Modeling Hair from an RGB-D Camera

MENG ZHANG, PAN WU, HONGZHI WU, and YANLIN WENG, State Key Lab of CAD&CG, Zhejiang University
YOUYI ZHENG and KUN ZHOU, Zhejiang University and ZJU-FaceUnity Joint Lab of Intelligent Graphics, China

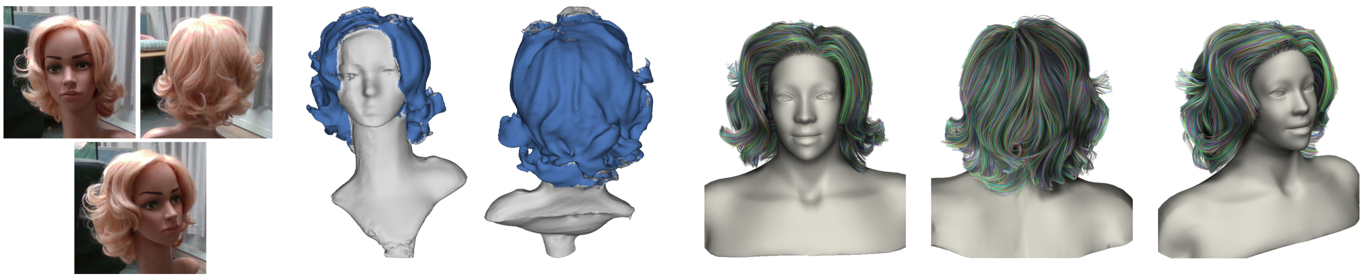

Fig. 1. Given a stream of color frames (left) and the corresponding fused geometry model (middle) from a single RGB-D camera, our method computes a complete strand-level 3D hair model (right) that closely resembles both the fusion model and the hair textures in the input images.

Creating realistic 3D hairs that closely match the real-world inputs remains challenging. With the increasing popularity of lightweight depth cameras featured in devices such as iPhone X, Intel RealSense and DJI drones, depth cues can be very helpful in consumer applications, for example, the Animated Emoji. In this paper, we introduce a fully automatic, data-driven approach to model the hair geometry and compute a complete strand-level 3D hair model that closely resembles the input from a single RGB-D camera. Our method heavily exploits the geometric cues contained in the depth channel and leverages exemplars in a 3D hair database for high-fidelity hair synthesis. The core of our method is a local-similarity based search and synthesis algorithm that simultaneously reasons about the hair geometry, strands connectivity, strand orientation, and hair structural plausibility. We demonstrate the efficacy of our method using a variety of complex hairstyles and compare our method with prior arts.

CCS Concepts: • **Computing methodologies** → *Image processing*; Mesh models;

Additional Key Words and Phrases: Fully automatic hair modeling, RGB-D camera, exemplar-based, 3D PatchMatch

*Corresponding authors: Yanlin Weng (weng@cad.zju.edu.cn) and Youyi Zheng (youyizheng@zju.edu.cn).
Authors' addresses: Meng Zhang; Pan Wu; Hongzhi Wu; Yanlin Weng, State Key Lab of CAD&CG, Zhejiang University; Youyi Zheng; Kun Zhou, Zhejiang University , ZJU-FaceUnity Joint Lab of Intelligent Graphics, Hangzhou, 310058, China.

0730-0301/2018/11-ART205 $15.00
https://doi.org/10.1145/3272127.3275039

ACM Reference Format:
Meng Zhang, Pan Wu, Hongzhi Wu, Yanlin Weng, Youyi Zheng, and Kun Zhou. 2018. Modeling Hair from an RGB-D Camera. *ACM Trans. Graph.* 37, 6, Article 205 (November 2018), 11 pages. https://doi.org/10.1145/3272127.3275039

1 INTRODUCTION

Hair modeling is one of the most crucial components in digital avatar creation. Modeling 3D hairs that closely resemble the real-world inputs is a demanding task in today's emerging VR and AR applications. Due to the inherently densely convoluted complex structures exhibited in the wide variety of real-world hairstyles, modeling of a complete and realistic 3D hair remains challenging.

Image-based hair modeling has shown promising progress in the past few years. Early researches rely on complex capture setups in controlled environments [Hu et al. 2014a; Luo et al. 2013; Paris et al. 2008] to achieve compelling reconstruction results but are less suitable for non-professional users. Single-view based methods can reconstruct results with complex strand-level 3D hairs [Chai et al. 2015, 2016, 2013, 2012; Hu et al. 2015, 2017] but often fall short in generating results that match the reality at views distant from the input one. Zhang et al. [2017] introduce a four-view image-based hair modeling method. They generate consistent hair textures over a smooth surface of rough hair shape to combine hairstyles from different views. Since there is no detail information along the normal direction of the surface, they adopt helix curves fitting to guide detail refinement which inevitably involves some artifacts. Except the single-view approach of Chai et al. [2016], most of the aforementioned methods require certain amounts of manual work. Moreover, the lack of detailed geometry information in the RGB images makes it difficult for these methods to model realistic 3D hairs that retain high fidelity to the real-world inputs, for example, to model the fine details shown in Fig. 1, middle.

In this paper, we present a fully automatic, data-driven approach to synthesize high-quality hair geometry from a consumer-level RGB-D camera. The increasing popularity of depth cameras featured in devices such as iPhone X and Microsoft XBox has brought in a variety of applications (e.g., animated emoji). It has also opened up research opportunities in the communities of visual recognition [Song and Xiao 2013] and 3D reconstruction [Dai et al. 2017; Zhou and Koltun 2014]. Hence we seek to model 3D hairs that closely match the real-world inputs by exploiting both the color and geometric cues provided by the RGB-D camera.

To this end, we develop a search-and-synthesis framework for modeling 3D hairs with an RGB-D camera. Given the streamed depth and color frames from the camera, our method first fuses the 3D geometry of the hair and automatically extracts consistent hair regions across all frames. We then define patch-based local similarity to search exemplars in a 3D hair database, and use them as references to synthesize a complete 3D orientation field which is consistent with both the fused surface orientation and the input color images. In a key stage, we perform 3D PatchMatch-based optimization to obtain a nearest neighbor field mapping between the synthesis result and the exemplars, from which we grow hair strands by the geometry guidance from the exemplars to ensure hair structural plausibility.

