Single-View 3D Hair Modeling With Clumping Optimization

Zhongsi Tang, Jiahao Geng, Yanlin Weng, Youyi Zheng, and Kun Zhou, *Fellow, IEEE*

***Abstract*—Deep learning advancements have enabled the generation of visually plausible hair geometry from a single image, but the results still do not meet the realism required for further applications (e.g., high quality hair rendering and simulation). One of the essential element that is missing in previous single-view hair reconstruction methods is the clumping effect of hair, which is influenced by scalp secretions and oils, and is a key ingredient for high-quality hair rendering and simulation. Inspired by common practices in industrial production which simulates realistic hair clumping by allowing artists to adjust clumping parameters, we aim to integrate these clumping effects into single-view hair reconstruction. We introduce a hierarchical hair representation that incorporates a clumping modifier into the guide hair and skinning-based hair expressions. This representation utilizes guide strands and skinning weights to express the basic geometric structure of the hair. The clumping modifier allows for the expression of more detailed and realistic clumping effects. Based on this representation, We design a fully differentiable framework integrating a neural measurement of clumping and a line-based rasterization renderer to iteratively solve guide strands positions and clumping parameters. Our method demonstrates superior performance both qualitatively and quantitatively compared to state-of-the-art techniques.**

***Index Terms*—Hair modeling, single-view reconstruction, differentiable rendering, contrastive learning.**

I. Introduction

REALISTIC hair modeling is essential for various applications. For an instance, in rendering of digital human portraits, high-quality hair models are needed to produce lifelike images. In dynamic simulations, including games, movies, and VR, both the initialization of high-quality hair models and the preservation of hairstyle characteristics throughout the simulation process are crucial.

Creating a realistic 3D hair is a complex endeavor, traditionally demanding weeks of meticulous effort from artists. Recently, advancements in computer graphics have introduced methods for reconstructing realistic hair from a single image, substantially easing the complexity of hairstyle modeling. Prior works like [31], [37] typically generate 3D volumetric hair representations from a single image using deep learning techniques, then grow hair strands from these volumes. However, these methods struggle with several issues, resulting in unsatisfactory rendering and simulation results. The most important point is that their hair modeling without hierarchical structure often fails to capture the intricate details and natural structure of hair, such as clumping effects caused by secretions and oils from the scalp, which detracts from the realism of the results. Second, due to the complex translucency and occlusion of hair, it is challenging to ensure consistency between the reconstructed result and the input image in single-image hair reconstruction.

Inspired by the industrial production tool Maya XGen, we introduce a hierarchical hair representation for single-image hair modeling. This representation incorporates a clumping modifier into the guide hair and skinning-based hair expressions. On the one hand, this representation utilizes guide strands and skinning weights to express the basic geometric structure of the hair. On the other hand, the clumping modifier allows for the expression of more detailed and realistic clumping effects, which are crucial for hairstyle realism. This hair representation not only allows us to express more detailed hair structures but also makes it easier to maintain hair characteristics during simulations.

To achieve a hairstyle consistent with the input image, we design a fully differentiable framework integrating a neural measurement of clumping and a line-based rasterization renderer to iteratively solve guide strands positions and clumping parameters. We find that optimizing both hair geometry and appearance properties simultaneously, as in existing works [12], [20], [24], often results in ambiguous outcomes and fails to reconstruct the geometry accurately. This is due to the excessive complexity of hair appearance modeling caused by the complex occlusion and semi-transparent of hair strands. To circumvent the complexities of hair appearance, we utilize geometric feature maps, such as orientation and depth fields, as intermediate representations. However, it is difficult to compare the input image with the geometric feature maps regarding clumping directly. Therefore, we leverage contrastive learning [2], a method capable of comparing multimodal similarity [18], to design clumpNet for measuring the similarity of clumping between geometric feature maps and the input image. Additionally, in order to computing derivative of rendering image with respect to the guide strands positions and clumping parameters, we design a line-based rasterization differentiable renderer to propagate the loss calculated by geometric

Received 13 September 2024; revised 27 February 2025; accepted 2 March 2025. Date of publication 20 March 2025; date of current version 5 September 2025. This work was supported in part by NSF China under Grant 62172357, Grant 62421003, and in part by XPLORER PRIZE. Recommended for acceptance by T. Wong. *(Corresponding author: Yanlin Weng.)*

Zhongsi Tang, Yanlin Weng, Youyi Zheng, and Kun Zhou are with the State Key Lab of CAD&CG, Zhejiang University, Zhejiang 310027, China (e-mail: zhongsitang@zju.edu.cn; weng@cad.zju.edu.cn; youyizheng@zju.edu.cn; kunzhou@zju.edu.cn).

Jiahao Geng is with Faceunity Inc., Hangzhou 310000, China (e-mail: jiahaogeng@outlook.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/TVCG.2025.3552919, provided by the authors.

Digital Object Identifier 10.1109/TVCG.2025.3552919

1077-2626 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:01:17 UTC from IEEE Xplore. Restrictions apply.

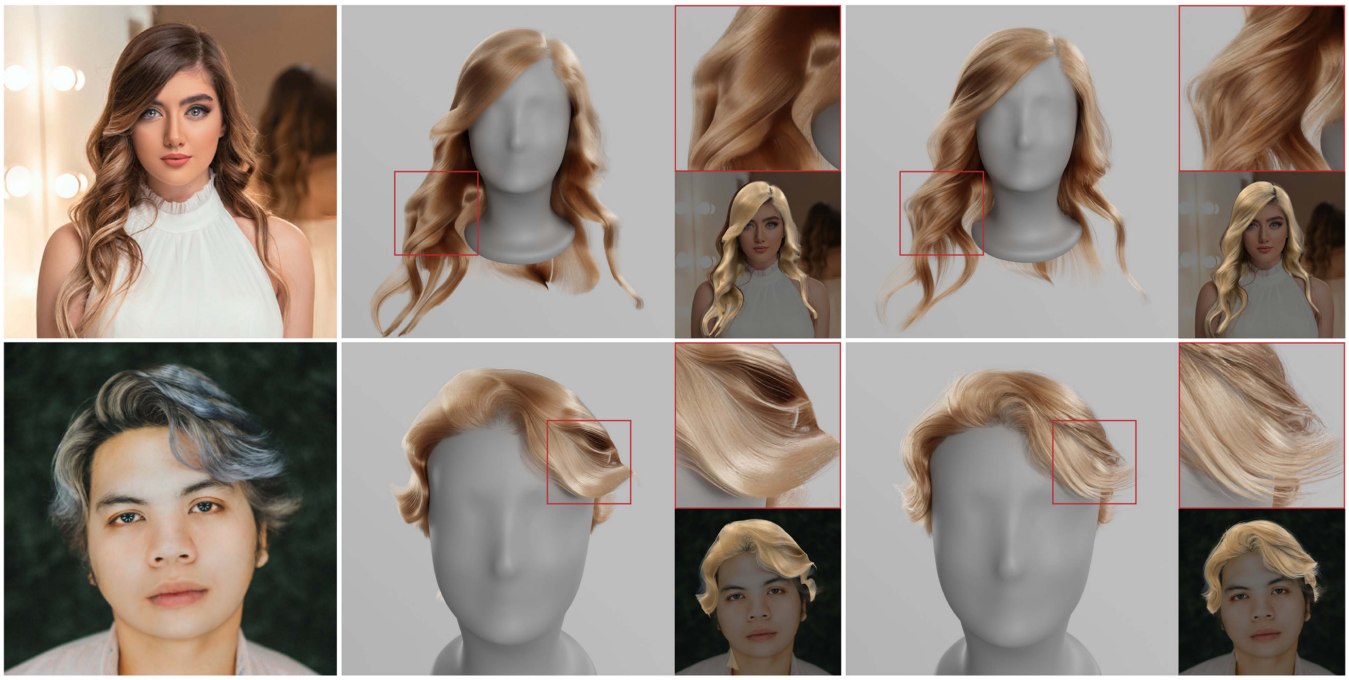


Fig. 1. Given an image, we obtain an initial hairstyle using existing methods and then transform it into our parametric hair representation combining guide strands and clumping modifier. We optimize the guide strands and clumping parameters using a differentiable renderer. This ensures that the resulting hairstyle's contour, growth direction and clumping characteristics match the input image. From left to right, the sequence includes input images, the results of the state-of-the-art method HairStep [37], and our results. In the right columns of HairStep and our results, from top to bottom, are the zoom-in and projection of the reconstructed hairstyle results shown in the left columns.

feature maps and clumpNet. Qualitatively and quantitatively comparative analyses confirm that our method outperforms state of the arts both in rendering and simulation (see Fig. 1 and Fig. 12).

In summary, the main contributions of our work are as follows:

- We introduce a novel parametric hair representation that consists of the guide hair and skinning parameters representing the basic geometry of the hair, and clumping parameters representing detailed hair structures. This representation facilitates more realistic outcomes through the refined optimization of model parameters and makes it easier to maintain hair characteristics during simulations.
- We devise a fully differentiable framework for single-view hair modeling and optimization, which is capable of generating higher quality hair conforming to the characteristics of the input image, compared to the prior art.
- We utilize contrastive learning to design a novel hair similarity module, termed clumpNet, which quantifies the clumping similarity between the input image and geometric feature maps.
- We introduce a line-based soft rasterization technique to establish a differentiable rendering optimization framework tailored explicitly for hair strands.

