Supplementary Material for Dr.Hair: Reconstructing Scalp-Connected Hair Strands without Pre-Training via Differentiable Rendering of Line Segments

Yusuke Takimoto[1*] Hikari Takehara[1*] Hiroyuki Sato[1*] Zihao Zhu[1,2†] Bo Zheng[1]

[1]Huawei Technologies Japan K.K. [2]Keio University

1. Experimental settings

1.1. Synthetic data: Cem Yuksel's hair models [16]

We used wStraight, wCurly, and wWavy models, all of which have 50,000 strands. The head model, accompanied by the hair models, was attached for rendering by Blender Cycles. A white uniform environment map was used for illumination. Camera parameters were set the same as the real studio data discussed later.

1.2. Real data

H3DS [11]

H3DS is a real-world multi-view dataset for head reconstruction. GT Head meshes were scanned by a laser scanner, and independently captured multi-view images surrounding the subject in 360° were registrated against the mesh. About 70 images are provided per subject. Top views are not well captured in both the mesh and the images. Moreover, some views are affected by strong flash lighting that deteriorates image quality. For NeuralHaircut, by following the official implementation setting, 32 clean views manually annotated by the dataset authors were used. For the other methods, all views were inputted.

Monocular hand-held video [14]

A subject asked to be as static as possible on a chair was captured in circular motion by a smartphone. Subsampled 60 frames are provided. Camera parameters were estimated with COLMAP [13].

Studio data

Original 58 images were captured in 1824x2736 pixels by DSLRs with hardware synchronized shooting. The cameras were evenly put on a hemisphere, and similarly positioned LEDs were illuminated for uniform lighting. Camera parameters and a raw mesh were estimated by MetaShape [2]. The images were resized to a height of 684 pixels for LPMVS and Strand Integration and 512 pixels for NeuralHaircut and ours.

1.3. Existing methods' settings

LPMVS [9] and Strand Integration [8]

Default values were used for most parameters. Reasonable values were set to scene-dependent minimum and maximum depth according to the distance between the camera and the subjects.

NeuralHaircut [14]

We followed the instructions to run the official implementation, including some manual processes. 50,000 strands were sampled for visualization and quantitative evaluation while 1,900 strands were used for training as in the default setting.

2. Additional Results

We show additional results on synthetic and real data.

2.1. Additional comparison and ablations on synthetic data

Qualitative comparison with existing methods is shown in the upper rows of Figure 1 and Figure 2. These results correspond to the quantitative comparison in the main paper, Table 2. The frontal scalp alignment of NeuralHaircut is not accurate, and for Straight Hair, NeuralHaircut confuses hair with the head. The results of LPMVS and Strand Integration are almost similar, showing many short strands, inconsistent 3D orientation, and no distinction between head and hair. Our full pipeline shows better precision for both cases.

The lower rows of Figure 1 and Figure 2 visualize the ablation study. Even in **w/o DR**, the outline is well estimated, but it leaves room for improvement in fine details. In **w/o guide opt.**, child strands become too smoother. In

