HairBrush for Immersive Data-Driven Hair Modeling

Jun Xing, Koki Nagano, Weikai Chen, Haotian Xu, Li-Yi Wei, Yajie Zhao, Jingwan Lu, Byungmoon Kim, Hao Li

HairBrush for Immersive Data-Driven Hair Modeling

Jun Xing
USC Institute for Creative Technologies

Koki Nagano
Pinscreen

Weikai Chen
USC Institute for Creative Technologies

Haotian Xu
Wayne State University

Li-Yi Wei
Adobe Research

Yajie Zhao
USC Institute for Creative Technologies

Jingwan Lu
Adobe Research

Byungmoon Kim
Adobe Research

Hao Li
USC Institute for Creative Technologies
Pinscreen

ABSTRACT

While hair is an essential component of virtual humans, it is also one of the most challenging digital assets to create. Existing automatic techniques lack the generality and flexibility to create rich hair variations, while manual authoring interfaces often require considerable artistic skills and efforts, especially for intricate 3D hair structures that can be difficult to navigate. We propose an interactive hair modeling system that can help create complex hairstyles in minutes or hours that would otherwise take much longer with existing tools. Modelers, including novice users, can focus on the overall hairstyles and local hair deformations, as our system intelligently suggests the desired hair parts. Our method combines the flexibility of manual authoring and the convenience of data-driven automation. Since hair contains intricate 3D structures such as buns, knots, and strands, they are inherently challenging to create using traditional 2D interfaces. Our system provides a new 3D hair authoring interface for immersive interaction in virtual reality (VR). Users can draw high-level guide strips, from which our system predicts the most plausible hairstyles via a deep neural network trained from a professionally curated dataset. Each hairstyle in our dataset is composed of multiple variations, serving as blend-shapes to fit the user drawings via global blending and local deformation. The fitted hair models are visualized as interactive suggestions that the user can select, modify, or ignore. We conducted a user study to confirm that our system can significantly reduce manual labor while improve the output quality for modeling a variety of head and facial hairstyles that are challenging to create via existing techniques.

UIST'19 , October 20–23, 2019, New Orleans, LA, USA

 ISBN 978-1-4503-6816-2/19/10…$15.00

DOI: https://doi.org/10.1145/3332165.3347876

CCS Concepts

•**Human-centered computing** → **Virtual reality;** •**Computing methodologies** → **Neural networks;** *Shape modeling;*

Author Keywords

hair, modeling, virtual reality, data-driven, machine learning, user interface

INTRODUCTION

Recent advances in modeling and rendering of digital humans have provided unprecedented realism for a variety of real-time applications, as exemplified in "Meet Mike" [33], "Siren" [34], and "Soul Machines" [1]. Compared to other human components, compelling 3D hair models are particularly challenging to create, render, and animate, due to their variety in styles and shapes.

In production, 3D hair models are still created manually with the help of advanced design softwares, such as XGen, Ornatrix, or Hairfarm. While these solutions incorporate a wide range of cutting edge tools for intuitive shape and procedural strand manipulation, they are developed for highly trained and experienced digital artists. Even for a skilled user, compelling and realistic hairstyles can easily take days or weeks to produce. Human hair is volumetric and often consists of highly intricate 3D structures, such as strands, wisps, buns, and braids, which are difficult to design with traditional 2D interfaces. 3D hair digitization and data-driven techniques can reduce the need for manual labor [27, 11, 10, 25, 19, 9], but afford limited control for real production environments.

We propose a a practical hair design system that combines the intuition and immersion of VR-based 3D interactions with the efficiency and automation of data-driven

(a) head hair gestures (b) prediction (c) facial hair gestures (d) prediction (e) result

Figure 1: *Immersive hairstyle authoring with our system.* Users can draw high-level hair gestures (green) in VR (a), based on which our system predicts the most plausible hairstyles (b). Our system can also help create facial hairs such as beards and eyebrows as shown in (c) and (d). Users can interact with the suggestions to maintain full control, including deforming hair structures and merging multiple hairstyles. The hair model produced by our system is composed of strips that can be rendered in high quality and in real-time. The final outcome (e) visualizes the underlying strips (left) and rendered hairs (right), and is completed by an novice in 10 minutes with 382 suggested and 71 manually-drawn strips. Please refer to the accompanying video for live actions.

modeling. In an immersive virtual environment, users can interactively express their intentions via natural gestures, such as brushes for braids and spins for afro-curls, and receive realistic 3D hairstyle suggestions by our system, similar to prior autocomplete techniques for 2D sketching [18, 45, 47] and 3D modeling [12, 28]. The suggested high-quality hairstyles are learned and computed from a hair model database created by professional artists. Our interface allows users to accept, modify, or ignore these suggestions to maintain full control over the design process. Figure 1 provides an example.

We represent our hair models as textured polygonal strips, which are widely adopted in AAA real-time games such as Final Fantasy 15 and Uncharted 4, and state-of-the-art real-time performance-driven CG characters such as the "Siren" demo shown at GDC 2018 [34]. Hairstrips are flexible to model and highly efficient to render [24], suitable for authoring, animating, and rendering multiple virtual characters. The resulting strip-based hair models can be also converted to other formats such as strands. In contrast, with strand-based models, barely a single character could be rendered on a high-end machine, and existing approaches for converting strands into poly-strips tend to cause adverse rendering effects due to the lack of consideration of appearances and textures during optimization.

To connect imprecise manual interactions with detailed hairstyles, we design a deep neural network architecture to classify sparse and varying number of input strokes to a matching hairstyle in a database. We first ask hair modeling artists to manually create a strip-based hair database with diverse styles and structures. We then expand our initial hair database using non-rigid deformations so that the deformed hair models share the same topology (e.g. ponytail) but vary in lengths and shapes. To simulate realistic usage scenarios, we train the network using varying numbers of sparse representative strokes. To amplify the training power of the limited data set and to enhance robustness of classification, the network maps pairs instead of individual strips into a latent feature space. The mapping stage has shared-parameter layers and max-pooling, ensuring our network scales well to arbitrary numbers of user strokes.

As each hairstyle consists of multiple geometry variations, we treat the retrieved hair models as blend-shapes [7] to fit the input strokes via a combination of global linear blending and local non-linear deformation. Instead of taking an entire hair model as blending basis, we blend at the strip level, to facilitate better expressiveness for each hair strip combinations. Following the blending operation, we perform real-time deformation to coherently propagate local details of key strips to a global scale. Our system also supports the creation of heterogeneous hairstyles by interactive merging of multiple hairstyles.

Even with a suggestive system, conventional 2D interfaces can still be difficult to create hairs due to their complex and volumetric 3D structures. We thus provide an immersive authoring interface in virtual reality (VR), which can facilitate 3D painting and sculpting for freeform design. Users can naturally model any hairstyles using a variety of brush types in 3D without physical limitations. When interacting freely in space, new challenges arise such as the difficulty to accurately perceive depth, position, and align objects [3]. We further propose new interface techniques to enable precise and intuitive interactions between the user and the hair model. We use our VR prototype for database creation, training data collection and subsequent user studies.

Experiments demonstrate that our system can save a significant amount of manual effort, while providing full degrees of freedom to create a large variation of compelling hairstyles that can be difficult to produce via

existing methods, such as ponytails (Figure 1b), braids (Figure 3), facial-hairs (Figure 1d), afro-curls (bottom row of Figure 11), hair buns, and long hair with small curls (Figure 17). Even novice users can create complex hairstyles with intricate geometry, texture, and shading in minutes that would otherwise take days for experts. The contributions of this paper are:

- An immersive and suggestive VR-based interface for intuitive and interactive hair authoring;
- A deep neural network that accurately predicts and suggests high-level hairstyles from sparse, imprecise brush strokes;
- A hair synthesis method that combines both global blending and local deformation;
- A new hair dataset with a wide diversity of hairstyles and structures that are manually created by professional artists.

RELATED WORK

Manual Hair Modeling

To generate photorealistic CG characters, sophisticated design tools have been proposed to model, simulate, and render human hair. We refer readers to [41] for an extensive overview. Manual modeling offers complete freedom in creating the desired hairstyles [13, 6, 48], but requires significant expertise and labor, especially for human hairstyles with rich and complex structures. The authoring interface needs to be easy to use (e.g. sketch-based for creation [44, 15] or posing [32]) without requiring detailed inputs such as individual strands [2] or clusters [23]. We present a system that produces realistic outputs in real-time given only a few sparse user strokes to facilitate interactive exploration.