II. Related Work

1) Single-view Hair Reconstruction: Single-view 3D hair modeling remains a challenging yet vibrant research area within computer graphics and computer vision domains. Although multi-view approaches have achieved strand-accurate hair reconstructions using multiple images [9], [15], [20], their deployment is often restricted to controlled environments and complex hardware setups. Conversely, single-view methods offer practicality and have found broad applications in real-world scenarios. The pioneering single-view techniques relied predominantly on geometric priors or synthetic data to approximate target hairstyles [1], [7], requiring extensive datasets for retrieving and thus limiting their practical utility. The advent of deep learning has revolutionized this field, with deep neural networks becoming prevalent in hair modeling. HairNet [41] was the inaugural network to generate 3D hair strands from a 2D orientation map directly. Subsequent efforts by [22] and [35] employed VAE and GAN architectures to create 3D orientation volumes, respectively, facilitating the generation of hair strands guided by these volumes. To enhance local detail for high-quality modeling, NeuralHDHair [31] introduced a coarse-to-fine strategy to obtain high-resolution 3D orientation fields. Further advancing the field, HairStep [37] proposed a strand-aware and depth-enhanced hybrid representation, which inferred volumetric hair from directed 2D orientation maps and depth maps, addressing the significant domain gap between synthetic and real data in single-view 3D hair modeling. These advancements have notably increased the fidelity of reconstructions yet continue to face challenges in detail accuracy and alignment consistency with the input image. Our methodology for the first time incorporates a clumping parametric representation to simulate the

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:01:17 UTC from IEEE Xplore. Restrictions apply.

natural clumping effects of hair, which introduces rich, lifelike details not attainable in previous models. Moreover, leveraging differential rendering optimization ensures precise alignment of the reconstructed hair with the input image through re-projection consistency.

2) Hair Representation: In the task of hair reconstruction, human hair, which comprises approximately 100,000 strands, exhibits diverse geometric structures and complex occlusion relationships. Typically, parametric models are employed to compress the representation of hair, including voxel-based and strand-based methods. Volume-based methods [9], [22], [31], [32], [35], [37], [40] convert hair into 3D orientation and occupancy fields, subsequently training neural networks to map geometric feature images onto these voxel-based representations. Although these methods generally yield reasonable results, the voxel-based representation of hair is inherently constrained by the voxel resolution, resulting in excessively smooth output and lacking fine details. Alternatively, several studies [20], [24], [25], [39] have explored strand-based methods, transforming hair strands into neural representations. For example, Neural Strands [20] adopt the modulated sine network architecture [13] to map individual strands into a low-dimensional latent space, then utilize UV unwrapping to allocate strands to the corresponding pixels on the scalp. GroomGen [39] developed a hierarchical hair model, parameterizing the representation into three levels: single strands, sparse guide hairs, and dense complete hairs. GroomGen parameterized the strands in the frequency domain using the discrete Fourier transform (DFT), establishing a strand latent space based on this frequency representation to enhance the retention of strand-level details such as curliness. Diverging from these implicit strand representations, our approach introduces an explicit parameterization using guide strands and clumping modifiers, creating a model that is more intuitive and manageable, and better suited for optimization through differentiable rendering.

3) Differentiable Rendering: Differentiable rendering has become a proper technique in the reconstruction of 3D scenes. Extensive research has been conducted not only on implicit functions [14], [29] but also on methods utilizing explicit geometric primitives such as meshes [3], [10], [11], [16], [19], [26] and point clouds [21], [33]. SoftRas [11] initially proposed a generic method for differentiable rendering aimed at optimizing explicit meshes. Subsequent developments, such as Neural Haircut [24] and DR.hair [27], have adapted hair into meshes, utilizing SoftRas for differentiable rendering. Besides, Neural Strands [20] employed neural rendering to optimize hair geometry and appearance simultaneously. GaussianHair [12] and GroomCap [40] utilizes a representation of each hair strand as a sequence of connected cylindrical 3D Gaussian primitives to facilitate differentiable rendering. However, these methods optimize both geometry and materials, which may lead to suboptimal geometry results due to the complexity of the hair appearance. Addressing these limitations, we adopt SoftRas specifically for hair strands, introducing a line-based differentiable renderer that directly renders geometric information into 2D feature images. This method explicitly optimizes hair strand geometry and significantly outperforms previous approaches in robustness, processing speed, and memory efficiency.

4) Image Similarity: Assessing the similarity between rendered images and target images is crucial for optimizing hair via differentiable rendering. Traditional methods often rely on pixel-wise differences such as Mean Squared Error (MSE) or Structural Similarity Index Measure (SSIM), which are effective for basic discrepancies but may not capture more nuanced perceptual qualities [30]. Advancements have introduced feature-based comparisons utilizing deep learning models, such as perceptual loss [8], [36] that employ features extracted by pre-trained networks like VGG. These methods seek to capture the higher-level perceptual differences that pixel-wise metrics often overlook. Recent advancements have incorporated contrastive learning [2], [18] into image similarity assessment. Initially utilized in self-supervised learning paradigms, contrastive learning facilitates the learning of representations by contrasting pairs of similar and dissimilar data. This method not only facilitates comparisons between images and maps images and textual descriptions to a shared feature space [18], enabling multimodal comparisons. Inspired by such approaches, we integrate the principles of contrastive learning to assess the similarity in geometric structure between input images and reconstructed hairstyles. To achieve this, we develop a network that evaluates the clumping similarity between input images and their corresponding rendered feature maps.

III. Method

We introduce our fully differentiable pipeline illustrated in Fig. 2. Our process begins by transforming the initial hairstyle obtained from HairStep [37] into a parametric model (Section III-A) that considers clumping. Given the complex shading inherent to realistic hair, which challenges the optimization of geometry, we utilize a line-based differentiable renderer (Section III-B) to generate geometric feature maps of the parameterized hair, such as orientation and depth field. We leverage contrastive learning to address the issue of comparing clumping similarities between geometric feature maps and input images (Section III-C). By integrating these components, our pipeline offers a robust differential framework for optimizing hair parameters (Section III-D).

A. Hierarchical Hair Model

Inspired by industrial production tools, we propose a hierarchical parametric model that stratifies hair complexity into three distinct levels: sparse, dense, and local detail. At the sparse level, guide strands, denoted as G, establish the foundational growth pattern for the denser hair levels. Dense hair is then generated through weighted interpolation from these guide strands with the weights expressed as W. The local detail level is enhanced by the application of clumping modifiers, parameterized by α, which are employed to apply clumping effects within the dense hair. This parameterization process and its impact on hair modeling fidelity are illustrated in Fig. 3.

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:01:17 UTC from IEEE Xplore. Restrictions apply.

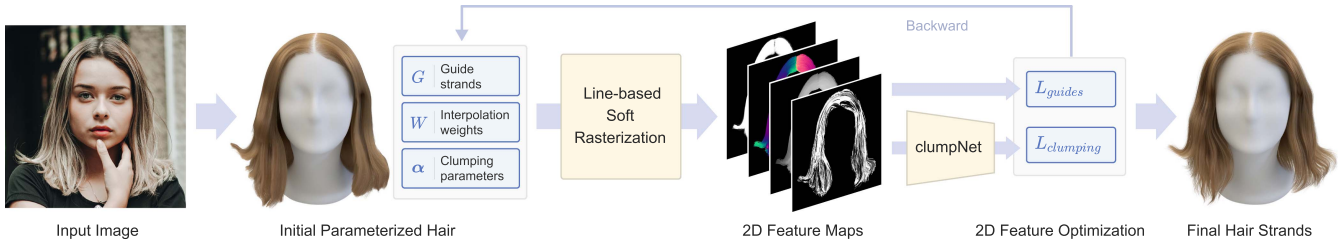


Fig. 2. Overall pipeline. We employ existing techniques to reconstruct the initial hair strands from the input image. Based on the initial values, a parameterized hair model is constructed, and the parameters of guide strands and modifiers are iteratively optimized through differentiable render to obtain results consistent with the input image features.

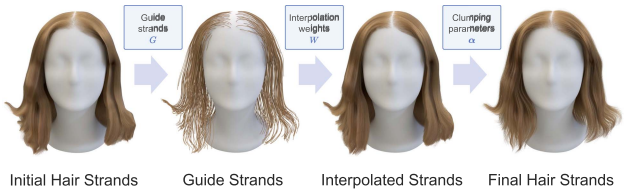


Fig. 3. Our hierarchical hair representation. The initial hair is represented by guide strands, interpolation weights, and the clumping modifier. After each parameter is optimized, the final hairstyle can be obtained.