Both the depth and color information from the camera are crucial to the high-quality modeling results - the depth frames provide a rough 3D geometry of hair shape, while the color frames contain the hair surface orientation. On the other hand, due to the inaccuracy of estimated camera poses and the fusion process, there are inevitably noise and ambiguity in the reconstructed surface orientation, which makes it difficult to directly grow hair strands along the diffused orientation field. Our 3D PatchMatch-based synthesis algorithm fills 3D orientation in the empty regions and also enables us to bypass the need for accurate reconstruction. Moreover, the locally mapped strand geometry of exemplars as guiding references ensures the structural plausibility of the synthesized result and also constrains the strand lengths.

Our method is capable of reconstructing a variety of real-world hairstyles which are challenging for the current state-of-the-art methods and is suitable for an end-user to create a 3D avatar with personalized hairstyles (see Fig.12). In summary, our contributions are:

- The first high-quality and fully automatic hair modeling pipeline that generates 3D hairs closely matching the real-world inputs using a single RGB-D camera;
- An efficient exemplar search method by local patch-based similarity to generate guide geometries;
- A 3D orientation synthesis method by 3D PatchMatch-based optimization;
- A hair synthesis method guided by 3D nearest neighbor mapping to propagate structural plausibility from exemplar strands to the target hair model.

2 RELATED WORK

Human hair modeling is extensively studied in computer graphics, in which professional skills and laborious manual work are often involved. A detailed discussion can be found in the survey of [Ward et al. 2007]. By far, there are only two approaches which have examined fully automatic hair modeling pipelines [Chai et al. 2016; Hu et al. 2017], however, both methods are single-view based and hence the hair at views distant from the front one is often hallucinated with deficient quality, and it is unknown how to extend their algorithms to multiple views.

Yuksel et al. [2009] first introduced hair meshes to model complex hairstyles, where coarse polygonal hair meshes from exemplars encode hair positions and directions and are used as guidance for strand generation. Our method falls into the category of image-based hair modeling which has shown to be a promising way to create compelling hair geometries from captured hair images. Below we review relevant works that are closely related to ours, i.e., those based on images.

Multi-view hair modeling methods [Echevarria et al. 2014; Herrera et al. 2012; Hu et al. 2014a; Jakob et al. 2009; Luo et al. 2013; Paris et al. 2008], which create high-quality 3D hair models from images taken from a number of views, often require complex capture setups and long processing cycles. Hu et al. [2014b] use a single RGB-D sensor as a multi-stereo acquisition hardware to address constrained braided hairstyle.

Single-view hair modeling methods have recently achieved impressive results but often require posing various priors such as layer boundary and occlusion [Chai et al. 2013, 2012], shading cues [Chai et al. 2015], or relying on 3D hair model database [Chai et al. 2016; Hu et al. 2015, 2017]. A major problem with single-view based techniques is the lack of control over the final result at views distant from the input one, as there is no input information at all.

In both multi-view and single-view methods, structural references are incorporated during hair reconstruction process. Luo et al. [2013] use a bottom-up strategy to connect local ribbons into wisps through purely geometry-inspired heuristics. Hu et al. [2014a] propose a strand fitting algorithm to find structurally plausible configurations among simulated strand examples. Hu et al. [2014b] use a braid patch fitting method to find a set of fitted structure patches. All these multi-view methods which use structural priors cost a lot of time in the stage of point cloud reconstruction and outlier point removal [Luo et al. 2013], strand fitting [Hu et al. 2014a], or structure patch fitting [Hu et al. 2014b]. Hu et al. [2015] and Chai et al. [2016] use the predefined database as initial hairstyles to reconstruct hair model from a single image. Hu et al. [2015] require user strokes to reveal the full hair connectivity and topology and remix all candidates, one of which is found for each stroke. Chai et al. [2016] use the mask-based search to find best matches from a populated database of 40k samples. Both methods are difficult to find a good hair model that matches all views in a limited database.

Zhang et al. [2017] introduce a four-view image-based hair modeling method, to fill in the gap between multi-view and single-view hair modeling. They require users to specify the camera configuration for each view and estimate a rough 3D shape of hair using a predefined database of hair models and then synthesize hair texture consistent with all four input images. Since the texture algorithm needs a smooth rough shape surface, they only use contours as matching metric and do not consider the internal hair structure. Piece-wise helix-fitting [Cherin et al. 2014] is involved to refine the strand details that are washed out by rough shape reconstruction and orientation diffusion. Our method uses KinectFusion [Newcombe et al. 2011] to generate a hair shape, along with the estimated camera poses. We use a localized 3D PatchMatch algorithm and exemplar hairs as guidance to generate hair strands to account for both the global hair shape and a plausible internal hair structure.

PatchMatch algorithm [Barnes et al. 2009, 2010] efficiently computes a Nearest-Neighbor Field (NNF) that stores the correspondences between patches of a 2D images. More discussion on 2D cases can be found in the comprehensive survey of [Barnes and Zhang 2017]. We adopt the original PatchMatch algorithm in the step of example retrieval, where the inputs are color images of the front and back views and the images of hair meshes which are rendered under the two constrained views respectively. Li et al. [2017] extend PatchMatch optimization to 3D volumetric voxels for shape completion. In our 3D orientation synthesis, where the source and target are represented as 3D vectors in volumetric voxels, we aim to calculate an NNF mapping from the target to the source as geometry guidance for hair synthesis.

3 OVERVIEW

The pipeline of our automatic system of portrait hair modeling is shown in Fig. 2. Given a stream of depth and color frames captured by a consumer-level RGB-D camera, we first reconstruct a rough 3D geometry and estimate camera poses using Kinect-Fusion algorithm [Newcombe et al. 2011]. Similar to Zhou and Koltun's method [2014], a subset of images is selected from the color frames as input. We pass selected color frames through the hair parsing network of [Chai et al. 2016] (*hairnet* we call hereafter), to automatically segment out hair region and remove directional sign ambiguity(**§4**). With the hair segments and the directional maps, we estimate 2D directional fields on each frame and project them to the fused hair model to obtain a 3D surface orientation field (Fig. 2, middle right), which is then diffused into a 3D volumetric orientation field.

