Hair Strand Reconstruction based on 3D Gaussian Splatting

Yimin Pan
yimin.pan@tum.de
Matthias Niessner
niessner@tum.de
Tobias Kirschstein
tobias.kirschstein@tum.de

The Visual Computing & Artificial Intelligence Group
Technical University of Munich
Munich, DE

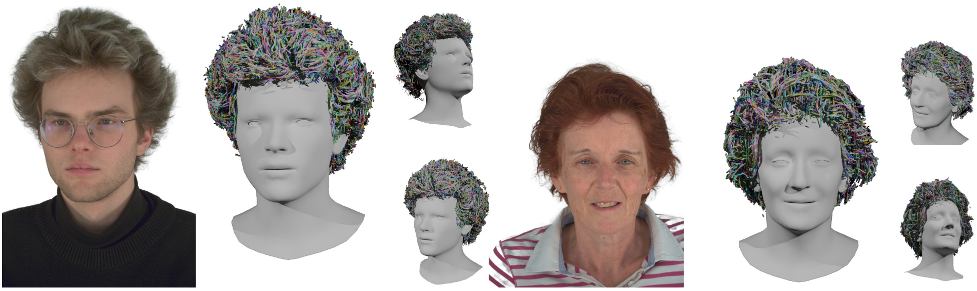

Figure 1: **Our method reconstructs realistic hair strands across a wide range of hairstyles.** Hair strands are rendered with different colors to facilitate visualization.

Abstract

Human hair reconstruction is a challenging problem in computer vision, with growing importance for applications in virtual reality and digital human modeling. Recent advances in 3D Gaussians Splatting (3DGS) provide efficient and explicit scene representations that naturally align with the structure of hair strands. In this work, we extend the 3DGS framework to enable strand-level hair geometry reconstruction from multi-view images. Our multi-stage pipeline first reconstructs detailed hair geometry using a differentiable Gaussian rasterizer, then merges individual Gaussian segments into coherent strands through a novel merging scheme, and finally refines and grows the strands under photometric supervision.

While existing methods typically evaluate reconstruction quality at the geometric level, they often neglect the connectivity and topology of hair strands. To address this, we propose a new evaluation metric that serves as a proxy for assessing topological accuracy in strand reconstruction. Extensive experiments on both synthetic and real-world datasets demonstrate that our method robustly handles a wide range of hairstyles and achieves efficient reconstruction, typically completing within one hour.

The project page can be found at: https://yimin-pan.github.io/hair-gs/

1 Introduction

Realistic human avatar modeling plays a crucial role in the gaming and virtual reality (VR) industry. Since the introduction of 3D Morphable Models (3DMM) by Blanz and Vetter [3], there has been a growing trend toward controllable head modeling. Recent advances in neural representations have further enhanced this field, enabling high-fidelity facial reconstructions [8, 13, 21, 29].

Despite these advancements, most head reconstruction methods focus primarily on facial geometry, often negleting the crucial role hair plays in defining personal identity. Traditional hair modeling tools are highly skill-intensive and time-consuming, as achieving realistic results requires the manual creation of a vast number of individual strands. This challenge has driven a shift toward image-based optimization techniques that aim to automate the hair reconstruction process.

However, hair reconstruction remains a challenging task due to the complex geometry of strands, frequent occlusions, and the limitations of conventional keypoint and multi-view stereo (MVS) techniques [11, 23]. Data-driven approaches [22, 24, 31] leverage learned priors to infer hair structure from images, enabling the recovery of even occluded inner volumes. Yet, their generalization capabilities are often limited by the scarcity of high-quality 3D data.

Recently, Gaussian Splatting [6] has emerged as a popular method for efficient and accurate scene reconstruction. However, its general-purpose design is not directly applicable for hair modeling. When applied directly, it results in numerous uncontrollable and disconnected Gaussians, and struggles particularly with modeling curly strands.

To address these limitations, we propose a novel multi-stage pipeline that adapts the 3DGS framework with constraints specifically tailored for hair strand reconstruction. In the first stage, we employ a differentiable rasterizer together with adaptive densification to obtain a dense and detailed representation of visible hair geometry. This step is essential, as successful merging relies on comprehensive geometry recovery to identify most suitable candidates. We then introduce a merging scheme based on distance and angle heuristics, enabling the combination of separated hair segments into longer strands. In the final stage, we refine the joint positions using a combination of photometric losses and smoothness regularization. Extensive quantitative and qualitative evaluations demonstrate that our method is general enough to reconstruct a wide variety of hairstyle and substantially faster than most data-driven approaches. Our main contributions can be summarized as follows:

- An efficient multi-stage optimization pipeline and a set of losses to supervise hair directions and prevent the formation of sharp angles.
- A merging scheme to combine short strands into longer ones using geometric heuristics.
- A novel metric for quantitative evaluation of topological strand accuracy.

