Curly-Cue: Geometric Methods for Highly Coiled Hair

HAOMIAO WU* and ALVIN SHI*, Yale University, USA
A.M. DARKE, University of California, Santa Cruz, USA
THEODORE KIM, Yale University, USA

Fig. 1. 234K highly coiled hairs generated by combining our *phase locking*, *period skipping*, and *switchback* methods. The head is modelled after *New York Times* bestselling author Carvell Wallace [Iguodala and Wallace 2019], and used with his permission.

We present geometric methods for generating shapes that are characteristic of highly coiled hair. Different features become visually relevant when hairs are well-approximated by high-frequency helices instead of low-frequency curves, so we present algorithms for three such phenomena. First, a Fourier-based method for *phase locking*, the process by which disparate helices near the scalp coalesce into a single curl. Second, a method for *period skipping* which models individual helices deviating from the coalesced curl. Third, a non-linear optimization that directly generates the shapes of *switchbacks*, a.k.a. helical perversions, which heretofore could only be produced through direct physical simulation. By applying all three methods in tandem, we show that we can achieve richly detailed depictions of highly coiled hair.

CCS Concepts: • **Computing methodologies → Shape modeling**.

Additional Key Words and Phrases: Hair Modeling, Hairstyling

*Joint first authors

Authors' Contact Information: Haomiao Wu, haomiao.wu@yale.edu; Alvin Shi, Yale University, USA, alvin.shi@yale.edu; A.M. Darke, University of California, Santa Cruz, USA, darke@ucsc.edu; Theodore Kim, Yale University, USA, theodore.kim@yale.edu.

SA Conference Papers '24, December 3–6, 2024, Tokyo, Japan

ACM ISBN 979-8-4007-1131-2/24/12
https://doi.org/10.1145/3680528.3687641

ACM Reference Format:
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 Conference Papers '24), December 3–6, 2024, Tokyo, Japan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3680528.3687641

1 Introduction

Hair modeling and simulation has been an active topic in computer graphics research for over three decades [Anjyo et al. 1992]. While many advances have been made, they are usually demonstrated on straight [McAdams et al. 2009], wavy [Kaufman et al. 2014], or loosely curled hair [Iben et al. 2013]. A handful of papers [Bertails et al. 2005; Shi et al. 2023] have dealt with highly coiled, Afro-textured hair, but only one full paper [Patrick et al. 2004] and one short paper [Ogunseitan 2022] has specifically addressed the geometric challenges of modelling this type of hair.

In this paper, we show that when highly coiled hairs, i.e. high-frequency helices, are assumed to be the base primitive, then a variety of challenging geometric phenomena appear that merit further algorithmic investigation.

First, we examine how a group of disparate hairs coalesce into a single coherent curl as they travel away from the scalp. We call this phenomenon *phase locking*, borrowing the term's use in neuroscience [Kolev and Yordanova 1997] to describe multiple wavy signals falling into lock-step. Subsequently, we propose a data-driven, Fourier-based method for computing this geometry to capture the "spongy" texture of highly coiled hair near the scalp.

Second, we present a direct geometric method for generating *switchbacks* along a helix. These structures are also known as “changes in handedness” or “helical perversions”. While they are common in curly hair, the only previous method for generating them has been high-resolution physics simulation [Bergou et al. 2008; Crespel et al. 2024]. We build on static analyses from physics [McMillen and Goriely 2002] to formulate a non-linear optimization that can insert physically valid switchbacks anywhere on a helix.

Finally, we present a method for *period skipping*, a phenomenon related to phase locking where an individual hair temporarily breaks away from its coalesced curl. Precisely capturing this feature is crucial to achieving the range of diffuse and shiny looks characteristic of highly coiled hair. Our contributions are as follows:

- A data-driven, Fourier method for computing phase locking between an ensemble of hairs.
- A non-linear optimization method for inserting physically valid switchbacks anywhere along a helix.
- A simple geometric method for computing period skipping within a coalesced curl of hair.

To demonstrate both the realism and scalability of our methods, we include human-scale examples and several swatches.

2 Related Works

Many works on hair rendering, modelling, and simulation have been developed over the years, and excellent surveys and courses are available [Ward et al. 2007; Yuksel and Tariq 2010]. Here, we list the works most relevant to our contributions.

2.1 Hair Interpolation

Directly simulating and modelling every hair on a human head has long been impractical, though methods are constantly improving [Daviet 2023]. Instead, a widespread practice is to select a subset of “guide” hairs for modelling and simulation, and to interpolate a full set of hairs from these guides. This approach has been used since at least Watanabe and Suenaga [1992], who observed that nearby hairs usually resemble each other, so duplicating a single guide into thicker *wisps* is a viable strategy. Subsequently, wisps have also accelerated simulations [Bertails et al. 2003; Choe et al. 2005].

Similar clustering notions appear in Disney’s “groom tubes” [Kaur et al. 2018], and Animal Logic’s “hair tubes” [Narayan 2023]. Pixar’s hair system performs nearest neighbor interpolation [Butts et al. 2018] while Sony Imageworks [Hasenbring and Karlsson 2021] interpolates between guides and “final” hairs. These approaches are mainly for straight or wavy hair. We instead present strategies that are specifically tailored to high-frequency hair.

Many existing systems support *scraggle* [Butts et al. 2018; Haapaoja and Genzwürker 2019; Narayan 2023], which uses noise functions (e.g. Perlin [1985]) to add variations to hair. While this approach is effective for straight and wavy hair, we will show that highly coiled hair contains features that cannot be captured using this family of spatially local functions alone.

2.2 Hair Modelling

Image-based methods for hair have been refined over the last two decades [Paris et al. 2004], but many report issues when trying to capture high-frequency features like curly hair [Jakob et al. 2009; Sun et al. 2021; Zhang et al. 2017] or braids [Liang et al. 2018]. Much of the difficulty stems from the images, which force the algorithms to reconstruct complex internal structures using only surface data. Recent methods have used volumetric CT data [Shen et al. 2023], which is significantly costlier than camera images, or neural methods [Rosu et al. 2022] that only demonstrate efficacy on straight hair. The geometric features for highly coiled hair we describe here could potentially be used to inform future image-based methods.

Mesh-based [Yuksel et al. 2009] and multi-resolution [Kim and Neumann 2002] methods have also been developed for direct manipulation of hair, but again are usually demonstrated on straight and wavy hair. We present techniques for highly coiled hair.

2.3 Hair Simulation

Guide hairs can be driven by spring-mass [Iben et al. 2013; Selle et al. 2008], Discrete Elastic Rod [Bergou et al. 2008], Super-Helix [Bertails et al. 2003], Position-Based [Umetani et al. 2015], or finite element-based [Shi et al. 2023] simulation methods.

Our methods are agnostic to the simulation method, including pre-processes like inverse [Derouet-Jourdan et al. 2013] or sag-free optimizations [Hsu et al. 2023]. However, to demonstrate generality, we show results that involve both Discrete Elastic Rods [Bergou et al. 2008] and Lifted Curls [Shi et al. 2023].

2.4 Hair Classification

We are investigating highly coiled hair, which under different classification systems might be called Type 4c, Type VIII [De La Mettrie et al. 2007], or Type-O [Mafro and Mafro 2013]. We forego committing to a specific typing system, because they can contain key limitations [Kim et al. 2022]. For example, Type 4c was informally added to the original Walker [1997] system after criticism that Walker had failed to include this historically neglected hair type [CurlCentric 2014]. Instead, we follow the recommendation of Darke et al. [2024] and use the qualitative hair texture descriptor “coiled”.

2.5 Related Phenomena

The graphics phenomenon most related to our notion of *phase locking* is the fiber-level modelling of yarns [Montazeri et al. 2019], where micro-fibers combine into a larger (straight) yarn. We instead track single helices that combine into a larger (non-straight) helix.

Our notion of switchbacks also appears in plants [Wang et al. 2013], bacteria [Mendelson 1978], and polymers [Wie et al. 2015]. While these prior works analyze the conditions under which switchbacks occur, none ever instantiate the geometry. We present the first method for directly computing the actual curve.

These shapes have also been called “spiral inversions” [Goriely and Tabor 1998], “helix reversals” [Hancock et al. 2020], “tendril perversions”, and ”helical perversions”. We choose to avoid using the term “perversion” to describe a human feature, and instead opt for the shorter and more descriptive term *switchback*.