We observe that direct methods [Chai et al. 2016; Paris et al. 2008] to grow hair strands along the diffused 3D orientation field could neglect the fine geometric details of the original hair model since these methods tend to grow hair in a globally smooth manner. In order to get hair strands that closely match the input model, we devise an exemplar-based hair synthesis paradigm to synthesize a full detailed 3D hair

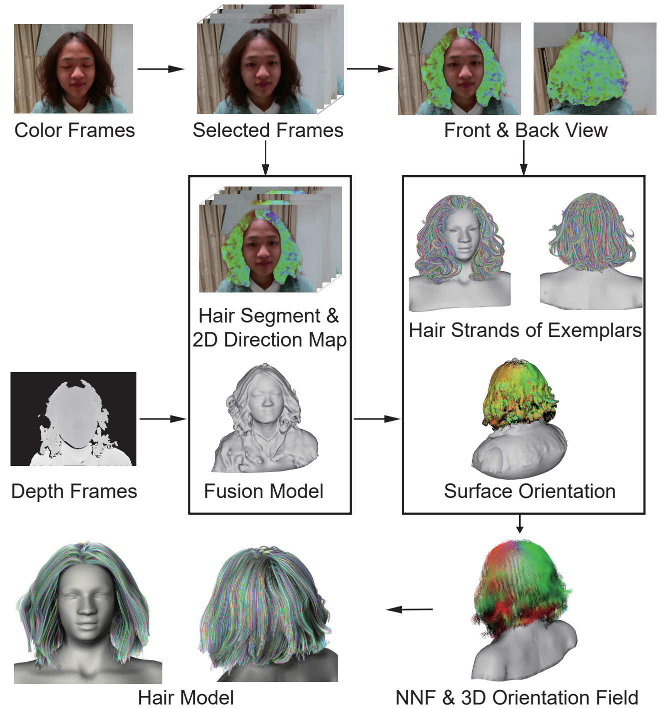

Fig. 2. Overview of our pipeline. Given color and depth frames captured by a consumer-grade RGB-D camera, we extract hair masks and 2D direction map using *hairnet* [Chai et al. 2016], and obtain a fusion model of the hair using the method of [Newcombe et al. 2011]. The hair masks and the 2D direction map are used to reconstruct 3D orientation on the hair shape surface. With two best-matching exemplars as guiding geometry, we construct nearest neighbor field mapping between the matched exemplars and the fusion model using a 3D patchmatch-based optimization. Finally, a strand-level 3D hair model is synthesized under the guidance of the mapping, with its colored 3D digitalized character obtained through a color-map method, which is visually similar to the real model.

by taking into consideration of both the global shape and local details of the fusion model. Specifically, we search in a 3D hair database and select two best-matching exemplars which have similar style patterns with the front and back views respectively. We convert the database models from mesh to hair strands and establish 3D hair orientation field in the corresponding hair volume, which are regarded as guiding geometry (**§5**). Given the 3D volumetric orientation fields from the exemplars and the orientation field from the fusion model, we perform 3D patchmatch-based optimization constrained by the surface orientation on the fusion model, to obtain an NNF mapping, which is then employed to guide the hair growth in the internal region. The NNF mapping helps us derive structure patterns from the guiding exemplars, which reason about the connectivity, direction and hair end, and help to synthesize the hair strands matching the fusion model and the hair textures in all input images (**§6**).

4 PREPROCESSING

Our input is a stream of depth images and the synchronous color images captured by Intel RealSense SR300 camera at 30 fps. The resolution of depth and color images are both set to 640×480. A hair model takes 1-2 rotations of 360 degrees in front of the RGB-D camera to get a full head scan.

Geometry fusion. The depth frames can produce an initial mesh G using the Kinect-Fusion pipeline [Newcombe et al. 2011], along with estimated camera poses that approximately register each depth image to the mesh. We perform re-meshing by uniformly sampling new vertices $\{p_j\}$ over the surface G via the method of ACVD [Valette and Chassery 2004].

Hair region segmentation and direction annotation. Similar to Zhou and Koltun's method [2014], to reduce the number of color images, we select one frame with the lowest blurriness in each interval of 50 color frames, and gather them as input color images $\{I_i\}$. We pass all selected color images through *hairnet* [Chai et al. 2016], to automatically obtain a set of hair region masks $\{M_i^I\}$ and direction label maps $\{M_i^D\}$.

With hair region mask, we can segment out hair region mesh G^M. A vertex p on G is defined as a hair surface vertex when its projection lies inside the hair region of the nearest view w.r.t. the normal of p. Please see Fig. 1 (middle) for an example of G^M.

With direction label map, since the network [Chai et al. 2016] is trained with images of hair distributed around the face, we assume that the label map has higher confidence c_i when the viewpoint is nearer to that of the front view. We also densely calculated a per-pixel orientation map O_i^I for I_i [Luo et al. 2013]. For a pixel p_i with 2D orientation o_i^I in the hair region, we can project it back to 3D space using estimated camera transformation, to get a 3D position p. Then we can estimate the view based 3D orientation vector $d_i(p)$, referring the method in [Luo et al. 2013]. In order to decide the direction sign for $d_i(p)$, we use a voting strategy. We project it to all visible views, and then sum up those voting view confidence: if $C_+ > C_-$, the direction sign for p_i is positive, and vice versa.

Head fitting. Given the front view, we locate a set of facial feature points using a face alignment algorithm [Cao et al. 2014]. By re-projection of the image feature points back onto the surface of G, we obtain 3D facial feature points. By extending the head mesh fitting method in [Chai et al. 2012] to 3D, we compute a new head shape S_{new} by the function:

$$S_{new} = TS^* = T(\bar{S} + V \cdot \beta)$$

$\bar{S}$ is the average head shape vector, V is the matrix of principal components and β is the coefficient vector, where all those three components are computed in the same way as in [Chai et al. 2012]. T is a rigid transformation computed by least squares on the corresponding feature points. We finally get S_{new} through an alternating optimization, i.e., S^* is optimized while T is kept fixed, and vice versa.