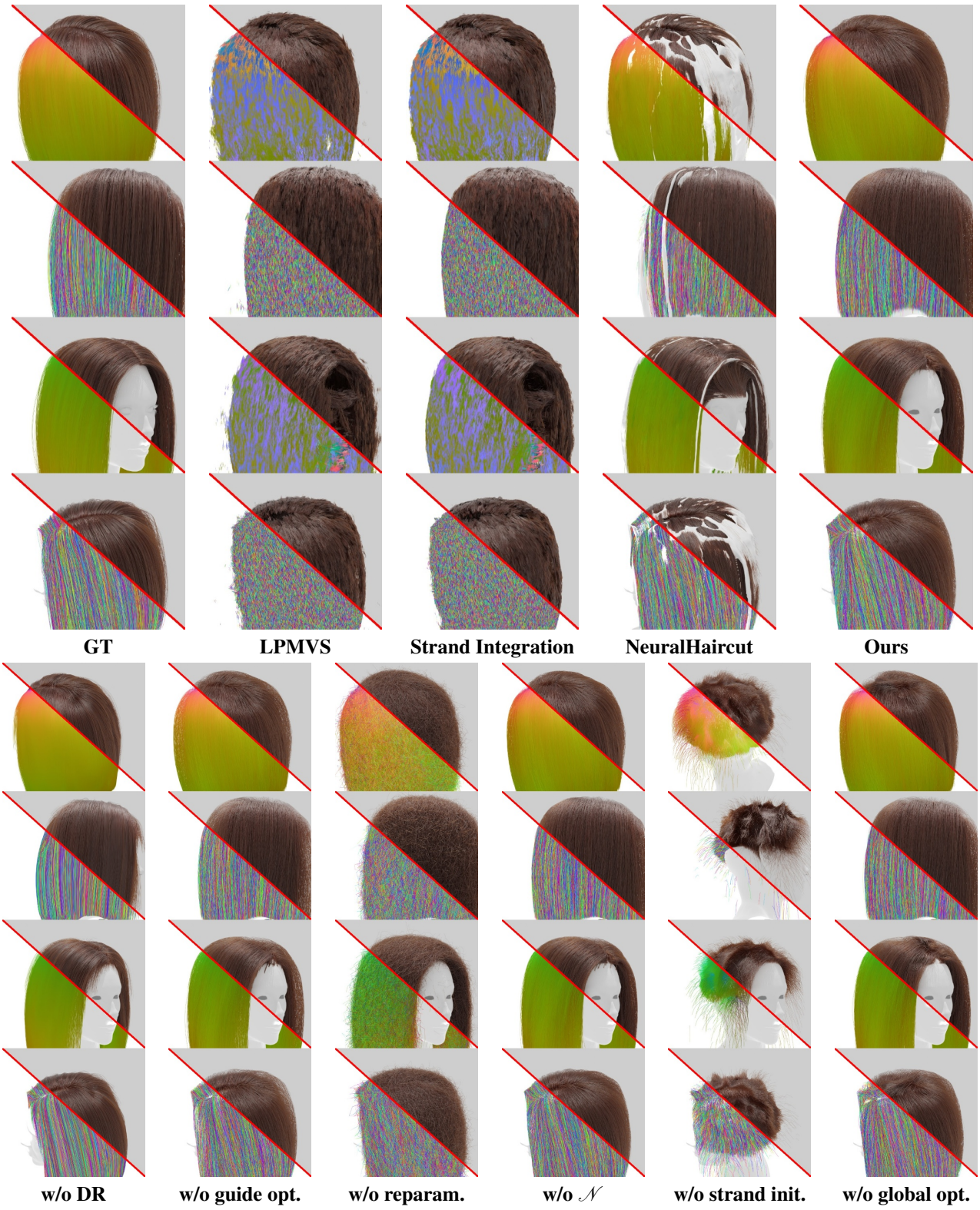


Figure 1. Qualitative evaluation on synthetic Straight Hair, corresponding to Table 2 of the main paper and Table 1 of this material. Four views per method are displayed with different strand visualization: Blender Cycles shading, 3D orientation, and random color. The upper row shows a comparison with existing methods, and the lower row displays the ablation study.

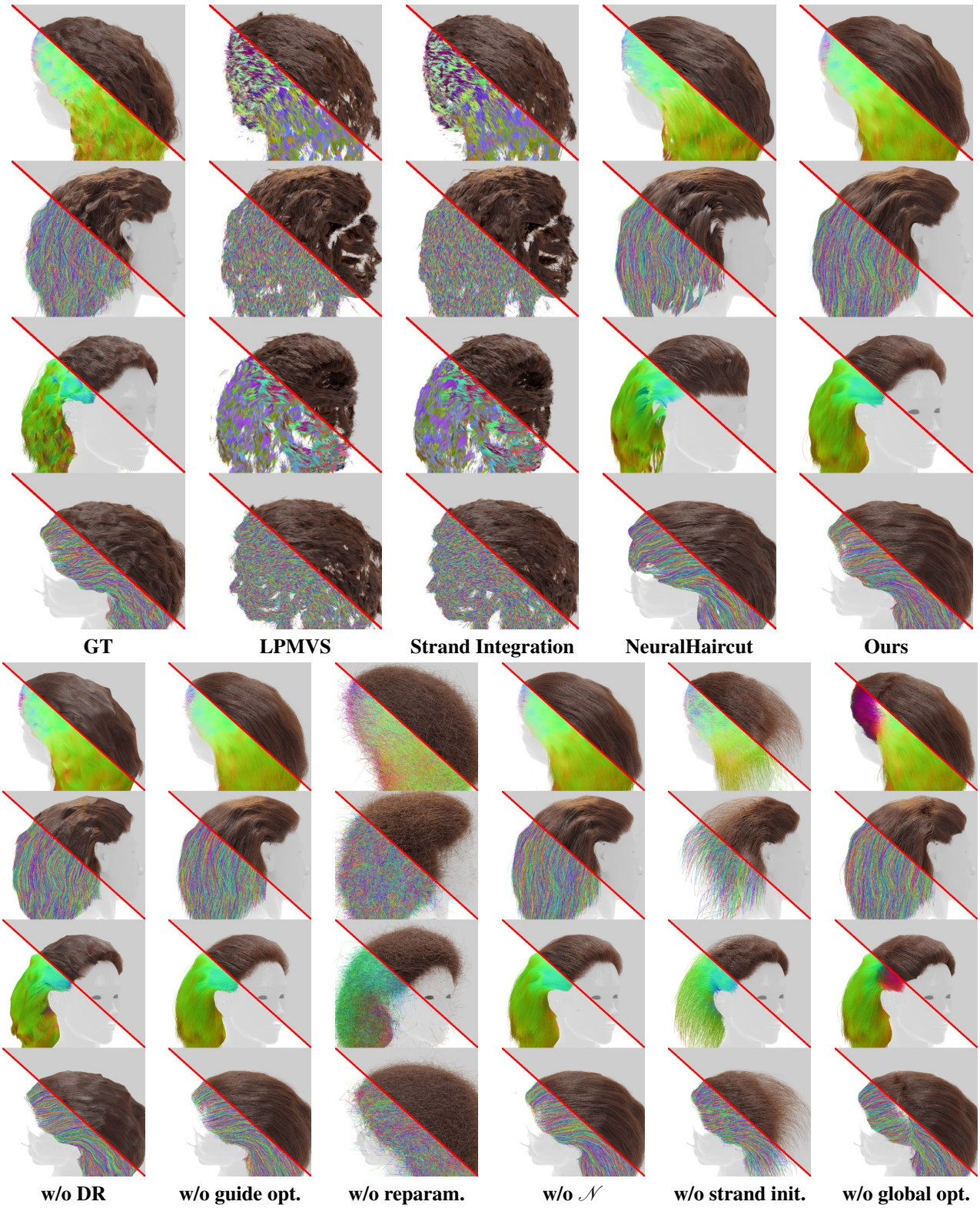


Figure 2. Qualitative evaluation on synthetic Curly Hair, corresponding to Table 2 of the main paper and Table 1 of this material. Four views per method are displayed with different strand visualization: Blender Cycles shading, 3D orientation, and random color. The upper row shows a comparison with existing methods, and the lower row displays the ablation study.