2 Related Work

Optimization-Based Hair Reconstruction. Early approaches to hair strand reconstruction typically rely on Structure from Motion (SfM) to estimate hair geometry. For instance, Paris *et al.* [20] formulated the optimization problem using 2D orientation fields derived from

Gabor filters as constraints. Luo *et al.* [15] introduced a Markov Random Field (MRF) optimization to generate oriented point clouds, which were further refined with Poisson surface reconstruction and extended through visual hull carving and orientation-based refinement [16]. Despite these advances, such methods often produce sparse and noisy results due to the planar neighborhood assumptions of stereo vision [11], which do not hold for hair strands. Works like [1, 2, 26] addressed the reconstructing problem using properties from line primitives. Nam *et al.* [19] similarly tackled the challenge using LP-MVS, incorporating line constraints within a multi-phase optimization to recover accurate strands. Nevertheless, estimating inner hair volumes from images remains impractical due to heavy occlusion. More recently, [25] address this issue by introducing guide strands to interpolate and reconstruct full hair volumes.

Learning-Based Hair Reconstruction. Learning-based approaches have progressed rapidly with the release of public datasets such as [4, 30], enabling model pretraining on synthetic hair data. Recent methods employ neural architectures to synthesize 3D hair strands from images. For example, [33] uses a CNN-based autoencoder to infer strands from a single image, while [32] generates volumetric hair fields using GANs. More recent works [22, 24, 27] leverage implicit neural representations, texture-based encodings, and diffusion priors to further improve fidelity and realism. Nevertheless, these methods are usually limited by the scarcity of high-quality training data, often resulting in reduced expressiveness and oversmoothing, as shown in the experiments (see Figure 4).

Radiance Fields and Volumetric Rendering. The introduction of Neural Radiance Fields (NeRF) [18] and volumetric rendering has greatly advanced photorealistic human avatar modeling, enabling flexible representation of line-like structures such as hair, which are difficult to capture with traditional mesh-based methods. More recently, 3DGS [6] has emerged as an explicit and efficient alternative, representing scenes with 3D Gaussians that naturally align with the structure of hair strands. While prior works [14, 31, 34] typically use 3DGS for refining coarse geometry, our approach leverages 3DGS from the begining to recover precise hair segments, which are then merged into complete strands.

3 Method

Given a set of multi-view images of a static scene, our method reconstructs an accurate 3D hair model of the subject in the form of polylines. Similar to [6], we begin by preprocessing the images using COLMAP [23] to obtain camera poses and a sparse point cloud. However, due to the sensitivity of 3DGS to initialization, we use the vertices obtained from fitting a FLAME model [10]. Additionally, we estimate 2D orientation maps and corresponding confidence maps using Gabor filters [20], and extract hair segmentation masks using off-the-shelf methods [5, 12]. The complete pipeline is illustrated in Figure 2, which consists of three main stages: geometry reconstruction (Section 3.1), strand generation (Section 3.2), and final refinement (Section 3.3). Unlike other concurrent works, our approach leverages the flexibility of unconstrained Gaussian splatting to recover as many hair segments as possible, which are then merged into coherent strands.

The final output is a set of strands $S = \{S_1, \dots, S_n\}$, each comprising a variable number of linked 3D points $S_i = \{p_1, \dots, p_m\}, p_j \in \mathbf{R}^3$. In contrast to most existing methods [17, 19, 22, 24, 34], we do not explicitly specify the number of points per strand, as this is typically unknown and highly dependent on the shape and complexity of the curve being modeled. Instead, we rely on the optimization process and gradient feedback to dynamically determine

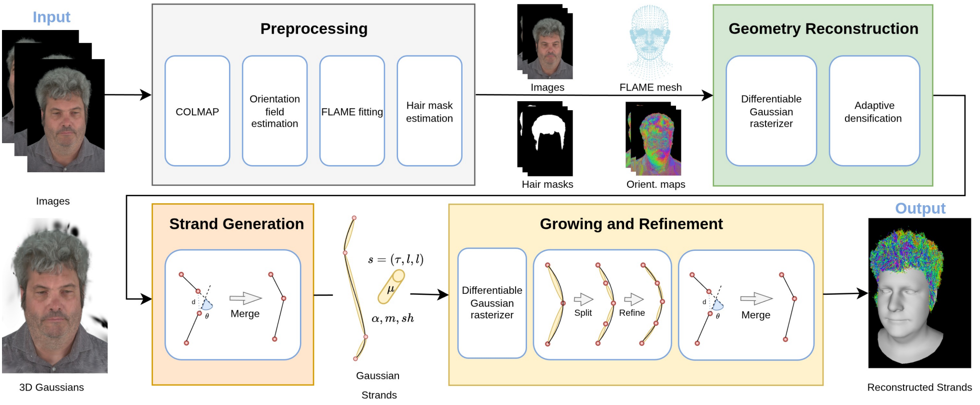

Figure 2: **An overview of the full pipeline.** The process begins with image preprocessing and geometry reconstruction using 3D Gaussians. Hair strands are created by merging Gaussians, followed by a refinement stage to optimize joint positions and the topology.

whether the current number of points is sufficient or if further densification is required.

