Curvature Space Editing of Highly-Coiled Hair

ALVIN SHI, Yale University, USA
FLORENCE BERTAILS-DESCOUBES, Université Grenoble-Alpes, Inria, CNRS, Grenoble INP, LJK, France
A.M. DARKE, University of California, Santa Cruz, USA
THEODORE KIM, Yale University, USA

Fig. 1. *Code My Crown*'s [Dove 2023] *Headwrap Curls* model (top left) with our ruffling (bottom left) and Blender's scraggle modifiers (top right). Our ruffling method ($\rho = 0.05$, $N_m = 30$) operates in curvature space, while Blender's noise-based scraggle operates in position space. The resultant texture from ruffling better resembles the right-hand side of the reference image from Curlsmith [2021] (bottom right).

Due to its highly curved geometry, tightly coiled hair is challenging to model and edit using standard position-based tools. In this work we propose using material curvatures and twists to analyze and edit tightly coiled hair styles. Our method relies on the geometry of super-helices, primitives parametrized by piecewise constant curvatures and twists, whose helical geometry naturally resembles a coiled hair strand. Using this curvature/twist space, we introduce new editing tools that allow us to expand, shrink, "ruffle", interpolate or guide the position of coiled hair in a natural way. We present analytical expressions for geometry and gradients that allow our method to run efficiently and without the need for any training data. We successfully apply our tools to highly coiled simulated hairs, as well as those generated procedurally.

CCS Concepts: • **Computing methodologies → Parametric curve and surface models**.

Additional Key Words and Phrases: Geometry, Strands, Mesh Processing, super-helices

ACM Reference Format:
Alvin Shi, Florence Bertails-Descoubes, A.M. Darke, and Theodore Kim. 2026. Curvature Space Editing of Highly-Coiled Hair. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '26), July 19–23, 2026, Los Angeles, CA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3799902.3811115

Authors' Contact Information: Alvin Shi, Yale University, New Haven, USA, alvin.shi@yale.edu; Florence Bertails-Descoubes, Université Grenoble-Alpes, Inria, CNRS, Grenoble INP, LJK, France, florence.descoubes@inria.fr; A.M. Darke, University of California, Santa Cruz, USA, darke@ucsc.edu; Theodore Kim, Yale University, New Haven, USA, theodore.kim@yale.edu.

SIGGRAPH Conference Papers '26, Los Angeles, CA, USA

ACM ISBN 979-8-4007-2554-8/2026/07
https://doi.org/10.1145/3799902.3811115

1 Introduction

Hair is one of the primary components of human appearance, and convincingly portraying all of its aspects is an ongoing research challenge. Part of this is due to hair's geometric complexity, which can vary widely based on the hair's inherent curvature (curly, coily, straight), as well as how the hair is styled.

While a variety of geometric editing [Kim and Neumann 2002; Yuksel et al. 2009], parameterization [Chang et al. 2025], and generation [Stuyck et al. 2025; Zhou et al. 2023] tools have been developed for hair, there is still a relative dearth of techniques aimed at highly coiled hair [Wu et al. 2024]. In this paper, we aim to fill the gap of *editing* this type of hair. For hair composed of tight helices with small radii, the frequency characteristics are sufficiently distinct from smooth hair that different representations and operations become more effective.

In particular, we propose the *super-helix* as an effective modeling primitive, due to its first-class treatment of curvature and twist. The super-helix has been extensively investigated as a simulation primitive [Bertails 2009; Bertails et al. 2006], especially in cases of physically-based hairstyling [Bertails et al. 2005], hair inverse design in static [Derouet-Jourdan et al. 2013b] and dynamic [Hu et al. 2017a] settings, and dynamic simulation with nonsmooth frictional contact [Bertails-Descoubes et al. 2011; Crespel et al. 2024; Daviet et al. 2011].

However, it has heretofore not been treated as an editing primitive that a user would directly manipulate, e.g. for post-simulation sculpting [Wong et al. 2018] or in response to director draw-overs [Fleischer et al. 2015]. We show that leveraging the geometric properties of super-helices allows one to achieve a variety of analysis and editing operations on highly coiled hair that are not possible using established tools like cage-based deformations [Jacobson et al. 2011; Joshi et al. 2007] or noise-based "scraggle" [Butts et al. 2018; Narayan 2023]. The editing tools we present are input-agnostic, and can be applied to existing hair grooms that were generated either procedurally or through simulation, and also to data that originated as poly-lines using existing conversion tools [Derouet-Jourdan et al. 2013a]. Our contributions are:

- A method of analyzing and identifying features on curves and strands based on the material curvatures and twists of fitted super-helices.
- A new "ruffle" operator that enables a variety of new looks in existing grooms, while still respecting the underlying characteristics of the original.
- A shape optimization method that allows for the direct manipulation of super-helices while preserving length and respecting curvature.
- An efficient analytic gradient strategy that is over an order of magnitude faster than automatic differentiation.

2 Related Works

Computer graphics research on the modeling and simulation of hair spans over 30 years [Anjyo et al. 1992]. Various modeling operations have been proposed, including fluid-based control [Hadap and Magnenat-Thalmann 2000], multi-resolution [Kim and Neumann 2002] and cage-based methods [Yuksel et al. 2009]. However, these methods were predominantly designed for straight or wavy hair, and are less effective on highly coiled hair.

Different methods are needed for different frequency regimes. This is apparent in card-based hair rendering [Tojo et al. 2025; Zheng et al. 2025], where performance degrades on highly coiled examples, as well as some hair capture algorithms [Chai et al. 2015; Saito et al. 2018] which list highly coiled hair as failure cases.

Many production systems have also been built for the modeling and editing of hair, such as Pixar's [Butts et al. 2018] real-time interpolation scheme, Sony's Fyber [Hasenbring and Karlsson 2021], or DreamWorks' Wig [Davalath et al. 2021; Somasundaram 2015]. These are again predominantly designed for, and demonstrated on, straight or wavy hair. Systems from Walt Disney Animation [Kaur et al. 2018], Animal Logic (now Netflix Animation) [Narayan 2023] and ILM's HairCraft [Bowline et al. 2016] all use volumes as an embedding primitive. We show here that background embeddings, such as free form deformation [Sederberg and Parry 1986], Kelvinlets [De Goes and James 2017; de Goes and James 2019], or cage based deformations [Jacobson et al. 2011; Joshi et al. 2007; Lipman et al. 2008] are unsuited for highly coiled hair operations (Fig. 14).

While some simulation methods are geared specifically towards curly hair [Iben et al. 2013; Shi et al. 2023], the number of techniques for modeling highly coiled hair is relatively small [Ogunseitan 2022; Patrick et al. 2004]. Using the statics of super-helices, Bertails et al. [2005] introduced a physically-based method to generate curly and "fuzzy" hairstyles along with cutting, wetting and drying tools, but no means to directly manipulate the hair geometry. Wu et al. [2024] proposed a procedural method specifically for creating highly curly hair with controlled geometric properties, but also does not provide intuitive tools for editing the results. Other techniques for removing "sag" [Hsu et al. 2023; Takahashi and Batty 2025] provide methods for initializing hair simulation rather than editing.

Methods for sculpting [Montell et al. 2022], stylizing [Coban et al. 2025], interpolating [Hsu et al. 2024], or adding noise ("scraggle") [Butts et al. 2018; Haapaoja and Genzwürker 2019; Narayan 2023] to straight or wavy hair have been explored extensively. We propose an alternative "ruffle" operator (Figure 1) that is more appropriate for coiled hair and avoids the blurring effects generated by existing frizzing operators.

Overall, working with super-helices allows us to build geometric tools that operate in *curvature* space rather than in the *position* space. In the case of tightly coiled hair, this offers more natural results.

Our proposed grooming tool performs an optimization over the material curvatures and twists of super-helices. Hence, like other methods for differentiable simulation [Du et al. 2021; Stuyck and Chen 2023], it requires the efficient computation of gradients. We propose an exact, analytical computation of gradients based on a recursive algorithm.

3 Geometry of a super-helix

Previously, helical elements have effectively been utilized as nonlinear constitutive elements in thin elastic rod simulation due to their natural incorporation of curvature and twist, leading to analytical kinematic terms, a linear expression of bending and twisting forces, and intrinsic inextensibility of the rod [Bertails et al. 2006]. For the

modeling and manipulation of highly-coiled strands, super-helix geometric properties also give rise to several novel operations and avenues for analysis.

3.1 Material rods and super-helices

An unshearable material rod is kinematically described by an arclength parameterized curve $\mathbf{r}(s)$, along with an orthonormal material frame $\mathbf{F}(s) = [\mathbf{t}(s)\ \mathbf{u}(s)\ \mathbf{v}(s)]$. These describe how the material composing the rod evolves around the centerline, through rotation around the tangent $\mathbf{t}$ – with rotation rate $\kappa_0(s)$ – and around each of the two axes $\mathbf{u}$ and $\mathbf{v}$ of the rod cross-section – with rotation rates $\kappa_1(s)$ and $\kappa_2(s)$. The rotation rates (or so-called *material strains*) κ_0, κ_1, and κ_2 are respectively called the material *twist* and the two material *curvatures* of the rod. In the following, we concatenate them in the vector $\mathbf{q}(s)$, called the *local curvature vector*. The evolution of the material frame $\mathbf{F}$ as a function of arclength s can be written according to the so-called *Darboux equations*,

$$\mathbf{t}'(s) = \Omega \times \mathbf{t}(s) \qquad \mathbf{u}'(s) = \Omega \times \mathbf{u}(s) \qquad \mathbf{v}'(s) = \Omega \times \mathbf{v}(s), \tag{1}$$

where $\Omega(s) = \kappa_0(s)\mathbf{t}(s) + \kappa_1(s)\mathbf{u}(s) + \kappa_2(s)\mathbf{v}(s)$ is known as the *Darboux vector*, and represents the instantaneous rotation vector of the material frame. Note that $\Omega(s) = \mathbf{F}(s)\mathbf{q}(s)$, meaning that Ω stands for the *global* curvature vector, i.e. the local curvature vector $\mathbf{q}(s)$ expressed in the world frame. For more detailed readings on principal curvatures and the Darboux equation, we refer the reader to Audoly and Pomeau [2010] and Banchoff and Lovett [2022].