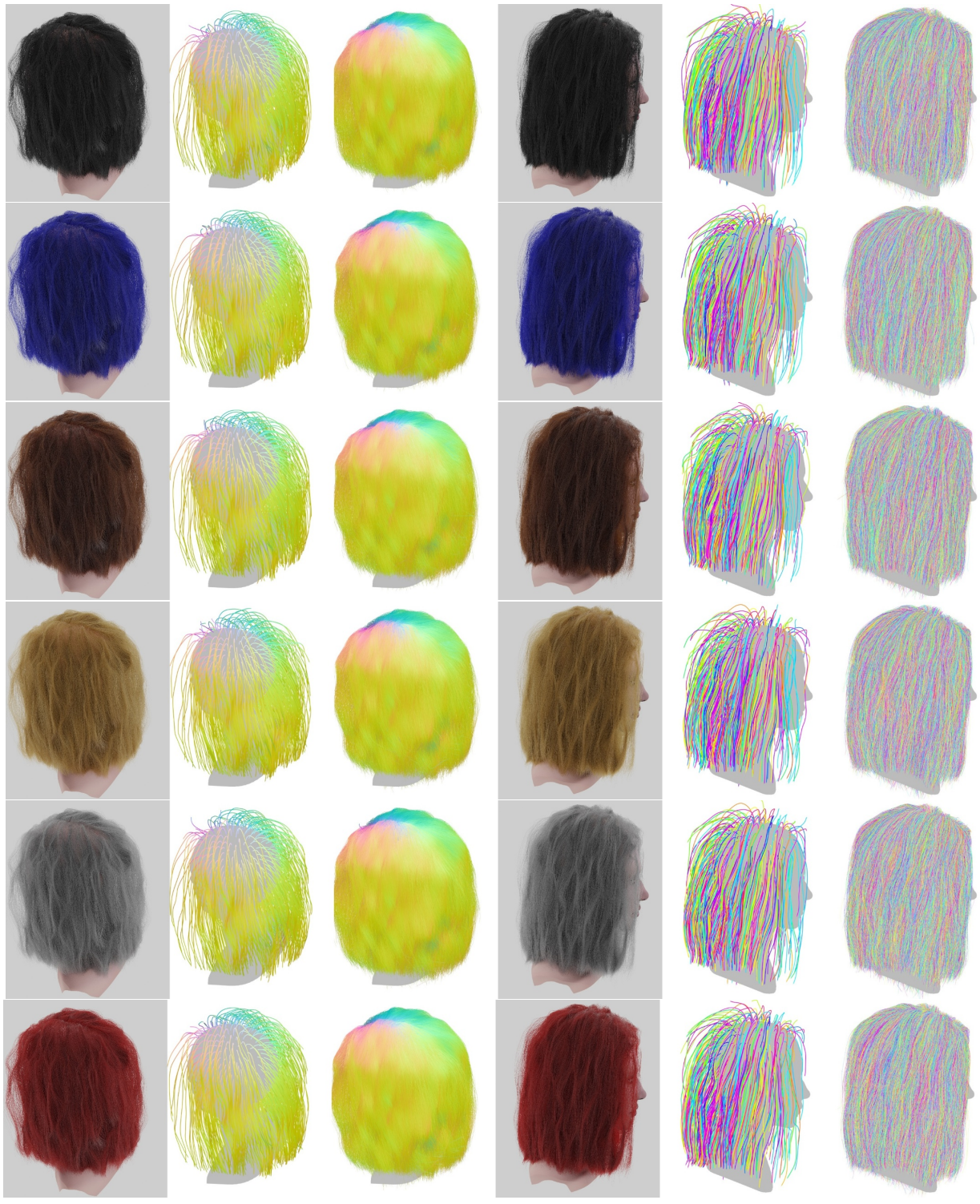


Figure 3. Additional results on synthetic Wavy hair with different colors. From top to bottom, black, blue, brown, gold, gray, and red colors are tested. From right to left, top view GT, top view guide with 3D orientation, top view child with 3D orientation, side view GT, side view guide with random color, and side view child with random color are shown. The results are almost same and accurate, which indicate our method is less sensitive to the color of the input hairs.

Table 1. Quantitative comparison with existing methods and ablation study on synthetic data tolerating **180° ambiguity**, which have been used in the previous studies. **P**, **R**, and **F1** denote precision, recall, and F1 score, respectively. Higher is better. The lower rows describe the values of our full pipeline and ours without individual modules. w/o DR: DR optimization is not applied, and the initialized strands are evaluated. w/o guide opt.: Child strands are optimized from the beginning of the DR step. w/o reparam.: Reparameterization is disabled. w/o $\mathscr{N}$: Only $\mathscr{N}$ is abandoned in the reparameterization. w/o strand init.: Strands are initialized by straight lines parallel to the normal of the scalp. w/o global opt.: Only gravity heuristic is applied to the initial 3D orientation, and 180° ambiguity is accepted on the other steps.

	Straight Hair									**Curly Hair**								
Threshold	1mm/10°			2mm/20°			3mm/30°			1mm/10°			2mm/20°			3mm/30°		
Measure	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
LPMVS [9]	61.0	37.1	46.1	81.1	62.8	70.8	87.5	76.0	81.4	36.9	8.1	13.3	65.6	18.6	28.9	74.0	28.1	40.7
Strand Integration [8]	**68.3**	42.0	52.0	86.9	62.2	72.5	91.6	72.6	81.0	**38.7**	8.8	14.3	68.2	18.8	29.4	76.2	26.6	39.4
NeuralHaircut [14]	50.3	15.0	23.1	76.4	29.3	42.4	85.9	38.6	53.3	21.0	3.9	6.6	58.6	14.7	23.6	80.8	28.1	41.7
Ours	60.3	46.4	52.5	88.2	84.3	86.2	**94.5**	93.6	**94.1**	38.4	23.6	**29.2**	**79.2**	61.1	**69.0**	**90.1**	81.2	**85.4**
Ours (w/o DR)	65.4	41.8	51.0	**88.6**	78.8	83.4	93.3	88.6	90.9	22.3	15.9	18.6	60.0	56.8	58.4	77.5	83.0	80.1
Ours (w/o guide opt.)	61.1	46.8	**53.0**	86.8	86.1	**86.5**	92.7	95.0	93.9	36.9	22.9	28.3	77.5	60.7	68.1	88.8	81.0	84.7
Ours (w/o reparam.)	8.2	43.2	13.8	24.9	**97.4**	39.7	40.5	**99.9**	57.6	8.1	**33.5**	13.1	30.1	**95.5**	45.8	52.1	**99.9**	68.5
Ours (w/o $\mathscr{N}$)	59.5	**46.9**	52.5	86.6	85.4	86.0	92.8	94.3	93.5	36.5	23.2	28.4	76.4	60.8	67.7	87.6	80.9	84.1
Ours (w/o strand init.)	5.6	1.0	1.7	24.3	7.1	11.0	40.2	19.1	25.8	10.1	5.1	6.8	27.6	24.8	26.2	45.0	50.2	47.5
Ours (w/o global opt.)	58.1	45.7	51.2	85.5	85.8	85.7	91.8	94.8	93.3	28.5	15.4	20.0	68.9	51.1	58.7	82.6	74.8	78.5

