

Wide-baseline Hair Capture using Strand-based Refinement

Linjie Luo
Princeton University

Cha Zhang
Microsoft Research

Zhengyou Zhang
Microsoft Research

Szymon Rusinkiewicz
Princeton University

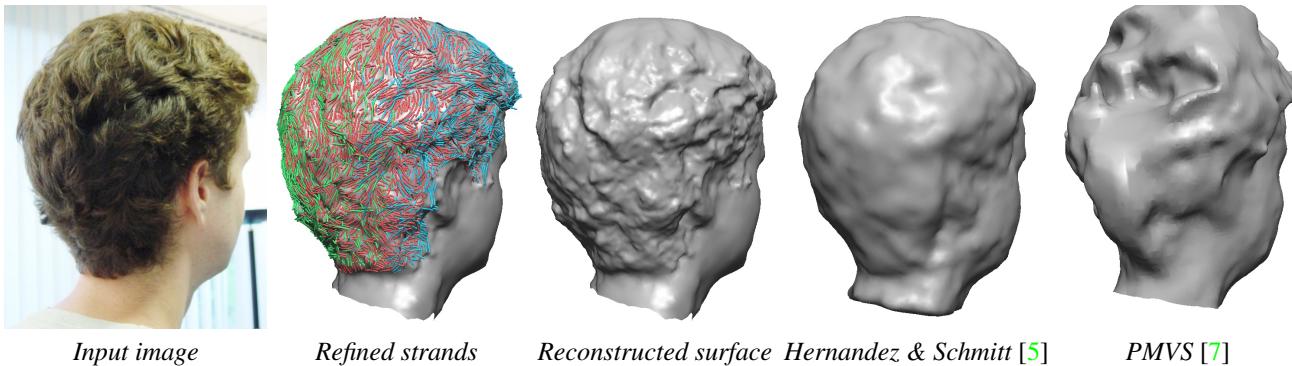


Figure 1: We begin with 8 input images from wide-baseline viewpoints, extract and refine the 3D strands (the strands from 3 adjacent views are shown in different colors), and reconstruct the surface from the positions of the refined strands. Our result is robust to the wide-baseline setup and reveals detailed hair structures. In contrast, general multi-view stereo methods based on texture are less accurate [5] or unable to converge in the wide-baseline setup [7].

Abstract

We propose a novel algorithm to reconstruct the 3D geometry of human hairs in wide-baseline setups using strand-based refinement. The hair strands are first extracted in each 2D view, and projected onto the 3D visual hull for initialization. The 3D positions of these strands are then refined by optimizing an objective function that takes into account cross-view hair orientation consistency, the visual hull constraint and smoothness constraints defined at the strand, wisp and global levels. Based on the refined strands, the algorithm can reconstruct an approximate hair surface: experiments with synthetic hair models achieve an accuracy of $\sim 3\text{mm}$. We also show real-world examples to demonstrate the capability to capture full-head hair styles as well as hair in motion with as few as 8 cameras.

1 Introduction

Multi-view stereo methods have been widely used to reconstruct real world objects with ever improving quality [7, 1]. However, hair reconstruction remains one of the most challenging tasks due to many unique hair characteristics. For instance, omni-present occlusions and complex strand geometry preclude general surface-based smoothness priors [26] for hair reconstruction. The highly specular nature of hair [17] also violates the Lambertian surface assumption employed in most multi-view stereo methods. Consequently, many practical systems have either completely

avoided hair reconstruction during facial capture (e.g. [1]), or relied on manual input to achieve plausible results [19].

Researchers have explored specialized hardware to facilitate hair capture, such as a fixed camera with moving light sources [21], a stage-mounted camera with macro lens [8], thermal imaging [12], etc. These methods are often costly, and require lengthy capture sessions that limit their applicability to only static hairstyles. An alternative approach is to deploy dense camera arrays that have small baselines. To capture complete full-head hairstyles, it is typical to have 20 to 30 camera views [25, 15, 12]. However, due to the complex hardware setup, it is challenging to adopt many cameras in real-world systems.

