



Inverse Dynamic Hair Modeling with Frictional Contact

Alexandre Derouet-Jourdan, Florence Bertails-Descoubes, Gilles Daviet,
Joëlle Thollot

► To cite this version:

Alexandre Derouet-Jourdan, Florence Bertails-Descoubes, Gilles Daviet, Joëlle Thollot. Inverse Dynamic Hair Modeling with Frictional Contact. ACM Transactions on Graphics, 2013, Proceedings of SIGGRAPH Asia 2013, 32 (6), pp.Article No. 159. 10.1145/2508363.2508398 . hal-00857559

HAL Id: hal-00857559

<https://inria.hal.science/hal-00857559>

Submitted on 3 Sep 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Inverse Dynamic Hair Modeling with Frictional Contact

Alexandre Derouet-Jourdan

Florence Bertails-Descoubes

Gilles Daviet

Joëlle Thollot

INRIA and Laboratoire Jean Kuntzmann (Grenoble University, CNRS), France*

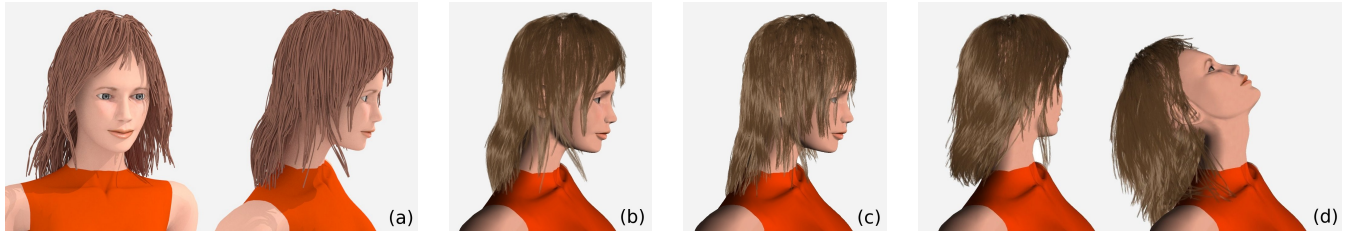


Figure 1: Our inversion method makes the hair synthesis pipeline consistent: (a) Raw hair geometry (a set of polylines) resulting from the manual design or the automatic capture of a static hairstyle (here, a capture from [Herrera et al. 2012]); (b) Input geometry is automatically converted into a dynamic hair model (a set of super-helices) at equilibrium under gravity and frictional hair-body and hair-hair contact forces; Unlike classical hair simulators (c) which ignore surrounding forces when initializing the hairstyle and are thus prone to undesired sagging, our simulator (b) exactly matches the original hair geometry at initial state and (d) yields a realistic, character-specific hair animation.

Abstract

In the latest years, considerable progress has been achieved for accurately acquiring the geometry of human hair, thus largely improving the realism of virtual characters. In parallel, rich physics-based simulators have been successfully designed to capture the intricate dynamics of hair due to contact and friction. However, at the moment there exists no consistent pipeline for converting a given hair geometry into a realistic physics-based hair model. Current approaches simply initialize the hair simulator with the input geometry in the absence of external forces. This results in an undesired sagging effect when the dynamic simulation is started, which basically ruins all the efforts put into the accurate design and/or capture of the input hairstyle. In this paper we propose the first method which consistently and robustly accounts for surrounding forces — gravity and frictional contacts, including hair self-contacts — when converting a geometric hairstyle into a physics-based hair model. Taking an arbitrary hair geometry as input together with a corresponding body mesh, we interpret the hair shape as a static equilibrium configuration of a hair simulator, in the presence of gravity as well as hair-body and hair-hair frictional contacts. Assuming that hair parameters are homogeneous and lie in a plausible range of physical values, we show that this large underdetermined inverse problem can be formulated as a well-posed constrained optimization problem, which can be solved robustly and efficiently by leveraging the frictional contact solver of the direct hair simulator. Our method was successfully applied to the animation of various hair geometries, ranging from synthetic hairstyles manually designed by an artist to the most recent human hair data automatically reconstructed from capture.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: Hair modeling, inverse statics, frictional contact

Links: [DL](#) [PDF](#) [WEB](#) [VIDEO](#) [CODE](#)

1 Introduction

Realistically synthesizing the shape and motion of hair is crucial for representing convincing virtual humans. In the two related fields of Computer Graphics, namely *hairstyling* and *hair animation*, considerable progress has been achieved these latest years. Yet, these two areas of research have remained fairly disconnected from each other. For the sake of flexibility and control, hairstyling is generally performed using a purely geometric process, either through the interactive editing of geometric primitives, or, more recently, with the help of automatic image-based capture methods. Resulting hairstyles are then represented as raw geometric data, with no connection to physics. In contrast, hair animation is often considered as a passive and complex phenomenon that can be captured realistically using physics-based simulation. Before simulation, the hairstyle is generally initialized with a rest pose in the absence of external forces and is typically prone to *sagging* when the simula-

*contact: {alexandre.derouet-jourdan, florence.descoubes} @inria.fr

tion is started (see Figure 1(c)). Such a sagging effect considerably limits control over the static hair shape; in particular, it prevents the simulator from accurately matching an input geometry at initial state. As a result, there exists nowadays a lack of compatibility between, on the one hand, the most realistic hairstyles that can be created or captured, and, on the other hand, the initial input of the most advanced simulators for animating hair. In this paper we present the first approach that consistently and robustly *bridges this gap*. The core of our method amounts to formulating and solving an *inverse problem*, which allows us to interpret raw hair geometry as plausible physical data. More precisely, our method automatically retrieves the hair physical parameters such that the input hairstyle geometry corresponds to a *static equilibrium* of hair under external forces, including gravity and contacts.

Preserving details Real hairstyles often consist of many thin distinctive features — typically, individual hair fibers or thin strands with a peculiar orientation and shape — which fully characterize the visual appearance of hair and play a key role in the person’s identity. Most advanced hair designing tools or image-based reconstruction systems now allow one to capture those features into detailed digital hairstyles. Such details are important to be preserved accurately during the inversion process so as to generate a faithful, character-specific hair simulation. Recent digital hairstyles are often represented as a large set of 3D geometric curves with arbitrarily curved shape, each one of them standing for a single hair fiber or a strand [Paris et al. 2008; Herrera et al. 2012; Luo et al. 2013]. To match such a thin geometry accurately at the initial time step of a dynamic simulation, it is natural to model hair physically using individual 1D deformable structures. In practice we use up to a few thousands thin elastic rods, modeled as super-helices [Bertails et al. 2006], to represent a collection of guide strands. Our inversion method is however generic and could accommodate any other rod model accounting for arbitrary hair curliness [Spillmann and Teschner 2007; Bergou et al. 2008].

Inversion with contacts The key point is to consider that each raw geometric curve intrinsically *encodes*, within its own shape, *all surrounding forces*; typically, gravity, but *also* frictional contact. While gravity is known, frictional contact forces — which encompass tens of thousands hair-body and hair-hair contact forces — are *not*. Accounting for gravity while ignoring the effect of contact forces considerably reduces the complexity of the problem [Derouet-Jourdan et al. 2010], but is not an acceptable option in the case of hair. Indeed, as illustrated in Figure 2 (b), considering that the intrinsic geometry of hair is due to gravity solely tends to increase artificially the rest curvatures of the physical model, making subsequent animation overly stiff. In contrast, in this paper we aim at properly accounting for frictional contact during the inversion phase. For stability and accuracy purposes, we furthermore choose to model frictional contact using unilateral constraints, accounting for non-penetration as well as exact Coulomb friction [Daviet et al. 2011]. We are thus faced with a *large, underdetermined* and *non-smooth* problem. Proposing a suitable, accurate and efficient solution to this challenging problem is the key contribution of our paper. To achieve this, we reasonably assume that the hair physical parameters (linear mass density, stiffness and friction coefficients) take typical values for hair, and that contact forces, which partly support the equilibrium, cannot be excessively large. We furthermore consider that the hair intrinsic curliness (straight, wavy, or curly rest shape) can be roughly estimated from the input geometry. Thinking of this estimation as a target, we formulate the inverse problem as a well-posed convex second-order cone quadratic program (SOCQP) whose unknowns are the frictional contact forces. Interestingly, this SOCQP can be solved robustly and efficiently by leveraging the solver from [Daviet et al. 2011], originally designed for the direct problem with frictional contact.