w/o reparam., hair moves freely to improve recall value but causes noised shapes due to a lack of regularization. In **w/o** $\mathscr{N}$, individual strands move freely. In the case of curly hair, the strands are easily entangled in close observation and quickly become stuck in the local minima. The same behavior was observed for straight hair, but the uniformity of the hair flow had less negative impact on the numerical evaluation. In **w/o strand init.**, because initial strands are far from actual, hair growing stops in the middle of a stretch. Note that increasing the learning rate may improve hair growth, but shape collapse may also happen. In **w/o global opt.**, the boundary condition becomes heuristic, and all 3D orientations on surface are treated as downward facing, resulting in partially wrong, opposite guide hair flow. Even if the DR process allows 180° ambiguity, it will never be the correct orientation because the initial value will settle to the local minima of the direction it is facing. Turning off the individual modules causes reasonable degradation, which indicates that the effectiveness of each component of our pipeline is validated.

Next, Figure 3 illuminates the robustness against hair color. Hair color mainly affects the former part of our pipeline, such as raw mesh reconstruction and 2D/3D orientation estimation. Six colors with the same hair geometry were tested, and our method reconstructed similar, accurate strands for all colors, which indicates that our method can handle various hair colors.

In Table 2 of the main paper, a quantitative comparison was performed in 360° range to evaluate absolute hair flow with synthetic data [16]. To align with the criteria used in the previous papers [9, 12, 14], we show the evaluation tolerating 180° ambiguity in Table 1. Note that only evaluation metrics were updated, and the same geometries shown in Figure 1 and Figure 2 were used. The values of LPMVS and Strand Integration become better because they are not aware of 180° ambiguity. NeuralHaircut keeps the most values because it estimates the correct absolute hair direction in this case. Ours is still best in most values.

2.2. Additional results on studio data

To demonstrate our method's robustness on various real hairstyles, Figure 4 and Figure 5 show more results on studio data. In addition to final child hair, guide hair is also shown for each subject. The robustness of our framework against diverse natural hairstyles is proven.

2.3. Additional physics simulation results

In Figure 6, the physics simulations with head movement were compared in time series. The full sequence is available in the supplementary video. The natural behavior of our strands indicates that our method is capable of reconstructing simulation-ready hair strands.

2.4. Additional results on USC-HairSalon [5]

To address robustness against artistic hairstyles, we conducted experiments using USC-HairSalon [5]. A total of 58 synthetic images were generated following the same procedure as Cem Yuksel's hair models as input. Figure 7 illustrates that our approach adeptly reconstructs artistic hairstyles.

3. Anti-aliasing validation

3.1. Comparison with existing AAs

Our anti-aliasing (AA) for line segments is validated on a toy problem that grows a strand by DR. The toy problem

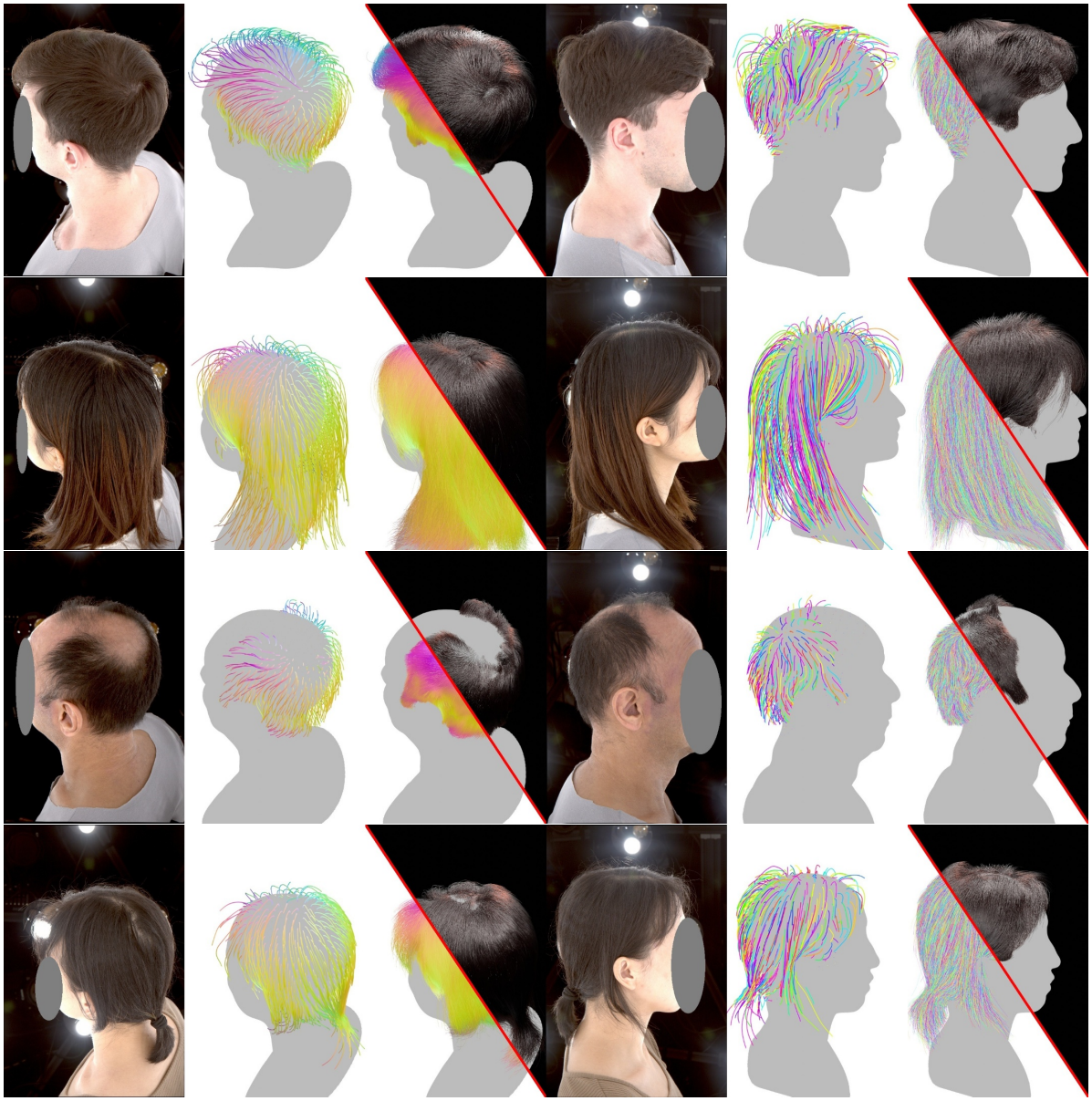


Figure 4. Additional results on studio data 1 / 2. From left to right, top view image, top view guide with 3D orientation, top view child with Blender Cycles shading and 3D orientation, side view image, side view guide with random color, and side view child with Blender Cycles shading and random color are shown. From top to bottom, the subjects with short hair, long hair, half bald head, and short-tied hair are displayed. Our method enables realistic reconstruction for a wide range of hairstyles in the wild.

