

Capturing Braided Hairstyles

Liwen Hu[†]

Chongyang Ma^{† *}

Linjie Luo[‡]

Li-Yi Wei[§]

Hao Li[†]

[†] University of Southern California

[‡] Adobe Research

[§] The University of Hong Kong

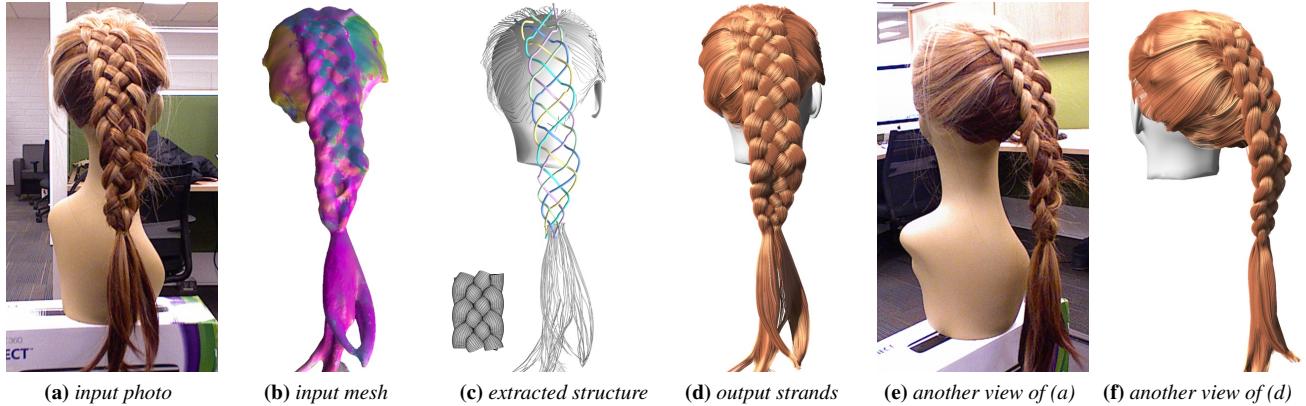


Figure 1: Capturing a five-strand Dutch braid. We capture the braided hairstyle (a) using a Kinect sensor and obtain an input mesh with a local 3D orientation for each vertex (b). Based on the information provided by the example patches in a database, we extract the centerlines (c) of the braid structure to synthesize final output strands (d). (e) and (f) show the input reference photo and our output strands from another viewpoint.

Abstract

From fishtail to princess braids, these intricately woven structures define an important and popular class of hairstyle, frequently used for digital characters in computer graphics. In addition to the challenges created by the infinite range of styles, existing modeling and capture techniques are particularly constrained by the geometric and topological complexities. We propose a data-driven method to automatically reconstruct braided hairstyles from input data obtained from a single consumer RGB-D camera. Our approach covers the large variation of repetitive braid structures using a family of compact procedural braid models. From these models, we produce a database of braid patches and use a robust random sampling approach for data fitting. We then recover the input braid structures using a multi-label optimization algorithm and synthesize the intertwining hair strands of the braids. We demonstrate that a minimal capture equipment is sufficient to effectively capture a wide range of complex braids with distinct shapes and structures.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms;

Keywords: hair capture, braids, data driven reconstruction

Links: DL PDF

*Corresponding author: chongyang.ma@usc.edu

1 Introduction

Whether as a fashion statement or an indicator of social status, braided hairstyles for both men and women have remained prevalent in many cultures for centuries. It is no surprise that digital storytellers frequently use characters with braids, ranging from casual ones to meticulously styled updos. In both film and game production, the hairstyles of digital humans need to be carefully designed as they often reflect a unique personality of the character. Typically, reference photographs or 3D scans of real actors are used as a starting point for the digitization. While sophisticated hair modeling tools [Choe and Ko 2005; Yuksel et al. 2009] have been introduced to improve the workflow of digital artists, significant effort is still required to create digital hair models that accurately match the input references. In feature films, modeling the hairstyle of a single character can take up to several weeks for an artist.

To facilitate the digitization of complex hair models, important advancements in 3D hair capture [Paris et al. 2008; Luo et al. 2013; Hu et al. 2014] have recently emerged to further reduce the manual effort of digital artists. Even though sophisticated hardware is necessary, these techniques can capture a wide spectrum of real world hairstyles using geometric and physics-driven priors for hair structural analysis. However, these structural analyses become problematic for *constrained* hairstyles such as braids since their priors cannot properly model the intertwined topologies of braids.

A great variety of complex braids can be generated by repeatedly applying several basic rules [Coefield 2013]. For instance, the most common basic braids with 3 strands (Figure 2a) are generated by repeatedly crossing the current middle strand under the outside one (left or right) in an interleaved fashion. Other braid styles extend the basic one by varying the number of strands, crossing over or under, or merging extra hair strands after each crossing. More complex braids include the Princess Anne style (Figure 14, bottom). These basic rules are generalized and systematically studied in braid theory [Artin 1947] as the generators for a braid group B_n with n strands. Although in theory, an infinite number of braids exist in B_n , braided hairstyles in daily life are typically limited to a couple of styles where



(a) basic braid



(b) Five-strand Dutch braid

Figure 2: Making braids. Top: a three-strand basic braid; bottom: a five-strand Dutch braid for Figure 1. Individual steps of making each braid style are shown from left to right.

each strand is symmetric to another for simplicity and aesthetics.

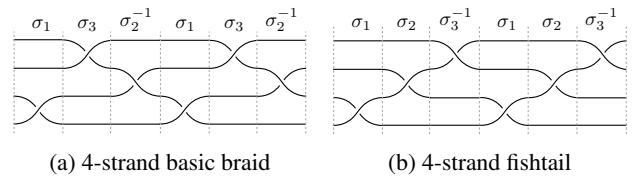
In this paper, we develop the first 3D hair reconstruction framework that enables the acquisition of wide range of different braided hairstyles. In addition to greatly increasing the space of capturable hair variation, our system requires only a commodity depth sensor as input, producing results comparable to existing high-end hair capture techniques. Inspired by the simplistic generative rules of braids in both theory and practice, we use a family of compact procedural models based on sinusoidal functions to represent common braided hairstyles. These procedural models can be used to fit the braid structures to the input data and resolve the structural ambiguities caused by occlusion. We show that our system can faithfully recover complex intertwined patterns and generate structurally plausible braided hair models. We adopt a patch-based fitting algorithm based on random sampling and perform structure analysis to connect them into consistent braids. Finally, we synthesize output strands in the braid structures and combine with the remaining hair by diffusing the 3D orientation fields. Since our system only requires a consumer level depth sensor such as Microsoft’s Kinect, we can greatly reduce the cost of the acquisition hardware as compared to previous multi-view stereo systems (e.g. [Luo et al. 2013; Hu et al. 2014]).