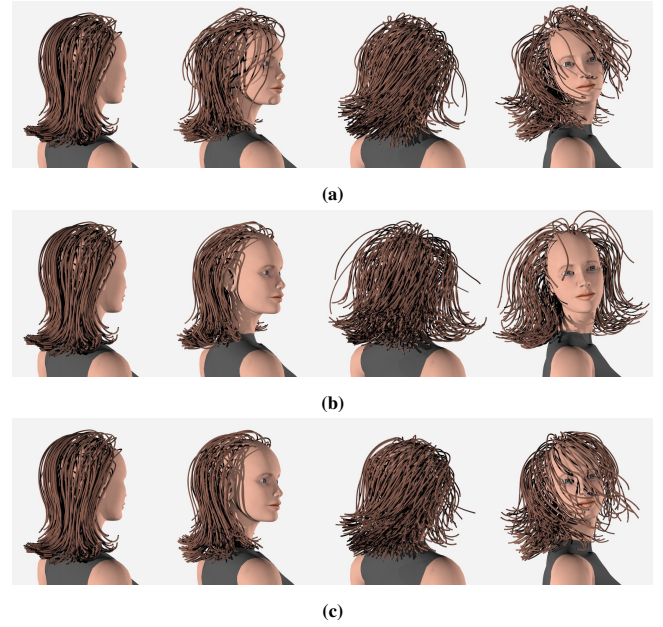


Figure 2: Dynamic simulation of an artistic hairstyle after three different static inversions: an inversion accounting for gravity only [Derouet-Jourdan et al. 2010], without (a) and with (b) stabilization, and our inversion method accounting for gravity and frictional contact forces (c). All three methods exactly match the input style at first frames, when the character is still (left). However, inversion (a) produces a very unstable hair configuration which causes the loss of the groom right at the beginning of the motion, while the stabilized version (b) requires a large stiffness ($E = 60 \text{ GPa}$ on average) and leads to an overly stiff animation, where hair hardly deforms. In contrast, our method (c) remains stable with a reasonable hair stiffness value ($E = 4 \text{ GPa}$) and yields plausible animation results.

2 Related Work

Realistic hair synthesis has drawn the interest of computer scientists since the 90’s, and there is now a large body of research on this topic [Ward et al. 2007a]. While photo-realistic hair rendering is now achievable provided enough computational resources are available, the realistic modeling and animation of hair remain very challenging issues and are still subject to active research. In the sequel we discuss the most relevant works of the latest years with respect to these two areas of research.

Geometry-based hairstyling Most of the hairstyling methods developed so far are purely *geometric*. They rely upon the creation and editing of simple geometric primitives, such as splines, so as to allow a user to carve directly the shape of hair as precisely as desired. Given the complexity of the hair geometry, a number of techniques has been proposed to offer the user interactive and intuitive carving tools. While some models structure hair into large wisps often represented as generalized cylinders [Yang et al. 2000; Kim and Neumann 2002], others view the visible part of hair as surface patches [Koh and Huang 2000] which can be further refined to create intricate styles [Yuksel et al. 2009]. At the final stage, textures or individual fiber geometries are added to produce the rendering output. To enhance visual realism, some authors have also developed physics-inspired tools allowing users to automatically add natural waviness or randomness to hair [Hadap and Magnenat-Thalmann 2000; Yu 2001]. Still, the digital creation of a complex hairstyle remains a lengthy process which definitely requires artistic skills.

Recently, considerable progress has been achieved towards the fast and accurate digital creation of real 3D hairstyles, thanks to the emergence of automatic image-based capture approaches [Kong et al. 1997; Grabli et al. 2002; Paris et al. 2004; Wei et al. 2005; Paris et al. 2008; Jakob et al. 2009; Herrera et al. 2012]. Unlike geometric editing techniques, the output amounts to a set of raw large unstructured data containing tens to hundreds of thousands 3D curves. Very recently, some authors proposed some hierarchical structured hair synthesis frameworks in order to edit a hair capture interactively [Wang et al. 2009] or to extract relevant hair features from the reconstruction process [Luo et al. 2013].

From this whole body of work, a large bunch of realistic, detailed hair geometries is now available in the Computer Graphics community. There is a great temptation to animate them, edit them physically, or even do some reverse engineering to retrieve hair physical properties. However, as the design/acquisition was processed with no connection to physics, output data cannot be interpreted straightforwardly as the result of a physics-based process.

Physics-based hairstyling To circumvent this issue, an alternative is to style hair directly in a plausible physics-based environment, as in a virtual hair salon [Lee and Ko 2001; Bertails et al. 2005; Ward et al. 2007b]. The advantage is that the dynamics of the hairstyle can then be run consistently at any time of the process, since gravity is already set active during hairstyling. However, as interactivity is desirable for the hairstyling step, in practice the user may use very few physics-based primitives combined with a drastic approximation of hair self-contact, which obviously limits the richness of the hairstyles that can be created. Moreover, even though some sketch-based interfaces were designed to alleviate the task of the user [Wither et al. 2007], a physics-based hairstyling process based on indirect operations, such as hair cutting or frizzing, may be tedious and lack intuitiveness. An artist who has a precise hairstyle idea in mind will definitely prefer the flexibility and control offered by geometric methods. Furthermore, accurately reproducing a real hairstyle through costly and hardy controllable physics-based styling tools would be, even for an expert, impracticable.

Hair animation At the other end of the hair synthesis pipeline, a number of works has been devoted to the realistic animation of hair. While some early methods have focused on hair keyframing animation [Koh and Huang 2000], most recent techniques have attempted to simulate hair using a physics-based model accounting for the law of mechanics [Anjyo et al. 1992; Plante et al. 2001; Hadap and Magnenat-Thalmann 2001; Chang et al. 2002; Bertails et al. 2006; Hadap 2006]. Such simulations were for long limited in complexity and resolution due to computational overhead. Recently, some authors demonstrated that the simulation of thousands guide strands subject to complex hair-body and hair-hair interactions was now affordable on a standard desktop computer [Selle et al. 2008; McAdams et al. 2009; Daviet et al. 2011].

One current limitation of hair physics-based simulation is the difficulty to initialize the model accurately with a sample geometry. While artists take hours to model a precise hairstyle, and while capture systems are designed to get every single detail of a hairstyle, those details are lost or severely altered when activating forces and starting a simulation. Some research has been started already in order to bridge this gap. Hadap [2006] and Derouet-Jourdan et al. [2010] leverage exact inverse statics for fibers under gravity, but consider only isolated (i.e., non contacting) fibers. Twigg and Kačić-Alesić [2011] propose to reduce the sagging effect of mass-springs systems (including hair) through nonlinear optimization. In their approach, the net force applied onto each particle (including internal elastic forces, gravity, and contacts modeled as penalty forces) is minimized over the physical parameters of the system. However, the minimum is not guaranteed to vanish, as this would

require the possibility for springs to have a negative rest length. This means that the proposed formulation does not strictly enforce the equilibrium of the resulting physical state, thus offering no guarantee that a true static pose will be recovered in the end. Furthermore, friction is not considered although it plays a crucial role in the statics of hair. In addition, the optimization of repulsion radii per particle for maintaining contact adds a significant cost to the global optimization process. Even with a splitting strategy aimed at solving elasticity and contact parameters separately, for hair scenarios similar to ours their paper reports hours of computations for the optimization process to be completed.

In contrast, our inverse formulation ensures the retrieval of a *perfect equilibrium configuration* of a large collection of hair strands subject to *frictional* contact, and is *efficiently* solved for. Typically, in our tests no more than a few minutes are needed for inverting a couple thousands contacting strands.

3 Contributions and Overview

To the best of our knowledge, computing the exact inverse statics of a fiber assembly while taking frictional contacts (including self-contacts) into account has never been tackled in the past. In this paper we attempt to address this challenging issue.

We first explain how the input geometric data is preprocessed to fit into our inverse static pipeline (Section 4). Then, relying upon the nonsmooth mathematical framework, we introduce the inverse static problem for a fiber assembly subject to frictional contact in the general case (Section 5). Leveraging a few reasonable assumptions about the hair material, we reformulate the general under-determined inverse problem as a well-posed convex optimization problem, which can be solved efficiently and robustly using an existing frictional contact solver (Section 6). Finally, we successfully apply our inversion algorithm to various input hairstyles — including most recent hair captures — and show they can then be animated (or physically edited) in a consistent way (Section 7); further improvements are also discussed so as to refine the accuracy of our technique, before concluding.

