Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2012)
P. Kry and J. Lee (Editors)

Multi-linear Data-Driven Dynamic Hair Model with Efficient Hair-Body Collision Handling

Peng Guan[1] Leonid Sigal[2] Valeria Reznitskaya[2] Jessica K. Hodgins[2,3]

[1]Brown University [2]Disney Research, Pittsburgh [3]Carnegie Mellon University

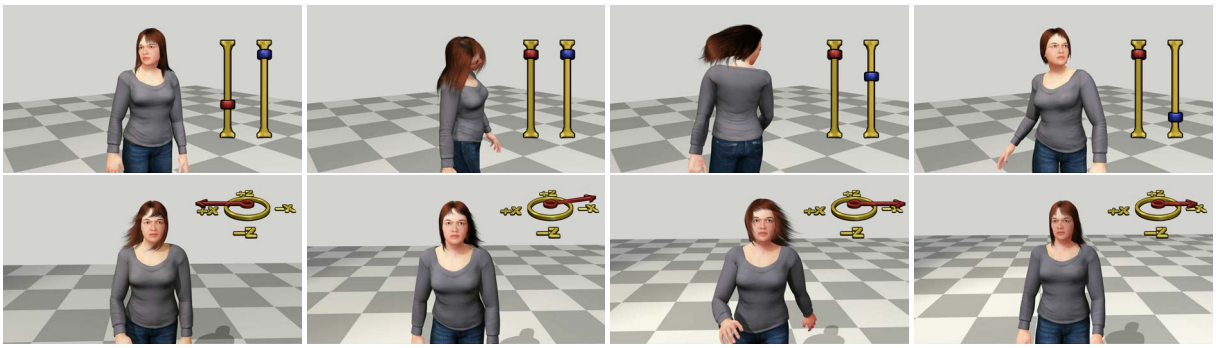

Figure 1: *Real-time animation of 900 guide multi-linear hair model, with interactive control over the hair softness (red slider, the higher the softer) and length (blue slider, the higher the longer); bottom row shows interactive control of wind strength (arrow length) and direction (arrow orientation).*

Abstract

We present a data-driven method for learning hair models that enables the creation and animation of many interactive virtual characters in real-time (for gaming, character pre-visualization and design). Our model has a number of properties that make it appealing for interactive applications: (i) it preserves the key dynamic properties of physical simulation at a fraction of the computational cost, (ii) it gives the user continuous interactive control over the hair styles (e.g., lengths) and dynamics (e.g., softness) without requiring re-styling or re-simulation, (iii) it deals with hair-body collisions explicitly using optimization in the low-dimensional reduced space, (iv) it allows modeling of external phenomena (e.g., wind). Our method builds on the recent success of reduced models for clothing and fluid simulation, but extends them in a number of significant ways. We model motion of hair in a conditional reduced sub-space, where the hair basis vectors, which encode dynamics, are linear functions of user-specified hair parameters. We formulate collision handling as an optimization in this reduced sub-space using fast iterative least squares. We demonstrate our method by building dynamic, user-controlled models of hair styles.

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

1. Introduction

Hair animation is a difficult task, primarily due to the large volume of hairs that need to be considered (a typical human head consists of 100,000 hair strands) and the complex hair motions and interactions. Despite this, there has been enormous success in model acquisition [PMC*08], simulation [SLF08, DDB11] and rendering of hair (e.g., Rapunzel's hair in Tangled [WSM*10]). Such high-quality simulations, however, are expensive and require off-line processing. The approach of Daviet and colleagues [DDB11]

DOI: 10.2312/SCA/SCA12/295-304

simulates 25 seconds of video in 48 hours (using 2,000 rods) and that of Salle and colleagues [SLF08] simulates 1 frame in 4-38 minutes. Real-time applications, such as prototyping and games, have more stringent computational budgets, and hence often rely on less realistic models which are either entirely procedural [CMM09], topologically constrained [YSK09], or approximate simulation using low-resolution (e.g., guide curves or strips [KHS04]) or level of detail models [WLL*05].

Rather than attempt to create a fast physically accurate simulation, our goal is to learn a flexible low-dimensional representation of dynamic hair motion that is compact and fast, but at the same time expressive enough to convey the dynamic behaviors seen in high-resolution simulations. Our data-driven approach generates motion that models the dynamics of hair and provides a level of accuracy comparable to the input data. Our method builds on the recent success of reduced models for clothing [dASTH10] and fluid simulation [TLP06], but extends them in a number of significant ways. It is a general method that is also applicable in other domains, for example for modeling clothing. The approach of de Aguiar and colleagues [dASTH10] is a special case of our model. Here we focus primarily on hair animation.

We leverage hair simulations produced by a standard simulation package (Shave and a Haircut [Alt06]) to build a highly efficient multi-linear model of hair motion as a function of several user-controlled parameters (hair length, softness and wind direction). To build this model, we make two basic assumptions: (i) characters in interactive domains typically exist in finite configuration spaces, where, for example, the user has control over the transitions between a finite set of motions (e.g., as in motion graphs); or has limited dynamic control over the raw character motion (e.g., as with most interactive controllers); and (ii) there exists a continuous manifold space of hair models parameterized by geometric, dynamic, and external factors acting on the hair. The second assumption is motivated by hair grooming and simulation tools that typically provide continuous control over similar parameters but off-line.

Our method takes, as input, multiple sets of hair motions produced by a simulator under various perturbations in the parameters of interest, and learns a reduced multi-linear dynamical model approximating the behavior of hair exhibited across all sets. As a consequence, one can think of the conditional dynamic base vectors, modeling hair evolution, as being functions of real-valued factors that can be specified by the user at test time. Thus using a discrete set of simulations, we are able to build a continuous and intuitive space of dynamic hair models. Because our learning method is statistical in nature, the raw results from the multi-linear model can only *approximately* resolve body-hair contacts. This limitation can cause unwanted hair-body penetrations. To explicitly handle this problem in our model, we propose an optimization step that resolves collisions by optimizing the reduced space representation directly. This process is efficient because we only need to optimize a small set of hair parameters, instead of raw hair strand vertex positions.

Unlike prior real-time hair-simulation methods that typically rely on low-resolution models (with a handful of strips or wisps), our model is considerably more efficient and can deal with up to 4,000 guide hair strands at a small fraction of the computational cost. In contrast to most model reduction approaches [TLP06], we assume no specific form for the dynamics. In contrast to data-driven methods [dASTH10], we do not learn a single linear dynamical model, but rather a family of models parameterized by semantic user-specifiable parameters (including external factors like the wind); we also explicitly and efficiently deal with hair-body collisions, which was a limitation of [dASTH10].

Contributions: We introduce a data-driven multi-linear reduced-space dynamical model for modeling hair. It is explicitly parameterized by a number of real-valued factors (e.g., hair length, hair softness, wind direction/strength, etc.) that make it easy to adjust the groom and motion of hair interactively at test time. We formulate our model using tensor algebra and illustrate how dynamics can be incorporated within this framework. Further, we explicitly address the issue of hair-body collisions by a very efficient optimization procedure formulated directly in the reduced space and solved using a form of iterative least squares. Our formulation goes substantially beyond current reduced-space dynamical models (e.g., [dASTH10]).