Contributions. In summary, our contributions are:

- A novel hair reconstruction framework for braided hairstyles capable of extracting complex intertwined structures;
- A family of compact procedural braid models that represent a wide range of common braided hairstyles.
- A robust patch-based braid fitting and analysis approach that addresses flexible braid patterns.

2 Previous Work

Braid theory. Braid theory [Artin 1947] is an abstract geometric theory that studies the topology of braids as mathematical groups. In an n -strand braid group B_n , each braid can be represented by a *braid word* concatenating the generators $\{\sigma_i\}$ and their inverse $\{\sigma_i^{-1}\}$. There are $n - 1$ generators in B_n in total. For example, the common 3-strand basic braid can be represented as $\sigma_1\sigma_2^{-1}$; the



(a) 4-strand basic braid

(b) 4-strand fishtail

Figure 3: Representing common braid styles by braid words in braid theory. As shown above, the 4-strand basic braid can be written as $\sigma_1\sigma_3\sigma_2^{-1}$ and 4-strand fishtail as $\sigma_1\sigma_2\sigma_3^{-1}$.

4-strand Fishtail can be represented as $\sigma_1\sigma_2\sigma_3^{-1}$ (Figure 3). These representations can be generalized to n -strand braids: n -strand basic braids as $\Pi_{i \text{ odd}} \sigma_i \Pi_{i \text{ even}} \sigma_i^{-1}$ and fishtails as $\Pi_{i \leq \frac{n}{2}} \sigma_i \Pi_{i > \frac{n}{2}} \sigma_i^{-1}$. In a similar spirit, our procedural braid models use periodic sinusoidal functions to model the braid strands as they are woven down.

Hair modeling. A long line of research has been devoted to hair modeling as surveyed in [Ward et al. 2007]. Pure manual editing remains a popular choice for experienced artists to create hair models from scratch for maximum flexibility. To avoid the daunting task of manipulating individual hair fibers, these systems usually provide some kinds of geometric proxy, such as guide strands or polygonal surfaces for more intuitive design and manipulation of hair [Choe and Ko 2005; Fu et al. 2007; Wither et al. 2007; Yuksel et al. 2009; Ghoniem and Museth 2013]. Further realism can be achieved through physical simulation as a post-process [Selle et al. 2008; Derouet-Jourdan et al. 2013]. While it is possible to manually create braided hairstyles from scratch as demonstrated in [Yuksel et al. 2009], complex and realistic ones are very difficult and time consuming to produce (often requiring up to several weeks for a skilled artist).

Hair capture. To reduce the amount of manual labor required, a common approach is to directly capture real hairstyles and reconstruct 3D hair models [Wei et al. 2005; Paris et al. 2008; Jakob et al. 2009; Herrera et al. 2012; Luo et al. 2012; Luo et al. 2013; Echevarria et al. 2014; Hu et al. 2014]. One well-established pipeline consists of first capturing the raw hair geometry through passive or active stereo techniques, then computing the orientation fields, and finally producing the hair strands following the geometry and input orientation field [Luo et al. 2012; Luo et al. 2013; Hu et al. 2014]. However, these approaches are not designed for constrained hairstyles, and typically fail on braided hairstyles, producing wrong hair strand connections during the reconstruction.

There are also a number of hybrid approaches in existence that attempt to combine the flexibility and robustness of manual modeling with the accuracy and efficiency of capturing techniques. For example, the methods by [Chai et al. 2012; Chai et al. 2013; Yu et al. 2014] allow the users to manually sketch over input images and/or videos to guide the modeling and editing process. However, as pointed out in [Yu et al. 2014], capturing and modeling braids remain a major challenge, especially for the more complex structures. Even manual sketching techniques over captured 3D data have had limited success in reconstructing constrained hairstyles and only the digitization of very simple ponytails has been demonstrated [Hu et al. 2014].

Structure-aware shape processing. Our proposed technique is related to existing structure-aware shape processing methods used for modeling man-made objects. A survey can be found in [Mitra et al. 2014]. By leveraging prior knowledge from existing databases,

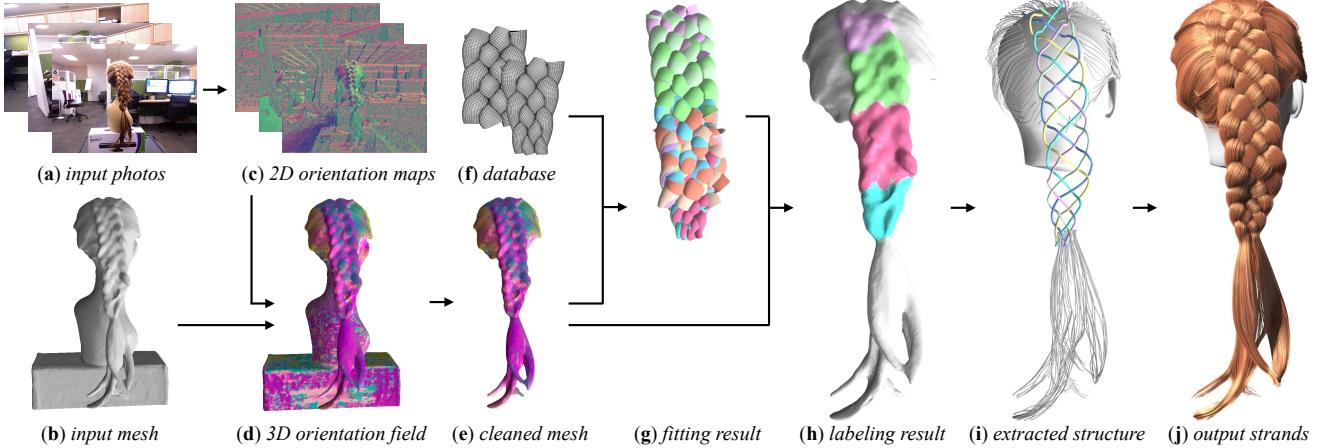


Figure 4: Overview of our system pipeline. More detailed descriptions are in Section 3.

different approaches have been proposed to analyze and reconstruct indoor scenes [Nan et al. 2012; Shao et al. 2012; Kim et al. 2012], component-based models [Li et al. 2011; Shen et al. 2012], and dense foliage [Bradley et al. 2013]. Compared to prior methods [Li et al. 2010; Huang et al. 2013] that recover 1D structures from point cloud data, our approach can handle multiple interleaved 1D structures such as braids.

