

Simulation-Ready Hair Capture

Liwen Hu^{1,2,3} Derek Bradley¹ Hao Li^{2,3,4} Thabo Beeler¹

¹Disney Research ²Pinscreen ³University of Southern California ⁴USC Institute for Creative Technologies

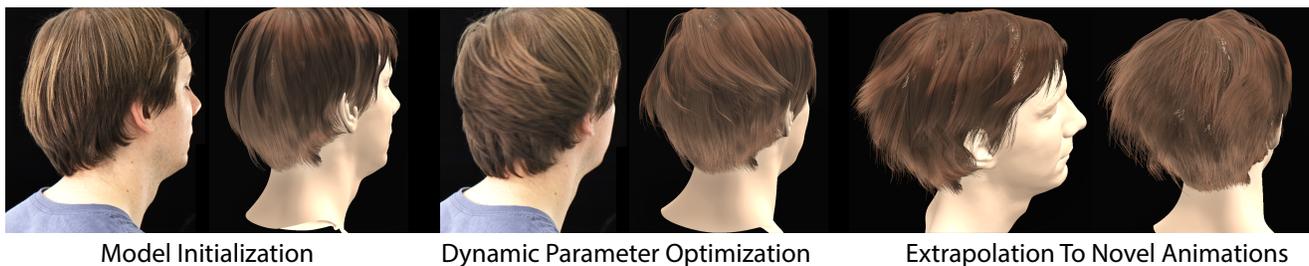


Figure 1: We present a framework to automatically determine optimal parameters of hair simulation models from video footage such that the simulation matches the motion of the real hair. This allows to simulate the same hairstyle under novel animations.

Abstract

Physical simulation has long been the approach of choice for generating realistic hair animations in CG. A constant drawback of simulation, however, is the necessity to manually set the physical parameters of the simulation model in order to get the desired dynamic behavior. To alleviate this, researchers have begun to explore methods for reconstructing hair from the real world and even to estimate the corresponding simulation parameters through the process of inversion. So far, however, these methods have had limited applicability, because dynamic hair capture can only be played back without the ability to edit, and solving for simulation parameters can only be accomplished for static hairstyles, ignoring the dynamic behavior. We present the first method for capturing dynamic hair and automatically determining the physical properties for simulating the observed hairstyle in motion. Since our dynamic inversion is agnostic to the simulation model, the proposed method applies to virtually any hair simulation technique, which we demonstrate using two state-of-the-art hair simulation models. The output of our method is a fully simulation-ready hairstyle, consisting of both the static hair geometry as well as its physical properties. The hairstyle can be easily edited by adding additional external forces, changing the head motion, or re-simulating in completely different environments, all while remaining faithful to the captured hairstyle.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Virtual characters have many distinguishing features that define the identity of the character. After the face, one of the most important features is the hairstyle. Modeling and animating hair has been a rich topic of research in computer graphics because creating believable CG hairstyles is extremely complicated. A real-world hairstyle consists of tens of thousands of hair strands with varying lengths, thicknesses and material properties, all of which affect the shape and the motion of the hair. On top of this, as hair moves it creates thousands of collisions with varying levels of friction depending on whether the hair is dry, wet, oily, or contains consumer hair prod-

ucts. All of these factors must be taken into consideration when modeling and animating the hairstyle of a virtual character, as inconsistencies in the hairstyle can contribute just as much to the infamous uncanny valley as inconsistencies in the facial performance.

Animating all these complex effects by hand is impractical, and so the traditional approach is to simulate hair physically. This approach allows the user to define physical parameters of the hair strands and then computationally simulate how the hairstyle should react to external forces such as gravity, head movement or wind. Modern simulators can compute and handle complex interactions such as collisions, friction and electrostatic forces and simulate

thousands of hair strands. Since the result is a physically plausible dynamic hairstyle, simulation is currently the industry method of choice for hair. One major drawback of hair simulation, however, is that it can be very challenging to define the physical parameters of the hair model in order to obtain a desired hairstyle with desired motion. Determining the parameters for every fiber is impractical, and typically animators resort to a single set of parameters for the complete hairstyle. But even just selecting a single set of parameters to match a particular hairstyle is challenging and typically requires a lot of trial and error, especially if the hair is for a digital double of a real person.

To obtain specific hairstyles, recent research has focused on capturing hair from the real world, following trends in facial performance capture. Hair capture, however, is a much more challenging problem because the majority of the hair strands are occluded, and hair usually has a shiny appearance which challenges existing computer vision algorithms. Nevertheless, a few techniques have emerged that can recover a static hairstyle from images. While static hair capture removes the need to model a hairstyle, it still leaves the animator with the challenging task of selecting appropriate physical parameters for simulation if the hair is to be animated. Some methods are able to reconstruct the motion of dynamic hair from videos, however these methods can only play-back and re-render the hair from different viewpoints without the ability to edit the animation in any way. Without allowing artist control, the usefulness of current dynamic hair digitization methods becomes limited as additional forces cannot be added and motion adjustments or collisions are not possible after the capture process.

The method proposed in this paper is fundamentally different in that it does not present another capture algorithm with the goal of acquiring a particular hair performance as accurately as possible, but rather a data-driven parameter estimation method that finds the optimal simulation parameters given a specific simulation method, such that physically simulating the hairstyle reproduces hair motion that closely resembles the observed hair dynamics. The main advantage of such an approach over pure capture methods is that the observed hairstyle can not only be played back, but can be re-simulated to create any other animation while remaining faithful to the physical behaviour of the captured hairstyle.

The seminal work of Derouet-Jourdan et al. [DJBDT13] presented a first step towards this goal, as they recover a simulation hair model from images of a hairstyle at a static point in time under gravity, which allows to animate the reconstructed hair using physical simulation. However, only considering the static state under gravity does not guarantee to extract the correct physical properties that will remain faithful to the observed hairstyle when in motion, since hair geometry and physical properties are directly connected and can only be separated under dynamics. As an example, demonstrated in Fig. 2, several sets of very different physical parameters can result in the same hairstyle under gravity, but very different dynamic behavior given head motion.

We present the first approach to reconstruct truly “simulation-ready” hairstyles, in that we produce both the geometry of a hairstyle and estimate simulation parameters that reproduce the observed dynamics as faithfully as possible given the chosen simulation method. In order to cater to different existing hair simulation



Figure 2: Simulation results with different physical parameters. Here we show simulation results for three different sets of physical parameters, which yield the same static result under gravity at time t_0 , but substantially differ under motion. As a consequence, computing physical parameters from a static frame under gravity as proposed by Derouet-Jourdan et al. [DJBDT13] is not sufficient.

techniques, we present a general framework for dynamic parameter inversion, which aims to treat simulation as a black-box. After minor initialization, our optimization method is agnostic to the chosen hair simulation model, allowing to find optimal parameters for virtually any hair simulation method, which we demonstrate on two state-of-the-art hair simulation models: (1) the *super-helices* model of Bertails et al. [BAC*06] and (2) the *discrete elastic rods* presented by Bergou et al. [BWR*08]. We employ particle swarm optimization (PSO) [KE95] and optimize for the simulation parameters of a number of guide strands by iteratively simulating and computing a residual error to the observed dynamic hair, ultimately minimizing an objective that takes into account the observed hair orientation field, the approximate silhouette of the hair volume, and the perpendicular strand tracking over time. The simulation model serves essentially as regularizer in this optimization, and it depends solely on the chosen model how closely the perceived hair motion can be matched. Hair dynamics are extremely complex and governed by various elements, such as hair-hair collisions, hair-hair friction, electrostatic forces, or hair-head collisions to name just a few, and various simulators implement more or fewer of these elements. Since the proposed dynamic inversion employs the simulator itself as part of the optimization, it will find parameters that optimally match the target hair motion within the capabilities of the simulator. The estimated parameters can then directly be used to simulate the hair under novel head motion, in the presence of external effects such as blowing wind, or artistically edited by adjusting the simulation parameters to make the animated hair look more wet or be more stiff due to hair products – all while remaining faithful to the hairstyle of the actor. As such, this work represents a major step towards practical hair modeling and animation.

2. Related Work

In most production settings, the modeling of realistic digital hair is still a fairly manual effort relying on sophisticated 3D modeling frameworks and often involves intuitive hair design tools [WBC07, FWTQ07, YSK09, WWL*13, YYCY14] or procedural hair generation techniques [WYZG09]. Due to the complex behavior and intricate interactions of individual hair strands, physically-based sim-

ulation [BAC*06, SLF08, BWR*08, BAV*10, KTS*14] is still the method of choice for animating compelling hair despite a usually tedious process of parameter tuning. A long thread of established techniques has been investigated in [WBK*07], trading off realism and computational cost. State-of-the-art collision response algorithms [KTS*14] and solvers that capture non-smooth friction in large hair assemblies [DBDB11] ensure realistic dynamic hair behavior. The idea of capturing hair directly from the real world is a relatively new paradigm, and we review recent advances from a modeling, animation, and simulation perspective.

