Struct2Hair: A hair shape descriptor for hairstyle modelling

Wenshu Zhang
Norwich University of the Arts
w.zhang@nua.ac.uk

Yinyu Nie
Technical University of Munich
yinyu.nie@tum.de

Shihui Guo
Xiamen University
guoshihui@xmu.edu.cn

Jian Chang
Bournemouth University
jchang@bournemouth.ac.uk

Jianjun Zhang
Bournemouth University
jzhang@bournemouth.ac.uk

Ruofeng Tong
Zhejiang University
trf@zju.edu.cn

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

Figure 1: A hair shape descriptor approach for hairstyle modelling. (a). Input portrait (b). Hair shape descriptor (c). Generated full-head hairstyle (d). Side view (e). Back view. Portrait originates from [1]

Abstract

In recent years, it becomes possible to extract hair information for hair reconstruction from multiple cameras or monocular camera. Using a single image as the input avoids the high cost setups and complex calibration compared to multi-viewed reconstruction. Taking advantage of an extendible hairstyle database, this paper introduced Struct2Hair, a novel single-viewed hair modelling approach by extracting hair shape descriptor (HSD). The HSD is defined as the fundamental structure-aware feature, which is a combination of critical shapes in a hairstyle. A complete dataset of critical hair shapes is constructed from a known database of 3D hair models. We first analyse the input 2D image to extract the orientation information and 2D hair sketch automatically. The extracted information is then used to retrieve the corresponding critical shapes with optimisation to build the robust HSD. Finally, the HSD constructs a weighted 3D hair orientation field to guide full-head hair model generation. Our method can preserve local geometric features of hair and retain the whole shape of the hairstyle globally owing to the HSD, which

will benefit further hair editing and stylisation.

Keywords: Hairstyle Modelling, Hair Shape Descriptor, Data-driven Modelling

1 Introduction

Hair is an important feature to form character appearance in both film and video game industry. Hair grooming and combing for virtual characters was traditionally an exclusive task for professional designers because of its requirements for both technical manipulation and artistic inspiration. However, this manual process is time-consuming and further limits the flexibility of customised hairstyle modelling. In addition, it is hard to manipulate virtual hairstyle due to intrinsic hair shape. The fast development of related industrial applications demands an intuitive tool for efficiently creating realistic hairstyle for non-professional users. Recently, image-based hair modelling has been investigated for generating realistic hairstyle [2, 3].

Early image-based hair modelling algorithm use multi-viewed images to recover missing depth information [4, 5, 6, 7]. They mainly rely on complicated set-up of capturing device, which is difficult to be implemented by common users. With the help of data-driven method, single portrait image based hair modelling research achieved great success in recent years [8, 9]. This approach does not require the setup of a specialised system and allows almost an arbitrary portrait image as input. It also reduces computation cost by eliminating the need for image registration. Furthermore, image based hair modelling is also the cornerstone for many relevant applications, including portrait manipulation by re-lighting hair and changing hairstyles, high quality 3D printable relief by adding hair details and 3D hairstyle space navigation etc [8, 10, 11, 12, 9, 3].

This paper proposed a pipeline of Struct2Hair, a single-viewed hair modelling method based on a novel hair shape descriptor (HSD). The HSD analysed hairstyle structure on both hair geometry shape and hair distribution. We built a large dataset of critical hair shapes and extracted robust HSD from 2D hair information. After this, the HSD constructed a weighted 3D hair orientation field to guide full-head hair model generation. Compared to the latest single-image based hair modelling research [2, 3], our new approach is capable of extracting and modelling the local feature of a representative hair, in addition to remixing candidate hairstyle models retrieved from database to obtain a similar hairstyle model. Our method enhanced the image-based hair modelling by preserving local geometric features of hair and retaining a global shape of a hairstyle owing to the introduction of HSD, which benefits further hair editing and stylisation.

In summary, this paper demonstrates a work flow (see Figure 2) that robustly captures a hairstyle from a single portrait input. Specifically, critical shapes of a hairstyle on a coarse level were extracted from generated 2D hair strands by clustering and optimisation first. We built a critical hair shape database by analysing an existing hairstyle model database. The critical hair shapes is a group of hair strands which possess similar shape appearance and close space location, whose centres are the HSD for each hairstyle in the original database. The robust HSD is constructed by retrieving and matching corresponding critical hair shape centres in the database. The full-head hairstyle was reconstructed by uniformly distributing the hair on the scalp with the guidance of HSD. The produced results were compared against with the original portraits to evaluate the quality of our output.

Main contributions of this paper listed below:

- Propose a hair shape descriptor (HSD) to analyse hair structure
- Generate HSD preserves local geometric features of hair as well as the global shape of the hairstyle
- Detailed critical hair shape database benefit other hair modelling research
- Compact hairstyle storage and fast reloading by HSD

2 Related Works

Image-based hair modelling. The image-based hair modelling method has been studied for

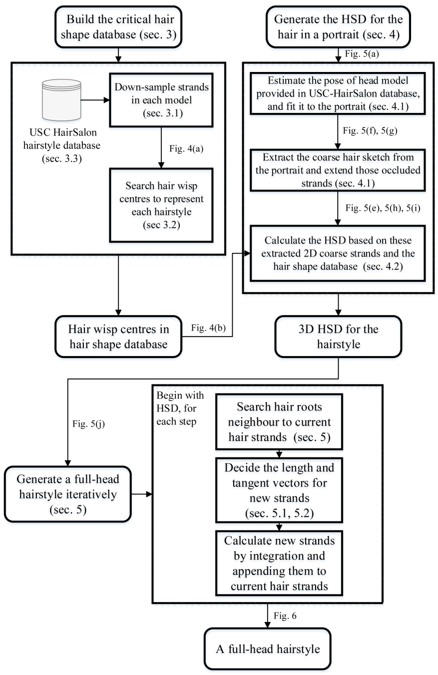


Figure 2: Struct2Hair: Work flow of full-head hairstyle generation

many years due to the intuitive sense of hairstyle given by images compared to traditional hair scratch. [4, 5, 6, 7] pioneered in modelling hair from images either by extracting hair silhouette or by recovering 3D hair orientation field to trace hair growth. With the development of dedicated equipments, more robust hair geometry has been captured. [13] used 16 cameras with a projector to reveal the rays of light for hair geometry capturing. [14] assembled hair fibre by fibre by recovering hair depth from defocused images. A thermal camera was used in [15] to segment hair between skin and background. [16, 17, 18, 11] leveraged specialised capture setups with multiple synchronised cameras to get high quality input images. Most multi-viewed hair modelling methods need complicated acquisition setups, which prevents prevalent usage.