4 3D Geometric Fitting

As previously mentioned, we choose to represent the hair geometry as a set of 3D curves (stored as splines or polylines). This format indeed corresponds to the standard output produced by hair designing tools and recent hair capture systems. Furthermore, representing hair with curves is in line with the input of the most advanced hair animation systems, which often simulate a discrete assembly of guide strands [Hadap 2006; Selle et al. 2008; McAdams et al. 2009; Bergou et al. 2010; Daviet et al. 2011].

In order to invert a given hairstyle geometry, we first need to make sure that the geometry is consistent and of reasonable size (Section 4.1). Then, we have to convert this input geometry into the strand geometry of the hair simulator: in our case, piecewise helices (Section 4.2). Finally, to account for contact forces during inversion, we need to infer the location of all potential contacts (hair-body and hair-hair contacts) as precisely as possible, through a geometric analysis of the input data (Section 4.3).

4.1 Suitable input geometric data

Our goal is to build a generic and robust inversion method able to handle a wide range of hairstyles produced by external sources, either through manual design or automatic capture. As far as possible, we wish to avoid laying constraints on the input dataset. Nevertheless, a minimum set of desiderata is required to be able to interpret the hair geometry consistently, as explained below.

Avoiding geometric discrepancies While artists naturally design a dense set of connected curves rooted on the scalp, automatic hair capture systems are often aimed at reconstructing visible parts of hair as accurately as possible, with no specific care for the geometric consistency of individual fibers. Typically, a single hair fiber may consist of many small disconnected curve pieces and/or may not be properly attached to the scalp [Grabli et al. 2002; Paris et al. 2004; Jakob et al. 2009]. Such discrepancies may not be an issue for rendering purposes, but clearly, they cannot be dealt with when attempting to animate the hairstyle. Fortunately, motivated by reusability of data, some recent hair capture systems strive to generate a dense and consistent set of hair curves which satisfy connectivity and are properly attached to the scalp [Paris et al. 2008; Herrera et al. 2012; Luo et al. 2013].

Pruning geometric data While a full head of hair is composed of around 150,000 hair fibers, currently most realistic hair simulators can only deal with a few thousands guide strands. In contrast, most capture systems attempt to reconstruct every single hair fiber composing a full head of hair, thus resulting in too large an amount of data (e.g., hair data generated by [Herrera et al. 2012] are composed of around 100,000 curves). In such a case, we need to keep only a representative subset of the original data. In practice we simply randomly pick a few thousands hair curves among the original data, preserving an even distribution of hair over the scalp. Note that very recently, in the hair capture community some authors strived to retrieve the structure of hair and provide as a result a reasonably sized set of wisps centerlines [Luo et al. 2013].

4.2 Converting input curves into piecewise helices

To model arbitrarily curled hair strands with bending and twisting elasticity, we choose to use the super-helix model [Bertails et al. 2006], parametrized by curvatures and twist. One advantage of this model is that inextensibility is intrinsically preserved during animation with no need for adding any stiff stretch energy nor any external constraint. Furthermore, it was showed that computing the (stable) inverse statics (under gravity, without contacts) of such a model is trivial [Derouet-Jourdan et al. 2010]. A non-trivial part however deals with the geometric fitting of the super-helix model. Unlike low-order (segment-based) models whose centerline is a simple polyline, the super-helix is supported by a G^1 -connected chain of helical arcs. The geometric fitting problem thus amounts to fitting a smooth piecewise helix of N arcs to an arbitrary 3D curve (the input). To solve it accurately and efficiently, we use the *floating tangents* algorithm which was initially introduced in 2D [Derouet-Jourdan et al. 2010] and recently extended to the 3D case [Derouet-Jourdan et al. 2013]. The key idea is to cut the initial curve into N pieces and to interpolate tangents at joints with helical arcs by slightly relaxing corresponding point positions. Once joint points are set into a feasible configuration, the (unique) short helix interpolating two successive points along with their tangents is built by leveraging Ghosh’s co-helicity condition [Ghosh 2010]. In practice, we used between $N = 6$ and $N = 40$ helical arcs for one input curve, depending on its geometric complexity (see Section 7). The maximum conversion time for a full hairstyle made of around 2,000 curves was less than 2 minutes on a standard desktop.

4.3 Detecting contacts from the input geometry

Because the input data represents a full head of hair rooted on a character head and possibly falling upon the shoulders, it necessarily encodes hair-hair and hair-body contacts. To retrieve the corresponding contact forces, a first (geometric) step is necessary to find out where contacts are likely to occur. Before inverting the hairstyle physically, we thus perform a collision detection pass in order to collect the locations and normals of all possible contacts.

Hair strands are approximated with bounding cylinders of constant width and we further assume that we have at our disposal, for each hairstyle, the body mesh corresponding to the underlying character. Using proximity queries and spatial hashing as in [Teschner et al. 2003], we can efficiently collect a plausible list of active contacts.

When experimenting with data however, we are faced with two kinds of issues. First, input curves generated by an artist or reconstructed from a real capture are generally not evenly distributed inside the hair volume. Typically, the visible part of hair may be more densely covered than the hidden part. This often results in a thin empty layer just above the body mesh. In such a case, we use a larger cylindrical approximation for the hair wisps close to the body (in practice, 1 to 10 times as large as other wisps), in order to correctly detect body-hair contacts. Second, for data generated by hair capture systems, we could not always get access to a full body mesh consistent with the data; and in the favorable case, only the head mesh was available. Indeed, from a reconstruction point of view, the only important part of the body that is worth reconstructing carefully is the head, so that captured hair strands can be rooted properly on the scalp [Herrera et al. 2012; Luo et al. 2013]. For our problem however, given that we consider long hairstyles, having a full and consistent head-and-shoulders mesh is required. In practice, each time the mesh of the original supporting body was unavailable, we manually deformed and adjusted a virtual home-made mannequin so that it roughly fits in with the input hairstyle.

5 3D General Inverse Statics

Let us consider an assembly of thin elastic rods with one end clamped (onto the scalp) and the other end free. To simulate the dynamics of this assembly realistically, we rely upon the recent hair simulator of [Daviet et al. 2011]. This simulator accurately accounts for dry (Coulomb) friction within hair, which is crucial for capturing the typical nonsmooth patterns visible on the outer surface of hair. Now, remember that our goal is to bridge the gap between hairstyling and hair animation. More precisely, we wish to initialize the hair simulator with our input hairstyle so that this input data corresponds to a *static equilibrium* configuration of the simulated hair under external forces. In other terms, we consider the *inverse* problem which consists in enforcing the static configuration of hair and retrieving the corresponding physical parameters.

After introducing our main notation (Section 5.1), we recall the broad lines of the direct simulator (Section 5.2), from which we formulate a general underdetermined inverse problem (Section 5.3).

5.1 Notation

As this is typically the case for natural hairstyles, we consider the hair material to be homogeneous. The cross-section area S , volumetric mass ρ , inertia momentum I and Young modulus E of the rods are thus assumed to be uniform (over each rod and over the full hair volume).

The whole fiber assembly can be described as a finite Lagrangian dynamical multi-body system with m generalized coordinates \mathbf{q} . Note that in the case of super-helices, \mathbf{q} collects the discrete material curvatures and twist of all rods composing the assembly. Finally, the \mathbf{q}^0 vector, collecting the generalized coordinates of the rod *at rest*, i.e., in the absence of external forces, models the *intrinsic curliness* of the rod. We shall see in Section 6.1 how to make reasonable assumptions regarding this quantity.

While we model here rods as super-helices, note that other rod models could be used. To make our exposition as general as possible, we keep generic notation until Equation (3).

5.2 Direct dynamic problem with contacts

The nonsmooth equations of motion for a Lagrangian dynamical system subject to contact forces and exact Coulomb friction (see, e.g., [Acary and Brogliato 2008] for a comprehensive introduction) were introduced to Computer Graphics in [Bertails-Descoubes et al. 2011; Daviet et al. 2011], and are briefly reminded here to the reader. In the absence of contact forces, the dynamics of the system is described as a smooth differential equation,

$$\mathcal{M}(\mathbf{q}) \frac{d\dot{\mathbf{q}}}{dt} + \mathbf{F}(t, \mathbf{q}, \dot{\mathbf{q}}) = 0$$

where \mathcal{M} is the generalized mass matrix and \mathbf{F} collects internal elastic forces and external forces such as gravity, as well as nonlinear inertial terms.