is to fit a minimum line segment with two vertices, one at the root and one at the tip, into the target image, where the silhouette of a long strand is depicted. The line segment is initialized with the length in 20% of the target strand. Its root is fixed, and the tip position is optimized. To validate gradient quality itself, we used a simple optimizer, stochastic gradient descent without momentum. The learning rate was set to 1.0, and DR optimization with an L2 silhouette loss was performed in 25,000 iterations. Per iteration, the line segment is converted to a triangle, as shown in Figure 5 of the main paper. The triangle is rasterized in 128x128 pixels, and then each AA is applied.

In the experiments of the main paper, the width of our strand is set to 0.2 mm, which is often thinner than one pixel. So, in this validation, we tested root thickness in 1.0, 0.8, and 0.6 pixels. Thin width is prone to cause jumping pixels by the nature of rasterization.

Figure 9 shows the quantitative comparison with nvdiffrast [6] and splatting [4], which are AAs for meshes. Similar to ours, nvdiffrast is based on geometric AA, but

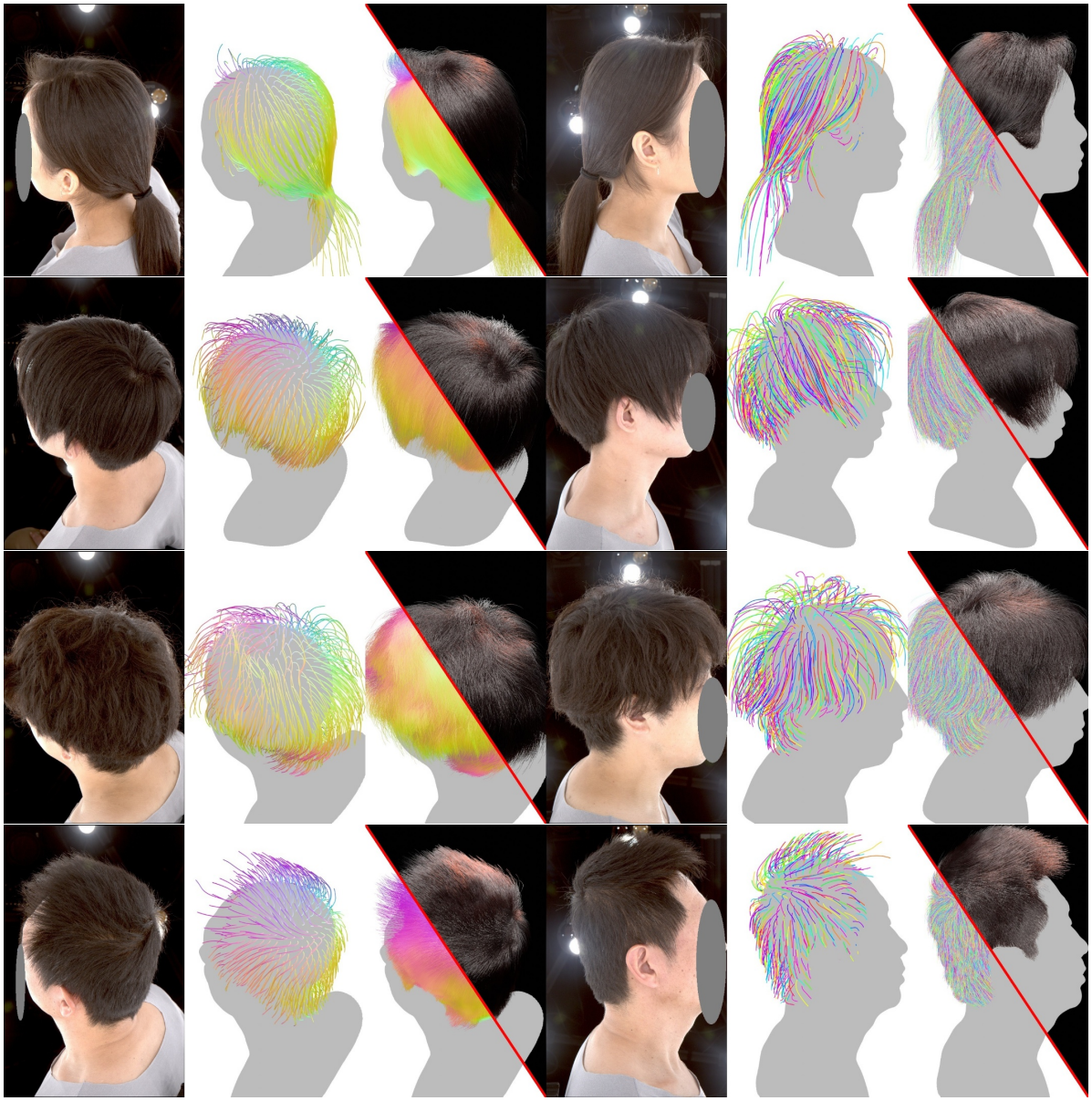


Figure 5. Additional results on studio data 2 / 2. From left to right, top view image, top view guide with 3D orientation, top view child with Blender Cycles shading and 3D orientation, side view image, side view guide with random color, and side view child with Blender Cycles shading and random color are shown. From top to bottom, the subjects with long-tied hair, short straight hair, wavy hair, stiff hair that is not squished by gravity are displayed. Our method enables realistic reconstruction for a wide range of hairstyles in the wild.

its gradient generation is selective. Splatting is another approach that propagates gradients through the weighted sum of neighbor pixels with differentiable screen space position interpolation. Our AA generates gradients for all line edges and propagates them to vertex positions via pixel-to-edge distance. Ours can reduce loss monotonically, while the other AAs show difficulty in handling tiny geometry. The qualitative comparison is displayed in Figure 8. Our AA generates a smooth gradient even with a very thin geometry, which leads to successful line segment alignment.