Hair Capture

Production-level capture typically requires controlled environments, manual tuning, and sophisticated acquisition devices, e.g. multi-view stereo rigs [14, 21, 25, 27]. To popularize hair capture for end-users, existing methods offer various tradeoffs among setup, quality, and robustness, e.g. thermal imaging [17], capturing from multiple views [19] versus single view [9], or requiring different amounts and types of user inputs [11, 10, 42]. Such methods can reduce manual edits but also limit the output effects to the captured data at hand.

Despite the large body of works on hair capture, facial hair reconstruction remains largely unexplored. Facial hairs can have more varied shape, density, and length than scalp hairs. In [4], a 3D reconstruction method is proposed to recover both the geometry of sparse facial hair and its underlying skin surface. Other recent works [22, 8, 29, 38] focus on generating or editing facial hair in 2D. To the best of our knowledge, our method provides the first suggestive system for intelligent 3D modeling of both scalp and facial hairs.

Data-driven Hair Modeling

Instead of using only data captured at the current session, a database or a set of exemplars can enrich the scope and diversity of the output hairs [40, 18, 49]. Inspired by these data-driven approaches and auto-compete authoring systems [12, 45, 47, 28, 37], we provide an auto-complete hair-modeling interface that suggests potential detailed hair structures from a database based on sparse user inputs. Instead of using hand-crafted metrics (e.g. [18]) that often fail to handle the large style variations and complexity of hairstyle (e.g., ponytail vs. braid), we propose a deep neural network for database retrieval that is able to exploit high-level hairstyle features. Moreover, a deep neural network is much faster, and scales well with the dataset size, as later shown by [20].

3D Deep Learning

Recent methods such as [50, 31] applied deep learning to reconstruct hair from a single image. In particular, Zhou et al. [50] take the 2D orientation field of a hair image as input and synthesize strand features that are evenly distributed on a parameterized 2D scalp. Saito et al. [31] represent hair geometry using 3D volumetric field, which can be efficiently encoded and learned through a volumetric variational autoencoder (VAE). However, their methods are limited to hair reconstruction from images and do not offer intuitive control to modify hair details. Different from these generation networks that rely on fixed and finite-dimensional input representations, our network is only used for nearest hairstyle retrieval via the hair strokes represented as point sequences. Our works is inspired by the seminal work on point clouds recognition [30]. PointNet uses multi-layer perceptron to encode individual 3D points into feature vectors and aggregate them into a global feature vector by max pooling. Though simple, the architecture has nice properties of being invariant to the number or order of points.

Modeling in VR

Even with the assistance of data and/or machine learning, traditional 2D interfaces such as [18] present inherent challenges for authoring 3D content, especially those with intricate 3D structures such as human hair buns, knots, and wisps. Recent advances in VR modeling [39, 16, 26] have offered a new venue for interactive 3D modeling. However, none of existing VR platforms supports direct editing on hair geometry, not to mention an automated system which can predict and complete a complex hair model from sparse painting strokes. We introduce the first hair modeling tool that leverages the unprecedented authoring freedom in VR and provides intelligent online hairstyle suggestion and manual authoring assistance.

DESIGN GOALS

To build a powerful, flexible, and easy to use hair authoring system, we have the following design goals in mind:

Wide audience Both novice and professionals can easily and quickly create high quality hair models with our system.

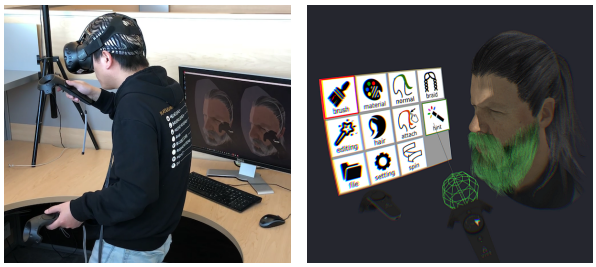
Flexible input Only high-level, sparse input gestures are required to indicate intended hairstyles. Users can choose to provide more inputs for finer controls.

Assistance Based on sparse user inputs, our system interactively suggests complete hair structures, which can be composed of parts from different hair styles.

Interactivity The suggestions should be dynamically updated in real-time during user interaction and scale well to different numbers of query strips.

Immersion Complex 3D structures, such as buns, knots, and strands, should be easy to specify.

Quality output The output models should be complete and realistic, regardless of the amounts and types of user inputs.



(a) setup (b) interface

Figure 2: *System setup and user interface.* One controller is used as the brush for freeform drawing, and the other as a UI panel from which the users can pinpoint to select different tools and change parameters (e.g. brush, color, and models).

USER INTERFACE

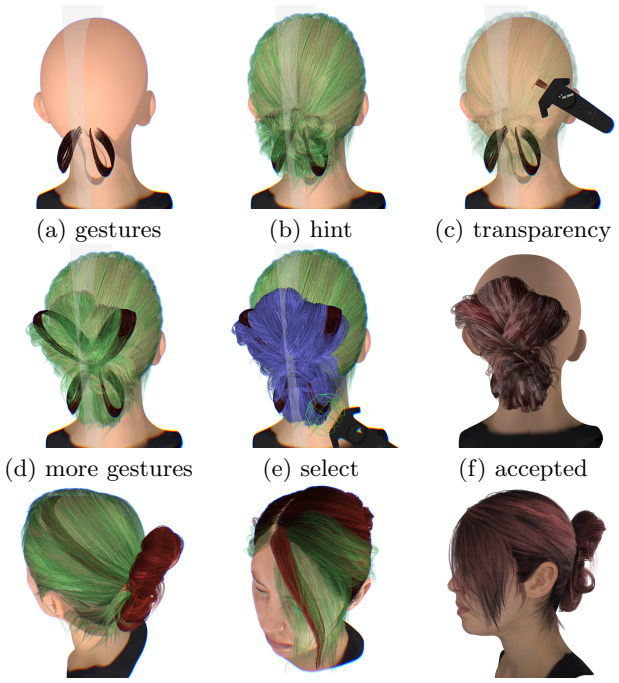
To meet the design goals in Section 3, we have built an assistant VR authoring system to help users easily and quickly create high-quality hair models. Users draw high-level, sparse hair strips to indicate intended hairstyles. Based on user inputs, our system interactively suggests complete, detailed, and realistic hair structures. Our interface design is inspired by systems in VR brushing (e.g., Tilt-Brush [16]) and geometry autocompletion (e.g., [12, 28]).

Basic user interaction

As shown in Figure 2b and the supplementary video, our system supports basic user interactions such as brushing strips, loading models, undo/redo, changing stroke colors and hair textures, etc. With VR handles (HTC Vive in our prototype), users can freely rotate, bend, and stretch hair strips in 3D (Figure 2a), as well as change the brush sizes and shapes via the touchpad of the freeform drawing controller.

Online modeling suggestion and hint update

As the user draws intended hair strips, our system predicts the most plausible hairstyles, rendered as transparent hints. Whenever a new user strip is detected, the

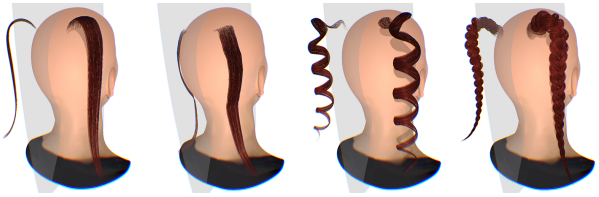


(a) gestures (b) hint (c) transparency

(d) more gestures (e) select (f) accepted

(g) back hair merge (h) top hair merge (i) merged result

Figure 3: *Examples of user interaction and system assistance.* The user starts drawing one guide strip at the lower back of the head in a mirror mode (a), and our system predicts the best matching hairstyle visualized in transparent green (b). The transparency can be changed by moving the controller closer (more opaque) or farther (more transparent) from the head (c). The user can ignore the suggestion and draw more gestures, and a different hairstyle is retrieved and deformed to fit the current set of user interactions (d). The user can accept part of the suggestion using the selection brush (e), and the accepted suggestion will be adjusted to the current brush color (f). When the user draws a new gesture on the top of the head, our system provides a new suggestion taking into account both the current guide strip and the existing bun (g). Then the user accepts the suggestions and draws another gesture in the front (h) and context-aware suggestions are updated. (i) shows the merged result.



