Shaping Strands with Neural Style Transfer

BEYZANUR COBAN, ETH Zürich, Switzerland
PASCAL CHANG, ETH Zürich & DisneyResearch|Studios, Switzerland
GUILHERME G. HAETINGER, ETH Zürich & DisneyResearch|Studios, Switzerland
JINGWEI TANG, DisneyResearch|Studios, Switzerland
VINICIUS C. AZEVEDO, DisneyResearch|Studios, Switzerland

Fig. 1. **Hair Stylization.** The baseline groom (left) and three variants generated by our strand-aware neural style transfer: a smooth spiral pattern, a tessellated triangle motif, and an angular spiral design. Each look is driven by a single 2D reference image and produced in minutes, mapping the painted strokes onto thousands of strands while keeping the volume and silhouette intact. This level of patterned detail would be impractically time-consuming to achieve with conventional grooming brushes.

The intricate geometric complexity of knots, tangles, dreads and clumps require sophisticated grooming systems that allow artists to both realistically model and artistically control fur and hair systems. Recent volumetric and 3D neural style transfer techniques provided a new paradigm of art directability, allowing artists to modify assets drastically with the use of single style images. However, these previous 3D neural stylization approaches were limited to volumes and meshes. In this paper we propose the first stylization pipeline to support hair and fur. Through a carefully tailored fur/hair representation, our approach allows complex, 3D consistent and temporally coherent grooms that are stylized using style images.

CCS Concepts: • **Computing methodologies** → *Physical simulation*; **Shape modeling**.

ACM Reference Format:
Beyzanur Coban, Pascal Chang, Guilherme G. Haetinger, Jingwei Tang, and Vinicius C. Azevedo. 2025. Shaping Strands with Neural Style Transfer. *ACM Trans. Graph.* 44, 6 (December 2025), 14 pages. https://doi.org/10.1145/3763365

Authors' addresses: Beyzanur Coban, bcoban@ethz.ch, ETH Zürich, Switzerland; Pascal Chang, pascal.chang@inf.ethz.ch, ETH Zürich & DisneyResearch|Studios, Switzerland; Guilherme G. Haetinger, haetinger@disneyresearch.com, ETH Zürich & DisneyResearch|Studios, Switzerland; Jingwei Tang, jingwei.tang@disneyresearch.com, DisneyResearch|Studios, Switzerland; Vinicius C. Azevedo, vinicius.azevedo@disneyresearch.com, DisneyResearch|Studios, Switzerland.

ACM 0730-0301/2025/12-ART
https://doi.org/10.1145/3763365

1 INTRODUCTION

The intricate geometric complexity of locks, tangles, dreads and clumps require sophisticated grooming [Iben et al. 2013; Hasenbring and Karlsson 2021; Jayaraman Mohan 2023] and simulation [Bergou et al. 2008] systems. Digital characters oftentimes carry tens of thousands — sometimes millions — of filaments forming silhouettes that require artistic control to convey a specific style or to enhance storytelling: may those be the choreographic acting beats of the complex hair motion in *Trolls* [Missey et al. 2017], or the plush, pillowy fur in *Turning Red* [Montell et al. 2022]. Current grooming pipelines arm artists with brushes, splines, shapes, and a myriad of strand-level settings; powerful controls, but oftentimes too granular that broad creative vision can be difficult to realize.

Recent breakthroughs in 3D Neural Style Transfer (NST) [Kim et al. 2019; Aurand et al. 2022; Gomes Haetinger et al. 2024] enabled a novel way to integrate artistic control: instead of low-level modifications, creators can efficiently deform 3D assets to match the aesthetics, patterns and motifs of a reference style image. These three dimensional style transfer pipelines have quickly found their way into production, enhancing the visual language of feature movies such as *Raya and the Last Dragon* [Navarro and Rice 2021], *Strange World* [Navarro 2023], *Wish* [Tollec and Navarro 2024], and, most prominently, used to stylize all fire characters of *Elemental* [Kanyuk et al. 2023; Hoffman et al. 2023].

In this paper we extend previous 3D Neural Style Transfer methods to work with digital grooms. Building on earlier volumetric and mesh stylization pipelines, we introduce a strand-aware neural style transfer algorithm that maps 2D style exemplars and motifs onto strands, producing groom-level deformations that remain coherent over time and over different views. Our proposed fur/hair NST extends previous pipelines with three contributions: a purpose-built volumetric displacement representation, a lightweight differentiable strand renderer, and a set of regularization techniques that give artists control over the output of the stylization. The result is a practical tool that lets production artists iterate complex hairstyles in novel ways that were not possible before.

2 RELATED WORK

Many previous works on hair and fur rendering [Kajiya and Kay 1989; Marschner et al. 2003; Wu and Yuksel 2017], modeling [Kim and Neumann 2002; Fu et al. 2007; Yuksel et al. 2009], and simulations [Anjyo et al. 1992; Bertails et al. 2006; Bergou et al. 2008; Müller et al. 2012] have been developed over the years, and excellent courses are available [Hadap et al. 2007]. Our work, more specifically, acts as a post-processing for hair/fur to modify previously groomed models and simulations. We notice that, however, our proposed method is the first to enable image-based neural style transfer on grooming setups. Therefore, the state-of-the-art discussions will focus on methodologies developed for related fields such as 2D and 3D Neural Style Transfer, art-directability and neural representations.

2D Neural Style Transfer. Pioneered by the seminal work of Gatys et al. [2016], Neural Style Transfer (NST) methods implement stylization as an iterative optimization in which the output image is steered by two complementary objectives: (1) a content loss that preserves the high-level Convolutional Neural Network (CNN) activations of the source image, and (2) a style loss that aligns second-order channel statistics (e.g., Gram-matrix correlations) to those of the reference style. Since its debut, image-based NST was extended for alternative style loss representations [Li et al. 2017a; Risser et al. 2017; Li et al. 2017b; Li and Wand 2016], improved temporal coherency in video stylization [Ruder et al. 2018; Jamriška et al. 2019; Ye et al. 2024], and improved performance by approximating the optimization problem through a feed-forward network [Huang and Belongie 2017]. Neural Neighbor Style Transfer (NNST) [Kolkin et al. 2022] method propose a different way to compute the style loss. Content features extracted from CNN activations are rearranged by finding nearest neighbour from the style features. Compared to gram-matrix correlations, NNST creates better style matching than the method of [Gatys et al. 2016]. Extracting style features from CNNs trained for classification tasks [Simonyan and Zisserman 2014; Szegedy et al. 2014] limits image style transfer capabilities. Thus, recent works [Chung et al. 2024a; Hertz et al. 2024; Rout et al. 2024] implement image-based stylization through text-to-image diffusion networks, achieving remarkable quality in stylization capacity and visual quality.

3D Neural Style Transfer. Besides images, several works extend stylization to three dimensions by iteratively optimizing 2D rendered views of the target asset, enabling the stylization of a variety of 3D assets such as meshes [Liu et al. 2018; Gomes Haetinger et al. 2024], implicit neural representations [Nguyen-Phuoc et al. 2022; Vu et al. 2025], particle systems [Cao et al. 2020; Zhang et al. 2024b], radiance fields [Jung et al. 2024] and volumetric simulations [Kim et al. 2019, 2020; Aurand et al. 2022]. The stylization of meshes was originally implemented by back-propagating gradients of a differentiable renderer by approximation [Kato et al. 2018] or computed analytically [Liu et al. 2018]. Customized energy-based formulations can be used for stylization without requiring neural networks: as-rigid-as-possible (ARAP) [Sorkine and Alexa 2007] deformations can be used to model cubic styles [liu and Jacobson 2019; Kohlbrenner et al. 2021]. Through a mesh Laplacian parametrization [Nicolet et al. 2021], Gomes Haetinger et al. [2024] lifted image-space gradients to 3D, enabling large silhouette-altering deformations that drastically improved the quality of mesh stylizations. Stylizing volumetric fluids comes with new challenges such as temporal coherency, efficiency and view consistency, and since the seminal transport-based NST (TNST) [Kim et al. 2019], subsequent work extended the method to particle-based [Kim et al. 2020] data, and accelerated inference with feed-forward 3D convolutional neural networks [Guo et al. 2021; Aurand et al. 2022; Wang et al. 2025]. Lastly, text-driven 3D stylization [Michel et al. 2022] offers artists a way to stylize meshes through text prompts, and recent works focused on improving control through GLIP [Li et al. 2022] embeddings [Chung et al. 2024b], tight integration between generative diffusion models and implicit representations [Yang et al. 2023] and regularizing deformations through a differentiable ARAP parametrization [Dinh et al. 2025]. For a more complete list of references regarding the stylization of 3D assets, we refer to [Chen et al. 2025].

Art-directability of hair and fur. Achieving set-piece moments — such as the swirling *hairnado* sequence in *Trolls* [Missey et al. 2017] — requires artists to creatively piece together multiple different modeling and simulation paradigms. Therefore several tools were developed to improve control over digital grooms, and early attempts were focused in adapting traditional 2D inputs such as sketches [Chen and Zhu 2006; Wither et al. 2007; Fu et al. 2007]. A more popular approach is to reduce the dimensionality of hair strands with custom-tailored controllable representations such as NURBS [Noble and Tang 2004], meshes [Yuksel et al. 2009], guide strands [Somasundaram 2015; Butts et al. 2018; Kaur et al. 2018] and tubes [Missey et al. 2017; Jayaraman Mohan 2023]. In our work we employ guide strands that reduce the groom dimensionality by selecting a few strands that control the fully discretized setup through interpolation. Several works try to improve the interpolation of guide strands through data-driven models [Wang et al. 2009; Chai et al. 2014; Xing et al. 2019], volumetric transport [Ghoniem and Museth 2013], inverse kinematics constraints [Missey et al. 2017], physically respecting interpolation [Hsu et al. 2024] and neural representations [He et al. 2024]. Lastly, special tools are required when targeting more specific hair/fur settings, such highly coiled hair [Wu et al. 2024], braids [Ogunseitan 2022], locks [Darke et al. 2024], and complex fur [Montell et al. 2022].

Neural Hair/Fur Representations. Neural methods can be employed as efficient data-driven primitives that can improve several downstream applications such as capturing [Wu et al. 2022; Sklyarova et al.