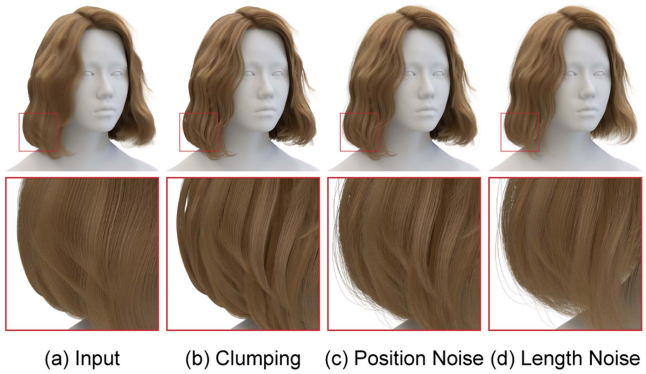


Fig. 4. The effects of clumping and noise within the clumping equation. From left to right: the original hairstyle, the hairstyle with added clumping, the hairstyle with added position noise, and the hairstyle with added length noise.

1) Guide Strands and Interpolation: Initially, for an original hair set $\{S_i|i=0,\ldots,N-1\}$ comprising N strands, we resample the number of vertices per strand to 32. We then apply the K-means algorithm to compute the N_g cluster centers, which serve as guide strands $\{G_j|j=0,\ldots,N_g-1\}$. Subsequently, we calculate the normalized inverse of the distance from each hair root to its nearest 32 guide strands, establishing these values as the initial interpolation weights W. Using these weights, dense hair strands are represented through a weighted interpolation based on proximity to the nearest guide strands.

$$S_i' = \sum_{j \in M} W_{i,j} \cdot G_j \tag{1}$$

Where M represents the set of 32 guide strands nearest to S_i. We optimize the guide strands and interpolation weights by minimizing the distance between each vertex of the interpolated hair and the initial hair, using the following energy term:

$$E_{geo} = \left(\frac{1}{N} \sum_{i=0}^{N-1} |S_i'(W,G) - S_i|^p \right)^{\frac{1}{p}}$$
$$\text{s.t.} \sum_{\mathrm{j}=0}^{\mathrm{N_g}-1} \mathrm{W_{i,j}} = 1, \mathrm{W_{i,j}} > 0. \tag{2}$$

Here, we employ the generalized mean (GeM) to compute the average difference across all hair strands, thereby reducing sensitivity to outliers. For more details, see supplementary materials. The parameter p in GeM is set to 2.7. Furthermore, we incorporate a laplacian smoothing term along with a regularization term designed to maintain uniform vertex spacing.

2) Clumping Modifiers: After obtaining the basic expression of the hair, we enhance it by adding clumping effects through a clumping modifier, which can be seen as an offset in the original hair strand. The visual effect of clumping can be seen in Fig. 4.

Initially, we apply the k-means algorithm to partition S' into N_c cluster centers, denoted by $\{C_i|i=0,\ldots,N_c-1\}$. Subsequently, we calculate the offset of each hair strand vertex relative to its corresponding cluster center:

$$D^c(S_i') = (S_i' - C_i) \odot \mathbf{s}_c \tag{3}$$

Considering the diverse scale of clumping between the roots and tips of real hair—with tips typically exhibiting tighter clumping than roots, we employ the vector $\mathbf{s}_c$ to represent the variation in clumping at different vertex along a hair strand. The vector $\mathbf{s}_c$ is of length n_s corresponding to the number of vertices in each hair strand.

We then add a random offset to the vertex of each hair strand in a direction that conforms to a 3D noise function, expressed as the following formula:

$$D^{np}(S') = \mathcal{N}(S') \odot \mathbf{s}_n \tag{4}$$

$\mathcal{N}$ represents a 3D noise function. Similar to $\mathbf{s}_c$, $\mathbf{s}_n$ is used to control the noise intensity at each vertex of a hair strand. In real hairstyles, the noise at the hair tips is often greater than at the roots. Therefore, we use an exponential function to approximate

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:01:17 UTC from IEEE Xplore. Restrictions apply.

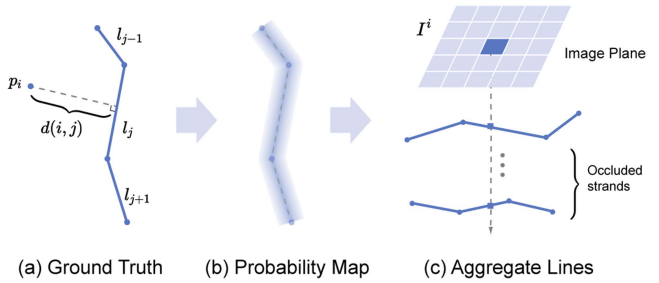


Fig. 5. Soft rasterization. (a) definition of pixel-to-line distance; (b) probability map for one strand; (c) final color is calculated by weighted sum of color of each line segment.

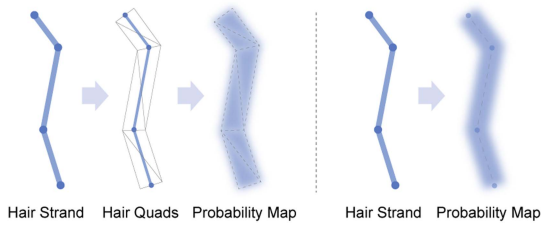


Fig. 6. Compare hair quads [24] (left) and ours (right). While the method of hair quads converts strands into mesh triangles, which may introduce artifacts, our method computes a probability map directly from the strands.

s_n and s_c.

$$\mathbf{s}(i) = \left(\frac{i}{n_s}\right)^t, i = 0, 1, \ldots, n_s - 1 \tag{5}$$

Where t controls the variation from the root to the tip of the hair. In $\mathbf{s}_c$, our t_c is fixed at 0.5, and in $\mathbf{s}_n$, our t_n is set at 0.2. Finally, we use $\boldsymbol{\alpha}$ to represent the clumping scale of each hair strand, and add length noise D^{nc} to each strand, which effectively randomizes the trimming of each hair strand's length.

$$S^m(\boldsymbol{\alpha}) = D^{nc}(S' + \boldsymbol{\alpha} \cdot D^c(S') + D^{np}(S')) \tag{6}$$

After establishing the hierarchical parametric hair model, we manipulate the morphology of the hairstyle by changing the positions of the guide strands G and adjusting the clumping scale of the hairstyle through the $\boldsymbol{\alpha}$ parameter. This approach enables the representation of diverse hairstyles observed in the real world. In subsequent sections of this paper, we iteratively optimize these parameters to achieve a hairstyle that aligns with the contour and clumping features of the input image.

B. Line-Based Soft Rasterizer

Differentiable rasterization is an important approach for achieving differentiable rendering, and many works [11], [24] have advanced its development. However, these methods, which are based on quads or triangles, are not the most effective for rendering line structures such as hair strands. Therefore, building on the success of the soft rasterization method [11], [19] developed for mesh structures, we introduce a line-based soft rasterizer which enables the optimization of hair strands through differentiable rendering, as illustrated in Fig. 5.

Our line-based soft rasterizer is designed to render hair strands as line segments. The full rendering pipeline is illustrated in Fig. 5. For each pixel p_i on the image plane, we define the probability of influence by each line segment l_j as follows:

$$P_j^i = 2 \cdot \text{sigmoid}\left(-\frac{d^2(i,j)}{\sigma}\right) \tag{7}$$

where σ controls the blurriness of the line segment edges. and $d(i,j)$ represents the minimum distance from p_i to l_j. Similar to [19], we constrain pixels to be influenced by only the nearest K line segments along the z-axis to improve efficiency. We set σ as 1×10^{-5} and K as 128 unless otherwise specified.

For each line segment l_j, we define its color C_j^i at pixel p_i on the image plane by interpolating vertex color using distance weights. The final color I_c^i and alpha value I_α^i of each pixel are calculated by the following formulas:

$$I_c^i = \sum_j \frac{P_j^i \exp(z_j^i/\gamma)}{\sum_k P_k^i \exp(z_k^i/\gamma)} C_j^i \tag{8}$$

$$I_\alpha^i = \tanh\left(\frac{1}{n}\sum_j^n P_j^i \cdot \eta\right) \tag{9}$$

Here, z_j^i denotes the reciprocal of the normalized depth of the 3D points on l_j, whose 2D projection is p_i. $\gamma = 1 \times 10^{-5}$ is used to control the depth weight of each line segment. Since the diameter of hair is often less than 1 pixel, we use (9) to calculate the alpha channel with soft edges, and $\eta = 16$ is used to control the contrast of the alpha channel edges at the pixel level.

Compared to previous approaches [24] that convert hair strands into hair quads [34], which may introduce artifacts into the probability map due to the presence of triangle edges when rendering the interior of individual hair strands, our method processes line segments directly. This approach ensures that pixels close to the line segment are more significantly influenced, while those further away are less affected. See close-ups in Fig. 6. Moreover, representing each segment with two triangular patches significantly increases the rendering overhead. Our method is more efficient and yields more accurate results. For a more quantitative comparison of performance and optimization results, see Section IV-C1.