(a) normal (b) attach (c) spin (d) braid

Figure 4: *Brush modes.* (a) shows the original gestures drawn by the user, which could have different effects under different brush modes, including the attach brush that attaches the strokes onto the scalp (b), and the spin (c) and braid (d) brushes for more complex structures.

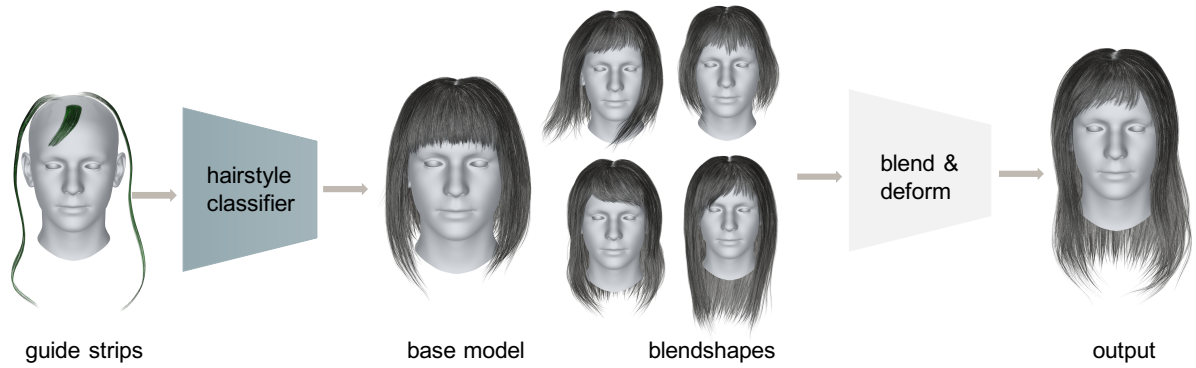


Figure 5: *System pipeline.* The user input gestures are classified by a neutral network to predict the most plausible hairstyle. Each hairstyle contains multiple blendshapes sharing the same topology while varying in shape and length. The blend shapes can be linearly blended to fit the user gestures.

interface updates and displays modeling suggestions in real time that fits the user inputs (Figures 3a and 3b). In order to distinguish user strokes from system suggestions, the hints are visualized with transparency (Figure 3c). The user can control the hint transparency by moving the brush closer to or farther from the head and move the head for different views. When the user changes the guide strips, e.g. add more or remove existing ones, the system updates the suggestion accordingly (Figure 3d). The user can press a button to activate the selection brush to partially select the suggestions or ignore the suggestions by continuing to draw (Figure 3e). If the system has achieved the desired hairstyle, the user can select all suggestions via a simple gesture to finalize the hair creation (Figure 3f).

Hybrid hairstyle

Our system supports creation of heterogeneous hairstyle by merging multiple hairstyles in a single output. Once a hairstyle is accepted or manually drawn in a local region (Figure 3g), the user can continue to draw new guide strips and the system will update the suggestions taking into account both current guide strips and previously accepted/drawn ones (Figure 3h), enabling a smooth blend of styles (Figure 3i).

Brush modes

Although VR provides the complete freedom for drawing, it can be very difficult to manually draw complicated hair structures, such as curls and braids. Thus, our system provides two special hair brushes that allow users to draw spin curves and braids with simple gestures (Figures 4c and 4d). For the spin brush, the spin size and curliness can be adjusted via the touchpad of the freeform drawing controller. The braid brush can guide our system to provide automatic suggestions of complex braids. We also provide the attach brush (Figure 4b) that automatically project the whole stroke onto the scalp surface, which is useful when creating the inner layer hair.

Manual mode

Apart from auto-completion, manual mode is also supported in our system. Users can manually draw either from scratch or on top of system suggestions. To facilitate easier drawing, our system supports *automatic anchoring* and *collision avoidance*. If automatic anchoring is activated, the strip root will be automatically anchored onto the scalp surface. Collision avoidance, if turned on, automatically pushes out the part of stroke inside the head to avoid strip-head collision. To allow symmetric drawing on both sides of a head, we also provide the mirror tool so users only need to draw on one side (Figure 3a). Users can also manually deform a strip and choose to propagate the deformation to the remaining strips to change the entire hairstyle.

METHOD OVERVIEW

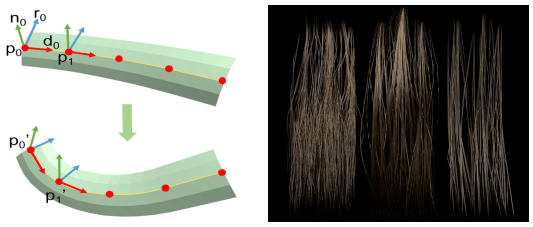
Our pipeline consists of the following steps, as illustrated in Figure 5.

Hairstyle prediction

Given sparse input user gestures as key strips, we estimate the user-intended hairstyle based on a nonlinear classifier trained by a deep neural network. The predicted hairstyle will match the input strips at a high level, providing the best-matched class label while being robust to local perturbations that may be introduced by novice users.

Hair strip generation

Our system then synthesizes the detailed hair strips of full scale that conform to the input key strips. The estimated hair class will direct the remaining algorithm steps to the corresponding set of hair templates/blendshapes (Figure 5) with the same topology but variations in size, length, and shape. We first initialize the output with a default hair model belonging to the predicted class label. For each key strip, we find the closest strip in each retrieved hair template and linearly blend the template strips to approximate the key strip. The blending coefficients are propagated to the remaining hair strips to obtain a smoothly interpolated hair model. Due to the limited expressiveness of linear blending, the result is



(a) geometry (b) texture

Figure 6: *Hair representation in geometry and texture.* (a) illustrates a hair-strip mesh, with samples shown in red and the medial axis shown in yellow. Each sample has a local coordinate frame as visualized by 3 color arrows. We can transform the samples to deform a template mesh (top) to a target geometry (bottom). (b) shows an albedo hair texture map.

prone to under-fitting. We therefore non-rigidly deform the matching strips in the output so that their geometry better matches the corresponding key strips. The deformation is again propagated to the rest of the strips to preserve local details.

Users can selectively accept the suggested hair model generated by our framework. Once selected, the hair strips will stay fixed unless they are manually deformed for further refinement. While users keep drawing in the void regions, the newly suggested hair model obeys both the new guide strokes and existing hair strips, to maintain coherence while avoiding overlap.

REPRESENTATION

We adopt a strip-based representation [24] to ease the manipulation of a large number of hair strands. With detailed texture maps and sophisticated rendering technique, a single strip can realistically depict multiple strands of hair (Figure 6b). To make the surface normal less flat, our system adopts an U-shape strip geometry (Figure 6a), where user can control the cross curvature of the strip.

Medial axis representation

We represent each hair strip with a fixed number of samples (30 in our implementation) evenly spaced on its medial axis (red dots in Figure 6). For each sample, we store both the point position p and a local coordinate system (d, r, n) (Figure 6) defined by the VR handle. The mesh geometry can be represented and reconstructed via the local frames.

Shape matching distance

Given a query strip, the system searches for its matching strip in the database that has the closest geometry. We denote the sample points of strip $\mathcal{P}_i$ and $\mathcal{P}_j$ as $\{p_m^i\}$ and $\{p_n^j\}$, respectively. We then define the shape matching distance between $\mathcal{P}_i$ and $\mathcal{P}_j$ as

$$\mathrm{d}_{SM}(\mathcal{P}_i, \mathcal{P}_j) = \Delta L \mathrm{d}(\mathcal{P}_i, \mathcal{P}_j) \quad (1)$$

$$\Delta L = 1 + \omega \frac{||l_i - l_j||}{||l_i + l_j||} \quad (2)$$

$$\mathrm{d}(\mathcal{P}_i, \mathcal{P}_j) = \sqrt{\sum_k (p_k^i - p_k^j)^2} \quad (3)$$

, where all coordinates of $\{p_m^i\}$ and $\{p_n^j\}$ are in the global space (with origin at the head center).

In Equation (2), l_i stands for the length of i-th strip $\mathcal{P}_i$ while ω is a scaling factor that is set as 3.0 in our implementation. The value of ΔL equals to 1.0 if $\mathcal{P}_i$ and $\mathcal{P}_j$ have the same length, and increases as the discrepancy between l_i and l_j increases. $\mathrm{d}(\mathcal{P}_i, \mathcal{P}_j)$ measures the piecewise distance between the corresponding points of $\mathcal{P}_i$ and $\mathcal{P}_j$. In sum, d_{SM} penalizes the case where there is a large difference between the lengths or shapes of the input strips.