3.1 Stage I: Geometry Reconstruction

The first stage of our pipeline reconstructs hair strand geometry by optimizing photometric losses within an analysis-by-synthesis framework, as shown in Figure 3. Following the original 3DGS, we initialize anisotropic 3D Gaussians from the input point cloud, each parameterized by a position mean $\mu \in \mathbf{R}^3$, per-axis scale $s \in \mathbf{R}^3$, a rotation matrix R (as a quaternion) and an opacity value $0 \leq \sigma(\alpha) \leq 1$, with $\sigma(x)$ denoting the sigmoid function. We introduce an additional mask value $0 \leq \sigma(m) \leq 1$, which is crucial for restricting the merging process to valid hair regions and for filtering out non-hair elements in the final output. The corresponding Gaussian distribution in world space is:

$$G(x) = e^{-\frac{1}{2}(x)^T \Sigma^{-1}(x)}, \quad \Sigma = RSS^T R^T \tag{1}$$

To focus on geometry rather than appearance, we represent the radiance field using spherical harmonics (SH) coefficients of degree 0, thereby avoiding view-dependent color effects.

We optimize the Gaussian parameters using the differentiable rasterizer $\mathcal{R}$ from [6]. The optimization is driven by a combination of photometric losses, $\mathcal{L}_1$ and $\mathcal{L}_{DSSIM}$, together with a bidirectional orientation loss:

$$\mathcal{L}_\theta = \frac{1}{HW} \sum_{y=1}^{H} \sum_{x=1}^{W} \Delta_\theta(\hat{\theta}(x,y), \theta(x,y)) C_\theta(x,y), \quad \Delta_\theta(\theta_1, \theta_2) = \min(|\theta_1 - \theta_2|, \pi - |\theta_1 - \theta_2|) \tag{2}$$

where $\hat{\theta}$ and θ represents the rendered and precomputed orientation maps, and C_θ the corresponding confidence map. This loss accelerates convergence toward the correct hair orientation.

To further supervise α, we introduce a mask loss based on the binary cross-entropy

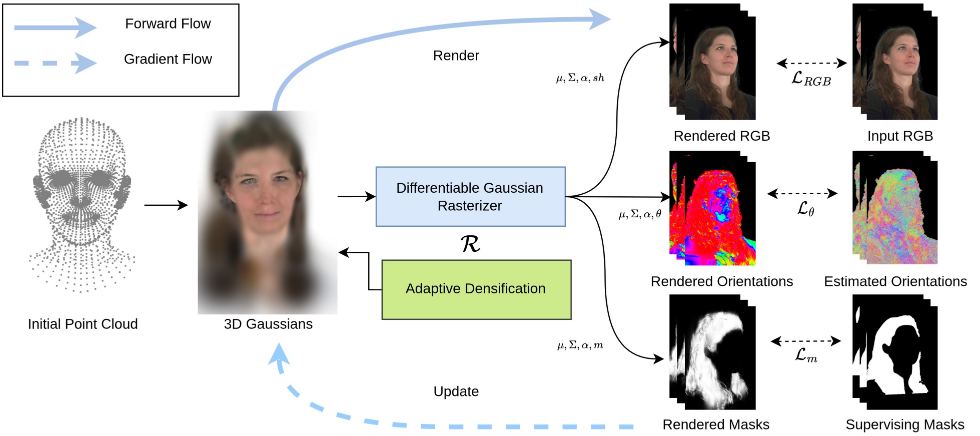

Figure 3: **Overview of the first-stage process**. From an initial point cloud, 3D Gaussians are optimized using the differentiable rasterizer with the supervision of a combination of RGB, orientation, and mask losses ($\mathcal{L}_{RGB}$, $\mathcal{L}_{\theta}$, $\mathcal{L}_m$). The densification process populates Gaussians in areas with missing geometry.

between the precomputed mask $M(x,y)$ and the rendered mask $\hat{M}(x,y)$:

$$\mathcal{L}_m = -\frac{1}{HW}\sum_{y=1}^{H}\sum_{x=1}^{W}\left[M(x,y)\log\left(\hat{M}(x,y)\right) + (1 - M(x,y))\log\left(1 - \hat{M}(x,y)\right)\right] \tag{3}$$

The overall objective for the first-stage is:

$$\mathcal{L}_{first} = (1 - \lambda_{DSSIM})\mathcal{L}_1 + \lambda_{DSSIM}\mathcal{L}_{DSSIM} + \lambda_{\theta}\mathcal{L}_{\theta} + \lambda_m\mathcal{L}_m \tag{4}$$

We use the default 3DGS densification mechanism to add Gaussians where geometry is missing, resulting in a dense and detailed scene representation for the next stages.