In addition to the Darboux equation, if the rod is considered inextensible, its centerline is coupled to the material frame $\mathbf{F}$ through $\mathbf{r}'(s) = \mathbf{t}(s)$, meaning that the tangent vector $\mathbf{r}'(s)$ should be unitary.

A *super-helix* [Bertails et al. 2006] is a special kind of material rod, defined by a local curvature vector $\mathbf{q}(s)$ which is *piecewise constant*. It can be shown that each piece, or *element*, with a constant local curvature vector $\mathbf{q} = (\kappa_0, \kappa_1, \kappa_2)$, has also a constant Darboux vector Ω, and that the shape $\mathbf{r}(s)$ of the element exactly takes that of a circular helix with main axis Ω. In the remainder of this article we assume that $\kappa = \sqrt{\kappa_1^2 + \kappa_2^2} \neq 0$, i.e. that the helix is not degenerate. In such a case, the material twist κ_0 corresponds to the (constant) Frenet torsion τ of a circular helix [Bertails-Descoubes et al. 2018], and κ to the geometric (or Frenet) curvature of the helix. As a result, the helix shape is characterized by a radius $R = \frac{\kappa}{\|\Omega\|^2}$ and by a step $\Delta = 2\pi\frac{\tau}{\|\Omega\|^2}$ (see Sec. 4.1 or [Bertails 2006, Appendix D]).

3.2 Geometry of a single element

Starting from an initial material frame $\mathbf{F}_b = [\mathbf{t}_b\ \mathbf{u}_b\ \mathbf{v}_b]$ and a given (constant) Darboux vector Ω, the Darboux equations (1) can be integrated analytically [Derouet-Jourdan et al. 2013a, Appendix A],

$$\begin{aligned}\mathbf{t}(\Omega, \mathbf{t}_b, s) = &\left(\mathbf{t}_b \cdot \frac{\Omega}{\|\Omega\|}\right)\frac{\Omega}{\|\Omega\|} \\ &+ \cos(\|\Omega\|\, s)\left(\mathbf{t}_b - \left(\mathbf{t}_b \cdot \frac{\Omega}{\|\Omega\|}\right)\frac{\Omega}{\|\Omega\|}\right) \\ &+ \sin(\|\Omega\|\, s)\left(\frac{\Omega}{\|\Omega\|} \times \mathbf{t}_b\right).\end{aligned} \tag{2}$$

Similar expressions follow for $\mathbf{u}$ and $\mathbf{v}$ by replacing $\mathbf{t}$ and $\mathbf{t}_b$ with their corresponding vectors.

The geometry of a single element $\mathbf{r}(\Omega, \mathbf{p}_b, \mathbf{t}_b, s)$ can be thought of as the trajectory traced out by a particle starting at $\mathbf{p}_b$ following $\mathbf{t}(\Omega, \mathbf{t}_b, s)$ as a tangent vector. Integrating $\mathbf{t}$ along s from 0 to l, we obtain the general equation for a single helical element of length l [Derouet-Jourdan et al. 2013a, Appendix A],

$$\begin{aligned}\mathbf{r}(\Omega, \mathbf{p}_b, \mathbf{t}_b, s) = \mathbf{p}_b &+ \tau\frac{\Omega}{\|\Omega\|^2}s \\ &+ \frac{\sin(\|\Omega\|\, s)}{\|\Omega\|}\left(\mathbf{t}_b - \tau\frac{\Omega}{\|\Omega\|^2}\right) \\ &+ \frac{1 - \cos(\|\Omega\|\, s)}{\|\Omega\|}\left(\frac{\Omega}{\|\Omega\|} \times \mathbf{t}_b\right),\end{aligned} \tag{3}$$

where the constant Frenet torsion $\tau = \Omega \cdot \mathbf{t}_b$ (Fig. 2).

3.3 Full chains of elements

Multiple elements with different Darboux vectors can smoothly connect at their endpoints to form a C^1 piecewise helical curve [1], which we will refer to as a super-helix $\mathcal{S}$. The geometry of a super-helix is thus fully determined by fixing an initial point $\mathbf{p}_0$, an initial tangent $\mathbf{t}_0$, and a sequence of Darboux vectors Ω_i and lengths l_i,

$$\mathcal{S}(s) = \mathbf{r}\left(\Omega_i, \mathbf{p}_i, \mathbf{t}_i, s - \sum_{n=0}^{i-1} l_n\right),\ 0 \leq s - \sum_{n=0}^{i-1} l_n \leq l_i, \tag{4}$$

where $\mathbf{p}_i \equiv \mathbf{r}_i(0)$ and $\mathbf{t}_i \equiv \mathbf{t}_i(0)$, the initial points and tangents of each element i, are defined recursively using continuity conditions at the junction between two elements,

$$\mathbf{p}_i = \mathbf{r}\left(\Omega_{i-1}, \mathbf{p}_{i-1}, \mathbf{t}_{i-1}, l_{i-1}\right) \tag{5}$$

$$\mathbf{t}_i = \mathbf{t}_{i-1}(l_{i-1}). \tag{6}$$

Additionally, having a full initial frame $\mathbf{F}_0 = [\mathbf{t}_0\ \mathbf{u}_0\ \mathbf{v}_0]$ and ensuring the continuity[2] of the two cross-section vectors,

$$\mathbf{u}_i = \mathbf{u}_{i-1}(l_{i-1}) \tag{7}$$

$$\mathbf{v}_i = \mathbf{v}_{i-1}(l_{i-1}), \tag{8}$$

finally defines the entire material frame $\mathbf{F}(s)$ recursively over the full super-helix chain.

3.4 Super-helix state

Each helical element has a total of four degrees of freedom: three from its Darboux vector and one from its prescribed length. With the addition of a root point and frame, a framed super-helix consisting of N elements therefore has $4N + 6$ degrees of freedom. Any modifications of a super-helix's initial frame or position respectively results in a global rotation/reflection or translation, while modifying element lengths can result in highly nonlinear deflection.

We choose to consider as our main state variable the full local curvature vector $\mathbf{q} \in \mathbb{R}^{3N}$, stacking all the local curvatures $\mathbf{q}_i = \mathbf{F}_i^\top \Omega_i$ for the entire super-helix. Indeed, these local curvature coordinates are invariant under rotations applied to the strand, making them

[1] The continuity of the tangent vector $\mathbf{t}(s)$ guarantees C^1 continuity of $\mathbf{r}(s)$, however note that there is possibly a curvature jump at each junction between two elements.

[2] This is the main difference between the material frame $\mathbf{F}$ of a super-helix and the Frenet frame $\mathbf{F}_f$ of a piecewise helical curve. While the former is continuous, the latter possibly has jumps of its normal and binormal at each junction. Actually, $\mathbf{F}$ evolves similarly to $\mathbf{F}_f$ (same Ω), albeit up to a constant rotation about the tangent.

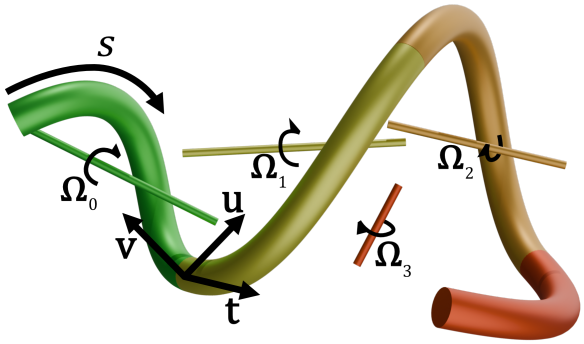


Fig. 2. A set of four helical elements with Darboux vectors Ω_i visualized as axes of rotation. $\mathbf{t}$, $\mathbf{u}$, and $\mathbf{v}$ are frame directions for $\mathbf{F}$. The entire strand is parameterized by arclength s.

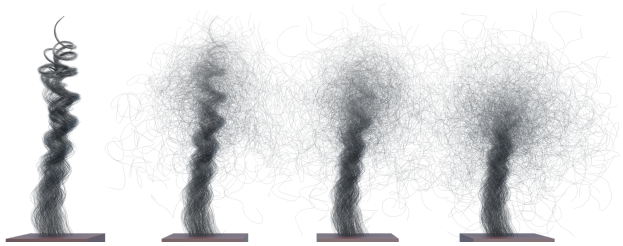


Fig. 3. Ruffling a single wisp. From left to right: $\rho = 0,\ 0.05,\ 0.1,\ 0.25$. Each strand has 90 elements, and the matching cutoff is $N_m = 30$.

attractive as an underlying set of parameters for interpolation and analysis (Fig. 9).