We now assume that the dynamical system is subject to n (external, mutual or self) contact forces with Coulomb friction. The equations of motion read¹

$$\begin{cases} \mathcal{M}(\mathbf{q}) \frac{d\dot{\mathbf{q}}}{dt} + \mathbf{F}(t, \mathbf{q}, \dot{\mathbf{q}}) = \mathcal{H}(\mathbf{q})^\top \mathbf{r} \\ \mathbf{u} = \mathcal{H}(\mathbf{q}) \dot{\mathbf{q}} + \mathbf{u}_f(t, \mathbf{q}) \\ \forall i = 1 \dots n, \quad (\mathbf{r}^i, \mathbf{u}^i) \in \mathcal{C}(\mu^i, \mathbf{e}^i), \end{cases} \quad \begin{matrix} (1a) \\ (1b) \\ (1c) \end{matrix}$$

where \mathbf{r} (resp. \mathbf{u}) collects the n contact forces \mathbf{r}^i (resp. the n relative spatial velocities \mathbf{u}^i) at contact points, expressed in the local contact basis defined by the normal \mathbf{e}^i at contact i , \mathcal{H} is the gradient matrix of size $(3n, m)$ relating the spatial velocities \mathbf{u} to the generalized velocities $\dot{\mathbf{q}}$, and \mathbf{u}_f represents the free spatial velocity of the rod at contacts points, transmitted through clamped motion. Inclusion (1c) abstractly represents the frictional contact law as a coupling between each relative velocity \mathbf{u}^i and each contact force \mathbf{r}^i , where μ^i is the value of the friction coefficient at contact i . This coupling, which makes the whole dynamical system *nonsmooth*, includes for each contact the three possible states of the Coulomb law: the *taking-off* case (when contact ceases and objects detach from each other), the *sticking* case, sometimes referred to as *static* friction (when there is no relative motion between objects in contact), and the *sliding* case, sometimes referred to as *dynamic* friction (when objects slide along each other while dissipating energy). Following De Saxcé and Feng [1998], this coupling can be formulated compactly as a complementarity constraint for each contact,

$$\forall i = 1 \dots n, \quad K_{\frac{1}{\mu}} \ni \tilde{\mathbf{u}}^i \perp \mathbf{r}^i \in K_{\mu^i}, \quad (2)$$

where $\tilde{\mathbf{u}}^i$ is the modified velocity $\tilde{\mathbf{u}}^i = \mathbf{u}^i + \mu^i \|\mathbf{u}^i - \mathbf{u}^i \cdot \mathbf{e}^i\| \mathbf{e}^i$, K_{μ} is the second-order cone of aperture μ (the so-called *Coulomb friction cone*), and $K_{\frac{1}{\mu}}$ its dual cone (see Figure 3). Using Moreau's time-stepping method [Moreau 1994] and a root-finding equivalence based on the generalized Fischer-Burmeister function [Fukushima et al. 2002], Daviet and colleagues [2011] designed a robust and efficient one-contact solver accounting for exact Coulomb friction. This solver was then run iteratively (contact by contact) so as to solve the full (nonsmooth and nonconvex) dynamic problem (1). The global solver could successfully handle up to a few thousands contacting rods (up to 50,000 contact points), in a reasonable amount of time (a few minutes per frame).

¹For simplicity we explicitly write the time derivative of the velocity, but this is a misused notation. Indeed, because of contact constraints, the velocity may not be differentiable. A rigorous way of writing the nonsmooth dynamics is to use measure differential inclusions [Moreau 1994].

5.3 Inverse static problem with contacts

Let us now consider the static case, where $\dot{\mathbf{q}} = \mathbf{u} = 0$.

Direct statics In the static case, System (1) and the friction law (2) boil down to

$$\begin{cases} \mathbf{F}(t, \mathbf{q}, \dot{\mathbf{q}} = 0) = \mathcal{H}^\top(\mathbf{q}) \mathbf{r} \\ \forall i = 1 \dots n, \quad \mathbf{r}^i \in K_{\mu^i}. \end{cases} \quad (3)$$

As expected, all contact forces are set in the *sticking* case and should thus belong to the Coulomb friction cone. Let us now explicitly rewrite System (3) in the case of contacting super-helices (made of N elements). Introducing the (constant) diagonal stiffness matrix \mathcal{K} with the j^{th} $3N \times 3N$ block $\mathcal{K}_{kk}^j = \ell_k^j EI$, where ℓ_k^j is the length of the k^{th} element of the j^{th} rod, we get

$$\begin{cases} \mathcal{K}(\mathbf{q} - \mathbf{q}^0) = \mathbf{F}_g(\mathbf{q}) + \mathcal{H}^\top(\mathbf{q}) \mathbf{r} \\ \forall i = 1 \dots n, \quad \mathbf{r}^i \in K_{\mu^i}, \end{cases} \quad (4)$$

where $\mathcal{K}(\mathbf{q} - \mathbf{q}^0)$ collects internal elastic forces, and the j^{th} block $\mathbf{F}_g^j(\mathbf{q})$ of the generalized force \mathbf{F}_g results from the contribution of gravity applied to the j^{th} rod. In the absence of contacts, we retrieve the simple balance condition between elastic and gravitational forces,

$$\mathcal{K}(\mathbf{q} - \mathbf{q}^0) = \mathbf{F}_g(\mathbf{q}). \quad (5)$$

Inverse statics In our problem, the configuration of the rod is *known*: the generalized coordinates \mathbf{q} (discrete curvatures and twists in our case) are directly provided by the geometric fitting of the input hairstyle (see Section 4). Our goal is thus not to solve (4) for the \mathbf{q} variable (direct problem), but instead the *inverse* problem taking as input the configuration \mathbf{q} of the rod. Without contact (see Equation (5)), the unknowns amount to the geometric and physical parameters of the rods: their linear mass density ρS , their stiffness EI , and their intrinsic curliness \mathbf{q}^0 . When the linear mass density and the stiffness are fixed, the equilibrium condition is trivially met by a unique solution for \mathbf{q}^0 , $\mathbf{q}^0 = \mathbf{q} - \mathcal{K}^{-1} \mathbf{F}_g(\mathbf{q})$, and a stability criterion can be stated depending on the ratio $EI/\rho S$ [Derouet-Jourdan et al. 2010]. When contact enters the game however (see System (4)), we have to deal with two types of supplementary unknowns: the n contact forces \mathbf{r}^i , and the corresponding friction coefficients μ^i . While friction coefficients can be reasonably fixed to plausible values ($0.1 \leq \mu \leq 0.3$), contact forces cannot be estimated easily and may take a wide range of values which remain compatible with the equilibrium condition (4). We are thus faced with a large underdetermined system. Yet, a number of reasonable assumptions about the hair material may help us formulate a well-posed problem, as shown below.

6 Inversion as a Well-Posed Convex SOCQP

By leveraging a few reasonable assumptions about hair physical parameters and active contact forces (Section 6.1), we come up with a well-posed, strictly convex second-order cone quadratic program (SOCQP), which can be solved efficiently in practice (Section 6.2)

6.1 Physical assumptions for a well-posed problem

To remove unnecessary underdetermination of our inverse problem (4), we first fix the hair physical parameters for which typical values are known (the linear mass density and the stiffness), and set the friction coefficients to a plausible value (typically, $\mu^i = 0.3$). We are then left with the intrinsic hair curliness \mathbf{q}^0 and the contact forces \mathbf{r} as unknowns. Even though neither one of them can be estimated easily, some reasonable assumptions can be made about them.

Intrinsic curliness as a target Consider the synthetic hairstyle in Figure 4(b). From our visual perception, it is obvious that the hair is intrinsically straight, and that its curved shape near the head and the shoulders is due to contact solely, and by no means to the hair intrinsic shape. Although $\mathbf{r} = 0$ appears to be a possible solution to the inverse statics (4), one would thus like to discard this choice in this case, and instead impose the intrinsic curliness \mathbf{q}^0 to be close to 0. More generally, provided it is possible to estimate a plausible hair intrinsic curliness, one would want to fix it, and then solve (4) to retrieve the corresponding contact forces. Unfortunately, it is not always possible to find a \mathbf{r} satisfying such a problem. Moreover, if the estimation of \mathbf{q}^0 is biased, unreasonably large contact forces may be necessary to satisfy it. Instead, we thus rather take the estimation of the intrinsic curliness as a *target* \mathbf{q}_{est}^0 and penalize the contact forces so that they always keep a reasonable value. This can be formulated mathematically as a well-posed optimization problem, as explained below.