DATABASE CONSTRUCTION

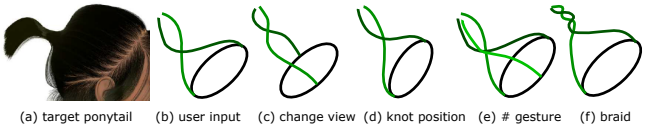


Figure 7: *Hair gestures example.* In order to create a ponytail in (a), user could draw some gestures (green lines), from which our system extracts high-level features for hairstyle retrieval. The high-level features should be insensitive to viewpoints (b, c), knot position (b, d), and gestures number (b, e), while being discriminative for different hairstyles, e.g. ponytail and braid (b, f). Image courtesy of [15] in (a).

We construct three separate databases for scalp hairs, beards and mustaches, and eye brows. The *scalp* hair database $\mathcal{D}$ contains 30 different styles $\{\mathcal{H}_i\}$ with various hair length (long, middle and short), hairline (left, right, middle or none), and hair parts (bun and ponytail). For each hairstyle $\mathcal{H}_i$, our collaborating artist created a base model $\mathcal{H}_i^0$ via our system (about 23 minutes), and expanded it into 10 more variations $\mathcal{H}_i^j, j = 1, 2, ..., 10$, via our deformation tool (around 55 minutes). These variations serve as the blend-shapes of our method as depicted in Figure 5. We further augment each hair model $\mathcal{H}_i^j$ with up to 5 different amounts of curliness. For example, our three results in Figure 14 (bottom) and Figure 15 share the same style but different curliness (wavy, straight, and braid). Since *facial* hairs tend to have simpler structures than scalp hairs, we created 5 different styles for both the beard/mustache and eyebrow databases, and each hairstyle consists of 5 different variations.

Hair models $\{\mathcal{H}_i^j\}$ that belong to the same hairstyle $\mathcal{H}_i$ share the same topology: 1) each strip in one model $\mathcal{H}_i^j$ can always find its corresponding strip in any other

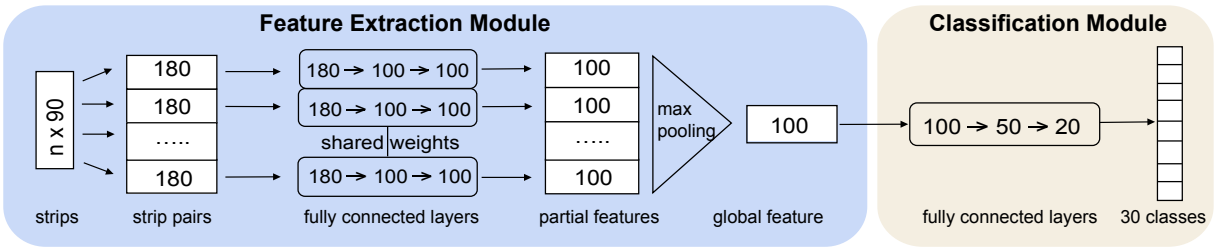


Figure 8: *Network architecture.* Given n sparse strips that depict a hairstyle, our network encodes them into a global feature vector (blue color), followed by classifying the global feature to get the probability of different hairstyles (yellow color).

models within the class label i; 2) corresponded strips share the same root point on the scalp.

Hair models typically consist of multiple layers of strips where the outer layers dominate the overall appearance. We thus manually label the hair strips into outer and inner layers and further accelerate the algorithm by only matching strips on the outer layers during query. UI-wise, users only need to draw outer strips while our system can automatically generate globally coherent inner layers.

To learn how ordinary users sketch key strips, we asked 31 participants (10 females and 21 males with 5 of them being experienced drawing artists) to manually segment each hairstyle $\mathcal{H}_i$ into 5 representative regions. The segmentation is utilized to extract a sparse set of representative hair strips for learning high-level hairstyle features (Section 8).

HAIRSTYLE PREDICTION

We need to bridge the gap between sparse strips $\{\mathcal{P}_i\}$ (Section 6) and complete hair models $\{\mathcal{H}_j\}$ (Section 7). Directly measuring the geometric difference using hand-crafted metric, e.g. Hausdorff distance, is expensive and sensitive to large hairstyle variations (e.g., ponytail vs. braid). To resolve this issue, we propose to compare their similarity in a latent feature space, which is more robust and invariant to spatial locations, drawing orders, and strip numbers (Figure 7). Towards this end, we adopt deep neural network for feature extraction due to its robustness to outliers, ability to exploit high-level hairstyle features, and computational efficiency at run time that scales well with the data size.

Our network architecture is illustrated in Figure 8. The input is a vectorized representation of strip geometry – the concatenated 3D coordinates of the sample points, and the output is the class label of the corresponding hairstyle. The network consists of two main parts: a feature extraction module that maps the input to a non-linear global feature, followed by a classification module which predicts the probability of different hairstyles. We elaborate on two components that are critical for robust and accurate prediction.

Sparse strips as training data

To resemble the real application scenario, we only extract a random number (ranging from 1 to 10) of sparse strips from each hair model during training. While some users sketch a hairstyle with the most representative strokes spread out on the head, others draw dense strips locally before moving to the next area. To consider both cases, we asked different users to segment the entire set of hair strips into 5 representative clusters (Section 7), with each cluster containing at most 10 nearest strips. The training data are generated by sampling from these clusters in a combinatorial way. As a result, large amount of training data can be produced from limited hair models.

Extract features from paired strips and apply max pooling

Instead of analyzing individual strips, we pair up every two strips to obtain $\binom{n}{2}$ (n is the number of hair strips sampled from each hair model) strip pairs $\{\mathcal{P}_i, \mathcal{P}_j\}$ and feed them into two fully connected layers with shared parameters. When only one strip is available, we duplicate it to form a strip pair. Partial features of the hairstyle will be extracted from each strip pair by mapping it to a fixed-size feature vector $\mathcal{F}_{i,j}$. In order to handle arbitrary number of strips, we aggregate all the partial features $\{\mathcal{F}_{i,j}\}$ via a max-pooling layer to obtain a global feature $\mathcal{F}$ of the hairstyle, similar to PointNet [30]. The rationale behind the use of strip pair as the basic feature is that it can capture more structural information than individual strips (as illustrated in Figure 7) and yet achieve comparable accuracy with denser strip groupings (e.g. 3 or more) with less computational cost.

Network details

Each input strip is represented with 30 uniformly sampled points, leading to a 180-dimensional ($30 \times 3 \times 2$) vector for each strip pair. The classification module consists of two fully-connected layers and a softmax layer that outputs a 30-dimensional vector which encodes the probability of each hairstyle.

Run time usage

At run time, up to the 6 latest strips are fed into the network for hairstyle prediction. The prediction accuracy of our network is 85% for top-1 and 94% for top-5 classification. Once users accept system suggestions, existing guide stripes will be removed. As our system provides

Figure 9: The hair models of AAA game characters (top: "The Last of Us", bottom: "The Witcher 3") took a professional artist several days to place the initial hair strips and 2 to 3 weeks to refine the geometry and texture, while our system could create the comparable hairstyle and quality much faster (top: 38 minutes using pure manual drawing, bottom: 6 minutes using suggestive drawing, for a non-professional user).