2. Related Work

A large body of work exists on hair modeling, simulation, and rendering. We refer the reader to a survey [WBK*07] and prior SIGGRAPH course notes [BHC*08, YT10] for an overview. Hair simulation approaches can loosely be organized into two classes of methods: those that model hair as a continuous medium and those that model it as a set of disjoint, possibly interacting, groups [WBK*07].

Continuous medium models, model hair as a continuum and model complex interactions between strands using fluid dynamics (smooth particle hydrodynamics) [BCN03, HMT01]. Such methods, however, are slow and do not capture the clustering effects observed in longer hair.

Disjoint models typically model hair using a sparse set of hair guides, hair strips, or wisps. Hair guides are *representative* strands that are simulated; the dense hair model is then created by interpolating the position of the remaining strands from a set of hair guides [CJY02]. This approximation allows nearly real-time performance with a moderate number of guides (a GPU implementation with 166 simulated strands can run at 15 FPS [TB08]). Hair strips model hair using thin flat patches (NURBS surfaces) [KH01]. Using a strip to represent tens or hundreds of individual strands leads to significant efficiencies, particularly in

collision handling, resulting in real-time performance; consequently this approach is often used in games. However, strips are unable to represent complex hair styles or motions. Wisps model bundles of hair strands as volumetric primitives [CCK05, WS92, YXWY00]. These approaches are particularly good at modeling hair styles with well-defined clusters; however, they are typically computationally expensive (e.g., requiring seconds per frame to compute [CCK05]). Another promising approach uses the hair mesh structure for modeling the hair volume; topological constraints allow an automatic and unique way to trace the path of individual hair strands through this volume [YSK09]. With a coarse resolution mesh this approach is able to simulate hair at 92 FPS. However, the coarse resolution of the mesh does not allow for fine movement of individual strands. Level of detail models such as one proposed by Ward and colleagues [WLL*05] can improve performance, on average, by leveraging lower resolution simulations when character is further away from the camera; however, at closeups the simulation is still relatively expensive (about 8 FPS).

Our model conceptually builds on the clothing model introduced by [dASTH10]. In particular, for a given setting of factors, our model reduces to a model very similar to theirs. However, our model is considerably more general as it allows for the real-valued factors, that account for the hair groom, style, and external phenomena, to be controlled interactively. These factors in essence modulate the basis of the learned dynamical model. In contrast, their model would require a separate model to be learned each time parameters of simulation or groom of the hair changed; producing models for only a discrete set of simulations performed at training. We are also able to model external factors, such as (controllable) wind, which would not be possible with their approach. In addition, we explicitly deal with collision detection and resolution in the reduced space; [dASTH10] only approximately maintains depth ordering and requires a custom rendering pipeline to resolve problematic cases. Self-collisions in the reduced spaces were previously addressed in [BJ10]. That approach is not applicable, however, to hair-body collisions as the topology of both the body and the hair is changing over time.

Our formulation builds on the formalism of multilinear subspace learning from tensor data; a survey of related methods is available from [LPV11]. In the past, multi-linear models have been used for face recognition [VT02b] and transfer [VBPP05], human motion modeling [MLC10] and image-based texture mapping [VT04]. Unlike prior methods that use multi-linear models as a way to build generative representations of spatial or spatio-temporal data, we utilize them to build *conditional* models of hair dynamics. Finally, because we use style (or groom) as one of the factors, the space of geometric variations in hair groom can also be modeled. This addition allows users to create styles on-line without explicit grooming by an animator in much the way that human body shapes are modeled in [ACP03].

3. Representation

We use a physically based hair simulation software (Shave and a Haircut [Alt06]) to simulate a large number of hair guides, each guide being the proxy for a bundle of hair strands. Our method operates on hair guides as this is a typical representation for hair simulators. However, unlike other methods (e.g., [TB08]) that use few (up to 200) hair guides to reduce simulation complexity, we utilize up to 4,000 hair guides with our model†. The hair guides are simulated on the head of the virtual character, animated and skinned using a set of 35 motion capture sequences. We adopt a standard approach to interpolate between hair guides to obtain a full set of hair strands [ND05].

Hair: We use N_g guides per-frame and every guide $\mathbf{g}_k (1 \leq k \leq N_g)$ is a curve represented by $N_m = 15$ points in 3D (see Figure 2). We generate three different datasets with N_g being 198, 962, and 3980 respectively. Let $\mathbf{g}_{k,1}, \mathbf{g}_{k,2}, ..., \mathbf{g}_{k,N_m} \in \mathbb{R}^3$ be the points on guide k. We concatenate the x,y,z coordinates of points from the guide and obtain $\mathbf{g}_k = [\mathbf{g}_{k,1}, \mathbf{g}_{k,2}, ..., \mathbf{g}_{k,N_m}] = [g_{k,1,x}, g_{k,1,y}, g_{k,1,z}, ..., g_{k,N_m,x}, g_{k,N_m,y}, g_{k,N_m,z}] \in \mathbb{R}^{3N_m}$. We put together all the guides and use a tall vector $\mathbf{h} = [\mathbf{g}_1, \mathbf{g}_2, ..., \mathbf{g}_{N_g}]^T \in \mathbb{R}^{N_h}$ to represent one frame of hairs, where $N_h = 3N_m N_g$.

Body: Similarly we represent the body using a set of vertices of the triangular mesh (see Figure 2). For the purposes of our model we only need to consider the head and the shoulders (the *bust*) of the mesh with which hair can potentially come in contact. Assuming that there are N_n vertices in the bust and that each vertex is represented as $\mathbf{b}_i = [b_{i,x}, b_{i,y}, b_{i,z}] \in \mathbb{R}^3$, at a single frame the body is represented using $\mathbf{b} = [\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_{N_n}]^T \in \mathbb{R}^{N_b}$, where $N_b = 3N_n$.

3.1. Dimensionality Reduction in Canonical Space

Given the correlation among the hair guides (and body vertices) and the constrained topology of hair points, the underlying number of degrees of freedom (DOF) is much less than N_h (or N_b in the case of the body). Hence, we adopt Principal Component Analysis (PCA) to reduce the dimensionality of the two representations. We are able to capture most of the variation in the geometric hair appearance using a much lower dimensional space (typically 50 to 100); as the configuration of the bust is much more constrained, we use only 10 dimensions to represent it. The choice of the space in which the hair and bust are represented is also an important practical issue. Representation in the original world space hinders generalization [dASTH10]. Therefore, we model the motion in a canonical space of the bust.

† In effect we show that by treating *all* hair strands as guide curves in our framework, we can forego the interpolation step as our model learns to incorporate "interpolation" as part of the mapping from the reduced space to the full-dimensional hair representation.

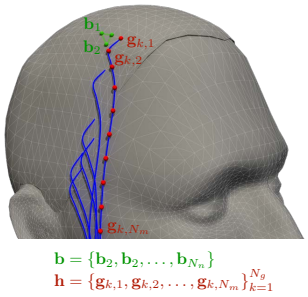

Figure 2: ***Hair and body parametrization:*** *We represent hair using control points on a sparse set of guides and body using vertices making up the triangular mesh of bust.*

We assume that the hair motion is only determined by the motion of the *bust*. We do not consider hair-hand interaction in this work. To normalize hairs and bust at frame t, we transform all the hair points and the bust vertices into a canonical space by: (1) subtracting the average position of the bust vertices, $\mathbf{p}_t = \frac{1}{N_n}\Sigma_{i=1}^{N_n}\mathbf{b}_{i,t} \in \mathbb{R}^3$ and (2) rotating the bust (and hairs) around the Y-axis, $r_t \in \mathbb{R}^1$ such that the head is facing towards the positive Z-axis; the negative Y-axis is the gravity direction. PCA is applied on the normalized data.