3.2 Stage II: Strand Generation

Strand Representation. To continue leveraging the 3DGS framework for strand optimization, we represent each hair strand as a chain of linked joints, similar to [14, 34]. In contrast to the first stage, rasterization is now performed on the segments connecting these joints, rather than on the joints themselves. Each segment is modeled as a cylinder aligned with the x-axis, with its scale defined as $(||p_{j+1} - p_j||_2, \tau_j, \tau_j)$, where τ_j is a learnable thickness parameter. The center μ of each segment is set to the midpoint between its endpoints, and the rotation matrix R is computed to align the x-axis vector $\vec{x} = (1,0,0)$ with the segment direction $\vec{p}j = \frac{pj+1-p_j}{|pj+1-p_j|2}$ using the Rodrigues rotation formula:

$$R = I + K + \frac{K^2}{1 + v \cdot d}, \quad K = \vec{x} \times \vec{p}_j \tag{5}$$

Other segment properties, such as opacity α, mask m, and color sh, are optimized independently.

Merging Scheme. Initially, each Gaussian from the first stage is converted into a short strand with two joints, based on its position and direction. To form longer and more realistic (curvy) strands, we iteratively merge these short strands. This merging is formulated as an assignment problem, where strand endpoints (tips and roots) are nodes in a bipartite graph, and edge costs are determined by spatial distance and angular difference. While the Hungarian Matching Algorithm [9] can find the optimal solution, its computational and memory requirements are prohibitive for our large-scale problem (up to half a million nodes). Therefore, we use a more efficient greedy approach: candidates are filtered by predefined distance and angle thresholds, sorted by cost, and only the lowest-cost matches are kept. We implement this efficiently using a K-D tree-based nearest neighbor search.

For each selected pair, a new joint is created at the midpoint, and the new strand is formed by connecting the original endpoints to this joint. This process is illustrated in Strand Generation of Figure 2.

To ensure high confidence in the initial merging process, we use very restrictive thresholds: $d_m = 2\,\mathrm{mm}$ and $\theta_m = 20^\circ$.

3.3 Stage III: Growing and Refinement

Geometry Refinement. After merging, we observe a gradual loss of geometric accuracy, mainly due to the naive placement of new joints at the midpoints of merged endpoints. To correct this, we refine joint positions by reintroducing image-based supervision with the losses from Stage 1.

Unconstrained optimization can also cause unnaturally sharp angles between connected segments. To prevent this, we add an angle smoothness loss that penalizes large direction changes. Let $\mathcal{C}$ be the set of all connected segment pairs (a,b), $\vec{p}_a$, $\vec{p}_b$ their normalized direction vectors, and θ_s an angle threshold. The loss defined as:

$$\mathcal{L}_{\text{smooth}} = \frac{1}{|\mathcal{C}|} \sum_{(a,b)\in\mathcal{C}} \theta_{a,b}^2, \quad \theta_{a,b} = \begin{cases} \cos^{-1}(\vec{p}_a \cdot \vec{p}_b) & \text{if } \vec{p}_a \cdot \vec{p}_b \leq \cos(\theta_s) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

The overall loss for optimizing the joint positions of our Gaussian strands $\mathcal{GS}$ is:

$$\mathcal{L}_{third} = \mathcal{L}_{first} + \lambda_{smooth}\mathcal{L}_{smooth} \tag{7}$$

Topology Refinement. Empirically, we find that sharp angles most often arise when modeling curly strands. While the smoothness loss helps prevent such artifacts, it alone is not sufficient to capture the complexity of curly hair. To address this, we split segments that exceed a length threshold, inserting new joints at their midpoints. This increases the degrees of freedom and allows the model to better represent curves. Subsequent image-based supervision further refines these new joint positions, resulting in more accurate strand shapes (see the Growing and Refinement stage in Figure 2).

Despite these improvements, reconstructed strands may still be shorter than in reality or appear as multiple disconnected segments. This occurs because the merging algorithm cannot always identify endpoints belonging to the same underlying strand if they are not sufficiently close. To alleviate this problem, we gradually relax the merging criteria during optimization by increasing the distance and angle thresholds from $d_m = 2mm$ and $\theta_m = 20^\circ$ at the start, to $d_m = 4mm$ and $\theta_m = 40^\circ$ by the end. This strategy enables more merges and nearly doubles the average strand length.

Finally, we filter out non-hair geometries using the learned mask values. The final result is a clean set of hair strands represented as polylines, constructed from the optimized joint positions and connectivity.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate our method and baselines on both synthetic and real-world datasets. For quantitative analysis, we use the USC-HairSalon [4] dataset, which is, to our knowledge, the largest publicly available hair dataset, containing 343 hairstyles with 10,000 strands and 100 joints per strand. We also use the Cem Yuksel hair models [30] as a test set; these models are smaller in scale and not used for training by learning-based baselines. For each sample, we generate 16 input images by rotating a virtual camera around the subject in OpenGL, using simple ambient and diffuse lighting. Hair strands are rendered as line primitives together with the head mesh and assigned brown color with slight variations for realism.