3.5 Polylines to super-helices

Derouet-Jourdan et al. [2013a]'s work allows for the efficient conversion of a polyline to a piecewise helical curve (transitioning through a spline representation $\boldsymbol{\alpha}(s)$) with guarantees for accuracy as the number of elements N increases. Matching terms from their formulation allows us to extract Darboux vectors Ω_i and lengths l_i. To build an initial frame, we start from the C^1 parameterized curve $\boldsymbol{\alpha}(s)$ and use the tangent $\boldsymbol{\alpha}'(0)$ and an arbitrary vector $\mathbf{k}$ non-parallel to the tangent,

$$\left[\ \boldsymbol{\alpha}'(0)\ \middle|\ \mathbf{k}\times\boldsymbol{\alpha}(0)\ \middle|\ \boldsymbol{\alpha}'(0)\times(\mathbf{k}\times\boldsymbol{\alpha}'(0))\ \right]. \tag{9}$$

Normalizing the columns then yields our initial frame $\mathbf{F}_0$. Transporting that frame according to the Darboux equations (1) then allows for the construction of the full frame $\mathbf{F}(s)$ and $\mathbf{q}$.

4 Strand Feature Analysis

Given curves parameterized as super-helices, analysis of their state vectors gives useful information that corresponds to qualitative features in the strand itself. Furthermore, since each element in the chain fits to a specific portion of the entire strand, this allows for a clean mapping between frequency-based features and specific locations along the strand while also allowing for quick shape exploration.

4.1 Helical offsets

Up to a rotation and translation, a circular helix can be expressed parametrically as

$$\mathbf{h}(t) = (t, R\sin(\omega t), R\cos(\omega t))\,. \tag{10}$$

Applying equation (10) as an offset to some central curve has been common practice for the generation of curly hair in procedural systems [Chang et al. 2025; Ogunseitan 2022], with controls over R and ω for curl radius and spatial frequency, respectively.

With arclength reparameterization, we obtain $\mathbf{h}(s)$ as

$$\left(\frac{s}{\sqrt{1+\omega^2R^2}}, R\sin\left(\frac{\omega s}{\sqrt{1+\omega^2R^2}}\right), R\cos\left(\frac{\omega s}{\sqrt{1+\omega^2R^2}}\right)\right). \tag{11}$$

Matching terms between $\mathbf{h}(s)$ and equation (3), we derive effective radii and spatial frequencies for helical elements based on their Darboux vectors,

$$R = \frac{\sqrt{\|\Omega\|^2-\tau^2}}{\|\Omega\|^2} = \frac{\kappa}{\|\Omega\|^2} \qquad \omega = \frac{\|\Omega\|^2}{\tau}. \tag{12}$$

Taking these values for each element in a super-helix yields a sequence of local radii and spatial frequencies. These are useful for the identification of regions of particular frequency, radii, or events where helical handedness flips, though sampling rates can introduce secondary beat effects (Fig. 4a).

4.2 Centerline alignment

Omitting the oscillatory parts of equation 3, the central axis of a single super-helix is as follows,

$$\mathbf{r}_C(\Omega, \mathbf{p}_b, \mathbf{t}_b, s) = \mathbf{p}_b + \tau\frac{\Omega}{\|\Omega\|^2}s + \frac{\Omega\times\mathbf{t}_b}{\|\Omega\|^2}. \tag{13}$$

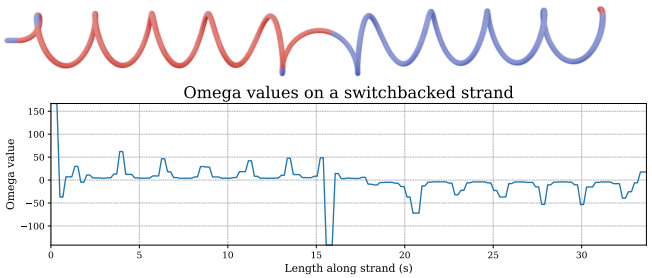
For super-helices, the sequence of axes is not guaranteed to be continuous, as each axis is parallel to the Darboux vector of its corresponding element.

However, in the case of a sequence of elements fitted to a perfect helix, the sequence of line segments equation (13) produces matches an intuitive notion of the helix's centerline. Towards generalizing this notion, we utilize the method of Wu et al. [2024] for obtaining a centercurve $\mathbf{c}(s)$ of $\mathcal{S}(s)$ using Fourier analysis and consider the quantity,

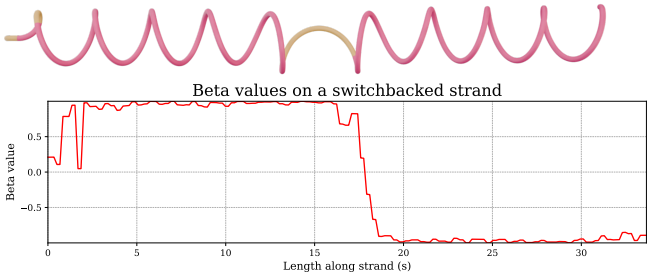
$$\beta(s) = \frac{\mathbf{c}'(s)\cdot\Omega(s)}{\|\Omega(s)\|}, \tag{14}$$

where $\Omega(s)$ is the Darboux vector of the super-helix $\mathcal{S}$ at s.

For regions of $\mathcal{S}$ with larger ω values, the sign of β coincides with the strand winding clockwise/counterclockwise. Furthermore, $|\beta|$ is an indicator for whether the strand is winding around the centercurve in a helical or planar wavy manner, in which case $|\beta|$ would be closer to one or zero, respectively. Since β is also bounded in $[-1, 1]$, it can serve as a smoother probe into helical handedness without the confounding factor of magnitude noisiness that tracking ω can introduce (Fig. 4b).



(a) Plotting ω. Note that handedness flips at the switchback. elements with positive ω are colored red, while elements with negative ω are colored blue.



(b) Plotting β. Elements with $|\beta| > 0.85$ are colored pink, while elements that aren't are colored yellow, isolating the switchback.

Fig. 4. Plotting ω and β on a switchback strand. ω tracks absolute angular velocity, while β more cleanly illustrates the handedness change.

4.3 Constructing novel q vectors

From these analyses, we propose that these state vectors $\mathbf{q}$ can serve as a stylistic signature for super-helices, providing a blueprint for how they should grow from root to tip. Careful authoring and modifications of $\mathbf{q}$ then lead to a variety of new strand shapes.

4.3.1 Ruffling modifier. Existing methods for hair randomization use noise textures to procedurally displace individual strands. These displacements act uniformly on the underlying spatial frequencies, skewing the resulting distribution towards uniformity. However, highly-coiled hair has more structured spatial frequencies in higher bands than straight hair, requiring more careful treatment.

Given a collection of super-helices, permutations of paired local curvatures/lengths between strands give rise to novel grooms from existing data. We call this procedure *ruffling*.

Since modifying earlier elements in a super-helix contributes more towards changing the final shape than later elements, it is useful to let some number of elements N_m match a single baseline super-helix before allowing any operations to occur. Additionally, we fix a single replacement probability ρ for any ith element past N_m, selecting a random element uniformly from the set of all ith elements in the super-helix group (Fig 3). Additionally, smooth local curvature interpolation between the two sets of strands serves as another degree of control after ruffling has been performed.

5 Groom Sculpting

Manipulating large volumes of hair to fit a certain silhouette is a common goal in groom modeling. Similar to other strain-based approaches [Fröhlich and Botsch 2011; Schumacher et al. 2012], we formulate the task as an optimization problem, allowing the super-helix to smoothly morph points towards user-defined anchors.

5.1 Shape optimization

Given a super-helix $\mathcal{S}$ with starting state $\mathbf{q}_0$, starting point $\mathbf{p}_0$, and starting frame $\mathbf{F}_0$, we specify M control points along the strand $\mathcal{S}(\lambda_k)$, each of which is assigned a corresponding anchor point $\mathbf{a}_k$.

The task is then to minimally modify $\mathbf{q}$ so that each control point reaches its corresponding anchor. Formulated as an optimization problem, we apply a quadratic penalty for each point/anchor pair, along with a quadratic penalty for the state vector's displacement,

$$\mathbf{q}^* = \underset{\mathbf{q}}{\operatorname{argmin}} \sum_{k=0}^{M-1} \|\mathcal{S}(\lambda_k) - \mathbf{a}_k\|^2 + c\,\|\mathbf{q} - \mathbf{q}_0\|^2 . \tag{15}$$

Here, c is a tunable parameter, allowing the user to specify how stiffly the strand should resist changes in local curvature relative to reaching the anchor points (Fig. 10). Note that individual element lengths are completely untouched in this process, meaning that all deformed strands are guaranteed to preserve their lengths.

5.2 Analytic gradients

To perform optimization (15) on a super-helix of N elements, differentiation becomes necessary. Using the chain rule,

$$\frac{\partial}{\partial \mathbf{q}} \|\mathcal{S}(\lambda_k) - \mathbf{a}_k\|^2 = 2\left(\frac{\partial \mathcal{S}}{\partial \mathbf{q}}(\lambda_k)\right)^{\top} \big(\mathcal{S}(\lambda_k) - \mathbf{a}_k\big). \tag{16}$$

The most involved term in this calculation is $\frac{\partial \mathcal{S}}{\partial \mathbf{q}} \in \mathbb{R}^{3\times 3N}$, describing super-helix shape variation with respect to its curvature parameters. Recalling that $\mathbf{q}$ is a stack of individual local curvatures $\mathbf{q}_j$ for $j \in [0, N-1]$, we narrow our focus specifically to blocks $\frac{\partial \mathcal{S}}{\partial \mathbf{q}_j} \in \mathbb{R}^{3\times 3}$.