3 Overview

The overview of our pipeline is shown in Figure 4. Our input is a point cloud with normals and local hair orientations at each point for the captured hairstyle. The point cloud can be derived from the vertices of a mesh reconstructed by a consumer depth camera [Newcombe et al. 2011] or directly from multi-view stereo [Furukawa and Ponce 2010]. The local 3D orientation for each point (Figure 4d) is computed by maximizing the consistency of projected directions across a set of 2D orientation maps (Figure 4c) as in [Luo et al. 2013]. We manually remove the non-hair part from the point cloud (Figure 4e). This is the only mandatory manual step of our entire pipeline.

We generate a database of a variety of example patches via procedural braid models (Figure 4f, Section 4). Each patch is a group of surface tubes with their centerlines. We then adopt a fitting method based on random sampling to align the input point cloud with the patches in the database and derive a set of candidate examples matching the braided part (Figure 4g, Section 5). To speed up computation, users can optionally perform the following two types of manual input: (1) choosing a braid example from the database based on visual observation, and (2) providing a rough initial scale which narrows the search range.

Next, we compute the optimal subset of the candidate patches to disjointly cover the input braid via multi-label optimization (Figure 4h), and extract the structure of the braided part as the connected centerlines of the covering braid patches (Figure 4i, Section 6). Finally, we follow the recovered braid structure with the rest of the unbraided hair region to synthesize the output strands guided by a diffused 3D orientation field (Figure 4j, Section 7).

4 Procedural Model

Inspired by the hair braiding process and braid theory, we use a few intertwining centerlines to represent the braid structures. These

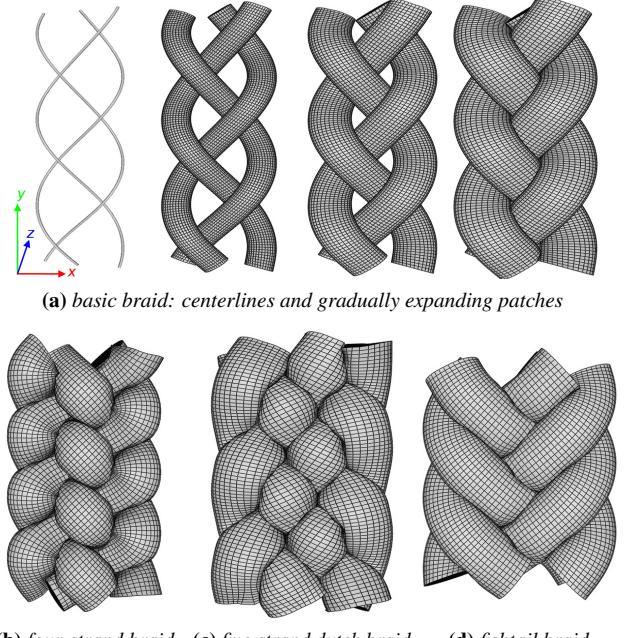


Figure 5: Database preparation. (a) shows the centerlines of a three-strand braid (left), two intermediate results of the patch expanding process (middle two), and the final expanded patch (right). (b), (c) and (d) show the expanded patches for a four-strand braid, a five-strand dutch braid and a fishtail braid respectively.

centerlines are symmetric to each other and have repeating patterns that can be characterized by periodic sinusoidal functions. Basic braids (Figure 2a) can be described by three centerlines $\{\mathbf{L}_i, i = 0, 1, 2\}$ in its natural frame as follows:

$$\begin{aligned} \mathbf{L}_0 : x &= a \sin(t), & y &= t, z = b \sin(2t) \\ \mathbf{L}_1 : x &= a \sin(t + 2\pi/3), & y &= t, z = b \sin(2(t + 2\pi/3)) \\ \mathbf{L}_2 : x &= a \sin(t + 4\pi/3), & y &= t, z = b \sin(2(t + 4\pi/3)) \end{aligned} \quad (1)$$

where the y is the braiding direction, and a and b are two constants that determine the shape of the braid. Other braids can be similarly modeled according to how the strands are woven. We provide a

few more procedural braid models as shown in Appendix A.1. Note that the procedural forms are by no means exhaustive: users can provide additional parametric forms either procedurally (as above) or manually (e.g. sketching a few curves via 3D authoring tools).

In order to model the strand thickness in real braids, we augment the procedural model by expanding the centerlines to tubes. To compute the proper radius of these tubes, we expand an initially small radius until inter-tube penetrations occur.

We use each procedural model to generate a segment of the target braid as an example patch \mathcal{P}_e and collect them into a single database \mathcal{D} (Figure 5). Each patch in the database is a set of helical tubes with local orientation defined at each vertex $v_i \in \mathcal{P}_e$. The vertex orientation is defined to be along the direction of the centerline.

5 Patch Fitting

Given the input geometry of the captured hair, our goal is then to identify the local braid structure of the hairstyle by fitting braid patches from the database to the point cloud. There are two major challenges we need to address here. First, almost all braided hairstyles in real life exhibit rich variations in terms of type, orientation, and size. For example, braids can have different numbers of strands, and the orientation and the size of each knot can change along the braids. Second, the input geometry is usually very noisy and largely occluded due to hair and body contacts.