To avoid the dilemma mentioned above, hair modelling based on single image was proposed by [8, 10]. They computed hair orientation map to generate sparse 2D hair strands. A visually plausible hair model was inferred from these strands combined with space constraints by a heuristic method. Inspired by their research, we traced 2D hair strands within the hair region by using the same Gabor filtering results in [8]. These strands were used to extract 2D hair sketch for our 3D HSD generation, which is different from referring 3D hair strands directly from 2D information. In [9], a shape from the shading algorithm was used to refine hair strands generation. They also added a helical prior for 3D hair strands optimisation to pop up high-quality portrait relief. However, the aforementioned researches were incapable of generating full-head hair model due to missing information in single image. The two most similar approaches to ours, which model full-head hair model from single image by data-driven method are[2] and [3]. [2] prototyped capturing whole hairstyle model with using a hairstyle database. Several user input strokes needed for guiding retrieve hair example in the database. To make the hair modelling even more easy handling, [3] introduced a fully automatic way to generate the hair model. The hair region in the portrait image was detected by a trained convolutional neutral network. They also constructed a database for data-driven hair modelling. The resulted hair model could benefit many future applications including physical-based hair animation and hair space navigation etc. Both [2] and [3] are top-down methods, they require hairstyle models remixing and adjustments to closely match the hairstyle of the original picture. We have proposed a way to analyse the key structure of a hairstyle and reconstruct the full-head hair model from the HSD. Since hair shapes are difficult to be analysed by a generative approach, this bottom-to-up method is quiet challenge currently.

Data-driven method. Data-driven approach has been used in many areas in addition to hair capturing. The success of data-driven based research shows its ability in many applications including modelling and character animation control etc [19, 20, 21, 22]. As a single portrait input naturally lacks of depth information, the invisible hairs in the back of the head can only be inferred by heuristic method. Leveraging a hairstyle database can generate reliable photo-realistic hair model and keep the global shape consistency at the same time [23, 2, 3]. Inspired by these fantastic works, this paper constructs a hair element database and uses it to model full-head hairstyle.

Sketch-based modelling. Sketch playing as guide strokes demonstrates structure information of the modelling object. It contributes to depth hint for image-based modelling and offers a way of user interaction. In addition, the sketch is also a robust feature for object retrieving combined with the data-driven method mentioned above. Recent research witnesses its achievements at various field including hair modelling [24, 2, 25, 26, 27]. In this paper, we extracted 2d hair sketch, which was incorporated with the aforementioned critical hair shape database to further generate 3D HSD for hairstyle structure analysing.

3 Critical hair shape database

This work proposed a data-driven approach to construct a database composed of critical hair shape. The main purpose of building such a database is to use a minority of strands from each full-head hair model to carry the representative structure of the overall hairstyle.

We propose a two-hierarchic clustering method to search the optimal key hair wisps to represent the basic structures of a hairstyle in distribution, length, shape, etc. This clustering method can be divided into two steps: scalp region segmentation and wisp centres searching. The first step is achieved by a fast down sampling. It aims to reduce the calculation burden of the following step without losing much information of strands' shape. The second step preliminarily classifies strands by their location. The critical hair shape database is built based on the wisp.

3.1 Scalp region segmentation

Here, we heuristically assume that the neighbouring hair strands share the similar geometrical properties including length and shape. Thus, the hair roots on the scalp have been divided into m non-overlapping regions by a down sampling plan named as Minimum and Maximum Distance Design [28, 29], which is well-developed for selecting samples to be evenly distributed in a spatial interval.

We denote the 3D Cartesian coordinates of all the hair roots as $\boldsymbol{X} = \left\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(n)}\right\}^T$, $\boldsymbol{x}^{(i)} \in \mathbf{R}^3, i = 1, 2, ..., n$, where $\boldsymbol{x}^{(i)}$ is the

Figure 3: Scalp region segmentation. 100 optimal samples are selected from 9977 hair roots.

root node of the i th strand, and n is the number of strands. Instead of sampling from a spatial interval, we need to sample from a dataset $\boldsymbol{X}$. Hence interval sampling problem in Minimum and Maximum Distance Design is transformed to the point sampling by selecting the best m samples $\boldsymbol{X}'$ from to $\boldsymbol{X}$ minimise [30]:

$$\min_{\boldsymbol{X}' \subset \boldsymbol{X}} \Phi_b\left(\boldsymbol{X}'\right) = \left(\sum_{j=1}^{k} J_j d_j^{-b}\right)^{1/b}. \quad (1)$$

$\{d_j\}$ is a vector of distinct distance values between all point pairs in $\boldsymbol{X}'$ (where the Euclidean distance is used in this paper). J_j is the number of point pairs which can be separated by d_j. k is the size of $\{d_j\}$. b is a positive integer which is generally set as $20 - 50$ if it is a large problem. For how to select a proper value, we recommend to further reading page 338 in [29].

Considering there are nearly 10,000 strands in each hair model from the database (about 100k hair for a real human), in this paper we set b as 50, and adopt the integral programming to heuristically solve the optimisation problem 1 with the genetic algorithm [30], which shows robustness in searching the global minimum with a heuristic manner. Then the optimal samples $\boldsymbol{X}'$ are evenly distributed on the scalp which can represent the whole hair roots $\boldsymbol{X}$. After that, for each hairstyle model, we can segment the scalp into 50 non-overlapping areas in this paper by searching for the neighbour hair roots of each sample in $\boldsymbol{X}'$ by the kNN algorithm (see Figure 3).

3.2 Hair wisp centres searching

For a hairstyle model, after segmenting the n strands into m groups, we apply the K-means++ algorithm [31] to categorize them into c clusters.

The hair strands within the same cluster share the similar geometric properties.