C. Clumping Similarity Network (clumpNet)

Hair clumping is a phenomenon where individual strands aggregate into larger bundles due to physical forces such as friction and static attraction, which are typically modeled in simulations to achieve realistic hair dynamics [23]. However, the clumping effect represents a feature of local structural aggregation rather than a characteristic manifested by individual pixels. Currently, there is no direct algorithm available for comparing the clumping similarity between two images. Drawing inspiration from contrastive learning [2], [18] that assesses multimodal similarity, we introduce clumpNet. This specialized network is designed to quantify the similarity in clumping features between the input image and the rendered hair strands.

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:01:17 UTC from IEEE Xplore. Restrictions apply.

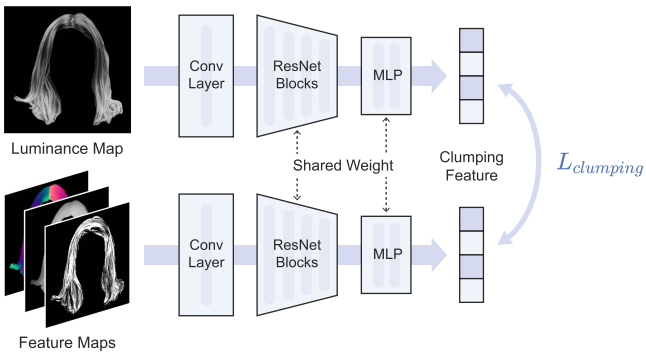


Fig. 7. Structure of clumpNet. The clumping similarity between the input image and the rendered image can be obtained by clumpNet.



Fig. 8. Example of training data. Each row represents a hairstyle, and different columns within a row show the results of that hairstyle under varying clumping parameters.

The inputs to clumpNet is divided into two groups: one consists of a luminance map I^l extract from the real image, and the other comprises feature images F, which include a depth map F^d, an orientation field F^o, and a clumping strength parameter map F^α. The luminance map is computed using the luminance formula [28]: $\mathrm{Y} = 0.299\times \mathrm{R} + 0.587\times \mathrm{G} + 0.114\times \mathrm{B}$. The feature maps are rendered using our differentiable renderer based on the reconstructed hair strands. Our purpose is to extract features related to clumping from I and F, and to measure the similarity of these features between them. Since I and F are not the same type of data but contain common attributes, we use contrastive learning to train clumpNet.

Our network is based on ResNet18, and its structure is shown in Fig. 7. We adapted the ResNet architecture to individually process the input luminance map and feature map through two distinct single-layer convolutional layers. Subsequently, we utilize the same ResNet blocks to extract image features. The process culminates with an MLP that projects these features into a 128-dimensional clumping feature space.

1) Dataset: For our clumpNet dataset construction, we employ synthetic data as depicted in Fig. 8. We start by generating 3D hair strands from the HiSa dataset using HairStep [37] and subsequently transform these into our parameterized hair model, as described in (2). This process results in 1,250 basic hairstyles. To enhance these styles, we randomly adjust the α parameter and apply (6) to introduce a variety of clumping effects, producing approximately 18,750 clumped hairstyles. These hairstyles are then rendered under diverse lighting conditions to simulate realistic hairstyle images. Additionally, we project the hairstyle's orientation, depth, and clumping parameters using a camera to generate orientation maps, depth maps, and clumping strength maps. It is crucial to note that the clumpNet is designed to assess the similarity in clumping between the input image I and the feature images F, rather than to validate the fidelity of the feature image structure. Thus, the diversity of hairstyles and clumping effects is more essential than the realism of the data. The efficacy of training with synthetic data has been substantiated through both qualitative and quantitative evaluations, see Section IV-C2.

During the training process, we randomly adjust the brightness and contrast of I to augment color variations. Additionally, we apply random translations, rotations, and flips to both I and F to prevent the network from learning only contour information and to enhance its generalization capabilities. Moreover, we manually label the clumping parameters from real hairstyle images to bridge the domain gap between our synthetic dataset and real-world data.

2) Loss Function: During the training process, our loss function utilizes the modified log softmax function as follows:

$$L = \frac{1}{N}\sum_{i=0}^{N} \log \frac{\exp(y_{i,i})}{\sum_{j=0}^{N} \exp(y_{i,j}\cdot m_{i,j})} \tag{10}$$

In this context, N represents the batch size. Within a batch, all samples except the corresponding I and F are considered negative samples. y represents the similarity, and the calculation formula is as follows:

$$y_{i,j} = \cos(\mathrm{clumpNet}(I_i), \mathrm{clumpNet}(F_j)) \tag{11}$$

Inspired by Arcface [5], we incorporate a penalty term $m_{i,j}$ to intensify the distinction between positive and negative samples.

$$m_{i,j} = 1 + (\mathrm{mean}(F_i^\alpha) - \mathrm{mean}(F_j^\alpha))^2 \tag{12}$$

D. Optimization

By integrating all the components discussed, we establish a comprehensive differentialable optimization framework for hair modeling. Initially, we generate the hair model using parameterized representations to render a series of feature maps, specifically silhouette, orientation, depth, and clumping maps. These maps facilitate a comparison with input images to refine the hair model parameters. The optimization process is bifurcated into two main components: optimization of guide strands and optimization of clumping parameters.

1) Guide Strands Optimization This phase aims to align the hair strands with the input contours, orientation fields, and depth fields. We define the energy functions for these features as follows:

$$\begin{aligned} E_{silh} &= \|\hat{I}_{silh} - I_{silh}\|_2 \\ E_{orien} &= 1 - \cos(\hat{I}_{orien}, I_{orien}) \\ E_{depth} &= 1 - \cos(\nabla\hat{I}_{depth}, \nabla I_{depth}) \\ E_{map} &= E_{silh} + E_{orien} + E_{depth} \end{aligned} \tag{13}$$

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:01:17 UTC from IEEE Xplore. Restrictions apply.

Here, I represents the reference map, $\hat{I}$ is the rendered map, and $\cos(\cdot)$ denotes the cosine similarity function. Since the reconstruction relies exclusively on single-view images, additional constraints are required to maintain modifications within acceptable deviations from the original geometry. One such constraint involves comparing the optimized guide strands with their initial states within the Laplacian space.

$$E_{geo} = \|\Delta\hat{G} - \Delta G\|_2 \tag{14}$$

Additionally, we incorporate a smoothness term to maintain the uniformity and continuity of the guide strands:

$$E_{smooth} = \|\Delta^2\hat{G}\|_2 \tag{15}$$

To account for the natural growth direction of hair, we align the tangent at the hair root to be parallel to the scalp's normal. This alignment strategy not only enhances the realism of hair direction but also effectively prevents geometric penetration of the scalp during the optimization process.

$$E_{root} = 1 - \cos(\mathbf{r}_{tangent}, \mathbf{r}_{normal}) \tag{16}$$

Here, $\mathbf{r}_{\text{tangent}}$ and $\mathbf{r}_{\text{normal}}$ refer to the tangent at the hair roots and the normal directions of the scalp, respectively. The final energy function for guide strand optimization is expressed as:

$$E_{guide} = E_{map} + \lambda_1 \cdot E_{geo} + \lambda_2 \cdot E_{root} + \lambda_3 \cdot E_{smooth} \tag{17}$$

The coefficients are set as $\lambda_1 = 1 \times 10^4$, $\lambda_2 = 10$, and $\lambda_3 = 1$ for all experimental setups described in this paper.

2) Clumping Parameter Optimization: The other part of the energy function primarily utilizes clumpNet to measure the clumping similarity between the input image I and the feature map F

$$E_{clumping} = ||\text{clumpNet}(I) - \text{clumpNet}(F)||_2. \tag{18}$$

3) Optimization Pipeline: During the solving process, We first search for a global clumping parameter to use as the initial optimization parameter. Following this, we iteratively optimize the guide strand positions and clumping parameters as delineated in (17) and (18). To effectively minimize the energy functions defined in these equations, we use a gradient-based optimization method with the Adam optimizer. Details of the optimization process are outlined in Algorithm 1.

Algorithm 1: Hair Optimization Process.

Input: initial hair strands S, reference image I_{ref}, epoch $= 500$
Output: optimized hair strands S^m
$F_{ref} \leftarrow \text{HairStep}(I_{ref})$
$G \leftarrow \text{Reparameterize}(S)$
$\boldsymbol{\alpha} \leftarrow \text{SearchForBestParam}(S)$
for $i \leftarrow 1$ to epoch **do**
  $S^m \leftarrow \text{EvaluateHairStrands}(G, \boldsymbol{\alpha})$
  $F_{render} \leftarrow \text{SoftRasterize}(S^m)$
  $L_{guide} \leftarrow \text{CalcGuideLoss}(F_{ref}, F_{render})$
  $G \leftarrow \text{Adam}(L_{guide})$
  if $i > 30$ **then**
    $L_{clump} \leftarrow \text{CalcClumpingLoss}(F_{ref}, F_{render})$
    $\boldsymbol{\alpha} \leftarrow \text{Adam}(L_{clump})$
  end if
end for

IV. Experimental Results

We conduct extensive experiments to validate the effectiveness of our hair reconstruction pipeline. In Section IV-A, we describe the training procedure and system runtime. In Section IV-B, we present both qualitative and quantitative results, along with comparisons to state-of-the-art methods. Finally, in Section IV-C, we provide ablation studies to further analyze our approach.