We first align the input point cloud \mathcal{C} with each braid patch in the database \mathcal{D} to collect a few candidate matches using a fitting approach based on random sampling similar to [Bradley et al. 2013; Hu et al. 2014]. Specifically, for each example patch $\mathcal{P}_e \in \mathcal{D}$, we apply the fitting algorithm N times, each time starting from a random initial position, orientation, and scale for the patch. The initial position of the example patch is determined by randomly sampling a point on the captured surface, and the initial orientation of the patch is computed from a randomly normalized quaternion. The initial scale is computed by randomly scaling between $0.9 \sim 1.1$ of an estimated scale, which is obtained by manually scaling and matching the braid patches to the braids in the input point cloud. Here, we use a simple interactive tool by drawing three lines on the mesh surface to indicate the x, y, z scales. Note that since we adopt a strategy based on random sampling, we only need a roughly estimated scale for our fitting algorithm to work and this has to be done only once for a given example patch. We found that satisfactory results are obtained for $N = 100$ in all our examples.

Next, we compute the optimal transformation \mathbf{T} to align the scaled patch \mathcal{P}_e with \mathcal{C} via Iterative Closest Point (ICP) method [Besl and McKay 1992]. To take into account the variations of braids in the input hairstyle and to achieve better fitness, we adopt both, a rigid and non-rigid ICP [Li et al. 2009] approach to deform \mathcal{P}_e . (An example comparison is shown in Figure 6.) The fitting error is computed as:

$$E(\mathcal{C}, \mathcal{P}_f) = \sum_i |\mathbf{p}(v_{c,i}) - \mathbf{p}(\hat{v}_{e,i})|^2 + \alpha |\mathbf{o}(v_{c,i}) - \mathbf{o}(\hat{v}_{e,i})|^2 \quad (2)$$

where \mathbf{p} and \mathbf{o} indicate the position and orientation of the corresponding point; $\mathcal{P}_f = \mathbf{T}(\mathcal{P}_e)$ the fitted patch; $\hat{v}_{e,i}$ the transformed position of the vertex in \mathcal{P}_f with closest point $v_{c,i} \in \mathcal{C}$, and α a constant weight which is fixed to be 100 for all of our results. Empirically, we have found that good results can be obtained with 30 iterations of rigid ICP followed by 30 iterations of non-rigid ICP.

Note that the tail of the braid are usually tapered and can cause large errors when fitting the braid patches. To faithfully fit the tail of

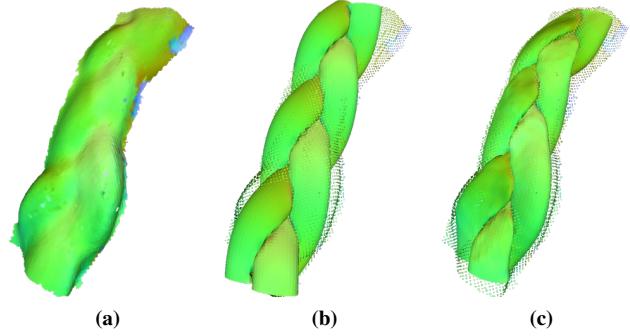


Figure 6: Comparison between fitting with rigid ICP and non-rigid ICP. Here we show (a) input mesh and the fitted patches with rigid ICP and non-rigid ICP in (b) and (c) respectively.

a braid, we also introduce a *tail patch* for each example patch in the database. Each tail patch is automatically created by linearly tapering the example patch towards the end such that the width and the thickness of the tubes are halved compared to the beginning along the y -axis. See Figure 4f for a concrete example.

6 Structure Analysis

Our goal in this section is to select the optimal set of fitted patches from the candidate set and extract the coherent braid structure of the input hairstyle. The process is visualized in Figure 7.

The patch fitting step (Section 5) produces a set of fitted patches $\{\mathcal{P}_f\}$. Each fitted patch *covers* a subset of points $\{v_c\}$ in the input point cloud \mathcal{C} if v_c is within a distance threshold to some vertex $\hat{v}_e \in \mathcal{P}_f$. We collect all the vertices $\{v_c\}$ covered by at least one fitted patch \mathcal{P}_f , and consider them as the braided part \mathcal{C}_b of \mathcal{C} , whose structure can be analyzed by the procedural braid models in the database. Since $\{\mathcal{P}_f\}$ may be redundant and overlap with each other, one v_c can be covered by multiple fitted patches (see Figure 7b). We need to select a subset of the fitted patches so that we can connect them together to form a complete and clean structure for the braided part.

We formulate the structure analysis task as a multi-label problem. Specifically, for each point $v_i \in \mathcal{C}_b$, we choose a single label l_i which corresponds to a fitted patch $\mathcal{P}_{f,i}$ by minimizing the following energy function:

$$E(\mathcal{C}_b, \{l_i\}) = \sum_i E_d(v_i, l_i) + \sum_{i,j \in N(i)} E_s(v_i, v_j, l_i, l_j) \quad (3)$$

The first term in Equation 3 is a data term to ensure that each point v_i is assigned to a label such that it is covered by the corresponding fitted patch $\mathcal{P}_{f,i}$:

$$E_d(v_i, l_i) = \begin{cases} E(v_i, \mathcal{P}_{f,i}) & \text{if } \mathcal{P}_{f,i} \text{ covers } v_i \\ 100 & \text{otherwise} \end{cases} \quad (4)$$

where $E(v_i, \mathcal{P}_{f,i})$ is the fitting error computed based on Equation 2. We also ensure that a sufficiently large penalty factor is used (100). The second term in Equation 3 is a smoothness term that tries to assign the same label to vertices that are within a distance threshold and have similar local orientations:

$$E_s(v_i, v_j, l_i, l_j) = \begin{cases} 0 & \text{if } l_i = l_j \\ 100 & \text{otherwise} \end{cases} \quad (5)$$