For qualitative evaluation, we also test on the real-world NeRSemble dataset [8], which features real head captures with diverse hairstyles.

Baselines. We compare our method against state-of-the-art 3D hair reconstruction approaches based on multiview images, including Neural Haircut [24] as a representative data-driven method, and classical SfM-based methods LP-MVS [19] and Strand Integration [17]. For LP-MVS, we use the implementation provided by the authors of Strand Integration, as no official code is available. Both methods return a directed point cloud without topological information, so we perform additional postprocessing to convert them into strands using the forward Euler method described in [19]. However, this lacks the mean-shift and growing steps which are essential for producing clean, long hair strands. We therefore keep both versions and refer to the postprocessed results as LP-MVS† and Strand Integration†.

Implementation Details. For the initial geometry reconstruction, we optimize the Gaussians for 30,000 iterations using the Adam optimizer [7]. The merging algorithm is run until no candidates remain, followed by an additional 30,000 iterations of refinement. All other hyperparameters are adopted directly from [6]. Experiments are conducted on a NVIDIA RTX 4090 GPU and AMD Ryzen 9 7950X CPU, with the entire pipeline typically completing in about 1 hour. Our fully optimization-based approach is significantly faster than learning-based methods, which are generally more computationally intensive. For example, Neural Strands [22] takes 48 hours on a NVIDIA V100, Neural Haircut [24] requires approximately 120 hours on our setup, and GroomCap [34] takes a similar amount of time on even higher-end hardware.

4.2 Evaluation Measures

Due to the difficulty of finding true correspondence between points. We follow common practices from[19, 24, 25] to define precision, recall and f1-score based on matched nearest neighbors under a distance and angle threshold.

Strand Consistency. A key limitation of these metrics is that they only assess geometric accuracy at the point cloud level, ignoring the essential topological structure of hair strands. As long as the location of the points is correct, the metrics do not penalize even if the edges are completely wrong. To address this, we introduce a novel metric, strand consistency (SC),

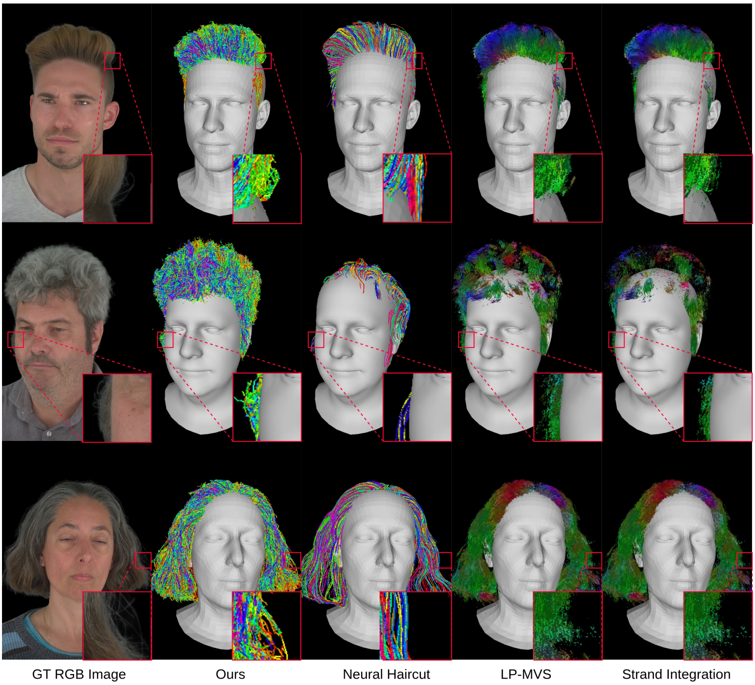

Figure 4: **Comparison on real-world subjects.** Our method excels on challenging hairstyles, especially curly hair. Colors either differentiate strands (Ours and Neural Haircut) or direction (others).

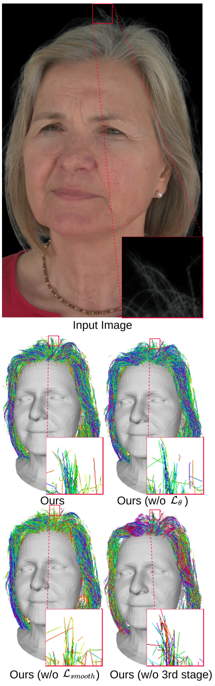
Input Image
Ours
Ours (w/o $\mathcal{L}_\theta$)
Ours (w/o $\mathcal{L}_{smooth}$)
Ours (w/o 3rd stage)

Figure 5: **Ablation study.** Full method gives best results.

which evaluates the connectivity of points within hair strands. Specifically, the metric measures, for each Ground truth (GT) strand, the highest fraction of its points that are matched to points in a single predicted strand, and then averages this value over all strands.