3 Synthesizing Highly Coiled Hairs

We first examine the phenomenon of *phase locking*, where disparate hairs coalesce into a single coherent curl (Fig. 2). Existing methods

Fig. 2. Phase locking in Instagram photo from @theesudani. The hairs form a "spongy" layer near the scalp (light blue), but then coalesce into wisps (light red). The phenomenon was accentuated using "finger coiling." ©Yar Sudani

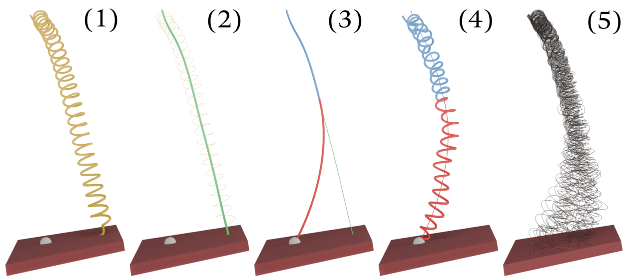


Fig. 3. Synthesizing a single strand. (1) Start with a root point (gray) and a guide strand (yellow). (2) Extract the guide's centerline (green), §3.2.1. (3) Interpolate from the root to the guide's centerline (red). (4) Apply data-driven helicity to the interpolated strand, §3.3.3. (5) Repeat to obtain a wisp.

that linearly interpolate guide hairs [Butts et al. 2018] do not suffice, because rich frequency information is destroyed under linear interpolation, e.g. averaging a sine wave with its $\pi/2$ shifted counterpart yields a line. We will instead present a frequency-aware alternative.

Distinct geometric regions emerge in highly coiled hair that are absent in straighter hair. The uncorrelated, highly coiled hairs near the scalp form a "spongy" texture, and neglecting this region creates the impression of wig instead of naturalistic growth. (Fig. 10, left) Further from the scalp, the hairs coalesce into curls. (Fig. 3, right) We explicitly model both of these regions.

3.1 Geometry Preliminaries

3.1.1 Piecewise Spline. We model each hair strand as a C^1 continuous curve $\mathbf{c}(t)$ in $\mathbb{R}^3$ composed of a piecewise cubic Hermite (Catmull-Rom) spline. Given a sequence of points $\{\mathbf{p}_0, ..., \mathbf{p}_n\}$ along a strand, the piecewise spline $\mathbf{c}_i(t)$ connecting $\mathbf{p}_i$ to $\mathbf{p}_{i+1}$ over the interval $t \in [i, i+1)$ is given by

$$\mathbf{c}_i(i) = \mathbf{p}_i \qquad \mathbf{c}_i(i+1) = \mathbf{p}_{i+1}$$

$$\frac{\partial \mathbf{c}_i(i)}{\partial t} = \frac{\mathbf{p}_{i+1} - \mathbf{p}_{i-1}}{2} \qquad \frac{\partial \mathbf{c}_i(i+1)}{\partial t} = \frac{\mathbf{p}_{i+2} - \mathbf{p}_i}{2}$$

At the ends $\mathbf{c}_0$ and $\mathbf{c}_{n-1}$, we do a weighted average of the slopes at the two closest points:

$$\frac{\partial \mathbf{c}_0(0)}{\partial t} = (\mathbf{p}_2 - \mathbf{p}_0) - \frac{(\mathbf{p}_3 - \mathbf{p}_1)}{2}$$

$$\frac{\partial \mathbf{c}_{n-1}(n)}{\partial t} = (\mathbf{p}_n - \mathbf{p}_{n-2}) - \frac{(\mathbf{p}_{n-1} - \mathbf{p}_{n-3})}{2}$$

The final piecewise curve is then $\mathbf{c}(t) = \mathbf{c}_i(t),\ t \in [i, i+1)$. For simplicity, we use $t \to nt$ to remap the domain of $\mathbf{c}(t)$ to $[0, 1]$.

3.1.2 Curve Frames. Our curve $\mathbf{c}(t)$, needs $\mathbf{u}(t)$ and $\mathbf{v}(t)$ directions that span the plane orthogonal to the curve's direction $\partial \mathbf{c}(t)/\partial t$, yielding an oriented orthogonal frame $\{\mathbf{u}, \mathbf{v}, \partial \mathbf{c}(t)/\partial t\}$. There are many ways to compute $\mathbf{u}(t)$ and $\mathbf{v}(t)$, such as Frenet frames or parallel transport. Our method is agnostic to any specific technique; we describe our specific strategy in §5.1.1.

3.2 Frequency Analysis of Strands

To convert a single guide hair into multiple hairs coalescing into a wisp, we perform two separate frequency analyses. First, a Fourier analysis extracts the *low-frequency centerline* of each guide hair. Second, we use the centerline algorithm to extract *high frequency spectra* from a dataset of full-resolution simulations. These spectra are then used in §3.3 to synthesize high-frequency "spongy" regions.

3.2.1 Centerline Extraction. An obvious candidate for Fourier analysis is the x, y, and z components of the curve points $\mathbf{p}_i$. We instead prefer the translation-agnostic method of Zhou et al. [2023], which constructs the displacement-based sequence $\mathbf{x}_i = \mathbf{p}_{i+1} - \mathbf{p}_i$, and then performs a separate DFT along each component of $\{\mathbf{x}_i\}_{i=0}^{N-1}$.

This results in a triplet of coefficient vectors $\{\mathbf{k}_i\}_{i=1}^{K} \in \mathbb{C}^3$ where $\mathbb{C}$ is the complex domain and $K = \lfloor N/2 \rfloor + 1$. We have found that applying a IDFT to the first three $\mathbf{k}_{1,2,3}$ yields displacements $\{\mathbf{x}_i^*\}_{i=0}^{N-1}$ that form an excellent centerline for the original helix.

By construction, this process yields a centerline rooted at the origin. We compute a suitable scalp root translation $\mathbf{t}^*$ using an approach similar to shape matching [Müller et al. 2005]. If $\mathbf{p}_i^* = \sum_{j=0}^{i-1} \mathbf{x}_j^*$ denotes the reconstructed centerline points at the origin, a simple quadratic energy yields a crisp, closed-form solution:

$$\mathbf{t}^* = \underset{\mathbf{y}}{\operatorname{argmin}} \sum_{i=0}^{N} \left\|\mathbf{p}_i^* + \mathbf{y} - \mathbf{p}_i\right\|^2 = \frac{1}{N+1} \sum_{i=0}^{N} (\mathbf{p}_i - \mathbf{p}_i^*). \qquad (1)$$

The final centerline $\mathbf{c}^c(t)$ is then the piecewise spline from §3.1.1 constructed through the points $\{\mathbf{p}_i^c\}_{i=0}^{N} = \{\mathbf{p}_i^* + \mathbf{t}^*\}_{i=0}^{N}$.

3.2.2 High-Frequency Feature Extraction. We can now extract the centerline for *any* helical curve, and will use this approach to extract high-frequency details from the strands of a full-resolution simulation. In §3.3, we will transfer these extracted details onto the smooth centerlines of interpolated strands.

Starting from curve points $\{\mathbf{p}_i\}_{i=0}^{N}$ from a full-resolution simulation, we compute its centerline $\{\mathbf{p}_i^c\}_{i=0}^{N}$, and radial offsets from that centerline, $\mathcal{R} = \{\mathbf{r}_i\}_{i=0}^{N} = \{\mathbf{p}_i - \mathbf{p}_i^c\}_{i=0}^{N}$. Then we project each $\mathbf{r}_i$ against the orthogonal basis $\{\mathbf{u}_i, \mathbf{v}_i, (\partial \mathbf{c}/\partial t)_i\}$ from §3.1.2, to obtain $\mathbf{r}_i = \alpha_i \mathbf{u}_i + \beta_i \mathbf{v}_i + \gamma_i (\partial \mathbf{c}/\partial t)_i$. Helices without switchbacks (§4) only need to project against $\mathbf{u}_i$ and $\mathbf{v}_i$. Taking the DFTs of the α_i and β_i sequences yields another coefficient vector $\{\mathbf{w}_i\}_{i=1}^{L} \in \mathbb{C}^2$ with $L = \lfloor N+1/2 \rfloor + 1$. Taking the norm and arg of $\mathbf{w}_i$ yields amplitude and angle spectra, respectively $\mathcal{A} = \{\mathbf{a}_i\}_{i=1}^{L}$ and $\mathcal{T} = \{\boldsymbol{\theta}_i\}_{i=1}^{L}$.

These $\mathcal{A}$ and $\mathcal{T}$ vectors are now centerline-agnostic descriptions of a strand's high-frequency features. These features can now be transferred, e.g. in the spirit of Sorkine et al. [2004], to any new spline by applying them to that curve's radial offsets $\mathcal{R}$.