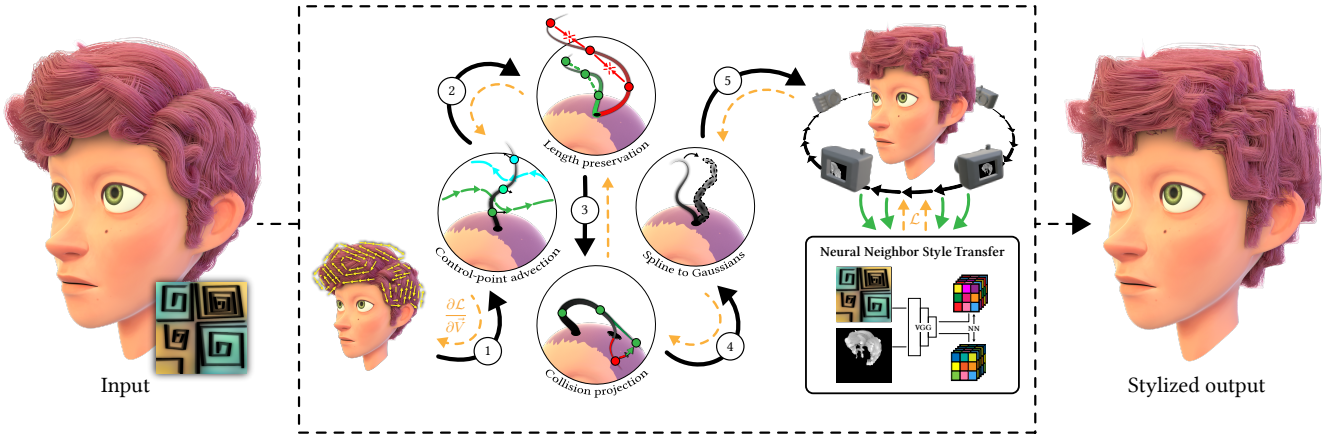


Fig. 2. **Hair and fur style transfer pipeline.** A learnable 3D velocity field first transports the spline control points of every strand. After advection and transporting the control points, we apply two strand-integrity constraints: length preservation and collision handling. The updated splines are then converted to a set of cylindrical Gaussians, and we render the groom differentiably via Gaussian splatting. The resulting image is matched to a single 2D style exemplar with a neural neighbor style-transfer module; gradients are back-propagated all the way back to the stylization deformation field.

2023], data-driven representations [He et al. 2024; Zhou et al. 2018; Saito et al. 2018] and rendering [Rosu et al. 2022]. Gaussian splatting was adopted for hair reconstruction as a lightweight differentiable rendering technique [Takimoto et al. 2024; Luo et al. 2024; Zakharov et al. 2025]. More recent works focus on adapting powerful diffusion backbones to generative hair setups: this can be accomplished by integrating Stable Diffusion [Rombach et al. 2022] into neural hair representations [Zhou et al. 2023; Sklyarova et al. 2024; Sun et al. 2024], working on generative pipelines with disentangled hair-head representations [Zhang et al. 2024a; Liu et al. 2024; Zhou et al. 2025], or creating new text-to-hair controls [Sklyarova et al. 2024; Sun et al. 2024].

3 METHOD

Our work follows a similar methodology as previous volumetric [Aurand et al. 2022; Kim et al. 2019], particle-based [Kim et al. 2020], and mesh style transfer [Gomes Haetinger et al. 2024] approaches. By rendering a set of Poisson-distributed views Θ around the target asset, a 2D image-based neural style transfer can be extended to generate 3D stylizations [Liu et al. 2018]. These renders are updated given a style image I_s by minimizing

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \sum_{\theta \in \Theta} \mathcal{L}(\mathcal{R}_\theta(\mathbf{x}), I_s), \tag{1}$$

where $\mathbf{x}$ is an arbitrary hair/fur parametrization, $\mathcal{R}$ represents a differentiable renderer, and θ is a camera configuration sampled from the set of camera views Θ. Cameras are placed on a spherical shell centered at the hair/fur object's centroid, with a configurable radius. The view set Θ is sampled from user-specified polar and azimuthal angle ranges using a Poisson-disk strategy, with a new camera selected at each optimization iteration. Broader angle ranges typically require more iterations to achieve adequate coverage.

Equation (1) is solved iteratively using a gradient-based optimization approach similar to the Neural Neighbor Style Transfer (NNST) method [Kolkin et al. 2022] to compute the style loss. NNST works by extracting features using a pre-trained classification [Simonyan and Zisserman 2014] network Φ from both content I_c and style I_s images. In our case, the content image is the hair/fur $\mathbf{x_0}$ rendered before stylization $I_c = \mathcal{R}_\theta(\mathbf{x_0})$. After zero centering both the content and style features, NNST utilizes a nearest-neighbor matching through cosine distance metric. Each hyper-column of content features is replaced by the closest style feature to obtain the target stylized features. Formally, at hyper-column i, the target stylized feature T is defined as

$$T_i = \arg\min_{\Phi(I_s)_j} \mathcal{D}_{cos}\left(\Phi(I_c)_i - \mu_c,\ \Phi(I_s)_j - \mu_s\right), \tag{2}$$

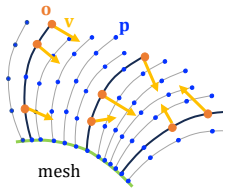
where $\mathcal{D}_{cos}$ denotes the cosine similarity, μ_c and μ_s are the average features of the content image and the style image respectively. The loss function can then be formulated as the L^2 distance between the features of the current render features $\Phi(\mathcal{R}_\theta)$ and the target stylized features T:

$$\mathcal{L} = \|\Phi(\mathcal{R}_\theta(\mathbf{x})) - T\|_2 . \tag{3}$$

Gomes Haetinger et al. [2024] showed NNST usually produces sharper results than computing secondary statistics such as the Gram Matrix [Gatys et al. 2016].

A simple way to parametrize strands is to represent them as natural cubic splines, and use the spline control points positions $\mathbf{p}$ as the optimization target $\mathbf{x}$. However, as we show in Figure 3, such option is sub-optimal as it creates high-frequency artifacts due to the lack of spatial smoothness. Thus, in this paper, we propose several novel contributions that enable an efficient hair/fur stylization pipeline: a novel volumetric displacement representation tailored for stylizing hair/fur data (Section 3.1); an efficient differentiable rendering algorithm for strands (Section 3.2); and a set of techniques to achieve temporal coherency (Section 3.3) and to enforce geometric constraints (Section 3.4).

3.1 Volumetric Hair/Fur Style Transfer



Following TNST [Kim et al. 2019], we adopt a volumetric velocity field to stylize individual hair/fur strands, ensuring that the stylization is both spatially and temporally smooth. Specifically, we optimize a velocity field $\mathbf{v}$ that is collocated with a subset of points uniformly sampled on the control point positions $\mathbf{p}$ of the original strands. We denote these positions for defining the velocity fields as $\mathbf{o}$. This setup is shown in the inset image. This contrasts with the original TNST [Kim et al. 2019], which uses a fixed volumetric grid for discretizing stylization velocities. Sampling velocity fields on hair strands simplifies the stylization, since a uniform volumetric velocity grid is prone to low voxel occupancy, and can also lead to artifacts at the geometry-hair interface. Our formulation transports the cubic spline control points by a velocity field during the stylization process as

$$\hat{\mathbf{v}} = \arg\min_{\mathbf{v}} \sum_{\theta \in \Theta} \mathcal{L}(\mathcal{R}_\theta(\mathcal{T}(\mathbf{p}, \mathbf{v})), I_s). \tag{4}$$

The transport function $\mathcal{T}$ in Equation (4) is computed by continuously sampling the volumetric velocity field with a grid-free interpolation function $\mathcal{I}$ [Müller et al. 2003] as

$$\mathcal{T}(\mathbf{p}_i, \mathbf{v}) = \mathbf{p}_i + \mathcal{I}(\mathbf{p}_i, \mathbf{v}), \quad \mathcal{I}(\mathbf{p}_i, \mathbf{v}) = \sum_{j \in \mathcal{N}} \phi(\|\mathbf{o}_j - \mathbf{p}_i\|_2^2)\mathbf{v}_j. \tag{5}$$

Here $\mathbf{o}_j$ denotes the $\mathcal{N}$ neighboring sampling locations defined in the vicinity of the i-th particle $\mathbf{p}_i$. $\mathbf{v}_j$ is the velocity field located at $\mathbf{o}_j$, and ϕ is an user-defined radial basis function kernel (e.g., cubic or inverse distance).

This formulation ensures that the resulting stylizations are spatially smooth, since the deformation at a strand is interpolated using neighboring velocity samples to update a given control point position. We highlight that Equation (5) discretizes the transport function by a simple forward Euler integrator with a timestep size of $\Delta t = 1$, since Aurand et al. [2022] demonstrates that higher order integrators make no difference to the quality of the stylization. Since $\mathbf{v}$ transports control points using a forward Euler integrator with a time step size $\Delta t = 1$, this velocity field can also be interpreted as a displacement/deformation field. The capacity of our volumetric discretization to produce smooth stylizations is demonstrated by comparing it against directly optimizing control points in Figure 3.

3.2 Differentiable Hair/Fur Rendering through 3D Gaussian Splatting

Gaussian Splatting [Kerbl et al. 2023] is a popular differentiable algorithm to render point primitives in an efficient manner. In the context of hair data, Luo et al. [2024] employed cylindrical 3D Gaussian primitives to more efficiently represent thin structures. In a similar spirit, we initialize cylindrical Gaussian splats along the cubic spline strands. To orient each Gaussian splat along its strand segment, a quaternion ($\mathbf{q_i}$) that rotates the canonical x-axis onto the spline's unit direction vector is computed. The size of each Gaussian splat is encoded by a diagonal scaling matrix that stretches it along the segment (s_{x_i}) and keeps a fixed, narrow cross-section (s_{y_i}, s_{z_i}). The $\Sigma_i^{3D} = 3 \times 3$ covariance matrix that represents a 3D cylindrical

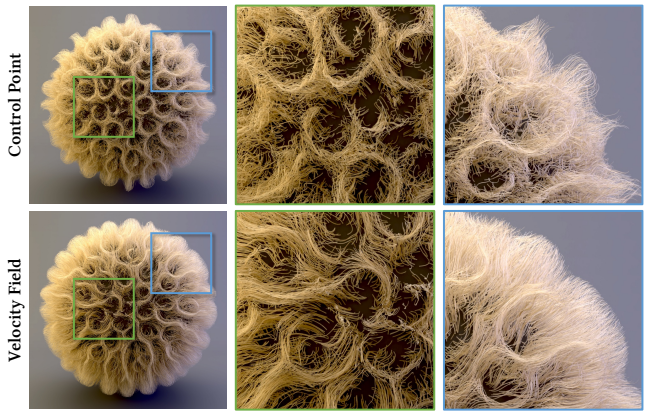


Fig. 3. **Comparison of different optimization parametrizations.** Control point parametrization creates high frequency artifacts while velocity field parametrization preserves the overall hair continuity structure.