5 GUIDING GEOMETRY GENERATION

Following [Hu et al. 2015], we collect an initial set $\{H\}$ of about 300 3D models of different hairstyles, downloaded from online public repositories [Electronic Arts 2014], and double the number of the database by flipping each exemplar. In this section, we describe how to generate 3D guide geometries by searching from the database.

Hu et al. [2015] demonstrate that, with the help of a rich set of hair geometry exemplars, excellent hair modeling results can be obtained, which match the image cues and retain the realism of hair geometry. However, in their method, the global hair structure cannot be robustly estimated from local image information, thus user interactions are needed. A subsequent time-consuming optimization is also required to adapt the exemplars to the user-specified global structure.

In order to avoid user interaction and improve run-time, Chai et al. [2016] dramatically expand the database to an amount of more than $40K$ on the precomputation stage by remixing initial models and organize them for compact storage. Moreover, they retrieve a few good-matching candidates from a large set of exemplars, typically 5-40 candidates, all of which are involved in the further reconstruction until the last step, when the closest matching model is selected from the candidates. Zhang et al. [2017] only need an approximately matching hair shape with the four view contour constraints. They only use contours as direct matching cues and do not consider the internal hair structure.

In our case, we do not require that the exemplar models in the database have the same style in global as our target hairstyle. We only expect that there are *local style patterns* provided in need to guide the target hair synthesis. Front and back are two views which are sufficient to express a hairstyle [Zhang et al. 2017], thus we regard them as guiding views to search exemplars from the database, one for each view. We pick the first input image as the front view (normally, the model scan starts from the front view with angle 0^o), and the back view is selected from $\{I_i\}$ as the one whose view angle is closest to 180^o.

5.1 Database View Representation

All these database models are composed of a number of independent thin polygon-strips. Each strip represents a coherent hair wisp, with relative strand growth directions encoded in parametric texture coordinates. We render each database at two views: front and back, with the projected direction encoded in color space for each pixel. See the left and middle column in Fig.3 for example. To search for the best matching exemplars, we consider the following two factors in order: the hair mask M^H for mask test to filter out less matched candidates, and the orientation map O^H for style patch match to search for the best ones.

5.2 Mask Test

After view rendering for each database exemplar, we can get a binary 2D hair mask M^H. The distance between mask M^H

Fig. 3. Guide exemplar search results for the front and back views, respectively. Left column: given a hair segmentation, the best matching exemplar rendered at the corresponding view. Middle column: the original mesh with color indicating the direction. Right column: hair strands grown from the scalp.

and the input image mask M^I is measured as:

$$dist(M^I, M^H) = \frac{M^I \cup M^H - M^I \cap M^H}{M^I \cup M^H}$$

The candidates whose distances to both the front and back views are within 0.5 are selected.

5.3 Style PatchMatch

Our next task is to select for each view one best matching exemplar, from the candidates which pass through the mask test. Given an input color image I_i and the orientation map O^H of an exemplar H at the corresponding view, we need a measurement to decide whether H can describe the hairstyle of the input image. First, we densely calculate a per-pixel orientation map O_i^I for I_i [Luo et al. 2013] and remove the directional ambiguity using the voting strategy described in **§4**. The distance between O^H and O_i^I is then defined as:

$$dist(O_i^I, O^H) = \sum_{X \in O_i^I} \min_{Y \in O^H} ||X - Y||^2$$

where X is an $m \times m$ orientation patch in O_i^I and $m = 9$ in our cases; Y is the same size of patch in O^H whose distance to X is the smallest. For each patch X in O_i^I, we run several PatchMatch sweeps [Barnes et al. 2009] to find a nearest patch Y in O^H.

After the selection of mask test described above, there are about $5 \sim 100$ candidates. In the style patchmatch step, we use 4 pyramid levels in the PatchMatch searching for speedup. On each resolution level ($l = 4, 3, 2, 1$), we keep the first k candidates with the smallest distance for efficiency ($k = 20, 10, 5, 1$, respectively).

5.4 Strand-level Exemplars and 3D Orientation Field

Given the two exemplars $\{H_{1,2}\}$ retrieved by the front and back views, following previous solutions [Chai et al. 2013; Hu et al. 2015; Paris et al. 2008], we convert $H_{1,2}$ to 3D orientation volume ($\sim 60 \times 60 \times 60$) within the bounding box of H_1, H_2, and G^M, and perform oorientation diffusion inside the volume by treating the direction vectors given by H and the head surface normal near the scalp region as constraints. We then grow $30,000$ strands from uniformly sampled seeds on the scalp, under the guidance of the 3D volumetric orientation field (see right column in Fig. 3). We record the following geometry information of $H_{1,2}$ for the further synthesis step: the first is the 3D volumetric orientation field D_h, with o^h_{xyz}, $h = 1, 2$ indicating the orientation at voxel (x, y, z); the second is the strand-voxel indexing dictionary $\{s_v\}^h_{xyz}(h = 1, 2)$ indicating the set of strand vertices of the model H_h which go through (x, y, z), s is the index of strand passing (x, y, z), v is the index of vertex on strand s which occupies (x, y, z). We sample the strand such that no two vertices on the same strand fall into the same voxel.

6 DATA-DRIVEN HAIR SYNTHESIS

After the guiding geometry generation step as described above, our next goal is to synthesize the strand level hair geometry that resembles both the fusion model and hair texture in the input images.

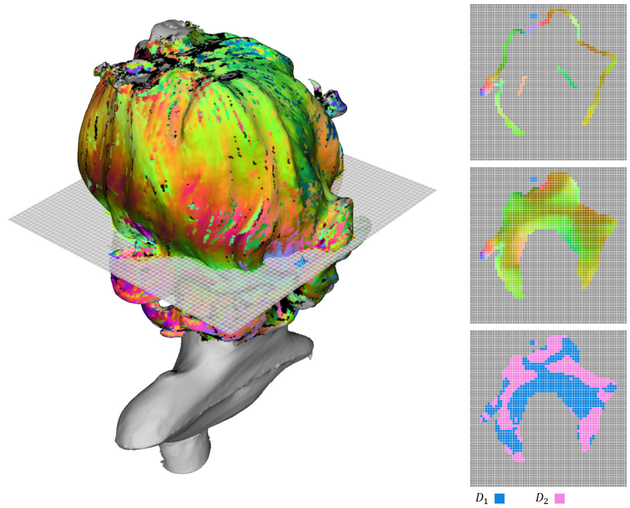

Fig. 4. 3D orientation field generation. Left: Surface 3D orientation field (colored per vertex). Right from up to bottom: a slice of surface orientation, a slice of 3D orientation synthesis result, and the source indexing.