A. Training and Runtime

For the implementation of our algorithm, we employ Python on a system equipped with an Intel Core i9-13900 K CPU and a NVIDIA GeForce RTX 4090 GPU. The training of clumpNet utilizes a synthetic dataset comprising 18,750 pairs of images, each of resolution 512x512, as detailed in Section III-C. The Adam optimizer is employed with a starting learning rate of 0.0003, and a cosine annealing scheduler is used to modulate the learning rate. Training parameters include a batch size of 128, a temperature setting of 0.07, and a weight decay of 0.0001. The network is trained over 1500 epochs until convergence, taking approximately 40 hours.

After training, we use the clumpNet to infer the clumping similarity in our optimization pipeline. The hair model is parameterized by 512 guide strands and 80,000 hair strands. Optimization of the guide strands is conducted using the Adam optimizer with a learning rate of 0.0001, while clumping parameters are optimized using the same optimizer set at a learning rate of 0.01. The entire optimization process, utilizing PyTorch [17] for the differentiable pipeline, is set to execute over 500 iterations and successfully completed in about 4 minutes.

B. Qualitative and Quantitative Results

We evaluate our algorithm on challenging internet portraits obtained from Unsplash and Pexels (Fig. 9). The reprojection results demonstrate that for various hairstyles, the contours and orientations of our reconstruction align well with the input images. The rendering results further show that by accounting for hair clumping, our outcomes appear more natural compared to the baseline [37].

To underscore the enhanced capabilities of our methodology, we conduct comprehensive qualitative and quantitative assessments comparing it to leading single-view hair reconstruction methods, such as HairNet [41], NeuralHDHair [31], and HairStep [37]. Fig. 10 illustrates that, while these methods are advanced, they often fall short in producing realistic renderings due to their limited ability to represent clumping effects. In contrast, our specially designed clumpNet significantly enhances the realism of the reconstructed hairstyles. A notable limitation of conventional techniques is their absence of a differentiable rendering framework, which restricts their capability to utilize

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:01:17 UTC from IEEE Xplore. Restrictions apply.

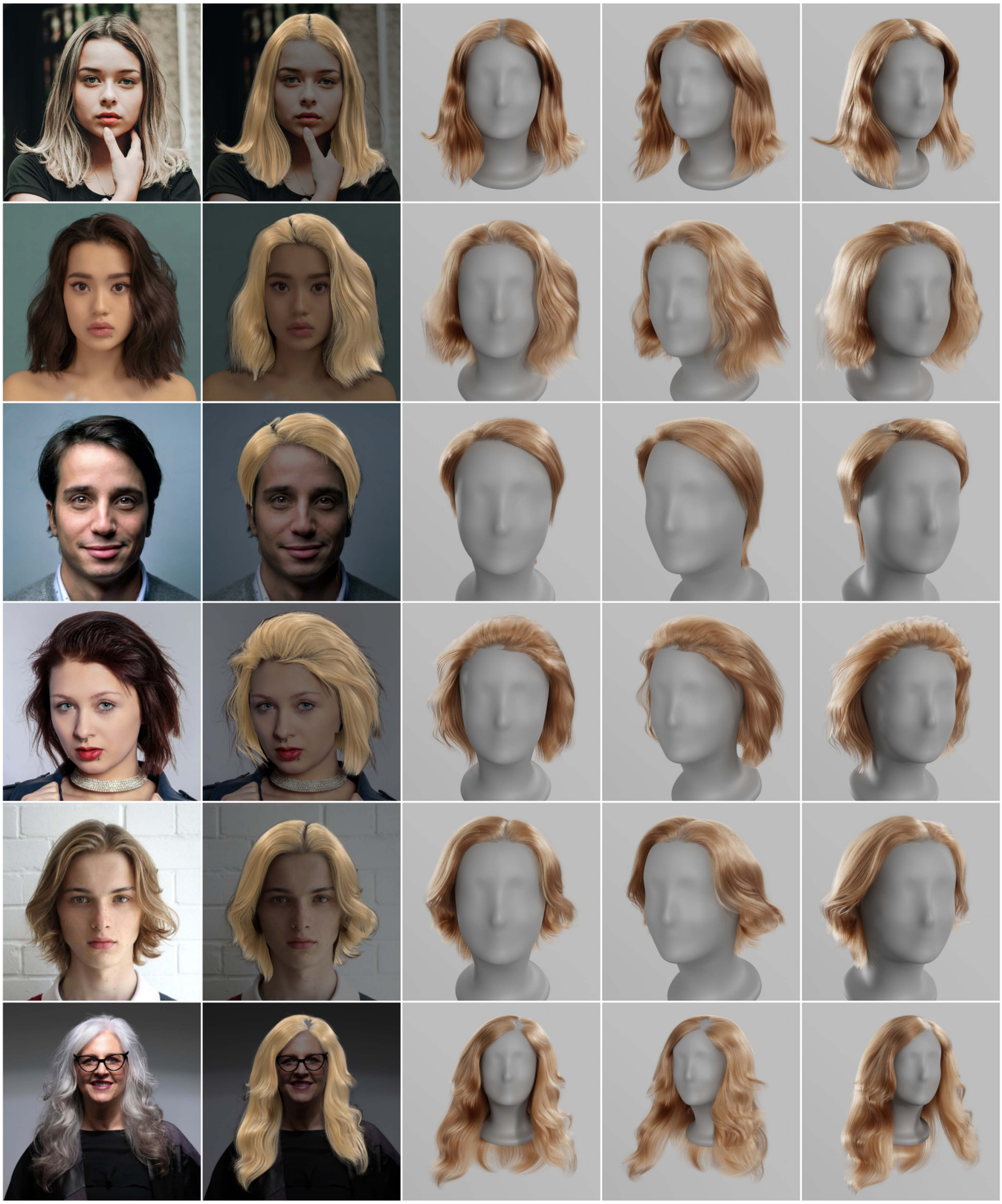


Fig. 9. Qualitative results. From left to right, the sequence includes the input image, the reprojection result, the front view of the reconstructed hairstyle, and the side views.

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:01:17 UTC from IEEE Xplore. Restrictions apply.