Gaussian splat is

$$\Sigma_i^{3D} = \mathbf{R}(\mathbf{q}_i) \begin{bmatrix} s_{x_i}^2 & 0 & 0 \\ 0 & s_{y_i}^2 & 0 \\ 0 & 0 & s_{z_i}^2 \end{bmatrix} \mathbf{R}(\mathbf{q}_i)^\mathsf{T}, \tag{6}$$

where $\mathbf{R}(\mathbf{q}_i)$ represents the rotation matrix relative to the input quaternion.

Each 3D cylindrical Gaussian splat can be projected to two dimensions by computing the 2D covariance $\Sigma_i^{2D} = JW\Sigma_i^{3D}W^\mathsf{T}J^\mathsf{T}$, where J represents the Jacobian of the projective transformation, and W is the view transformation matrix. Then, the color of pixel value C is computed by aggregating the depth-ordered set of points r_i in a ray $\mathcal{R}$ that intersect with the Gaussian splat positions by

$$C = \sum_{r_i \in \mathcal{R}} T_i w_i c_i, \quad T_i = \prod_{j=1}^{i-1}(1 - w_j), \\ w_i = \alpha_i \, e^{-\frac{1}{2}(\mathbf{y}-\hat{\mu_i})^T(\Sigma_i^{2D})^{-1}(\mathbf{y}-\hat{\mu_i})}, \tag{7}$$

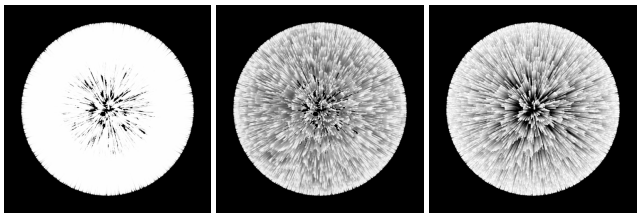
where $\mathbf{y}$ is the 2D image-plane sample location, $\hat{\mu}$ is the two dimensional Gaussian center projection onto the image plane, and α_i and c_i are the opacity and color values, respectively. In the next paragraphs, we discuss additional features of our differentiable render tailored to hair/fur style transfer.

Rendering Heuristics. Adding light ambient occlusion and transparency can help the optimization process to differentiate between distinct hair strands, as well as model natural strand color differences from root to the tip. Properly modeling colors due to light transport and scattering inside hair strands is a computationally intensive process [Marschner et al. 2003; Moon and Marschner 2006], and correctly modeling it would make our differentiable rendering unnecessarily expensive. To mimic ambient occlusion, we adopt a simpler heuristic, centered in the observation that hair regions closer to the root inherently receive less light, as these regions are densely populated due to neighboring strands. Therefore, we assign a darker color at the root and a brighter color to the tip. Similarly

to ambient occlusion, we include a translucency prior, by parametrically varying strand opacity along its length using fully opaque values at the root and transparent values at the tip. We compare the results of a simple differentiable render, a more sophisticated ambient occlusion algorithm [Jansson et al. 2019] and our rendering heuristics in Figure 4.

Visibility. Since we do not render the mesh that the hair/fur is attached to, strands that would be invisible during render time are visible to our differentiable render. To prevent these strands from being mistakenly optimized, we use a simple visibility calculation to remove those from our renders. We compute the visibility by checking if the dot product between the viewing direction and the normal direction of the mesh at the location of the strand root is smaller than a user-defined threshold ϵ. We allow this to be different than zero, because sometimes long strands can still be visible to the camera even if they originate from the back of the mesh.

Densification. A production groom typically contains only a few thousand *guide* strands, each standing in for hundreds of final hairs. Only rendering guide strands directly would introduce aliasing and visible gaps, whereas naively instancing the full groom would blow up memory during back-propagation. We therefore insert a lightweight densification stage that expands each guide into a compact set of proxy strands. Root positions are first randomly sampled on the underlying scalp mesh. For each sampled point, we identify its nearest guide-strand origins and compute barycentric weights inversely proportional to the squared Euclidean distances. These weights are then used to interpolate the control-point positions of the neighbors, producing a new proxy strand whose shape inherits more strongly from its closest guides. The stylization velocity fields are then used to update these newly created strands as well, which raise pixel coverage of the stylized rendering, providing the visual density expected of a full groom. We note that both the guide and dense strands are rendered to compute the style loss, but the optimization only targets the guide strands. The gradients on the dense strands only indirectly affect guide strands through the back-propagation process.



(a) No AO (b) Volumetric AO (c) Heuristic AO

Fig. 4. **Comparisons among different shading algorithms.** (a) The differentiable renderer uses no ambient occlusion, and since the fur is mostly opaque, the generated image produces saturated results. (b) By using a volumetric ambient occlusion algorithm [Jansson et al. 2019], images present meaningful color variations that can be picked up by the optimizer. (c) Using our simpler heuristic ambient occlusion, which produces results with similar qualities but in a much faster speed.

Algorithm 1: Strand stylization at frame t

Input:

- $\mathbf{p}_t$: control point positions
- $\hat{\mathbf{p}}_{t-1}$: optimized control point positions from last frame
- $\hat{\mathbf{v}}_{t-1}$: optimized velocity field from last frame
- $\mathbf{s}_{t-1}, \mathbf{s}_t$: simulation velocity
- $\mathcal{M}$: geometry mesh
- I_s: style image

Parameters:

Camera

- ψ, δ: polar and azimuthal angle ranges
- r: distance between the camera and the object centroid
- ϵ: visibility threshold
- w_c, h_c: render image size

Artistic Controls

- ρ: velocity field sampling density
- M: number of densified strands
- $\mathbf{s}^*_{t-1}, \mathbf{s}^*_t$: (optional) style flow

Multi-scale ▷ Each set has the cardinality of N_{scales}.

- W_s, H_s: set of style image sizes
- η: set of learning rates
- N_{iters}: set of number of iterations

Temporal coherency

- σ: EMA coefficient
- ν: EMA decaying period
- α: 1€ filter blending parameter
- β: 1€ filter scaling coefficient
- f_{min}: 1€ filter min. cut-off frequency

Output:

- $\hat{\mathbf{p}}_t$: optimized control point positions
- $\hat{\mathbf{v}}_t$: optimized velocity field

$\Theta \leftarrow \textsc{PoissonSample}(\psi, \delta)$ ▷ $|\Theta| = \textsc{Sum}\ (N_{iters})$
If $\mathbf{s}^*$ exists **then** $\mathbf{u} \leftarrow \mathbf{s}^*$ **Else** $\mathbf{u} \leftarrow \mathbf{s}$
$\mathbf{o}_t \leftarrow \textsc{SubsetSample}(\mathbf{p}_t, \rho)$ ▷ Sec. 3.1
$\mathbf{p}_t \leftarrow \textsc{DensifyStrands}(\mathbf{p}_t, M)$ ▷ Sec. 3.2
$\mathbf{v}_t(\mathbf{o}_t) \leftarrow \mathbf{0}$ ▷ Initialization at locations $\mathbf{o}_t$
for $\ell \leftarrow 1 : N_{scales}$ **do**
  for $m \leftarrow 1 : N^{\ell}_{iters}$ **do**
    $\mathbf{v}_t \leftarrow \text{EMA}(\mathbf{v}_t, \hat{\mathbf{v}}_{t-1}, \mathbf{u}_{t-1}, \nu, \sigma)$ ▷ Eq. 8
    $\mathbf{p}_t \leftarrow \textsc{Transport}(\mathbf{p}_t, \mathbf{v}_t)$ ▷ Eq. 5
    $\mathbf{p}_t \leftarrow \textsc{LengthPreserve}(\mathbf{p}_t)$ ▷ Sec. 3.4
    $\mathbf{p}_t \leftarrow \textsc{HandleCollisions}(\mathbf{p}_t, \mathcal{M})$ ▷ Sec. 3.4
    $\mathcal{R}_{\Theta^{\ell}_m}(\mathbf{p}_t) \leftarrow \textsc{Render}(\mathbf{p}_t, \Theta^{\ell}_m, r, \epsilon, h_c, w_c)$ ▷ Sec. 3.2
    $\hat{I}_s \leftarrow \textsc{Resize}(I_s, H^l_s, W^l_s)$
    $L \leftarrow \mathcal{L}\left(\mathcal{R}_{\Theta^{\ell}_m}(\mathbf{p}_t), \hat{I}_s\right)$ ▷ Eq. 4
    $\mathbf{v}_t \leftarrow \textsc{AdamW}(\mathbf{v}_t, L, \eta_\ell)$
  end
end
$\hat{\mathbf{v}}_t \leftarrow \mathbf{v}_t$
$\hat{\mathbf{p}}_t \leftarrow \textsc{OneEuro}(\mathbf{p}_t, \hat{\mathbf{p}}_{t-1}, \mathbf{u}_t, \mathbf{u}_{t-1}, \alpha, \beta, f_{min})$ ▷ Eq. 9, Eq. 10

3.3 Enforcing Temporal Coherency

Equation (4) defines a style loss for a single frame. Simply using it for a rendered animation sequence generates results that are not temporally coherent, since the resulting image-space NST optimization is highly sensitive to the input. One common approach to alleviate this issue is to align stylizations computed for individual frames with the velocities defined by the underlying animation [Kim et al. 2019; Aurand et al. 2022; Gomes Haetinger et al. 2024], and blend them with an exponential moving average (EMA) smoothing filter. Applying this procedure for the volumetric deformation velocities of Equation (4) yields:

$$\begin{aligned} &\hat{\mathbf{v}}^t = (1 - f(\nu, \sigma))\, \mathbf{v}^t + f(\nu, \sigma)\, \mathcal{T}(\mathbf{v}^{t-1}, \mathbf{u}^{t-1}), \\ &\text{where } f(\nu, \sigma) = \left(1 - \min(1, \frac{m}{\nu})\right) \sigma. \end{aligned} \tag{8}$$

Here $\mathbf{u}^t$ represents the control point velocity at timestep t and $f(\nu, \sigma)$ represents the EMA blending parameter, m denotes the optimization iteration step, σ is the smoothing weight and $\nu > 0$ modulates EMA amplification factor. The function $f(\nu, \alpha)$ modulates how the smoothing is applied throughout the optimization: as iterations progress the EMA smoothing is gradually decreased. We also highlight that the same transport function $\mathcal{T}$ previously defined can also be used to transport velocity fields between time-steps.

While previous EMA-based NST pipelines [Aurand et al. 2022; Gomes Haetinger et al. 2024] are able to produce temporally coherent stylizations for their target 3D assets, we found that simply translating their approach produces sub-par results when stylizing hair/fur. This happens because dynamic modes in the animation can have distinct oscillations that contribute to the temporal coherency differently. This difference can translate to jitters if one uses a naive EMA filtering.