As a result, the hair at frame t, $\mathbf{h}_t \in \mathbb{R}^{N_h}$ can be written as:

$$\mathbf{h}_t = R_y(r_t)[\mathbf{Q}^h\mathbf{y}_t + \mu^h] + \mathbf{p}_t, \quad (1)$$

where $R_y(r_t)$ is a 3×3 rotation matrix around the Y-axis that rotates the hairs from a canonical space back to world space, $\mathbf{Q}^h \in \mathbb{R}^{N_h \times d_h}$ are the eigenvectors learned by the hair PCA, d_h is the dimension we choose to represent the hair, μ^h is the mean location of hairs in the canonical space, and $\mathbf{y}_t$ is a vector of hair PCA coefficients for frame t.

The bust vertices are represented in a similar way:

$$\mathbf{b}_t = R_y(r_t)[\mathbf{Q}^b\mathbf{x}_t + \mu^b] + \mathbf{p}_t, \quad (2)$$

where $\mathbf{Q}^b \in \mathbb{R}^{N_b \times d_b}$ are the eigenvectors learned by the bust PCA, d_b is the dimension we choose to represent the bust, μ^b is the mean location of bust vertices learned from training data, and $\mathbf{x}_t$ is a vector of bust PCA coefficients for frame t.

4. Multi-linear Hair Framework

The appearance of hair is a composite effect of many factors, such as length, softness, head pose and motion. We explicitly parameterize our hair model using these real-valued factors. By changing the values of any of these factors, we are able to synthesize hair with different appearance, configuration and motion. To simplify the formulation, we first introduce a generative multi-linear model for hair appearance in a given frame and then illustrate how that model can be extended to incorporate dynamics for synthesis.

Multi-linear algebra provides us with a mathematical framework to factorize hair appearance. The simulated hair exemplars, parameterized by reduced representation, are built into a data tensor $\mathcal{D}$ that is later decomposed in order to separate and represent each constituent factor. We use the Matlab Tensor Toolbox [BK06] to perform tensor operations. Hair data is built into a $N > 2$ tensor or N-way array $\mathcal{D}$, and N-mode singular value decomposition (N-SVD) orthogonalizes N spaces and decomposes the tensor into the $mode-N$ product [VT02a, VT02b]:

$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \ldots \times_i \mathbf{U}_i \ldots \times_N \mathbf{U}_N. \quad (3)$$

The core tensor $\mathcal{Z}$ governs the interaction between the mode matrices $\mathbf{U}_1$,...,$\mathbf{U}_N$, and each mode matrix $\mathbf{U}_i$ is obtained by $mode-i$ flattening of $\mathcal{D}$ [BK06].

We introduce the formulation in terms of a simple model with two factors, but build and discuss a variety of other models of this form in the results section. We prepare the training dataset such that we have $N_l = 2$ different hair lengths (*short* and *long*) and $N_s = 2$ different hair softnesses (*soft* and *stiff*). Note that all hair models in our dataset are in correspondence, i.e., contain the same number of hair strands, the same number of points per strand and the same scalp attachment points. Each hair length and softness combination corresponds to approximately $N_f = 12000$ frames of different head poses from 35 training sequences (animated using motion capture data). The total size of the training set is $N_l \times N_s \times N_f$ frames. We now show how we can represent hair, $\mathbf{y} \in \mathbb{R}^{d_h}$, using a multi-linear generative model.

For the simple case of the two factors of length and softness, our hair data tensor $\mathcal{D}$ is a $d_h \times N_l \times N_s \times N_f$ array, which is decomposed to:

$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_{hair} \times_2 \mathbf{U}_{length} \times_3 \mathbf{U}_{softness} \times_4 \mathbf{U}_{config}. \quad (4)$$

$\mathcal{Z} \in \mathbb{R}^{d_h \times N_l \times N_s \times N_f^*}$, with $N_f^* = min(N_f, d_h \cdot N_l \cdot N_s) = d_h \cdot N_l \cdot N_s$, is the core tensor and $\mathbf{U}_{hair}$ is the hair mode matrix which will be projected out (see Figure 3). The $N_l \times N_l$ mode matrix $\mathbf{U}_{length}$ spans the space of hair length parameters, each row of which corresponds to a different hair length in our dataset. Similarly, the $N_s \times N_s$ mode matrix $\mathbf{U}_{softness}$ spans the space of hair softness parameters, each row of which corresponds to a different hair softness in our dataset. $\mathbf{U}_{config}$ spans the space of hair configurations that encode variations in hair appearance as the body moves. This model characterizes how hair length, softness and configuration interact and multiplicatively modulate the appearance of hair.

We can synthesize novel hair length and softness by interpolating between the rows in $\mathbf{U}_{length}$ ($\mathbf{U}_{softness}$). This interpolation corresponds to convex combination of bases, using barycentric coordinates, and can be extended to a dataset with $N_l > 2$ and/or $N_s > 2$. Let $\mathbf{v}_{length} \in \mathbb{R}^{N_l}$ ($\mathbf{v}_{softness} \in \mathbb{R}^{N_s}$) be a vector of coefficients that interpolates between the rows of $\mathbf{U}_{length}$ ($\mathbf{U}_{softness}$). Note that for our simple dataset, where $N_l = 2$, $\mathbf{v}_{length} = \mathbf{U}_{length}^T \cdot [\alpha, (1-\alpha)]^T$, where $\alpha \in (0,1)$.

We can *generate* hair coefficients, $\mathbf{y}$, by specifying all the

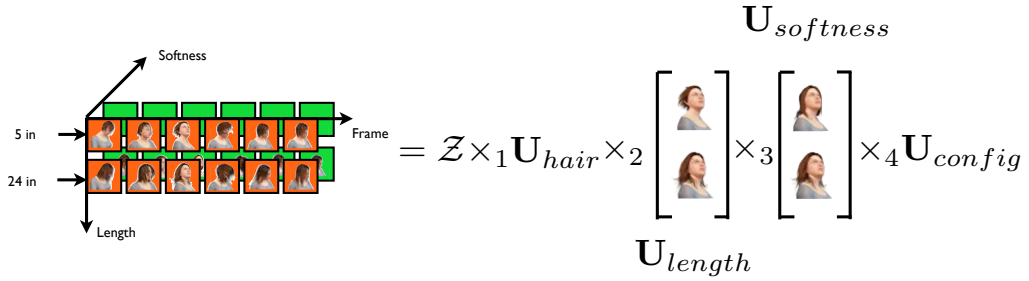

Figure 3: ***Multi-linear hair model:*** *The representation of the hair tensor $\mathcal{D}$ (left) as a core tensor and mode matrices (right).*

constituent factors (length, softness, and configuration):

$$\mathbf{y} = \mathcal{Z} \times_1 \mathbf{U}_{hair} \times_2 \mathbf{v}_{length} \times_3 \mathbf{v}_{softness} \times_4 \mathbf{v}_{config}. \quad (5)$$