Let $\mathcal{S}_G$ and $\mathcal{S}_P$ denote the sets of ground truth and predicted strands, respectively. And let g and p be points on s_G and s_P. $\mathbf{I}(\cdot)$ is the indicator function and d_τ and θ_τ are the distance and angle thresholds. The computation is as follows:

$$C = \frac{1}{|\mathcal{S}_G|} \sum_{s_G \in \mathcal{S}_G} \max_{s_P \in \mathcal{S}_P} \left(\frac{1}{|s_G|} \sum_{g \in s_G} \mathbf{I}\left(\exists p \in s_P : \|g - p\|_2 \leq d_\tau \wedge \theta_{g,p} \leq \theta_\tau\right) \right) \tag{8}$$

Intuitively, this metric looks at each GT strand and finds the predicted strand with the most matching directed points.

4.3 Comparison

Quantitative Results. On the USC-HairSalon dataset, our method achieves higher precision but lower recall than Strand Integration, indicating that while MVS-based methods are robust to input variations, they often produce noisy results with incorrect directions due to reliance on hand-crafted filters (see Table 1). Neural Haircut shows consistently lower scores across all metrics, reflecting limited generalization to the synthetic evaluation data. In contrast, our

Method	Thresholds: mm / degree							
	Precision		Recall		F-score		SC	
	2/20	4/40	2/20	4/40	2/20	4/40	2/20	4/40
Ours	0.445	**0.769**	0.182	**0.545**	0.256	**0.636**	**0.050**	**0.189**
Neural Haircut	0.187	0.603	0.040	0.192	0.064	0.291	0.023	0.094
LP-MVS	0.388	0.643	0.209	0.505	0.271	0.565	–	–
LP-MVS†	0.229	0.474	0.098	0.345	0.136	0.398	0.019	0.046
Strand Integration	**0.497**	0.712	**0.243**	0.533	**0.326**	0.608	–	–
Strand Integration†	0.258	0.486	0.117	0.384	0.161	0.427	0.022	0.053

Table 1: Quantitative comparison on aggregated samples from the USC-HairSalon dataset. Higher values indicate better results and are marked in bold.

Method	Thresholds: mm / degree							
	Precision		Recall		F-score		SC	
	2/20	4/40	2/20	4/40	2/20	4/40	2/20	4/40
Ours	**0.411**	**0.736**	**0.139**	**0.468**	**0.207**	**0.572**	**0.041**	**0.166**
Neural Haircut	0.190	0.639	0.008	0.066	0.015	0.120	0.006	0.042
LP-MVS	0.302	0.532	0.052	0.289	0.089	0.375	–	–
LP-MVS†	0.212	0.472	0.033	0.226	0.056	0.306	0.015	0.064
Strand Integration	0.154	0.346	0.021	0.136	0.037	0.195	–	–
Strand Integration†	0.106	0.300	0.012	0.107	0.022	0.157	0.007	0.038

Table 2: Quantitative comparison on a challenging curly hair sample from the Cem-Yuksel hair model. Higher values indicate better results.

fully optimization-based approach, which makes fewer assumptions from the input, demonstrates greater robustness and generalization. On the challenging curly hair sample from the Cem-Yuksel hair model, our method outperforms all others across every metric (Table 2), and achieves the highest strand consistency, indicating more reliable reconstruction of correct strand connectivity.

Qualitative Results. Further analysis on the NeRSemble dataset [8] (Figure 4) shows results for three hairstyles: straight, curly, and long hair. Neural Haircut improves on real-world data, generally matching the overall shape for straight hair. However, only our method accurately captures fine details, such as thin and floating strands in the female subject. For curly hair, which is especially challenging due to rapid directional changes, LP-MVS and its variants capture only the sparse outer geometry, while Neural Haircut fails to represent the overall shape. In contrast, our optimization-based method, free from prior assumptions, robustly reconstructs all tested hairstyles.

4.4 Ablation Study

Merging thresholds	Thresholds: mm / degree							
	Precision		Recall		F-score		SC	
	2/20	4/40	2/20	4/40	2/20	4/40	2/20	4/40
$1mm/10^\circ \rightarrow 2mm/20^\circ$	0.4264	0.7606	0.1281	0.4819	0.197	0.59	0.048	0.1822
$2mm/20^\circ \rightarrow 4mm/40^\circ$	**0.4639**	0.8037	0.15	**0.54**	0.2277	0.6438	**0.055**	0.2196
$4mm/40^\circ \rightarrow 6mm/60^\circ$	0.4618	**0.8213**	**0.155**	0.5383	**0.233**	**0.65**	0.055	**0.2262**
$6mm/60^\circ \rightarrow 8mm/80^\circ$	0.4355	0.775	0.1376	0.5182	0.271	0.565	0.05	0.1866

Table 3: Impact of different initial and final merging thresholds on reconstruction performance. Results show that performance remains stable within a reasonable range, while overly extreme thresholds degrade accuracy. Bold numbers indicate the best results.