Like [Kim et al. 2020], we apply a per-point filtering to the stylization deformations that is similar to the 1€ filter [Casiez et al. 2012]. Our newly proposed strand-aware temporal coherency first computes the weighted average between the point velocities and control point positions for the i−th location after all optimization iterations at timestep t:

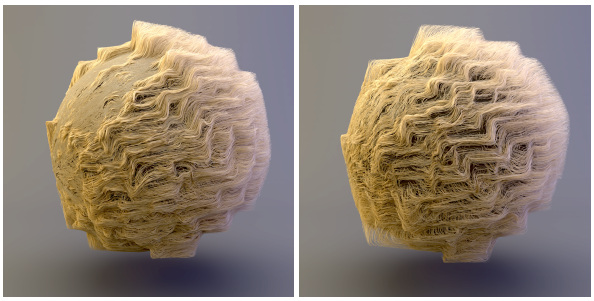
$$\hat{\mathbf{u}}_i^t = \alpha \mathbf{u}_i^t + (1 - \alpha)\hat{\mathbf{u}}_i^{t-1}, \quad \hat{\mathbf{p}}_i^t = \gamma_i \mathbf{p}_i^t + (1 - \gamma_i)\hat{\mathbf{p}}_i^{t-1}, \tag{9}$$

with α is a constant blending coefficient, and γ_i is a per-point smoothing coefficient. This per-point coefficient allows different strands to be smoothed out with the previous time-steps positions at different rates, depending on how much the underlying animation is changing over time.

The smoothing coefficient computed for the i-th location is

$$\gamma_i = \frac{2\pi\omega_i \Delta t}{2\pi\omega_i \Delta t + 1}, \quad \text{where } \omega_i = f_{min} + \beta \left\| \hat{\mathbf{u}}_i^t \right\|. \tag{10}$$

Here $\Delta t = 1/\text{FPS}$, f_{min} and β are the user-defined minimum cut-off frequency and scaling factor, respectively. The 1€ filter acts effectively as a spatially-varying treatment on the temporal smoothness, which depends on the per-point velocity magnitude $\left\| \hat{\mathbf{u}}_i^t \right\|$. Fast-moving points receive a higher γ_i, which decreases the influence of the temporal coherence from the previous frame and allows the particle to better follow rapid motion. Conversely, near-stationary points receive a lower γ_i, resulting in stronger temporal smoothing and greater visual stability.



(a) w/o thresholding (b) w/ thresholding

Fig. 5. **Effect of projection threshold in collision handling.** (a) Collision handling without projection threshold, where penetrated control points are projected to the closest mesh surface. (b) Collision handling with a projection threshold. Snapping artifacts are eliminated, producing more physically plausible collision handling.

3.4 Geometry-Aware Regularization Controls

While our differentiable pipeline can sculpt a groom into visually compelling new shapes, unconstrained gradient descent is prone to artifacts: strands may stretch far beyond their rest length or tunnel through the mesh. To remedy these issues, we augment our stylization loss with physically motivated constraints. In practice we employ two complementary constraints —collision handling and length preservation — which together keep the groom plausible yet responsive to style cues.

Collision Handling. We reduce collisions due to the stylization deformations by projecting penetrating control points back onto the mesh and folding that correction directly into the stylization offset. After each stylization iteration, we map every penetrating strand control point $\mathbf{p}$ onto the closest surface point $\mathbf{x}$ on the triangle mesh using a pre-computed BVH. If the signed dot product between the collision direction and the corresponding closest point mesh normal $\mathbf{n}$ is smaller than zero (i.e., the vertex lies inside the mesh half-space), we replace the stylization displacement with a projected offset. Formally, we denote $\mathbf{p}^0$ as the undeformed control point positions, so the stylization update as $\Delta\mathbf{p} = \mathbf{p} - \mathbf{p}^0$. The collision-free update for the point is

$$\Delta\mathbf{p}^* = \Delta\mathbf{p} - (\mathbf{p} - \mathbf{x}), \quad \text{if } (\mathbf{p} - \mathbf{x}) \cdot \mathbf{n} < 0. \tag{11}$$

Projecting penetrating control points directly onto the surface causes the strands to snap abruptly to the mesh, as shown in Figure 5 (a). A straightforward improvement is to introduce a projection threshold τ:

$$\begin{aligned} &\Delta\mathbf{p}^* = \Delta\mathbf{p} + \kappa\,(\tau - \|\mathbf{d}\|)\,\frac{\mathbf{d}}{\|\mathbf{d}\|}, \quad \text{if } (\mathbf{p} - \mathbf{x}) \cdot \mathbf{n} < \tau, \\ &\text{with } \mathbf{d} = \begin{cases} \mathbf{p} - \mathbf{x}, & \text{if } (\mathbf{p} - \mathbf{x}) \cdot \mathbf{n} < 0, \\ \mathbf{x} - \mathbf{p}, & \text{otherwise.} \end{cases} \end{aligned} \tag{12}$$

Fig. 6. **Short hair view-independent style transfer.** The short hair model is stylized statically.

Fig. 7. **Long hair style transfer.** The long hair model is simulated with time varying wind and gravity. We show the stylization results for the first frame.

We set the threshold $\tau > 0$ to proactively repel points that are outside the mesh but are close to the surface. We also add a stiffness coefficient κ to control the magnitude of the projection.

In practice this projection operator eliminates strand–mesh intersections without introducing stiff penalty weights or extra energy terms, preserving convergence speed while keeping the deformation field free of mesh penetrations. The threshold effectively projects the strand control points away from the mesh surface. It resolves the snapping issue when no thresholding is used, and is thus adopted in all our examples. The comparison is shown in Figure 5.

Length Preservation. Hair plays a key role in defining a character's visual identity through its silhouette and dynamic motion, both of which are sensitive to strand length. To prevent stretching or shrinking, we constrain deformations to closely follow the rest length measured on the input groom. During the optimization process, when transporting the control points in Equation (5), each hair strand is traversed from the root to the tip. We compute the distance between two consecutive control points and project their length to match the length of the original input groom. Since we use a natural cubic spline that entirely depends on the control points positions, the simplification of preserving the distances between the control points works well. Combined with the collision term, this length regularizer keeps the groom's original size intact while still allowing rich surface-level stylizations.

4 EXPERIMENTS AND RESULTS

The proposed stylization pipeline for hair and fur is implemented in PyTorch [Paszke et al. 2019]. At each optimization iteration, a perspective camera is sampled using spherical coordinates within predefined intervals. Following Gomes Haetinger et al. [2024], we facilitate the synthesis of multi-scale, coarse-to-fine structures, we adopt a hierarchical scheme in which the model is optimized sequentially across different image scales. We denote the number of scales as N_{scales}, which is set as $N_{\text{scales}} = 3$ for all our examples. At each scale, the style image is resized to emphasize patterns characteristic of that resolution. Correspondingly, the number of optimization iterations is incremented at finer scales to ensure convergence and detail refinement. Detailed configurations for each example can be seen in Table 1. We use collision handling and length preservation regularizations from Section 3.4 in all our examples. The groom models and simulations (Figure 8, 6, 9, 7, 12) are created in Blender and Houdini, and rendered with Houdini's Karma renderer. We use a NVIDIA RTX 3090 GPU for all our experiments. We refer to the supplemental video for a more comprehensive visualization of

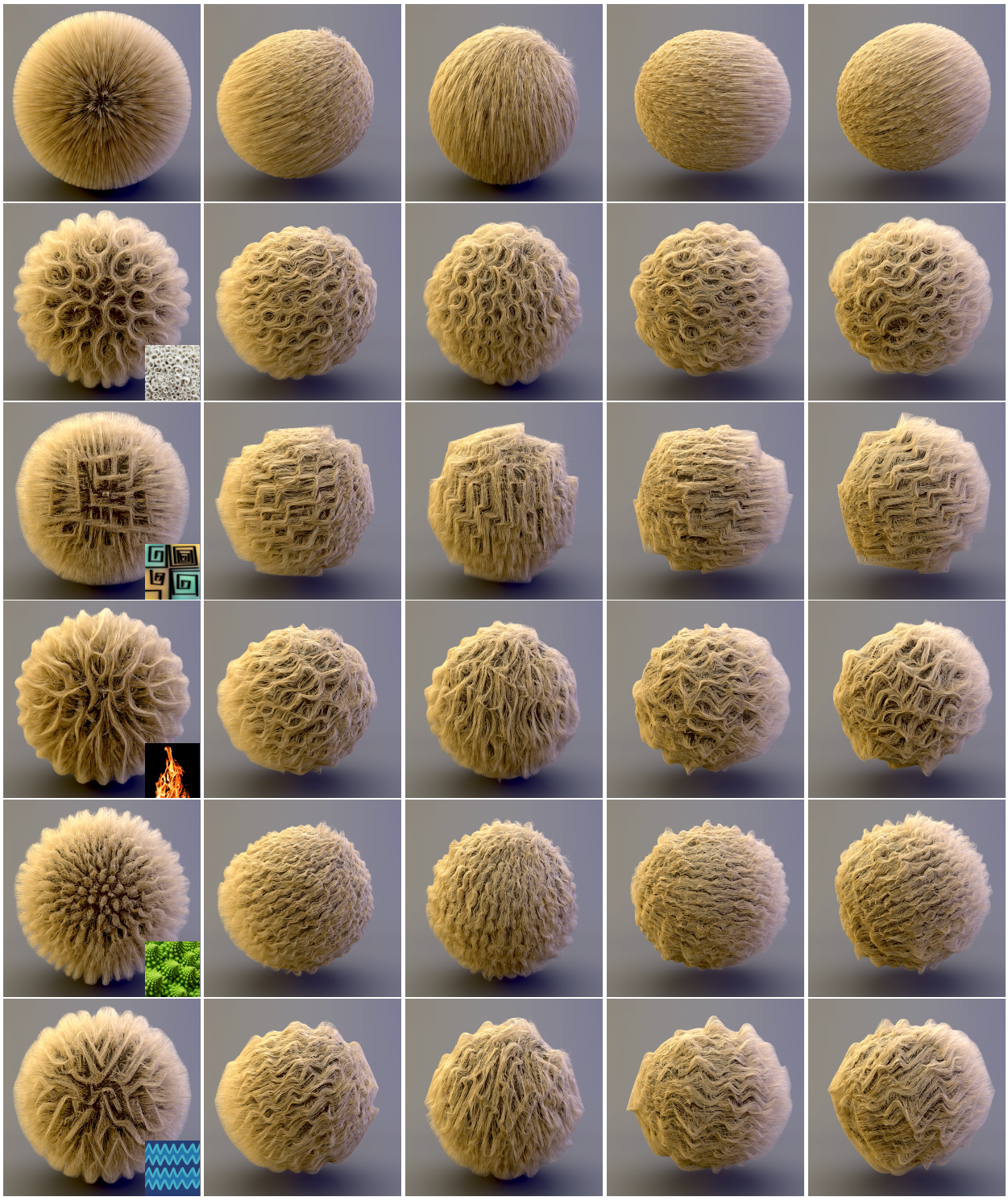
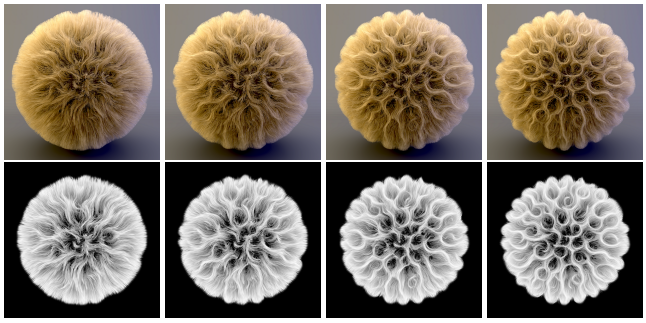