A well-posed convex SOCQP Our goal is to minimize the drift $\|\mathbf{q}^0 - \mathbf{q}_{est}^0\|$ while exactly satisfying Equilibrium (4) and preventing \mathbf{r} from reaching too large values in norm. This amounts to formulating the following optimization problem

$$\begin{aligned} \min_{\mathbf{r}} \quad & \frac{1}{2} \left(\|\mathbf{q}^0(\mathbf{r}) - \mathbf{q}_{est}^0\|^2 + \gamma \|\mathbf{r}\|^2 \right) \\ \text{s.t.} \quad & \mathbf{r}^i \in K_{\mu^i} \quad \forall i = 1 \dots n, \end{aligned} \quad (6)$$

where $\gamma > 0$ is the regularization coefficient for the contact forces. In practice we fixed γ to $1.10^{-3} \text{ N}^{-2} \cdot \text{m}^{-2}$ for all our tests. Expanding the expression of $\mathbf{q}^0(\mathbf{r})$ using Equation (4) yields

$$\begin{aligned} \min_{\mathbf{r}} \quad & \frac{1}{2} \left(\|\mathcal{H}^{-1} \mathcal{H}^\top \mathbf{r} + \mathcal{H}^{-1} (\mathbf{F}_g(\mathbf{q}) - \mathcal{H}(\mathbf{q} - \mathbf{q}_{est}^0))\|^2 + \gamma \|\mathbf{r}\|^2 \right) \\ \text{s.t.} \quad & \mathbf{r}^i \in K_{\mu^i} \quad \forall i = 1 \dots n, \end{aligned}$$

which formally reads

$$\begin{aligned} \min_{\mathbf{r}} \quad & \frac{1}{2} \mathbf{r}^\top \mathcal{W} \mathbf{r} + \mathbf{b}^\top \mathbf{r} \\ \text{s.t.} \quad & \mathbf{r}^i \in K_{\mu^i} \quad \forall i = 1 \dots n, \end{aligned} \quad (7)$$

with $\mathcal{W} = \mathcal{H} \mathcal{H}^{-2} \mathcal{H}^\top + \gamma \mathcal{I}$ (where \mathcal{I} is the identity matrix) and $\mathbf{b} = \mathcal{H} \mathcal{H}^{-2} (\mathbf{F}_g(\mathbf{q}) - \mathcal{H}(\mathbf{q} - \mathbf{q}_{est}^0))$. Note that the objective function $f(\mathbf{r}) = \frac{1}{2} \mathbf{r}^\top \mathcal{W} \mathbf{r} + \mathbf{b}^\top \mathbf{r}$ is quadratic, and has to be minimized under second-order conic constraints: our problem is a second-order cone quadratic program (SOCQP), and belongs to the family of the so-called quadratically constrained quadratic programs (QCQP) [Boyd and Vandenberghe 2004]. Since \mathcal{W} is symmetric positive-definite (definiteness comes from the regularization term), our problem is strictly convex and admits a unique solution \mathbf{r} . We have thus transformed the initial underdetermined problem (4) into the well-posed convex problem (7).

Estimation of the intrinsic curliness For now we have assumed the intrinsic curliness of a given hairstyle could be guessed easily. If one adopts the simplistic assumption that hair grows regularly on the scalp with a uniform curliness, it is indeed possible to retrieve this value approximately. Boundary conditions for thin elastic rods subject to gravity tell us that at the free end of hair fibers, the actual curvature equals the intrinsic curliness. Provided no contact substantially deforms the tip of the hair, one may thus simply measure the actual curvature at the tip to get a plausible estimation of the intrinsic curliness.

For complex, realistic hairstyles however, estimating the intrinsic curliness accurately is not easy. Look at the hairstyle in Figure 1(a). One may guess this hairstyle is *almost* straight, however it seems

that the hair bump above the neck and the slight curvature of the hair strands falling upon the forehead and the cheek are due to some past deformation, as if hair had been deformed and had memorized its history. In such a case, we found out that taking as an estimation the full actual configuration \mathbf{q} yielded better results than taking a uniform intrinsic curvature matching \mathbf{q} at the tip only. Looking back to our minimization problem (6), this choice can be interpreted mechanically. Taking $\mathbf{q}_{est}^0 = \mathbf{q}$ means that we are searching for contact forces that guarantee an exact equilibrium state for hair while minimizing hair internal elastic energy. This implies that, *as far as possible*, we rely on the contact forces to compensate for gravity. Of course, as the number of contact points is arbitrarily sparse and as the contact forces are bounded and constrained to belong to the friction cone, contact is unlikely to support any arbitrary input shape on its own. In the (common) case where contacts are not sufficient to compensate for gravity, then the elastic energy becomes active and contributes to the equilibrium. Note that in the extreme case where not a single contact is declared, then the elastic energy has to compensate for gravity on its own, and the formulation boils down to Equation (5).

The accompanying video illustrates, for the captured hairstyle depicted in Figure 1(a), the difference between choosing \mathbf{q}_{est}^0 as uniform vs. taking \mathbf{q}_{est}^0 equal to \mathbf{q} . Both methods lead to a perfect equilibrium but different behaviors are visible once animation is started. In the former case, some artificial tufts pop out of the hair during motion. Indeed, as hair strands gradually get moving, some initial contact constraints are lost and strands then strive to reach their rest shape, which was set up with a uniform vanishing intrinsic curvature. These undesirable effects are mitigated in the latter case as the actual (non-uniform) curvature \mathbf{q} integrates the small variations along the fibers which are more likely due to their past history (combing, wetting, drying, etc.) rather than contact. Note finally that at the end of the motion, in both cases hair straightens as expected after the character has leant her head back and hair has stopped contacting with the shoulders.

6.2 An efficient and robust SOCQP solver

A challenging optimization problem As our inverse problem (7) is convex, one may naturally attempt to solve it using standard algorithms from convex analysis. Typically, interior points methods constitute a prime family of solvers for quadratic problems subject to quadratic constraints [Boyd and Vandenberghe 2004]. Unfortunately, because our problem is very large in size (thousands to tens of thousands contacts are accounted for), solving it in a reasonable amount of time is likely to be out of reach for standard solvers.

We have tried the interior point method provided by the free Ipopt package [Wächter and Biegler 2006]. Although this solver yielded good convergence results for small size problems (a few hundreds contacts), it could not scale up well to larger problems. To come up with a working solution, we instead turned to a solver dedicated to our type of problem. Actually, as explained below, we managed to find an equivalent formulation of our inverse problem (7) which exactly fits in with the input format of Daviet et al.'s solver, initially introduced for solving the direct problem [Daviet et al. 2011]. Solving (7) then simply boiled down to applying Daviet et al.'s one-step solver, which in practice converged well and efficiently (in a few seconds) for all our examples.

Formulation as a conic complementarity problem Our objective function f is differentiable and convex, and is minimized over a convex set $L = K_{\mu^1} \times \dots \times K_{\mu^n}$ (a product of cones). A standard result of convex analysis (see, e.g., [Boyd and Vandenberghe 2004])

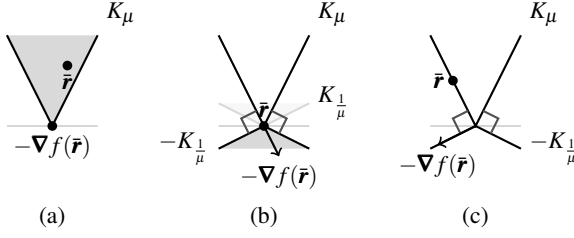


Figure 3: Optimality conditions for a convex minimization problem under a conic constraint, split into three different cases: (a) $\bar{\mathbf{r}}$ lies in the cone and $\nabla f(\bar{\mathbf{r}}) = 0$; (b) $\bar{\mathbf{r}} = 0$ and $-\nabla f(\bar{\mathbf{r}})$ lies in the dual cone; (c) $\bar{\mathbf{r}} \neq 0$ lies on the boundary of the cone and $-\nabla f(\bar{\mathbf{r}})$ lies on the boundary of the dual cone and is normal to $\bar{\mathbf{r}}$.

states that $\bar{\mathbf{r}}$ is a solution to the convex problem (7) if and only if

$$-\nabla f(\bar{\mathbf{r}})^\top (y - \bar{\mathbf{r}}) \leq 0 \quad \forall y \in L \quad (8)$$

where $\nabla f(\mathbf{r})$ is the gradient of f . In our case, $\nabla f(\mathbf{r}) = \mathcal{W}\mathbf{r} + \mathbf{b}$.