In this work, we study hair capture with a wide-baseline camera setup. Merely 8 cameras are used to capture the complete hair geometry, with each adjacent pair of cameras having a large 45-degree wide angular baseline. Under such a setup, stereo matching based on aggregation schemes such as local window or surface patch in existing methods [15, 25] is unreliable and error-prone. Instead, we propose that 3D strand is a better “aggregation unit” for stereo matching in hair reconstruction because it models hair’s characteristic “strand-like” structural continuity and thus yields improved robustness against matching ambiguities in wide-baseline setups. The 3D strands are first generated separately from a 2D strand extraction step in each view and then jointly optimized in a *strand-based refinement* step. We also introduce a novel formulation of smoothness energy that regularizes the optimization at the *strand*, *wisp* and *global* levels to better account for real hair dynamics, hair wisp structures and

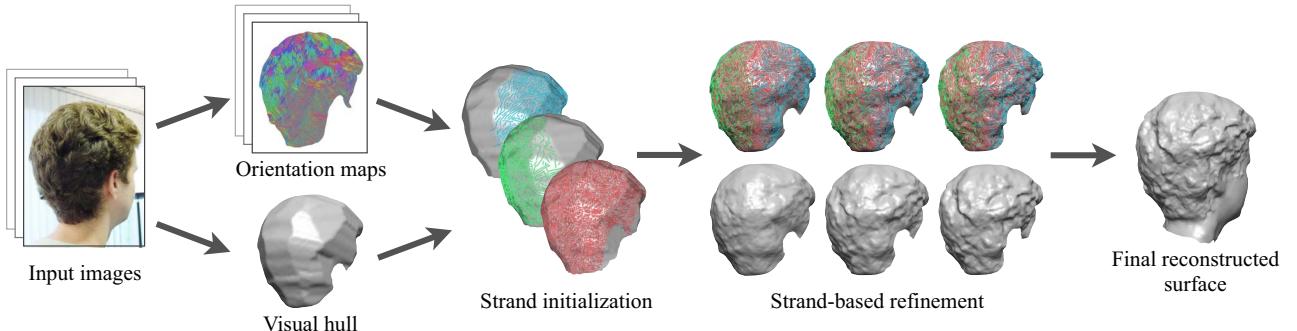


Figure 2: The overview of our reconstruction method. We take 8 input images from different views and compute the orientation map for each. Using the visual hull constructed from the segmented images, we extract strands on the orientation maps and project them from each view onto the visual hull for strand initialization. Finally we perform strand-based refinement to obtain the final strand positions. The hair surface can then be reconstructed from the refined strands using [10].

cross-view reconstruction consistency.

We quantitatively evaluate our reconstruction method on synthetic hairstyles, and achieve an accuracy of $\sim 3\text{mm}$. In addition, the approach can handle a wide variety of hairstyles in static images and dynamic sequences, as demonstrated with real examples in the results section and the supplemental materials.

2 Related Work

In this section, we review existing technologies for hair capture, including those using dedicated setups and dense camera arrays. In addition, we survey a few traditional multi-view stereo methods that are closely related to the proposed algorithm, including refinement-based reconstruction and wide-baseline stereo.

2.1 Hair Capture

A few dedicated systems in the literature have been designed for hair capture. Paris et al. [20] proposed to estimate the hair orientation in images and analyze the highlights on the hair. This analysis requires a fixed camera with a light source moving along a predefined trajectory. Later, Paris et al. [21] presented *Hair Photobooth*, a complex system made of several light sources, projectors, and video cameras that capture a rich set of data to extract the hair geometry and appearance. Jakob et al. [8] showed how to capture individual hair strands using focal sweeps with a macro-lens equipped camera controlled by a robotic gantry. Recently, thermal imaging has been applied for hair reconstruction to avoid shadowing and anisotropic reflectance [12]. While accurate, these techniques are expensive, and the capture process is usually slow and only applicable for static hairs.

Work has also been done to capture hair with more flexible setups. Wei et al. [25] proposed a technique based on many hand-held photographs. Their approach uses a coarse *visual hull* as the approximate bounding geometry for hair growing constrained with orientation consistency.

Yamaguchi et al. [27] used an array of 12 cameras to capture partial geometry of straight hair in moderate motion. Guided by hair simulation, Zhang et al. [29] reconstructed smooth hair dynamics with 7 cameras. Using only a single view, Chai et al. [4] proposed a method to generate a depth map for convincing view interpolation of different hairstyles. Beeler et al. [2] used a high resolution dense camera array to reconstruct facial hair strand geometry by matching distinctive strands. In contrast with these approaches, our method is capable of reconstructing accurate hair geometry from a wide-baseline sparse camera array.

2.2 Related Multi-view Stereo Methods

There have been many multi-view stereo methods presented in the literature [22]. The proposed method belongs to the general category of refining a rough initial geometry (e.g., a visual hull) by optimizing for cross-view consistency. And the consistency is measured in novel ways in order to handle the wide-baseline and challenging hair characteristics.