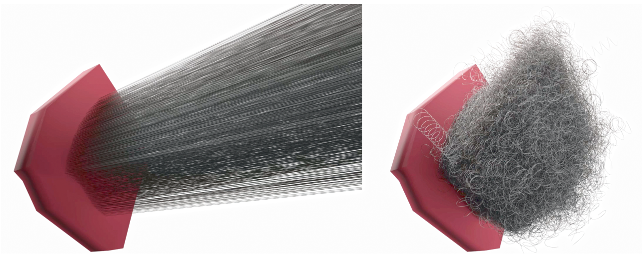


Fig. 4. Simulating 2592 strands at human-scale hair density using Shi et al. [2023]'s simulator. Initially set in a perfectly straight "comb" (left), they settle into a "spongy" texture (right) after simulating for 31 hours. Simulating 200K hairs would take ~100 days. Instead, we use the spectra from this simulation to synthesize a "spongy" layer.

3.2.3 Dataset Construction. Direct simulation of 200K strands of hair on a head remains daunting, especially for the ~80 vertices needed per highly coiled hair, in lieu of the 30 vertices needed in straight-hair examples [Daviet 2020; Kaufman et al. 2014]. Collision processing also presents new challenges.

We simulated a small patch of highly coiled hair at human-scale density, allowing the dynamics to settle and intricate entanglements characteristic of the "spongy" layer to form (Fig. 4). Direct simulation of a full head would have taken ~100 days, so we instead extract $\mathcal{A}$ and $\mathcal{T}$ spectra from each hair in this sequence. Next, we will transfer these details onto interpolated strands.

3.3 Strand Interpolation and Synthesis

3.3.1 Assigning Roots to Guides. Given a sparse set of guides and a dense set of hair root positions, we must first assign each root to a guide. The simplest strategy is nearest-neighbor; assigning based on the Voronoi diagram of the guides. However, this creates hard boundaries along the scalp, and the underlying Voronoi cells become visible. While this may be desirable for precision hairstyles like box braids, fuzzier boundaries are expected in general. We instead used a "noisy" Voronoi assignment by picking randomly from the N-closest guides to each root point (see supplemental video).

3.3.2 Single-Strand Interpolation. We now focus on a single strand, starting with its guide strand $\mathcal{V}^{\mathrm{G}} = \{\mathbf{p}_0, \mathbf{p}_1, ..., \mathbf{p}_N\}$, and its extracted centerline $\mathbf{c}^{\mathrm{G}}(t)$. The root position $\mathbf{q}_0$ of the new hair is already known from §3.3.1, so our goal is to compute the centerline $\mathbf{c}^{\mathrm{I}}$ of a new, interpolated strand, and then generate the new hair spline $\mathcal{V}^{\mathrm{I}} = \{\mathbf{q}_0, \mathbf{q}_1, ..., \mathbf{q}_N\}$.

The interpolated strand consists of two parts:

- A loosely-guided region corresponding to the "spongy" layer that is only loosely influenced by the guide strand.
- A strictly-guided region, where the strand has phase-locked coherently into the overall curl.

We set the boundary between the loosely- and strictly-guided region at $l \in [0, 1]$ (for *loose*). The full strand is then the concatenation of the two sequences $\mathcal{V}^{\mathrm{I}} = \{\mathcal{V}^{\mathrm{L}}\mathcal{V}^{\mathrm{S}}\}$. We describe the construction of $\mathcal{V}^{\mathrm{L}}$ and $\mathcal{V}^{\mathrm{S}}$ separately.

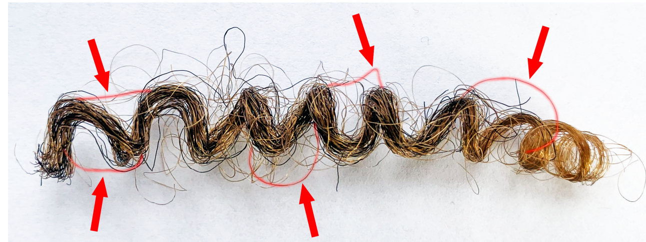


Fig. 5. Period skipping in a hair sample photo. Red arrows point to highlighted hairs that temporarily drop out of phase, but later rejoin the helix, essentially stretching out one of their periods.

3.3.3 Loosely Guided Region. For $t \in [0, l]$, the strand has not yet coalesced into a path following the guide strand. However, it still has an intrinsic curvature and helical frequency influenced by its guide. We use the centerline of the guide strand, $\mathbf{c}^{\mathrm{G}}(t)$, to set the boundary conditions for the centerline $\mathbf{c}^{\mathrm{I}}(t)$ of the interpolated strand:

$$\mathbf{c}^{\mathrm{I}}(0) = \mathbf{q}_0 \qquad \frac{\partial \mathbf{c}^{\mathrm{I}}(0)}{\partial t} = \frac{\left(\mathbf{c}^{\mathrm{G}}(l/2) - \mathbf{q}_0\right)\left\|\mathbf{c}^{\mathrm{G}}(l) - \mathbf{q}_0\right\|}{\left\|\mathbf{c}^{\mathrm{G}}(l/2) - \mathbf{q}_0\right\|}$$

$$\mathbf{c}^{\mathrm{I}}(l) = \mathbf{c}^{\mathrm{G}}(l) \qquad \frac{\partial \mathbf{c}^{\mathrm{I}}(l)}{\partial t} = \frac{\partial \mathbf{c}^{\mathrm{G}}(l)/\partial t \left\|\mathbf{c}^{\mathrm{G}}(l) - \mathbf{q}_0\right\|}{\left\|\partial \mathbf{c}^{\mathrm{G}}(l)/\partial t\right\|}$$

The derivative boundary conditions ease the interpolated spline into the guide's centerline, while the $\left\|\mathbf{c}^{\mathrm{G}}(l) - \mathbf{q}_0\right\|$ term ensures that derivative magnitudes scale with the geometry. (Fig. 3, middle)

Letting $k = \lfloor l \times (N+1) \rfloor$ and sampling k points equidistantly along $\mathbf{c}_{\mathrm{I}}$ gives a sequence of centerline points $\{\mathbf{c}_i^{\mathrm{I}}\}_{i=0}^{k-1}$. We can then transfer radial offsets according to a randomly selected $\mathcal{A}$ and $\mathcal{T}$ from the dataset of fully-simulated strands from §3.2.2. The sequence of points $\mathbf{q}_i = \{\mathbf{c}_i^{\mathrm{I}} + \mathbf{r}_i\}_{i=0}^{k-1}$ then define $\mathcal{V}_{\mathrm{L}}$.

3.3.4 Strictly Guided Region. For $t \in [l, 1]$, every strand follows along with the guide strand's shape, similar to Narayan [2023]. Beginning with $\{\mathbf{p}_i\}_{i=k}^{N}$, we compute $\mathbf{u}(t)$ and $\mathbf{v}(t)$ at any $\mathbf{p}(t)$ according to §3.1.2, and randomly sample α, β from the unit square to create a transported displacement function

$$\mathbf{d}(t) = w(t)\left(\alpha \mathbf{u}(t) + \beta \mathbf{v}(t)\right) \tag{2}$$

where $w(t)$ is a wisp radius function. Sampling at the same rates as $\{\mathbf{p}_i\}_{i=k}^{N}$, we obtain $\mathcal{V}^{\mathrm{S}} = \{\mathbf{p}_i + \mathbf{x}_i\}_{i=k}^{N}$.

3.4 Period Skipping

3.4.1 Geometric Observations. The strictly guided regions of the wisp (§3.3.4) form coherent helical bundles, and even with the variation inserted via the $w(t)$ function, the wisps look shinier and more uniform than in real-life photographs.

Closer examination of real-life hair samples reveals what is causing this difference. In the strictly guided region, while hairs are largely phase-locked into a coherent helix, many occasionally *skip* some of the helical turns (Fig. 5). They drop out of phase for a period,

Fig. 6. Photo from Darke et al. [2023] of a switchback in human hair.

then later rejoin the helix. These hairs can run orthogonal to the rest of the helix, which then break up the coherency of the highlights.