ALGORITHM 1: 3D orientation synthesis algorithm

```
for iteration = 1 → n do
  for h = 1 → 2 do
    NNF(h) ← SingleSourceSynthesis(D_G, D_h)
  end for
  for each patch O^G_xyz do
    if dist(O^G_xyz, O^1_xyz) < dist(O^G_xyz, O^2_xyz) then
      nnf ← nnf(1)
    else
      nnf ← nnf(2)
    end if
  end for
  D_G ← FieldReconstruction(NNF, D_1, D_2)
end for
```

6.1 3D Orientation Reconstruction

Given a set of 2D orientation maps $\{O_i^I\}$ and the corresponding camera transformations $\{T_i^I\}$, referred to [Wei et al. 2005], for each point p on G^M, the 3D orientation $d(p)$ with all visible views of p is optimized as:

$$d(p) = arg\min_{d} \sum_{i} \delta_i^2 (N_i(p) \cdot d)^2$$

$N_i(p)$ is the normal of a plane defined by the cross of the line-of-sight vector and the 2D orientation direction projected into 3D in the camera space. d is the 3D vector to optimize. The line-of-sight vector is computed by p and view center of I_i, and the 2D orientation direction is obtained from the pixel of I_i where p is projected by T_i^I. δ is the confidence of pixel orientation. This can be solved efficiently by singular value decomposition. See the left in Fig.4 for an example.

The above process gives us a 3D orientation reconstruction for vertices on G^M. We then convert surface orientation to the 3D volumetric orientation field as defined in section 5.4, denoted as D_G.

6.2 NNF Mapping Generation

Given the 3D orientation field D_G, we perform 3D patchmatch-based orientation synthesis algorithm to update D_G, with guiding volumetric orientation fields D_1 and D_2. This operates in the following three stages.

Single-Source-Synthesis. Similar to [Darabi et al. 2012], our goal is to fill the content in the target region D_G with those from the source exemplars D_1 and D_2. We minimize the sum of squared differences between the corresponding voxel via patch-based operation:

$$E(D_G, D_h) = \sum_{O_{xyz}^G \in D^G} \min_{O_{xyz}^h \in D^h} ||O_{xyz}^G - O_{xyz}^h||^2$$

Here, O_{xyz} is an $m \times m \times m$ 3D voxel patch with 3D orientation vectors as elements and we set $m = 5$ in our experiments. The search travels in the order of the left-and-right and the up-and-down.

NNF Mapping. After Single-Source-Synthesis for both D_1 and D_2, we will determine for each target voxel patch whether NNF for D_1 or that for D_2 better matches. Here, we just naively keep the one with the smaller distance.

Field Reconstruction. Once all valid patches got their NNF, we update the 3D orientation field D_G in accordance with the computed NNF mapping. Specifically, the orientation of a particular voxel is computed as the average of all local patches (from D_1 and D_2) that cover it [Darabi et al. 2012]. The updated D_G will be used for the next iteration.

The above three steps are iterated many times (20 iterations in our experiments) to get a stable D_G as well as the NFF mapping (see an overview in ALGORITHM 1). In our subsequent hair synthesis step, we rely on the NNF mapping function to guide the hair growth.

6.3 Hair Synthesis

Although the 3D orientation field is reconstructed as described above, we still do not know the structure of strand connection and growth end. Therefore, directly growing strands from roots in scalp [Zhang et al. 2017] cannot cover the complete hair region (See Fig.11). In order to grow natural strands uniformly in the hair region, we process hair synthesis in 3 steps: hair growing, root assignment, and strand augmentation.

Hair Growing. The computed NNF mapping implicitly assigns a one-to-one mapping from a target patch position (x, y, z) to that of the exemplars $(x^h, y^h, z^h)(h \in \{1, 2\})$, represented as $nnf(x, y, z) \Leftarrow (x^h, y^h, z^h)$. As described in **§5.4**, we also obtain the strand-voxel indexing dictionary $\{s_v\}$. We grow strands starting from the uniformly sampled seeds inside the hair volume and along the direction guided by the strand geometry of the exemplars. For a seed s^*, positioned at voxel (x, y, z), its NNF maps p' in (x^h, y^h, z^h) where a set strands pass through with a set of vertices occupying it, $\{s_v\}_{xyz}^h$, we

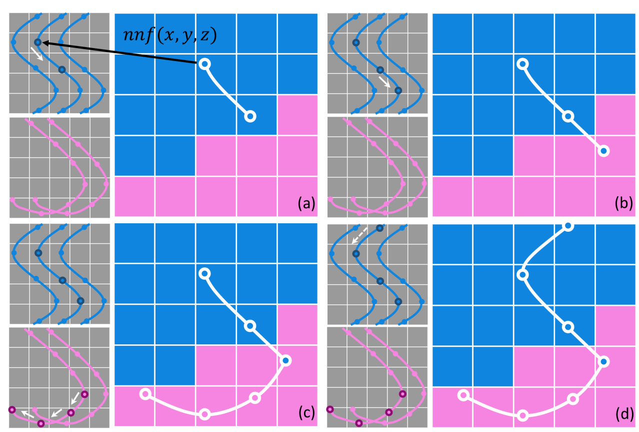

Fig. 5. Hair synthesis. For a sampled seed, nnf indexes its guide strand vertex in sources (a). Hair grows along the direction guided by source (b) until a source strand reaches its end (c). We also need to trace along the negative direction from the beginning seed (d).