3.2. "CVPR" drawing by hair growing

Although the proposed pipeline utilizes the AA for fine-tuning following initialization, it possesses sufficient capability for growth. The depiction of the "CVPR" drawing with strands is presented in Figure 10. Each GT letter is constructed using one, two, two, and three bundles for "C", "V", "P", and "R" respectively, wherein a bundle comprises 500 strands with 50 segments functioning as child hair, each capable of independent movement. Target images were ren-

Figure 6. Comparison of physics simulation with head motion. Full sequences are available in the supplemental video. Starting from the reconstructed strands, gravity, hair stiffness and head rotation are applied to NeuralHaircut and ours. On the top two rows, the subjects of studio data in Figures 1 and 10 of the main paper are shown. The bottom row shows the subject of H3DS in the top row of Figure 7 of the main paper. Thanks to the correct hair growing direction, our hair shows more natural behavior under strong head movement. The original scalp shapes and hair root positions are kept while the head model is replaced for privacy protection.

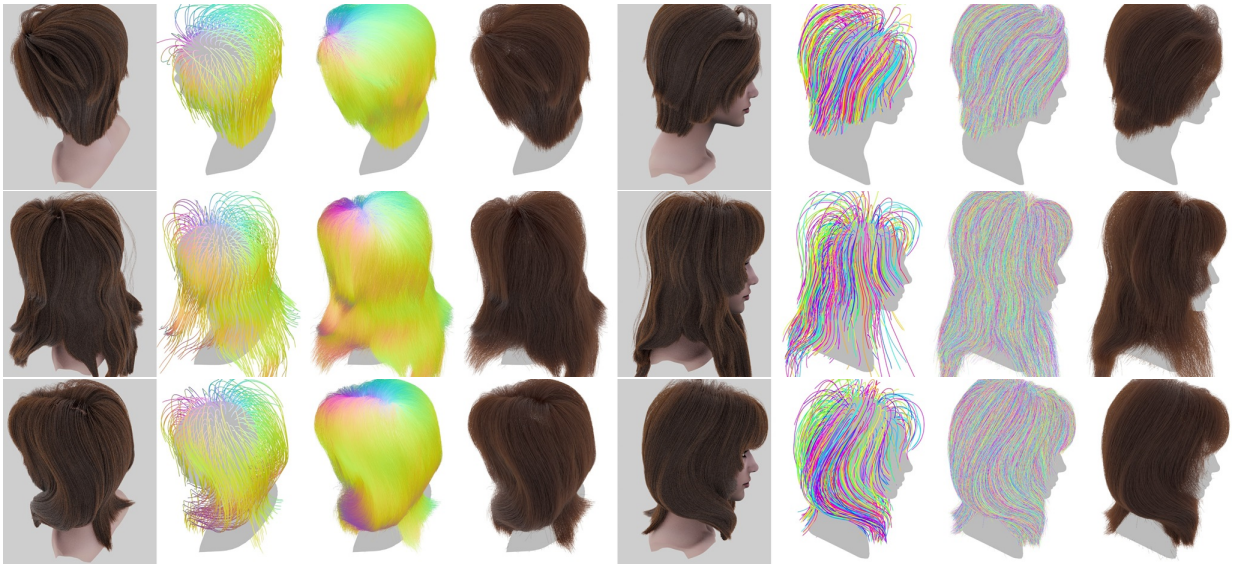
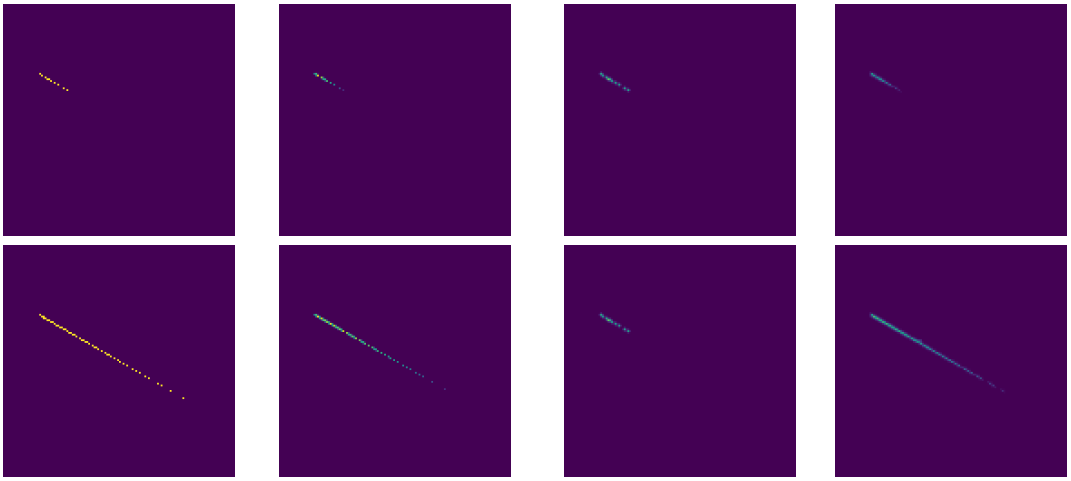
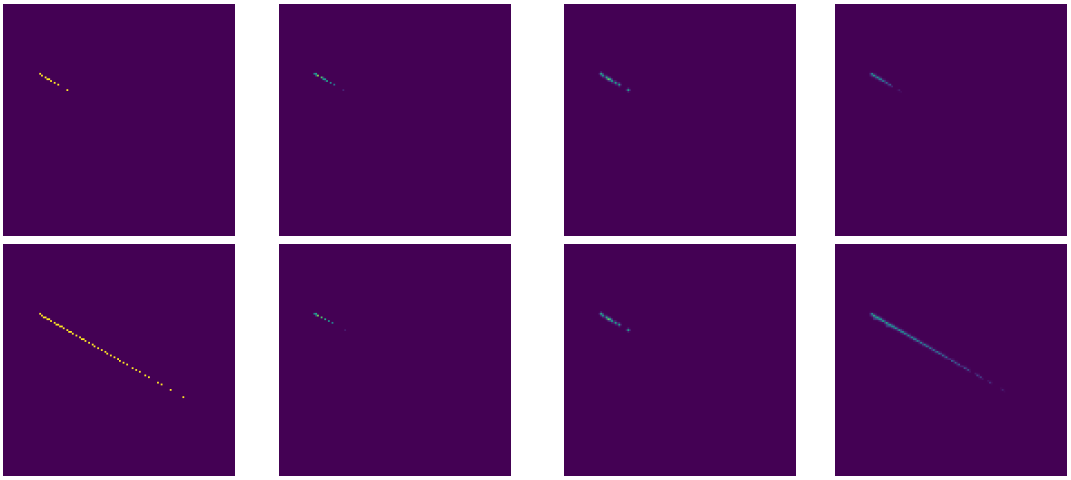