Eq. 5 allows us to generate hair with different appearance using only a few matrix multiplications. In fact, to synthesize hairs with fixed style (length and softness), we can precompute $\mathcal{M} \in \mathbb{R}^{d_h \times (N_l \cdot N_s \cdot N_f^*)}$

$$\mathcal{M} = \mathcal{Z} \times_1 \mathbf{U}_{hair} \times_2 \mathbf{v}_{length} \times_3 \mathbf{v}_{softness}, \quad (6)$$

which corresponds to a linear space that spans the hair PCA coefficients. Only one matrix multiplication is needed to obtain $\mathbf{y} = \mathcal{M} \cdot \mathbf{v}_{config}$, where $\mathbf{v}_{config}$ is the set of coefficients that encode hair configuration. However, for a given frame we do not have explicit knowledge of $\mathbf{v}_{config}$ *a priori*, instead in the next section we show how we can solve for $\mathbf{v}_{config}$ by conditioning the model on the bust pose and previous hair configurations; conditioning on previous hair configurations allows us to model dynamics.

5. Dynamics

The simple formulation above is unable to model dynamics and there is no intuitive way to condition the model to obtain $\mathbf{v}_{config}$ for a given frame. To address the first limitation, we build a generative model over a short (3-frame) temporal window of hair and bust configurations. This allows us to model the relationship between the (presumably unknown) hair configuration at the current frame and the (presumably known) body as well as (presumably known) hair configurations at the past frames. To address the second limitation, we show how this model can then be conditioned to predict/simulate the configuration of the hair at the current frame. More specifically, we assume a second order dynamical model on the hair (consistent with a second order ODE governing the true dynamics and empirical observations in [dASTH10]). We also assume a control signal $\mathbf{x}_t$, in the form of a bust at time t, that governs the motion of the hair and (later) collision detection.

Dynamic Multi-linear Hair Model: We learn a multi-linear model as in Section 4, but with augmented vectors $\mathbf{w}_t = [\mathbf{x}_t; \mathbf{y}_{t-2}; \mathbf{y}_{t-1}; \mathbf{z}_{t,t-2}; \mathbf{z}_{t-1,t-2}; \mathbf{y}_t] \in \mathbb{R}^{d_a}$, where $d_a = d_b + 3d_h + 10$, and $\mathbf{z}_{t,j} \in \mathbb{R}^5$ encodes the relative global bust translation and rotation at frame t with respect to frame j:

$$\mathbf{z}_{t,j} = \begin{bmatrix} R_y(-r_j)(\mathbf{p}_t - \mathbf{p}_j) \\ sin(r_t - r_j) \\ cos(r_t - r_j) \end{bmatrix}. \quad (7)$$

Note that we need to add $\mathbf{z}_{t,t-2}$ and $\mathbf{z}_{t-1,t-2}$ because the body and hair are normalized into a canonical space, so the incremental global motion is lost and needs to be added back (in the form of these auxiliary variables). The resulting hair tensor is $\mathcal{D} \in \mathbb{R}^{d_a \times N_l \times N_s \times N_f^*}$, where $N_f^* = d_a \cdot N_l \cdot N_s$. We also experimented with a complete generative model over the 3-frame temporal window (by adding $\mathbf{x}_{t-1}$ and $\mathbf{x}_{t-2}$ to the augmented vector $\mathbf{w}_t$) as well as with longer temporal windows but longer windows did not result in better performance, often led to over-fitting, and resulted in higher dimensional (more expensive) model inference.

Simulation as Inference: For every time instant, we need to estimate $\mathbf{y}_t$ to animate the hair. To do so, we treat $\mathbf{y}_t$ as missing data and infer it using the generative multi-linear model above operating on the augmented representation $\mathbf{w}_t$; we do so by conditioning on the part of the vector $\mathbf{w}_t^o$ that is observed at a given time instance. In the general case ($t \geq 3$), $\mathbf{w}_t^o = [\mathbf{x}_t; \mathbf{y}_{t-2}; \mathbf{y}_{t-1}; \mathbf{z}_{t,t-2}; \mathbf{z}_{t-1,t-2}] \in \mathbb{R}^{d_a - d_h}$. For every time instance, we condition our model on the observed part, $\mathbf{w}_t^o$, and infer/predict the missing part, $\mathbf{y}_t \in \mathbb{R}^{d_h}$ (i.e., hair coefficients for the current frame). For a given hair style (fixed hair length and softness), our pre-computed matrix $\mathcal{M} = [\mathcal{M}^o; \mathcal{M}^{\mathbf{y}}]$ computed using Equation 6, can be decomposed into two parts, consisting of bases for reconstruction of observed variables, $\mathcal{M}^o$, and $\mathbf{y}_t$ itself.

From Section 4, we know that $\mathbf{w}_t = [\mathbf{w}_t^o; \mathbf{y}_t] = \mathcal{M} \cdot \mathbf{v}_{config,t}$. Hence, we can solve for the linearly *optimal* $\mathbf{v}_{config,t}$ for the current frame t by doing a linear sub-space solve, $\mathbf{v}_{config,t} = (\mathcal{M}^o)^\dagger \cdot \mathbf{w}_t^o$, where $\dagger$ is the pseudo inverse. We can then reconstruct $\mathbf{y}_t$ from $\mathbf{v}_{config,t}$, resulting in a very efficient and compact iterative simulation equation,

$$\mathbf{y}_t = \mathcal{M}^{\mathbf{y}} \cdot (\mathcal{M}^o)^\dagger \cdot \mathbf{w}_t^o. \quad (8)$$

Note, that if we want to change the hair style anywhere (or

continuously) within a sequence, we just need to re-compute $\mathcal{M}^{\mathbf{y}} \cdot (\mathcal{M}^{o})^{\dagger}$. For a general case we then have,

$$\mathbf{y}_t = \mathcal{M}^{\mathbf{y}} \cdot (\mathcal{M}^{o})^{\dagger} \cdot [\mathbf{x}_t; \mathbf{y}_{t-2}; \mathbf{y}_{t-1}; \mathbf{z}_{t,t-2}; \mathbf{z}_{t-1,t-2}]. \quad (9)$$

For a given set of factors, the model can be interpreted as a second order conditional linear dynamical system, similar to the one proposed in [dASTH10], i.e.,

$$\mathbf{y}_t = \mathbf{A}\mathbf{x}_t + \mathbf{B}_1\mathbf{y}_{t-2} + \mathbf{B}_2\mathbf{y}_{t-1} + \mathbf{C}_1\mathbf{z}_{t,t-2} + \mathbf{C}_2\mathbf{z}_{t-1,t-2}, \quad (10)$$

where

$$\mathcal{M}^{\mathbf{y}} \cdot (\mathcal{M}^{o})^{\dagger} = [\mathbf{A} \quad \mathbf{B}_1 \quad \mathbf{B}_2 \quad \mathbf{C}_1 \quad \mathbf{C}_2]. \quad (11)$$

Therefore, the model proposed by de Aguiar and colleagues is a special case of our more general formulation.