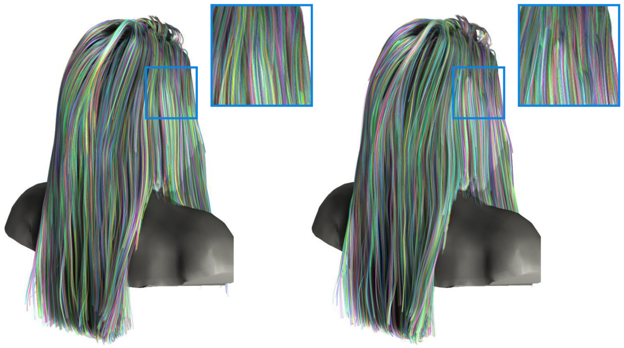

Fig. 6. Hair strands using our SDF distance-based root assignment (left) and using a direct greedy assignment (right). Strands at further distance will collide with closer strands when they are later connected to the scalp (the right blue square).

grow the strand along the direction d_+:

$$d_+ = \frac{\sum_{s_v \in NeighborReg(x^h,y^h,z^h)} w(s_v) \cdot \frac{s_{v+1}-s_v}{||s_{v+1}-s_v||}}{\sum_{s_v \in NeighborReg(x^h,y^h,z^h)} w(s_v)}$$

where $w(s_v) = \exp\left(-10 \cdot ||p' - s_v||^2/g^2\right)$, and g is the voxel width. $NeighborReg(x^h, y^h, z^h)$ is a $3 \times 3 \times 3$ neighboring voxel region centered at (x^h, y^h, z^h). Then we calculate the next strand vertex as $s^{n+1} \leftarrow s^n + \Delta \cdot d_+$, Δ is decided by the nearest source vertex s_v^*, which is set to $\|s_{v+1}^* - s_v^*\|$. The growing stops, if s_v^* is a strand end or p is outside the hair region.

Since the growing does not start from the scalp root, we also need to trace along the opposite direction d_- from the seed s^*:

$$d_- = \frac{\sum_{s_v \in NeighborReg(x^h,y^h,z^h)} w(s_v) \cdot \frac{s_v-s_{v-1}}{||s_v-s_{v-1}||}}{\sum_{s_v \in NeighborReg(x^h,y^h,z^h)} w(s_v)}$$

The next strand vertex is computed by $s^{n-1} \leftarrow s^n - \Delta \cdot d_-$, with $\Delta = \|s_v^* - s_{v-1}^*\|$. See Fig.5 for an illustration.

Root Assignment. After hair growing, we usually get about 3000 hair strands, denoted as S, which should be linked to the scalp. A direct assignment to their nearest scalp point could lead to undesired arrangements of the hair strands (see Fig. 6). We resort to a simple heuristic. We first, on the volume, calculate the sign distance field (SDF) to the head model. Larger SDF values mean further to the head model. We sort these strands by the value $\psi(s)$ from large to small. Given a strand $s \in S$, $\psi(s)$ is calculated by the SDF of all strand vertices of s:

$$\psi(s) = \frac{1}{N}\sum_v SDF(s_v)$$

where N is the number of vertices. We then uniformly sample more over 3000 roots on the scalp. The strands with a larger value of ψ have priority to be assigned first to its nearest root samples. We use the similar cut-and-connect procedure as in [Chai et al. 2016] to connect the strand to the scalp. This simple heuristic allows hair strands at further distances to get connected first to avoid the collision when closer strands take up the nearby roots too quickly (Fig. 6).

Strand Augment. After the strands set S obtained as described above, we augment the number of strands in our experiments by adding new strands to our result model as in [Zhang et al. 2017]. For each new strand, we again uniformly sample the scalp region as the root point, then copy the nearest guide strand from S. After that, we adopt the linear blend skinning approach in [Hu et al. 2015] to deform the new strand to be coherent with the neighboring ones. Finally, we obtain a hair model with $35,000$ strands.

7 RESULTS AND DISCUSSION

We run our modeling method on a variety of challenging hairstyles, ranging from short/straight to long/curly (see Fig.12). All experiments are run on a computer with an Intel Core i7-4790 CPU and 32GB of memory. With hair masks, color images and fusion model as input, a complete strand-level hairstyle model is calculated, taking about 10 minutes using our un-optimized automatic pipeline. The majority of the time is spent on the iterations of 3D orientation reconstruction. For comparison, the total processing time of [Zhang et al. 2017] for an example is about 25 minutes (including 6 minutes for user interactions). Fig. 12 shows the results. Our method is capable of successfully modeling various complex hairstyles and generating results that closely resemble the input images, including detail regions where it is very challenging for previous methods to reconstruct, and in the meantime, being fully automatic, thanks to our depth fusion and the local 3D patch guided strand synthesis algorithm.

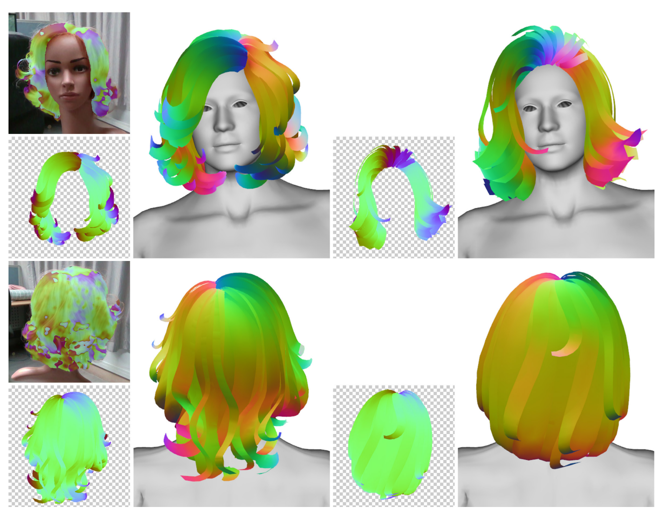

Fig. 7. Comparison of exemplar searching results with AutoHair [Chai et al. 2016]. Left two columns: the 2D hair mask and the orientation map (top), the best matching exemplar using our method (bottom) and the corresponding mesh model (right). Right two columns: the corresponding best matching results of AutoHair.

Fig. 8. Comparisons with a state-of-the-art four-view modeling method [Zhang et al. 2017]. From left to right, input images for [Zhang et al. 2017], our results (left three), and the results using [Zhang et al. 2017] (right three).

To evaluate our method, we first compare the exemplar matching strategy with Autohair [Chai et al. 2016]. In [Chai et al. 2016], they search results based on the symmetric difference between two masks and the pixel orientation difference. They keep 4-50 candidates for their further deformation steps. Here we show the results with the smallest distance drawn from the 478 exemplars. In their method, the pixel orientation difference in the union mask region cannot describe the similarity in style between the hair models of exemplar and the input image. In comparison, we use local style patterns as matching metric to find exemplars that provide patch patterns as many as we need to describe our target hairstyle model. Visual examples are shown in Fig. 7.