Static Hair Digitization. As an effort to advance and scale the production of digital humans, automatic 3D digitization techniques have been introduced to generate realistic static hair models at the strand level using various sensing techniques. Paris and colleagues [PnS04] introduced a method that captures static hair geometry by extracting the 2D orientation field of highlights using a moving light source with known positions. They later expand on this technique using multiple projectors and cameras to capture both, shape and appearance of a wide range of complex hairstyles [PCK*08]. Herrera and colleagues [LHZW12] used thermal imaging to handle issues caused by shadowing and anisotropic reflectance and multiple macrophotographs were used by Jakob et al. [JMM09] to capture individual hair fibers within a small working volume. Because the slightest motion can deform hair, synchronized multi-view stereo systems are particularly popular since a subject can be instantaneously captured. Robust 2D orientation filters [PCK*08] have been developed to extract individual strand structures from the raw images and to reconstruct both short [BBN*12] and long 3D hair strands [LLP*12, LLR13, HMLL14]. Due to complex occlusions and the intertwined geometry of many hairstyles, locally coherent wisp structures are discovered by Luo et al. [LLR13] to bridge disconnected hair strands and complete missing data. Even though visually pleasing strands are re-synthesized from a scalp model, this local hair growing strategy is sensitive to local minima and often produces implausible hair strands, visible during animation. The data-driven method of Hu et al. [HMLL14] uses physically-based hair simulations to generate a database of structurally plausible samples and improved control during hair digitization, which makes it suitable for animation purposes as shown in this work. While the acquisition of 3D hair from a single image [CWW*12, CWW*13, HMLL15, CLS*15, CSW*16] or depth sensor [HML*14] offers the most flexibility, they often rely on manual guidance and non-visible regions have to be estimated with data-driven techniques.

Dynamic Hair Capture. Techniques for acquiring complex hair movements through optical sensing were introduced to address the difficulty of tweaking simulation parameters and slow turnaround time of producing realistic hair animation. In analogy to body motion capture systems, Ishikawa and coworkers track reflective markers on a few hair strands to animate the full hairstyle by interpolating guide strand motions [IKSM07]. These markers can potentially interfere with the hair motion and only coarse hair dynamics can be recovered. The video-based technique of Yamaguchi et al. [YWO08], which extends the work of Wei et al. [WOQS05], can generate dynamic hair sequences that have consistent 2D orientations with input images from multiple views, but is limited to

straight hairstyles as it enforces temporal coherence of the shapes between frames. Dynamic hair capture has also been demonstrated on a single-view video input using optical flow-based feature tracking [CWW*13], but only simple hair motions can be obtained. Luo and colleagues [LLW*11, LLP*12, LZZR13] exploit the consistency of 2D orientation maps across multiple views to produce accurate and temporally smooth hair surfaces as an intermediate representation. Even though a wide range of highly complex hairstyles were captured, the resulting strands suffered from flickering artifacts since they were synthesized independently for each frame. Zhang and coworkers [ZTW*12] used a mass-spring simulation model to ensure the capture of temporally coherent hair motion, but most high-frequency dynamics are lost. Furthermore, simulation parameters are not optimized for the entire hairstyle, and therefore they cannot re-simulate the result directly and create new animations with different head motions or additional external forces. The state of the art work of Xu et al. [XWW*14] introduces a robust motion path analysis algorithm, which produces significantly more complex hair motions while ensuring temporal coherence. While impressive results were demonstrated, the recorded motions can only be replayed exactly as captured, and complex interactions with other objects (e.g., comb) are difficult to achieve. Consequently, the benefits of purely captured hair animations are still dubious in production.

Data-driven Simulation. Our work focuses on measuring simulation parameters from real-world dynamic hair performances and to produce new hair animations under different environmental conditions. Techniques for measuring simulation parameters have been developed in other domains, such as cloth [BTH*03, MBT*12, WOR11] and soft body objects [BBO*09, WWY*15]. When it comes to hair, Twigg and Kačić-Alesić [TKA11] introduced a method for static inversion of mass-spring systems, which optimizes parameters such as the spring rest length so that shapes represent the equilibrium when settled under gravity. Derouet-Jourdan and collaborators [DJBDDT13] proposed the first work to compute hair strand parameters based on the Super-Helices model [BAC*06]. Simulation parameters of hairstyles in static equilibrium are estimated subject to gravity and frictional contact for automatic simulation purposes. In addition to the assumption that hair parameters are homogeneous, these parameters cannot be properly estimated solely from static geometry, since different combinations of shape and physical properties could yield the same result under gravity (see Fig. 2). We propose to disambiguate these parameters by observing the hair in motion with a technique that can compute hair simulation parameters for a given simulation model by observing dynamic hair in the real world.

3. Capture and Preprocessing

Our goal is to capture the physical parameters of specific hairstyles by observing the dynamic motion in the real world. In this section we discuss our capture setup for acquiring video data, and describe the preprocessing steps required to obtain initial hair geometry for the first frame and extract the rigid head motion over time, which are both required for the dynamic optimization procedure described in Section 4. We also preprocess the input videos to obtain constraints for the dynamic optimization.

Data Acquisition. Our method requires calibrated video reference of dynamic hair motion. We have no specific constraints on the captured data, other than the hair should be well-lit and recorded from sufficiently many viewpoints to see the complete hairstyle, at a high-enough frame-rate to avoid severe motion blur. For the hairstyles in this paper we use a multi-view array of ten cameras recording at 35 frames-per-second, where two are dedicated to track the head motion and eight are spread around the back and sides of the hairstyle (see Fig. 3). The hair is evenly lit by LED lighting strips. Note that for validating our method on small examples of individual hair strands (as described in Section 6) we employed only two views.



Figure 3: Our capture setup consists of 8 video cameras spaced evenly around the hairstyle, and 2 additional cameras tracking head motion from front. The hair is uniformly lit using LED strips.

In addition to the video data we acquire a facial scan of the actor [BBS*10] and align a full head model to the scan in order to obtain a complete scalp for rooting the hair. We place a small number of markers (14) on the actor’s face in order to compute the per-frame motion of the head from the two front cameras. The actor maintains a neutral expression with closed eyes during acquisition.

Hair Initialization. Just as with traditional hair simulation for animation, our optimization method for simulation parameters requires the topology and initial geometry of the hair to be given as input. We therefore wish to reconstruct the hairstyle at the starting frame of the video, where the hair should be static. This is accomplished by starting with a multi-view reconstruction step to obtain a 3D point cloud [Agi16]. In practice, one frame of ten viewpoints with wide baselines is not sufficient to obtain a dense point cloud, so we begin each sequence by rotating the actor as if on a turntable to simulate many narrow baseline views, and then extract 100 reference views for reconstruction. From the point cloud data we recover the 3D hair geometry using the state-of-the-art hair capture method from multi-view images of Hu et al. [HMLL14], which provides plausible hair synthesis for a wide range of possible hairstyles. This geometry is then set as the initial geometry of the first frame. Since the hairstyle can have tens or even hundreds of thousands of strands, we follow the work of [DJBDDT13] as well as industry practice, and extract a subset of *guide strands* in order to decrease computation time. To this end, we group all the hair strands into

400 clusters via k-means clustering based on the root positions and strand shapes as in Wang et al. [WYZG09]. For each cluster we compute the center strand, which represents the shape of the cluster, as well as the attachment point to the scalp. Throughout this work we will simulate and optimize for only these 400 guide strands and generate final hairstyles by interpolating the in-between strands, a common practice in hair simulation. Fig. 4 shows the first-frame reconstructed hair geometry using Hu et al. [HMLL14] and a subset (100 for visualization) of the corresponding guide strands for two actors.

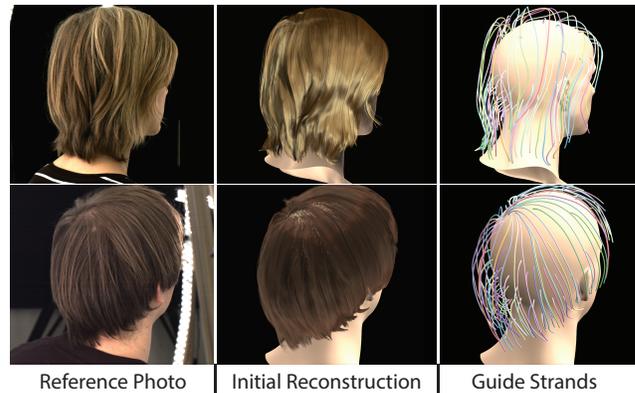


Figure 4: From reference photos (left) we reconstruct initial hair geometry (center) and extract a subset of guide strands (right).

Video Preprocessing. Our parameter optimization method will be guided by constraints from the input videos of the dynamic hair motion. To facilitate the optimization, we preprocess the videos to make constraint computation faster. We argue that the overall appearance of a dynamic hairstyle projected into a 2D video can be characterized by three main properties: (1) the orientation of hair strands, (2) the approximate silhouette of the hair volume, and (3) the general direction of motion of the strands from one frame to the next. We therefore precompute orientation maps in each frame of every video using oriented Gabor filters, which have been shown to work well for identifying hair orientations [BBN*12]. We choose the highest filter response under 16 orientations for each pixel. Silhouettes are extracted using Video Snapcut [BWSS09], which allows us to manually segment a small subset of key-frames and the silhouettes are automatically propagated over time. Approximate hair motion is computed using optical flow [BBPW04]. Since projected hair strands have high gradients orthogonal to the strand and are virtually indistinguishable along the strand, we use only the motion component that is orthogonal to the strand orientation as a constraint. An example orientation map, silhouette and motion field for one video frame are shown in Fig. 5.