As seen in equation (4), sampling a super-helix farther down its length depends on the endpoints and frames sampled from earlier elements in the chain. Let $\mathbf{r}_i$ be the individual element on which $\mathcal{S}(\lambda_k)$ resides and let s_k be the length along $\mathbf{r}_i$ such that $\mathbf{r}_i(s_k) = \mathcal{S}(\lambda_k)$. Using the chain rule,

$$\begin{aligned}\frac{\partial \mathbf{r}_i}{\partial \mathbf{q}_j} &= \frac{\partial}{\partial \mathbf{q}_j}\mathbf{r}(\Omega_i, \mathbf{p}_i, \mathbf{t}_i, s_k) \\ &= \frac{\partial \mathbf{r}_i}{\partial \Omega_i}\frac{\partial \Omega_i}{\partial \mathbf{q}_j} + \frac{\partial \mathbf{r}_i}{\partial \mathbf{p}_i}\frac{\partial \mathbf{p}_i}{\partial \mathbf{q}_j} + \frac{\partial \mathbf{r}_i}{\partial \mathbf{t}_i}\frac{\partial \mathbf{t}_i}{\partial \mathbf{q}_j}.\end{aligned} \tag{17}$$

The leftmost terms in each product have direct derivatives. However, Ω_i depends on all $\mathbf{q}_j$ for $j \in [0, i]$, while $\mathbf{p}_i$, $\mathbf{t}_i$, $\mathbf{u}_i$, and $\mathbf{v}_i$ depend on all $\mathbf{q}_j$ for $j \in [0, i-1]$.

In the case of $j > i$, changing parameters later in the chain will obviously not affect the shape of $\mathbf{r}_i$, so those Jacobians vanish.

In the case of $j = i$, we note that from equations (5)-(8) $\frac{\partial \mathbf{p}_i}{\partial \mathbf{q}_i} = \frac{\partial \mathbf{t}_i}{\partial \mathbf{q}_i} = 0$ and $\frac{\partial \Omega_i}{\partial \mathbf{q}_i} = \mathbf{F}_i$, leaving a relatively direct expression,

$$\frac{\partial \mathbf{r}_i}{\partial \mathbf{q}_i} = \frac{\partial \mathbf{r}_i}{\partial \Omega_i}\mathbf{F}_i. \tag{18}$$

Letting $\hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$, $\hat{\tau}_i = \mathbf{t}_i \cdot \hat{\Omega}_i$, and $[\mathbf{v}]_\times$ be the skew-symmetric matrix such that $[\mathbf{v}]_\times \mathbf{u} = \mathbf{v} \times \mathbf{u}$, the gradient $\frac{\partial \mathbf{r}_i}{\partial \Omega_i}$ reads

$$\begin{aligned}\frac{\partial \mathbf{r}_i}{\partial \Omega_i} &= \frac{\hat{\Omega}_i \mathbf{t}_i^\top}{\|\Omega_i\|} s + \left(\mathbf{I} - 2\hat{\Omega}_i \hat{\Omega}_i^\top\right) \frac{\hat{\tau}_i s}{\|\Omega_i\|} \\ &+ \left(\mathbf{t}_i - \hat{\tau}_i \hat{\Omega}_i\right) \left(s\cos(\|\Omega_i\| s) - \frac{\sin(\|\Omega_i\| s)}{\|\Omega_i\|}\right) \frac{\hat{\Omega}_i^\top}{\|\Omega_i\|} \\ &- \frac{\sin(\|\Omega_i\| s)}{\|\Omega_i\|^2} \left(\hat{\Omega}_i \mathbf{t}_i^\top + \hat{\tau}_i \left(\mathbf{I} - 2\hat{\Omega}_i \hat{\Omega}_i^\top\right)\right) \\ &+ (\hat{\Omega}_i \times \mathbf{t}_i) \left(s\sin(\|\Omega_i\| s) - \frac{2 - 2\cos(\|\Omega_i\| s)}{\|\Omega_i\|}\right) \frac{\hat{\Omega}_i^\top}{\|\Omega_i\|} \\ &- \frac{1 - \cos(\|\Omega_i\| s)}{\|\Omega_i\|^2} [\mathbf{t}_i]_\times.\end{aligned} \tag{19}$$

In the case of $j < i$, we must rely on recursion to the case of $j = i - n$ by repeatedly decrementing i. We have $\frac{\partial \mathbf{r}_i}{\partial \mathbf{p}_i} = \mathbf{I}$, while the $\frac{\partial \mathbf{p}_i}{\partial \mathbf{q}_j}$ term reduces to

$$\frac{\partial \mathbf{p}_i}{\partial \mathbf{q}_j} = \frac{\partial}{\partial \mathbf{q}_j} \mathbf{r}(\Omega_{i-1}, \mathbf{p}_{i-1}, \mathbf{t}_{i-1}, l_{i-1}). \tag{20}$$

For the remaining $\frac{\partial \Omega_i}{\partial \mathbf{q}_j}$ and $\frac{\partial \mathbf{t}_i}{\partial \mathbf{q}_j}$ terms, we invoke a secondary recursive stack. To motivate this, let us write out $\frac{\partial \Omega_i}{\partial \mathbf{q}_j}$:

$$\begin{aligned}\frac{\partial \Omega_i}{\partial \mathbf{q}_j} &= \frac{\partial}{\partial \mathbf{q}_j} \left(\left[\begin{array}{c|c|c} \mathbf{t}_i & \mathbf{u}_i & \mathbf{v}_i \end{array}\right] \mathbf{q}_i\right) \\ &= \frac{\partial \mathbf{t}_i}{\partial \mathbf{q}_j} \kappa_{i,0} + \frac{\partial \mathbf{u}_i}{\partial \mathbf{q}_j} \kappa_{i,1} + \frac{\partial \mathbf{v}_i}{\partial \mathbf{q}_j} \kappa_{i,2}.\end{aligned} \tag{21}$$

Here, $\kappa_{i,k}$ are the components of $\mathbf{q}_i$. Knowledge of $\frac{\partial \mathbf{t}_i}{\partial \mathbf{q}_j}$ is necessary to begin differentiating Ω_i. Referring to equations (6)-(8) and equation (2), we reduce towards $j = i - n$,

$$\frac{\partial \mathbf{t}_i}{\partial \mathbf{q}_j} = \frac{\partial^2 \mathbf{r}(\Omega_{i-1}, \mathbf{t}_{i-1}, l_{i-1})}{\partial \mathbf{q}_j \partial s} \tag{22}$$

with similar expressions for the derivatives of $\mathbf{u}_i$ and $\mathbf{v}_i$.

Seeing this as a reduction rule, let us then analyze $\frac{\partial^2 \mathbf{r}(\Omega_i, \mathbf{t}_i, l_i)}{\partial \mathbf{q}_j \partial s}$ for general i and j and keeping in mind that analysis for the derivatives of $\mathbf{u}_i$ and $\mathbf{v}_i$ follow from swapping variables:

$$\frac{\partial}{\partial \mathbf{q}_j} \frac{\partial \mathbf{r}}{\partial s}(\Omega_i, \mathbf{t}_i, l_i) = \frac{\partial^2 \mathbf{r}}{\partial \Omega_i \partial s} \frac{\partial \Omega_i}{\partial \mathbf{q}_j} + \frac{\partial^2 \mathbf{r}}{\partial \mathbf{t}_i \partial s} \frac{\partial \mathbf{t}_i}{\partial \mathbf{q}_j}. \tag{23}$$

In the case of $j > i$ this Jacobian goes to zero since future parameters do not affect past element shapes.

In the case of $j = i$, $\frac{\partial \mathbf{t}_i}{\partial \mathbf{q}_j}$ goes to zero while the rest of the terms can be calculated directly as

$$\frac{\partial}{\partial \mathbf{q}_i} \frac{\partial \mathbf{r}}{\partial s}(\Omega_i, \mathbf{t}_i, l_i) = \frac{\partial^2 \mathbf{r}_i}{\partial \Omega_i \partial s} \frac{\partial \Omega_i}{\partial \mathbf{q}_i} = \frac{\partial^2 \mathbf{r}_i}{\partial \Omega_i \partial s} \mathbf{F}_i. \tag{24}$$

Differentiating equation (19) with respect to s gives

$$\begin{aligned}\frac{\partial^2 \mathbf{r}_i}{\partial \Omega_i \partial s} &= \frac{\hat{\Omega}_i \mathbf{t}_i^\top}{\|\Omega_i\|} + \left(\mathbf{I} - 2\hat{\Omega}_i \hat{\Omega}_i^\top\right) \frac{\hat{\tau}_i}{\|\Omega_i\|} \\ &- \left(\mathbf{t}_i - \hat{\tau}_i \hat{\Omega}_i\right) (s\sin(\|\Omega_i\| s))\, \hat{\Omega}_i^\top \\ &- \frac{\cos(\|\Omega_i\| s)}{\|\Omega_i\|} \left(\hat{\Omega}_i \mathbf{t}_i^\top + \hat{\tau}_i\right) \left(\mathbf{I} - 2\hat{\Omega}_i \hat{\Omega}_i^\top\right) \\ &+ (\hat{\Omega}_i \times \mathbf{t}_i) (s\|\Omega_i\|\cos(\|\Omega_i\| s) - \sin(\|\Omega_i\| s)) \frac{\hat{\Omega}_i^\top}{\|\Omega_i\|} \\ &- \frac{\sin(\|\Omega_i\| s)}{\|\Omega_i\|} [\mathbf{t}_i]_\times.\end{aligned} \tag{25}$$

This concludes the case of $j = i$.