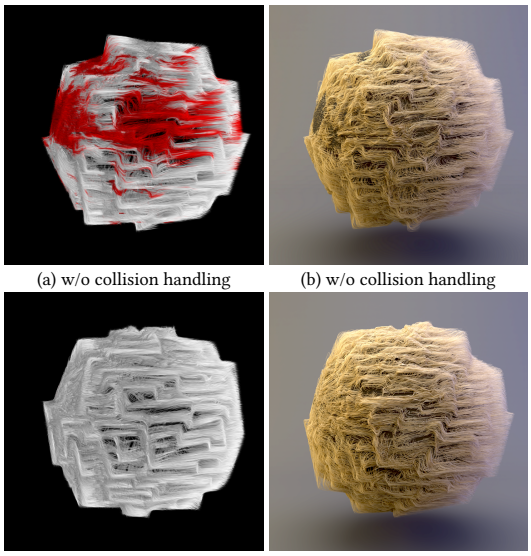
Fig. 8. **Fur ball style transfer.** Furs are initialized on a sphere surface in the normal direction with uniform lengths. A time varying wind is applied in the fur simulation.

Fig. 9. **Cat fur style transfer.** The furs on the cat body below the neck are stylized statically.



(a) $k = 9000$ (b) $k = 18000$ (c) $k = 36000$ (d) $k = 72000$

Fig. 10. **Effect of velocity field samples on the style.** Lower number of velocity field samples k leads to smoother stylizations (a) (b) while higher number of samples captures better high-frequency details (c) (d).



(a) w/o collision handling (b) w/o collision handling

(c) w/ collision handling (d) w/ collision handling

Fig. 11. **Collision handling.** We visualize the penetrating strands in red in the differentiable rendering setting (a). The final rendering (b) of the same result shows up the inter-penetrations. Our in-optimization collision handling effectively removes collisions as shown in both the differentiable rendering (c) and the final rendering (d).

the animated sequences. Algorithm 1 provides a comprehensive overview of the steps in our hair/fur stylization pipeline. No prior work performs image-based neural stylization on hair or fur grooms. Existing 3D NST methods target meshes, volumes, or implicit representations, making them unsuitable as baselines. We therefore evaluate our method solely via an ablation study of our pipeline's key components.

4.1 Ablation Studies

We validate the different features and parameters of our method, including collision handling, length preservation and temporal coherency. Additional ablations related to temporal coherency are provided in the supplementary video.

Velocity field resolution. Our proposed volumetric velocity field parametrization is essential to produce smooth stylization flows over the strands, as we have shown in Figure 3. The number of optimizable velocity field samples, which are always collocated with a subset of the strand control points, is a controllable hyperparameter of our method. In Figure 10, we illustrate the effect of varying this parameter for the fur ball scene using the spiral style: as the number of samples k decreases from (d) to (a), the stylization gets less capable of capturing high-frequency details of the reference style.

Geometry-Aware Constraints. Figure 11 demonstrates the effect of applying collision constraints. Fur strands penetrating the mesh surface at a given frame of the dynamic simulation are visualized in red in (a) and (c). Without collision handling (a), this becomes a significant issue, which translates into bald spots in the rendered final result (b), as can be seen in the left part of the sphere. Figure 13 illustrates the effect of applying length preservation in the grass example. Deviations from the original strand length are visualized using increasingly darker shades of red (positive deviation) and blue (negative deviation). Without length preservation, strands may undergo extreme deformations to match the style in (a) and (b), which, while potentially enhancing stylization, compromises the realism of the result.

Temporal Coherency. In the accompanying supplementary video we demonstrate the effect of using the volumetric EMA smoothing and our new strand-aware approach. Not using any temporal smoothness techniques results in animations that exhibit pronounced jitteriness. While a simple volumetric EMA smoothing can partially deal with this problem, we demonstrate that our strand-aware approach produces the most visually compelling results.

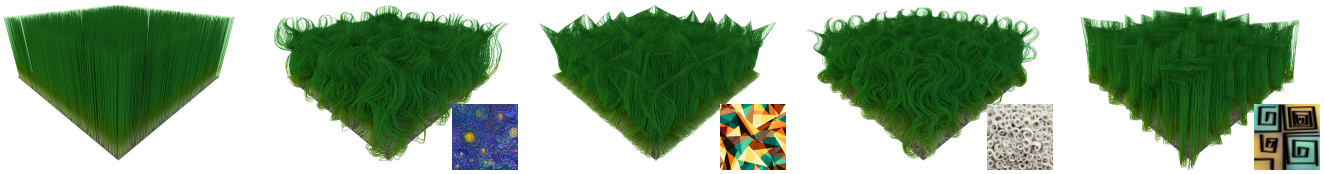
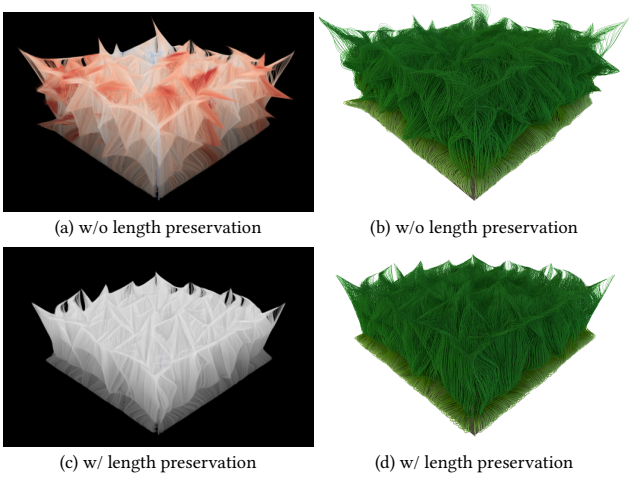


Fig. 12. **Grass patch style transfer.** The grasses are initialized as vertical unit-length strands on a plane. The grasses are simulated as fur with time varying wind forces. We show the first frame style transfer results.



(a) w/o length preservation

(b) w/o length preservation

(c) w/ length preservation

(d) w/ length preservation

Fig. 13. **Length preservation.** We use a colormap in (a) and (c) to where elongated strands are marked as red and shortened strands as blue. Without length preservation, most of the strands change their length in (a) and (b). When length preservation is used, the strand lengths got preserved and the stylization quality is least affected (c) (d).

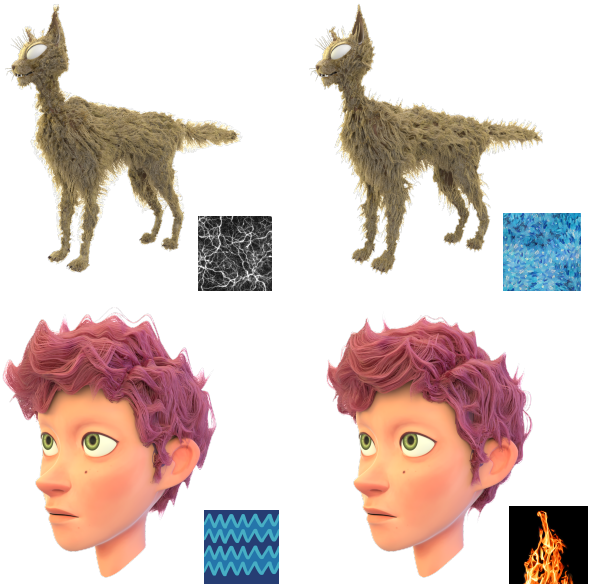


Fig. 14. **Failure cases.** The first row illustrate that highly detailed style image patterns is difficult to be faithfully captured in the final renderings. The second row demonstrates that certain style images produce similar results due to the shared prominent visual features.

Moreover, our results specify two different setups for point velocities $\mathbf{u}^t$: they can be chosen to follow the underlying simulation velocity, or defined by a user-defined style flow field. In the former case, the style pattern typically follows the natural hair movement. In the latter, the user-defined velocity causes the style pattern itself to drift over the hair surface towards the specified direction, producing the impression that the style is flowing through the groom.

The parameters of the 1€ filter also offer additional stylization control. A higher cut-off frequency f_{min} is able to preserve detailed style textures, while lower f_{min} yield smoother transitions. The β parameter controls how the stylization is able to track rapid changes in previous frames. Smaller β values will create a more uniform response throughout the groom, and different strands will respond to the EMA smoothness more similarly. The derivative smoothing coefficient α balances responsiveness and robustness: moderate values enable useful adaptivity without amplifying noise, whereas very small or very large values could under- or over-react. Since our examples, the per-point velocity $\mathbf{u}$ do not change much over time, the effect of α is not obvious in the results. We demonstrate the ablations of each parameter in the supplementary video.

4.2 Style Transfer Results

Figures 1, 7, 9, 12 showcase our fur stylization method applied under a small camera angle range. The results span a variety of 3D assets, including fur, hair, and grass-like structures, and demonstrate the method's versatility and ability to capture a broad range of styles. Additionally, in Figure 6, we show examples of view-independent hair stylization. This is achieved by sampling camera positions along a 360° orbit around the object to ensure that stylization is visible from any viewing angle.

In addition to static stylizations, our method produces temporally stable results on animated sequences involving hair and fur, achieved through a combination of EMA smoothing and a 1€ filter (Section 3.3). Figure 8 depicts a dynamic scene in which a fur ball is subjected to wind with varying direction. Although the original simulation causes the fur to lie very close to the mesh surface after the first frame, our stylization method still generates appealing deformations that capture the essence of the target styles, even in this challenging scenario. We refer to our supplemental video for other dynamic scene results.