Ultimately, the acquired data for our dynamic optimization includes a 3D head model with rigid motion per frame, initial 3D guide strands attached to the head in a starting static pose, and a multi-view array of orientation maps, silhouettes and perpendicular motion vectors for a dynamic hair sequence.



Figure 5: Our optimization will use cues from the video frames. The constraints include the 2D orientation of strands, the silhouette of the hairstyle volume, and approximate motion vectors.

4. Dynamic Parameter Optimization

We now describe our data-driven approach to optimize hair simulation parameters, by comparing simulated hair animations to observed hair motion.

4.1. Physical Hair Simulation

As mentioned previously, physical simulation is the method of choice for generating CG hair animations, and several hair models and simulation approaches have been proposed [BAC*06, BWR*08, SLF08], each with their own particular strengths. A key component of our work is that we design a framework for optimizing the parameters of virtually *any* hair model, by treating the simulation as an unknown function of a parameter set Φ . Our approach is independent of the simulator and even the actual parameters, as long as the resulting simulated hair can be evaluated as follows.

Let us define a generic hair strand as a 1D curve in space, discretely sampled at regular intervals along the curve. Each sample point \mathbf{x}_i contains an orientation \mathbf{o}_i , which represents the tangent to the curve at \mathbf{x}_i . Accounting for motion over time t , and combining samples from all hair strands, the complete hair animation can be characterized by $\mathcal{S} = \{\mathbf{x}_i^t, \mathbf{o}_i^t\}$. Our optimization method will aim to produce an optimal set of parameters Φ such that the simulator will generate a result \mathcal{S} that closely matches the observed hairstyle under the measured head motion. Our goal is similar to the work of Derouet-Jourdan et al. [DJBDDT13], but it is very important to note that we go beyond inverting static hairstyles by considering the dynamic motion of the hair in our optimization, which is essential for guaranteeing that the estimated parameters will remain faithful to the observed hairstyle under motion. Next, we describe the energy function that we minimize in order to achieve this goal.

4.2. Data-Driven Hair Evaluation

Given a simulated hair animation that corresponds to a captured video sequence, we can evaluate the suitability of the parameter set used for simulation by formulating an error function over all strands \mathcal{S} , given the video data constraints computed in Section 3. Note that our goal is not to obtain an exact strand-by-strand reconstruction of the observed performance such as Xu et al. [XWW*14], but rather obtain a feasible set of simulation parameters Φ that will allow plausible re-animations of the hair in different scenarios.

We formulate the difference between simulated and observed hair motion as an energy term that we minimize. Specifically,

$$E(\mathcal{S}) = \sum_{t=1}^{N_f} \sum_{v=1}^{N_v} E_{view}(\mathcal{S}, t, v), \quad (1)$$

where N_f is the number of video frames and N_v is the number of viewpoints. For each viewpoint v we evaluate the error for only the visible strand points $\{\tilde{\mathbf{x}}_i\}^t$ with corresponding orientations $\{\tilde{\mathbf{o}}_i\}^t$ at time t , which we denote as $\tilde{\mathcal{S}}^t$. For ease of notation, we omit the viewpoint index v . Visibility is computed by the simple heuristic of checking occlusion with the head model from the viewpoint. The error for a single viewpoint is thus defined as

$$E_{view}(\tilde{\mathcal{S}}^t) = E_{orient}(\tilde{\mathcal{S}}^t) + \lambda_1 E_{flow}(\tilde{\mathcal{S}}^t) + \lambda_2 E_{silh}(\tilde{\mathcal{S}}^t), \quad (2)$$

where the first term is an orientation term to ensure that the projected strand orientations at the current time follow the corresponding orientation map. This orientation error can be described as

$$E_{orient}(\tilde{\mathcal{S}}^t) = \sum_{i=1}^{|\tilde{\mathcal{S}}^t|} (1 - |\langle \tilde{\mathbf{o}}_i^t, \mathcal{O}^t(\tilde{\mathbf{x}}_i^t) \rangle|), \quad (3)$$

where $\tilde{\mathbf{x}}_i^t$ and $\tilde{\mathbf{o}}_i^t$ are the projections of the sample point $\tilde{\mathbf{x}}_i^t$ and orientation $\tilde{\mathbf{o}}_i^t$ onto the image plane, and \mathcal{O}^t is the orientation field computed in Section 3. Both orientation vectors are normalized, and $\langle \cdot, \cdot \rangle$ denotes the inner vector product. The second term in Eq. 2 is a flow term to ensure that the motion of the strand points follow the approximate motion field \mathcal{F}^t , also from Section 3, and is defined as

$$E_{flow}(\tilde{\mathcal{S}}^t) = \sum_{i=1}^{|\tilde{\mathcal{S}}^t|} \left\| \left(\tilde{\mathbf{x}}_i^t - \tilde{\mathbf{x}}_i^{t-1} \right) - \mathcal{F}^t(\tilde{\mathbf{x}}_i^t) \right\|^2. \quad (4)$$

Finally, the third term is a silhouette term, which constrains the projection of the strands to fill the hair region of the image, since the overall volumetric shape of the hairstyle in motion is an important cue for its appearance. The silhouette term is defined over all silhouette pixels $\{s_i\}^t$ in the image, and attempts to minimize the distance to the closest projected hair strands as

$$E_{silh}(\tilde{\mathcal{S}}^t) = \sum_{s \in \{s_i\}^t} \min_{\tilde{\mathbf{x}} \in \tilde{\mathcal{S}}^t} \|s - \tilde{\mathbf{x}}\|^2. \quad (5)$$

Note that this term does not prevent the hair from extending beyond the silhouette, but in that case the orientation term (Eq. 3) will be very poorly satisfied for those strands.

The result of this evaluation is an error metric that quantifies how well the simulated dynamic hairstyle \mathcal{S} matches the captured hair animation. Since the flow term and the silhouette term are measured in pixels, their influence can be independently down-weighted through λ_1 and λ_2 , respectively, to better match the orientation term, which is constructed as an inner product. In our examples shown later, we operate on IMP video frames and use $\lambda_1 = 0.01$ and $\lambda_2 = 0.01$.

4.3. Optimization

Given a captured hair sequence and the error function described in Section 4.2 we can now optimize for the simulation parameters Φ that will yield a hairstyle animation that closely matches the input, and also allow the hairstyle to be simulated for new animations.

Inverting a dynamic hair simulation directly is a daunting task, as the complex interplay of hair motion including collisions, contacts and friction makes the problem highly non-linear and non-continuous. Furthermore, there is no such thing as a single best hair simulation model, and over the past years different models with different properties and simulation parameters have evolved. In order to cater to as many simulation methods as possible, we leverage particle swarm optimization (PSO) [KE95], a general computational method that iteratively solves optimization problems with respect to an objective function. The main benefit of using PSO for this task is that it is a form of “black-box” optimization, in that PSO is oblivious to the actual function being optimized. The overall approach of PSO is to maintain a population of candidate solutions, named *particles*, and iteratively evolve the particles in the parameter search space towards other particles that found better solutions to the objective function. After a fixed number of iterations, the best fitting particle (i.e. the one with lowest objective error) is chosen as the solution to the optimization.

In our solution, each particle is a simulator that initializes a physical model given the 3D reconstructed hair from Section 3, and generates a hair animation \mathcal{S} given the captured head motion and the current parameter guess Φ . The result of the optimization will be the physical parameters that minimize our objective. Using this approach is very powerful, as it allows us to optimize for any simulation model, with any set of physical parameters. Furthermore, if the simulation models hair-hair interaction, such as friction, the optimization will also find suitable parameters for the inner fully occluded hair strands, since every strand indirectly influences the overall perceived motion. The obvious drawback over optimization methods that compute analytic derivatives is convergence speed and hence computation time, as many simulation runs may be required to achieve optimal parameters. We will assess convergence behaviour in Section 6 and discuss computation time in Section 7.

5. Application to Simulation Models

To demonstrate the effectiveness and universality of our optimization procedure, we show its application to two well-known state-of-the-art hair simulation models; the *super-helices* model of Bertails et al. [BAC*06] in Section 5.1, and the *discrete elastic rods* model proposed by Bergou et al. [BWR*08] in Section 5.2. Once optimal simulation parameters are found, they can be used to simulate the hairstyle under any novel input animation. In Section 5.3 we describe how the simulated guide strands are interpolated to a full hairstyle, which is then rendered to generate the final visuals.

5.1. Super-Helices

In this section we first give an overview of the *super-helices* model of Bertails et al. [BAC*06] and then show how to use it with the proposed dynamic parameter inversion.

Model Description. The super-helices model is defined for a single hair strand, which is divided into N helical segments. Each segment is parameterized by three values: twist (κ_0) and two orthogonal curvatures (κ_1 and κ_2). Thus the entire strand has $3N$ parameters denoted as \mathbf{q} , the *generalized coordinates* of the strand. Given coordinates $\mathbf{q}(t)$ for the strand over time t , it is possible to reconstruct the full animation of the strand.