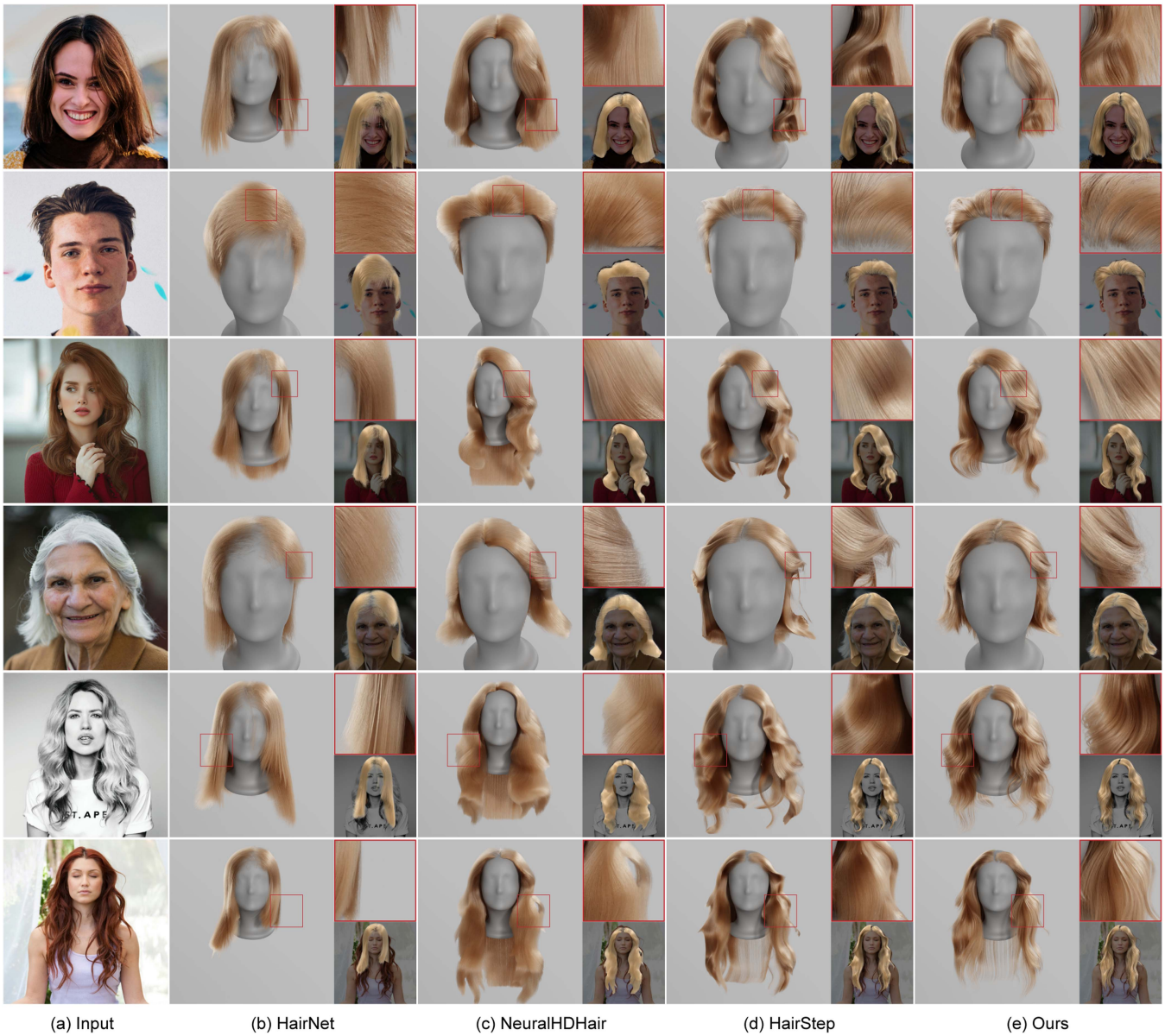


Fig. 10. Qualitative comparisons. From left to right show the results from HairNet [41], NeuralHDHair [31], HairStep [37], and ours.

input image for guiding hairstyle reconstructions. Our method overcomes this limitation by integrating a differentiable framework with soft rasterization. This integration allows for refined optimization of both guide strands and clumping parameters, aligning the reconstructed hairstyles more closely with the input image in terms of contours, direction, and clumping features.

For a more comprehensive evaluation, we compare our method with the recent work UniHair [38], which represents hairstyle as 3D Gaussians (3DGS) with joint geometry-appearance optimization, as shown in Fig. 11. While UniHair's reconstruction produces a point cloud that approximates the overall morphology of reference hairstyles, it fails to capture the underlying strand-level topology. In contrast, Our method can reconstruct the complete hair strand geometry, which is essential for downstream applications such as relighting and simulation. Qualitative evaluation of novel view rendering reveals that although UniHair's novel views retain photometric similarity to the input reference, they exhibit noticeable rendering artifacts such as blurring and structural inconsistencies compared to our strand-level geometric reconstruction.

Quantitative assessments further substantiate the advantages of our method. Following the evaluation method established by HairStep [37], we calculate errors in the 2D orientation field and relative depth field, termed HairSale and HairRida, respectively, between our reconstructed hairstyles and the corresponding ground truth annotations (see Table I). The results indicate that our method significantly outperforms existing techniques across all 2D metrics, underscoring the efficacy of our differential rendering framework. Additionally, we evaluate the accuracy of the 3D geometry of the reconstructed hairstyles using a synthetic

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:01:17 UTC from IEEE Xplore. Restrictions apply.

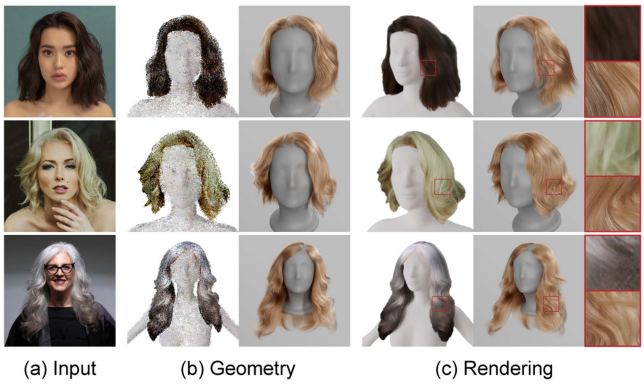


Fig. 11. Comparisons between our method and UniHair [38]: (a) Input reference image. (b) Comparison of reconstruct geometry: left shows UniHair's point cloud, right shows ours hair strands. (c) Comparison of novel-view renderings: left shows UniHair, center shows our method, and right highlights close-up view.

TABLE I
QUANTITATIVE COMPARISONS ON HAIRSALE AND HAIRRIDA OF HAIRNET [41], NEURALHDHAIR [31], HAIRSTEP [37] AND OURS

Method	IoU.↑	HairSale↓	HairRida↑
HairNet	33.51%	36.38	66.26%
NeuralHDHair	79.76%	18.29	73.83%
HairStep	77.19%	13.96	79.31%
Ours	**86.30%**	**11.70**	**82.50%**

TABLE II
QUANTITATIVE COMPARISONS ON SYNTHETIC DATASET OF HAIRNET [41], NEURALHDHAIR [31], HAIRSTEP [37] AND OURS

Method	Orien. err.↓	Occ. acc.↑
HairNet	0.1695	50.53%
NeuralHDHair	0.1176	55.07%
HairStep	0.1137	67.26%
Ours	**0.1127**	**69.46%**

dataset (see Table II). Despite the primary optimization of hair strands based on 2D inputs, our approach also yielded improvements in the metrics for 3D reconstruction, demonstrating the robustness and versatility of our methodology in enhancing both 2D and 3D aspects of hair modeling.

Our approach not only enables more accurate reconstruction of hair geometry but also achieves superior simulation results. Leveraging a coarse-to-fine hierarchical structure, we simulate the guide strands using Blender's simulation system [4]. The complete hair strands are subsequently generated via our parametric model. Since prior methods [37] did not incorporate guide information, we use two simulation methods for a fair comparison. First, we reference the methodology employed by Unreal Engine [6]. We utilize the K-means algorithm to calculate 512 guide strands, matching our count for simulation. The offsets for the hair strands in each frame are linearly interpolated based on distance-related weights. Additionally, we directly simulate all the hair strands. The simulation results are shown in Fig. 12, demonstrating that previous methods using guide simulation often produced artifacts, whereas simulating all strands is very

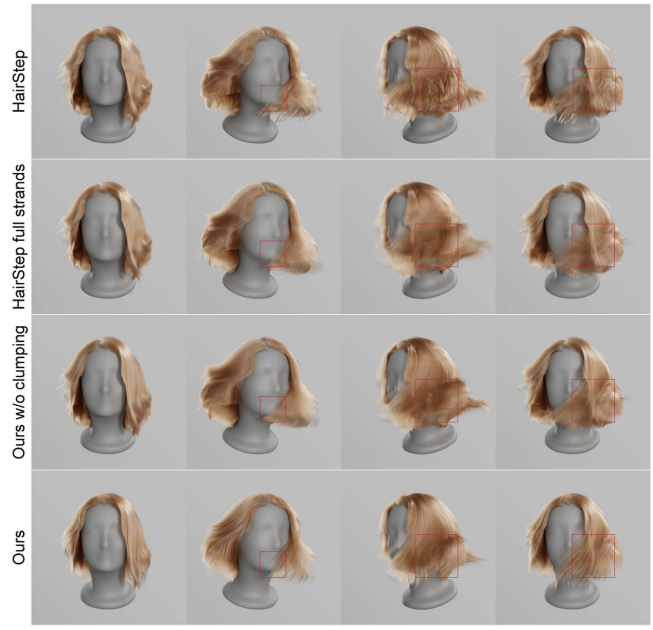


Fig. 12. Comparison of hair simulation results between HairStep [37] and our method. From top to bottom are HairStep with guide simulation, HairStep with full hair simulation, our model without clumping and our full results. Note that the simulation results of HairStep with guide simulation contain many artifacts, while our results are more robust and maintain the clumping effects.

TABLE III
QUANTITATIVE COMPARISIONS BETWEEN HAIR QUADS [24] AND OURS

	IoU.↑	Orien. Err.↓	Time↓	Memory↓
Hair Quads	0.9182	0.0067	402.01 s	7277 M
Ours	0.9535	0.0034	150.38 s	4355 M

slow. In contrast, our approach achieves more robust simulation results, maintaining the clumping effects of the hair and significantly enhancing the realism of hair simulations.

C. Ablation Study

To better study the effect of each design in our framework, we perform ablation studies on the parameterization, rasterization, clumpNet inputs, and optimization energy terms.