Finally, in the case of $j < i$, equations (25), (21), and (22) can be used recursively to obtain $\frac{\partial^2 \mathbf{r}_i}{\partial \Omega_i \partial s}$, $\frac{\partial \Omega_i}{\partial \mathbf{q}_j}$, and $\frac{\partial \mathbf{t}_i}{\partial \mathbf{q}_j}$, respectively. This yields the term

$$\begin{aligned}\frac{\partial^2 \mathbf{r}_i}{\partial \mathbf{t}_i \partial s} &= \hat{\Omega}_i \hat{\Omega}_i^\top + \cos(\|\Omega_i\| s) \left(\mathbf{I} - \hat{\Omega}_i \hat{\Omega}_i^\top\right) \\ &+ \sin(\|\Omega_i\| s) [\hat{\Omega}_i]_\times.\end{aligned} \tag{26}$$

These results then allow us to recursively calculate $\frac{\partial \Omega_i}{\partial \mathbf{q}_j}$ and $\frac{\partial \mathbf{t}_i}{\partial \mathbf{q}_j}$. In conjunction with equation (20), we have everything we need for equation (17) to evaluate $\frac{\partial \mathbf{r}_i}{\partial \mathbf{q}_j}$ for all cases $0 \le i, j \le N - 1$.

5.2.1 Efficient Jacobian building. Evaluating $\frac{\partial \mathbf{r}_i}{\partial \mathbf{q}_j}$ analytically through equation (17) is achievable through recursion, but recursive calls quickly inflate evaluation time to unreasonable levels, especially whenever $\frac{\partial \Omega_i}{\partial \mathbf{q}_j}$ for $j < i$ appears and applying equation (21) results in the creation of three more call stacks. Compounded with the fact that constructing $\frac{\partial \mathbf{r}_i}{\partial \mathbf{q}}$ requires *all* $\frac{\partial \mathbf{r}_i}{\partial \mathbf{q}_j}$ for $j \in [0, N-1]$, an iterative solution that does not perform excess computation is preferable.

Iteratively building $\frac{\partial^2 \mathbf{r}}{\partial \mathbf{q}_n \partial s}(\Omega_m, \mathbf{t}_m, l_m)$, $\frac{\partial^2 \mathbf{r}}{\partial \mathbf{q}_n \partial s}(\Omega_m, \mathbf{u}_m, l_m)$, and $\frac{\partial^2 \mathbf{r}}{\partial \mathbf{q}_n \partial s}(\Omega_m, \mathbf{v}_m, l_m)$ for $0 \le n \le m \le i - 1$ allows for efficient reuse of results. Arraying each of these intermediate Jacobian sets in an $(i-1) \times (i-1)$ table and indexing by (m, n), the diagonal cases $n = m$ come directly from equations (24) and (25), while cases where $n < m$ can reference previously calculated derivatives in $(m-1, n)$ to plug into equation (23).

We additionally construct a fourth table $\frac{\partial \mathbf{r}}{\partial \mathbf{q}_n}(\Omega_m, \mathbf{p}_m, \mathbf{t}_m, l_m)$ for $0 \le n \le m \le i - 1$ and index by (m, n). We use equation (18) to fill in the cases where $n = m$, then use equation (17) along with the already-filled information in the $(m-1, n)$ entries of all four tables to calculate cases where $n < m$ (Alg. 1).

With all four tables filled, we apply equation (17) one last time, making sure to evaluate at s_k for the $\frac{\partial \mathbf{r}_i}{\partial \Omega_i}$ and $\frac{\partial \mathbf{r}_i}{\partial \mathbf{t}_i}$ terms while referencing tables at entry indices $(i-1, n)$ for other terms, resulting in a fully calculated Jacobian $\frac{\partial \mathbf{r}_i}{\partial \mathbf{q}}$. These tables are also reusable for any index $a < i$, referencing the indices at $(a-1, n)$ instead to calculate $\frac{\partial \mathbf{r}_a}{\partial \mathbf{q}}$ (Alg. 2).

Algorithm 1 Algorithm for filling intermediate endpoint and frame derivative tables

Input: Strand state $\mathbf{q}$, $\mathbf{p}_m$, $\mathbf{F}_m$, and l_m for $0 \le m \le i-1$
Output: Endpoint table $\frac{\partial \mathbf{r}}{\partial \mathbf{q}_n}(\Omega_m, \mathbf{p}_m, \mathbf{t}_m, l_m)$ and all frame tables $\frac{\partial^2 \mathbf{r}}{\partial \mathbf{q}_n \partial s}(\Omega_m, \mathbf{t}_m, l_m)$, $\frac{\partial^2 \mathbf{r}}{\partial \mathbf{q}_n \partial s}(\Omega_m, \mathbf{u}_m, l_m)$, and $\frac{\partial^2 \mathbf{r}}{\partial \mathbf{q}_n \partial s}(\Omega_m, \mathbf{v}_m, l_m)$ for $0 \le n \le m \le i-1$

```
for m = 0 to m = i − 1 do
    for n = 0 to n = m do
        if n == m then
            Use (24) to fill (m, n) on the three frame tables
            Use (18) to fill (m, n) on the endpoint table
        else
            Use (23) and (m − 1, n) to fill frame entries (m, n)
            Use (17) and (m − 1, n) to fill endpoint entry (m, n)
        end if
    end for
end for
```

Algorithm 2 Algorithm for finding full derivatives of multiple control point positions with respect to the full state vector $\mathbf{q}$

Input: Strand state $\mathbf{q}$, $\mathbf{p}_m$, $\mathbf{F}_m$ for $0 \le m \le N-1$, l_m for $0 \le m \le N-1$, k control points $\mathbf{r}_{i_k}(s_k)$
Output: All $\frac{\partial \mathbf{r}_{i_k}}{\partial \mathbf{q}}(\Omega_i, \mathbf{p}_i, \mathbf{t}_i, s_k)$

```
Obtain I = max(i_k)
Calculate size I × I endpoint/frame derivative tables (Alg. 1)
Calculate ∂r_{i_k}/∂q_j for j ≤ i_k using equation (17) and (i_k − 1, j) entries.
```

5.3 Strand collection control

As individual strands begin numbering in the thousands to hundred-thousands for full-resolution grooms, manually setting anchor points $\mathbf{a}_k$ for equation (15) becomes tedious.

To make control easier, we group the strands using k-means clustering [Hu et al. 2017b; Lloyd 1982; Tojo et al. 2025] over stacked root/endpoint position vectors. We then assign each cluster a set of sampling lengths λ_k to define control points for each strand, then set anchor points by applying affine transformations on the group of control points.

For interactivity, we set the affine transformation by allowing a user to manipulate a sequence of oriented frames whose initial positions and rotations are determined by their corresponding control point cluster centroids and principal directions, visualized as an armature system with frames at each joint (Fig. 11).

6 Implementation And Results

To perform optimization (15), we used an Adam optimizer [Kingma and Ba 2017] with initial conditions set to the original strand shape.

6.1 Cage deformation comparison

Optimizing strand geometries towards individual anchor points allows for the creation of more expressive motion with fewer degrees of freedom. To demonstrate this, we analyze the effectiveness of our method compared to cage deformation [Joshi et al. 2007] on the task of "twisting out" a lock of hair (Fig. 14).

Using cage deformation for the motion already requires a cage at a relatively high resolution, and multiple control meshes of higher resolution would be required to replicate sliding effects. On the other hand, our technique implicitly preserves strand lengths and allows for large degrees of relative motion between nearby super-helices, so intermediate geometries are less distorted.

Table 1. Timing information for groom morphing. Strand counts are specifically the number of edited strands. Timings are in minutes:seconds, with stars indicating the timings were taken from the slowest batch of 500 strands in a distributed run. All optimizations ran on 2.9GHz Intel Xeons.

Figure	Strands	Elements	c	Cores	Timing
Coily (Fig. 11)	100	45	30	1	03:40
Wavy (Fig. 11)	2025	45	10	1	143:34
Cage (Fig. 14)	2024	45	100	1	35:07
Eye cover (Fig. 8)	3292	90	20	8	13:41*
Ponytails (Fig. 5)	18000	45	0.01	8	04:29*
Stretch (Fig. 13)	90314	90	20	8	36:24*

6.2 Groom morphing

Responding to artist drawovers, we performed styling operations on various high-resolution polyline grooms that would have otherwise required resimulation or painstaking selection and deformation.

For *Code My Crown's* [Dove 2023] *Twist Out* groom, we pulled a part of the sweep down to cover the eye (Fig. 8). Additionally, we stretched an existing *long natural* groom to achieve a longer look by moving control points downwards (Fig. 13) and edited a straighter ponytail example (Fig. 5).

6.3 Ruffling

To showcase the effectiveness of ruffling, we use it to add naturalistic weathering to *Code My Crown's* [Dove 2023] *Headwrap Curls* groom. Compared to simply adding procedural frizz through Blender's geometry nodes [Blender 2025], our technique splays strands out in a more naturalistic manner while keeping control over individual wisp coherence (Fig. 1).