To simulate a strand physically and generate an animation $\mathbf{q}(t)$, several mechanical properties are required. The first of which are the values of \mathbf{q} under no external forces, deemed \mathbf{q}^n and called the “natural curvatures and twist”. These $3N$ parameters define the curliness of the strand, where values of zero yield straight hair, and non-zero values give more wavy or curly strands. In addition, the mechanical properties include the bending stiffness for both curvature directions and the torsional stiffness, typically expressed as two scalar values (known as Young’s modulus and Poisson’s ratio). While the original model defines uniform stiffness over the entire strand to avoid lengthy parameter tuning, we can allow per segment stiffnesses in order to, for example, model hair that becomes frizzier towards the tips, since our values will be optimized automatically, yielding an additional $2N$ parameters. Finally, each strand contains an additional parameter defining the volumetric mass ρ , the cross section of the strand given by two radii η_1 and η_2 , a viscous air drag coefficient ν , and an internal friction coefficient γ_1 , all of which are constant across the segments of the strand. In addition to the original super-helix model we add a per-strand hair-hair friction coefficient γ_2 , a hair-head friction coefficient γ_3 , and a cohesion coefficient τ that controls how closely the velocities of nearby guide strands should match. In our implementation, we use a more approximate and simpler model without Coulomb friction compared to Derouet-Jourdan et al. [DJBDT13] to decrease computation time (please refer to Choe et al. [CCK05] for more details of these hair interaction forces). We found that these additional parameters give the simulator more flexibility to match observed real world hair. In total, the complete hairstyle is governed by $(5N + 8)G$ parameters, where G is the number of guide strands.

A strand is simulated by solving the equations of motion of a super-helix to get $\mathbf{q}(t)$ given the mechanical parameters, a starting point $\mathbf{q}(0)$ with corresponding head geometry and root attachment points, the time-varying head motion, and the known gravity direction. Elaborating further on the model, the equations of motion or the simulation approach is beyond the scope of this paper, and we refer to Bertails et al. [BAC*06] for more details.

Model Initialization. So far we have captured the static hair of the first video frame as a set of piecewise linear segments representing the G guide strands, together with the initial head geometry. In order to initialize the simulation model we have to convert the guide strands into super-helices. We use the floating tangent algorithm introduced by Derouet-Jourdan et al. [DJBDT13], which approximates each strand by a smooth super-helix consisting of N segments. For the datasets in this paper we use $N = 5$ for short hairstyles, and increase the number of segments up to $N = 10$ for longer, curlier hairstyles. The resulting super-helices define the configuration of \mathbf{q} , i.e. the topology of the hair, and specifically they represent $\mathbf{q}(0)$, the starting point of our simulation. Note that

the natural curvatures and twist \mathbf{q}^n are still unknown, as the captured guide strands are reconstructed under gravity and contacts.

To finish the initialization we compute the root attachment points of the strands on the head model, plus define a coordinate frame for the first segment of each super-helix at each attachment point. We set the gravity vector based on the initial head position.

Parameter Optimization. We are now ready to optimize the physical simulation parameters. Our parameters include the Young's modulus per strand segment, as well as the mass ρ , the hair-hair and hair-head friction coefficients γ_2 and γ_3 , and the cohesion coefficient τ per guide strand, yielding a parameter vector $\Phi \in \mathbb{R}^{(N+4)G}$. We fix the Poisson's ratio (0.48), the two radii η_1 ($5e^{-5}m$) and η_2 ($5e^{-5}m$), as well as internal friction γ_1 ($1e^{-10}kg \cdot m^3 \cdot s^{-1}$) and air drag ν ($1e^{-5}kg \cdot (m \cdot s)^{-1}$) for all hairstyles as suggested by Bertails et al. [BAC*06]. The natural curvatures and twist \mathbf{q}^n will be solved for as described below.

For our PSO-based optimization we initialize a swarm of particles, each one a thread that will run an instance of the simulation with a different guess of parameters Φ^* . Before simulating, each particle must compute \mathbf{q}^n , which can be determined given the static hairstyle of the first frame $\mathbf{q}(0)$ and the parameters Φ^* through static inversion. Similar to the recent work on static hair inversion [DJBDT10, DJBDT13] we estimate the \mathbf{q}^n that produces $\mathbf{q}(0)$ under known gravity vector, hair-head collisions and hair-hair interactions. Instead of optimizing the interaction forces as well [DJBDT13], we pre-compute them by using the explicit hair interaction models with γ_2 and γ_3 .

The particle now has everything required to simulate the hairstyle given the sequence of tracked head motion. The result is a set of configurations $\mathbf{q}(t)$, which can readily be converted to our representation of a hairstyle animation \mathcal{S} by sampling the reconstructed super-helices at regular intervals. We empirically found that a spacing of 3mm per sample provided sufficient points to evaluate the animation, and adding more samples only increased computation time. The particle finally determines the suitability of the parameter set Φ by evaluating Eq. 1. Given the residuals of all particles, the particles are re-initialized with a new set of parameters and the process is repeated using traditional particle swarm optimization [KE95].

For the results in this paper we use 31 particles, optimizing over 50 video frames for 400 iterations. To accelerate the convergence of our algorithm, we employ a two-step method. In the first 200 iterations, we estimate the Young's modulus as a uniform value per strand in order to reduce the dimensionality of the problem. In the next 200 iterations we relax this constraint to allow Young's modulus to vary along the strand, using the estimated uniform value as the initial guess. Allowing the Young's modulus to vary spatially is beneficial to account for the fact that a single guide strand represents a collection of fibers of different length. The fitted parameters produce a plausible match to the observed hairstyle in motion, and can be immediately used for additional simulations with different external forces, as we will show in Section 6.

5.2. Discrete Elastic Rods

As a second example, we show how our dynamic inversion strategy can be applied to the *discrete elastic rods* model of Bergou et al. [BWR*08]. Since the proposed dynamic inversion strategy is agnostic to the underlying simulation method, optimal simulation parameters can be found in the same way as for the super-helices model, even though the two models are conceptually very different (i.e. one is continuous the other discrete). The major difference between these two models are the number of parameters to be optimized and the static inversion problem. We will first briefly describe the discrete elastic rods model and then elaborate on these differences.

Model Description. A discrete rod (hair strand) consists of a centerline r with n vertices x_0, \dots, x_{n-1} and $n-1$ straight edges. Each rod has $4n-1$ degrees of freedom (DOF) ($3n$ for vertices, and $n-1$ for the angle of rotation θ relative to the natural Bishop frame for each edge). As internal forces we consider twisting, bending, as well as stretching, which leaves each segment with four physical parameters: twist (κ_0), two orthogonal curvatures (κ_1 and κ_2) and rest length (l_0). To simulate a strand animation $r(t)$, we first compute all the internal and interaction forces as well as any external forces acting on the rod. Then, by solving for the velocity of each vertex and the rotation angle at each segment, we can update the strand shape in the next time step. For more details on the model, we would like to refer the reader to Bergou et al. [BWR*08].

Model Initialization. The discrete representation of hair strands as piecewise linear curves renders model initialization straightforward. In our implementation, we resample each hair strand into 1cm long segments. As in the previous model, we only consider guide strands for simulation, yielding a total G rods to be simulated for the full hairstyle.

Parameter Optimization. Our optimization parameters for *discrete elastic rods* include the Young's moduli per five strand segments, as well as the mass ρ , the hair-hair and hair-head friction coefficients γ_2 and γ_3 , and the cohesion coefficient τ per guide strand as detailed above, yielding a parameter vector $\Phi \in \mathbb{R}^{((n-1)/5+4)G}$. Since the rods have many more segments than the super-helices ($n \gg N$), we chose to estimate the Young's modulus over 5 segments, yielding a similar discretization.

Static inversion takes place the same way as described above for the super-helices model, with the difference that now the stiffness matrix is banded. We solve for κ_0 , κ_1 , κ_2 and l_0 for each segment to ensure the hairstyle is static under gravity in the absence of any other external forces.

5.3. Hair Interpolation and Rendering

Throughout simulation and optimization we follow the standard practice of reducing computation time by operating only on guide strands, which represent just a subset of the hairstyle. In order to create realistic complete hairstyles we return to the dense set of captured strands reconstructed by Hu et al. [HMLL14] in Section 3, represented as piecewise linear hair strands \mathcal{H} , and we also return to the piecewise linear strands \mathcal{G} from which the super-helices and

discrete rods were generated. The simulation provides the motion of the guide strands in \mathcal{G} , and these strands will be used to interpolate the motion of the rest of the hairstyle \mathcal{H} . Since the time-varying head position is known and the root points of all hair strands as well as the initial coordinate frame of the first segments remain fixed, obtaining per-segment coordinate rotations is sufficient to reconstruct all the hairs in \mathcal{H} . Thus, we aim to interpolate rotations from the guide strands. We adopt a linear blend skinning approach [LCF00] for the interpolation, which is similar to the method of Chai et al. [CZZ14]. For each \mathcal{H}_i in the initial video frame we find its k_0 nearest guide strands $\{\mathcal{G}_i\}$ and use them as skeletons, where each edge segment between two vertices is considered as a bone. For each edge segment ℓ_j in \mathcal{H}_i we search for the k_1 nearest bones among all $\{\mathcal{G}_i\}$ and compute skinning weights based on the distances. Then, for every point in time, we blend the rotations of the k_1 bones according to the weights to get the rotation of ℓ_j . The final hairstyle is reconstructed by following the segments in order from root to tip and applying the rotations per segment, and then rendered using the method of Marschner et al. [MJC*03]. We use $k_0 = 2$ and $k_1 = 4$ for all results in this paper.