For the cases where $t = 1, 2$, the process is very similar except that $\mathbf{w}_t^o = [\mathbf{x}_t]$, and the missing part becomes $[\mathbf{y}_{t-2}; \mathbf{y}_{t-1}; \mathbf{z}_{t,t-2}; \mathbf{z}_{t-1,t-2}; \mathbf{y}_t]$.

5.1. Stability of dynamics

Similarly to [dASTH10], we can measure the stability of the learned model by looking at the largest eigenvalue, λ_{max}, of linear dynamics matrix of the dynamical system, namely:

$$\begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 \\ \mathbf{I}_{d_h \times d_h} & \mathbf{0}_{d_h \times d_h} \end{bmatrix}. \quad (12)$$

The key difference between our approach and [dASTH10] is that $\mathbf{B}_1$ and $\mathbf{B}_2$ are both functions of the factors, $\mathbf{v}_{length}$ and $\mathbf{v}_{softness}$, in the multi-linear model. Hence, to prove stability we need to ensure that the largest eigenvalue λ_{max} is ≤ 1 for *any* value of factors in our model, i.e., we need to show that:

$$\lambda_{max} = \arg \max_{\alpha \in [0,1], \beta \in [0,1]} \lambda_{max}(\alpha, \beta) \leq 1. \quad (13)$$

where α and β are parameters interpolating between the bases of $\mathbf{U}_{length}$ and $\mathbf{U}_{softness}$ respectively. Taking arg max is difficult in practice, therefore we resort to an approximation obtained by evaluating arg max using a set of discrete samples (by uniformly and finely sampling α and β in the range of 0 to 1) and assuming eigenvalues are locally smooth as a function of α and β. We report the λ_{max} for several hair configurations in Table 1. We observe that all trained models are stable with λ_{max} consistently < 1.

6. Collision Handling

The reconstructed hairs $\mathbf{h}_t$, which are a function of predicted hair coefficients $\mathbf{y}_t$, may penetrate the bust. We propose a simple and efficient method to resolve collisions. This method is based on minimizing hair-bust penetration while keeping the predicted hair coefficients unchanged as much as possible. Collision handling is done in the normalized coordinates in the reduced PCA space (for efficiency).

Our measurement of collision is based on a simple approximation of the signed distance to the body mesh. For a hair point $\mathbf{h}_i(\mathbf{y})$, we find its nearest neighbor vertex on the bust $\mathbf{b}_j$. (We drop the temporal subscript for clarity.) Then the dot product of $\mathbf{b}_j$'s surface normal vector and the offset vector $\mathbf{h}_i(\mathbf{y}) - \mathbf{b}_j$ locally approximates the signed distance to the body mesh for $\mathbf{h}_i(\mathbf{y})$.

$$p_{\mathbf{C}}(\mathbf{y}) = \sum_{(i,j) \in \mathbf{C}} \rho\left(\mathbf{n}_{\mathbf{b}_j}^T \cdot (\mathbf{h}_i(\mathbf{y}) - \mathbf{b}_j)\right), \quad (14)$$

where $\mathbf{C}$ is a set of correspondences between hair guide point $\mathbf{h}_i$ and its closest bust vertex $\mathbf{b}_j$, $\rho(x) = \begin{cases} 0 & x \geq 0 \\ x^2/(\sigma^2 + x^2) & x < 0 \end{cases}$ is a robust error function which only penalizes negative signed distance (i.e., penetrating guide points), $\mathbf{n}_{\mathbf{b}_j}$ is the normal for bust vertex $\mathbf{b}_j$.

Method $\mathcal{A}$: A straightforward way to remove collisions is to minimize the energy function

$$E_{\mathbf{C}}(\mathbf{y}) = \pi_1 p_{\mathbf{C}}(\mathbf{y}) + \pi_2 d_{\mathbf{C}}(\mathbf{y}) + \pi_3 s_{\mathbf{C}}(\mathbf{y}) \quad (15)$$

with respect to the hair PCA coefficients $\mathbf{y}$. The first term, defined in Eq. (14), minimizes penetration. The second term,

$$d_{\mathbf{C}}(\mathbf{y}) = ||\mathbf{y} - \mathbf{y}_0||^2, \quad (16)$$

ensures that the resulting hair coefficients are close to the prediction from the model (to preserve dynamics); where $\mathbf{y}_0$ are the predicted hair PCA coefficients from the multi-linear dynamical model. The third term,

$$s_{\mathbf{C}}(\mathbf{y}) = \sum_{k \in [1, N_g]} ||\mathbf{g}_{k,1}(\mathbf{y}) - \tilde{\mathbf{g}}_{k,1}||^2 \quad (17)$$

ensures that the hair roots are at correct positions on the scalp; where $\tilde{\mathbf{g}}_{k,1}$ is the true hair root position on the scalp for the k-th guide and $\mathbf{g}_{k,1}(\mathbf{y})$ is the model position. π_1, π_2 and π_3 are the relative weights for each of the terms.

Assuming, $\mathbf{y}_t^* = \arg\min E_{\mathbf{C}}(\mathbf{y}_t)$ are the optimized hair coefficients for frame t, the final hair guides in the world space are obtained by: $\mathbf{h}_t^* = R_y(r_t)[\mathbf{Q}^h \mathbf{y}_t^* + \mu^h] + \mathbf{p}_t$. For efficiency, the nearest neighbor correspondences $\mathbf{C}$ are pre-computed, at each frame, based on the model prediction before we use gradient decent optimization on Eq. (15).

Method $\mathcal{B}$: Method $\mathcal{A}$ is fast but still involves a relatively expensive gradient optimization. We propose an approximation scheme which is around 50X faster than Method $\mathcal{A}$ while producing very similar results. The key idea is to reformulate the optimization in Method $\mathcal{A}$ in terms of a series of linear least squares (LLS) problems that can be solved extremely efficiently in closed form. $d_{\mathbf{C}}(\mathbf{y})$ and $s_{\mathbf{C}}(\mathbf{y})$ in Eq. (15) already have a convenient quadratic form and require no special treatment. The first term in Eq. (15), $p_{\mathbf{C}}(\mathbf{y})$, however, is an asymmetric error function and requires approximation. We approximate $p_{\mathbf{C}}(\mathbf{y})$ by taking into account only the set of hair points that currently penetrate $\mathcal{P}$:

$$p_{\mathbf{C}}(\mathbf{y}) \approx \sum_{(i,j) \in \mathbf{C} \bigcap i \in \mathcal{P}} ||\mathbf{n}_{\mathbf{b}_j}^T \cdot (\mathbf{h}_i(\mathbf{y}) - \mathbf{b}_j)||^2 \quad (18)$$

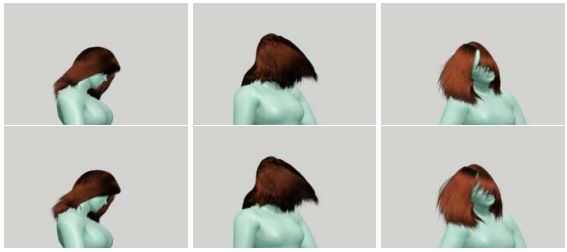

Figure 4: ***Sub-sampling factor:*** *Illustrated are sub-sampling factors of 1 (top) and 15 (bottom) on the 3,980 hair guide dataset. There is almost no visual difference among the hairs corresponding to different sub-sampling factors.*

With this approximation, every term in Eq. (15) takes quadratic form and all the variables are linear functions of unknowns $\mathbf{y}$, resulting in a standard LLS problem. Because the approximation in Eq. (18) is instantaneous and only deals with the current penetrating guide vertices, new penetrations may be introduced in the solution. To address this, we iteratively solve the optimization in Eq. (15), and for each iteration, re-compute Eq. (18), including the current set of penetrating points. However, we only compute hair-body correspondences $\mathbf{C}$ once at the beginning of the optimization and use it throughout the iterations (three to five iterations are sufficient in practice).