To distinguish strands with different shapes, we provide a metric (Equation 3) where three factors are mainly considered: strand length, spatial position and tangent value:

$$d_s(\boldsymbol{s}_i, \boldsymbol{s}_j) = \frac{d_H(\boldsymbol{s}_i, \boldsymbol{s}_j)}{\tau_{i,j}} \tag{2}$$

$d_H(\boldsymbol{s}_i, \boldsymbol{s}_j)$ is the Hausdorff distance between two hair strands. As a popular approach in measuring the distance between two curves in a metric space [32], it is presented as:

$$\mathrm{d}_H(\boldsymbol{s}_i, \boldsymbol{s}_j) = \max\left\{\sup_{p\in\boldsymbol{s}_i}\inf_{q\in\boldsymbol{s}_j} d(p,q), \sup_{q\in\boldsymbol{s}_j}\inf_{p\in\boldsymbol{s}_i} d(p,q)\right\} \tag{3}$$

In equation 3, s_i, s_j are denoted as the i th and j th strand of a hair model, p and q are the nodes distributed on them respectively, where $p, q \in \mathbf{R}^3$. $d(p,q)$ here we adopt the Euclidean distance.

To avoid two hair strands with different shapes are too close to have a small d_H to be grouped into one cluster, a coefficient $\tau_{i,j}$ is added onto the Hausdorff distance. $\tau_{i,j}$ is the variable to represent the tangent information between two hair strands. It is computed as inner product between the tangent vectors of the two strands, see Equation 4.

$$\tau_{i,j} = \frac{1}{l-1}\sum_{k=1}^{l-1}\langle \tan(p_k), \tan(q_k)\rangle, l \geq 2. \tag{4}$$

$\tan(p_k)$ and $\tan(q_k)$ respectively denote the unit tangent vector at point p_k (on s_i) and q_k (on s_j). l is the number of points on the shorter strand, wherein, to make points on all strands distributed equally in distance, the cubic spline interpolation is used to re-sample these points beforehand.

The metric in equation 2 shows that two strands which have a close location, similar length and tangents will contribute to a smaller distance.

In section 3.1, we have got the down-sampled strands. Each strand there represents an individual spatial area on the scalp. Based on our metric, we apply the K-means ++ method to gather these strands into c clusters. Then it yields c centre strands, each of which can largely reflect the overall style of strands in a cluster. The clustering result of a hair model is shown in Figure 4(a), 4(b).

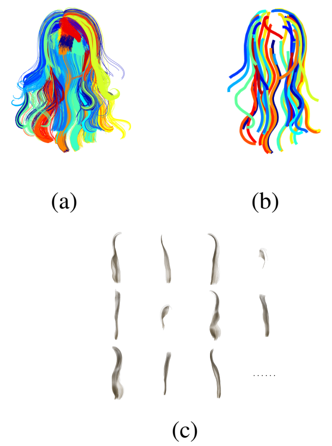


Figure 4: Hair strand clustering. (a). A hairstyle model from [2] grouped into 50 clusters. (b). Corresponding 50 cluster centres. (c). Extracted hair wisps used in section 3.3

3.3 Critical hair shape database building

We develop our critical hair shape database based on USC-HairSalon hairstyle database [2]. By doing the clustering on 343 models in the hairstyle database in section 3.2, we obtained a large set of hair wisps (see Figure 4(c)). Those hair wisps exhaustively demonstrate natural hair shapes ranging from short straight hairstyles to long curly hairstyles. Therefore, we call such hair wisps as critical hair shapes and collect them as a database.

4 Hair shape descriptor (HSD)

Due to the complicated intrinsic shape of individual hair strand, it is difficult to describe the structure of hairstyle. In order to analyse and capture hairstyle model from single 2D image, we introduce hair shape descriptor (HSD) to demonstrate the representation of hair structure. The Hair Shape Descriptor (HSD) is explained in this section.

4.1 Image Pre-processing

2D hair sketch extraction. We first cut the hair region from the input portrait image by image editing software Photoshop, this process is mostly automatic, excluding some challenging cases where hair color is similar with the background. Then we use the Gabor filter in [8] to compute 2D hair orientation map. Instead of computing seed points to grow strand, we evenly sample hair seed points within the region of hair to preserve the possible shape information. The 2D hair strands are then traced on the image plane (see Figure. 5(b), 5(c)). In some cases, a hair strand might be cut into several line segments after the tracing due to occluded by other strands or inevitable image noises. Thus the segments belonging to the same strand should be connected together first. Here we define an asymmetric distance between any two segments as:

$$\mathrm{d}_c\left(\boldsymbol{s}_i, \boldsymbol{s}_j\right) = \begin{cases} d\left(p_i^t, q_j^h\right) & \text{if } d\left(p_i^t, q_j^h\right) < \epsilon \;\&\; \theta < \frac{\pi}{6} \\ \infty & \text{otherwise} \end{cases}, \tag{5}$$

where p_i^t, p_j^h are respectively denoted as the tail and head nodes of segment $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$, which means the distance cost when attaching the head of $\boldsymbol{s}_j$ to the tail of $\boldsymbol{s}_i$. $d\left(p_i^t, q_j^h\right)$ is the Euclidean distance, and the threshold value ϵ is used to filter out those pairs that are too large to be connected together. θ is the orientation different between p_i^t and q_j^h to enforce the hair holding a natural bending angle.