1) Evaluation on Soft Rasterizer: In this comparision, we render a parametric hair model using both Hair Quads [24] and our proposed method to generate silhouette maps and orientation maps. To ensure comparability, identical parameters are maintained across both rendering techniques while assessing their performance and optimization results. To mitigate extraneous influences, the computation of the energy term is restricted to the L2 distance between the silhouette map and the orientation map, augmented by original geometric constraints. The results of these experiments are detailed in Table III. The Intersection over Union (IoU) quantifies the contour discrepancies between the rendered outputs and the reference, whereas the Orientation Error (Orien. Err.) evaluates the accuracy of the orientation field in the optimized renderings, employing the methodology described in HairSale [37]. Our findings reveal that our approach,

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:01:17 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
QUANTITATIVE EVALUATION ON INPUTS OF SNET

	Clumping	+Depth	+Orien.
clumping err.↓	0.0649	0.0616	**0.0583**

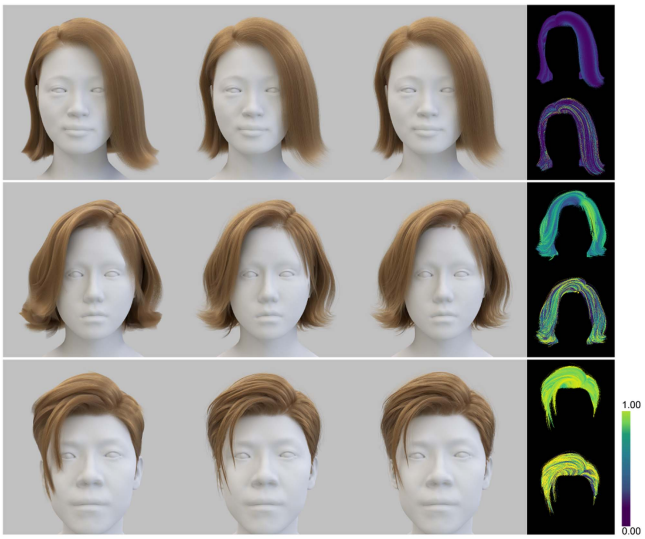


Fig. 13. Validation of clumping parameters on synthetic data. (a) Initial hairstyle. (b) Input image. (c) Optimized results. (d) Clumping parameter color map (from top to bottom: Ground Truth and optimized results respectively).

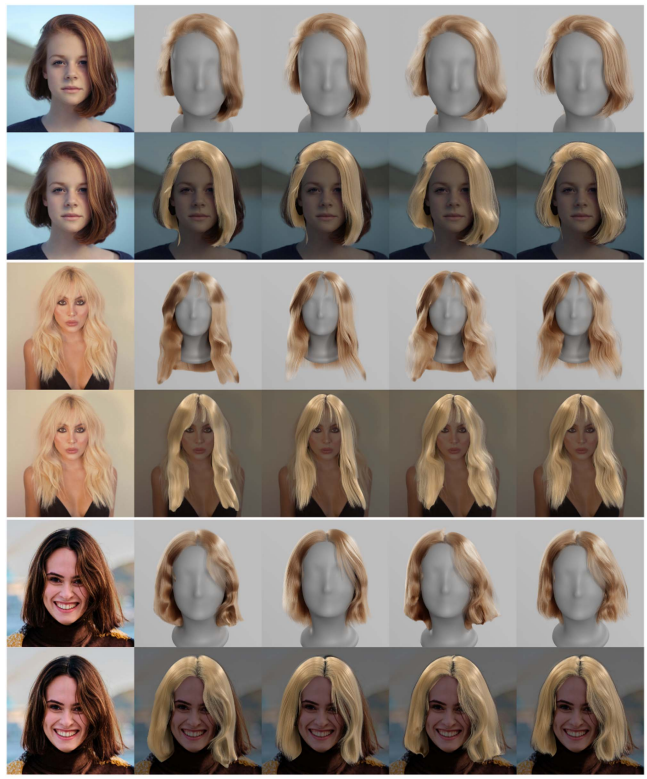


Fig. 14. Qualitative results of guide strands and clumping optimization. From left to right, the sequence includes the input image, the original hairstyle, the result from optimizing only the guide strands, the result from optimizing only the clumping parameters, and our final result.

leveraging native line segments for soft rasterization, significantly surpasses Hair Quads in terms of optimization outcomes. Furthermore, our method achieves a 40% reduction in VRAM usage and a 63% decrease in optimization time.

2) Evaluation on clumpNet: We primarily conduct a detailed evaluation of the inputs to clumpNet, as referenced in Table IV. By incorporating depth fields and orientation fields into the clumping feature map, we are able to further reduce errors in clumping.

We conducted a qualitative analysis of the accuracy of the optimized clumping parameters using synthetic data, with results shown in Fig. 13. For the initial hairstyle (Fig. 13(a)), we manually generated a hairstyle with clumping effects. The rendered result (Fig. 13(b)) was then used as the input reference for clumpNet. Next, we applied our optimization framework to solve for the clumping parameters of each hair strand based on the initial hairstyle, with the optimization results presented in Fig. 13(c). In order to clearly compare the clumping parameters, the reference and optimized clumping parameters are visualized in Fig. 13(d). The experimental results demonstrate that our optimization framework achieves accurate alignment with the reference data across different clumping scales. For additional visualizations of clumping parameters, see the supplementary materials.

3) Evaluation on Optimization: We first evaluate the impact of optimizing guide strands and clumping parameters, which can be seen in Fig. 14. Optimizing guide strands alone, although it produces accurate hair contours, lacks naturalness and still shows differences from the hairstyle characteristics of the input image. Optimizing only the clumping parameters ensures realism, but there are discrepancies between the contour and the input image. In contrast, our results not only maintain consistency in contour and direction with the input image, but the hair's clumping characteristics are also more similar to those in the input image. In the process of hair simulation, our method can keep the clumping effect of hairstyle, which makes the result of hair simulation more realistic and natural (Fig. 12).

We also visualize the impact of each component in the guide strands loss on the optimizated results (Fig. 15). In our experiments, it is noted that reliance solely on the feature map error E_{map} for optimization led to considerable noise in the results. To mitigate this, the original geometry is introduced as a constraint E_{geo}; however, this adjustment do not sufficiently smooth some local areas. Subsequently, smoothness constraints E_{smooth} are incorporated, addressing these irregularities, though they introduce a new challenge of penetration at the top of the head. Ultimately, the integration of root constraints E_{root} prove effective in significantly reducing the issues related to the intersection of the scalp and hairstyle.

D. User Study

To further evaluate the quality of our method, we also conduct a user study to gather feedback on the realism and accuracy of the images produced. For this study, we randomly select 10

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:01:17 UTC from IEEE Xplore. Restrictions apply.

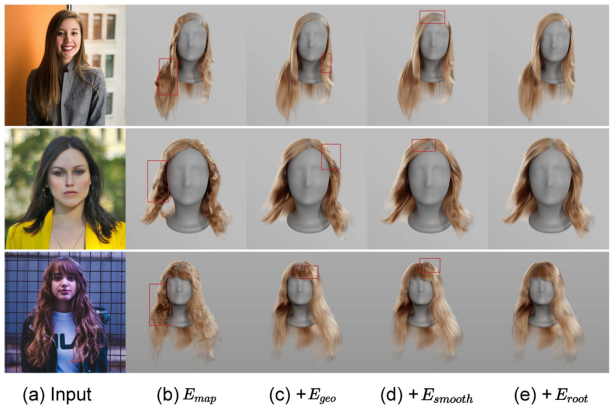


Fig. 15. Qualitative results of outline energy. From left to right: input image, result optimized with E_{map}, result after adding E_{geo}, further refined by adding E_{smooth}, and the final result after adding E_{root}.

TABLE V
USER STUDY ON NEURALHDHAIR [31], HAIRSTEP [37] AND OURS

Method	Similarity↑	Realism↑
NeuralHDHair	20.8%	20.8%
HairStep	31.6%	30.8%
Ours	**47.6**%	**48.4**%

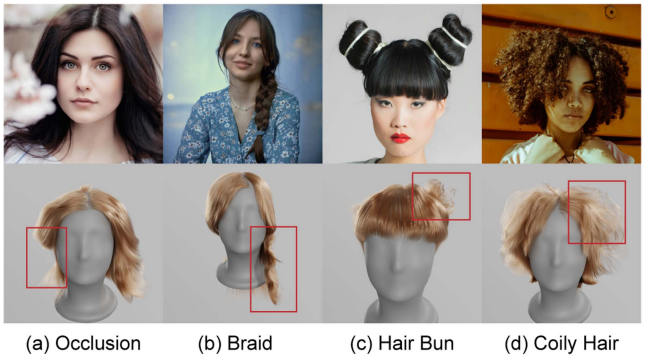


Fig. 16. Limitations. Our method still struggles with occluded hair regions and highly complex hairstyles, such as braids, buns, and coily hair.

images from our test dataset, ensuring a variety of subjects and styles to evaluate the performance of our approach effectively. These images are processed using our method and several other competitive methods for comparison. We invited 110 participants to take part in this study, asking them to rate the images based on their realism and resemblance to the original input image. Approximately half of the participants perceived the images generated by our method as the most realistic and most closely aligned with the input images. This indicates a significant preference for our approach over others, demonstrating its ability to produce high-quality and realistic image transformations. The results and statistical analysis of the user ratings are presented Table V.

E. Limitations

Our method exhibits inherent limitations rooted in its image-based strand reconstruction paradigm. As shown in Fig. 16(a), occluded hair regions remain fundamentally unrecoverable from single-view inputs. Furthermore, the framework demonstrates constrained capability in modeling topologically complex hair structures including braids, hair buds, and high-frequency curls (Fig. 16(c) and (d) - a recognized open challenge in single-view hair modeling. These technical boundaries will be systematically investigated in our future research trajectory.