Additionally, using *Twist Out's* modifier stack [Dove 2023], we compare our ruffling against Maya's *XGen* grooming system [Autodesk 2026]. Their radius and frequency randomization achieves a different frizzy look compared to our ruffling operator (Fig. 6).

6.4 Feature analysis

Considering ω, R, and β from equations (12) and (14) allow us to analytically describe key features in highly-coiled hair. In a physically simulated model obtained after virtually combing a tightly coiled hair wisp [Crespel et al. 2024], we specifically identify instances of switchbacks [Wu et al. 2024] by visualizing such values with coloring and explicit plotting (Fig. 12).

6.5 Performance

Because each super-helix optimization is self-contained, multithreading and batching allows for further increases in speed. In practice,

Fig. 5. An artist provides direction for bending ponytails as though hair rollers were being inserted (first/second). Our armature moves the hair into place (third), allowing the artist to visualize a composite (fourth). Reference photo from Dano [2024].

Fig. 6. An XGen-generated swatch (left) with native noise modifiers applied (middle) compared to our ruffling operation ($\rho = 0.01,\ N_m = 30$) (right). Ruffling achieves a distinct look compared to noise, even when starting from modifier-based procedural geometry.

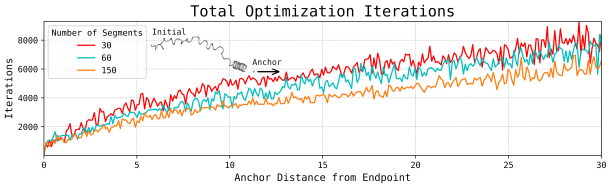
we ran on either one or eight cores with chunking in groups of 500 strands for larger grooms. Strand counts, per-strand segment counts, and timing statistics are available in Table 1.

6.5.1 Optimizer details. For every figure, we set the Adam step size to 0.001, the first moment β_1 to 0.9, the second moment β_2 to 0.999, and the denominator stabilizer ϵ to 10^{-8}. Stepping concluded when the residual was less than 10^{-6}, when 4000 steps had passed, or when 50 steps had passed without any improvement in the residual.

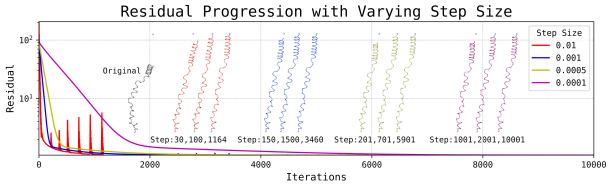
Iteration counts varied depending on target location, number of helical segments, and optimizer parameters. Running single-strand optimizations, the number of iterations until convergence generally increases with initial anchor-point distance, decreases with the number of segments, and decreases as step size increases up to some stability threshold (Fig. 7). Code for running strand endpoint optimizations is also available in the supplemental materials.

6.6 Gradient calculations

Derivative calculations for every optimization step was the primary timing bottleneck in computing equation (15). Compared to PyTorch's automatic differentiation —the fastest alternative we could find—we observed speedups around 47x, which can be attributed to our analytic method leveraging repeated values in what would otherwise be an unwieldy recursive calculation tree, possibly over the span of several calls when considering multiple control points.



(a) As helical segment count decreases/anchor point distance increases, the number of iterations to convergence increases.



(b) As Adam step size decreases, the number of required iterations increases. Several intermediate strand geometries at pre-convergence steps are also overlayed.

Fig. 7. Single-strand optimization analyses.

7 Conclusions

By constructing general curves in terms of their local curvatures, super-helices are naturally suited for the representation and manipulation of highly-coiled hair. However, there are several limitations to these current techniques that are worthy of future work.

7.1 Limitations

The optimization of single super-helix strands of up to 100 elements is fast enough to be considered interactive, but our techniques lose interactivity with multistrand grooms. Furthermore, limiting the state vectors $\mathbf{q}$ to only consider local curvatures while ignoring the length of each individual element narrows the range of curves that can interpolate between one another. In general, modifying local curvatures towards particular goals like increasing or decreasing spatial frequency or radii remains difficult, and preserving the overall direction of a curly strand without large deflections is nontrivial.

7.2 Future work

Higher-order methods for optimizing (15) may improve solving time, and further applications of our analytic super-helix derivatives in strand geometry processing and simulation also seem promising. Exploring other interaction modes such as sculpting, cutting, or combing, might be more well-suited towards the notion of a 'digital salon.' Finally, modifying the penalty energy to incorporate contact interactions or promote particular strand shapes are also interesting directions for further investigation.

Acknowledgments

This work was supported by the Teng and Han Family Fund, the Bungie Foundation, and NSF IIS-2132280. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

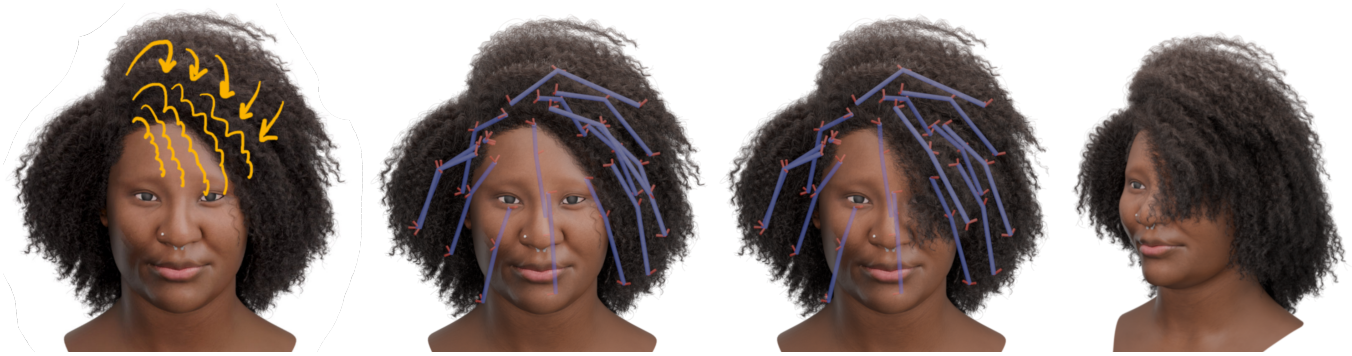


Fig. 8. An artist provides direction for moving the hair down to cover the eye (first), we fit an armature over the groom (second), and through moving one joint, achieve the desired pose (third/fourth).

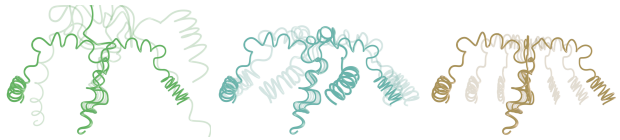


Fig. 9. Interpolating between two strand shapes using global Darboux coordinates (left), local curvature coordinates (middle), and shape keys (right). The two endpoints, along with the exact halfway point, are shown at full transparency. Both the global Darboux interpolation and shape keys distort the strand, while interpolating over local curvatures gives satisfactory results. See the video for more detail.

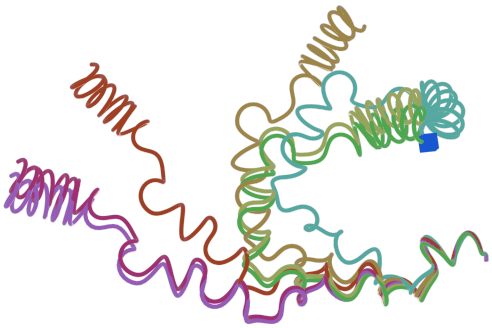


Fig. 10. Optimizing a single strand (pink) to have its endpoint reach the blue cube. c is set at 100,000 (purple), 10,000 (red), 1,000 (orange), 100 (yellow), 10 (green), and 0 (light blue).

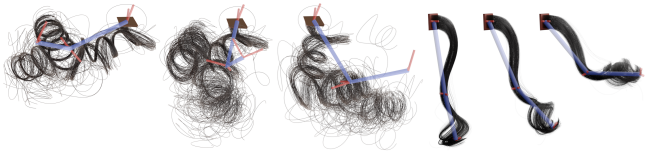


Fig. 11. Manipulating a highly-coiled wisp of 100 super-helices using a single-jointed armature (left), along with a wavier coil of 2025 super-helices (right). Both wisps were originally generated from physically-based combing simulation by Crespel et al. [2024]. See the video for more detail.

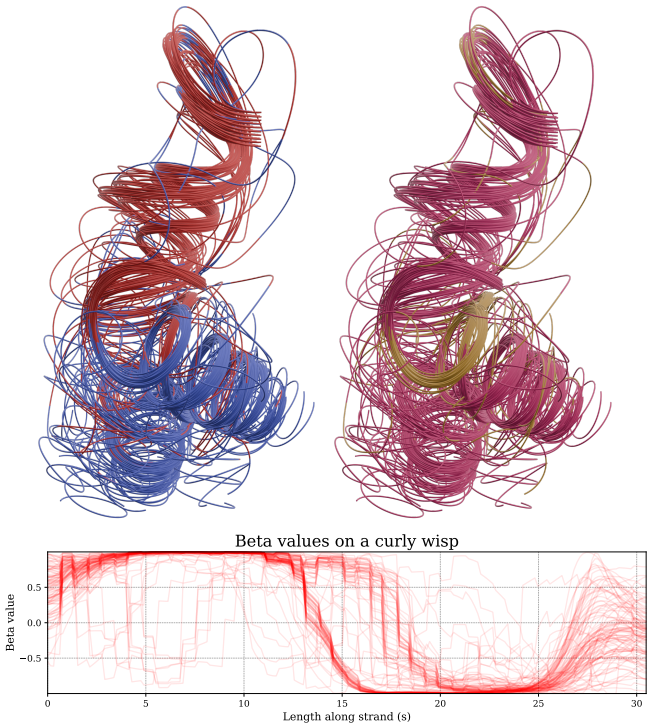


Fig. 12. Visualization of ω (Eqn. 12) and β (Eqn. 14) on a simulated strand example (highly-coiled wisp of Fig. 11, left) provided by Crespel et al. [2024]. On the left, positive ω transitions over to negative ω, as colored by red and blue, respectively. On the right, pink and yellow represent $|\beta| > 0.5$ and $|\beta| \leq 0.5$, allowing for easier localization of wavy features. On the bottom, a line graph of every strand's β as a function of strand length.