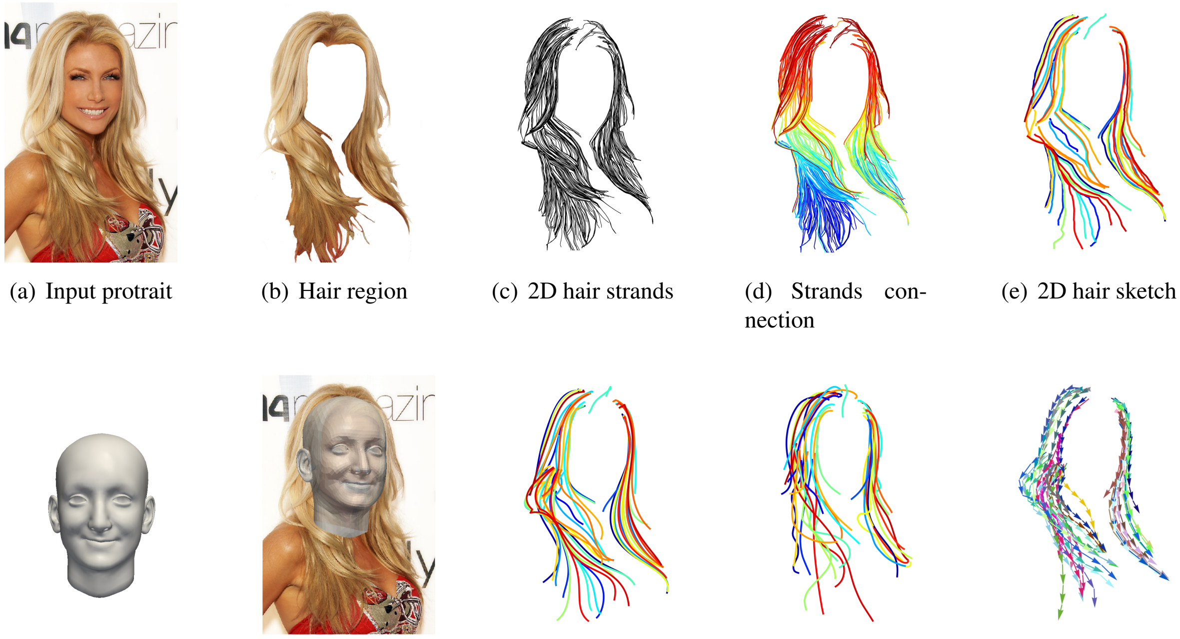
After the asymmetric distance matrix $\mathbf{D} = (d_c\left(\boldsymbol{s}_i, \boldsymbol{s}_j\right))$ is obtained, we use a searching algorithm to connect these segment pairs by the order from the minimal distance value to the higher. For a connection, once meeting the end of this searching, such that no extra segment can be connected to the current hair strand, then delete the indexes of these involved segments in matrix $\mathbf{D}$ and begin a new connection (see Figure 5(d)).

After the complete strands are obtained, we apply the same K-means++ algorithm in section 3.2 on the generated 2D hair strands to cluster them. The collection of the cluster centres is our 2D hair sketch. We set 10 as the initial number of clusters. But our result shows that 10 is not enough for full-head hairstyle recover. We tested 10, 25, 50 and 75 clusters, and 50 comes out as a trade off between reconstruction satisfaction and computation efficiency (see Figure 5(e)).

2D hair sketch extension. However, the generated 2D hair strands didn't reveal the missing information from the original image. Consider the hairstyle is a combination of different hair wisps, hair strands grown from the back side of the head also participant to add variation to the visible hair. Thus, we extend the extracted hair sketch to be attached to the scalp of the head aligned to the portrait.

Figure 5(e) shows that some extracted strands are occluded by the front hair, which partly appear in the current view. To recover the occluded part for these strands, we extend the extracted hair sketch with the following steps to let them grow from hair roots in 2D. The extension result is shown in Figure 5(h). There is a standard head model provided in USC-HairSalon database aligned with each hairstyle. In the following work, in order to do the 2D shape centres matching, we project the critical hair shape centres to the 2D portrait plane through fitting the head model to the face of the portrait by the pose estimation and shape fitting method in [33].

1. Pose estimation & Head fitting
   a) Detect 2D facial landmarks of input portrait, and manually label their correspondences on the given standard head model.
   b) Estimate the pose of the standard head model by solving camera matrix using the Gold Standard Algorithm [34].
   c) Rotate and scale the standard head model by obtained camera matrix to make its face aligned to the portrait, see Figure 5(g).
2. Attach 2D hair sketch
   a) Project the 3D hair roots on the standard head model to the 2D image plane.
   b) Search the nearest hair root for each strand in the 2D hair sketch, and use polynomials to fit these strands along with their nearest hair root, which ensures that the occluded strands are at-



(a) Input protrait (b) Hair region (c) 2D hair strands (d) Strands connection (e) 2D hair sketch

(f) Standard head model (g) Head fitting (h) Extended 2D hair sketch (i) Matched 3D centres from database (j) Generated HSD

Figure 5: Hair shape descriptor generation. We first segment hair region (b) from input portrait (a), then trace 2D hair segments (c) on image plane. We perform a segment connection on (c) to obtain more reliable 2D hair strands (d), and extract the 2D hair sketch (e) from it. To optimise the 2D hair sketch, the standard head model (f) provided in [2] has been mapped (g) onto the input portrait. Then we extend 2D hair sketch (h) to make it grow from projected 2D scalp and search the best matched 3D centres (i) from our critical hair shape database. The hair shape descriptor (HSD) (j) is generated by adding depth information on 2D extended hair skecth. Input choose from [1]

tached to the hair roots on the back head.

Our tests show that for very curly hair styles, a high-order polynomial sequence is required to represent their strands. For general cases, the fifth-order polynomials are enough to ensure a great performance. Figure 5(h) displays that those occluded hair strands in Figure 5(e) are connected to the hair roots on the back head, which are used to generate the back hair in following steps.

4.2 HSD generation

For each 2D cluster centre, we search the closest critical hair shape from the table maintained for the hair shape database according to the shape and distance constraints. Therein, we firstly project every entry of the table to the 2D image plane based on the pose estimation process. Then the distance metric in equation 2 is used for matching the closet 2D critical hair shape centre to each strand of hair sketch. The matched 3D critical hair shape centre is shown in Figure 5(i).

We build frenet-serret frame for both 2D strand in hair sketch and its corresponding 3D critical hair shape centre. The depth of 3D critical hair shape centre is then applied to guide the generation of 3D hair sketch. This 3D hair sketch is our hair shape descriptor (HSD), which plays as an important structure feature to reconstruct full-head hairstyle generation in section 5. For each pair of 2D strand and matched hair shape centre, we can build the corresponding parametric curve functions as follows.

$$\begin{cases} x = x_1(s_1) \\ y = y_1(s_1) \end{cases}, s_1 \in [0, L_1], \tag{6}$$

$$\begin{cases} x = x_2(s_2) \\ y = y_2(s_2) \\ z = z_2(s_2) \end{cases}, s_2 \in [0, L_2], \quad (7)$$

where $s_{i,\ i\in[1,2]}$ represents the curvilinear abscissa along the curve i. x, y and z are the Cartesian coordinates of the nodes on a curve. the x-y plane is coplanar with the portrait plane. For 2D hair sketch, the coordinates along z-axis (depth direction) are zeros. x_1, y_1, x_2, y_2, z_2 are the parametric functions which are approximated by the cubic spline functions. L_1 and L_2 are denoted as the total arc length of the two curves.