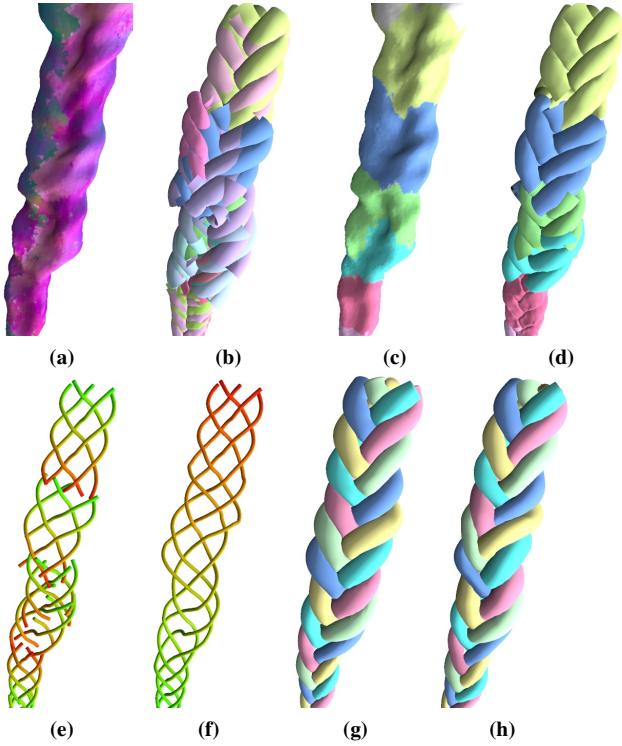


Figure 7: Illustration of our structure analysis algorithm. Here we show (a) input mesh; (b) candidate patches collected during the fitting stage; (c) colorized visualization of labeling result on input mesh; (d) patches selected by the labeling step; (e) centerlines of the selected patches; (f) connected centerlines; (g) surface expanded from the connected centerlines; (h) fitted surface after non-rigid ICP.

The energy function in Equation 3 can be efficiently minimized using a graph-cuts algorithm [Delong et al. 2012]. See Figure 7c for a color visualization of the labeling result on the input mesh.

Based on the labeling result, we can select an optimal subset of fitted patches $\{\mathcal{P}_f\}$ whose corresponding labels have been assigned to certain points in C_b (Figure 7d). Now we need to connect different sets of centerlines for adjacent patches to generate a complete braid structure. We pose this task as an assignment problem. Specifically, we consider all possible matching combinations of the centerlines for every two adjacent patches. For each possible combination, we compute an assignment cost as the average distance between the corresponding points in the overlapping regions of all the combined pairs of centerlines. The overlapping region is defined to be the points on one centerline within a distance threshold from the other centerline. If the overlapping region for any pair of centerlines has less than two points, the assignment cost is infinite since there is insufficient overlap. We use the Hungarian algorithm [Kuhn 1955] to compute the optimal assignment and smoothly connect the two sets of centerlines based on the optimal combination. The final connected centerlines represents the structure of the entire braided part of the input hairstyle (Figure 7f).

Finally, we expand the connected centerlines using the same method for the database described in Section 4, in order to obtain the surface of the volumetric tubes for the braid (Figure 7g). To further improve the fitness to the input hairstyle, we use the non-rigid ICP algorithm to align the expanded surface with the captured geometry. The deformed surface represents the reconstructed volumetric tubes for the braided part of input hair (Figure 7h).

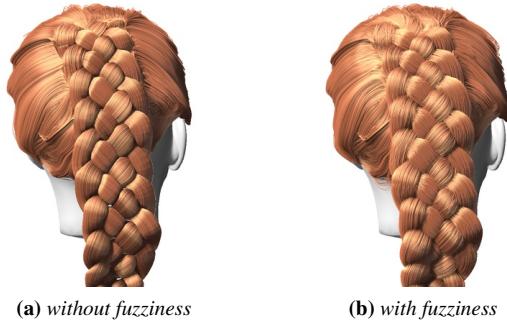


Figure 8: Adding fuzziness [Choe and Ko 2005] over the final output strands.

Discussion. An alternative way to obtain the structure of the braided part is to connect these selected surface patches $\{\mathcal{P}_f\}$ directly using a mesh composition approach [Huang et al. 2012]. We found that connecting the centerlines first, followed by expanding the surface with non-rigid ICP is easier to implement and produces good results in all our cases.

7 Strand Synthesis

After extracting the braid structure of the input hair, we synthesize full hair strands in both the braided and unbraided parts. The unbraided part consists of the points \mathcal{C} that are not covered by the fitted braid patches in the braid obtained in the fitting step in Section 5.

We first synthesize the strands on the unbraided part by growing bidirectionally from the uncovered points based on the local orientations as described in [Luo et al. 2013]. These synthesized strands are then used to build a 3D orientation field using the method of [Wang et al. 2009] and diffused into the entire output volume as in [Paris et al. 2008]. During this diffusion process we consider the centerline directions in the braided parts to ensure continuity with the unbraided parts. Next, we determine the root positions by randomly sampling a user-specified region on a manually prepared scalp. Finally we grow the output strands from the roots following the 3D orientation field. Once a strand grows into the braided part, i.e. one of the volumetric tubes obtained in Section 6, we compute the barycentric coordinates of the entry point with respect to the cross section of the tube, and interpolate all subsequent growing positions using the same barycentric coordinates along the tube.

To further improve the visual realism of the reconstruction results, we introduce some fuzziness into the sample positions of the final strands using the method of [Choe and Ko 2005] (See Figure 8).

8 Results

Capture setup. Our algorithm supports different capturing systems, and does not require precise control of the environments. Except for the basic braid case in Figure 14 which is captured using a multi-view stereo (as shown in Figure 9b), all other results are obtained by slowly moving a hand-held Kinect camera around the targets (as shown in Figure 9a) for realtime geometry reconstruction using KinectFusion [Newcombe et al. 2011].

Hairstyles. To show the effectiveness of our method, we have tested our framework with a variety of different braided hairstyles. As illustrated in Figure 1 and 14, our method can successfully

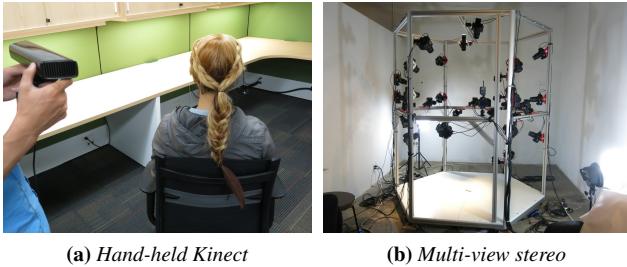


Figure 9: Our capture setups.