Sub-sampling: Method $\mathcal{B}$ allows real-time hair collision handling when the number of hair guides N_g is moderate, but is still expensive for large number of strands. In this scenario, the computational bottleneck of Method $\mathcal{B}$ becomes computing the nearest neighbor correspondences $\mathcal{C}$. To address this, we sub-sample the hair guide strands and only perform collision handling on selected guides. The intuition is that because we are modeling hair in the PCA sub-space, the hair guides are correlated and guides within some neighborhood will generally move together. Assuming this is the case, resolving collisions for some hair guides will implicitly help resolve collisions for nearby hair guides. To achieve this goal we re-write Eq. (18) once again, resulting in the final form for $p_{\mathbf{C}}(\mathbf{y})$:

$$p_{\mathbf{C}}(\mathbf{y}) \approx \tau \sum_{(i,j)\in\mathbf{C}\bigcap i\in\mathcal{P}\bigcap i\in\mathcal{S}_\tau} ||\mathbf{n}_{\mathbf{b}_j}^T \cdot (\mathbf{h}_i(\mathbf{y}) - \mathbf{b}_j)||^2, \quad (19)$$

where τ is the sub-sample factor (e.g., $\tau = 2$ will choose every other hair guide for collision handling), and S_τ is the selected subset of hair strands corresponding to τ.

7. Experiments

We generate three datasets with different numbers of hair guides N_g: a *sparse* hair dataset with $N_g = 198$, a *main* hair dataset with $N_g = 962$, and a *dense* hair dataset with $N_g = 3980$. For the *sparse* hair dataset, we synthesize four sets of hair simulations (long soft, long stiff, short soft, and short stiff) to learn a two factor model. The *main* hair dataset is separated into two parts. The first part has the same four styles as the *sparse* dataset. The second part consists of long soft hairstyle (i) without wind and with wind directions of (ii) +z, (iii) +x, and (iv) -x. We use these four simulation datasets to learn a multilinear model with external wind strength and directions as constituent factors. The *dense* hair dataset has only one style (long soft) because it is expensive to generate training data due to the memory constraints and computing resources. We use the *dense* hair dataset to demonstrate the sub-sampling strategy for collision handling. Each dataset consists of 35 different training body motions from which we learn our multi-linear dynamic hair model and 7 test motions on which we perform our experiments; our test and training sets are disjoint. We choose a dimensionality of $d_h = 100$ for hair coefficients, which represents around 98% energy of the PCA subspace. We set $\pi_1 = 0.08$, $\pi_3 = 1.5$ in Equation 15 for all the experiments.

Model Highlights: A key property of our model is that users are able to interactively change the style of the hair, including softness and length, or apply external forces such as wind (Figure 1). We show side-by-side comparison of different hair lengths in Figure 5 (a)-(d), where (a)-(c) show the interpolated hair lengths with hair length factor being 0 (shortest), 0.5 (interpolated median), and 1 (longest), while (d) shows an extrapolation example where the length factor is 1.35. We can generate additional hair styles, not part of our training sets, by mixing the long and short hair styles we model. Figure 5 (e)(f) show two examples. This functionality opens a possibility for interactive hairstyle design. The comparison to the raw simulation can be found in the accompanying video and shows that our data-driven method can adequately approximate dynamic behaviors of hair (sometimes with fewer body-hair collisions as compared to the original).

Collision Handling: We show the performance of collision handling algorithms on the *sparse* hair dataset ($N_g = 198$), but also find similar trends in all other datasets. We define the following measurements for quantitative evaluation: (1) Penetration rate: the ratio of penetrating hair points to the total hair points. Penetration is defined by Equation 14. (2) The mean of maximal penetration amount over all frames in a sequence. The maximal penetration amount for each frame is defined as $\max|\mathbf{n}_{\mathbf{b}_j}^T \cdot (\mathbf{h}_i - \mathbf{b}_j)|$, where $\mathbf{h}_i$ is a penetrating hair point (see Equation 14). "penetration rate" is the most straightforward measurement while the "maximal penetration amount" provides an upper-bound of how deep a hair point penetrates. These two quantities are informative but not necessarily perceptual; we can arbitrarily decrease π_2 in Equation 15 to achieve better collision handling. Therefore, we use the third measurement: (3) deviation from the hair coefficients prediction: $||\tilde{\lambda}^T(\mathbf{y}^* - \mathbf{y}_0)||/||\mathbf{y}_0||$, where $\mathbf{y}_0$ is the model prediction, $\mathbf{y}^*$ are the hair coefficients after collision handling, and $\tilde{\lambda} = [\lambda_1, \lambda_2, ..., \lambda_{d_h}]^T / \sum_{i=1}^{d_h} \lambda_i$ are the normalized eigenvalues of the hair PCA subspace. We weight the

(a) (b) (c) (d) (e) (f)

Figure 5: ***Creating different grooms:*** *(a) short, (b) interpolated medium, (c) long, (d) extrapolated long, (e) and (f) new hair grooms created by segmenting hair guides into segments and mixing long and short lengths (in (a) and (c)) for each segment.*

hair coefficients deviation $\mathbf{y}^* - \mathbf{y}_0$ according to the importance of the principal directions.

In Figure 6, we show the above-mentioned three measures versus different hair coefficients prior weight π_2. The plot is based on a representative *test* sequence with long soft hairs which includes complex motions. Note that the ground truth hair simulations from Shave and a Haircut in themselves have a non-negligent penetration rate of 1.9% and the mean of maximal penetration amount of 3.8 *mm*. Our collision handling algorithms (both Method $\mathcal{A}$ and $\mathcal{B}$) significantly reduce the penetration. The final penetration rates and amount of Method $\mathcal{B}$ are very similar to $\mathcal{A}$ which indicates that the iterative least squares does approximate the asymmetric error function in $\mathcal{A}$ well. As we can see from Figure 6 (right), the deviation from the original hair coefficients prediction varies between 1.3% and 5.5%, which visually corresponds to very natural dynamics. See the accompanying video for the visual results. Based on these three curves, our selection of π_2 is 0.13 for all the experiments.