References

Ken-ichi Anjyo, Yoshiaki Usami, and Tsuneya Kurihara. 1992. A simple method for extracting the natural beauty of hair. In *Proceedings of SIGGRAPH*. 111–120. doi:10.1145/142920.134021

Basile Audoly and Yves Pomeau. 2010. *Elasticity and Geometry: From hair curls to the nonlinear response of shells.* Oxford University Press, Oxford, New York.

Autodesk. 2026. Maya Help: XGen Interactive Grooming, Autodesk. https://help.autodesk.com/view/MAYAUL/2024/ENU/?guid=GUID-496603B0-

Fig. 13. Fitting an armature over a *long natural* groom, we follow artist directions to produce a longer groom by stretching the control points outwards and optimizing (top). The resulting texture closely matches what is achieved by stretching afro hair in real life by using a blowdryer and paddle brush or comb. An example is referenced from GARSHELL [2023] (bottom left).

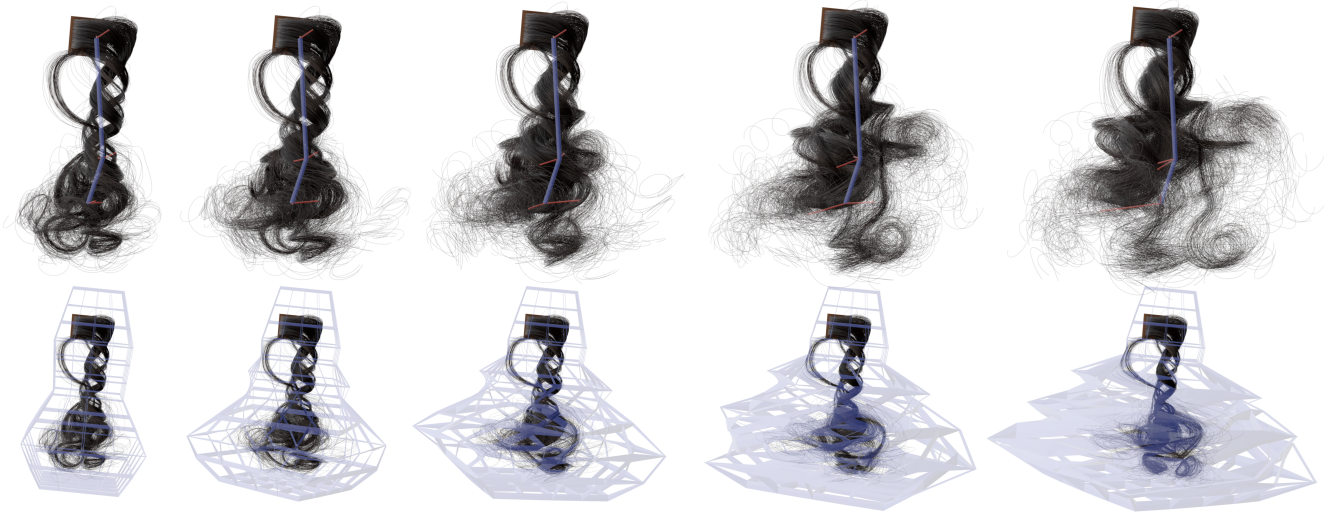


Fig. 14. Twisting control frames versus twisting a cage on the curly wisp of Crespel et al. [2024] composed of 2025 super-helices. Even with significant subdivision, the twisting and scaling of the cage simply distorts the strand while our method splays individual strands outwards. See the video for more detail.

F929-45CD-B607-1CFCD3283DBE
Thomas F. Banchoff and Stephen Lovett. 2022. *Differential Geometry of Curves and Surfaces*. CRC Press. Google-Books-ID: awJ3EAAAQBAJ.
F. Bertails. 2006. *Simulation of Virtual Hair*. Ph. D. Dissertation. Institut National Polytechnique de Grenoble.
Florence Bertails. 2009. Linear time super-helices. In *Computer graphics forum*, Vol. 28. Wiley Online Library, 417–426. doi:10.1111/j.1467-8659.2009.01381.x
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. In *ACM SIGGRAPH 2006 Papers (SIGGRAPH '06)*. Association for Computing Machinery, New York, NY, USA, 1180–1187. doi:10.1145/1179352.1142012
Florence Bertails, Basile Audoly, Bernard Querleux, Frédéric Leroy, Jean-Luc Lévêque, and Marie-Paule Cani. 2005. Predicting Natural Hair Shapes by Solving the Statics of Flexible Rods. In *Eurographics Short Papers*, Dingliana J. et Ganovelli F. (Ed.). Eurographics, Dublin, Ireland. doi:10.2312/egs.20051029
Florence Bertails-Descoubes, Florent Cadoux, Gilles Daviet, and Vincent Acary. 2011. A nonsmooth Newton solver for capturing exact Coulomb friction in fiber assemblies. *ACM Transactions on Graphics (TOG)* 30, 1 (2011), 1–14. doi:10.1145/1899404.1899410
Florence Bertails-Descoubes, Alexandre Derouet-Jourdan, Victor Romero, and Arnaud Lazarus. 2018. Inverse design of an isotropic suspended Kirchhoff rod: theoretical

and numerical results on the uniqueness of the natural shape. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 474, 2212 (2018). arXiv:http://rspa.royalsocietypublishing.org/content/474/2212/20170837.full.pdf doi:10.1098/rspa.2017.0837

Blender. 2025. *Blender - Frizz Hair Curves.* Blender Foundation, Blender Institute, Amsterdam. https://docs.blender.org/manual/en/latest/modeling/geometry_nodes/hair/deformation/frizz_hair_curves.html

Stephen D. Bowline, Andrew A. Johnson, and Ryan Gillis. 2016. HairCraft: spells and incantations for digital hair. In *ACM SIGGRAPH 2016 Talks* (Anaheim, California) *(SIGGRAPH '16)*. Association for Computing Machinery, New York, NY, USA, Article 31, 2 pages. doi:10.1145/2897839.2927401

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*. 1–2. doi:10.1145/3214745.3214798

Menglei Chai, Linjie Luo, Kalyan Sunkavalli, Nathan Carr, Sunil Hadap, and Kun Zhou. 2015. High-quality hair modeling from a single portrait photo. *ACM Trans. Graph.* 34, 6 (2015), 204–1. doi:10.1145/2816795.2818112

Wesley Chang, Andrew L. Russell, Stephane Grabli, Matt Jen-Yuan Chiang, Christophe Hery, Doug Roble, Ravi Ramamoorthi, Tzu-Mao Li, and Olivier Maury. 2025. Transforming Unstructured Hair Strands into Procedural Hair Grooms. *ACM Trans. Graph.* 44, 4 (July 2025), 105:1–105:20. doi:10.1145/3731168

Beyzanur Coban, Pascal Chang, Guilherme Gomes Haetinger, Jingwei Tang, and Vinicius C. Azevedo. 2025. Shaping Strands with Neural Style Transfer. *ACM Trans. Graph.* 44, 6, Article 239 (Dec. 2025), 13 pages. doi:10.1145/3763365

Octave Crespel, Emile Hohnadel, Thibaut Métivet, and Florence Bertails-Descoubes. 2024. Contact detection between curved fibres: high order makes a difference. *ACM Transactions on Graphics* 43, 4 (Aug. 2024), 132:1–23. doi:10.1145/3658191 Publisher: Association for Computing Machinery.