We evaluate the impact of key design choices in our method (Figure 5). Removing the angle smoothness loss leads to unrealistic, spiky strands due to the lack of regularization on segment directions. Skipping the third-stage optimization, which refines joint positions and adaptively inserts points, produces coarse and inaccurate strands, as the initial merging only yields rough approximations. Restrictive merging criteria in the second stage prevent short strands from being combined into longer ones, while enforcing a limited number of joints constrains the representation to short, straight segments. Without orientation supervision, hair segments often deviate in incorrect directions. In contrast, the full method integrates all components and achieves the most faithful strand reconstructions.

We further analyze the impact of different merging thresholds (Table 3). As shown in the first two rows, varying the thresholds within a reasonable range leads to only minor performance differences. A noticeable drop in accuracy occurs only when the thresholds are pushed to extremes—either overly restrictive or overly permissive, as in the first and fourth rows—indicating that our method remains robust to threshold selection as long as it stays within practical bounds.

5 Conclusion

In this work, we introduce a fully optimization-based method for hair reconstruction from multi-view images, capable of handling a wide range of hairstyles without relying on priors from synthetic datasets. Our multi-stage pipeline first estimates an accurate Gaussian representation using 3DGS, then merges individual segments to form complete hair strands.

To address the limitations of existing evaluation metrics, which overlook the structural connectivity of hair strands, we propose a novel metric as a proxy for topological accuracy.

Extensive experiments on both synthetic and real-world datasets show that our method outperforms existing approaches, producing highly detailed hair models—including fine, floating strands—while being considerably faster than data-driven methods, with reconstructions completed in under one hour.

Beyond hair, our framework could naturally extend to other line-like structures such as cables or wires, requiring minimal adaptation of the segmentation model.

Despite these strengths, some limitations remain. Our merging criteria can prevent effective merging of Gaussians from the same strand, resulting in shorter reconstructed strands. As discussed before using a variation of Hungarian algorithm for optimal matching could address this. Additionally, reconstructed strands are not necessarily attached to the scalp, limiting their use in rendering engines; future work could address this by pinning strand roots to the surface and growing strands as in [28, 34].

6 Acknowledgements

This work was supported by the ERC Consolidator Grant Gen3D (101171131) and the German Research Foundation (DFG) Research Unit “Learning and Simulation in Visual Computing”.

References

[1] Adrien Bartoli and Peter Sturm. Structure-from-motion using lines: Representation, triangulation, and bundle adjustment. *Computer Vision and Image Understanding*, 100(3):416–441, 2005. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2005.06.001. URL https://www.sciencedirect.com/science/article/pii/S1077314205000846.

[2] Herbert Bay, Vittorio Ferrari, and Luc Van Gool. Wide-baseline stereo matching with line segments. volume 1, pages 329–336, 01 2005. doi: 10.1109/CVPR.2005.375.

[3] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive*

Techniques, SIGGRAPH '99, page 187–194, USA, 1999. ACM Press/Addison-Wesley Publishing Co. ISBN 0201485605. doi: 10.1145/311535.311556. URL https://doi.org/10.1145/311535.311556.

[4] Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. Single-view hair modeling using a hairstyle database. *ACM Trans. Graph.*, 34(4), July 2015. ISSN 0730-0301. doi: 10.1145/2766931. URL https://doi.org/10.1145/2766931.

[5] Zhanghan Ke, Jiayu Sun, Kaican Li, Qiong Yan, and Rynson W.H. Lau. Modnet: Real-time trimap-free portrait matting via objective decomposition. In *AAAI*, 2022.

[6] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42, 8 2023. ISSN 15577368. doi: 10.1145/3592433.

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 12 2014. URL http://arxiv.org/abs/1412.6980.

[8] Tobias Kirschstein, Shenhan Qian, Simon Giebenhain, Tim Walter, and Matthias Nießner. Nersemble: Multi-view radiance field reconstruction of human heads. *ACM Trans. Graph.*, 42(4), July 2023. ISSN 0730-0301. doi: 10.1145/3592455. URL https://doi.org/10.1145/3592455.

[9] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 52, 1955. URL https://api.semanticscholar.org/CorpusID:9426884.

[10] Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. In *ACM Transactions on Graphics*, volume 36. Association for Computing Machinery, 11 2017. doi: 10.1145/3130800.3130813.

[11] Tony Lindeberg. *Scale Invariant Feature Transform*, volume 7. 05 2012. doi: 10.4249/scholarpedia.10491.

[12] Kunliang Liu, Ouk Choi, Jianming Wang, and Wonjun Hwang. Cdgnet: Class distribution guided network for human parsing. pages 4463–4472, 06 2022. doi: 10.1109/CVPR52688.2022.00443.

[13] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering, 2021. URL https://arxiv.org/abs/2103.01954.

[14] Haimin Luo, Min Ouyang, Zijun Zhao, Suyi Jiang, Longwen Zhang, Qixuan Zhang, Wei Yang, Lan Xu, and Jingyi Yu. Gaussianhair: Hair modeling and rendering with light-aware gaussians. 2 2024. URL http://arxiv.org/abs/2402.10483.