As a final caveat, after generating the simulation results using the optimized parameters it is sometimes apparent that the initial static reconstruction of Hu et al. [HMLL14] can create some hair strands that are actually longer than the real counterparts. Identifying the exact strand lengths from hair at rest is a very challenging problem, but when animated the difference in length becomes more visible. For this reason, after observing the simulated result we typically return to the initial static hair reconstruction and manually apply a “virtual haircut” to shorten strands whose excessive length only became apparent during motion. After cutting, we re-run the optimization to ensure we obtain the optimal parameter set.

6. Results and Evaluation

We start by evaluating convergence properties and performance of the proposed algorithm, first on synthetic data, followed by experiments on real single hair strands. Then we show results on full hairstyles for two different subjects, with different hair lengths. In this first part, results are computed using the super-helices model such that they can be easily compared to one another for analysis. In a second part we then compute results with both models to show the generality of the approach, and how it enables the simulation methods to optimally approximate the captured hair animation within their respective limitations. Our results demonstrate that the method succeeds at recovering simulation parameters from captured input videos, which not only resemble the captured animation but allow to simulate the hairstyle under any novel input animation, including external forces such as wind.

Synthetic Validation. We validate the proposed method using a synthetic hairstyle, which gives us ground-truth geometry and allows us to quantitatively assess the performance. Also, in this case we know that the animation can be matched by the simulation, which is not always the case for real data where the simulation method only approximates the many effects that influence the hair. To this end we simulate and render a short animation of 200 frames using $G = 100$ guide strands. The rendered images then serve as the only input to our method, as if we were to operate on captured

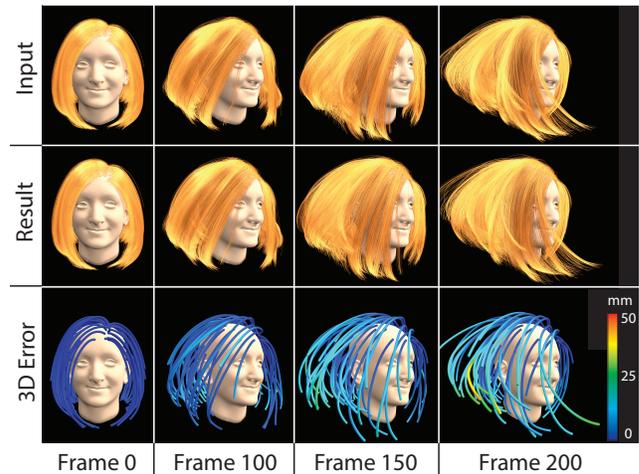


Figure 6: We validate our method on a synthetic dataset and show that we can accurately recover simulation parameters that reproduce the same input animation. The top row shows the input images to our optimization, the middle row shows a rendering of a simulation with the recovered parameters, and the bottom row shows the 3D error color coded from blue (0mm) to red (50mm).

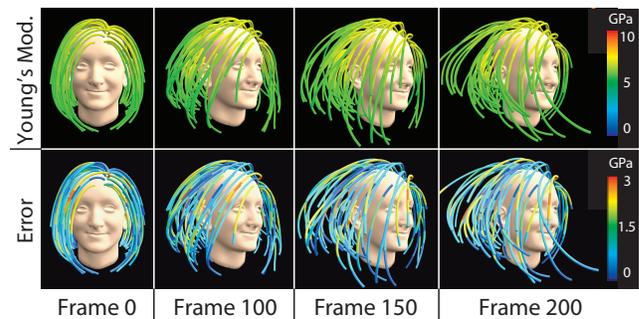


Figure 7: Here we show the input Young’s modulus (top) for the synthetic dataset in Fig. 6, color coded from blue (0GPa) to red (10GPa), and we visualize the error of our recovered Young’s modulus (bottom) which is on a scale of 0 to 3GPa, as illustrated by the color bars.

imagery. As can be seen in Fig. 6, the re-simulated result with the estimated parameters (middle row) resembles the ground-truth input (top row) very well. This is corroborated by the Euclidean error computed on the guide strands and visualized in the bottom row. In Fig. 7 we visualize the error between the original and recovered Young’s moduli and show that overall our method finds reasonable parameters and not just parameters that yield a similar motion. Since our measurement is in a 2D scenario, our estimated parameters are only accurate with respect to projection of the hairstyle in the rendered images and as a consequence, the recovered parameters are not exactly identical to ground truth.

Convergence Analysis. In order to gain some intuition on the error surface of the objective function, and how suited the chosen optimization strategy is to find its minimum, we conduct a

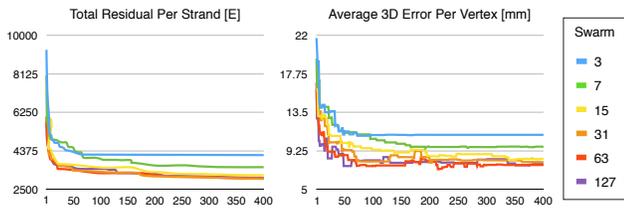


Figure 8: We assess the convergence of our algorithm on the synthetic dataset by varying the swarm size, i.e. the number of particles. We first generate 127 random parameter vectors, and for each test that involves k particles, we pick the first k vectors as initial values. The left plot shows the total energy residual per strand over all 200 frames, and the right plot the average 3D distance to ground truth per vertex. The two are clearly correlated, which means that minimizing the proposed objective function indeed yields similar 3D hair motion. Furthermore, the plots show that swarms with more than ~ 30 particles exhibit similar convergence properties, which is why we use 31 particles in our experiments.

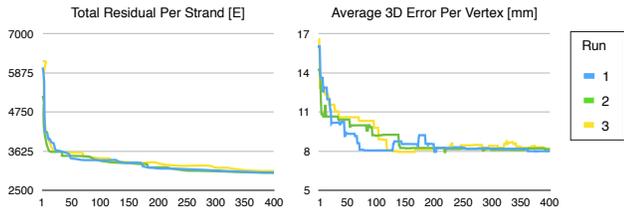


Figure 9: We test the convergence of our algorithm w.r.t. different initial values using the synthetic dataset. Each run consists of 31 particles with random initial parameter values and all successfully converge to similar low energy states. This indicates that the energy landscape is suited for PSO optimization.

series of convergence experiments on the synthetic data described above. The first experiments investigate the influence of the swarm size (number of particles) on convergence. As can be seen from Fig. 8 small swarms converge slowly and fail to find a low minimum. Swarms with more than ~ 30 particles perform similarly well, which is why we use 31 particles for optimization. The second batch of experiments gives an intuition of the smoothness of the objective function. Following from the previous experiments, we randomly initialize three different swarms of 31 particles each. As can be seen in Fig. 9, all three swarms exhibit very similar convergence behaviour and converge to similar low-energy states, which is a strong indicator that the objective function is well suited for the proposed optimization method. Finally, in Fig. 10 we investigate how close the estimated parameters resemble the input parameters. As can be seen, the parameter residual is reduced during optimization which indicates that there is a strong correlation between parameters and objective residual. This would for example be less obvious if the parameters would be highly redundant since then very different parameters could produce similar output animations.

Single Strand Experiments. Adding a level of complexity over synthetic validation, we conduct a series of experiments on sin-

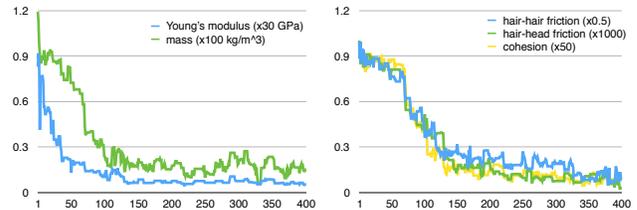


Figure 10: These two plots show the progression of the parameter vector residual during optimization. As can be seen, the residuals of all parameters are being reduced during optimization, which indicates that the optimization indeed gets close to the global optimum. Note that the curves show the averaged convergence over all three runs displayed in Fig. 9 and that the y-axis has been scaled per parameter as indicated in brackets.

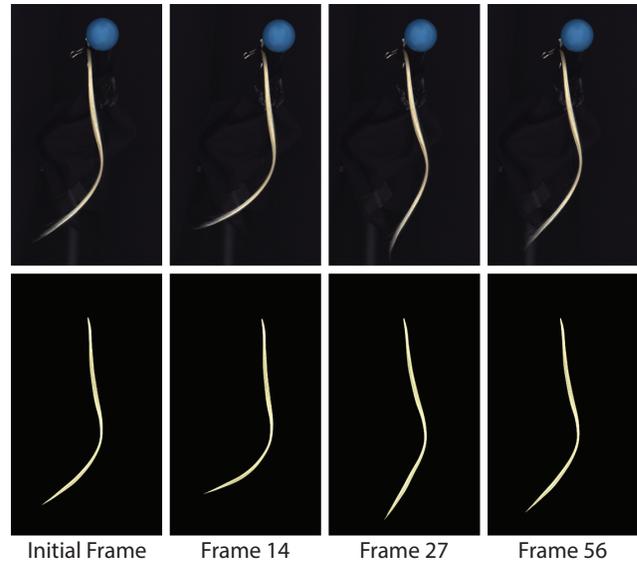


Figure 11: The top row shows input frames used to estimate the simulation parameters, with which the strand is simulated (bottom row). As can be seen, the simulation results closely match the input.

gle, real hair strands. While the simulation method may already not model all the physical effects that govern the animation, at the very least we can ensure that the initialization is accurate and we know the exact motion at the root of the strand, both of which will be less accurate for the full hairstyle. Fig. 11 shows our result for a single hair strand. Starting from the initial frame on the left, the method recovers parameters that yield a very similar motion of the strand when re-simulated (bottom row). We estimate the motion of the root by tracking the position of the blue ball.