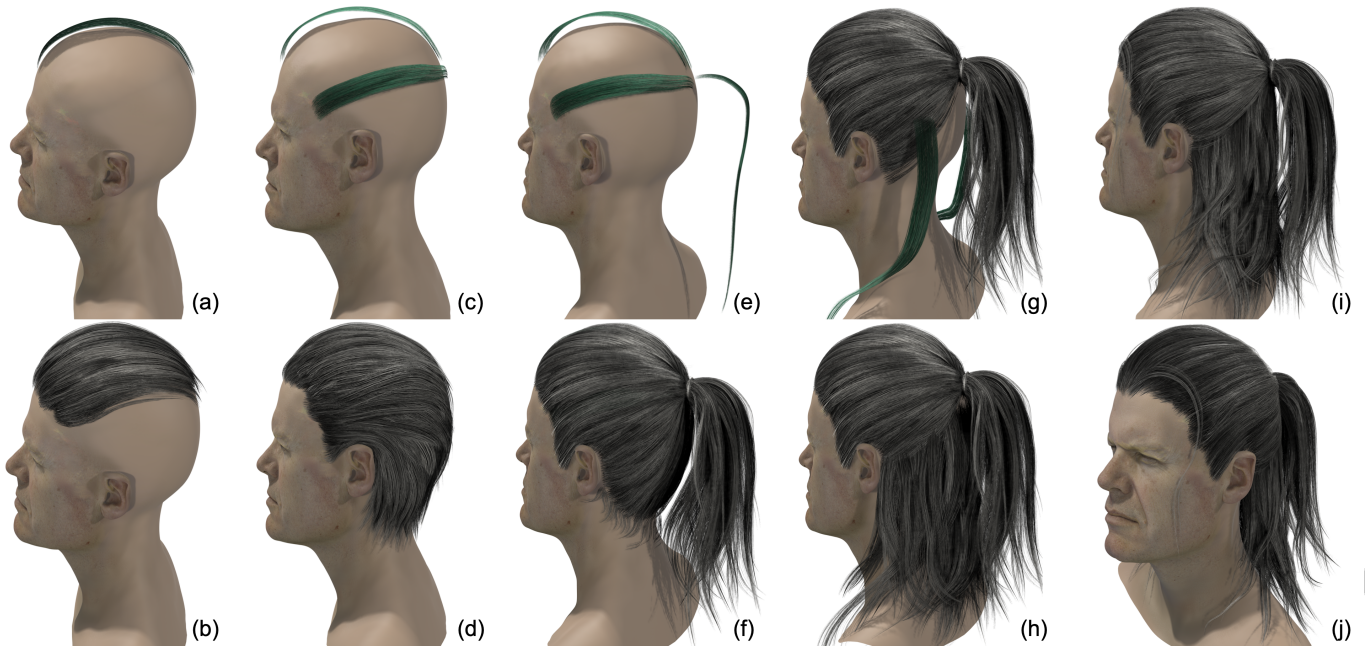


Figure 10: *Authoring process and intermediate results of our system.* (a) User first draws a guide strip (shown in green) on top of the head, and our system provides online hairstyle suggestion that best matches the input guide strip (b). User can ignore the suggestion by continue drawing (c), and the suggestion will be updated accordingly in realtime (d). Such process is iterated until satisfactory suggestion is obtained (e, f). User can partially accept the suggestion and continue such interaction (g), and the new suggestion will be aware of the existing strips (h). Users can also manually draw more strips, or delete and deform the existing strips to achieve desired appearance (i,j).

real-time feedback, users could accept system suggestion if they find it appropriate. Therefore discarding older operations enables the algorithm to provide more accurate updates according to the latest user inputs.

FULL-SCALE HAIR GENERATION

With the hairstyle predicted by the network, our system then generates full hair details: blend and deform variations from the predicted hairstyle to fit user strokes, repeat these steps to merge multiple hairstyles, and reconstruct full hair geometry from the hair samples.

Blending

Given the target hairstyle $\mathcal{H}_i$ classified by the network in Figure 8, we use its corresponding hair models $\{\mathcal{H}_i^j\}$ (Section 7) as "blendshapes" [7] to fit the input key strips $\{\mathcal{P}_\ell\}$. The method in [7] treats each complete face model as a blend shape. Applying the same approach by treating entire hair meshes as linear bases does not work well as hairstyles can have more shape variations than faces. We thus perform local blending at the hair-strip level for better expressiveness and accuracy. The hair models $\{\mathcal{H}_i^j\}$ from the same hairstyle $\mathcal{H}_i$ have the same size and topology. Therefore, we represent each strip in the generated model $\mathcal{V}$ as a linear combination of its corresponding strips in $\{\mathcal{H}_i^j\}$ and optimize the blend weights such that the resulting hair model matches the key strips well. The steps are as follows:

Strip matching

The purpose of strip matching is to find a suitable blending basis for each key strip $\mathcal{P}_\ell$. Every strip in $\mathcal{H}_i^j$ has corresponding strips in all other hair models that belong to the same hairstyle $\mathcal{H}_i$. Suppose hairstyle $\mathcal{H}_i$ has n strips. Then $\mathcal{H}_i$ has n blending bases: $\{c_j\}_{j=0,1,...,n-1}$, where $c_j = \{c_j^k\}_{k=0,1,...,m-1}$ are the individual strips in basis c_j and m is the number of hair models belonging to style $\mathcal{H}_i$. For each key strip $\mathcal{P}_\ell$, we search for its closest hair strip $\mathcal{M}_i$ from the retrieved hair models $\{\mathcal{H}_i^j\}$. The similarity between the strips is measured using the shape matching metric d_{SM} defined in Equation (1). Hence, the strip with the lowest d_{SM} to $\mathcal{P}_\ell$ will be considered as its matching strip. For $\mathcal{P}_\ell$, if its matching strip $\mathcal{M}_i$ belongs to the j-th basis $\{c_j^k\}$, then $\{c_j^k\}$ will become $\mathcal{P}_\ell$'s blending basis in the following steps. The matching process could be accelerated via *kd*-tree search. As discussed in Section 7, we only search for the matching strip in the outer layer of hairstyle.

Key strips fitting

Given a key strip $\mathcal{P}_\ell$ and its blending basis $c_i = \{c_i^k\}$, we first calculate the mean shape B_i of $\{c_i^k\}$ and then compute the displacement vector d_i^k as follows:

$$d_i^k = c_i^k - B_i \tag{4}$$

Hence, each key strip $\mathcal{P}_\ell$ can be approximated as linear combination of the displacement vectors and the mean shape:

$$\mathcal{P}_\ell \simeq t_i \sum_k \alpha_i^k d_i^k + s_i B_i \tag{5}$$

$\alpha_i = \{\alpha_i^k\}$ are the blending coefficients; t_i and s_i are positive scaling factors for linear blending and mean shape, respectively. We introduce t_i and s_i to enhance the expressiveness and smoothness of linear model. As the linear model restricts the coefficient to stay between 0 and 1 to avoid unstable extrapolations, it can under-fit large variations and produce non-smooth, rigid results. The additional degrees of freedom would help smooth the outcome geometry while providing more capability in representing largely deformed structures.

We optimize the set of weights, t_i, α_i and s_i, to minimize the following fitting error using real-time L-BFGS solver:

$$\begin{aligned} &\arg\min_{t_i,\alpha_i,s_i} \left\| \mathcal{P}_\ell - \left(t_i \sum_k \alpha_i^k d_i^k + s_i B_i \right) \right\| \\ &\text{s.t.} \quad \alpha_i^k \in [0,1], \quad \sum_i \alpha_i^k = 1 \end{aligned} \tag{6}$$

Coefficient propagation

After key strip fitting, part of the blending bases $\{\{c_0^k\}, \{c_1^k\}, ..., \{c_{n-1}^k\}\}$ has received blending coefficients, producing corresponding strips in the fitting result $\mathcal{V}$. We then propagate the computed coefficients to the remaining bases.

Each basis c_j has three different sets of coefficients: t_j, α_j and s_j, which are propagated using the same formula in Equation (7). In particular, suppose $\{c_i\}$ is the set of bases that have received blending coefficients. Then the coefficients for any other basis w_j can be calculated based on the spatial distance and shape similarity between B_i and B_j:

$$w_j = \sum_i e^{-\frac{\mathrm{d}_{SM}(B_i,B_j)}{\sigma}} w_i \tag{7}$$

where d_{SM} is the distance function defined in Equation (1); B_i and B_j are the mean shapes; w_i and w_j are the corresponding coefficients (i.e. $w_j = \{t_j, \{\alpha_j^k\}, s_j\}$); σ is a constant, which is set to 0.08 for distance normalization. The coefficient propagation can be computed efficiently since $\mathrm{d}_{SM}(B_i, B_j)$ can be pre-computed after the database construction.

After all the blending coefficients are solved, each strip $\mathcal{V}_j$ of a full-scale hair model $\mathcal{V}$ can be calculated by applying the linear model in Equation (5).

Deformation

Although linear blending provides robust and smooth fitting to the key strips, it is prone to under-fit due to limited variations in the linear bases. Therefore we further deform the blending result towards key strips. Given the key strips $\{\mathcal{P}_i\}$ and the linear blending results $\{\mathcal{V}_i\}$, we compute the target displacement for each $\mathcal{V}_i$ as:

$$\Delta \mathrm{d}_i = k_i(\mathcal{P}_i - \mathcal{V}_i) \tag{8}$$

$\Delta \mathrm{d}_i$ consists of point-wise translation vectors, and k_i is a weight that goes from 0 (the hair root) to 1 (the hair tip) smoothly, making sure the hair roots are always fixed. We then propagate the displacement vectors computed at the $\{\mathcal{V}_i\}$ to the remaining strips, analogous to the coefficient propagation for blending in Equation (7).