Figure 10: Comparisons with state-of-the-art hair capture methods.

reconstruct complex braided hairstyles with different numbers of strands per braid, different numbers of braids, different braid geometry (varying width and thicknesses with global twist and bending), and different topology (e.g., the merged style at the bottom of Figure 14).

Comparisons. In Figure 10 we compare our method with two state-of-the-art hair capture techniques [Luo et al. 2013; Hu et al. 2014], using the same input point cloud and orientation field. Luo et al. [2013] connects hair threads via geometric heuristics. Such heuristics often fail when there are significant occlusions in braids, causing broken hair threads in the final reconstruction. Hu et al. [2014] uses database of example strands from pre-simulation and user sketches to produce better customized hair connections. However, the strand groups can still yield incorrect topological configuration due to potential ambiguity in the orientation fields, e.g. at adjacent segments coming from different braids. In contrast, our approach produces structurally correct strands for braided hairstyles by leveraging our procedurally generated patches.

Evaluations. To evaluate the robustness of our algorithm (see Figure 11), we reconstruct the same input hairstyle of a fishtail braid (Figure 11a) using different example patches. Specifically, we restrict the number and type of example patches within the database and first check if our method produces smaller fitting errors for more suitable patches. We also see if our technique generates plausible results even with examples that are different from the captured input. As demonstrated, our method can successfully distinguish between different example patches based on the fitting errors. Moreover plausible reconstructions can be obtained even without structurally correct examples in the database. Figure 12 further demonstrates the robustness of our method against the estimated scale of example patches. Specifically, we uniformly scale the example patches in Figure 4f by different factors s before running our patch fitting and structure analysis algorithm. As shown in Figure 12, our method produces good results with a wide range of estimated scales.

Implementation. The input hairstyles shown in this paper are manually woven and styled, taking several minutes for an inexperienced person per hairstyle. The most time-consuming stage in our pipeline is the input pre-processing part. Computing a 3D orientation field from a set of 2D images takes about 40 minutes, while manual clean-up of the mesh takes about half an hour. The patch fitting step requires five minutes of computation for one patch using 100 different initial poses (parallelized on four cores and eight threads). In our implementation, all the numbers length related measurements are in millimeters and all angles are in radians. We compute the distance in Section 6 according Section 5, and set the threshold to 100 in all our results except 200 for Figure 11c and Figure 11d. The grid size for an orientation field in Section 7 is set to 1mm , and building a vector field takes about five minutes. The number of final output strands ranges from $30K$ to $50K$. All the other computations can be done within seconds (on a 2.6 GHz Intel Core i7 and 16 GB RAM machine).

9 Discussion and Limitations

We have introduced a data-driven framework to reconstruct large varieties of complex braided hairstyles. Even with a consumer level RGB-D camera (Kinect) as input, we can generate compelling results on par with existing state-of-the-art multi-view stereo techniques. Our method is robust against parameter selection and can produce plausible reconstructions even when examples in the database are completely different from the input braids.

As a data-driven approach, the effectiveness of our method and the quality of our reconstruction results is determined by the input data, including both capture accuracy as well as the prior braided structures in the database. We cannot ensure that the recovered structure fully complies with the input hairstyles, especially when the captured data is noisy and incomplete. Furthermore, the tail part of a braid often exhibits highly irregular deformations. We have incorporated additional patch tapering heuristics to improve the visual appearance of the results (see Figure 13 for a failure case).

Similar to prior methods on hair capture [Luo et al. 2013; Echevarria et al. 2014], our method requires the user to manually segment the hair parts from the captured geometry. Fully automatic hair segmentation techniques based on data-driven and orientation field analysis could significantly reduce the manual effort in the reconstruction process.

Another limitation of our method is the need of specifying a rough scale for the example patches in the database. Inappropriate scaling factors could lead to inaccurate reconstructions since the example patches can fail to fit the captured input geometry. One solution is to enumerate more examples of different scales in the database, but the computational cost can increase significantly. It would be interesting to adopt an analysis method such as [Huang et al. 2014] to automatically extract the scaling factors by detecting the repetitive and near-regular structure in the braided hairstyles.

While a large range of braided hairstyles are covered by our approach, there are still certain types that are difficult to handle. For instance, a French rope braid has clusters of strands that twist around the centerline, causing the local orientation of hair strands to no longer follow the centerline of the cluster. Here, our database generation can be extended with additional procedural rules. We leave the construction of a universal hairstyle database as future work. Such a database could be obtained through manually designed priors (e.g. [Yuksel et al. 2009]) and parametric forms as proposed in this work.

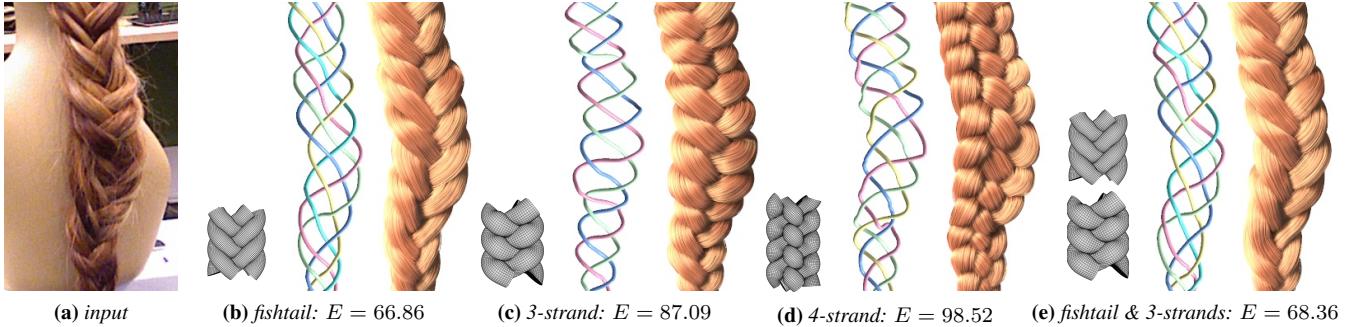


Figure 11: Comparisons with different example patches. We fit a fishtail braid (a) using different example patches in our database (b) - (e); shown in each are the example patch(es), the extracted structure, the output strands, and the fitting error E computed via Section 5.