The main advantage of our technique over pure capture methods is that the recovered parameters can now be used to simulate the strand under any novel animation. Fig. 12 shows a novel input animation on top, and our simulated result at the bottom, this time given only the motion of the blue ball as input. As can be seen, the simulation closely resembles the reference images, even though the

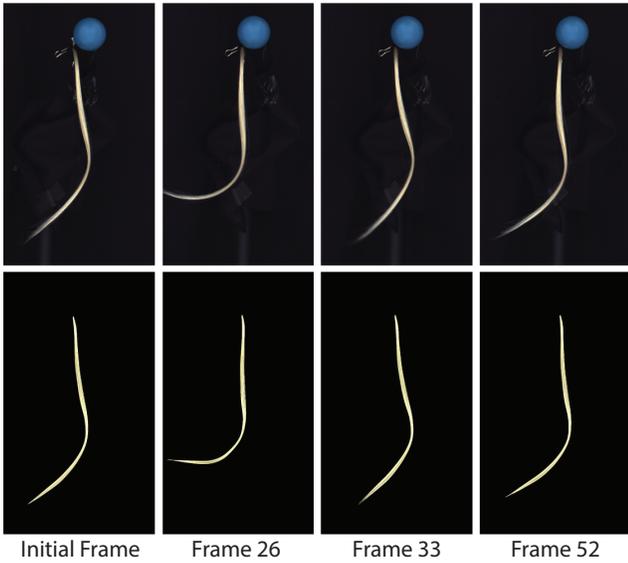


Figure 12: To validate that the properties estimated in Fig. 11 are meaningful in that they allow the simulation to extrapolate to novel animations, we simulate the strand under a different animation. The simulation again visually matches the reference frames very closely.

parameters were retrieved from a different clip and we only use the motion of the root as input.

Finally, we show that we can recover different physical parameters for hair that has been treated with different products. Fig. 13 shows three strands with different properties; treated with a stiffening varnish (left), wet with water (center) and dry (right). The method successfully retrieves these properties and can thus capture the physical appeal of hair, as demonstrated in Fig. 14, where one strand is simulated three times with the different properties extracted with the three different treatments. To better show the difference, we overlaid the three strands in pink (varnish), blue (wet), and yellow (dry). It is clearly visible how the varnish hair for example is much stiffer and exhibits a very different motion style.

Full Hairstyle Results. Lastly, we show results on complete hairstyles for four different subjects with hair of different length. For all subjects we use $G = 400$ guidestrands. Fig. 15 shows input images and simulation results using the super-helix model for a short and medium-length hairstyle and Fig. 16 for a long hairstyle using discrete elastic rods. The simulation parameters are estimated from the first 50 frames only, and extrapolation to subsequent frames additionally validates the recovered parameters. Even though the two simulation models are fundamentally different, the proposed framework succeeds at finding suitable parameters for both since it is agnostic to the actual simulator used.

In Fig. 17 we recaptured the shorter hairstyle after applying hair product to it. Specifically, we applied hair spray to the right half of the head, which increases stiffness in that area. As can be seen, our method recovers the fact that the two sides exhibit different stiffness, which can be seen on the right where we visualize the

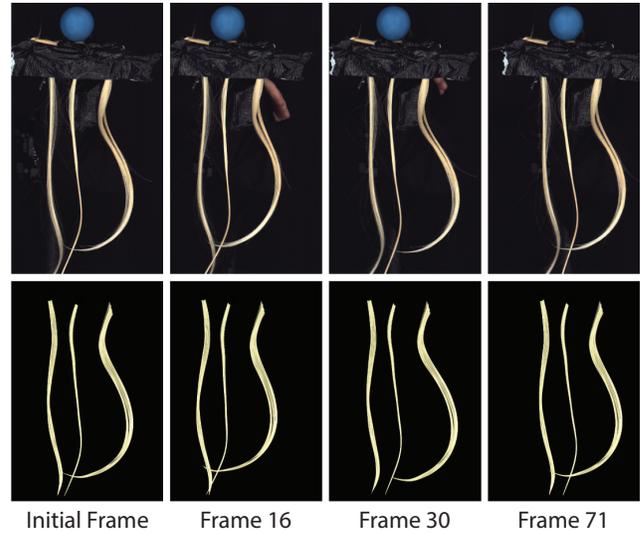


Figure 13: This figure shows three different strands that exhibit different physical properties (from left to right: with varnish, wet, and dry). The simulations with the estimated parameters shown in the bottom row match the input images.



Figure 14: Applying the parameters estimated from the strands with different physical properties (Fig. 13) to the same strand clearly shows how the physical properties influence the style of the animation and that we can successfully capture such properties. For visualization purposes, the three simulation runs are overlaid in pink (varnish), blue (wet), and yellow (dry).

estimated Young's modulus on the guide strands, the parameter that represents stiffness.

As discussed earlier, the main advantage of the proposed approach over pure capture methods is that the recovered parameters may be used to simulate the hair under novel input head animations and external forces as shown in Fig. 18, where we apply a wind force to the actors after optimizing for the hair properties. In the top row, we show the actor with longer hair, and in the bottom row we show two versions of the actor with shorter hair, the one with firming hair product on the side of his head and one without. Here we can clearly see the benefit of capturing the physical parameters that match the specific hairstyle of the actor.

The ability to estimate hair properties and apply them to novel animations is of central importance for practical applications, but of course the accuracy at which an animation can be repro-

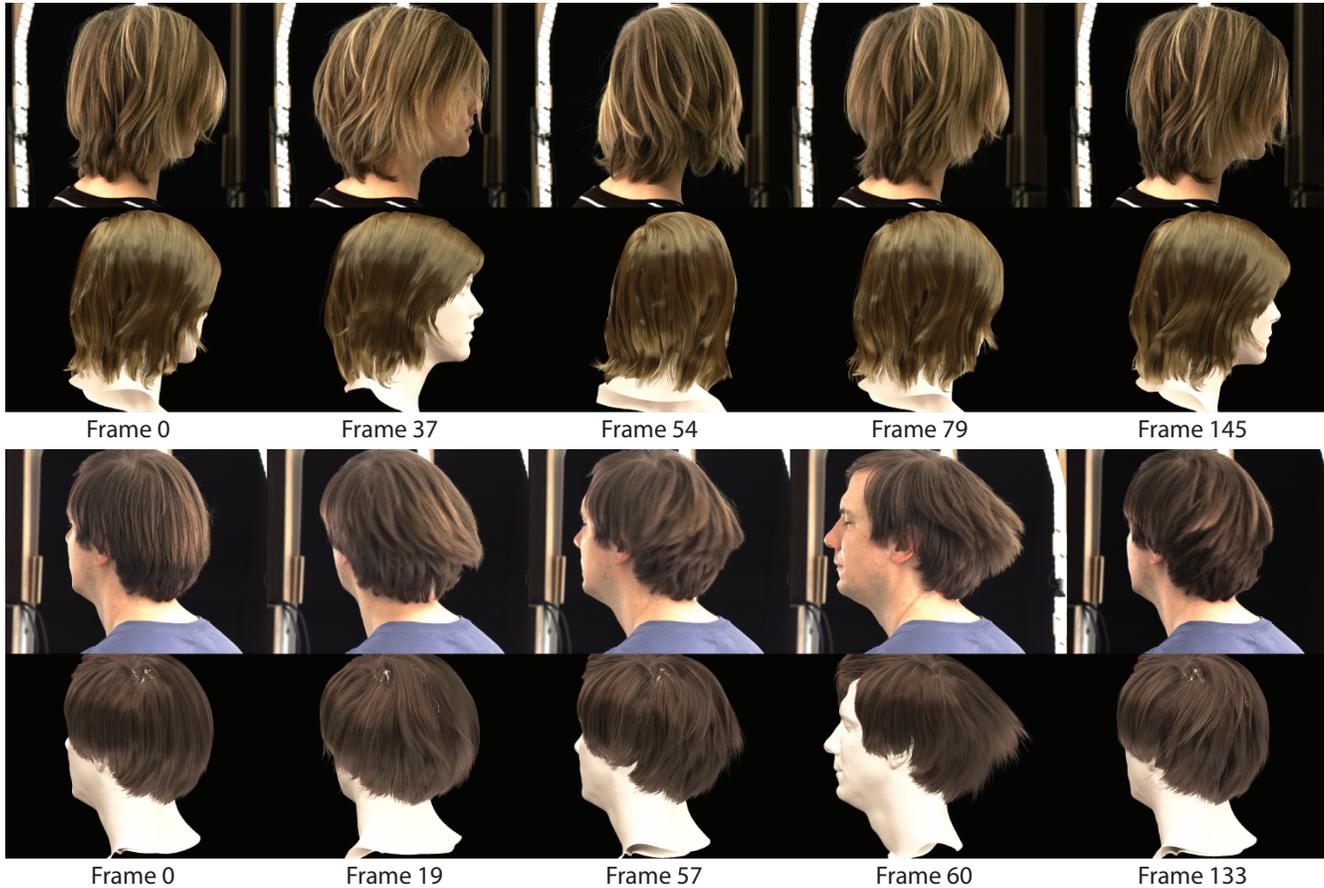


Figure 15: We show simulations with the super-helix model using the physical parameters that resulted from our dynamic optimization. Note that we optimized over the first 50 frames of each sequence, and the remaining frames additionally validate that our parameters are plausible.