3.4.2 Algorithmic Treatment. This visual phenomenon is difficult to achieve with *scraggle* [Butts et al. 2018; Haapaoja and Genzwürker 2019; Narayan 2023] because it must alter low-frequency components in a strand. Scraggle usually excels at introducing high-frequency variations. (Fig. 10, middle)

In contrast, period skipping is straightforward with our model. If we let each strand's guide curve be a *subset* of the the guide points $\overline{\mathcal{V}^{\mathrm{G}}} \subset \mathcal{V}^{\mathrm{G}}$, generated by discarding points $\mathbf{p}_i$ according to probability ρ, then the piecewise spline will automatically generate the missing features. Periods are skipped, but the boundary conditions derived from the remaining $\mathbf{p}_i$ maintain the hair's smoothness. Despite its apparent simplicity, we found this approach has a *dramatic* impact on the overall look of the hair. Altering ρ produces a range of appearances that were previously difficult to achieve. (Fig. 12)

4 Generating Switchbacks

The geometry of *switchbacks* can be found throughout nature, from plants [Wang et al. 2013] to polymers [Wie et al. 2015] to hair (Fig. 6), and appear whenever a change of handedness occurs in a helix. They are distinct from the skipped periods in §3.4, where one errant period in a helix spans multiple periods in a coalesced curl.

Despite its apparent ubiquity, we have not found any method for computing this shape other than direct physical simulation [Bergou et al. 2008; Crespel et al. 2024], i.e. stretch out a simulated helix until it deforms into a switchback. While the geometry appears simple, extensive mechanical studies [McMillen and Goriely 2002; Wang et al. 2020] have shown that it has no compact, closed form expression. Thus, we present here the first non-linear optimization capable of directly computing the shape of a switchback.

4.1 Elastic Rod Preliminaries

4.1.1 Arc Lengths and Directors. We use s to denote arc length along a rod. This is distinct from the $t \in [0, 1]$ parameterization from §3.1.1, particularly because we will be examining boundaries as $s \longrightarrow \pm\infty$. We adopt the Lagrange notation of $(\cdot)' = \partial(\cdot)/\partial s$, as derivatives in terms of arc length will arise often. The shape of the curve is then $\mathbf{c}(s) : \mathbb{R} \rightarrow \mathbb{R}^3$. The curve has an orthonormal director basis $\{\mathbf{d}_1(s), \mathbf{d}_2(s), \mathbf{d}_3(s)\}$, where $\mathbf{d}_3(s)$ is the tangent vector (Fig. 7)

$$\mathbf{d}_3(s) = \frac{\partial \mathbf{c}(s)}{\partial s} \tag{3}$$

For any orthonormal basis of the curve, there is a twist vector $\boldsymbol{\kappa}(s) = \kappa_1(s)\mathbf{d}_1 + \kappa_2(s)\mathbf{d}_2 + \kappa_3(s)\mathbf{d}_3$ such that

$$\mathbf{d}_i'(s) = \boldsymbol{\kappa}(s) \times \mathbf{d}_i(s). \tag{4}$$

The κ_1 and κ_2 terms are often referred to as curvature and κ_3 as torsion. $\mathbf{d}_1(s)$ and $\mathbf{d}_2(s)$ lie in the cross-sectional plane of the rod. This basis can be defined by the material axes and is distinct from the Frenet frame of the curve because of torsion.

4.1.2 Dimensionless Form. With the directors defined, we can introduce mechanical properties. We have found that it is more compact and structure-revealing to write things in dimensionless form. Let $\tilde{\cdot}$ denote the original dimensional forms. We scale by the rest curvature k^0 of the helix to obtain the dimensionless twist vector $\boldsymbol{\kappa} = \tilde{\boldsymbol{\kappa}}/k^0$, arc length $s = \tilde{s}k^0$, and overall curve $\mathbf{c} = \tilde{\mathbf{c}}k^0$. Forces $\mathbf{f} = \tilde{\mathbf{f}}/EI_1(k^0)^2$ and moments $\mathbf{m} = \tilde{\mathbf{m}}/EI_1 k^0$ are also scaled, where E is Young's modulus and I_1 is the scaled principal moment of inertia. This dimensionless form reveals that our analysis can proceed independent of Young's modulus (E) and rest state coil radius (k^0).

4.1.3 Static Analysis. A static rod can be described using the Kirchhoff equations:

$$\mathbf{f}' = 0 \qquad\qquad \mathbf{m}' + \mathbf{d}_3 \times \mathbf{f} = 0. \tag{5}$$

We can then use linear elasticity to compute forces:

$$\mathbf{m} = \left(\kappa_1 - \kappa_1^0\right)\mathbf{d}_1 + \Lambda\left(\kappa_2 - \kappa_2^0\right)\mathbf{d}_2 + \Gamma\left(\kappa_3 - \kappa_3^0\right)\mathbf{d}_3 \tag{6}$$

where κ_i^0 are the rest state twist vector components, and $\Lambda = I_2/I_1$ is the ratio of the moments of inertia along $\mathbf{d}_2$ and $\mathbf{d}_1$. We will assume $\Lambda = 1$ going forward, because while elliptical cross-sections can induce helicity in plants [Farhan et al. 2023], curliness in human hair is caused by inhomogeneous fiber distributions [Wortmann et al. 2020]. Using Poisson's ratio μ, we set $\Gamma = 1/(1+\mu)$. Forces can now be related to $\boldsymbol{\kappa}(s)$, which can then be used to compute a curve $\mathbf{c}(s)$ by integrating Eqns. 3 and 4.

4.2 Helix Solution

4.2.1 Energy Formulation. We can further tailor these equations to solve for a helix. Once that solution is known, we will modify it to accommodate a switchback. For a helix, the rest state twist vector is

$$\boldsymbol{\kappa}^0(s) = \mathbf{d}_1 + \kappa_3^0\mathbf{d}_3, \tag{7}$$

and the equation for a curve running along $\hat{\mathbf{x}} = [1\ 0\ 0]^\top$ is

$$\mathbf{c}(s) = \frac{1}{\lambda^2}\cos(\lambda s)\hat{\mathbf{y}} + \frac{1}{\lambda^2}\sin(\lambda s)\hat{\mathbf{z}} + \frac{\kappa_3^0}{\lambda}s \tag{8}$$

where $\lambda = \sqrt{1 + \left(\kappa_3^0\right)^2}$, $\hat{\mathbf{y}} = [0\ 1\ 0]^\top$, and $\hat{\mathbf{z}} = [0\ 0\ 1]^\top$. In this case, $\kappa_2^0 = 0$ and κ_1^0 and κ_3^0 are constants with respect to s. Furthermore, $\kappa_1^0 = 1$ since the unscaled $\tilde{\kappa}_1^0$ is equal to the rest state curvature k_0 .

By plugging Eqns. 6 and 4 into the static Kirchhoff equations, we obtain static equilibrium conditions in terms of twist and forces:

$$f_1' = f_2\kappa_3 - f_3\kappa_2 \qquad \kappa_1' = f_2 - (\Gamma - 1)\kappa_2\kappa_3 + \Gamma\kappa_3^0\kappa_2 \tag{9}$$

$$f_2' = f_3\kappa_1 - f_1\kappa_3 \qquad \kappa_2' = -f_1 + (\Gamma - 1)\kappa_2\kappa_3 + \kappa_3 - \Gamma\kappa_3^0\kappa_1 \tag{10}$$

$$f_3' = f_1\kappa_2 - f_2\kappa_1 \qquad \kappa_3' = -\kappa_2/\Gamma \tag{11}$$

We still cannot solve for $\boldsymbol{\kappa}(s)$ because the above system is underconstrained. Building from Eqn. 23 in McMillen and Goriely [2002], we observe that a stable shape would minimize the following energy density function

$$\mathcal{E}\left(\boldsymbol{\kappa}\right)=\frac{1}{2}(\kappa_1-1)^2+\frac{1}{2}\kappa_2^2+\frac{\Gamma}{2}\left(\kappa_3-\kappa_3^0\right)^2-\|\mathbf{f}\|\nu \tag{12}$$

which sums the elastic and potential energy of an external stretching force $\mathbf{f}$ running down the centerline of the helix. Here, $\nu=\kappa_3/\sqrt{\kappa_1^2+\kappa_3^2}$ denotes the coil frequency. Solving for the shape of a helix deforming under $\mathbf{f}$ can be viewed as minimizing

$$\boldsymbol{\kappa}(s)=\underset{\boldsymbol{\kappa}}{\operatorname{argmin}}\int_s \mathcal{E}\left(\boldsymbol{\kappa}\right)\,ds, \tag{13}$$

subject to the constraints in Eqns. 9-11.