Afterwards we take a linear mapping from s_1 to s_2 as:

$$s_2 = \frac{L_2 s_1}{L_1}, s_2 \in [0, L_2], \quad (8)$$

and transfer the depth information to the 2D hair sketch, then each 3D strand of the HSD can be represented by

$$\begin{cases} x = x_1(s_1) \\ y = y_1(s_1) \\ z = z_2\left(\frac{L_2 s_1}{L_1}\right) \end{cases}, s_1 \in [0, L_1]. \quad (9)$$

A generated HSD is shown in Figure 5(j).

5 Full-head hairstyle generation

Based on the HSD obtained above, full-head hairstyle is generated by treating the whole hairstyle as a 3D orientation field. All the hair strands are grown on a predefined region on the scalp with the evenly distributed hair roots (we generate 10000 hair strands for the full-head hairstyle in this paper).

Before that, we firstly propose two assumptions rules for generating hair from HSD.

1. For strands whose hair roots are located within a close distance, their strand lengths are similar, except some special areas (e.g. the fringe hair).

2. The orientations of neighbouring nodes on two strands (attached together) are similar in common cases.

When doing full-head hair generation, the HSD represents the shape information as a collection of local hair details. Each strand of HSD is treated as an initial hair cluster centre to diffuse the entire hairstyle around the scalp. We adopt an iterative hair generation method to keep local hair details when generating new hairs. New hair strands with close distance to HSD will be grown first, then expanded until full-head strands generation is completed. The main steps of the hair generation algorithm are shown as following:

1. Build a dataset $\boldsymbol{X} = \{\boldsymbol{x}_i\}, \boldsymbol{x}_i \in \mathbf{R}^3$ to store all the hair roots and a dataset $\boldsymbol{S} = \{S_i\}$ to store the current strands, which is initiated by strands in HSD.

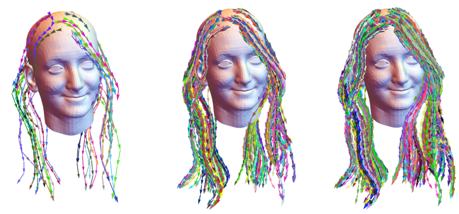
2. Loop until $\boldsymbol{X}$ is empty.

   a) For each strand $S_i \in \boldsymbol{S}$, search m nearest hair roots in $\boldsymbol{X}$.

   b) Collect all the m nearest hair roots, and generate strands based on their neighbour strands in $\boldsymbol{S}$.

   c) Update the current strands set $\boldsymbol{S}$ by appending the generated strands, and remove their hair roots from $\boldsymbol{X}$.

3. $\boldsymbol{S}$ is the generated full-head hair.



(a) 50 (b) 425 (c) 1218

Figure 6: Increasing hair strands around scalp

The number of new generated strands in step two is increasing with the expanding of $\boldsymbol{S}$, which ensures that the local shape information of the original centres is kept while the shape of new generated strands between these centres varies smoothly. This will be reflected in the local generation step, please see Figure 6 for details.

Taking the first hair generation step as an example, given a set of hair roots and initial hair strands from HSD, there are two phases in generating the new strands: deciding the strand length and the corresponding growth direction.

5.1 Deciding strand length

Based on the first priori assumption, for each hair root, we firstly use the k-NN algorithm to search m neighbouring hair roots among these centres, then adopt the Multi-quadric (M-Q) radial basis function [35] to interpolate its strand length based on the length data of these centres. k-NN method used here is aimed at keeping the local length and avoiding a large coefficient matrix in M-Q interpolation (see the matrix $(\phi(r_{i,j}))$ in Equation 12). The form of M-Q radial function is:

$$\phi(r) = (-1)^{\lceil \beta \rceil} \left(r^2 + c^2\right)^{\beta}, c \geq 0, \beta > 0, \beta \notin \mathbf{N}, \tag{10}$$

and the interpolated strand length function is written as:

$$l(\boldsymbol{x}) = \sum_{i=1}^{m} a_i \phi(r),\ r = ||\boldsymbol{x} - \boldsymbol{x}_i||. \tag{11}$$

In equation 10, the parameters β and c are used to control the shape of M-Q function where we recommend to refer [35] for their definitions. r is the variable of the radial Euclidean distance between two points. $\lceil \beta \rceil$ denotes the ceiling function, we set $\beta = 0.5$ in our experiment. $\mathbf{N}$ is the natural number.

In equation 11, $\boldsymbol{x}$ is the 3D position of the hair root with its corresponding length $l(\boldsymbol{x})$ to be interpolated. $\{\boldsymbol{x}_i\}$ is the closest m neighbouring hair roots to $\boldsymbol{x}$, and a_i is the interpolation coefficient of each radial basis $\phi(r)$. Based on the HSD, their hair roots $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_m\}$ and corresponding hair lengths $\{l(\boldsymbol{x}_1), l(\boldsymbol{x}_2), ..., l(\boldsymbol{x}_m)\}$ can be used to solve $\{a_i\}$ as the following:

$$\begin{pmatrix} l(\boldsymbol{x}_1) \\ l(\boldsymbol{x}_2) \\ ... \\ l(\boldsymbol{x}_m) \end{pmatrix} = \begin{pmatrix} \phi(r_{1,1}) & \phi(r_{1,2}) & ... & \phi(r_{1,m}) \\ \phi(r_{2,1}) & \phi(r_{2,2}) & ... & \phi(r_{2,m}) \\ ... & ... & ... & ... \\ \phi(r_{m,1}) & \phi(r_{m,2}) & ... & \phi(r_{m,m}) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ ... \\ a_m \end{pmatrix}, \tag{12}$$

where $r_{i,j} = ||\boldsymbol{x}_i - \boldsymbol{x}_j||$.