Intuitively, $\mathcal{V}_i$ is deformed to resemble $\mathcal{P}_i$ while maintaining its original details. Compared to optimation-based mesh deformation [36, 35], this simple one-pass method deforms hair strips in real time with sufficient quality.

Merging

In addition to fitting a single hairstyle , our system can also merge multiple hairstyles as exemplified in Figure 3h. To keep track of multiple hairstyles, we build a volumetric field to represent the occupancy of hair strips in 3D. We initialize a dense volumetric grid that is large enough to encompass any hair strip anchored on the scalp with all cell values set to 0. For each sample point from an existing hair strip, we set the occupied grid cell to 1, and propagate the occupancy to its neighboring cells using a Gaussian kernel. For the newly suggested hair model, we remove those strips that overlap with the existing strips, e.g. with more than 60% of its points locate in grid cells with positive occupancy values.

Hair Geometry Reconstruction

We first resolve the strip-head collision. To accelerate the computation, we pre-compute a dense volumetric levelset field with respect to the scalp surface. For each grid cell, its direction and distance to the scalp are also pre-computed and stored. During run time, samples inside the head are detected and projected back to their nearest points on the scalp, and samples outside the head remain fixed.

With the calculated sample locations, we proceed to reconstruct the full geometry of hair details. In particular, we build the Bishop frame [5] for each strip, and use the parallel transport method to calculate the transformation at each sample point. The full output hair mesh could be easily transformed via linear blending of these sample transformations.

EVALUATION

Our system is able to model highly complex and diverse hairstyles, such as curly hairs (Figure 14), ponytails (Figure 10), braids (Figure 11), afro braids (Figure 15), buns (Figures 3 and 17), and beard (Figures 1 and 9). These are extremely difficult or time-consuming to create from scratch using existing modeling techniques (e.g., [23, 15, 18, 20, 31]) and commercial systems (e.g., XGen, Ornatrix). We evaluate our system via sample outcomes, comparisons with prior methods, and a pilot user study.

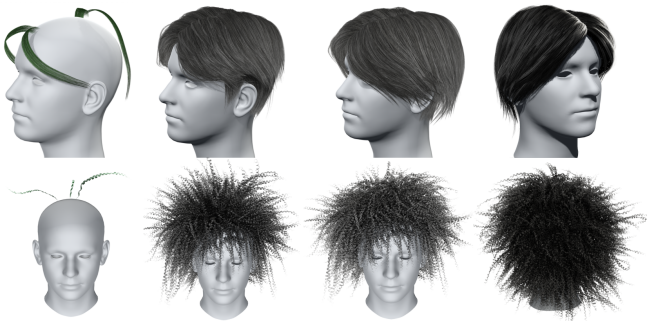
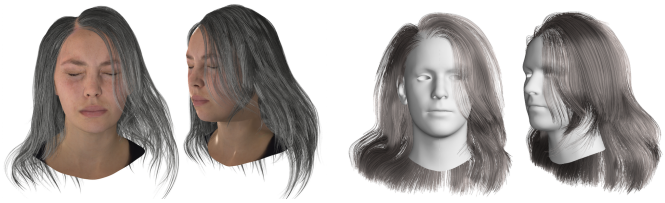


Figure 11: *Results of network prediction and our hairstyle auto-complete.* From left to right: input guidance strips, suggested default hairstyle (network output), deformed and blended result according to key strips in two different views.



(a) strip model (b) strand model

Figure 12: *Our system supports converting strip- to strand-based representations.*

Results and Analysis

We show how our system can help author high-quality hair models with large variations. Figure 11 shows the modeling results created with very sparse set of guide strips. We first validate the performance of hairstyle prediction. As seen from the first and second column, our prediction network is capable of capturing high-level features of input strips, such as hair length and curliness. We then present the effect of blending and deformation in fitting the hair models to the input key strips (third column). Although the base model (second column) matches the query strips at a high level, details deviate from the user's intentions. The proposed blending and deformation algorithm produces results with realistic local details better following input guide strokes.

We evaluate the merging operation in Figure 10. Given a partially accepted hair model, users can create a heterogeneous hairstyle by either drawing in a different style, e.g. invoking more curliness, or changing the hair topology, e.g. adding a hair knot. On top of the merged result, users can further refine the hair model with manual operations, e.g. bridge the topology transition or add/remove hair strips. Figure 12 shows that our strips can be converted to strand-based representation for rendering. In particular, we compute the orientation field based on the

hair strips, and grow the hair strands from the fixed roots [18].

Comparisons

We first compare our system with the professional hair modeling tools, such as Maya XGen, to demonstrate that our system could be applied to real-world AAA game hair asset creation. We then compare our method with a wide range of prior techniques: semi-automatic image-based hair modeling [18], and fully automatic hair reconstruction approaches [20, 31]. While previous hair modeling approaches mainly focus on generating hair models from 2D images, our framework enables hair creation and authoring in 3D space. Since none of these previous systems can handle facial hairs, we only compare scalp hair authoring.

Hu et al. [18] ours

Figure 13: *Comparison with [18].* The blue and green color of our guide strips (third column) indicates two suggestion selection sessions. The results by the 2D method [18] may lack realistic 3D structures. For each example, [18] took around 1 minute to draw and 2 minutes to process, and our system took 5/12 (top/bottom) minutes to complete (including manual refinement).

Professional tools

In Figure 9, we compare the scalp and facial hair models generated using our system with that of the modern hair modeling software deployed in most industrial studios. For AAA games, one hair model could easily take days for a professional artist to simply place the strips, as manipulating the mesh geometry (vertices, edges, and faces) could be extremely tedious in a 2D interface. It could take more days or even weeks to iterate the geometry and texture refinement before reaching a satisfying rendering. In contrast, our VR tool provides the full expressiveness to directly place the strips in 3D, as well as realtime immersive feedback of the final rendering. Even for non-professional users, they could create a comparable hairstyle within an hour using pure manual drawing in VR (top row). With system suggestion turned on, the authoring time can be further reduced to only a few minutes (bottom row).

Hu et al. [20] ours

Figure 14: *Comparison with [20].* The top case of [20] has a wrong style while the bottom case lacks detailed curls. The processing time is around 2 seconds for [20] and 2 minutes for our system (without manual refinement).

Sketching from photos

We also compare our approach with the semi-automatic image-based hair modeling algorithm [18] (Figure 13). As driven by a hair database, their approach retrieves the best matched hair examples based on the reference image and the user strokes drawn on 2D. However, due to the ambiguity of 3D projection and the view occlusion, the 2D strokes may not faithfully reflect the 3D structures. Our method, in contrast, allow users to directly sketch key hair strips in 3D, enabling more accurate retrieval of hairstyle and higher quality of detailed hair geometry, as demonstrated in Figure 13. Moreover, the output of our blendshape method always falls in the plausible space, while the deformation-based method may cause artifacts.

Synthesis from photos

We further compare our algorithm with the state-of-the-art automatic hair reconstruction approach [20] in Figure 14. To bridge the discrepancy of inputs, we asked a novice user to draw sparse guide strips that best represent the hairstyle in the input image. Our result is then automatically generated without any manual refinement. As shown in Figure 14, though Hu et al. [20] can generate a hair geometry that roughly matches the global shape of the hair in the input, it fails to reproduce the fine details. The deviation of local details may be either due to an inaccurate retrieval of hairstyle and deformation (first row of Figure 14) or the limited capability of dataset (second row of Figure 14). We also compare our approach with [31], a more recent deep learning-based method for reconstructing 3D hair from a single image. Though [31] is highly robust to the variations of inputs, their method also fail to capture local details as manifested in the second result of Figure 15. In contrast to both approaches, our approach offers significantly more accurate approximation of the input image allowing users to create sophisticated hairstyles with just a few gestures.

Saito et al. [31] ours

Figure 15: *Comparison with [31].* The results by [31] may not capture sufficient details from the input images. The processing time is about 1 minute for [31] and 3/4 (top/bottom) minutes for our system (without manual refinement).

User study

(a) reference (b) manual (c) ours

Figure 16: *User study drawing session.* From the reference image (a), a participant manually created the result in (b) in 18 minutes, and the result in (c) using our suggestive system in 5 minutes.