4.2.2 Force Substitution. For an infinitely long helix, the twist vector about $s\longrightarrow-\infty$ should be

$$\boldsymbol{\kappa}^h=\kappa_1^h\mathbf{d}_1+\kappa_2^h\mathbf{d}_2+\kappa_3^h\mathbf{d}_3, \tag{14}$$

where we constrain $\kappa_2^h=0$, and use the h superscript to denote the helix solution we seek. Plugging the above into Eqns. 9-11 yields

$$\mathbf{f}^h=\left(\frac{1}{\kappa_1^h}-1+\Gamma\left(1-\frac{\kappa_3^0}{\kappa_3^h}\right)\right)\kappa_3^h\boldsymbol{\kappa}^h. \tag{15}$$

By Eqn. 5, this force is a constant along the entire curve, and should be equal to the external force at both ends. We can eliminate $\mathbf{f}$ from Eqn. 12 by substituting in Eqns. 14 and 15:

$$\mathcal{E}\left(\boldsymbol{\kappa}^h\right)=\frac{1}{2}\left(\kappa_1^h-1\right)^2+\frac{\Gamma}{2}\left(\kappa_3^h-\kappa_3^0\right)^2-\left(\frac{1}{\kappa^h}-1+\Gamma\left(1-\frac{\kappa_3^0}{\kappa_3^h}\right)\right)\left(\kappa_3^h\right)^2.$$

This energy density is constant with respect to s, and parameterized entirely by κ_1^h and κ_3^h.

4.2.3 Final Helix Expressions. For the final solution, we consider $\nu\in[-1,1]$. If we can find the $\boldsymbol{\kappa}^h$ that corresponds to the energy gradient vanishing as $\frac{\partial\mathcal{E}}{\partial\kappa_1^h}=\frac{\partial\mathcal{E}}{\partial\kappa_3^h}=0$ at a fixed external force $\mathbf{f}^h$, then we have the solution. We can solve for this state analytically:

$$\kappa_1^h=\frac{1-\nu^2+\Gamma\nu\sqrt{1-\nu^2}\kappa_3^0}{1+(\Gamma-1)\nu^2}\qquad\qquad \kappa_2^h=0 \tag{16}$$

$$\kappa_3^h=\frac{\kappa_3^0}{2}+\mathbf{sgn}(\nu)\sqrt{\frac{\left(\kappa_3^0\right)^2}{4}+\frac{\kappa_1^h}{\Gamma}(1-\kappa_1^h)}. \tag{17}$$

This forms a family of ellipses in the $\left(\kappa_1^h,\kappa_3^h\right)$ plane. Combining the above equation with $\nu=\kappa_3/\sqrt{\kappa_1^2+\kappa_3^2}$, the twist vector at equilibrium can be completely determined by ν. These helix solutions will now serve as boundary conditions when we insert switchbacks.

4.3 Switchback Insertion

4.3.1 Helix Perturbation. With the helical solution $\boldsymbol{\kappa}^h$ established, we now perturb it compute a switchback solution $\boldsymbol{\kappa}^\Omega$. We use Ω because it looks like a switchback.

If we insert a switchback centered at $s=0$, we can assume that as $s\longrightarrow\pm\infty$, the shape reverts to a helix. Previous work [McMillen and Goriely 2002] showed that switchbacks decay exponentially quickly

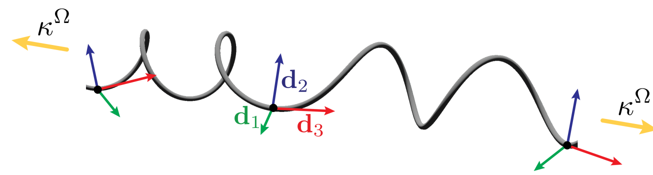


Fig. 7. The curve director basis along the curve and the twist boundaries.

in space. Thus, we can apply helical boundary conditions to κ_1 and κ_3 at the ends of a coil containing a switchback, even if the total arc length is finite. However, in the case of a switchback curve, the handedness of the boundaries flip (Fig. 7):

$$\boldsymbol{\kappa}^\Omega\longrightarrow(\kappa_1^h\mathbf{d}_1+\kappa_3^h\mathbf{d}_3)\qquad\text{when}\qquad s\longrightarrow\infty \tag{18}$$

$$\boldsymbol{\kappa}^\Omega\longrightarrow(\kappa_1^h\mathbf{d}_1-\kappa_3^h\mathbf{d}_3)\qquad\text{when}\qquad s\longrightarrow-\infty. \tag{19}$$

While this narrows down the possibilities for $\boldsymbol{\kappa}(s)$, solving the variational problem without a specific family of functions remains challenging. Consequently, we adopt the Rayleigh-Ritz approximation from Wang et al. [2020], which propose that κ_1 and κ_2 can be approximated by:

$$\kappa_1^\Omega=a_1e^{-b_1s^2}\cos(c_1s)+a_2e^{-b_2s^2}\cos(c_2s)+\kappa_1^h \tag{20}$$

$$\kappa_2^\Omega=a_3e^{-b_3s^2}\cos(c_3s)+a_4e^{-b_4s^2}\cos(c_4s)+\kappa_2^h. \tag{21}$$

The original helix solutions are highlighted as the red κ_1^h and κ_2^h. The remaining terms are exponentially decaying cosine "bumps" that perturb the original helix. The parameters $\{a_i,b_i,c_i\}$ are what we must now optimize for. The remaining κ_3^Ω can then be solved for by integrating Eqn. 11 and applying two boundary conditions:

$$\kappa_3^\Omega=-\frac{1}{\Gamma}\int_s\kappa_2^\Omega\,ds\qquad\lim_{s\to-\infty}\kappa_3^\Omega=\kappa_3^h\qquad\lim_{s\to\infty}\kappa_3^\Omega=-\kappa_3^h \tag{22}$$

The solution has the analytic form:

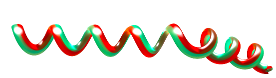
$$\kappa_3^\Omega=-\alpha\operatorname{Re}\left(\operatorname{erf}\left(\sqrt{b_3s}+\frac{c_3}{2\sqrt{b_3s}}i\right)\right)-(\kappa_3^h-\alpha)\operatorname{Re}\left(\operatorname{erf}\left(\sqrt{b_4s}+\frac{c_4}{2\sqrt{b_4s}}i\right)\right)$$

where $\alpha=\frac{a_3}{2\Gamma}\sqrt{\frac{\pi}{b_3}}e^{\frac{-c_3^2}{4b_3}}$, $\operatorname{Re}(\cdot)$ extracts the real component, and $\operatorname{erf}(\cdot)$ is the Gauss error function. Again, the helix solution κ_3^h gets perturbed by the other terms. There are now 12 free parameters, $\{a_i,b_i,c_i\}$, but one can be set using the boundary condition:

$$a_4=\sqrt{b_4}e^{\frac{c_4^2}{4b_4}}\left(\frac{2\Gamma\kappa_3^h}{\sqrt{\pi}}-\frac{a_3}{\sqrt{b_3}}e^{-\frac{c_3^2}{4b_3}}\right) \tag{23}$$

This leaves 11 parameters to optimize. We write them as a vector $\mathbf{n}=[a_1,b_1,c_1,a_2,b_2,c_2,a_3,b_3,c_3,b_4,c_4]^\top$, and minimize them over the energy density from Eqn. 12:

$$\mathbf{n}=\underset{\mathbf{n}}{\operatorname{argmin}}\int_s\mathcal{E}(\boldsymbol{\kappa}^\Omega)\,ds \tag{24}$$



4.3.2 Symmetry Regularization. Until now, we have focused on switchback insertion. However, Eqn. 24 lacks a key feature: left-right symmetry. Without it, we have found that half of the helix shoots off at a random and

Table 1. Performance of our Curly-Cue geometric methods. Our **unoptimized, single-threaded Python** implementation is on par with the running times of the simulation [Shi et al. 2023] and rendering in Blender. All timings are per frame, and in **minutes:seconds**, unless otherwise indicated.

Fig.	Sim. Strands	Render Hairs	Sim. Time	Curly-Cue Time	Render Time	Sim. Verts	Curly-Cue Verts
1, 8	8,546	233,933	03:10	44:57	28:05	552,271	13,118,079
10, 12	98	3,176	00:05	00:55	14:50	10,499	309,296
13	1	161	0.05s	0.2s	01:44	103	15,024
14	6,525	165,077	00:23	05:01	09:52	44,425	914,492
9	2,038	119,840	03:19	55:10	32:50	546,130	17,658,566

uncontrollable angle (see left inset), which is inconsistent with the helix boundaries.