Table 1. **Model parameters and performance statistics.** We use an NVIDIA RTX 3090 GPU for all our experiments. For the static models, the temporal coherency parameters are omitted.

Example	Guide Strands	Densified Strands	Control Points	Velocity Samples	Diff Render Resolution	No. of Iterations	Style Image Sizes	Learning Rates
		M			$w_c \times h_c$	N_{iters}	$[W_s \times H_s]$	η
Short Hair (Fig. 1, 14)	4 K	22.6 K	2.66 M	266 K	600×600	[20, 30, 50]	$[160 \times 160, 240 \times 240, 300 \times 300]$	[0.0026, 0.0016, 0.0009]
Short Hair 360 (Fig. 6)	4 K	22.6 K	2.66 M	266 K	600×600	[70, 100, 170]	$[160 \times 160, 240 \times 240, 300 \times 300]$	[0.0026, 0.0016, 0.0009]
Long Hair (Fig. 7)	8.18 K	4.12 K	1.23 M	409 K	600×600	[20, 30, 50]	$[200 \times 200, 240 \times 240, 320 \times 320]$	[0.0034, 0.0025, 0.0016]
Cat (Fig. 9, 14)	75.75 K	177.25 K	2 M	202 K	800×800	[20, 30, 50]	$[80 \times 80, 160 \times 160, 260 \times 260]$	[0.002, 0.0012, 0.0006]
Fur Ball (Fig. 8, 11)	8 K	42 K	450 K	72 K	500×500	[20, 30, 50]	$[30 \times 30, 150 \times 150, 240 \times 240]$	[0.0025, 0.0018, 0.0009]
Grass Patch (Fig 12, 13)	3 K	9 K	108 K	18 K	600×600	[20, 30, 50]	$[120 \times 120, 180 \times 180, 225 \times 225]$	[0.0028, 0.002, 0.0012]

Example	Polar Angle Range	Azimuthal Angle Range	Camera Distance (m)	Visibility Threshold	Vel. Field Sampling Density	EMA Coefficient	EMA Decaying Period	1€ filter Blending Parameter	1€ filter Scaling Coefficient	1€ filter Min. Cut-off Frequency (Hz)	Runtime (s/frame)
	ψ	δ	r	ϵ	ρ	σ	ν	α	β	f_{min}	
Short Hair (Fig. 1, 14)	[−10, 10]	[−10, 10]	5	0.3	0.665	-	-	-	-	-	138
Short Hair 360 (Fig. 6)	[−20, 20]	[−180, 180]	5	0.3	0.665	-	-	-	-	-	512
Long Hair (Fig. 7)	[−20, 20]	[−10, 10]	2.5	0.6	0.5	1	0.7	0.3	0.25	8.0	149
Cat (Fig. 9, 14)	[−10, 10]	[−10, 10]	2	0.2	0.333	-	-	-	-	-	179
Fur Ball (Fig. 8, 11)	[−15, 15]	[−15, 15]	3	0.5	1.0	1	0.6	0.34	0.2	9.0	53
Grass Patch (Fig 12, 13)	[−10, 10]	[−10, 10]	2	0.5	0.666	1	0.6	0.34	0.3	9.0	26

5 CONCLUSIONS

In this paper we introduced a strand-aware neural style transfer algorithm that brings the expressive power of single-image stylization to digital grooms. Our pipeline (i) represents each groom as a volumetric displacement field anchored to sparse guide strands, (ii) employs a lightweight differentiable strand renderer with densification that is tailored for style transfer optimization, and (iii) couples the style objective with strand-integrity constraints that enforce collision-free, length-preserving deformations. Taken together, these components allow artists to project hand-painted motifs, graphic patterns, or painterly strokes directly onto tens of thousands of hairs or fur fibers while maintaining temporal and viewpoint coherence.

Our method still has some limitations. First, the current implementation accepts only a single global style image; spatially varying style maps, orientation fields [Gomes Haetinger et al. 2024] or text-prompt diffusion guidance are left for future work. Second, while the proxy renderer is memory-efficient, ultra-dense, film-quality grooms can still overwhelm GPU memory at very high output resolutions. Third, we operate purely on geometry, so strand colours are untouched and multihued hair/fur stylization is not supported. Fourth, the collision handling critically depends on the SDF gradient field, since repulsion directions for each control point are sampled from it. Around strong non-convex regions, the gradient of SDF can be ambiguous or discontinuous due to resolution limits. It may produce misdirected or unstable repulsive forces. Finally, in Figure 14, we observe styles containing dense, fine-grained structure are difficult to be captured faithfully, causing the rendered results to lose characteristic patterning, and styles that are visually dominated by the same salient elements frequently produce similar outputs.

Despite these challenges, we believe the technique represents an important step towards closing the gap between 2D concept art and 3D grooming. By turning a style image into a direct, differentiable control knob, the method empowers artists to iterate rapidly, explore bolder looks, and deliver more distinctive characters. So the lays a foundation for ever richer visual storytelling in animation, VFX, and real-time experiences.

ACKNOWLEDGMENTS

We thank Sergio Sancho for his valuable insights and his help with the implementation of the collision handling methods.

REFERENCES

Ken-ichi Anjyo, Yoshiaki Usami, and Tsuneya Kurihara. 1992. A simple method for extracting the natural beauty of hair. *SIGGRAPH Comput. Graph.* 26, 2 (July 1992), 111–120. https://doi.org/10.1145/142920.134021

Joshua Aurand, Raphael Ortiz, Silvia Nauer, and Vinicius C. Azevedo. 2022. Efficient Neural Style Transfer for Volumetric Simulations. *ACM Trans. Graph.* 41, 6 (Nov. 2022), 257:1–257:10. https://doi.org/10.1145/3550454.3555517

Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. 2008. Discrete elastic rods. In *ACM SIGGRAPH 2008 papers (SIGGRAPH '08)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/1399504.1360662

Florence Bertails, Basile Audoly, Marie-Paule Cani, Bernard Querleux, Frédéric Leroy, and Jean-Luc Lévêque. 2006. Super-helices for predicting the dynamics of natural hair. *ACM Trans. Graph.* 25, 3 (July 2006), 1180–1187. https://doi.org/10.1145/1141911.1142012

Andrew Butts, Ben Porter, Dirk Van Gelder, Mark Hessler, Venkateswaran Krishna, and Gary Monheit. 2018. Engineering full-fidelity hair for Incredibles 2. In *ACM SIGGRAPH 2018 Talks*. ACM, Vancouver British Columbia Canada, 1–2. https://doi.org/10.1145/3214745.3214798

Xu Cao, Weimin Wang, Katashi Nagao, and Ryosuke Nakamura. 2020. PSNet: A Style Transfer Network for Point Cloud Stylization on Geometry and Color. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, Snowmass Village, CO, USA, 3326–3334. https://doi.org/10.1109/WACV45572.2020.9093513

Géry Casiez, Nicolas Roussel, and Daniel Vogel. 2012. 1 € filter: a simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. Association for Computing Machinery, New York, NY, USA, 2527–2530. https://doi.org/10.1145/2207676.2208639

Menglei Chai, Changxi Zheng, and Kun Zhou. 2014. A reduced model for interactive hairs. *ACM Transactions on Graphics* 33, 4 (July 2014), 1–11. https://doi.org/10.1145/2601097.2601211

Hong Chen and Song-Chun Zhu. 2006. A generative sketch model for human hair analysis and synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 7 (July 2006), 1025–1040. https://doi.org/10.1109/TPAMI.2006.131

Yingshu Chen, Guocheng Shao, Ka Chun Shum, Binh-Son Hua, and Sai-Kit Yeung. 2025. Advances in 3D Neural Stylization: A Survey. *International Journal of Computer Vision* (March 2025). https://doi.org/10.1007/s11263-025-02403-9

Jiwoo Chung, Sangeek Hyun, and Jae-Pil Heo. 2024a. Style Injection in Diffusion: A Training-Free Approach for Adapting Large-Scale Diffusion Models for Style Transfer. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Seattle, WA, USA, 8795–8805. https://doi.org/10.1109/CVPR52733.2024.00840

SeungJeh Chung, Joohyun Park, and Hyeongyeop Kang. 2024b. 3DStyleGLIP: Part-Tailored Text-Guided 3D Neural Stylization. *Pacific Graphics Conference Papers and Posters* (2024). https://doi.org/10.2312/PG.20241320 Artwork Size: 12 pages Edition: 1320 ISBN: 9783038682509 Publisher: The Eurographics Association.

A. M. Darke, Isaac Olander, and Theodore Kim. 2024. More Than Killmonger Locs: A Style Guide for Black Hair (in Computer Graphics). In *ACM SIGGRAPH 2024 Courses (SIGGRAPH Courses '24)*. Association for Computing Machinery, New York, NY, USA, 1–251. https://doi.org/10.1145/3664475.3664535

Nam Anh Dinh, Itai Lang, Hyunwoo Kim, Oded Stein, and Rana Hanocka. 2025. Geometry in Style: 3D Stylization via Surface Normal Deformation. https://doi.org/10.48550/arXiv.2503.23241 arXiv:2503.23241 [cs].