Figure 17: *More hair results created by users with our system.* Authoring time (minutes) from left to right: 5, 6, 4, and 8.

We conducted a preliminary user study to evaluate the usability of our suggestive system. We recruited 1 experienced hair modeling artist and 8 novice users with different levels of sketching experiences as participants.

Procedure

The study consisted of three sessions: warm-up (10 min), target session (60 min), open session and interview (20 min). For the target session, the participants were given a reference portrait image (loaded into VR) and asked to create the hair model via (1) manual drawing, and (2) suggestive modeling in our VR system (Figure 16). The orders of these three conditions were counter-balanced among all participants. For the open session, the goal is to let the participants explore the full functionality of our system and uncover potential usability issues.

Outcome

We measured the completion time and stroke counts for the target session. The results show that using our suggestive system could save both the authoring time (average/min/max time: 6/3/10 minutes) and number of strokes (average/min/max: 22/11/32 strokes), compared to the manual drawing (average/min/max time: 18/15/25 minutes, average/min/max: 71/58/95 strokes). The reported time of our suggestive approach includes both the manual drawing and editing operations as refinement. Without refinement, the users can create a desired hairstyle in just 2 to 4 minutes with 2 to 8 guide strips. The total stroke counts include those undone and deleted by the users.

Feedback

Overall, the participants found our system novel and useful, and liked the auto-complete function of our system. The participants reported the real-time suggestion was gratifying and accurate, which provided them helpful visual guidance for VR drawing. The artist pointed out that controlling and adjusting the strip position and angle is very time-consuming in professional hair modeling software like Maya XGen. Our VR-based system can provide more freedom and easiness to achieve similar operations. The artist also suggested integrating our system with professional modeling tools for smooth authoring experience and instant preview of the final rendering effects.

Figure 16 shows the results of our study participants. With manual drawing, the participants have complete freedom in drawing, and thus are able to author the desired details. However, it is difficult for people without VR drawing experience to create an accurate hair model from scratch, even when a reference image is loaded in VR. Our interactive system, on the other hand, suggests complete hair models to match sparse user strokes, and allows merging and deformation of different hairstyles at ease. The hair models in Figure 17 were created by our participants in the open session.

VR interface

Both the VR interface and the autocomplete function of our system can help improve usability. We omitted

the evaluation of the VR interface against traditional desktop interfaces, as the benefits of VR interaction are not unique to this work and have been demonstrated in other platforms such as VR brushing and painting. Instead, we asked three professional hair modeling artists about the efforts to create our paper results using their desktop tools, and all of them told us achieving the same complexity and quality of strip placement would take them more than 2 days on average. The VR interaction aspect of our interface has only been evaluated with anecdotal evidence, and we leave full evaluation as a future work.

LIMITATIONS AND FUTURE WORK

Since our system could be used for efficient and high-quality strip-based hair modeling, it has practical value for the scalable production of games and immersive content. Our results may not be realistic enough for main characters in some high-end AAA game titles, which usually take several months for a single hairstyle (e.g., Uncharted 4), our system can quickly generate reasonably high-fidelity hair models with superior quality than in game characters such as GTA V. We also argue that for secondary and crowd characters, there is no viable alternative solution for efficient polystrip modeling. Application-wise, our system is also useful for rapid prototyping tasks and concept design during pre-production, and can achieve faster turn-around times for asset creation. Our system also provides a solution to generate polystrip hairstyles for large number of real-time characters under a tight deadline. As far as we know, there is no other approach that can handle this.

We would like to emphasize that our system is designed as an extension, rather than a replacement, for the professional desktop tools. For example, the initial hair models could also be exported into 3D modeling tools, like Maya, for further refinement. Using our models as priors can significantly increase the production speed.

(a) reference (b) parts (c) result

Figure 18: *Failure case*. Our system may not merge between two very different hair parts, as visualized with green (the braids) and brown (the rest hair) colors in (b).

In traditional strip-based hair modeling, texture and geometry are highly coupled. In this paper, we mainly focus on the modeling of hair geometry, but leave the optimization of hair textures less explored. Even so, we demonstrate that our system can greatly accelerate the modeling and designing of complex hairstyles, enabling realistic hair modeling in minutes. The strip-strip intersection is not explored in our work as it is highly depended on the texture. For instance, in professional strip hair modeling, strip-strip intersections are often utilized to create volumetric effects. Investigating how to arrange hair geometries based on the texture to produce more realistic rendering would be an interesting future work. Integrating our system with the professional hair modeling tools like XGen to provide a smooth authoring experience would also be of interest.

In our system, a strip cannot change between different types, such as from a flat strip to a braid strip. This may cause artifacts at the connection area of different hair parts, as shown in Figure 18. One future work is to allow fine-level control of the strip geometry.

In games, rigging via bones/joints is a popular approach for animating hair strips, often combined with some efficient simulation techniques. Simulating complex hair strip animations and collisions are somewhat less explored. A potential direction is to add efficient secondary animation authoring [46, 43] with our hair geometry modeling.

As a data-driven approach, the quality and expressiveness of our results are limited by the variations and capacity of the dataset. Preparing a high-quality dataset with large variations in both geometry and texture still requires huge amount of artistic effort, especially for curly hairs.

ACKNOWLEDGMENTS

This research was funded by in part by Adobe, the ONR YIP grant N00014-17-S-FO14, the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, the Andrew and Erna Viterbi Early Career Chair, the U.S. Army Research Laboratory (ARL) under contract number W911NF-14-D-0005, and Sony. Hao Li is affiliated with USC, USC/ICT, and Pinscreen. This project was not funded by nor conducted at Pinscreen. Koki Nagano is affiliated with Pinscreen but worked on this project through his affiliation at USC/ICT. We would like to thank Aviral Agarwal for his help and professional advice on hair modeling, Liwen Hu for providing us the code for strip-to-strand conversion, Emily O'Brien and Mike Seymour for the scanned head models, and the anonymous reviewers for their valuable suggestions. The content of the information does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred.

REFERENCES

[1] 2017. Soul Machines. (2017). https://www.soulmachines.com/.

[2] Rıfat Aras, Barkın Başarankut, Tolga Çapın, and Bülent Özgüç. 2008. 3D Hair sketching for real-time

dynamic & key frame animations. *The Visual Computer* 24, 7 (2008), 577–585.

[3] Rahul Arora, Rubaiat Habib Kazi, Fraser Anderson, Tovi Grossman, Karan Singh, and George Fitzmaurice. 2017. Experimental Evaluation of Sketching on Surfaces in VR. In *CHI '17*. 5643–5654.

[4] Thabo Beeler, Bernd Bickel, Gioacchino Noris, Paul Beardsley, Steve Marschner, Robert W Sumner, and Markus Gross. 2012. Coupled 3D reconstruction of sparse facial hair and skin. *ACM Transactions on Graphics (ToG)* 31, 4 (2012), 117.

[5] Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. 2008. Discrete Elastic Rods. *ACM Trans. Graph.* 27, 3, Article 63 (2008), 12 pages.

[6] Florence Bertails, Basile Audoly, Marie-Paule Cani, Bernard Querleux, Frédéric Leroy, and Jean-Luc Lévêque. 2006. Super-helices for Predicting the Dynamics of Natural Hair. *ACM Trans. Graph.* 25, 3 (2006), 1180–1187.

[7] Volker Blanz and Thomas Vetter. 1999. A Morphable Model for the Synthesis of 3D Faces. In *SIGGRAPH '99*. 187–194.

[8] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. 2016. Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093* (2016).

[9] Menglei Chai, Tianjia Shao, Hongzhi Wu, Yanlin Weng, and Kun Zhou. 2016. AutoHair: Fully Automatic Hair Modeling from a Single Image. *ACM Trans. Graph.* 35, 4, Article 116 (2016), 12 pages.

[10] Menglei Chai, Lvdi Wang, Yanlin Weng, Xiaogang Jin, and Kun Zhou. 2013. Dynamic Hair Manipulation in Images and Videos. *ACM Trans. Graph.* 32, 4, Article 75 (2013), 8 pages.

[11] Menglei Chai, Lvdi Wang, Yanlin Weng, Yizhou Yu, Baining Guo, and Kun Zhou. 2012. Single-view Hair Modeling for Portrait Manipulation. *ACM Trans. Graph.* 31, 4, Article 116 (2012), 8 pages.