Thus, we add a regularizer that encourages the overall curve to remain straight by requiring that the tangent at $s = 0$ run parallel to the centerline. Since we assumed in Eqn. 8 that the curve's centerline runs along $\hat{\mathbf{x}} = [1\ 0\ 0]^\top$, the symmetry condition is:

$$1 - (\mathbf{d}_3(0) \cdot \hat{\mathbf{x}})^2 = 0. \tag{25}$$

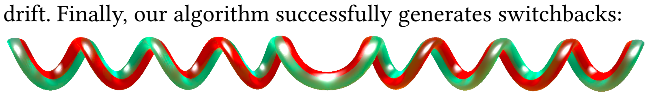
Adding this to the optimization yields our final energy:

$$\mathbf{n} = \underset{\mathbf{n}}{\operatorname{argmin}} \left(\left(1 - (\mathbf{d}_3(0) \cdot \hat{\mathbf{x}})^2\right) + \int_s \mathcal{E}(\boldsymbol{\kappa}^\Omega)\, ds \right) \tag{26}$$

Computing $\mathbf{d}_3(0)$ will be described in §4.3.3.

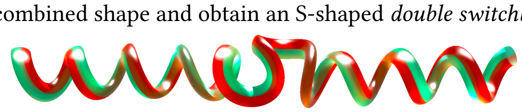
4.3.3 Curve Generation. After obtaining $\boldsymbol{\kappa}^\Omega(s)$ via non-linear optimization (§5.1.4), we can use it to integrate Eqn. 4, obtain a director basis $\{\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3\}$, and integrate Eqn. 3 to obtain the final curve $\mathbf{c}(s)$.

Integrating Eqn. 3 using forward Euler is straightforward, but Eqn. 4 involves rotating an orthonormal basis. We have found that brute-force basis renormalization introduces visible numerical drift, so in the spirit of Lie group integrators [Angelidis 2017], we build a rotation matrix $\mathbf{R}(s)$ for each Euler step. The directors $\mathbf{d}_i$ rotate about the axis $\boldsymbol{\kappa}^\Omega$, so for a step size Δs, the angle is $\Delta\phi = \|\boldsymbol{\kappa}\|\Delta s$. After constructing $\mathbf{R}(s)$, we integrate the directors along s according to $\mathbf{d}_i(s + \Delta s) = \mathbf{R}(s)\, \mathbf{d}_i(s)$. This ensures that the directors remain orthonormal, and we have found it dramatically reduces numerical drift. Finally, our algorithm successfully generates switchbacks:



4.3.4 Rest State Symmetry Analysis. To simplify their analyses, previous mechanics works have limited their discussions to circular rest states of $\boldsymbol{\kappa}^0 = (1, 0, 0)$. Our derivation generalizes to helical rest states with any number of switchbacks. However, this can imply that a helical solution with a single change-of-handedness is no longer a suitable boundary condition. Instead, an even number of switchbacks is required along a helix. In this case, the boundary conditions should be $\boldsymbol{\kappa}^\Omega \longrightarrow \left(\kappa_1^h \mathbf{d}_1 + \kappa_3^h \mathbf{d}_3\right)$ as $s \longrightarrow \pm\infty$.

If multiple switchbacks are sufficiently separated, their perturbation terms (Eqns. 20 and 21) can be linearly summed and translated. When two switchbacks are close to each other, we can optimize for their combined shape and obtain an S-shaped *double switchback*:



We provide source code in our supplemental materials.

5 Implementation and Results

5.1 Implementation Details

5.1.1 Frame Generation. In §3.1.1 every point on $\mathbf{c}(t)$ needed a frame $\{\mathbf{u}, \mathbf{v}\}$ orthonormal to $\partial \mathbf{c}(t)/\partial t$. While many methods are available [Duff et al. 2017; Max 2017], we found that following sufficed:

$$\mathbf{u}(t) = \partial \mathbf{c}(t)/\partial t \times \hat{\mathbf{x}} \qquad\qquad \mathbf{v}(t) = \partial \mathbf{c}(t)/\partial t \times \mathbf{u}(t). \tag{27}$$

If $\partial \mathbf{c}(t)/\partial t^\top \hat{\mathbf{x}} = 1$, then $\hat{\mathbf{x}}$ is randomly sampled from the unit sphere.

5.1.2 Strand Variation. To vary hair lengths within each wisp, we compressed the range over which points are sampled along each strand curve, to $[0, 1 - \epsilon]$, where ϵ is chosen randomly.

5.1.3 Fades. For regions with hairs shorter than a single curl, coalescing behaviour is not expected, so individual strands duplicate their guide strand (Fig. 11). By transitioning between this and the method of §3 for longer hair, we can achieve a range of looks.

5.1.4 Non-Linear Optimization. Computing switchbacks involves optimizing Eqn. 26, and we found derivative-free methods such as Nelder-Mead [Nocedal and Wright 2006] sufficed, so deriving gradient terms is unnecessary. To improve convergence, we replaced the integral in Eqn. 26 with a sparsely sampled ∞-norm in the region around the switchback. Each optimization took ~10s, or ~1.5s when warm-started with a solution from a similar ν.

5.2 Results

The spongy layer created by our phase locking algorithm can be seen on the right of Fig. 10. Traditional wisp [Watanabe and Suenaga 1992] and scraggle [Butts et al. 2018] methods (Fig. 10, left, middle) maintain the curls up to the scalp, and create a wig-like look.

By changing the skipping probability ρ in §3.4, we can achieve a range of different looks (Fig. 12), from shiny with highly visible coils, to diffuse with brushed apart coils. In the supplemental video, we show that these features stay coherent under animation.

Our switchback method is flexible enough that it can insert the shape anywhere along an existing wisp (Fig. 13, middle), and even multiple times on a single wisp (Fig. 13, right). They are inserted prior to simulation so they can deform under external forces. We are able to generate looks that are distinct from previous approaches [Daviet 2023; Hsu et al. 2023; Shen et al. 2023; Zhou et al. 2023]. While these papers presented methods that tried to encompass multiple hair types, we focused on a single, under-investigated hair type to achieve its characteristic look. Our algorithm generalizes to human-scale, as seen on our models of *New York Times* bestselling author Carvell Wallace [Iguodala and Wallace 2019] (Figs. 1, 8).

Fig. 8. Close-ups and additional angles on the Carvell Wallace model from Fig. 1. The simulation consisted of 8K strands that were interpolated using our methods into 233K hairs for rendering. Period skipping was set to $\rho = 0.5$ and switchbacks were inserted prior to simulation (bottom row, middle).

We inserted switchbacks into the hair, and one can be seen on the bottom row of Fig. 8. Alternate renders with different ρ are shown in the supplemental video. Figs. 14 and 9 showcase additional styles.

We verified our switchback geometry using Discrete Elastic Rods [Bergou et al. 2008] by running an example that verifies a helix is also a stable state. The simulation is in the supplemental video. As seen in Table 1, the timings for our Curly-Cue methods are on par with the simulation and rendering. Our current implementation is **unoptimized, single-threaded Python**, so we'd expect a high-performance, parallel implementation to run ~10× faster.

5.3 Limitations

Phase locking is currently applied as a geometric post-process, and does not play an active role in the simulation. A more realistic approach would directly simulate the spongy layer near the scalp.

The effects the geometric phenomena we have modelled decrease as hair becomes straighter. For example, period skipping loses meaning for straight hair that does not turn through a single helical period. Finding the precise frequencies at which these phenomena become visually irrelevant will need further empirical investigation. The switchback optimization can still take a few seconds to compute the shape, especially when solving for small coil frequencies from a cold start. The current method also cannot directly solve for shapes under complex external forces. We instead approximate them by integrating the switchback into the hair simulator.

6 Conclusions and Future Work

We have presented algorithms for three geometric phenomena characteristic of highly coiled hair: phase locking, period skipping, and switchbacks. Our switchback formulation applies to other scenarios where they occur, such as plant synthesis [Niese et al. 2022]. Our methods are for relatively unstyled hair, but a wide range of highly coiled styles exist, such as box braids, twist-outs and natural locs [Darke et al. 2023]. Investigating whether our current methods generalize to these styles, or if they involve new geometric phenomena that require new methods, remains future work. Finally, efficiently simulating highly coiled hairs, with their larger vertex counts and unique collision behaviors, remains a challenge.

Acknowledgments

The authors thank Jet Appling, H.D. Harris, Clíona Murray, Elijah Richmond, and Yar Sudani for their help. This work was supported by the Bungie Foundation, Teng and Han Family Fund, 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. 9. Close-ups and additional angles on the Carvell Wallace model with longer hair.