Sub-sampling for collision handling: When we model the *dense* hair dataset (3980 hair guides and 59700 hair points), the cost of Method $\mathcal{B}$ is dominated by determining nearest neighbor correspondences $\mathbf{C}$ and penetration set $\mathbf{P}$. Therefore, we show in Figure 7 that we can efficiently sub-sample the hair guides to perform collision handling while still achieving almost the same results. When the sub-sample factor τ increases (see Equation 19), the curve of the penetration rate is almost flat, which indicates that we can sub-sample the hairs significantly without sacrificing the collision handling performance, because the dense hair guides are highly correlated. Figure 4 visually shows the collision handling results using a sub-sample factor of 1 and 15. There is almost no visual difference between two cases. There is, however, a large computational gain; we achieve a 12X speed up by using a sub-sample factor of 15. We plot the time cost of the collision handling procedure (from the predicted hair coefficients $\mathbf{y}_0$ to final hair coefficients $\mathbf{y}^*$) versus the sub-sample factor τ. As τ increases, the time cost drops significantly. The sub-sampling strategy makes it possible for our method to potentially deal with even more hair strands in real time.

Quantitative Evaluation: We show the hair vertex location differences between the Shave and a Haircut simulation and our end results in Table 1. Stiff hairs have much lower errors compared to soft hairs, because the motion of the stiff hairs are more constrained. The long soft hairs with wind have high errors, because wind leads to less predictable hair behavior. The fact that training and testing sequences get similar errors (in some cases the errors of the testing sequences are lower) indicates the generalization power of our method. The stability measurement λ_{max} for each dataset is also shown in the table. These values are all below 1, which shows that our models are stable.

Sparse	L-soft	L-stiff	S-soft	S-stiff	λ_{max}
Training	3.39	2.09	1.73	1.23	0.9792
Testing	3.61	1.93	1.91	1.14	
Main	L-soft	L-stiff	S-soft	S-stiff	
Training	2.85	1.66	1.20	0.84	0.9646
Testing	2.93	1.57	1.22	0.78	
Main	L-soft	L-wind+z	L-wind+x	L-wind-x	
Training	2.97	4.23	4.50	4.32	0.9663
Testing	3.12	4.27	4.47	4.21	
Dense	L-soft				
Training	2.76				0.9621
Testing	2.71				

Table 1: ***Average vertex error and stability:*** *Average vertex error for all the datasets (using Euclidean distance measured in* cm*) and the stability measurement λ_{max} computed over 35 training and 7 testing sequences. "L" and "S" represent long and short hair styles respectively.*

Performance: The speed of our method and the Shave and a Haircut (Shave) simulation package are shown in Table 2. Maya and Shave were run on an Intel Core 2 Extreme X9650, 3.67 GHz processors with 4 GB of RAM. Our model was implemented on a GPU and run on a comparable AMD Phenom(tm) 2 X4 965 processor, 3.4GHz with 8 GB of RAM. The amount of RAM is irrelevant as the model is very compact and easily fits in memory. Note that most of our model (everything except collision detection) is perfectly parallelizable; collision detection requires LLS implementation which could also be made efficient on parallel architectures. Our method easily achieves *real-time* (45-70 FPS) and it is $9-25X$ faster than Shave. We report the results based on long soft hairs, which tend to have the most col-

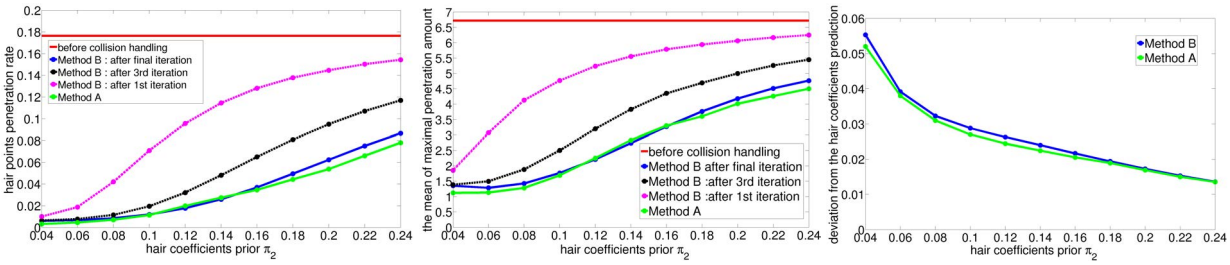

Figure 6: ***Collision handling measurements versus hair coefficients prior.*** *We show the comparison results for method $\mathcal{A}$, two intermediate steps of method $\mathcal{B}$, final results of method $\mathcal{B}$, and before collision handling.*

Hair		Runtime (FPS) Our Method				Shave
#Strands	#Vert	Syn	Col	Recon	Total	
198	2970	1429	74.6	5000	70	7.7
962	14430	1429	52.9	2500	50	5.5
3980	59700	1429	49	909	45	1.8

Table 2: ***Runtime performance:*** *Speed comparison (frames per second) between our method and Shave software package. We divide our method into "Syn" (computing the initial hair coefficients from the multi-linear dynamic model), "Col" (remove hair-body penetration), and "Recon" (reconstruction of the hair vertices from hair coefficients). We choose the sub-sample factor* $\tau = 1, 5, 15$ *for hair strands* $= 198, 962, 3980$ *respectively.*

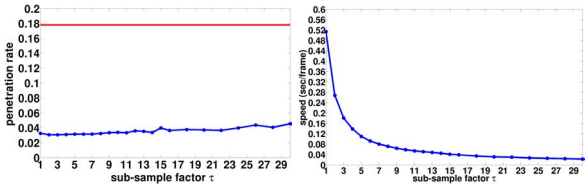

Figure 7: ***Dense hair sub-sampling:*** *Left: penetration rates comparisons between "before collision handling" (red) and various sub-sample factors on a representative sequence (blue). The penetration rates are computed on the full hairs. Right: Time cost of collision handling procedure versus various sub-sample factors.*

lisions. As the number of hair guides increases, our method becomes comparatively much more efficient, due to the benefit of a low-dimensional model that can capture correlations. These results show the potential of our method to deal with large number of hair guides; in doing so it also alleviates the need for additional hair interpolation for rendering. Further speedups are possible using level of detail models and/or culling for parts of the groom that are not visible.

The memory requirement for the our model is quite low. A hair tensor model which has 2 factors with hair dimension of 100 takes approximately 15mb. Adding another factor will double the size, however, too many factors are unlikely. The key advantage of modeling hair in the reduced space is that dimensionality of the hair coefficients is a function of the underlying complexity of the hair motion, and is highly sub-linear with respect to the number of hair strands and the number of vertices per strand. This property also makes the model scalable with the complexity of the hair.

Collision handling for cloth: Our collision handling approach is not limited to hair; it can be used, for example, to resolve body-cloth collisions for clothing. We test our approach on the results of [dASTH10] (see supplemental video). In [dASTH10], the collisions were not resolved explicitly. In contrast, we resolve all the interpenetrations without resorting to a custom rendering pipeline.

8. Discussion and Limitation

We present a method for data-driven animation of hair. The multi-linear nature of our model allows us to control the appearance and motion of hair in real-time. Our method efficiently deals with collisions by formulating collision handling as an iterative least squares optimization in the reduced space. While we illustrate our model on hair, the formulation is general and would work for other physical simulations such as clothing and fur.

Because our model lives in a low-dimensional subspace we are not able to resolve hair-to-hair collisions *explicitly*, as the motion of individual hair strands is not independent. That said, our model is able to resolve hair-to-hair collisions *implicitly* by learning how hair strands should interact based on the input simulations. However, because there is no commercially available simulator that is able to produce such effects, we cannot show any results with hair-to-hair collisions.