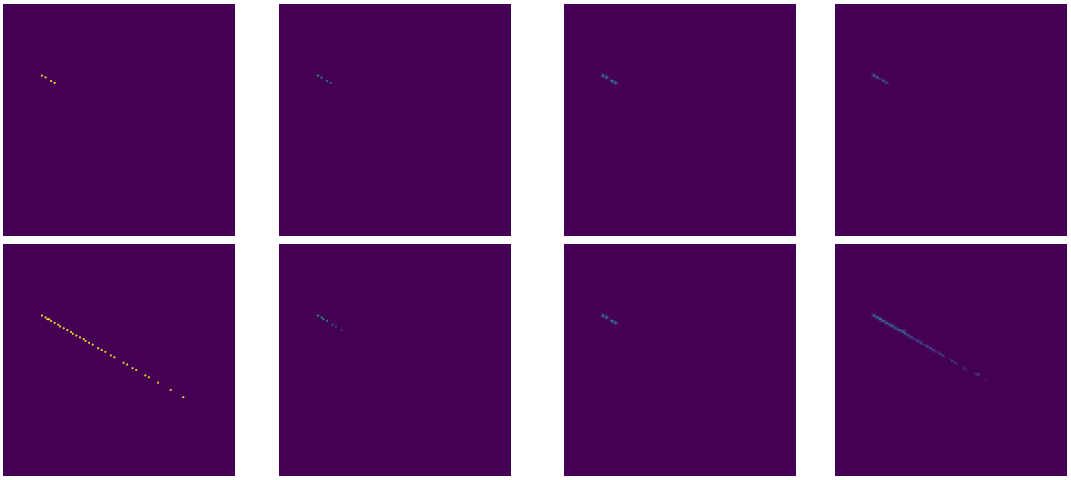
Figure 7. Additional results on USC-HairSalon [5]. From right to left, top view GT, top view guide with 3D orientation, top view child with 3D orientation, top view shaded child, side view GT, side view guide with random color, side view child with random color, and side view shaded child are shown. Our method successfully handles various hairstyles.



(a) 1.0 pixel width



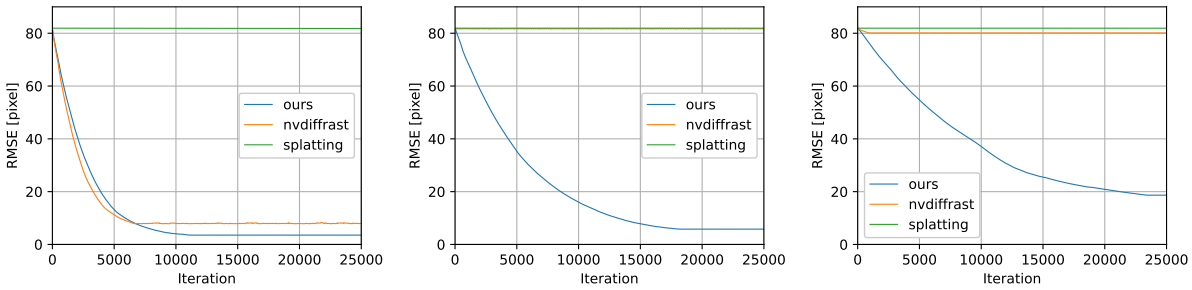
(b) 0.8 pixel width



(c) 0.6 pixel width

GT init / target **nvdiffrast** [6] init / final **splatting** [4] init / final **Ours** init / final

Figure 8. Qualitative validation of our AA. Digital zoom is recommended. The fixed root vertex is on the top left, and the tip vertex to be optimized is placed on the lower left of the root. Initial and final strands of various pixel widths are visualized. Our broader gradient compared to other AAs demonstrates that the strands grow even when the width is much narrower than one pixel.



(a) 1.0 pixel width (b) 0.8 pixel width (c) 0.6 pixel width

Figure 9. Quantitative validation of our AA. Error curves of various pixel widths are shown. The proposed method reduces errors in the long term, whereas the existing method is stuck in the early stages of optimization.

dered from random views of GT hairs on a per-letter basis. GT hairs were shortened to 10% of their original length as starting values for optimization utilizing DR. Each letter was optimized independently by Adam optimizer. The loss comprised L_m, L_o, R_{root}, R_c, alongside regularizer for equalizing segment lengths. For reparameterization, kNN with $k = 10$ was performed for $\mathscr{N}$ at 50% of the original length to address artifacts occurring at junctions. The "CVPR" drawing demonstrates that our AA can optimize complex shapes effectively.