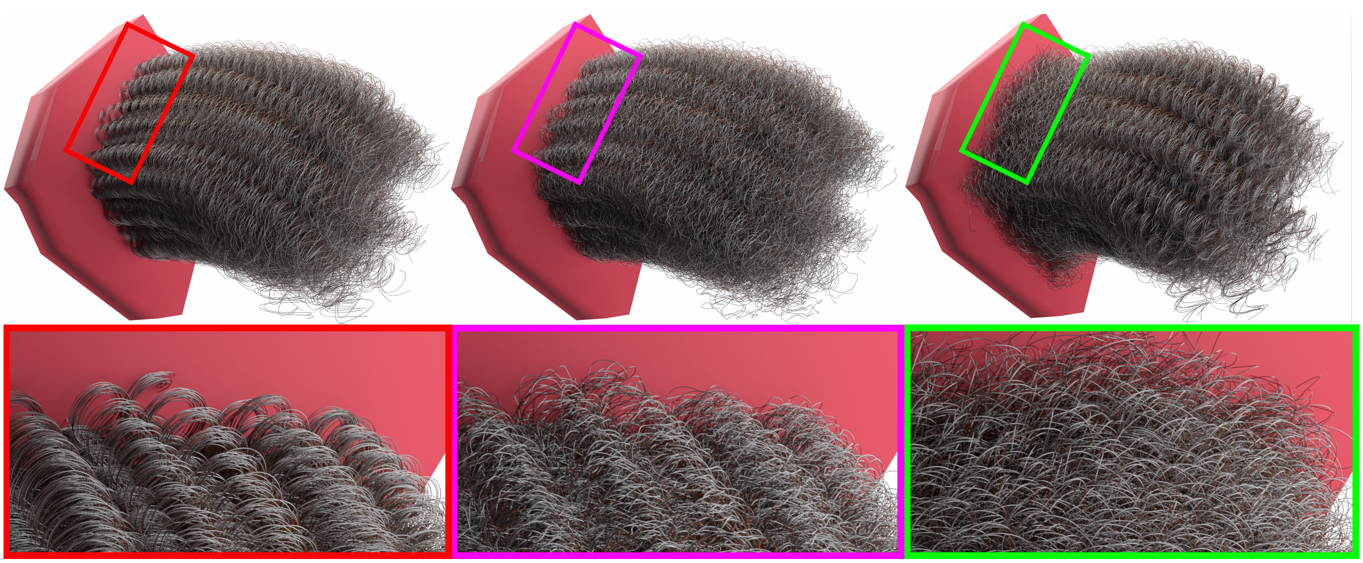


Fig. 10. Wisps generated with Watanabe and Suenaga [1992] are coherent at the scalp (red box), creating the appearance of a wig. Adding *scraggle* (magenta box) wrinkles the hair, but the wig look remains. Our **phase locking** method (green box), creates a spongy layer that transitions into coils.

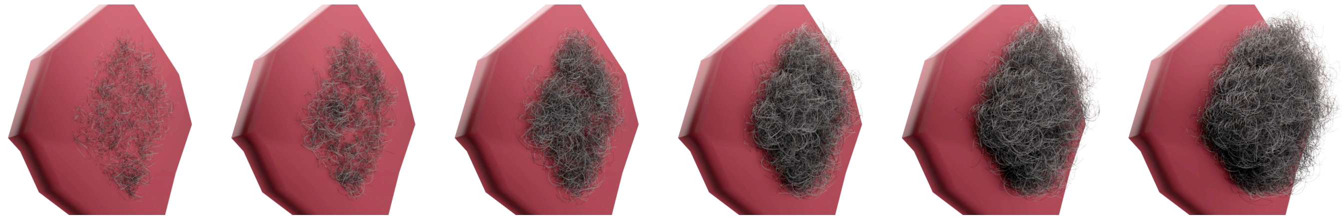


Fig. 11. Our approach handles a range of hair lengths, from close cut fades (left) to longer coils (right).

Fig. 12. Left to right: **Period skipping** setting $\rho = 0.0, 0.25, 0.50$, and an extreme 0.75. The appearance of each wisp changes, but so does the overall look, even though the rendering material parameters are the same. As more periods are skipped, the sharp highlights on the coils become more diffuse.

Fig. 13. Left to right: single wisp with no **switchbacks**, a switchback inserted near the wisp tip, one inserted near the middle. We can insert multiple switchbacks on one wisp, so on the right shows insertions near the middle and tip.

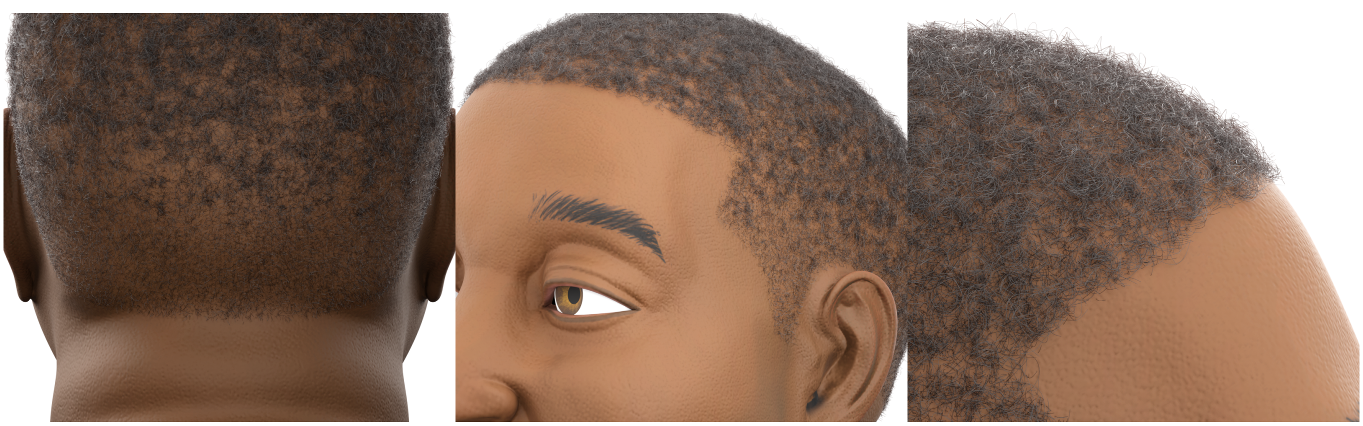


Fig. 14. Close-ups and additional angles on the Carvell Wallace model with shorter hair.

References

Alexis Angelidis. 2017. Multi-scale vorticle fluids. *ACM Trans. Graph.* 36, 4, Article 104 (jul 2017), 12 pages.

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.

Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. 2008. Discrete Elastic Rods. *ACM Trans. Graph.* 27, 3 (aug 2008), 63:1–63:12.

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.* Eurographics.

F. Bertails, T-Y. Kim, M-P. Cani, and U. Neumann. 2003. Adaptive Wisp Tree: a multiresolution control structure for simulating dynamic clustering in hair motion. In *Proceedings of the ACM SIGGRAPH/Eurographics Symp. Comp. Anim.* 207–213.

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 Talks.* 1–2.

Byoungwon Choe, Min Gyu Choi, and Hyeong-Seok Ko. 2005. Simulating complex hair with robust collision handling. In *Proceedings of the ACM SIGGRAPH/Eurographics Symp. Comp. Anim.* 153–160.

Octave Crespel, Emile Hohnadel, Thibaut Métivet, and Florence Bertails-Descoubes. 2024. Contact detection between curved fibres: high order makes a difference. In *Proceedings of SIGGRAPH.*

CurlCentric. 2014. *Hair Type Chart: How to Find Your Curl Pattern with Pictures.* https://www.curlcentric.com/hair-typing-system/

A.M. Darke, H.D. Harris, Nai'Vasha Grace, Isaac Olander, Malika Sajhmori Dunn, Diandra Rose, Adésayo Adéoyé, and Chrissy Powell. 2023. *Code My Crown: A guide to the creation and celebration of Black hair in video games.* Dove.

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 Courses* (Denver, CO, USA) *(SIGGRAPH Courses)*. Article 18, 251 pages.

Gilles Daviet. 2020. Simple and scalable frictional contacts for thin nodal objects. *ACM Trans. Graph.* 39, 4 (2020), 61–1.

Gilles Daviet. 2023. Interactive Hair Simulation on the GPU using ADMM. In *ACM SIGGRAPH 2023 Conference Proceedings.* 1–11.

R. De La Mettrie, D. Saint-Léger, G. Loussouarn, A. Garcel, C. Porter, and A. Langaney. 2007. Shape variability and classification of human hair: a worldwide approach. *Human biology* 79, 3 (2007), 265–281.

Alexandre Derouet-Jourdan, Florence Bertails-Descoubes, Gilles Daviet, and Joëlle Thollot. 2013. Inverse dynamic hair modeling with frictional contact. *ACM Trans. Graph.* 32, 6 (2013), 1–10.