The variational inequality (8) can be interpreted as follows. Either the minimum of the unconstrained problem belongs to the convex set L , in which case it obviously corresponds to the solution $\bar{\mathbf{r}}$ of the constrained problem (7) and it is reached for $\nabla f(\bar{\mathbf{r}}) = 0$ (see Figure 3 (a)). Or, the minimum of the unconstrained problem falls outside L , in which case $\bar{\mathbf{r}}$ should lie on the border of L , and $\nabla f(\bar{\mathbf{r}}) \neq 0$ should make an obtuse angle with any vector of L passing through $\bar{\mathbf{r}}$. Indeed, if this was not the case, it would be still possible to minimize f while remaining in the constraints set L , by descending along the projection of $-\nabla f(\bar{\mathbf{r}})$ onto the convex set L .

Let us have a deeper look at the latter case, when L is, for simplicity, a single cone K_μ . The analysis below can be generalized easily to a product of cones. First, consider that $\bar{\mathbf{r}} = 0$. In this case, the inequality (8) is satisfied if and only if ∇f lies anywhere inside the opposite of the dual cone K_μ^\perp (see Figure 3 (b)). Now assume that $\bar{\mathbf{r}} \neq 0$ and that $\bar{\mathbf{r}}$ lies on the (smooth) boundary of K_μ . Then one may identify (8) as the definition of the normal vector to a smooth surface, and realize that ∇f is the outward normal of the cone K_μ at point $\bar{\mathbf{r}}$ (see Figure 3, (c)).

Unifying the three cases described above, the optimality conditions of our SOCQP are thus equivalent to the following complementarity condition

$$\forall i = 1 \dots n, \quad K_\mu^\perp \ni (\nabla f(\mathbf{r}))^i \perp \mathbf{r}^i \in K_\mu^i. \quad (9)$$

Interestingly, this condition is mathematically similar to the complementarity condition (2) of the dynamic case, except that the dynamic variable $\tilde{\mathbf{u}}^i$ has been replaced with $(\nabla f(\mathbf{r}))^i$, i.e., the i^{th} 3×1 block of the gradient. Now, the solver of [Daviet et al. 2011] was designed to solve exactly this kind of conic complementarity, i.e., problems which can be formulated as

$$\forall i = 1 \dots n, \quad K_\mu^\perp \ni (\mathbf{W}\mathbf{r} + \mathbf{c})^i \perp \mathbf{r}^i \in K_\mu^i,$$

where \mathbf{W} is a symmetric positive (semi-)definite matrix of size $(3n, 3n)$, and \mathbf{c} a vector of size $3n$. In the direct (dynamic) case, \mathbf{W} was identified to the sparse (discrete) Delassus operator $\mathcal{H}\mathcal{M}^{-1}\mathcal{H}^\top$ [Daviet et al. 2011]. In our static inversion problem, \mathbf{W} is identified to our \mathcal{W} matrix, which is sparse, symmetric and positive-definite.

We can thus apply Daviet et al.’s solver straightforwardly to solve our inverse problem (9), and thus our initial SOCQP (7). Unlike the

Table 1: Input and performance of our inversion algorithm.

Input hairstyle	$nb_{strands}$	$nb_{contacts}$	$T_{geom.fitting}$ (s)	$T_{inv.statics}$ (s)
straightSynthSim	240	2,353	-	2.6
wavySynthArtist	240	8,922	26	5
straightCapture	1,980	30,381	21	19
curlyCapture	1,105	14,358	95	15

dynamic problem (2), our inverse problem (9) is convex. It is thus likely to be easier to solve. In practice, simply using the primary Fischer-Burmeister solver of [Daviet et al. 2011] (with no fail-safe) proved to converge well for all the problems we have tested.

7 Results and Discussion

Our inversion method was applied to four long hairstyles coming from various sources:

- **straightSynthSim**: a synthetic straight hairstyle generated by simulation (see Figure 4(b));
- **wavySynthArtist**: a synthetic wavy hairstyle designed by an artist (see Figure 2, left);
- **straightCapture**: a real straight hairstyle recently captured by Herrera and colleagues [2012] (see Figure 1(a));
- **curlyCapture**: a real curly wig (resting on the head of a mannequin with no shoulders) recently captured by Luo and colleagues [2013] (see Figure 5(a)).

In this section and in the accompanying video, we present and analyze the results yielded by our inversion method, for each input hairstyle. We also validate our inversion process on the simple synthetic case **straightSynthSim**, where all physical parameters are known and can thus be compared against the retrieved values. Then, we show how each input hairstyle can be animated consistently and even further edited in a physically-based manner. Finally, we discuss the strengths and limitations of our approach.

7.1 Results

Parameters and performance Table 1 summarizes the main features of the input hairstyles we have tested, as well as the computational timings for the full pipeline.

Unlike other hairstyles, **straightCapture** was provided with no consistent body mesh attached to it. For this hairstyle we have thus manually adjusted our body mesh so that it roughly fits in with the hair, with no interpenetration. To be able to handle the extremely intricate style **curlyCapture**, we have allowed ourselves to slightly smooth out the original shape by using a limited number of elements per super-helix (up to 40).

Except for **straightSynthSim** where we took $\mathbf{q}_{est}^0 = 0$, we have found out that taking $\mathbf{q}_{est}^0 = \mathbf{q}$ yielded better results, as explained in Section 6.1 and illustrated in the video. The γ parameter was fixed to $1.10^{-3} \text{ N}^{-2} \cdot \text{m}^{-2}$ for all our tests.

Timings were measured on a standard desktop computer featuring a CPU running at 2.3 GHz. Overall, it turns out that our method is pretty fast. Even for complex hairstyles made of almost 2,000 strands, the total amount of time necessary for fitting curves to piecewise helices and solving the SOCQP problem never exceeds 2 minutes. Compared to other hairstyles, **wavySynthArtist** is longer to fit geometrically because it is originally composed of many small Bézier patches, which slows down the sampling step.

Validation under controlled conditions To check that our inversion process computes plausible physical parameters, we performed a simple experiment using the hairstyle **straightSynthSim** synthesized with the hair simulator from [Daviet et al. 2011].

Figure 4 illustrates the physical process (a) for creating our synthetic hairstyle (b) and shows the resulting hair rest pose once body contacts have been removed, in 3 different cases: after the original simulation was continued (c), after a new simulation was initialized from (b) using inversion accounting for gravity only (d), and after a new simulation was initialized from (b) using our new inversion algorithm (e). As expected, hair falls straight in the original simulation (c). Our inverted hairstyle (e) behaves similarly, meaning that our computed parameters are plausible. In practice we measured an average error of 0.9 m^{-1} between the original intrinsic curliness and the one retrieved by inversion (compared to an average error of 9.9 m^{-1} for the inversion without contacts, with the same Young modulus). In contrast, as the inversion in (d) does not account for contact, hair artificially keeps its original pose, as if the body was still there. Furthermore, to be stable, it requires a large Young modulus ($E = 100 \text{ GPa}$), making the animation overly stiff.

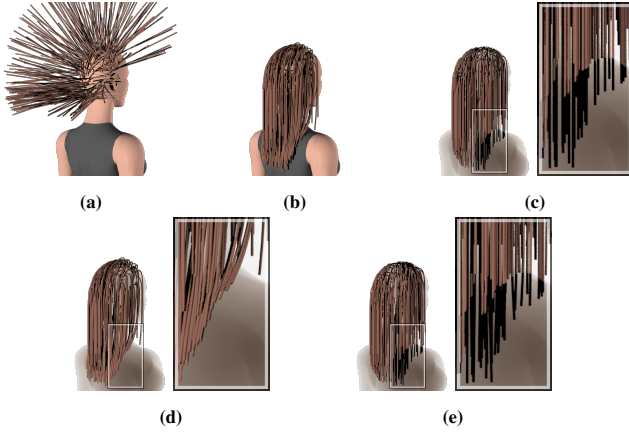


Figure 4: Generation of a synthetic hairstyle by simulation: initial hair strands (a) are simulated under gravity to form a static shape (b). When removing the body this hairstyle falls straight (c). Inverting the input style (b) with our method and then applying gravity to it (e) yields the same behavior as the reference (c). In contrast, the curved shape of hair due to contacts is artificially preserved (d) when naïvely using the inversion process from [Derouet-Jourdan et al. 2010] which accounts for gravity only.