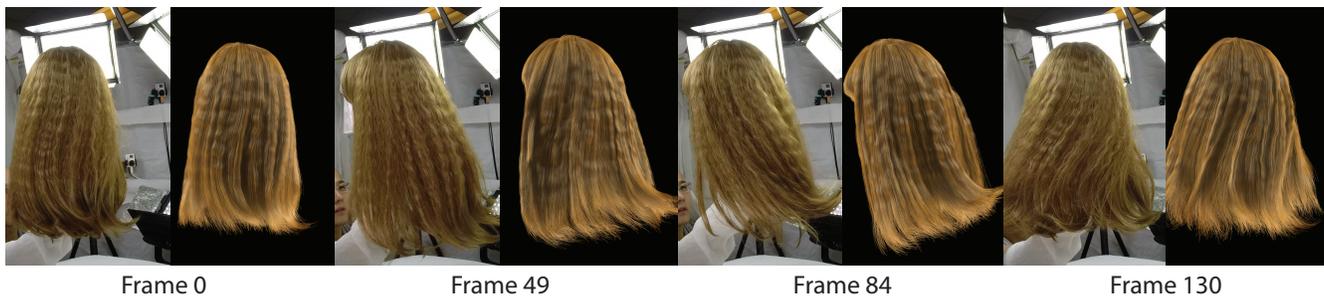


Figure 16: This figure shows results using the discrete elastic rods model. Just as in Fig. 15, simulation parameters are estimated from the first 50 frames only and the remaining frames additionally validate that the recovered parameters yield plausible simulations.

duced depends on the capabilities of the simulation method. When comparing to the state-of-the-art hair capture method of Xu et al. [XWW*14] (Fig. 19), it is obvious that the chosen simulation method cannot reproduce the animation at the same level of accuracy. Thus, if simply playing back a captured animation is desired, then dense hair capture methods (i.e. Xu et al. [XWW*14]) are currently the best option. However, since the proposed optimization framework is agnostic to the actual simulator, employing a more

sophisticated simulator in the future would yield improved reproduction of the hairstyle.

This dependency on the simulator is nicely shown in Fig. 20 where there are clearly small differences between the two simulation models employed. Overall, however, both resemble the input motion well since our optimization strategy will determine parameters such that the resulting simulation matches the input animation as closely as possible globally.



Figure 17: Here, the right half of the hairstyle has a firming hair product, and our method is able to determine a largely different bending stiffness (Young's modulus), as visualized on the right. Young's modulus on the left side stays within [1, 10GPa], which has the same order of magnitude as real hair, while the right side has much larger values because of the firming hair product.



Figure 18: After acquiring parameters we can simulate with additional external forces, and the simulated hair will remain faithful to the captured actor. The top row shows three animation frames from one actor with strong wind. The bottom row shows that the same wind force applied to the short-haired actor with and without hair product gives very different (and expected) behavior.

7. Conclusion

We present a new framework for simulation-ready hair capture - a method to automatically estimate dynamic hair simulation parameters by observing real hair motion. The resulting parameters can be used to re-simulate the captured hair, and also to edit the animation by changing the head motion, the physical properties of the hair such as wetness, and external forces such as wind.

In comparison to current state-of-the-art methods for parameter estimation of hair, ours is the first to consider its dynamics and fit parameters to captured hair in motion, which, as we have shown, is essential for obtaining reliable parameters. The suggested dynamic inversion is agnostic to the underlying simulation method, which we demonstrate using two state-of-the-art simulation models, super-helices [BAC*06] and discrete elastic rods [BWR*08]. We show that our method can estimate parameters for both models, such that the simulated results closely resemble target real-world hairstyles, even though the models are fundamentally different. Our method also has potential for mass-spring simulation

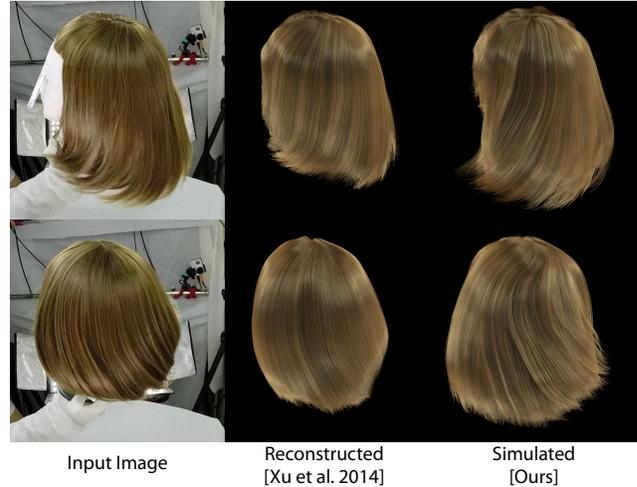


Figure 19: Our method can be applied to other hair capture setups, such as the input of Xu et al. [XWW*14], which was recorded for space-time reconstruction. Our simulated hair has comparable motion but can be completely edited or re-animated after capture.



Figure 20: Our dynamic inversion can be readily applied to other hair simulation models. Here we show an application to the super-helices model [BAC*06] (middle row) and the discrete elastic rods [BWR*08] (bottom row) on the same input data (top row). Since the two models have different capabilities, there are clear differences in the results, but overall the simulations match the input animation well since our optimization finds optimal parameters for both, such that they match the real world hairstyle.

models [SLF08]. We would like to explore the problem of static inversion for mass-spring systems and incorporate this model into our pipeline in the future.

The main limitation of our method is computation time, since a full dynamic simulation must be solved for every particle evaluation, over several particles, across many iterations. In practice, this was accomplished on a compute cluster using 480 cores (30 nodes

for the particles with 16 cores each for multi-threaded simulation). Still, optimizing over 50 frames for 200 iterations of PSO requires approximately 40 hours. Improving the convergence time of this approach is a primary topic of future work. As we have the ability to collect accurate 3D hair motion and their simulation parameters at scale, we wish explore data-driven and learning algorithms to predict these parameters directly.

We note again that our goal is not to recover the strand-by-strand reconstruction of the observed hair motion, and so the re-simulated animations with the optimized parameters may not perfectly match the inputs, in particular if the hair configuration goes outside of the simulation model that is used. However, the resulting hair properties computed in this work recover plausible motions and allow for artistic editing as mentioned above, and our framework can be immediately used to estimate parameters for even more sophisticated simulation models as they are developed in the future. A key feature of our framework is that it is generic, and can be applied to different physical models independent of the actual simulator. As such, this work represents a major step towards practical hair modeling and animation.

Acknowledgements

We wish to thank Xin Tong and Zexiang Xu for sharing their dynamic hair capture data. We would also like to thank all of our actors, who made this work possible.