Hongbo Fu, Yichen Wei, Chiew-Lan Tai, and Long Quan. 2007. Sketching hairstyles. In *Proceedings of the 4th Eurographics workshop on Sketch-based interfaces and modeling (SBIM '07)*. Association for Computing Machinery, New York, NY, USA, 31–36. https://doi.org/10.1145/1384429.1384439

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2016. Image Style Transfer Using Convolutional Neural Networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Las Vegas, NV, USA, 2414–2423. https://doi.org/10.1109/CVPR.2016.265

Ashraf Ghoniem and Ken Museth. 2013. Hair growth by means of sparse volumetric modeling and advection. In *ACM SIGGRAPH 2013 Talks*. ACM, Anaheim California, 1–1. https://doi.org/10.1145/2504459.2504502

Guilherme Gomes Haetinger, Jingwei Tang, Raphael Ortiz, Paul Kanyuk, and Vinicius Azevedo. 2024. Controllable Neural Style Transfer for Dynamic Meshes. In *ACM SIGGRAPH 2024 Conference Papers (SIGGRAPH '24)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3641519.3657474

Jie Guo, Mengtian Li, Zijing Zong, Yuntao Liu, Jingwu He, Yanwen Guo, and Ling-Qi Yan. 2021. Volumetric appearance stylization with stylizing kernel prediction network. *ACM Trans. Graph.* 40, 4 (July 2021), 162:1–162:15. https://doi.org/10.1145/3450626.3459799

Sunil Hadap, Marie-Paule Cani, Ming Lin, Tae-Yong Kim, Florence Bertails, Steve Marschner, Kelly Ward, and Zoran Kačić-Alesić. 2007. Strands and hair: modeling, animation, and rendering. In *ACM SIGGRAPH 2007 courses*. ACM, San Diego California, 1–150. https://doi.org/10.1145/1281500.1281689

Daniela Hasenbring and Henrik Karlsson. 2021. Hair Grooming with Imageworks' Fyber. In *ACM SIGGRAPH 2021 Talks*. ACM, Virtual Event USA, 1–2. https://doi.org/10.1145/3450623.3464668

Chengan He, Xin Sun, Zhixin Shu, Fujun Luan, Soren Pirk, Jorge Alejandro Amador Herrera, Dominik Michels, Tuanfeng Yang Wang, Meng Zhang, Holly Rushmeier, and Yi Zhou. 2024. Perm: A Parametric Representation for Multi-Style 3D Hair Modeling. https://openreview.net/forum?id=WKfb1xGXGx

Amir Hertz, Andrey Voynov, Shlomi Fruchter, and Daniel Cohen-Or. 2024. Style Aligned Image Generation via Shared Attention. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Seattle, WA, USA, 4775–4785. https://doi.org/10.1109/CVPR52733.2024.00457

Jonathan Hoffman, Te Hu, Paul Kanyuk, Stephen Marshall, George Nguyen, Hope Schroers, and Patrick Witting. 2023. Creating Elemental Characters: From Sparks to Fire. In *ACM SIGGRAPH 2023 Talks*. ACM, Los Angeles CA USA, 1–2. https://doi.org/10.1145/3587421.3595467

Jerry Hsu, Tongtong Wang, Zherong Pan, Xifeng Gao, Cem Yuksel, and Kui Wu. 2024. Real-time Physically Guided Hair Interpolation. *ACM Transactions on Graphics* 43, 4 (July 2024), 1–11. https://doi.org/10.1145/3658176

Xun Huang and Serge Belongie. 2017. Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, Venice, 1510–1519. https://doi.org/10.1109/ICCV.2017.167

Hayley Iben, Mark Meyer, Lena Petrovic, Olivier Soares, John Anderson, and Andrew Witkin. 2013. Artistic simulation of curly hair. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, Anaheim California. https://doi.org/10.1145/2485895.2485913

Ondřej Jamriška, Šárka Sochorová, Ondřej Texler, Michal Lukáč, Jakub Fišer, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. 2019. Stylizing video by example. *ACM Trans. Graph.* 38, 4 (July 2019), 107:1–107:11. https://doi.org/10.1145/3306346.3323006

Erik Sven Vasconcelos Jansson, Matthäus G. Chajdas, Jason Lacroix, and Ingemar Ragnemalm. 2019. Real-Time Hybrid Hair Rendering. *Eurographics Symposium on Rendering - DL-only and Industry Track* (2019), 8 pages. https://doi.org/10.2312/SR.20191215 Artwork Size: 8 pages ISBN: 9783038680956 Publisher: The Eurographics Association Version Number: 001-008.

Soorya Narayan Jayaraman Mohan. 2023. Hair Tubes: Stylized Hair from Polygonal Meshes of Arbitrary Topology. In *SIGGRAPH Asia 2023 Technical Communications*. ACM, Sydney NSW Australia, 1–4. https://doi.org/10.1145/3610543.3626157

Hyunyoung Jung, Seonghyeon Nam, Nikolaos Sarafianos, Sungjoo Yoo, Alexander Sorkine-Hornung, and Rakesh Ranjan. 2024. Geometry Transfer for Stylizing Radiance Fields. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Seattle, WA, USA, 8565–8575. https://doi.org/10.1109/CVPR52733.2024.00818

J. T. Kajiya and T. L. Kay. 1989. Rendering fur with three dimensional textures. *SIGGRAPH Comput. Graph.* 23, 3 (July 1989), 271–280. https://doi.org/10.1145/74334.74361

Paul Kanyuk, Vinicius Azevedo, Raphael Ortiz, and Jingwei Tang. 2023. Singed Silhouettes and Feed Forward Flames: Volumetric Neural Style Transfer for Expressive Fire Simulation. In *ACM SIGGRAPH 2023 Talks*. ACM, Los Angeles CA USA, 1–2. https://doi.org/10.1145/3587421.3595435

Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Neural 3D Mesh Renderer. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Salt Lake City, UT, 3907–3916. https://doi.org/10.1109/CVPR.2018.00411

Avneet Kaur, Maryann Simmons, and Brian Whited. 2018. Hierarchical controls for art-directed hair at Disney. In *ACM SIGGRAPH 2018 Talks*. ACM, Vancouver British Columbia Canada, 1–2. https://doi.org/10.1145/3214745.3214774

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.* 42, 4 (July 2023), 139:1–139:14. https://doi.org/10.1145/3592433

Byungsoo Kim, Vinicius C. Azevedo, Markus Gross, and Barbara Solenthaler. 2019. Transport-based neural style transfer for smoke simulations. *ACM Trans. Graph.* 38, 6 (Nov. 2019), 188:1–188:11. https://doi.org/10.1145/3355089.3356560

Byungsoo Kim, Vinicius C. Azevedo, Markus Gross, and Barbara Solenthaler. 2020. Lagrangian neural style transfer for fluids. *ACM Trans. Graph.* 39, 4 (Aug. 2020), 52:52:1–52:52:10. https://doi.org/10.1145/3386569.3392473

Tae-Yong Kim and Ulrich Neumann. 2002. Interactive multiresolution hair modeling and editing. *ACM Trans. Graph.* 21, 3 (July 2002), 620–629. https://doi.org/10.1145/566654.566627

M. Kohlbrenner, U. Finnendahl, T. Djuren, and M. Alexa. 2021. Gauss Stylization: Interactive Artistic Mesh Modeling based on Preferred Surface Normals. *Computer Graphics Forum* 40, 5 (2021), 33–43. https://doi.org/10.1111/cgf.14355 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14355.

Nicholas Kolkin, Michal Kucera, Sylvain Paris, Daniel Sykora, Eli Shechtman, and Greg Shakhnarovich. 2022. Neural Neighbor Style Transfer. https://doi.org/10.48550/arXiv.2203.13215 arXiv:2203.13215 [cs].

Chuan Li and Michael Wand. 2016. Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Las Vegas, NV, USA, 2479–2486. https://doi.org/10.1109/CVPR.2016.272

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. 2022. Grounded Language-Image Pre-training. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, New Orleans, LA, USA, 10955–10965. https://doi.org/10.1109/CVPR52688.2022.01069

Shaohua Li, Xinxing Xu, Liqiang Nie, and Tat-Seng Chua. 2017b. Laplacian-Steered Neural Style Transfer. In *Proceedings of the 25th ACM international conference on Multimedia*. ACM, Mountain View California USA, 1716–1724. https://doi.org/10.1145/3123266.3123425

Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. 2017a. Demystifying Neural Style Transfer. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, Melbourne, Australia, 2230–2236. https://doi.org/10.24963/ijcai.2017/310

Hongyu Liu, Xuan Wang, Ziyu Wan, Yujun Shen, Yibing Song, Jing Liao, and Qifeng Chen. 2024. HeadArtist: Text-conditioned 3D Head Generation with Self Score Distillation. In *ACM SIGGRAPH 2024 Conference Papers (SIGGRAPH '24)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3641519.3657512

Hsueh-Ti Derek liu and Alec Jacobson. 2019. Cubic stylization. *ACM Trans. Graph.* 38, 6 (Nov. 2019), 197:1–197:10. https://doi.org/10.1145/3355089.3356495

Hsueh-Ti Derek Liu, Michael Tao, and Alec Jacobson. 2018. Paparazzi: surface editing by way of multi-view image processing. *ACM Trans. Graph.* 37, 6 (Dec. 2018), 221:1–221:11. https://doi.org/10.1145/3272127.3275047

Haimin Luo, Min Ouyang, Zijun Zhao, Suyi Jiang, Longwen Zhang, Qixuan Zhang, Wei Yang, Lan Xu, and Jingyi Yu. 2024. GaussianHair: Hair Modeling and Rendering with Light-aware Gaussians. https://doi.org/10.48550/arXiv.2402.10483 arXiv:2402.10483 [cs].

Stephen R. Marschner, Henrik Wann Jensen, Mike Cammarano, Steve Worley, and Pat Hanrahan. 2003. Light scattering from human hair fibers. *ACM Trans. Graph.* 22, 3

(July 2003), 780–791. https://doi.org/10.1145/882262.882345

Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. 2022. Text2Mesh: Text-Driven Neural Stylization for Meshes. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, New Orleans, LA, USA, 13482–13492. https://doi.org/10.1109/CVPR52688.2022.01313

Brian Missey, Amaury Aubel, Arunachalam Somasundaram, and Megha Davalath. 2017. Hairy effects in *Trolls*. In *ACM SIGGRAPH 2017 Talks*. ACM, Los Angeles California, 1–2. https://doi.org/10.1145/3084363.3085070

Brandon Montell, Fernando de Goes, and Jacob Brooks. 2022. Hair Emoting with Style Guides in Turning Red. In *ACM SIGGRAPH 2022 Talks (SIGGRAPH '22)*. Association for Computing Machinery, New York, NY, USA, 1–2. https://doi.org/10.1145/3532836.3536253

Jonathan T. Moon and Stephen R. Marschner. 2006. Simulating multiple scattering in hair using a photon mapping approach. *ACM Trans. Graph.* 25, 3 (July 2006), 1067–1074. https://doi.org/10.1145/1141911.1141995

Matthias Müller, David Charypar, and Markus Gross. 2003. Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA '03)*. Eurographics Association, Goslar, DEU, 154–159.

Matthias Müller, Tae-Yong Kim, and Nuttapong Chentanez. 2012. *Fast Simulation of Inextensible Hair and Fur*. The Eurographics Association. https://doi.org/10.2312/PE/vriphys/vriphys12/039-044

Mike Navarro. 2023. Diving Deeper into Volume Style Transfer. In *ACM SIGGRAPH 2023 Talks*. ACM, Los Angeles CA USA, 1–2. https://doi.org/10.1145/3587421.3595448

Mike Navarro and Jacob Rice. 2021. Stylizing Volumes with Neural Networks. In *ACM SIGGRAPH 2021 Talks*. ACM, Virtual Event USA, 1–2. https://doi.org/10.1145/3450623.3464652

Thu Nguyen-Phuoc, Feng Liu, and Lei Xiao. 2022. SNeRF: stylized neural implicit representations for 3D scenes. *ACM Trans. Graph.* 41, 4 (July 2022), 142:1–142:11. https://doi.org/10.1145/3528223.3530107

Baptiste Nicolet, Alec Jacobson, and Wenzel Jakob. 2021. Large steps in inverse rendering of geometry. *ACM Trans. Graph.* 40, 6 (Dec. 2021), 248:1–248:13. https://doi.org/10.1145/3478513.3480501

P. Noble and W. Tang. 2004. Modelling and animating cartoon hair with NURBS surfaces. In *Proceedings Computer Graphics International, 2004.* 60–67. https://doi.org/10.1109/CGI.2004.1309193 ISSN: 1530-1052.