To evaluate the influence of the number of exemplars on the final result, we use 4 views (front, back and two sides, cf. [Zhang et al. 2017]) as guidance to search exemplars from the database. Representative comparison results are shown in the last row of Fig. 10. We find that the front and back views can already provide enough geometry guidance for hair synthesis and produce the modeling results that almost have the same quality as those by four exemplars. Thus for efficiency reason, we use the two-view scheme throughout our experiments.

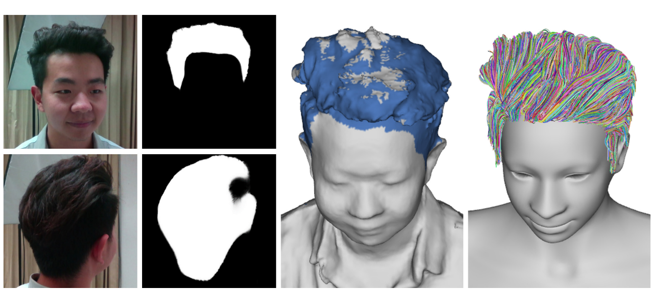

Fig. 9. Our algorithm is not sensitive to imperfect segmentations. From left to right: color images, hair masks generated by *hairnet* [Chai et al. 2016], hair region segmentation on the fusion model, and the 3D hair result by our method.

We segment out hair mask from color images using the *hairnet* in [Chai et al. 2016]. Although their network is trained with hair images with front views, we find it to perform consistently well for views distant from the front one. Occasionally, small artifacts could happen round vague regions such as the corner part of the side view hair in Fig. 9. Such improper 2D hair mask may result in imperfect hair segmentation on the 3D fusion model (Fig. 9 middle). Additionally, the incorrectly estimated camera poses from KinectFusion may also lead to outliers in the hair masks (e.g., Fig. 12, the forehead). Nevertheless, our data-driven hair synthesis algorithm is robust enough to overcome such incorrect mask regions and produces 3D hair model that resembles the input images considerably well (see Fig. 9, right).

We also compare our method with [Zhang et al. 2017], a state-of-the-art light-weighted multi-view hair modeling technique, in Fig. 8. Zhang et al. [2017] construct a smooth rough hair shape with a matching database shape as reference retrieved by the contours of the front, back and two side views. Thus, there's no information along the surface normal direction. In order to generate more geometry detail, they use a piece-wise helix fitting method for detail refinement. In comparison, our approach makes use of the fusion model computed from the depth frames, which provides a lot of shape details. Our hair synthesis method ensures that there are hair strands growing through even some small detail regions (Fig. 8). Moreover, the method of Zhang et al. [2017] requires user interactions for camera specification, hair segmentation, and directional ambiguity removal while ours runs in a fully automatic manner.

In addition, direct growing hair strands from scalp roots along the orientation field cannot guarantee to fill the entire

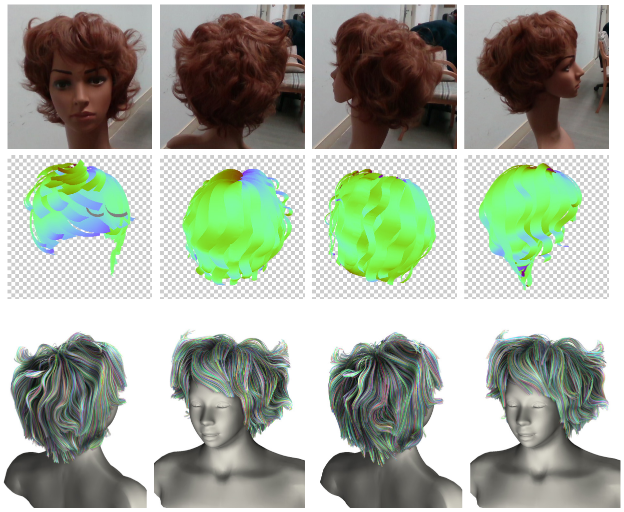

Fig. 10. Impact of the number of guiding exemplars. The top row shows the 4 input views (front/back/left/right) and the middle row shows their corresponding best-matching exemplars. The last row are modeling results using 2 views (front and back) and 4 views, respectively.

hair volume (See Fig. 11). Although cut-and-connect strategy [Chai et al. 2013] can refine the overall hair model, it unavoidably introduces some artifacts (see Fig. 8 and [Zhang et al. 2017]). Our hair synthesis method allows a plausible strands distribution inside the hair volume.

Limitations. Our hair modeling framework has limitations, which may inspire interesting future work. First, our head model fitting only considers a rigid transformation to align the head of the orthogonal view when fitting to the 3D fusion model. Thus the geometry of our fitted head may deviate from the fusion model (see Fig. 9). Non-rigid transformation might be considered to get more seamless head fitting (e.g., using the method of [Li et al. 2017]). Second, although our algorithm is robust against missing regions, the final 3D hair could still get imperfect results in some fine detail regions if they are not fully captured by the fusion model, i.e., there are no geometric cues to guide our strand synthesis (see Fig. 8, 12). We believe this is a common drawback for all current lightweight depth sensors. Third, we do not handle constrained hairstyles such as braids and buns. To create constrained braids which are different from the natural hairs, separate identification and processing of the braid structure should be done such as the methods in [Hu et al. 2014a,b]. In addition, our method may not work well for long highly curly hairstyles such as the failure case shown in [Zhang et al. 2017] due to the highly convoluted structures. Finally, our framework is off-line, and we hope that we can extend our data-driven method to online screened hair modeling.

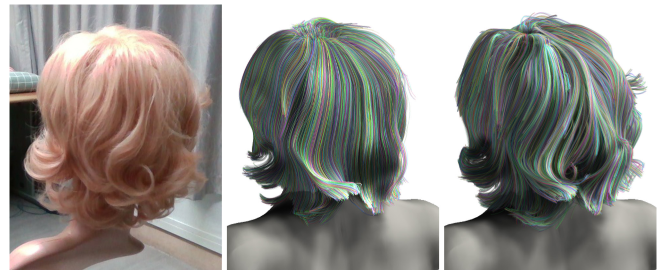

Fig. 11. A direct growing of hair along the 3D orientation field may lead to over-smoothed result (middle), while our data-driven hair synthesis produces higher quality hair strands (right).