The strand length of the new strand can be figured out by giving the 3D position of its hair root in equation 11. In some special scenarios, especially the fringe hairs, there is a length gap between them and their neighbours. Since the M-Q function is continuous, it makes the hair at the boundary of the gap a little longer than it suppose to be. In this case, a pre-defined area for fringe hair should be given in advance to ensure the length of new hair strands located in it can only be affected by the hair strands in this area. For hair styles with a fringe hair, we make the 1/5 area in front of the scalp as the special area. A hairstyle with a choppy full fringe hairs is used to demonstrate how these two parameters work, see Table 1 for a detailed discussion with fringe hair editing.

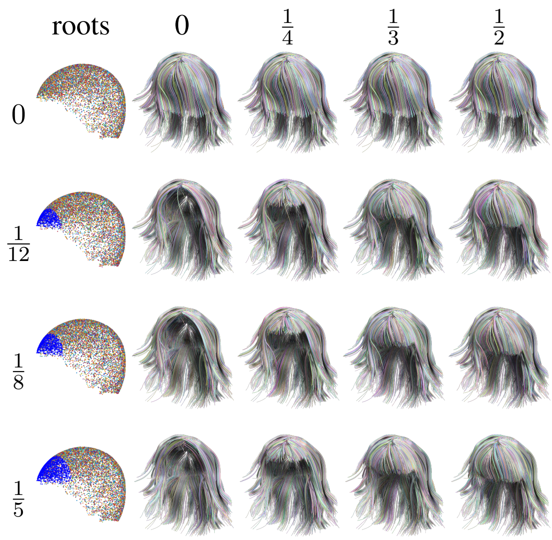


Table 1: Parameters to edit the shape of fringe hairs. The parameters in top row are the proportions of the front hair length, which is used to cut the fringe hair. There is no fringe hair when length proportion equals 0. The parameters in the first column is the ratios, which indicate how much front hairs will account for fringe area. When both ratio and length proportion equal 0, the hairstyle is the original generated from HSD. For ensure the natural look of the hairstyle, ratio is set to $\frac{1}{5}$ and the length proportion is set between $\frac{1}{3}$ and $\frac{1}{2}$.

5.2 Deciding growth direction

After figuring out the length of this new strand, based on the second priori assumption, we reconstruct the entire curve by integrating along its growth direction. Beginning with its hair root, by using the weighted k-NN regression, we interpolate the orientation of a strand node from the u neighbour nodes on other centre strands.

Denote the node on a strand as $\boldsymbol{r}(s)$, and its u neighbours from different centre strands as $\boldsymbol{r}(s_1), \boldsymbol{r}(s_2), ..., \boldsymbol{r}(s_u)$. $\boldsymbol{r}(s), \boldsymbol{r}(s_i) \in \mathbf{R}^3, i = 1, 2, ..., u$. Then the orientation vector $\boldsymbol{r}'(s)$ can be solved by the weighted regression as :

$$\boldsymbol{r}'(s) = \frac{\sum_{i=1}^{u} \omega_i^2 \boldsymbol{r}'(s_i)}{\sum_{i=1}^{u} \omega_i^2}, \tag{13}$$

where $\omega_i = (\sum_{i=1}^{u} d_i) - d_i$, $d_i = ||\boldsymbol{r}(s) - \boldsymbol{r}(s_i)||$, d_i is the Euclidean distance between $\boldsymbol{r}(s)$ and $\boldsymbol{r}(s_i)$. $\{\omega_i^2\}$ are the weight values.

Starting from the hair root, we iteratively calculate its orientation vector and update the new node by $\boldsymbol{r}(s + \delta s) = \boldsymbol{r}(s) + \boldsymbol{r}'(s)\,\delta s$ until meeting the end of the strand length. Besides, to make the shape of strands vary smoothly between centre strands, we adaptively change the searching range u in a new hair generation step by $u_{i+1} = 1.5u_i$, $u_0 = 1.5$.

6 Evaluation

Modelling result. We first test our HSD method on hairstyle models (as ground truth data) from USC-HairSalon database to evaluate the feasibility of our algorithm. Through generating hair from extracted HSD of hairstyle models, the results show the HSD based hairstyle modelling can reconstruct a hairstyle as similar as possible compared to the original hairstyle, see Figure 7.

Next, we test the novel HSD approach on portraits with different hairstyles. The results prove our algorithm's capability of dealing various hairstyles. In Figure 8, the rendered hairstyle models follow the natural way of hair growth.

We also compared our method with the state-of-the-art single image-based hair modelling algorithms [2, 3]. Figure 9 shows that for the original view, our method generate the hairstyle model closely matched the input image, which has the similar quality of the previous single-view hair modelling approaches. However, different from their methods, our framework focus on reconstruction of the reference hairstyle from its basic structure, which benefits a presence of rich local details of hair shape.

Furthermore, the Struct2Hair framework is also compared with the cutting-edge multi-viewed hair modelling technique by [36]. They

Figure 7: Hairstyle synthesized by HSD approach compared with ground truth hairstyle. From left to right: 1 column is a short straight hairstyle from USC-HairSalon, 2 column is the corresponding synthesized short straight hairstyle by HSD, 3 column is a long curly hairstyle from USC-HairSalon, 4 column is the corresponding synthesized hairstyle by HSD.

use four-view images to reconstruct a target hairstyle. The result from our Struct2Hair is rendered under the four reference views to compare with their model in Figure 10. It presents the capability of our framework to generate reliable hairstyle model with only a single view reference image.

Meanwhile, as the full-head hair is fully automatically generated from HSD, we consider it is a compact feature to save and reload hairstyle. The result in Figure 7 proves its capability of reconstructing hairstyle. We take this advantages and can down-sized the USC-HairSalon database from 3.53GB to 41.2MB by representing hairstyle of HSD.

Limitation. The HSD extraction is based on image pre-processing. In some cases, portrait images are lack of clear hair information due to high reflection or bad illumination which cause unreliable HSD generation, which result in unrealistic hairstyle capturing. Our HSD method is incapable of dealing with the fuzzy hairstyle at

Figure 8: Hairstyle models generated by HSD approach from single image. From left to right, 1 column is the original portrait, 2 column is the generated hairstyle model by HSD approach, 3 column is the corresponding side view. Portraits originate from [1]

the current stage. The intrinsic fuzzy hairstyle degenerates the consistency of 2D hair strands extraction. In that case, a better performance 2D hair strands extraction approach should be considered. As the HSD descriptor is designed for capturing hairstyle from single image, the generated hair model can only keep the hairstyle matched the original view, and preserve a plausible visual effect when rotating the head due to the unknown groundtruth of the back side.