Curlsmith. 2021. Bonding Oil. https://eu.curlsmith.com/products/bonding-oil

Afnan Dano. 2024. Hair Products Repurchasing. TikTok. https://www.tiktok.com/@afnandano/video/7351854634190851334

Megha Davalath, Terran Boylan, Rob O'Neill, and Rob Vogt. 2021. Wig: The Hair Story From Shrek 2 to The Croods: A New Age. In *ACM SIGGRAPH 2021 Talks*. 1–2. doi:10.1145/3450623.3464667

Gilles Daviet, Florence Bertails-Descoubes, and Laurence Boissieux. 2011. A hybrid iterative solver for robustly capturing coulomb friction in hair dynamics. *ACM Transactions on Graphics* 30, 6 (2011), 1–12. doi:10.1145/2024156.2024173

Fernando De Goes and Doug L James. 2017. Regularized kelvinlets: sculpting brushes based on fundamental solutions of elasticity. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–11. doi:10.1145/3072959.3073595

Fernando de Goes and Doug L James. 2019. Sharp kelvinlets: Elastic deformations with cusps and localized falloffs. In *Proceedings of the 2019 Digital Production Symposium*. 1–8. doi:10.1145/3329715.3338884

Alexandre Derouet-Jourdan, Florence Bertails-Descoubes, Gilles Daviet, and Joëlle Thollot. 2013b. Inverse dynamic hair modeling with frictional contact. *ACM Transactions on Graphics (TOG)* 32, 6 (2013), 1–10. doi:10.1145/2508363.2508398

Alexandre Derouet-Jourdan, Florence Bertails-Descoubes, and Joëlle Thollot. 2013a. Floating tangents for approximating spatial curves with G1 piecewise helices. *Computer Aided Geometric Design* 30, 5 (June 2013), 490–520. doi:10.1016/j.cagd.2013.02.007

Dove. 2023. GitHub - dove-us/code-my-crown. https://github.com/dove-us/code-my-crown/tree/main

Tao Du, Kui Wu, Pingchuan Ma, Sebastien Wah, Andrew Spielberg, Daniela Rus, and Wojciech Matusik. 2021. Diffpd: Differentiable projective dynamics. *ACM Transactions on Graphics (ToG)* 41, 2 (2021), 1–21. doi:10.1145/3490168

Kurt Fleischer, Paul Isaacs, Bret Parker, Bernhard Haux, Sarah Shen, Venkat Krishna, Chen Shen, Andrew Butts, Jayson Price, Tom Hahn, et al. 2015. Silhouette sketching on" inside out". In *ACM SIGGRAPH 2015 Talks*. 1–1. doi:10.1145/2775280.2792566

Stefan Fröhlich and Mario Botsch. 2011. Example-Driven Deformations Based on Discrete Shells. *Computer Graphics Forum* 30, 8 (2011), 2246–2257. doi:10.1111/j.1467-8659.2011.01974.x _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2011.01974.x.

GARSHELL. 2023. How I Blowout My Thick Natural Hair | No Heat Damage (Type 4). https://www.youtube.com/@GARSHELL

Rasmus Haapaoja and Christoph Genzwürker. 2019. Mesh-driven generation and animation of groomed feathers. In *ACM SIGGRAPH Talks*. Article 61, 2 pages. doi:10.1145/3306307.3328178

Sunil Hadap and Nadia Magnenat-Thalmann. 2000. Interactive hair styler based on fluid flow. In *Proceedings of Computer Animation and Simulation*. Eurographics Association. doi:10.1007/978-3-7091-6344-3_7

Daniela Hasenbring and Henrik Karlsson. 2021. Hair Grooming with Imageworks' Fyber. In *ACM SIGGRAPH Talks*. Article 37, 2 pages. doi:10.1145/3450623.3464668

Jerry Hsu, Tongtong Wang, Zherong Pan, Xifeng Gao, Cem Yuksel, and Kui Wu. 2023. Sag-free initialization for strand-based hybrid hair simulation. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–14. doi:10.1145/3592143

Jerry Hsu, Tongtong Wang, Zherong Pan, Xifeng Gao, Cem Yuksel, and Kui Wu. 2024. Real-time Physically Guided Hair Interpolation. *ACM Trans. Graph.* 43, 4 (July 2024), 95:1–95:11. doi:10.1145/3658176

Liwen Hu, Derek Bradley, Hao Li, and Thabo Beeler. 2017a. Simulation-Ready Hair Capture. *Computer Graphics Forum* (2017). doi:10.1111/cgf.13126 Publisher: The Eurographics Association and John Wiley & Sons Ltd..

Liwen Hu, Shunsuke Saito, Lingyu Wei, Koki Nagano, Jaewoo Seo, Jens Fursund, Iman Sadeghi, Carrie Sun, Yen-Chun Chen, and Hao Li. 2017b. Avatar digitization from a single image for real-time rendering. *ACM Trans. Graph.* 36, 6 (Nov. 2017), 195:1–195:14. doi:10.1145/3130800.31310887

Hayley Iben, Mark Meyer, Lena Petrovic, Olivier Soares, John Anderson, and Andrew Witkin. 2013. Artistic simulation of curly hair. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 63–71. doi:10.1145/2485895.2485913

Alec Jacobson, Ilya Baran, Jovan Popovic, and Olga Sorkine. 2011. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.* 30, 4 (2011), 78. doi:10.1145/2010324.1964973

Pushkar Joshi, Mark Meyer, Tony DeRose, Brian Green, and Tom Sanocki. 2007. Harmonic coordinates for character articulation. *ACM Trans. Graph.* 26, 3 (July 2007), 71–es. doi:10.1145/1275808.1276466

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

Tae-Yong Kim and Ulrich Neumann. 2002. Interactive multiresolution hair modeling and editing. *ACM Transactions on Graphics (TOG)* 21, 3 (2002), 620–629. doi:10.1145/566570.566627

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG] doi:11245/1.505367

Yaron Lipman, David Levin, and Daniel Cohen-Or. 2008. Green Coordinates. *ACM Transactions on Graphics* 27, 3 (2008), 1–10. doi:10.1145/1399504.1360677

S. Lloyd. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory* 28, 2 (March 1982), 129–137. doi:10.1109/TIT.1982.1056489

Brandon Montell, Fernando de Goes, and Jacob Brooks. 2022. Hair Emoting with Style Guides in Turning Red. In *ACM SIGGRAPH 2022 Talks* (Vancouver, BC, Canada) *(SIGGRAPH '22)*. Association for Computing Machinery, New York, NY, USA, Article 25, 2 pages. doi:10.1145/3532836.3536253

Soorya Narayan. 2023. Hair Tubes: Stylized Hair from Polygonal Meshes of Arbitrary Topology. In *SIGGRAPH Asia 2023 Technical Communications*. 1–4. doi:10.1145/3610543.3626157

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. doi:10.1145/3532836.3536277

Deborah Patrick, Shaun Bangay, and Adele Lobb. 2004. Modelling and rendering techniques for african hairstyles. In *Proceedings of Afrigraph*. 115–124. doi:10.1145/1029949.1029971

Shunsuke Saito, Liwen Hu, Chongyang Ma, Hikaru Ibayashi, Linjie Luo, and Hao Li. 2018. 3D hair synthesis using volumetric variational autoencoders. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–12. doi:10.1145/3272127.3275019

Christian Schumacher, Bernhard Thomaszewski, Stelian Coros, Sebastian Martin, Robert Sumner, and Markus Gross. 2012. Efficient simulation of example-based materials. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '12)*. Eurographics Association, Goslar, DEU, 1–8. https://dl.acm.org/doi/10.5555/2422356.2422358

Thomas W Sederberg and Scott R Parry. 1986. Free-form deformation of solid geometric models. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*. 151–160. doi:10.1145/15922.15903

Alvin Shi, Haomiao Wu, Jarred Parr, A. M. Darke, and Theodore Kim. 2023. Lifted Curls: A Model for Tightly Coiled Hair Simulation. *Proc. ACM Comput. Graph. Interact. Tech.* 6, 3, Article 42 (aug 2023), 19 pages. doi:10.1145/3606920

Arunachalam Somasundaram. 2015. Dynamically controlling hair interpolation. In *ACM SIGGRAPH 2015 Talks* (Los Angeles, California) *(SIGGRAPH '15)*. Association for Computing Machinery, New York, NY, USA, Article 36, 1 pages. doi:10.1145/2775280.2792541

Tuur Stuyck and Hsiao-yu Chen. 2023. Diffxpbd: Differentiable position-based simulation of compliant constraint dynamics. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 6, 3 (2023), 1–14. doi:10.1145/3606923

Tuur Stuyck, Gene Wei-Chin Lin, Egor Larionov, Hsiao-yu Chen, Aljaz Bozic, Nikolaos Sarafianos, and Doug Roble. 2025. Quaffure: Real-Time Quasi-Static Neural Hair Simulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 239–249. doi:10.1109/CVPR52734.2025.00031

Tetsuya Takahashi and Christopher Batty. 2025. Rest Shape Optimization for Sag-Free Discrete Elastic Rods. *Computer Graphics Forum* 44, 2 (2025), e70019. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.70019 doi:10.1111/cgf.70019

Kenji Tojo, Liwen Hu, Nobuyuki Umetani, and Hao Li. 2025. Strands2Cards: Automatic Generation of Hair Cards from Strands. In *Proceedings of the SIGGRAPH Asia 2025*

Conference Papers. 1–11. doi:10.1145/3757377.3763864

Audrey Wong, David Eberle, and Theodore Kim. 2018. Clean cloth inputs: Removing character self-intersections with volume simulation. In *ACM SIGGRAPH 2018 Talks*. 1–2. doi:10.1145/3214745.3214786

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 (SA '24)*. Association for Computing Machinery, New York, NY, USA, 1–11. doi:10.1145/3680528.3687641

Cem Yuksel, Scott Schaefer, and John Keyser. 2009. Hair meshes. *ACM Transactions on Graphics (TOG)* 28, 5 (2009), 1–7. doi:10.1145/1661412.1618512

Zhongtian Zheng, Tao Huang, Haozhe Su, Xueqi Ma, Yuefan Shen, Tongtong Wang, Yin Yang, Xifeng Gao, Zherong Pan, and Kui Wu. 2025. Auto Hair Card Extraction for Smooth Hair with Differentiable Rendering. *ACM Transactions on Graphics (TOG)* 44, 6 (2025), 1–13. doi:10.1145/3763295

Yuxiao Zhou, Menglei Chai, Alessandro Pepe, Markus Gross, and Thabo Beeler. 2023. Groomgen: A high-quality generative hair model using hierarchical latent representations. *ACM Transactions on Graphics (TOG)* 42, 6 (2023), 1–16. doi:10.1145/3618309