Consistent physics-based animation and editing Once a hairstyle has been inverted by our method, we have at our disposal a plausible hair physics-based model whose static shape perfectly matches the input geometry. Straightforward applications of our technique then include physics-based animation as well as physical editing (such as trimming the hairstyle).

Figures 1(d), 2(c), and 5(b) show the animation of the input hairstyles **straightCapture**, **wavySynthArtist**, and **curlyCapture** respectively, on a rotating and leaning head motion. Compared to the naïve approach without inversion, which causes undesired sagging, our method nicely preserves the initial equilibrium configuration once the dynamic simulation is started, as long as the character remains still. Once the character moves, hair starts to flow naturally. Note that the captured hairstyle **curlyCapture** was particularly challenging to invert and animate, due to its intricate geometry. To remove instabilities, we had to increase the accuracy of the contact solver. In practice we decreased the tolerance of the solver ($\times 10^{-3}$) for this simulation.

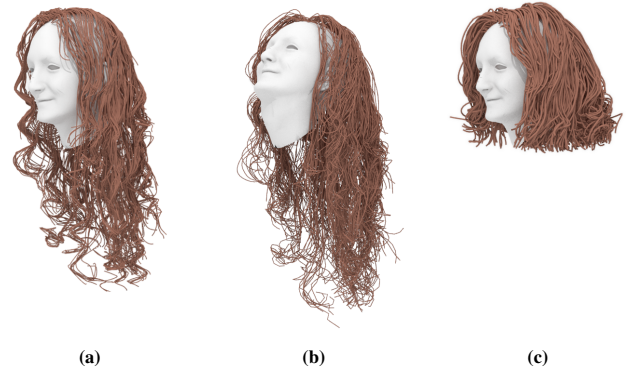


Figure 5: Real curly wig (a) captured from [Luo et al. 2013], inverted by our method and physically animated (b) and trimmed (c).

We have also applied a virtual trimming to the two captured hairstyles and observed that hair tended to pull up and expand, as expected (see Figure 5(c)). Unlike geometric methods that simply truncate the strands, our method allows one to *predict* the physical effect of removing mass to the material.

Finally, as previously mentioned, note that our inversion method is versatile and could be applied to hair simulators relying upon other rod models. For instance, the Discrete Elastic Rods model [Bergou et al. 2008] would naturally fit into our framework, the sole difference being the structure of the stiffness matrix \mathcal{K} which would be banded instead of diagonal.

7.2 Limitations

Although we were able to apply our inversion technique to challenging input hairstyles, we feel our method could be improved in a number of ways.

Refining input captured data One important issue that our practical experiments revealed is that the information provided by hair capture systems is insufficient compared to our needs. Our goal was to start from raw hair data with no a priori knowledge about how hairstyles were captured and reconstructed. However we realized that supplementary information w.r.t. the accurate geometry of contacting objects would have helped us a lot for the inversion. Indeed, one critical step of our inversion process is the declaration of active contacts. If some contacts are not set active where they should be, the solver may find excessively large intrinsic hair curliness for accounting for the static shape, causing further instability during animation. Conversely, if too many contacts are declared, this may cause unnecessarily large computational timings for the inversion and the subsequent simulation.

One key information that would be first required is the accurate geometry of the supporting body mesh. Except for **curlyCapture** where we had at our disposal the original mesh of the supporting head, for other captured styles that we have tested, we asked an artist to manually adjust a virtual mannequin so that it roughly fits in with the hair data. This was feasible for **straightCapture** (although we have probably altered the original position of the real person), but not for the other styles we have tested (either the hair was not contacting the surface of the body mesh at some places, or it was penetrating through the mesh at some other places).

Second, when input curves represent the centerlines of large wisps (this is the case, e.g., for artistic hairstyles or structured hair captures [Luo et al. 2013]), the geometric representation of contacting wisps as cylinder chains with constant width may be too far from their real shape. Our ideal input would consist of a set

of wisps centerlines evenly distributed, together with the accurate geometry of their envelope. Envelopes should be reconstructed or approximated in such a way that neighboring wisps do contact each other, with no penetration.

Given the fact that the level of accuracy reached by our inversion method is tightly bound to the hair acquisition process, we think that in the future, an optimal way of refining the full digital hair cloning process from capture to animation will be to join efforts from both communities.

Towards the inversion of hair motion Overall, we believe the geometry of a single, static hairstyle already reveals a lot about the mechanical state of hair. With simple assumptions about hair homogeneity and very few parameters to tune, we have shown that our method could automatically retrieve plausible values for the hair intrinsic curliness as well as hair-body and hair-hair contact forces. Compared to recent hair animation approaches which re-synthesize hair motion from the capture of real hair motion [Luo et al. 2011; Zhang et al. 2012] or the learning of precomputed hair simulations [Guan et al. 2012], our method offers an interesting alternative since it relies upon limited input data (a single hair geometry) and benefits from the versatility of physics-based approaches.

However, our approach may suffer from multiple inaccuracies due to incomplete input information. Although our simple estimation of the hair intrinsic curliness yielded satisfying results in our tests, it may be a poor approximation if contacts severely alter the shape of hair. Likewise, tuning the parameter γ to better balance the role of contacts w.r.t. elasticity in the equilibrium configuration may not be very intuitive for the user. Moreover, in our tests we arbitrarily fixed the friction coefficients to a plausible value, but the *stability* of the static hairstyle may be fairly sensitive to the choice of this value. Combining our approach to others which analyze data of hair in motion would certainly help us refine the estimation of such parameters, and thus improve the accuracy of our inversion process.

Conclusion

We have presented the first robust method that converts an arbitrary hairstyle geometry into a plausible physics-based hair model suitable for subsequent animation as well as for further physical editing. Because our method relies on detailed but affordable geometry data, we believe it could pave the way for a number of new exciting applications, such as the *dynamic* digital cloning of characters or the *predictive, customer-specific* virtual styling of hair. In the future we would like to compare our method against experimental hair measurements so as to better assess and possibly refine the range of validity of our approach, with the aim to develop an accurate reverse engineering approach for hair. Moreover, we would be interested in applying our method to other materials whose static shape may also reveal a lot of interesting underlying mechanics, such as folded cloth. Finally, we are intrigued by the good robustness and efficiency of our new SOCQP solver w.r.t. large problems, and plan to perform thorough comparisons with current state-of-the-art optimization approaches, on an extended range of benchmarks.

Acknowledgments

We would like to thank Laurence Boissieux for creating the character meshes and motions as well as the synthetic wavy hairstyle, and Romain Casati for producing many of the final renderings, using the opensource YafaRay raytracer (<http://www.yafaray.org>). We are also very grateful to Tomas Lay Herrera, Arno Zinke, Andreas Weber (Bonn University) and Linjie Luo, Hao Li, Szymon Rusinkiewicz (Princeton University) for sharing with us their latest data of captured hair. Finally, we would like to thank the anonymous reviewers for their useful comments.