Tom Duff, James Burgess, Per Christensen, Christophe Hery, Andrew Kensler, Max Liani, and Ryusuke Villemin. 2017. Building an Orthonormal Basis, Revisited. *Journal of Computer Graphics Techniques (JCGT)* 6, 1 (27 March 2017), 1–8.

M. Farhan, F. Klimm, M. Thielen, A. Rešetič, A. Bastola, M. Behl, T. Speck, and A. Lendlein. 2023. Artificial tendrils mimicking plant movements by mismatching modulus and strain in core and shell. *Advanced Materials* 35, 22 (2023), 2211902.

Alain Goriely and Michael Tabor. 1998. Spontaneous helix hand reversal and tendril perversion in climbing plants. *Physical Review Letters* 80, 7 (1998), 1564.

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

J. Hancock, M. Nisbet, W. Zhang, P. Halasyamani, and K. Poeppelmeier. 2020. Periodic tendril perversion and helices in the AMoO2F3 (A= K, Rb, NH4, Tl) family. *Journal of the American Chemical Society* 142, 13 (2020), 6375–6380.

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

Jerry Hsu, Tongtong Wang, Zherong Pan, Xifeng Gao, Cem Yuksel, and Kui Wu. 2023. Sag-Free Initialization for Strand-Based Hybrid Hair Simulation. *ACM Trans. Graph.* 42, 4 (2023), 1–14.

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.

Andre Iguodala and Carvell Wallace. 2019. *The Sixth Man.* Penguin Random House.

Wenzel Jakob, Jonathan T Moon, and Steve Marschner. 2009. Capturing Hair Assemblies Fiber by Fiber. *ACM Transactions on Graphics* 28, 5 (2009), 164–1.

Danny M. Kaufman, Rasmus Tamstorf, Breannan Smith, Jean-Marie Aubry, and Eitan Grinspun. 2014. Adaptive Nonlinearity for Collisions in Complex Rod Assemblies. *ACM Trans. Graph.* (2014).

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

Theodore Kim, Holly Rushmeier, Julie Dorsey, Derek Nowrouzezahrai, Raqi Syed, Wojciech Jarosz, and A.M. Darke. 2022. Countering racial bias in computer graphics research. In *ACM SIGGRAPH Talks.* ACM, 1–2.

Tae-Yong Kim and Ulrich Neumann. 2002. Interactive multiresolution hair modeling and editing. *ACM Trans. Grap.* 21, 3 (2002), 620–629.

Vasil Kolev and Juliana Yordanova. 1997. Analysis of phase-locking is informative for studying event-related EEG activity. *Biological cybernetics* 76, 3 (1997), 229–235.

S. Liang, X. Huang, X. Meng, K. Chen, L. Shapiro, and I. Kemelmacher-Shlizerman. 2018. Video to fully automatic 3d hair model. *ACM Trans. Graph.* 37, 6 (2018), 1–14.

Mariama Mafro and Zainab Mafro. 2013. *The LOIS Hair Typing System.* http://themafrosisters.blogspot.com/2013/04/curl-class-7-lois-hair-typing-system.html

Nelson Max. 2017. Improved accuracy when building an orthonormal basis. *Journal of Computer Graphics Techniques (JCGT)* 6, 1 (27 March 2017), 9–16.

A. McAdams, A. Selle, K. Ward, E. Sifakis, and J. Teran. 2009. Detail preserving continuum simulation of straight hair. *ACM Trans. Graph.* 28, 3 (2009), 1–6.

T McMillen and A Goriely. 2002. Tendril Perversion in Intrinsically Curved Rods. *J. Nonlinear Science* 12, 3 (2002), 241–281.

Neil H Mendelson. 1978. Helical Bacillus subtilis macrofibers: morphogenesis of a bacterial multicellular macroorganism. *Proceedings of the National Academy of Sciences* 75, 5 (1978), 2478–2482.

Zahra Montazeri, Chang Xiao, Yun Fei, Changxi Zheng, and Shuang Zhao. 2019. Mechanics-aware modeling of cloth appearance. *IEEE transactions on visualization and computer graphics* 27, 1 (2019), 137–150.

M. Müller, B. Heidelberger, M. Teschner, and M. Gross. 2005. Meshless deformations based on shape matching. *ACM Trans. Graph.* 24, 3 (2005), 471–478.

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

Till Niese, Sören Pirk, Matthias Albrecht, Bedrich Benes, and Oliver Deussen. 2022. Procedural urban forestry. *ACM Transactions on Graphics (TOG)* 41, 2 (2022), 1–18.

Jorge Nocedal and Stephen J Wright. 2006. *Numerical optimization.* Springer.

Sofya Ogunseitan. 2022. Space Rangers with Cornrows: Methods for Modeling Braids and Curls in Pixar's Groom Pipeline. In *ACM SIGGRAPH Talks.* Article 49, 2 pages.

Sylvain Paris, Hector M. Briceño, and François X. Sillion. 2004. Capture of hair geometry from multiple images. *ACM Trans. Graph.* 23, 3 (aug 2004), 712–719.

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

Ken Perlin. 1985. An image synthesizer. *Proceedings of SIGGRAPH* 19, 3 (1985), 287–296.

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 *Proceedings of ECCV.* Springer, 73–89.

Andrew Selle, Michael Lentine, and Ronald Fedkiw. 2008. A mass spring model for hair simulation. *ACM Trans. Graph.* 27, 3 (aug 2008), 1–11.

Yuefan Shen, Shunsuke Saito, Ziyan Wang, Olivier Maury, Chenglei Wu, Jessica Hodgins, Youyi Zheng, and Giljoo Nam. 2023. CT2Hair: High-Fidelity 3D Hair Modeling using Computed Tomography. *ACM Trans. Graph.* 42, 4 (July 2023), 75:1–75:13.

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. https://doi.org/10.1145/3606920

Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and H-P Seidel. 2004. Laplacian surface editing. In *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry processing.* 175–184.

T. Sun, G. Nam, C. Aliaga, C. Hery, and R. Ramamoorthi. 2021. Human Hair Inverse Rendering using Multi-View Photometric data. In *Eurographics Symposium on Rendering - DL-only Track.* https://doi.org/10.2312/sr.20211301

N. Umetani, R. Schmidt, and J. Stam. 2015. Position-based elastic rods. In *Proceedings of the ACM SIGGRAPH/Eurographics Symp. Comp. Anim.* 21–30.

Andre Walker. 1997. *Andre Talks Hair.* Simon & Schuster.

D. Wang, M.D. Thouless, W. Lu, and J.R. Barber. 2020. Generation of perversions in fibers with intrinsic curvature. *Journal of the Mechanics and Physics of Solids* 139 (2020), 103932.

Jian-Shan Wang, Gang Wang, Xi-Qiao Feng, Takayuki Kitamura, Yi-Lan Kang, Shou-Wen Yu, and Qing-Hua Qin. 2013. Hierarchical chirality transfer in the growth of Towel Gourd tendrils. *Scientific reports* 3, 1 (2013), 3102.

Kelly Ward, Florence Bertails, Tae-Yong Kim, Stephen R Marschner, Marie-Paule Cani, and Ming C Lin. 2007. A survey on hair modeling: Styling, simulation, and rendering. *IEEE transactions on visualization and computer graphics* 13, 2 (2007), 213–234.

Y. Watanabe and Y. Suenaga. 1992. A trigonal prism-based method for hair image generation. *IEEE Computer Graphics and Applications* 12, 1 (1992), 47–53.

Jeong Jae Wie, Kyung Min Lee, Taylor H Ware, and Timothy J White. 2015. Twists and turns in glassy, liquid crystalline polymer networks. *Macromolecules* 48, 4 (2015), 1087–1092.

Franz J Wortmann, Gabriele Wortmann, and Therakanya Sripho. 2020. Why is hair curly?—Deductions from the structure and the biomechanics of the mature hair shaft. *Experimental dermatology* 29, 3 (2020), 366–372.

Cem Yuksel, Scott Schaefer, and John Keyser. 2009. Hair meshes. *ACM Trans. Graph.* 28, 5 (2009), 1–7.

Cem Yuksel and Sarah Tariq. 2010. Advanced techniques in real-time hair rendering and simulation. In *ACM SIGGRAPH Courses.* 1–168.

M. Zhang, M. Chai, H. Wu, H. Yang, and K. Zhou. 2017. A data-driven approach to four-view image-based hair modeling. *ACM Trans. Graph.* 36, 4 (2017), 156–1.

Y. Zhou, M. Chai, A. Pepe, M. Gross, and T. Beeler. 2023. GroomGen: A High-Quality Generative Hair Model Using Hierarchical Latent Representations. *ACM Trans. Graph.* 42, 6 (Dec. 2023), 270:1–270:16.