Moreover, similar to [3], the head pose estimation method fails when extreme side-viewed face or tilted head appear in the input portrait. This causes unreliable critical hair shape manipulation and unsatisfied hair model generation. This problem can be solved by the improvement of face alignment method. Also, like the previous data-driven based single-view hair modelling methods, our Struct2Hair relied on the critical hair shape dataset constructed from the USC-HairSalon database. When there is no desired critical hair shape matched to the HSD, the quality of the reconstructed hair model is less satisfactory compared to [2, 3]. This requires us to build a large critical hair shape database in the future work.

Future work. The HSD method is based on the analysis of the hair structure, which could further contribute to a hairstyle editing application. A common user with no professional knowledge can easily comb the hair by changing the HSD to control the hairstyle.

The compact structure of HSD will help us enlarge the hairstyle database by adding new generated HSD into existing database. With the expansion of database, we could improve our algorithm to capture more realistic and fine local hair geometric features. This database will benefit future sketch based hair modelling methods. We have done some tests to re-mixing two HSDs to populate new hairstyle models for the USC-HairSalon database. Please see Figure 11 for details, the γ_1 and γ_2 represent the corresponding source hairstyles' contributions to the new hair model. We will continue to explore generating new hairstyle model by adding several source models in the future.

The HSD could also perform as a skeleton for hair simulation. Take the HSD as the guide to control hair movement to save computation cost. We will improve the current work-frame to test video input based hair simulation as our future work.

7 Conclusions

The Struct2Hair, a hair shape descriptor (HSD) approach for hair modelling is introduced in this paper. Compared to the state-of-the-art single-viewed hair modelling methods, HSD based hair modelling provides a bottom-to-up pipeline, which focuses the basic structure of a hairstyle. The generated hairstyle models show the ability of capturing both global and local feature. The full-head hair grown around the scalp follows the natural behaviour which retain the whole shape of the hairstyle globally. Our compact HSD structure also enables a way of hairstyle management like saving and reloading the hairstyle, as well as enlarge the hairstyle database by appending new extracted HSD. Promising hairstyle editing and simulation applications can be developed by manipulating the HSD in future.

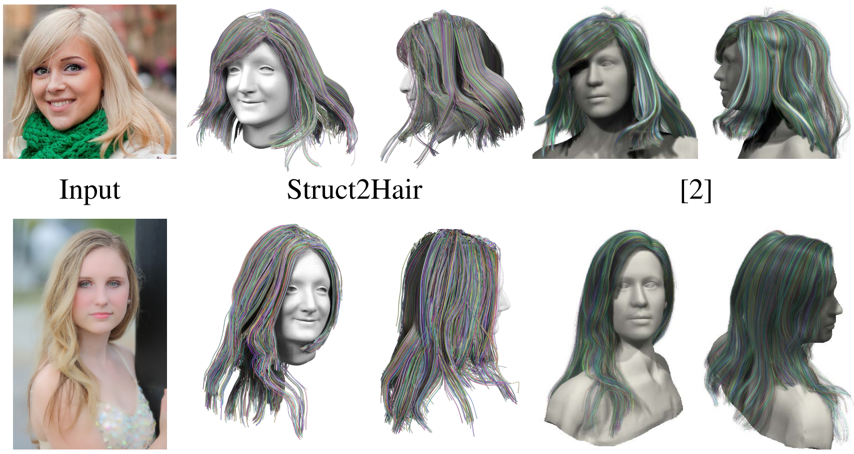


Figure 9: Compare with the state-of-the-art single-viewed hair modelling techniques. From left to right are the input portraits, hairstyle model generated by Struct2Hair, [2] and [3] respectively. Original image courtesy of Chris Zerbes and Bob HARRIS.

Figure 10: Comparison with the state-of-the-art multi-viewed hair modelling techniques. From left to right are the input portraits, hairstyle model generated by Struct2Hair using from the front view input, [36] and [11] respectively.

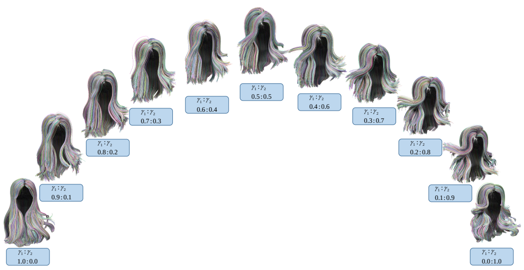


Figure 11: Blend hairstyle one with hairstyle two by setting different coefficients γ_k. From left to right, the hairstyle is morphing from pure hairstyle one to pure hairstyle two with the $\gamma_1 : \gamma_2$ changing from $[1:0]$ to $[0:1]$.

8 Acknowledgement

The authors would like to thank Liwen Hu for helping with the comparison and making their hairstyle models available for research use, the Menglei Chai for his kind suggestion. The research leading to these results has received partial support from the EU FP7 funded project AniNex (FP7-IRSES-612627).

References

[1] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision (IJCV)*, July 2016.

[2] Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. Single-view hair modeling using a hairstyle database. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2015)*, 34(4), July 2015.

[3] Menglei Chai, Tianjia Shao, Hongzhi Wu, Yanlin Weng, and Kun Zhou. Autohair: Fully automatic hair modeling from a single image. *ACM Trans. Graph.*, 35(4):116:1–116:12, July 2016.

[4] Waiming Kong and Masayuki Nakajima. Generation of 3d hair model from multiple pictures. *The Journal of the Institute of Image Information and Television Engineers*, 52(9):1351–1356, 1998.

[5] Stephane Grabli, François X Sillion, Stephen R Marschner, Jerome E Lengyel, et al. Image-based hair capture by inverse lighting. In *Proceedings of Graphics Interface (GI)*, pages 51–58, 2002.

[6] Sylvain Paris, Hector M Briceño, and François X Sillion. Capture of hair geometry from multiple images. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 712–719. ACM, 2004.

[7] Yichen Wei, Eyal Ofek, Long Quan, and Heung-Yeung Shum. Modeling hair from multiple views. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 816–820. ACM, 2005.