[15] Linjie Luo, Hao Li, S. Paris, T. Weise, Mark Pauly, and S. Rusinkiewicz. Multi-view hair capture using orientation fields. pages 1490–1497, 06 2012. ISBN 978-1-4673-1226-4. doi: 10.1109/CVPR.2012.6247838.

[16] Linjie Luo, Cha Zhang, Zhengyou Zhang, and Szymon Rusinkiewicz. Wide-baseline hair capture using strand-based refinement. pages 265–272, 06 2013. doi: 10.1109/CVPR.2013.41.

[17] Ryota Maeda, Kenshi Takayama, and Takafumi Taketomi. Refinement of hair geometry by strand integration initial point cloud from lpmvs result of refinement strand integration, 2023. URL https://github.com/elerac/strand_integration.

[18] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. 3 2020. URL http://arxiv.org/abs/2003.08934.

[19] Giljoo Nam, Chenglei Wu, Min H Kim, and Yaser Sheikh. Strand-accurate multi-view hair capture, 2019. URL https://api.semanticscholar.org/CorpusID:199016694.

[20] Sylvain Paris, Hector Briceño, and François Sillion. Capture of hair geometry from multiple images. *ACM Transactions on Graphics*, 23, 06 2004. doi: 10.1145/1015706.1015784.

[21] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. *arXiv preprint arXiv:2312.02069*, 2023.

[22] Radu Alexandru Rosu, Shunsuke Saito, Ziyan Wang, Chenglei Wu, Sven Behnke, and Giljoo Nam. Neural strands: Learning hair geometry and appearance from multi-view images. 7 2022. URL http://arxiv.org/abs/2207.14067.

[23] Johannes L. Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 501–518, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46487-9.

[24] Vanessa Sklyarova, Jenya Chelishev, Andreea Dogaru, Igor Medvedev, Victor Lempitsky, and Egor Zakharov. Neural haircut: Prior-guided strand-based hair reconstruction. 6 2023. URL http://arxiv.org/abs/2306.05872.

[25] Yusuke Takimoto, Hikari Takehara, Hiroyuki Sato, Zihao Zhu, and Bo Zheng. Dr.hair: Reconstructing scalp-connected hair strands without pre-training via differentiable rendering of line segments, 2024. URL https://arxiv.org/abs/2403.17496.

[26] Anil Usumezbas, Ricardo Fabbri, and Benjamin Kimia. From multiview image curves to 3d drawings. volume 9908, pages 70–87, 10 2016. ISBN 978-3-319-46492-3. doi: 10.1007/978-3-319-46493-0_5.

[27] Keyu Wu, Yifan Ye, Lingchen Yang, Hongbo Fu, Kun Zhou, and Youyi Zheng. Neuralhdhair: Automatic high-fidelity hair modeling from a single image using implicit neural representations, 2022. URL https://arxiv.org/abs/2205.04175.

[28] Keyu Wu, Lingchen Yang, Zhiyi Kuang, Yao Feng, Xutao Han, Yuefan Shen, Hongbo Fu, Kun Zhou, and Youyi Zheng. Monohair: High-fidelity hair modeling from a monocular video, 2024. URL https://arxiv.org/abs/2403.18356.

[29] Wangbo Yu, Yanbo Fan, Yong Zhang, Xuan Wang, Fei Yin, Yunpeng Bai, Yan-Pei Cao, Ying Shan, Yang Wu, Zhongqian Sun, and Baoyuan Wu. Nofa: Nerf-based one-shot facial avatar reconstruction. pages 1–12, 07 2023. doi: 10.1145/3588432.3591555.

[30] Cem Yuksel, Scott Schaefer, and John Keyser. Hair meshes. *ACM Transactions on Graphics*, 28:1–7, 12 2009. ISSN 15577368. doi: 10.1145/1618452.1618512.

[31] Egor Zakharov, Vanessa Sklyarova, Michael J Black, Giljoo Nam, Justus Thies, and Otmar Hilliges. Human hair reconstruction with strand-aligned 3d gaussians. *ArXiv*, September 2024.

[32] Meng Zhang and Y. Zheng. Hair-gan: Recovering 3d hair structure from a single image using generative adversarial networks. *Visual Informatics*, 3:102–112, 6 2019. ISSN 2468502X. doi: 10.1016/j.visinf.2019.06.001.

[33] Yi Zhou, Liwen Hu, Jun Xing, Weikai Chen, Han-Wei Kung, Xin Tong, and Hao Li. Hairnet: Single-view hair reconstruction using convolutional neural networks. 6 2018. URL http://arxiv.org/abs/1806.07467.

[34] Yuxiao Zhou, Menglei Chai, Daoye Wang, Sebastian Winberg, Erroll Wood, Kripasindhu Sarkar, Markus Gross, and Thabo Beeler. Groomcap: High-fidelity prior-free hair capture, 09 2024.