Sofya Ogunseitan. 2022. Space Rangers with Cornrows: Methods for Modeling Braids and Curls in Pixar's Groom Pipeline. In *ACM SIGGRAPH 2022 Talks (SIGGRAPH '22)*. Association for Computing Machinery, New York, NY, USA, 1–2. https://doi.org/10.1145/3532836.3536277

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: an imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Number 721. Curran Associates Inc., Red Hook, NY, USA, 8026–8037.

Eric Risser, Pierre Wilmot, and Connelly Barnes. 2017. Stable and Controllable Neural Texture Synthesis and Style Transfer Using Histogram Losses. http://arxiv.org/abs/1701.08893 arXiv:1701.08893 [cs].

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, New Orleans, LA, USA, 10674–10685. https://doi.org/10.1109/cvpr52688.2022.01042

Radu Alexandru Rosu, Shunsuke Saito, Ziyan Wang, Chenglei Wu, Sven Behnke, and Giljoo Nam. 2022. Neural Strands: Learning Hair Geometry and Appearance from Multi-view Images. In *Computer Vision – ECCV 2022*. Vol. 13693. Springer Nature Switzerland, Cham, 73–89. https://doi.org/10.1007/978-3-031-19827-4_5 Series Title: Lecture Notes in Computer Science.

Litu Rout, Yujia Chen, Nataniel Ruiz, Abhishek Kumar, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. 2024. RB-Modulation: Training-Free Stylization using Reference-Based Modulation. https://openreview.net/forum?id=bnINPG5A32&noteId=obWvN2p4mW

Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. 2018. Artistic Style Transfer for Videos and Spherical Images. *International Journal of Computer Vision* 126, 11 (Nov. 2018), 1199–1219. https://doi.org/10.1007/s11263-018-1089-z

Shunsuke Saito, Liwen Hu, Chongyang Ma, Hikaru Ibayashi, Linjie Luo, and Hao Li. 2018. 3D hair synthesis using volumetric variational autoencoders. *ACM Trans. Graph.* 37, 6 (Dec. 2018), 208:1–208:12. https://doi.org/10.1145/3272127.3275019

Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. https://arxiv.org/abs/1409.1556v6

Vanessa Sklyarova, Jenya Chelishev, Andreea Dogaru, Igor Medvedev, Victor Lempitsky, and Egor Zakharov. 2023. Neural Haircut: Prior-Guided Strand-Based Hair Reconstruction. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Paris, France, 19705–19716. https://doi.org/10.1109/ICCV51070.2023.01810

Vanessa Sklyarova, Egor Zakharov, Otmar Hilliges, Michael J. Black, and Justus Thies. 2024. Text-Conditioned Generative Model of 3D Strand-Based Human Hairstyles. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Seattle, WA, USA, 4703–4712. https://doi.org/10.1109/CVPR52733.2024.00450

Arunachalam Somasundaram. 2015. Dynamically controlling hair interpolation. In *ACM SIGGRAPH 2015 Talks*. ACM, Los Angeles California, 1–1. https://doi.org/10.1145/2775280.2792541

Olga Sorkine and Marc Alexa. 2007. As-rigid-as-possible surface modeling. In *Proceedings of the fifth Eurographics symposium on Geometry processing (SGP '07)*. Eurographics Association, Goslar, DEU, 109–116.

Xiaokun Sun, Zeyu Cai, Ying Tai, Jian Yang, and Zhenyu Zhang. 2024. StrandHead: Text to Strand-Disentangled 3D Head Avatars Using Hair Geometric Priors. https://doi.org/10.48550/arXiv.2412.11586 arXiv:2412.11586 [cs].

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2014. Going Deeper with Convolutions. https://doi.org/10.48550/arXiv.1409.4842 arXiv:1409.4842 [cs].

Yusuke Takimoto, Hikari Takehara, Hiroyuki Sato, Zihao Zhu, and Bo Zheng. 2024. Dr.Hair: Reconstructing Scalp-Connected Hair Strands without Pre-Training via Differentiable Rendering of Line Segments. 20601–20611. https://openaccess.thecvf.com/content/CVPR2024/html/Takimoto_Dr.Hair_Reconstructing_Scalp-Connected_Hair_Strands_without_Pre-Training_via_Differentiable_Rendering_CVPR_2024_paper.html

Marie Tollec and Mike Navarro. 2024. Making Magic with 3D Volume Style Transfer. In *ACM SIGGRAPH 2024 Talks*. ACM, Denver CO USA, 1–2. https://doi.org/10.1145/3641233.3664340

Nhat Phuong Anh Vu, Abhishek Saroha, Or Litany, and Daniel Cremers. 2025. GAS-NeRF: Geometry-Aware Stylization of Dynamic Radiance Fields. https://doi.org/10.48550/arXiv.2503.08483 arXiv:2503.08483 [cs].

Dongqing Wang, Ehsan Pajouheshgar, Yitao Xu, Tong Zhang, and Sabine Süsstrunk. 2025. Volumetric Temporal Texture Synthesis for Smoke Stylization using Neural Cellular Automata. https://doi.org/10.48550/arXiv.2502.09631 arXiv:2502.09631 [eess].

Lvdi Wang, Yizhou Yu, Kun Zhou, and Baining Guo. 2009. Example-based hair geometry synthesis. *ACM Transactions on Graphics* 28, 3 (July 2009), 1–9. https://doi.org/10.1145/1531326.1531362

Jamie Wither, Florence Bertails, and Marie-Paule Cani. 2007. Realistic Hair from a Sketch. In *IEEE International Conference on Shape Modeling and Applications 2007 (SMI '07)*. 33–42. https://doi.org/10.1109/SMI.2007.31

Haomiao Wu, Alvin Shi, A.M. Darke, and Theodore Kim. 2024. Curly-Cue: Geometric Methods for Highly Coiled Hair. In *SIGGRAPH Asia 2024 Conference Papers*. ACM, Tokyo Japan, 1–11. https://doi.org/10.1145/3680528.3687641

Keyu Wu, Yifan Ye, Lingchen Yang, Hongbo Fu, Kun Zhou, and Youyi Zhengl. 2022. NeuralHDHair: Automatic High-fidelity Hair Modeling from a Single Image Using Implicit Neural Representations. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, New Orleans, LA, USA, 1516–1525. https://doi.org/10.1109/CVPR52688.2022.00158

Kui Wu and Cem Yuksel. 2017. Real-time fiber-level cloth rendering. In *Proceedings of the 21st ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D '17)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3023368.3023372

Jun Xing, Koki Nagano, Weikai Chen, Haotian Xu, Li-yi Wei, Yajie Zhao, Jingwan Lu, Byungmoon Kim, and Hao Li. 2019. HairBrush for Immersive Data-Driven Hair Modeling. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. ACM, New Orleans LA USA, 263–279. https://doi.org/10.1145/3332165.3347876

Haibo Yang, Yang Chen, Yingwei Pan, Ting Yao, Zhineng Chen, and Tao Mei. 2023. 3DStyle-Diffusion: Pursuing Fine-grained Text-driven 3D Stylization with 2D Diffusion Models. In *Proceedings of the 31st ACM International Conference on Multimedia*. ACM, Ottawa ON Canada, 6860–6868. https://doi.org/10.1145/3581783.3612363

Zixuan Ye, Huijuan Huang, Xintao Wang, Pengfei Wan, Di Zhang, and Wenhan Luo. 2024. StyleMaster: Stylize Your Video with Artistic Generation and Translation. https://doi.org/10.48550/arXiv.2412.07744 arXiv:2412.07744 [cs].

Cem Yuksel, Scott Schaefer, and John Keyser. 2009. Hair meshes. *ACM Transactions on Graphics* 28, 5 (Dec. 2009), 1–7. https://doi.org/10.1145/1618452.1618512

Egor Zakharov, Vanessa Sklyarova, Michael Black, Giljoo Nam, Justus Thies, and Otmar Hilliges. 2025. Human Hair Reconstruction with Strand-Aligned 3D Gaussians. In *Computer Vision – ECCV 2024*. Vol. 15074. Springer Nature Switzerland, Cham, 409–425. https://doi.org/10.1007/978-3-031-72640-8_23 Series Title: Lecture Notes in Computer Science.

Dingxi Zhang, Yu-Jie Yuan, Zhuoxun Chen, Fang-Lue Zhang, Zhenliang He, Shiguang Shan, and Lin Gao. 2024b. StylizedGS: Controllable Stylization for 3D Gaussian Splatting. https://doi.org/10.48550/arXiv.2404.05220 arXiv:2404.05220 [cs].

Hao Zhang, Yao Feng, Peter Kulits, Yandong Wen, Justus Thies, and Michael J. Black. 2024a. TECA: Text-Guided Generation and Editing of Compositional 3D Avatars. In *2024 International Conference on 3D Vision (3DV)*. 1520–1530. https://doi.org/10.

1109/3DV62453.2024.00151 ISSN: 2475-7888.

Yuxiao Zhou, Menglei Chai, Alessandro Pepe, Markus Gross, and Thabo Beeler. 2023. GroomGen: A High-Quality Generative Hair Model Using Hierarchical Latent Representations. *ACM Trans. Graph.* 42, 6 (Dec. 2023), 270:1–270:16. https://doi.org/10.1145/3618309

Yi Zhou, Liwen Hu, Jun Xing, Weikai Chen, Han-Wei Kung, Xin Tong, and Hao Li. 2018. HairNet: Single-View Hair Reconstruction Using Convolutional Neural Networks. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XI.* Springer-Verlag, Berlin, Heidelberg, 249–265. https://doi.org/10.1007/978-3-030-01252-6_15

Zhenglin Zhou, Fan Ma, Hehe Fan, Zongxin Yang, and Yi Yang. 2025. HeadStudio: Text to Animatable Head Avatars with 3D Gaussian Splatting. In *Computer Vision – ECCV 2024.* Vol. 15090. Springer Nature Switzerland, Cham, 145–163. https://doi.org/10.1007/978-3-031-73411-3_9 Series Title: Lecture Notes in Computer Science.