[8] Menglei Chai, Lvdi Wang, Yanlin Weng, Yizhou Yu, Baining Guo, and Kun Zhou. Single-view hair modeling for portrait manipulation. *ACM Transactions on Graphics (TOG)*, 31(4):116, 2012.

[9] Menglei Chai, Linjie Luo, Kalyan Sunkavalli, Nathan Carr, Sunil Hadap, and Kun Zhou. High-quality hair modeling from a single portrait photo. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 34(6), November 2015.

[10] Menglei Chai, Lvdi Wang, Yanlin Weng, Xiaogang Jin, and Kun Zhou. Dynamic hair manipulation in images and videos. *ACM Trans. Graph*, 32:4, 2013.

[11] Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. Robust hair capture using simulated examples. *ACM Transactions on Graphics*, 33(4), July 2014.

[12] W. Zhang, J. Chang, J. J. Zhang, M. Wang, and R. Tong. Image-based hair preprocessing for art creation: A case study of bas-relief modelling. In *2015 19th International Conference on Information Visualisation*, pages 411–418, July 2015.

[13] Sylvain Paris, Will Chang, Oleg I Kozhushnyan, Wojciech Jarosz, Wojciech Matusik, Matthias Zwicker, and Frédo Durand. Hair photobooth: geometric and photometric acquisition of real hairstyles. *ACM Trans. Graph*, 27(3):30, 2008.

[14] Wenzel Jakob, Jonathan T Moon, and Steve Marschner. Capturing hair assemblies fiber by fiber. In *ACM Transactions on Graphics (TOG)*, volume 28, page 164. ACM, 2009.

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

[16] Linjie Luo, Hao Li, Sylvain Paris, Thibaut Weise, Mark Pauly, and Szymon Rusinkiewicz. Multi-view hair capture using orientation fields. In *Computer Vision and Pattern Recognition (CVPR)*, June 2012.

[17] Linjie Luo, Hao Li, and Szymon Rusinkiewicz. Structure-aware hair capture. *ACM Transactions on Graphics (TOG)*, 32(4):76, 2013.

[18] Linjie Luo, Cha Zhang, Zhengyou Zhang, and Szymon Rusinkiewicz. Wide-baseline hair capture using strand-based refinement. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 265–272. IEEE, 2013.

[19] Lin Gao, Yu-Kun Lai, Dun Liang, Shu-Yu Chen, and Shihong Xia. Efficient and flexible deformation representation for data-driven surface modeling. *ACM Transactions on Graphics (TOG)*, 35(5):158, 2016.

[20] Desai Chen, David IW Levin, Shinjiro Sueda, and Wojciech Matusik. Data-driven finite elements for geometry and material

design. *ACM Transactions on Graphics (TOG)*, 34(4):74, 2015.

[21] Eunjung Ju, Jungdam Won, Jehee Lee, Byungkuk Choi, Junyong Noh, and Min Gyu Choi. Data-driven control of flapping flight. *ACM Transactions on Graphics (TOG)*, 32(5):151, 2013.

[22] Andreas Baak, Meinard Müller, Gaurav Bharaj, Hans-Peter Seidel, and Christian Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *Consumer Depth Cameras for Computer Vision*, pages 71–98. Springer, 2013.

[23] Lvdi Wang, Yizhou Yu, Kun Zhou, and Baining Guo. Example-based hair geometry synthesis. In *ACM SIGGRAPH 2009 Papers*, SIGGRAPH '09, pages 56:1–56:9, New York, NY, USA, 2009. ACM.

[24] Jamie Wither, Florence Bertails, and Marie-Paule Cani. Realistic hair from a sketch. In *Shape Modeling and Applications, 2007. SMI'07. IEEE International Conference on*, pages 33–42. IEEE, 2007.

[25] Alex Shtof, Alexander Agathos, Yotam Gingold, Ariel Shamir, and Daniel Cohen-Or. Geosemantic snapping for sketch-based modeling. In *Computer Graphics Forum*, volume 32, pages 245–253. Wiley Online Library, 2013.

[26] Kun Xu, Kang Chen, Hongbo Fu, Wei-Lun Sun, and Shi-Min Hu. Sketch2scene: Sketch-based co-retrieval and co-placement of 3d models. *ACM Transactions on Graphics (TOG)*, 32(4):123, 2013.

[27] Changqing Zou, Xiaojiang Peng, Hao Lv, Shifeng Chen, Hongbo Fu, and Jianzhuang Liu. Sketch-based 3-d modeling for piecewise planar objects in single images. *Computers & Graphics*, 46:130–137, 2015.

[28] Mark E Johnson, Leslie M Moore, and Donald Ylvisaker. Minimax and maximin distance designs. *Journal of statistical planning and inference*, 26(2):131–148, 1990.

[29] Max D Morris and Toby J Mitchell. Exploratory designs for computational experiments. *Journal of statistical planning and inference*, 43(3):381–402, 1995.

[30] Mehdi Taheri and Mehdi Ahmadian. Machine learning from computer simulations with applications in rail vehicle dynamics. *Vehicle System Dynamics*, 54(5):653–666, 2016.

[31] Ragesh Jaiswal and Nitin Garg. Analysis of k-means++ for separable data. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 591–602. Springer, 2012.

[32] Xiao-Diao Chen, Weiyin Ma, Gang Xu, and Jean-Claude Paul. Computing the hausdorff distance between two b-spline curves. *Computer-Aided Design*, 42(12):1197–1206, 2010.

[33] Patrik Huber, Guosheng Hu, Rafael Tena, Pouria Mortazavian, Willem P Koppen, William Christmas, Matthias Rätsch, and Josef Kittler. A multiresolution 3d morphable face model and fitting framework. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 1–8, 2016.

[34] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[35] A.H.-D. Cheng. Multiquadric and its shape parameter¡ªa numerical investigation of error estimate, condition number, and round-off error by arbitrary precision computation. *Engineering Analysis with Boundary Elements*, 36(2):220 – 239, 2012.

[36] Meng Zhang, Menglei Chai, Hongzhi Wu, Hao Yang, and Kun Zhou. A data-driven approach to four-view image-based hair modeling. *ACM Transactions on Graphics (TOG)*, 36(4):156, 2017.