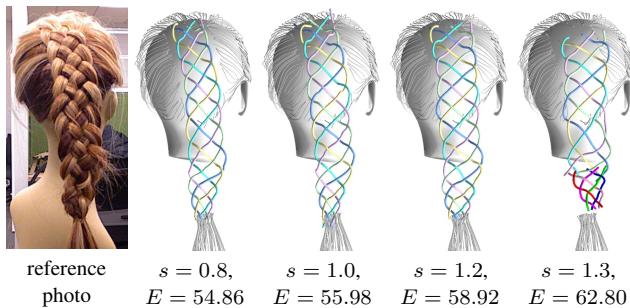


Figure 12: Comparisons with different estimated scales for the example patch in the database.



Figure 13: Failure case.

Acknowledgements

We would like to thank Frances Chen, Iris Wu, and Kirsty Lingman for being our capture models; Szymon Rusinkiewicz for the rendering code; Kyle Olszewski and Jessica Rossky for proofreading; Ecco Studios for the multi-view stereo capture; and the reviewers for their valuable suggestions. This project was supported in part by the USC's Integrated Media Systems Center (IMSC), the general research fund *Dynamic Element Textures* (HKU 717112E), Adobe Systems, Intel, NVIDIA, and Pelican Imaging.

References

- ARTIN, E. 1947. Theory of braids. *Annals of Mathematics Second Series* 48, 1, 101–126.
- BESL, P., AND MCKAY, N. D. 1992. A method for registration of 3-d shapes. *IEEE Trans. on PAMI* 14, 2, 239–256.
- BRADLEY, D., NOWROUZEZAHRAI, D., AND BEARDSLEY, P. 2013. Image-based reconstruction and synthesis of dense foliage. *ACM Trans. Graph.* 32, 4, 74:1–74:10.
- CHAI, M., WANG, L., WENG, Y., YU, Y., GUO, B., AND ZHOU, K. 2012. Single-view hair modeling for portrait manipulation. *ACM Trans. Graph.* 31, 4, 116:1–116:8.
- CHAI, M., WANG, L., WENG, Y., JIN, X., AND ZHOU, K. 2013. Dynamic hair manipulation in images and videos. *ACM Trans. Graph.* 32, 4, 75:1–75:8.
- CHOE, B., AND KO, H.-S. 2005. A statistical wisp model and pseudophysical approaches for interactive hairstyle generation. *IEEE Trans. Vis. Comput. Graph.* 11, 2, 160–170.
- COEFIELD, S. 2013. *DIY Braids: From Crowns to Fishtails, Easy, Step-by-Step Hair Braiding Instructions*. Adams Media.
- DELONG, A., OSOKIN, A., ISACK, H. N., AND BOYKOV, Y. 2012. Fast approximate energy minimization with label costs. *International journal of computer vision* 96, 1, 1–27.
- DEROUET-JOURDAN, A., BERTAILS-DESCOBES, F., DAVIET, G., AND THOLLOT, J. 2013. Inverse dynamic hair modeling with frictional contact. *ACM Trans. Graph.* 32, 6, 159:1–159:10.
- ECHEVARRIA, J. I., BRADLEY, D., GUTIERREZ, D., AND BEELER, T. 2014. Capturing and stylizing hair for 3d fabrication. *ACM Trans. Graph.* 33, 4, 125:1–125:11.
- FU, H., WEI, Y., TAI, C.-L., AND QUAN, L. 2007. Sketching hairstyles. In *SBIM '07*, 31–36.
- FURUKAWA, Y., AND PONCE, J. 2010. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. PAMI* 32, 1362–1376.
- GHONIEM, A., AND MUSETH, K. 2013. Hair growth by means of sparse volumetric modeling and advection. In *SIGGRAPH '13 Talks*, 34:1–34:1.
- HERRERA, T. L., ZINKE, A., AND WEBER, A. 2012. Lighting hair from the inside: A thermal approach to hair reconstruction. *ACM Trans. Graph.* 31, 6, 146:1–146:9.
- HU, L., MA, C., LUO, L., AND LI, H. 2014. Robust hair capture using simulated examples. *ACM Trans. Graph.* 33, 4, 126:1–126:10.



Figure 14: Reconstruction results of different braided hairstyles. From left to right: original photo, 2D orientation map, input mesh to our core algorithm, labeling result, extracted structure, and rendering of final strands from two different viewing points.