References

- ACARY, V., AND BROGLIATO, B. 2008. *Numerical methods for nonsmooth dynamical systems*, vol. 35 of *Lecture Notes in Computational and Applied Mechanics*. Springer.
- ANJYO, K., USAMI, Y., AND KURIHARA, T. 1992. A simple method for extracting the natural beauty of hair. In *Computer Graphics Proceedings (Proc. ACM SIGGRAPH'92)*, 111–120.
- BERGOU, M., WARDETZKY, M., ROBINSON, S., AUDOLY, B., AND GRINSPUN, E. 2008. Discrete elastic rods. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH'08)* 27, 3, 1–12.
- BERGOU, M., AUDOLY, B., VOUGA, E., WARDETZKY, M., AND GRINSPUN, E. 2010. Discrete viscous threads. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH'10)* 29, 4.
- BERTAILS-DESCOUBES, F., CADOUX, F., DAVIET, G., AND ACARY, V. 2011. A nonsmooth Newton solver for capturing exact Coulomb friction in fiber assemblies. *ACM Transactions on Graphics* 30 (February), 6:1–6:14.
- BERTAILS, F., AUDOLY, B., QUERLEUX, B., LEROY, F., LÉVÊQUE, J.-L., AND CANI, M.-P. 2005. Predicting natural hair shapes by solving the statics of flexible rods. In *Eurographics'05 (short papers)*, J. Dingliana and F. Ganovelli, Eds., Eurographics. Eurographics'05 (short papers).
- BERTAILS, F., AUDOLY, B., CANI, M.-P., QUERLEUX, B., LEROY, F., AND LÉVÊQUE, J.-L. 2006. Super-helices for predicting the dynamics of natural hair. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH'06)* 25, 1180–1187.
- BOYD, S., AND VANDENBERGHE, L. 2004. *Convex Optimization*. Cambridge University Press.
- CHANG, J., JIN, J., AND YU, Y. 2002. A practical model for hair mutual interactions. In *ACM SIGGRAPH - EG Symposium on Computer Animation (SCA'02)*, 73–80.
- DAVIET, G., BERTAILS-DESCOUBES, F., AND BOISSIEUX, L. 2011. A hybrid iterative solver for robustly capturing Coulomb friction in hair dynamics. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH Asia'11)* 30, 139:1–139:12.
- DE SAXCÉ, G., AND FENG, Z.-Q. 1998. The bipotential method: a constructive approach to design the complete contact law with friction and improved numerical algorithms. *Math. Comput. Modelling* 28, 4-8, 225–245.
- DEROUET-JOURDAN, A., BERTAILS-DESCOUBES, F., AND THOLLOT, J. 2010. Stable inverse dynamic curves. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH Asia'10)* 29 (December), 137:1–137:10.
- DEROUET-JOURDAN, A., BERTAILS-DESCOUBES, F., AND THOLLOT, J. 2013. Floating tangents for approximating spatial curves with G^1 piecewise helices. *Computer Aided Geometric Design* 30 (June).
- FUKUSHIMA, M., LUO, Z.-Q., AND TSENG, P. 2002. Smoothing functions for second-order-cone complementarity problems. *SIAM J. on Optimization* 12 (February), 436–460.
- GHOSH, S. 2010. *Geometric approximation of curves and singularities of secant maps. A differential geometric approach*. PhD thesis, University of Groningen, Johann Bernoulli Institute for Mathematics and Computer Science.

- GRABLI, S., SILLION, F., MARSCHNER, S., AND LENGUEL, J. 2002. Image-based hair capture by inverse lighting. In *Proc. Graphics Interface*, 51–58.
- GUAN, P., SIGAL, L., REZNITSKAYA, V., AND HODGINS, J. 2012. Multi-linear data-driven dynamic hair model with efficient hair-body collision handling. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '12, 295–304.
- HADAP, S., AND MAGNENAT-THALMANN, N. 2000. Interactive hair styler based on fluid flow. In *EG workshop on Computer Animation and Simulation (EG CAS'00)*, 87–100.
- HADAP, S., AND MAGNENAT-THALMANN, N. 2001. Modeling dynamic hair as a continuum. *Computer Graphics Forum (Proc. Eurographics'01)* 20, 3, 329–338.
- HADAP, S. 2006. Oriented strands - dynamics of stiff multi-body system. In *ACM SIGGRAPH - EG Symposium on Computer Animation (SCA'06)*, ACM-EG SCA, 91–100.
- HERRERA, T. L., ZINKE, A., AND WEBER, A. 2012. Lighting hair from the inside: A thermal approach to hair reconstruction. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia'12)* 31.
- JAKOB, W., MOON, J., AND MARSCHNER, S. 2009. Capturing hair assemblies fiber by fiber. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia'09)* 28.
- KIM, T.-Y., AND NEUMANN, U. 2002. Interactive multiresolution hair modeling and editing. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH'02)* 21, 3 (July), 620–629.
- KOH, C., AND HUANG, Z. 2000. Real-time animation of human hair modeled in strips. In *EG workshop on Computer Animation and Simulation (EG CAS'00)*, 101–112.
- KONG, W., TAKAHASHI, H., AND NAKAJIMA, M. 1997. Generation of 3D hair model from multiple pictures. *Proceedings of Multimedia Modeling*, 183–196.
- LEE, D.-W., AND KO, H.-S. 2001. Natural hairstyle modeling and animation. *Graphical Models* 63, 2 (March), 67–85.
- LUO, L., LI, H., WEISE, T., PARIS, S., PAULY, M., AND RUSINKIEWICZ, S. 2011. Dynamic hair capture. Tech. rep., Princeton University, August.
- LUO, L., LI, H., AND RUSINKIEWICZ, S. 2013. Structure-aware hair capture. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH'13)*. to appear.
- MCADAMS, A., SELLE, A., WARD, K., SIFAKIS, E., AND TERAN, J. 2009. Detail preserving continuum simulation of straight hair. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH'09)* 28, 3, 1–6.
- MOREAU, J. 1994. Some numerical methods in multibody dynamics: Application to granular materials. *European Journal of Mechanics - A/Solids* 13, 4, 93–114.
- PARIS, S., BRICEÑO, H., AND SILLION, F. 2004. Capture of hair geometry from multiple images. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH'04)*, 712–719.
- PARIS, S., CHANG, W., KOZHUSHNYAN, O., JAROSZ, W., MATUSIK, W., ZWICKER, M., AND DURAND, F. 2008. Hair photobooth: geometric and photometric acquisition of real hairstyles. In *ACM Transactions on Graphics (Proc. of SIGGRAPH 2008)*, ACM, New York, NY, USA, 1–9.
- PLANTE, E., CANI, M.-P., AND POULIN, P. 2001. A layered wisp model for simulating interactions inside long hair. In *EG workshop on Computer Animation and Simulation (EG CAS'01)*, Springer, Computer Science, 139–148.
- SELLE, A., LENTINE, M., AND FEDKIW, R. 2008. A mass spring model for hair simulation. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH'08)* 27, 3, 1–11.
- SPILLMANN, J., AND TESCHNER, M. 2007. CoRdE: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects. In *ACM SIGGRAPH - EG Symposium on Computer Animation (SCA'07)*, ACM-EG SCA, 63–72.
- TESCHNER, M., HEIDELBERGER, B., MÜLLER, M., POMERANERTS, D., AND GROSS, M. 2003. Optimized spatial hashing for collision detection of deformable objects. In *Vision, Modeling, Visualization (VMV 2003)*, 47–54.
- TWIGG, C., AND KAČIĆ-ALESIĆ, Z. 2011. Optimization for sag-free simulations. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, SCA '11, 225–236.
- WÄCHTER, A., AND BIEGLER, L. 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.* 106, 1 (May), 25–57.
- WANG, L., YU, Y., ZHOU, K., AND GUO, B. 2009. Example-based hair geometry synthesis. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH'09)* 28, 3 (July), 56:1–56:9.
- WARD, K., BERTAILS, F., KIM, T.-Y., MARSCHNER, S., CANI, M.-P., AND LIN, M. 2007. A survey on hair modeling: Styling, simulation, and rendering. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 13, 2 (Mar-Apr), 213–24.
- WARD, K., GALOPPO, N., AND LIN, M. 2007. Interactive virtual hair salon. *Presence: Teleoper. Virt. Environ.* 16, 3, 237–251.
- WEI, Y., OFEK, E., QUAN, L., AND SHUM, H.-Y. 2005. Modeling hair from multiple views. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH'05)*, 816–820.
- WITHER, J., BERTAILS, F., AND CANI, M.-P. 2007. Realistic hair from a sketch. In *International Conference on Shape Modeling and Applications, SMI 2007, June, 2007*, IEEE, Lyon, France, IEEE, 33–42.
- YANG, X., XU, Z., YANG, J., AND WANG, T. 2000. The cluster hair model. *Graphics Models and Image Processing* 62, 2 (Mar.), 85–103.
- YU, Y. 2001. Modeling realistic virtual hairstyles. In *Proceedings of Pacific Graphics'01*, 295–304.
- YUKSEL, C., SCHAEFER, S., AND KEYSER, J. 2009. Hair meshes. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH Asia'09)* 28, 5 (Dec.), 166:1–166:7.
- ZHANG, Q., TONG, J., WANG, H., PAN, Z., AND YANG, R. 2012. Simulation guided hair dynamics modeling from video. *Comp. Graph. Forum* 31, 7pt1 (Sept.), 2003–2010.