4. Implementation details

4.1. Scalp fitting and hair region extraction

We describe the details of scalp fitting and hair region extraction, corresponding to **3.1. Initialization** of the main paper. We project semantic segmentation [7] onto a raw mesh from each view while extracting the hair region by vertex-wise voting. Facial landmarks [1] are also projected. Subsequently, 3D correspondences between the raw mesh and the head template model are established, and similarity transform is estimated using Umeyama's method [15]. Non-rigid registration by deforming vertices is then carried out. During the non-rigid registration process, as the raw mesh included hairs, but some scalp regions were not visible, we only considered regions other than the hair, such as the ears, face, and neck. More specifically, we rendered depth images from each view and optimized the vertex positions with reparameterization [10] via differentiable rendering to minimize an L1 depth loss within the facial area and an L2 3D landmark loss. At this stage, the scalp area might extend beyond the hair region in the raw mesh. To address this, we performed a post-process to push the scalp area into the raw mesh. Based on the same non-rigid registration framework, an L1 silhouette loss between the head model's scalp area and the raw mesh's hair region is minimized with a regularization term to keep facial depth values. The final scalp mesh is obtained from the scalp area of the head mesh through linear interpolation.

4.2. Detailed description of the scalp boundary condition

To clarify $S(p_s)$ in the Equation 2 of the main paper, Figure 11 illustrates the contrast between the scalp normal $n_s(p_s)$ and $S(p_s)$. Our $S(p_s)$ reflects the natural directions of scalp pores.

4.3. Detailed description of the motivation for reparameterization

We will explain our motivation of **3.2. Hierarchical Strand Optimization | Reparameterization** of the main paper in detail. Our hair is represented as a set of thin geometries of less than one pixel, and the visibility in screen-space is stochastic due to the nature of hardware rasterization. The outermost strands are not always rasterized, and the inner strands may show through. Naïve DR optimization can easily collapse hairstyles since it makes only the visible division points of strands move at each iteration as w/o reparam. in Figure 1 and Figure 2. On the other hand, reparameterization, countermeasures for sparse gradients in geometry, has been studied in the context of meshes [10]. We, therefore, proposed the reparameterization for line segments, where each Laplacian element has the following regularizing effect: $\mathscr{F}$: If a small part of a strand is visible, the whole strand moves smoothly based on the visible part; $\mathscr{N}$: Always, even if a strand is not visible at all, the strand moves smoothly based on the other visible strands in the neighborhood.

4.4. Hair mask for DR loss

At **3.2. Hierarchical Strand Optimization | Guide/Child Hair Optimization** of the main paper, we generate hair masks for L_m through the hair region of the raw mesh in a similar manner to NeuralStrands [12]. First, silhouettes are rendered with the hair region mesh onto each view. Because a multi-view voting scheme estimates the hair region mesh, our silhouette extraction is more tolerant of severe failures than applying 2D silhouette extraction to the input

(a) GT

(b) Initial strands (10% length of the GT)

(c) Intermediate strands

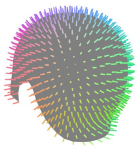
(d) Optimized strands

Figure 10. "CVPR" drawing by DR-based hair growing. The entire sequence is available in the supplementary video. The images were ray-traced by Blender Cycles after each letter was optimized independently.

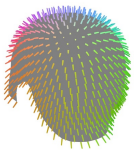
images individually. Then, the tri-map is made by erosion and dilation. KNN Matting [3] with the tri-map and an input color image is finally applied to generate an alpha hair mask.

4.5. Module-level performance measurement

We report relative time spent on each module in Table 2.



(a) Normals at scalp, $n_s(p_s)$



(b) Our $S(p_s)$

Figure 11. Comparison of scalp normal and our $S(p_s)$. (a) Normals at the scalp, $n_s(p_s)$. Growing directions at the side and back are different from real humans. (b) Our hair orientation at scalp $S(p_s)$. Strands can grow more naturally at the side and back.

Table 2. Time consumption ratio per module

Raw mesh reconstruction	4%
Scalp fitting	20%
3D Orientation estimation	18%
Strand initialization	5%
Hair mask generation	15%
Strand optimization by DR	38%

References

[1] Insightface, 2023. https://github.com/deepinsight/insightface. 10

[2] Agisoft. Metashape, 2023. https://www.agisoft.com/. 1

[3] Qifeng Chen, Dingzeyu Li, and Chi-Keung Tang. Knn matting. *IEEE transactions on pattern analysis and machine intelligence*, 35(9):2175–2188, 2013. 11

[4] Forrester Cole, Kyle Genova, Avneesh Sud, Daniel Vlasic, and Zhoutong Zhang. Differentiable surface rendering via non-differentiable sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6088–6097, 2021. 6, 9

[5] Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. Single-view hair modeling using a hairstyle database. *ACM Trans. Graph.*, 34(4), 2015. 5, 8

[6] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics*, 39(6), 2020. 6, 9

[7] Kunliang Liu, Ouk Choi, Jianming Wang, and Wonjun Hwang. Cdgnet: Class distribution guided network for human parsing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4473–4482, 2022. 10

[8] Ryota Maeda, Kenshi Takayama, and Takafumi Taketomi. Refinement of hair geometry by strand integration. *Computer Graphics Forum (proceedings of Pacific Graphics)*, 42 (7), 2023. 1, 5

[9] Giljoo Nam, Chenglei Wu, Min H. Kim, and Yaser Sheikh. Strand-accurate multi-view hair capture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 5

[10] Baptiste Nicolet, Alec Jacobson, and Wenzel Jakob. Large

steps in inverse rendering of geometry. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 40(6), 2021. 10

[11] Eduard Ramon, Gil Triginer, Janna Escur, Albert Pumarola, Jaime Garcia, Xavier Giro-i Nieto, and Francesc Moreno-Noguer. H3d-net: Few-shot high-fidelity 3d head reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5620–5629, 2021. 1

[12] Radu Alexandru Rosu, Shunsuke Saito, Ziyan Wang, Chenglei Wu, Sven Behnke, and Giljoo Nam. Neural strands: Learning hair geometry and appearance from multi-view images. In *ECCV*, 2022. 5, 10

[13] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[14] Vanessa Sklyarova, Jenya Chelishev, Andreea Dogaru, Igor Medvedev, Victor Lempitsky, and Egor Zakharov. Neural haircut: Prior-guided strand-based hair reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19762–19773, 2023. 1, 5

[15] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04): 376–380, 1991. 10

[16] Cem Yuksel, Scott Schaefer, and John Keyser. Hair meshes. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2009)*, 28(5):166:1–166:7, 2009. 1, 5