- HUANG, H., GONG, M., COHEN-OR, D., OUYANG, Y., TAN, F., AND ZHANG, H. 2012. Field-guided registration for feature-conforming shape composition. *ACM Trans. Graph.* 31, 6, 179:1–179:11.
- HUANG, H., WU, S., COHEN-OR, D., GONG, M., ZHANG, H., LI, G., AND CHEN, B. 2013. L1-medial skeleton of point cloud. *ACM Trans. Graph.* 32, 4, 65:1–65:8.
- HUANG, Q., GUIBAS, L. J., AND MITRA, N. J. 2014. Near-regular structure discovery using linear programming. *ACM Trans. Graph.* 33, 3, 23:1–23:17.
- JAKOB, W., MOON, J. T., AND MARSCHNER, S. 2009. Capturing hair assemblies fiber by fiber. *ACM Trans. Graph.* 28, 5, 164:1–164:9.
- KIM, Y. M., MITRA, N. J., YAN, D.-M., AND GUIBAS, L. 2012. Acquiring 3d indoor environments with variability and repetition. *ACM Trans. Graph.* 31, 6, 138:1–138:11.
- KUHN, H. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly* 2, 1-2, 83–97.
- LI, H., ADAMS, B., GUIBAS, L. J., AND PAULY, M. 2009. Robust single-view geometry and motion reconstruction. *ACM Trans. Graph.* 28, 5, 175:1–175:10.
- LI, G., LIU, L., ZHENG, H., AND MITRA, N. J. 2010. Analysis, reconstruction and manipulation using arterial snakes. *ACM Trans. Graph.* 29, 6, 152:1–152:10.
- LI, Y., WU, X., CHRYSATHOU, Y., SHARF, A., COHEN-OR, D., AND MITRA, N. J. 2011. Globfit: Consistently fitting primitives by discovering global relations. *ACM Trans. Graph.* 30, 4, 52:1–52:12.
- LUO, L., LI, H., PARIS, S., WEISE, T., PAULY, M., AND RUSINKIEWICZ, S. 2012. Multi-view hair capture using orientation fields. In *CVPR*, 1490–1497.
- LUO, L., LI, H., AND RUSINKIEWICZ, S. 2013. Structure-aware hair capture. *ACM Trans. Graph.* 32, 4, 76:1–76:12.
- MITRA, N. J., WAND, M., ZHANG, H., COHEN-OR, D., KIM, V., AND HUANG, Q.-X. 2014. Structure-aware shape processing. In *ACM SIGGRAPH 2014 Courses*, 13:1–13:21.
- NAN, L., XIE, K., AND SHARF, A. 2012. A search-classify approach for cluttered indoor scene understanding. *ACM Trans. Graph.* 31, 6, 137:1–137:10.
- NEWCOMBE, R. A., IZADI, S., HILLIGES, O., MOLYNEAUX, D., KIM, D., DAVISON, A. J., KOHLI, P., SHOTTON, J., HODGES, S., AND FITZGIBBON, A. 2011. Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE ISMAR*, IEEE.
- PARIS, S., CHANG, W., KOZHUSHNYAN, O. I., JAROSZ, W., MATSIK, W., ZWICKER, M., AND DURAND, F. 2008. Hair photobooth: Geometric and photometric acquisition of real hairstyles. *ACM Trans. Graph.* 27, 3, 30:1–30:9.
- SELLÉ, A., LENTINE, M., AND FEDKIW, R. 2008. A mass spring model for hair simulation. *ACM Trans. Graph.* 27, 3, 64:1–64:11.
- SHAO, T., XU, W., ZHOU, K., WANG, J., LI, D., AND GUO, B. 2012. An interactive approach to semantic modeling of indoor scenes with an rgbd camera. *ACM Trans. Graph.* 31, 6, 136:1–136:11.
- SHEN, C.-H., FU, H., CHEN, K., AND HU, S.-M. 2012. Structure recovery by part assembly. *ACM Trans. Graph.* 31, 6, 180:1–180:11.
- WANG, L., YU, Y., ZHOU, K., AND GUO, B. 2009. Example-based hair geometry synthesis. *ACM Trans. Graph.* 28, 3, 56:1–56:9.
- WARD, K., BERTAILS, F., KIM, T.-Y., MARSCHNER, S. R., CANI, M.-P., AND LIN, M. C. 2007. A survey on hair modeling: Styling, simulation, and rendering. *IEEE Trans. Vis. Comput. Graph.* 13, 2, 213–234.
- WEI, Y., OFEK, E., QUAN, L., AND SHUM, H.-Y. 2005. Modeling hair from multiple views. *ACM Trans. Graph.* 24, 3, 816–820.
- WITHER, J., BERTAILS, F., AND CANI, M.-P. 2007. Realistic hair from a sketch. In *SMI '07*, 33–42.
- YU, X., YU, Z., CHEN, X., AND YU, J. 2014. A hybrid image-cad based system for modeling realistic hairstyles. In *3D '14*, 63–70.
- YUKSEL, C., SCHAEFER, S., AND KEYSER, J. 2009. Hair meshes. *ACM Trans. Graph.* 28, 5, 166:1–166:7.

A Appendix

A.1 Parametric Forms of Different Braids

The centerlines of a typical five-strand Dutch braid (Figure 5c) can be described as:

$$\begin{aligned} \mathbf{L}_0 : x &= a \sin(t), & y = t, z = b \sin(4t) \\ \mathbf{L}_1 : x &= a \sin(t + 2\pi/5), & y = t, z = b \sin(4(t + 2\pi/5)) \\ \mathbf{L}_2 : x &= a \sin(t + 4\pi/5), & y = t, z = b \sin(4(t + 4\pi/5)) \\ \mathbf{L}_3 : x &= a \sin(t + 6\pi/5), & y = t, z = b \sin(4(t + 6\pi/5)) \\ \mathbf{L}_4 : x &= a \sin(t + 8\pi/5), & y = t, z = b \sin(4(t + 8\pi/5)) \end{aligned} \quad (6)$$

Replacing the factor 4 for z coordinate in Equation 6 above will lead to a fishtail-like braid (Figure 5d). The centerlines of a typical four-strand braid (Figure 5b) can be described as:

$$\begin{aligned} \mathbf{L}_0 : x &= a \sin(t), & y = t, z = b \cdot f(t) \\ \mathbf{L}_1 : x &= a \sin(t + \pi/2), & y = t, z = b \cdot f(t + 1\pi/2) \\ \mathbf{L}_2 : x &= a \sin(t + \pi), & y = t, z = b \cdot f(t + \pi) \\ \mathbf{L}_3 : x &= a \sin(t + 3\pi/2), & y = t, z = b \cdot f(t + 3\pi/2) \end{aligned} \quad (7)$$

where

$$f(t) = \begin{cases} \sin(2t) & \text{if } 2n\pi \leq t < 2n\pi + \pi, n \in \mathbb{Z} \\ \sin(4t) & \text{otherwise} \end{cases} \quad (8)$$

In general, the following term describes the centerlines $\{\mathbf{L}_i\}$ of a braid with odd number of clusters:

$$\mathbf{L}_i : x = a \sin(t + i\pi/n), \quad y = t, \quad z = b \sin(m(t + i\pi/n)) \quad (9)$$

where m is an even number smaller than n . The centerlines $\{\mathbf{L}_i\}$ of a braid with even number of clusters are slightly more complex and can be described in a closed form as below:

$$\mathbf{L}_i : x = a \sin(t + i\pi/n), \quad y = t, \quad z = b \cdot f(t + i\pi/n) \quad (10)$$

where

$$f(t) = \begin{cases} \sin((n-2)t) & \text{if } 2m\pi \leq t < 2m\pi + \pi, m \in \mathbb{Z} \\ \sin(n \cdot t) & \text{otherwise} \end{cases} \quad (11)$$

In both Equation 9 and Equation 10, $i = 0, 1, \dots, n - 1$, while n is an odd number and an even number respectively. Please note that the parametric forms above only cover a subset of possible braids and there are many more possibilities in practice.