We construct two types of reduced models with a number of parameters: short vs. long, stiff vs. soft, and with/without wind as an external force. Naturally there are many other parameters that we might want to include in the hair model: curly vs. straight, dry vs. wet, greasy vs. clean as well as other external forces such as tugs, barrettes, and headbands. In our experiments, the existing model was robust to the modeled parameter space, with no combination of parameters within the modeled ranges (or slightly outside) producing unnatural results. The tensor algebra is general enough

to extend to more factors. However, with more factors being used, the size of the training examples grows exponentially. In theory, if we want to model x factors simultaneously, we need 2^x sets of training data to represent all the combinations of all factors. It would be challenging to keep all the data in memory for training (requiring out of core SVD algorithm). In practice, the animators can choose what factors they want to model to avoid the explosion problem.

In conclusion, the approach to creating a reduced space model presented here appears quite powerful. We expect that implementing this approach for other physical simulations such as clothing and fur would be easy and would result in approximate dynamic models that could be computed and rendered many times faster than real time. This functionality would be useful in visualization of character motion during the animation process as well as allowing rich secondary motion to be added to real-time applications such as games.

References

[ACP03] ALLEN B., CURLESS B., POPOVIC Z.: The space of human body shapes: reconstruction and parameterization from range scans. *ACM Transactions on Graphics (TOG) 22*, 3 (2003).

[Alt06] ALTER J.: Shave and a haircut. *http://www.joealter.com/* (2006).

[BCN03] BANDO Y., CHEN B.-Y., NISHITA T.: Animating hair with loosely connected particles. *Computer Graphics Forum 22*, 3 (2003), 411–418.

[BHC*08] BERTAILS F., HADAP S., CANI M.-P., LIN M., KIM T.-Y., MARSCHNER S., WARD K., KAČIĆ-ALESIĆ Z.: Realistic hair simulation: animation and rendering. In *ACM SIGGRAPH 2008 courses* (2008), pp. 89:1–89:154.

[BJ10] BARBIC J., JAMES D. L.: Subspace self-collision culling. *ACM Transactions on Graphics (TOG) 29*, 3 (2010), 1–9.

[BK06] BADER B. W., KOLDA T. G.: Algorithm 862: Matlab tensor courses for fast algorithm prototyping. *ACM Trans. Mathematical Software 32*, 4 (2006), 635–653.

[CCK05] CHOE B., CHOI M. G., KO H.-S.: Simulating complex hair with robust collision handling. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2005), SCA '05, pp. 153–160.

[CJY02] CHANG J. T., JIN J., YU Y.: A practical model for hair mutual interactions. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2002), SCA '02, pp. 73–80.

[CMM09] CASSOL V., MARSON F., MUSSE S.: Procedural hair generation. In *VIII Brazilian Symposium on Games and Digital Entertainment (SBGAMES)* (2009).

[dASTH10] DE AGUIAR E., SIGAL L., TREUILLE A., HODGINS J. K.: Stable Spaces for Real-time Clothing. *ACM Transactions on Graphics (TOG) 29*, 4 (2010), 1–9.

[DDB11] DAVIET G., DESCOUBES F. B., BOISSIEUX L.: A hybrid iterative solver for robustly capturing coulomb friction in hair dynamics. *ACM Transactions on Graphics (TOG) 30*, 6 (2011), 336–342.

[HMT01] HADAP S., MAGNENAT-THALMANN N.: Modeling dynamic hair as a continuum. *Computer Graphics Forum 20*, 3 (2001), 329–338.

[KH01] KOH C. K., HUANG Z.: A simple physics model to animate human hair modeled in 2d strips in real time. In *Eurographics Workshop on Animation and Simulation* (2001).

[KHS04] KOSTER M., HABER J., SEIDEL H.-P.: Real-time rendering of human hair using programmable graphics hardware. In *Computer Graphics International* (2004).

[LPV11] LU H., PLATANIOTIS K. N., VENETSANOPOULOS A. N.: A survey of multilinear subspace learning for tensor data. *Pattern Recognition 44*, 7 (2011).

[MLC10] MIN J., LIU H., CHAI J.: Synthesis and editing of personalized stylistic human motion. *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (2010).

[ND05] NGUYEN H., DONNELLY W.: Hair animation and rendering in the Nalu demo. *GPU Gems 2* (2005).

[PMC*08] PARIS S., MATUSIK W., CHANG W., JAROSZ W., ZWICKER M., KOZHUSHNYAN O. I., DURAND F.: Hair photobooth: Geometric and photometric acquisition of real hairstyles. *ACM Transactions on Graphics (TOG) 27*, 3 (2008).

[SLF08] SELLE A., LENTINE M., FEDKIW R.: A mass spring model for hair simulation. *ACM Transactions on Graphics (TOG) 27*, 3 (2008).

[TB08] TARIQ S., BAVOIL L.: Real time hair simulation and rendering on the GPU. In *ACM SIGGRAPH Talk* (2008).

[TLP06] TREUILLE A., LEWIS A., POPOVIC Z.: Model reduction for real-time fluids. *ACM Transactions on Graphics (TOG) 25*, 3 (2006).

[VBPP05] VLASIC D., BRAND M., PFISER H., POPOVIC J.: Face transfer with multilinear models. *ACM Transactions on Graphics (TOG) 24*, 3 (2005), 426–433.

[VT02a] VASILESCU M. A. O., TERZOPOULOS D.: Multilinear analysis of image ensembles: Tensorfaces. In *European Conf. on Computer Vision* (2002), pp. 447–460.

[VT02b] VASILESCU M. A. O., TERZOPOULOS D.: Multilinear image analysis for facial recognition. In *International Conf. on Pattern Recognition* (2002), pp. 511–514.

[VT04] VASILESCU M. A. O., TERZOPOULOS D.: Tensortextures: Multilinear image-based rendering. *ACM Transactions on Graphics (TOG)* (2004), 336–342.

[WBK*07] WARD K., BERTAILS F., KIM T.-Y., MARSCHNER S., CANI M.-P., LIN M.: A survey on hair modeling: Styling, simulation, and rendering. *IEEE Transactions on Visualization and Computer Graphics 13*, 2 (2007), 213–234.

[WLL*05] WARD K., LIN M. C., LEE J., FISHER S., MACRI D.: Modeling hair using level-of-detail representations. In *Proceedings of the 16th Int Conf on Computer Animation and Social Agents* (2005), CASA '03, pp. 153–160.

[WS92] WATANABE Y., SUENAGA Y.: A trigonal prism-based method for hair image generation. *IEEE Computer Graphics and Applications 12*, 1 (1992), 47–53.

[WSM*10] WARD K., SIMMONS M., MILNE A., YOSUMI H., ZHAO X.: Simulating rapunzel's hair in Disney's Tangled. *ACM SIGGRAPH (talk article as ACM Trans. on Graphics)* (2010).

[YSK09] YUKSEL C., SCHAEFER S., KEYSER J.: Hair meshes. *ACM Transactions on Graphics (TOG) 28*, 5 (2009).

[YT10] YUKSE C., TARIQ S.: Advanced techniques in real-time hair rendering and simulation. In *ACM SIGGRAPH 2010 courses* (2010).

[YXWY00] YANG X. D., XU Z., WANG T., YANG J.: The cluster hair model. *Graph. Models 62* (March 2000), 85–103.