References

- [Agi16] AGISOFT: Agisoft, 2016. <http://www.agisoft.com/>. 4
- [BAC*06] BERTAILS F., AUDOLY B., CANI M.-P., QUERLEUX B., LEROY F., LÉVÊQUE J.-L.: Super-helices for predicting the dynamics of natural hair. *ACM Trans. Graph.* 25, 3 (2006), 1180–1187. 2, 3, 5, 6, 7, 12
- [BAV*10] BERGOU M., AUDOLY B., VOUGA E., WARDETZKY M., GRINSPUN E.: Discrete viscous threads. *ACM Trans. Graph.* 29, 4 (2010). 3
- [BBN*12] BEELER T., BICKEL B., NORIS G., MARSCHNER S., BEARDSLEY P., SUMNER R. W., GROSS M.: Coupled 3d reconstruction of sparse facial hair and skin. *ACM Trans. Graph.* (2012). 3, 4
- [BBO*09] BICKEL B., BÄCHER M., OTADUY M. A., MATUSIK W., PFISTER H., GROSS M.: Capture and modeling of non-linear heterogeneous soft tissue. *ACM Trans. Graph.* 28, 3 (2009). 3
- [BBPW04] BROX T., BRUHN A., PAPPENBERG N., WEICKERT J.: High accuracy optical flow estimation based on a theory for warping. In *ECCV* (2004), pp. 25–36. 4
- [BBS*10] BEELER T., BICKEL B., SUMNER R., BEARDSLEY P., GROSS M.: High-quality single-shot capture of facial geometry. *ACM Trans. Graphics (Proc. SIGGRAPH)* (2010). 4
- [BTH*03] BHAT K. S., TWIGG C. D., HODGINS J. K., KHOSLA P. K., POPOVIĆ Z., SEITZ S. M.: Estimating cloth simulation parameters from video. In *Proc. SCA* (2003), pp. 37–51. 3
- [BWR*08] BERGOU M., WARDETZKY M., ROBINSON S., AUDOLY B., GRINSPUN E.: Discrete elastic rods. *ACM Trans. Graph.* 27, 3 (2008), 63:1–63:12. 2, 3, 5, 6, 7, 12
- [BWSS09] BAI X., WANG J., SIMONS D., SAPIRO G.: Video snapshot: Robust video object cutout using localized classifiers. *ACM Trans. Graph.* 28, 3 (2009). 4
- [CCK05] CHOE B., CHOI M. G., KO H.-S.: Simulating complex hair with robust collision handling. *SCA '05*, pp. 153–160. 6
- [CLS*15] CHAI M., LUO L., SUNKAVALLI K., CARR N., HADAP S., ZHOU K.: High-quality hair modeling from a single portrait photo. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34, 6 (2015). 3
- [CSW*16] CHAI M., SHAO T., WU H., WENG Y., ZHOU K.: Auto-hair: fully automatic hair modeling from a single image. *ACM Trans. Graphics* 35, 4 (2016). 3
- [CWW*12] CHAI M., WANG L., WENG Y., YU Y., GUO B., ZHOU K.: Single-view hair modeling for portrait manipulation. *ACM Trans. Graph.* 31, 4 (2012), 116:1–116:8. 3
- [CWW*13] CHAI M., WANG L., WENG Y., JIN X., ZHOU K.: Dynamic hair manipulation in images and videos. *ACM Trans. Graph.* 32, 4 (2013), 75:1–75:8. 3
- [CZZ14] CHAI M., ZHENG C., ZHOU K.: A reduced model for interactive hairs. *ACM Trans. Graph.* 33, 4 (2014). 8
- [DBDB11] DAVIET G., BERTAILS-DESCOUBES F., BOISSIEUX L.: A hybrid iterative solver for robustly capturing coulomb friction in hair dynamics. In *Proc. SIGGRAPH Asia* (2011). 3
- [DJBDDT13] DEROUET-JOURDAN A., BERTAILS-DESCOUBES F., DAVIET G., THOLLOT J.: Inverse dynamic hair modeling with frictional contact. *ACM Trans. Graph.* 32, 6 (2013). 2, 3, 4, 5, 7
- [DJBDDT10] DEROUET-JOURDAN A., BERTAILS-DESCOUBES F., THOLLOT J.: Stable inverse dynamic curves. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 29, 6 (2010), 137:1–137:10. 7
- [DJBDDT13] DEROUET-JOURDAN A., BERTAILS-DESCOUBES F., THOLLOT J.: Floating tangents for approximating spatial curves with g1 piecewise helices. *Comp. Aid. Geom. Des.* 30, 5 (2013), 490–520. 6
- [FWTQ07] FU H., WEI Y., TAI C.-L., QUAN L.: Sketching hairstyles. In *Proc. Sketch-based Interfaces and Modeling* (2007), pp. 31–36. 2
- [HML*14] HU L., MA C., LUO L., WEI L.-Y., LI H.: Capturing braided hairstyles. *ACM Trans. Graph.* 33, 6 (2014), 225:1–225:9. 3
- [HMLL14] HU L., MA C., LUO L., LI H.: Robust hair capture using simulated examples. *ACM Trans. Graph.* 33, 4 (2014), 126:1–126:10. 3, 4, 7, 8
- [HMLL15] HU L., MA C., LUO L., LI H.: Single-view hair modeling using a hairstyle database. *ACM Trans. Graph.* 34, 4 (2015), 125:1–125:9. 3
- [IKSM07] ISHIKAWA T., KAZAMA Y., SUGISAKI E., MORISHIMA S.: Hair motion reconstruction using motion capture system. In *SIGGRAPH Posters* (2007). 3
- [JMM09] JAKOB W., MOON J. T., MARSCHNER S.: Capturing hair assemblies fiber by fiber. *ACM Trans. Graph.* 28, 5 (2009), 164:1–164:9. 3
- [KE95] KENNEDY J., EBERHART R.: Particle swarm optimization. In *IEEE Neural Networks* (1995), vol. 4, pp. 1942–1948 vol.4. 2, 6, 7
- [KTS*14] KAUFMAN D. M., TAMSTORF R., SMITH B., AUBRY J.-M., GRINSPUN E.: Adaptive nonlinearity for collisions in complex rod assemblies. *ACM Trans. Graph.* 33, 4 (2014), 123:1–123:12. 3
- [LCF00] LEWIS J. P., CORDNER M., FONG N.: Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *SIGGRAPH '00* (2000), pp. 165–172. 8
- [LHZW12] LAY HERRERA T., ZINKE A., WEBER A.: Lighting hair from the inside: A thermal approach to hair reconstruction. *ACM Trans. Graph.* 31, 6 (2012), 146:1–146:9. 3
- [LLP*12] LUO L., LI H., PARIS S., WEISE T., PAULY M., RUSINKIEWICZ S.: Multi-view hair capture using orientation fields. In *IEEE CVPR* (2012). 3
- [LLR13] LUO L., LI H., RUSINKIEWICZ S.: Structure-aware hair capture. *ACM Trans. Graph.* 32, 4 (2013), 76:1–76:12. 3
- [LLW*11] LUO L., LI H., WEISE T., PARIS S., PAULY M., RUSINKIEWICZ S.: *Dynamic Hair Capture*. Tech. rep., Princeton University, 2011. 3

- [LZZR13] LUO L., ZHANG C., ZHANG Z., RUSINKIEWICZ S.: Wide-baseline hair capture using strand-based refinement. In *IEEE CVPR* (2013). 3
- [MBT*12] MIGUEL E., BRADLEY D., THOMASZEWSKI B., BICKEL B., MATUSIK W., OTADUY M. A., MARSCHNER S.: Data-driven estimation of cloth simulation models. *Comput. Graph. Forum* 31, 2 (2012), 519–528. 3
- [MJC*03] MARSCHNER S. R., JENSEN H. W., CAMMARANO M., WORLEY S., HANRAHAN P.: Light scattering from human hair fibers. *ACM Trans. Graphics (Proc. SIGGRAPH)* 22, 3 (2003), 780–791. 8
- [PBnS04] PARIS S., BRICEÑO H. M., SILLION F. X.: Capture of hair geometry from multiple images. In *ACM SIGGRAPH 2004 Papers* (2004), SIGGRAPH '04, pp. 712–719. 3
- [PCK*08] PARIS S., CHANG W., KOZHUSHNYAN O. I., JAROSZ W., MATUSIK W., ZWICKER M., DURAND F.: Hair photobooth: Geometric and photometric acquisition of real hairstyles. *ACM Trans. Graph.* 27, 3 (2008). 3
- [SLF08] SELLE A., LENTINE M., FEDKIW R.: A mass spring model for hair simulation. *ACM Trans. Graph.* 27, 3 (2008). 3, 5, 12
- [TKA11] TWIGG C. D., KAČIĆ-ALESIĆ Z.: Optimization for sag-free simulations. In *Proc. SCA* (2011), SCA '11, pp. 225–236. 3
- [WBC07] WITHER J., BERTAILS F., CANI M.-P.: Realistic hair from a sketch. *Shape Modeling and Applications 0* (2007), 33–42. 2
- [WBK*07] WARD K., BERTAILS F., KIM T.-Y., MARSCHNER S. R., CANI M.-P., LIN M. C.: A survey on hair modeling: Styling, simulation, and rendering. *IEEE TVCG* 13, 2 (2007), 213–234. 3
- [WQOS05] WEI Y., OFEK E., QUAN L., SHUM H.-Y.: Modeling hair from multiple views. *ACM Trans. Graph.* 24, 3 (2005), 816–820. 3
- [WOR11] WANG H., O'BRIEN J. F., RAMAMOORTHY R.: Data-driven elastic models for cloth: Modeling and measurement. *ACM Trans. Graph.* 30, 4 (2011). 3
- [WWL*13] WENG Y., WANG L., LI X., CHAI M., ZHOU K.: Hair Interpolation for Portrait Morphing. *Comp. Graph. Forum* (2013). 2
- [WY*15] WANG B., WU L., YIN K., ASCHER U., LIU L., HUANGM H.: Deformation capture and modeling of soft objects. *ACM Transactions on Graphics(Proc. of SIGGRAPH)* 34, 4 (2015), 94:1–94:12. 3
- [WYZG09] WANG L., YU Y., ZHOU K., GUO B.: Example-based hair geometry synthesis. *ACM Trans. Graph.* 28, 3 (2009), 56:1–56:9. 2, 4
- [XWW*14] XU Z., WU H.-T., WANG L., ZHENG C., TONG X., QI Y.: Dynamic hair capture using spacetime optimization. *ACM Trans. Graph.* 33, 6 (2014), 224:1–224:11. 3, 5, 11, 12
- [YSK09] YUKSEL C., SCHAEFER S., KEYSER J.: Hair meshes. *ACM Trans. Graph.* 28, 5 (2009), 166:1–166:7. 2
- [YWO08] YAMAGUCHI T., WILBURN B., OFEK E.: Video-based modeling of dynamic hair. In *Proc. Advances in Image and Video Technology* (2008), pp. 585–596. 3
- [YYCY14] YU X., YU Z., CHEN X., YU J.: A hybrid image-cad based system for modeling realistic hairstyles. pp. 63–70. 2
- [ZTW*12] ZHANG Q., TONG J., WANG H., PAN Z., YANG R.: Simulation guided hair dynamics modeling from video. *Comp. Graph. Forum* 31, 7 (2012), 2003–2010. 3