V. CONCLUSION

In conclusion, our study advances the realistic reconstruction of 3D hair from single images by introducing a novel hierarchical hair representation integrated with contrastive learning and line-rasterized differentiable rendering techniques. Our approach leverages a clumping modifier that significantly enhances alignment and detail fidelity over previous methods. This improvement is quantifiable through our clumpNet, which measures clumping similarity effectively. Additionally, we optimize the hair model parameters using a line-based soft rasterization within a differentiable rendering framework, further refining the quality and accuracy of our results. Our method improves upon existing methods in hair reconstruction and simulation, and show superior qualitative and quantitative performance. It has potential impacts across further applications such as virtual reality and digital film-making. Future work will focus on refining these techniques and expanding their application to a wider range of hair types and interactive environments. Additionally, we will also attempt to further improve the quality of hairstyle reconstruction in unseen areas of a single image by using more complex datasets or a multi-view framework.

REFERENCES

[1] M. Chai, T. Shao, H. Wu, Y. Weng, and K. Zhou, "Autohair: Fully automatic hair modeling from a single image," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 116:1–116:12, 2016.
[2] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.
[3] F. Cole, K. Genova, A. Sud, D. Vlasic, and Z. Zhang, "Differentiable surface rendering via non-differentiable sampling," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6088–6097.
[4] B. O. Community, *Blender - a 3D Modelling and Rendering Package*. Amsterdam, Netherlands: Blender Foundation, 2018.
[5] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4690–4699.
[6] Epic Games, "Unreal engin," 2024.
[7] L. Hu, C. Ma, L. Luo, and H. Li, "Single-view hair modeling using a hairstyle database," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 125:1–125:9, 2015.
[8] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. 14th Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, Springer, 2016, pp. 694–711.
[9] Z. Kuang, Y. Chen, H. Fu, K. Zhou, and Y. Zheng, "Deepmvshair: Deep hair modeling from sparse views," in *Proc. SIGGRAPH Asia*, 2022, pp. 10:1–10:8.
[10] S. Laine, J. Hellsten, T. Karras, Y. Seol, J. Lehtinen, and T. Aila, "Modular primitives for high-performance differentiable rendering," *ACM Trans. Graph.*, vol. 39, no. 6, pp. 1–14, 2020.
[11] S. Liu, T. Li, W. Chen, and H. Li, "Soft rasterizer: A differentiable renderer for image-based 3D reasoning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7708–7717.
[12] H. Luo et al., "GaussianHair: Hair modeling and rendering with light-aware gaussians," 2024, *arXiv:2402.10483*.

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:01:17 UTC from IEEE Xplore. Restrictions apply.

[13] I. Mehta, M. Gharbi, C. Barnes, E. Shechtman, R. Ramamoorthi, and M. Chandraker, "Modulated periodic activations for generalizable local functional representations," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 14194–14203.
[14] B. Mildenhall, P.P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *Proc. Int. Conf. Comput. Vis.*, 2020, pp. 405–421.
[15] G. Nam, C. Wu, M. H. Kim, and Y. Sheikh, "Strand-accurate multi-view hair capture," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 155–164.
[16] B. Nicolet, A. Jacobson, and W. Jakob, "Large steps in inverse rendering of geometry," *ACM Trans. Graph*, vol. 40, no. 6, pp. 1–13, Dec. 2021.
[17] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.
[18] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.
[19] N. Ravi et al., "Accelerating 3D deep learning with pytorch3D," 2020, *arXiv: 2007.08501*.
[20] R. A. Rosu, S. Saito, Z. Wang, C. Wu, S. Behnke, and G. Nam, "Neural strands: Learning hair geometry and appearance from multi-view images," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 73–89.
[21] D. Rückert, L. Franke, and M. Stamminger, "ADOP: Approximate differentiable one-pixel point rendering," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 1–14, 2022.
[22] S. Saito, L. Hu, C. Ma, H. Ibayashi, L. Luo, and H. Li, "3D hair synthesis using volumetric variational autoencoders," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–12, 2018.
[23] A. Selle, M. Lentine, and R. Fedkiw, "A mass spring model for hair simulation," in *Proc. ACM SIGGRAPH 2008 Papers*, 2008, pp. 1–11.
[24] V. Sklyarova, J. Chelishev, A. Dogaru, I. Medvedev, V. Lempitsky, and E. Zakharov, "Neural haircut: Prior-guided strand-based hair reconstruction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 19762–19773.
[25] V. Sklyarova, E. Zakharov, O. Hilliges, M. J. Black, and J. Thies, "Text-conditioned generative model of 3D strand-based human hairstyles," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 4703–4712.
[26] Y. Takimoto et al., "Dressi: A hardware-agnostic differentiable renderer with reactive shader packing and soft rasterization," *Comput. Graph. Forum*, , vol. 41, no. 2, pp. 13–27, 2022.
[27] Y. Takimoto, H. Takehara, H. Sato, Z. Zhu, and B. Zheng, "Dr.Hair: Reconstructing scalp-connected hair strands without pre-training via differentiable rendering of line segments," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 20601–20611.
[28] ITU, *Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide-Screen 16:9 Aspect Ratios*, ITU-R BT.601-7, International Telecommunication Union, Geneva, Switzerland, 2011.
[29] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 27171–27183.
[30] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[31] K. Wu, Y. Ye, L. Yang, H. Fu, K. Zhou, and Y. Zheng, "Neuralhdhair: Automatic high-fidelity hair modeling from a single image using implicit neural representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. 2022*, 2022, pp. 1516–1525.
[32] L. Yang, Z. Shi, Y. Zheng, and K. Zhou, "Dynamic hair modeling from monocular videos using deep neural networks," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 235:1–235:12, 2019.
[33] W. Yifan, F. Serena, S. Wu, C. Öztireli, and O. Sorkine-Hornung, "Differentiable surface splatting for point-based geometry processing," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–14, Nov. 2019.
[34] C. Yuksel, S. Schaefer, and J. Keyser, "Hair meshes," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 1–7, 2009.
[35] M. Zhang and Y. Zheng, "Hair-GAN: Recovering 3D hair structure from a single image using generative adversarial networks," *Vis. Informat.*, vol. 3, no. 2, pp. 102–112, 2019.
[36] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 586–595.
[37] Y. Zheng et al., "HairStep: Transfer synthetic to real using strand and depth maps for single-view 3D hair modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 12726–12735.
[38] Y. Zheng et al., "Towards unified 3D hair reconstruction from single-view portraits," in *Proc. SIGGRAPH Asia 2024 Conf. Papers*, 2024, pp. 1–11.
[39] Y. Zhou, M. Chai, A. Pepe, M. Gross, and T. Beeler, "GroomGen: A high-quality generative hair model using hierarchical latent representations," *ACM Trans. Graph.*, vol. 42, no. 6, pp. 1–16, 2023.
[40] Y. Zhou et al., "GroomCap: High-fidelity prior-free hair capture," *ACM Trans. Graph.*, vol. 43, no. 6, pp. 1–15, 2024.
[41] Y. Zhou et al., "HairNet: Single-view hair reconstruction using convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 249–265.



Zhongsi Tang received the bachelor's degree in electrical engineering and automation from South China University of Technology. He is currently working toward the PhD degree in computer science with Zhejiang University under the supervision of Prof. Kun Zhou. His research interests focus on computer graphics and computer vision.



Jiahao Geng received the bachelor's degree in computer science from the University of Electronic Science and Technology of China. He later studied under professor Kun Zhou and earned his PhD degree in computer science from Zhejiang University, in 2023. Currently, he works as an algorithm engineer at FaceUnity. His research interests include modeling and animation.



Yanlin Weng received the master's and bachelor's degrees in control science and engineering from Zhejiang University and the PhD degree in computer science from the University of Wisconsin-Milwaukee. She is currently an associate professor with the School of Computer Science and Technology, Zhejiang University. Her research focuses on computer graphics and multimedia, with notable publications in leading journals including *ACM Transactions on Graphics (TOG)* and *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, as well as premier international conferences.



Youyi Zheng received the bachelor's in mathematics from Zhejiang University and the PhD degree in computer science from HKUST. He is a ZJU100 Young professor with the State Key Lab of CAD&CG, College of Computer Science, Zhejiang University. Prior to that, he was a PI at SIST of ShanghaiTech. From 2011 to 2015, he worked closely with Prof. Julie Dorsey with Yale and Prof. Niloy Mitra with UCL



Kun Zhou (Fellow, IEEE) received the PhD degree from Zhejiang University. He is a Cheung Kong professor of computer science with Zhejiang University and the director of the state Key Lab of CAD&CG. After graduation, he spent six years with Microsoft Research Asia, serving as a lead Researcher of the graphics group before returning to Zhejiang University. He was named one of the world's top 35 young innovators by MIT Technology Review (2011), and received the Asiagraphics Outstanding Technical Contributions Award (2022) and the ACM SIGGRAPH Test-of-Time Award (2024). He is a Fellow of ACM.

Authorized licensed use limited to: Zhejiang University. Downloaded on November 06,2026 at 06:01:17 UTC from IEEE Xplore. Restrictions apply.