[12] Siddhartha Chaudhuri and Vladlen Koltun. 2010. Data-driven Suggestions for Creativity Support in 3D Modeling. *ACM Trans. Graph.* 29, 6, Article 183 (2010), 10 pages.

[13] Byoungwon Choe and Hyeong-Seok Ko. 2005. A statistical wisp model and pseudophysical approaches for interactive hairstyle generation. *IEEE Transactions on Visualization and Computer Graphics* 11, 2 (2005), 160–170.

[14] Jose I Echevarria, Derek Bradley, Diego Gutierrez, and Thabo Beeler. 2014. Capturing and stylizing hair for 3D fabrication. *ACM Transactions on Graphics (ToG)* 33, 4 (2014), 125.

[15] Hongbo Fu, Yichen Wei, Chiew-Lan Tai, and Long Quan. 2007. Sketching Hairstyles. In *SBIM '07*. 31–36.

[16] Google. 2016. Tilt Brush. (2016). `https://www.tiltbrush.com/`.

[17] Tomas Lay Herrera, Arno Zinke, and Andreas Weber. 2012. Lighting hair from the inside: A thermal approach to hair reconstruction. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 146.

[18] Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. 2015. Single-view Hair Modeling Using a Hairstyle Database. *ACM Trans. Graph.* 34, 4, Article 125 (2015), 9 pages.

[19] Liwen Hu, Chongyang Ma, Linjie Luo, Li-Yi Wei, and Hao Li. 2014. Capturing Braided Hairstyles. *ACM Trans. Graph.* 33, 6, Article 225 (2014), 9 pages.

[20] Liwen Hu, Shunsuke Saito, Lingyu Wei, Koki Nagano, Jaewoo Seo, Jens Fursund, Iman Sadeghi, Carrie Sun, Yen-Chun Chen, and Hao Li. 2017. Avatar Digitization from a Single Image for Real-time Rendering. *ACM Trans. Graph.* 36, 6, Article 195 (2017), 14 pages.

[21] Wenzel Jakob, Jonathan T. Moon, and Steve Marschner. 2009. Capturing Hair Assemblies Fiber by Fiber. *ACM Trans. Graph.* 28, 5, Article 164 (2009), 9 pages.

[22] Ira Kemelmacher-Shlizerman. 2016. Transfiguring portraits. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 94.

[23] Tae-Yong Kim and Ulrich Neumann. 2002. Interactive Multiresolution Hair Modeling and Editing. *ACM Trans. Graph.* 21, 3 (2002), 620–629.

[24] Chuan Koon Koh and Zhiyong Huang. 2000. Real-time animation of human hair modeled in strips. In *Computer Animation and Simulation 2000*. 101–110.

[25] Linjie Luo, Hao Li, and Szymon Rusinkiewicz. 2013. Structure-aware Hair Capture. *ACM Trans. Graph.* 32, 4, Article 76 (2013), 12 pages.

[26] Oculus. 2016. Quill. (2016). `https://www.oculus.com/story-studio/quill/`.

[27] Sylvain Paris, Will Chang, Oleg I. Kozhushnyan, Wojciech Jarosz, Wojciech Matusik, Matthias Zwicker, and Frédo Durand. 2008. Hair Photobooth: Geometric and Photometric Acquisition of Real Hairstyles. *ACM Trans. Graph.* 27, 3, Article 30 (2008), 9 pages.

[28] Mengqi Peng, Jun Xing, and Li-Yi Wei. 2018. Autocomplete 3D Sculpting. *ACM Trans. Graph.* 37, 4, Article 132 (2018), 15 pages.

[29] Tiziano Portenier, Qiyang Hu, Attila Szabó, Siavash Arjomand Bigdeli, Paolo Favaro, and Matthias Zwicker. 2018. Faceshop: Deep Sketch-based Face Image Editing. *ACM Trans. Graph.* 37, 4, Article 99 (2018), 13 pages.

[30] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *CVPR '17*. 652–660.

[31] Shunsuke Saito, Liwen Hu, Chongyang Ma, Hikaru Ibayashi, Linjie Luo, and Hao Li. 2018. 3D Hair Synthesis Using Volumetric Variational Autoencoders. *ACM Trans. Graph.* 37, 6, Article 208 (2018), 12 pages.

[32] Shogo Seki and Takeo Igarashi. 2017. Sketch-based 3D Hair Posing by Contour Drawings. In *SCA '17*. Article 29, 2 pages.

[33] Mike Seymour. 2017. Meet Mike. (2017). https://www.fxguide.com/featured/real-time-mike/.

[34] Mike Seymour. 2018. Siren. (2018). https://www.fxguide.com/featured/epics-state-of-unreal-virtual-human-gdc-day-2-part-1/.

[35] Olga Sorkine and Marc Alexa. 2007. As-rigid-as-possible Surface Modeling. In *SGP '07*. 109–116.

[36] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. 2004. Laplacian Surface Editing. In *SGP '04*. 175–184.

[37] Ryo Suzuki, Koji Yatani, Mark D. Gross, and Tom Yeh. 2018. Tabby: Explorable Design for 3D Printing Textures. *CoRR* abs/1810.13251 (2018).

[38] Paul Upchurch, Jacob R Gardner, Geoff Pleiss, Robert Pless, Noah Snavely, Kavita Bala, and Kilian Q Weinberger. 2017. Deep Feature Interpolation for Image Content Changes.. In *CVPR '17*. 7064–7073.

[39] HTC Vive. 2017. MakeVR. (2017). https://www.viveport.com/apps/23d40515-641c-4adb-94f5-9ba0ed3deed5.

[40] Lvdi Wang, Yizhou Yu, Kun Zhou, and Baining Guo. 2009. Example-based Hair Geometry Synthesis. *ACM Trans. Graph.* 28, 3, Article 56 (2009), 9 pages.

[41] Kelly Ward, Florence Bertails, Tae-Yong Kim, Stephen R Marschner, Marie-Paule Cani, and Ming C Lin. 2007. A survey on hair modeling: Styling, simulation, and rendering. *IEEE Transactions on Visualization and Computer Graphics* 13, 2 (2007).

[42] Yanlin Weng, Lvdi Wang, Xiao Li, Menglei Chai, and Kun Zhou. 2013. Hair interpolation for portrait morphing. *Computer Graphics Forum* 32, 7 (2013), 79–84.

[43] Nora S. Willett, Wilmot Li, Jovan Popovic, Floraine Berthouzoz, and Adam Finkelstein. 2017. Secondary Motion for Performed 2D Animation. In *UIST '17*. 97–108.

[44] Jamie Wither, Florence Bertails, and Marie-Paule Cani. 2007. Realistic Hair from a Sketch. In *SMI '07*. 33–42.

[45] Jun Xing, Hsiang-Ting Chen, and Li-Yi Wei. 2014. Autocomplete Painting Repetitions. *ACM Trans. Graph.* 33, 6, Article 172 (2014), 11 pages.

[46] Jun Xing, Rubaiat Habib Kazi, Tovi Grossman, Li-Yi Wei, Jos Stam, and George Fitzmaurice. 2016. Energy-Brushes: Interactive Tools for Illustrating Stylized Elemental Dynamics. In *UIST '16*. 755–766.

[47] Jun Xing, Li-Yi Wei, Takaaki Shiratori, and Koji Yatani. 2015. Autocomplete Hand-drawn Animations. *ACM Trans. Graph.* 34, 6, Article 169 (2015), 11 pages.

[48] Cem Yuksel, Scott Schaefer, and John Keyser. 2009. Hair Meshes. *ACM Trans. Graph.* 28, 5, Article 166 (2009), 7 pages.

[49] Meng Zhang, Menglei Chai, Hongzhi Wu, Hao Yang, and Kun Zhou. 2017. A Data-driven Approach to Four-view Image-based Hair Modeling. *ACM Trans. Graph.* 36, 4, Article 156 (2017), 11 pages.

[50] Yi Zhou, Liwen Hu, Jun Xing, Weikai Chen, Han-Wei Kung, Xin Tong, and Hao Li. 2018. HairNet: Single-View Hair Reconstruction Using Convolutional Neural Networks. In *ECCV '18*. 235–251.

APPENDIX

DATASET

In Figures 19 and 20, we provide the 30 base models in our dataset of the scalp hair, with variations in hair length, hairline, curliness, and hair parts (e.g. ponytail and bun).

Figure 19: *Our data set, part 1.* We use the symbol "×2" at the top-right corner of a hairstyle to indicate it has a flipped counterpart.

Figure 20: *Our data set, part 2.*