8 CONCLUSION

We have presented a data-driven approach to model the hair geometry and compute a strand-level 3D hair model by using a single RGB-D camera. The core of our pipeline is a 3D PatchMatch-based local search and volumetric synthesis algorithm that simultaneously reasons about the hair geometry, strands connectivity, strand orientation, as well as hair structural plausibility. We extend the single-view and multi-view image-based 3D hair modeling to achieve more complete and realistic 3D hairs by utilizing depth fusion and 3D PatchMatch. To the best of our knowledge, our method is the first fully automatic method that can generate 3D hairs of higher quality than those of the state-of-the-art approaches. Finally, our approach also opens up possibilities for future researches including depth-based style-constrained hair modeling and online screened hair modeling.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their constructive comments; the gamers and artists for making their hair models available on The Sims Resource and Newsea platform; Monika Teiserskyte, Thauana de Morais, Kaizhang Kang, Xinru Hu, and Qin Lv for being our hair models. This work is partially supported by the NSF China (61572429, 61502306, U1609215, 61772457), and the China Young 1000 Talents Program.

REFERENCES

Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. 2009. PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing. *ACM Trans. Graph.* 28, 3 (2009), 24:1–24:11.

Connelly Barnes, Eli Shechtman, Dan B. Goldman, and Adam Finkelstein. 2010. The Generalized Patchmatch Correspondence Algorithm. In *ECCV'10*. 29–43.

Connelly Barnes and Fang-Lue Zhang. 2017. A survey of the state-of-the-art in patch-based synthesis. *Computational Visual Media* 3, 1 (2017), 3–20.

Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. 2014. Face alignment by explicit shape regression. *International Journal of Computer Vision* 107, 2 (2014), 177–190.

Menglei Chai, Linjie Luo, Kalyan Sunkavalli, Nathan Carr, Sunil Hadap, and Kun Zhou. 2015. High-quality hair modeling from a single portrait photo. *ACM Trans. Graph.* 34, 6 (2015), 204:1–204:10.

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 (2016), 116:1–116:12.
Menglei Chai, Lvdi Wang, Yanlin Weng, Xiaogang Jin, and Kun Zhou. 2013. Dynamic Hair Manipulation in Images and Videos. *ACM Trans. Graph.* 32, 4 (2013), 75:1–75:8.
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 (2012), 116:1–116:8.
Nicolas Cherin, Frederic Cordier, and Mahmoud Melkemi. 2014. Modeling piecewise helix curves from 2D sketches. *Computer-Aided Design* 46 (2014), 258–262.
Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. 2017. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Trans. Graph.* 36, 3 (2017), 76:1–76:19.
Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B. Goldman, and Pradeep Sen. 2012. Image Melding: Combining Inconsistent Images Using Patch-based Synthesis. *ACM Trans. Graph.* 31, 4 (2012), 82:1–82:10.
Jose I Echevarria, Derek Bradley, Diego Gutierrez, and Thabo Beeler. 2014. Capturing and stylizing hair for 3D fabrication. *ACM Trans. Graph.* 33, 4 (2014), 125:1–125:11.
Electronic Arts. 2014. The Sims Resource. http://www.thesimsresource.com/.
Tomas Lay Herrera, Arno Zinke, and Andreas Weber. 2012. Lighting hair from the inside: A thermal approach to hair reconstruction. *ACM Trans. Graph.* 31, 6 (2012), 146:1–146:9.
Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. 2014a. Robust hair capture using simulated examples. *ACM Trans. Graph.* 33, 4 (2014), 126:1–126:10.
Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. 2015. Single-view hair modeling using a hairstyle database. *ACM Trans. Graph.* 34, 4 (2015), 125:1–125:9.
Liwen Hu, Chongyang Ma, Linjie Luo, Li-Yi Wei, and Hao Li. 2014b. Capturing Braided Hairstyles. *ACM Trans. Graph.* 33, 6 (2014), 225:1–225:9.
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 (2017), 195:1–195:14.
Wenzel Jakob, Jonathan T Moon, and Steve Marschner. 2009. Capturing hair assemblies fiber by fiber. *ACM Trans. Graph.* 28, 5 (2009), 164:1–164:9.
D. Li, T. Shao, H. Wu, and K. Zhou. 2017. Shape Completion from a Single RGBD Image. *IEEE Trans. Vis. Comp. Graph.* 23, 7 (2017), 1809–1822.
Linjie Luo, Hao Li, and Szymon Rusinkiewicz. 2013. Structure-aware hair capture. *ACM Trans. Graph.* 32, 4 (2013), 76:1–76:12.
Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. 2011. KinectFusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*. IEEE, 127–136.
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 (2008), 30:1–30:9.
Shuran Song and Jianxiong Xiao. 2013. Tracking revisited using RGBD camera: Unified benchmark and baselines. In *IEEE International Conference on Computer Vision (ICCV)*. 233–240.
Sebastien Valette and Jeanmarc Chassery. 2004. Approximated Centroidal Voronoi Diagrams for Uniform Polygonal Mesh Coarsening. *Computer Graphics Forum* 23, 3 (2004), 381–389.
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 Trans. Vis. Comp. Graph.* 13, 2 (2007), 213–234.
Yichen Wei, Eyal Ofek, Long Quan, and Heung-Yeung Shum. 2005. Modeling Hair from Multiple Views. *ACM Trans. Graph.* 24, 3 (2005), 816–820.
Cem Yuksel, Scott Schaefer, and John Keyser. 2009. Hair meshes. *ACM Trans. Graph.* 28, 5 (2009), 166:1–166:7.
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 (2017), 156:1–156:11.
Qian-Yi Zhou and Vladlen Koltun. 2014. Color Map Optimization for 3D Reconstruction with Consumer Depth Cameras. *ACM Trans. Graph.* 33, 4 (2014), 155:1–155:10.

Fig. 12. Our hair modeling results. Each row shows the selected reference color images, the fusion model, and four views of the hair modeling results.