Space carving [11] reconstructs objects by eliminating voxels in a volume with low photo-consistency across visible views. Inspired by the active contour method [9], many reconstruction methods iteratively refine a rough initial shape (usually the visual hull) to obtain the final reconstruction by optimizing cross-view photo-consistency and surface smoothness. For instance, Hernandez and Schmitt [5] proposed a visual hull refinement method by iteratively minimizing the texture, silhouette and surface smoothness energies. Furukawa and Ponce [6] segmented the initial visual hull into surface areas between the rims and refined each via graph cuts.

Another important line of research for wide-baseline camera setups is to find robust feature correspondences between images. The well-known SIFT descriptor [14] is capable of finding feature points on images with significantly different viewpoints and illuminations. Tola et al. [23] extended the idea of SIFT to find dense correspondences



Figure 3: Our hair capture setup and a few sample images. We use 8 cameras (outlined in red) in wide-baseline to capture the complete hair styles. Four area lamps (outlined in blue) are used to compensate for the short exposure time.

across views for high quality reconstruction. Local regions with view-invariant properties have also been studied such as affinely invariant regions [24] and maximally stable extremal regions [18]. However, due to the lack of reliable texture and corner-like features on hair, it is difficult to apply these methods on hair reconstruction.

3 Overview

Given a set of wide-baseline images (see Fig. 3 for some sample images), our goal is to compute a shape that best approximates the captured hair volume. We achieve this by refining the positions of a dense set of representative 3D hair strands derived from each camera view.

Fig. 2 gives an overview of the various steps involved in our hair capture algorithm. We defer the description of our acquisition setup to Sec. 6. To create the initial 3D strands for refinement, we first compute the hair orientation map for each input image, and extract the 2D strands by tracking the confident ridges on the orientation map. The 2D strands are then back-projected onto the visual hull constructed from the segmented foreground of all input images to form the initial 3D strands. An iterative strand refinement algorithm is then applied to optimize the orientation consistency of the projected strands on all the orientation maps. We regularize the optimization with the silhouette constraint as well as a set of specialized smoothness priors for hair. The final hair shape is obtained using Poisson surface reconstruction [10] from the refined 3D strands.

In the rest of the paper, we will describe strand initialization in Sec. 4, and present the novel strand-based refinement algorithm in Section 5. Experimental results and conclusions are given in Sections 6 and 7, respectively.

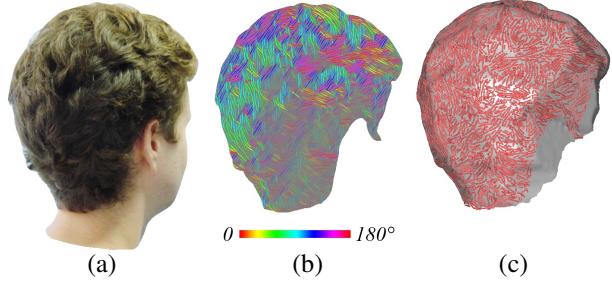


Figure 4: The steps of strand initialization in Sec. 4. For each input image (a), we compute the orientation map (b). Then we extract strands on the orientation map and project them onto the visual hull for initialization (c).

4 Strand initialization

We first compute an orientation map for each image using the method proposed in [15], which uses a bank of rotated filters to detect the dominant orientation at each pixel. The orientation map is enhanced with 3 passes of iterative refinement to improve the signal-to-noise ratio as in [4]. To further reduce noises in regions with low confidence, we apply the bilateral filtering method in [20] to diffuse the orientations of the high confidence regions.

We then track the confidence ridges of each orientation map (Fig. 4(b)) using hysteresis thresholding similar to [4]. The result is a set of poly-line 2D strands consisting of densely sampled vertices in about 1-pixel steps. We back-project each vertex of the resulting 2D strands onto the visual hull to determine the initial position of the 3D strands, as shown in Fig. 4(c). Note that the 3D strands are generally over-sampled after back-projection from 2D strands. Thus we down-sample each 3D strand by uniformly decimating the vertices to 20% of the original vertex count in order to reduce the computation cost in the following steps.

5 Strand-based refinement

After initializing the 3D strands from the 2D strands in each *reference view* (the view from which the strands were extracted), we iteratively refine all the strands by optimizing the projected orientation consistency across all visible views with silhouette and smoothness constraints (Fig. 5).

The optimization is formulated as an energy minimization problem. The total energy is defined as the weighted sum of a few specific energies, such as orientation energy, silhouette energy and smoothness energy:

$$E = \sum_{\star} \alpha_{\star} E_{\star} \quad (1)$$

where \star denotes each specific energy term as we will describe in detail in the following sections. All the energy terms are formulated in squared forms so that we can minimize the total energy with efficient non-linear solvers such as Levenberg-Marquardt [16].

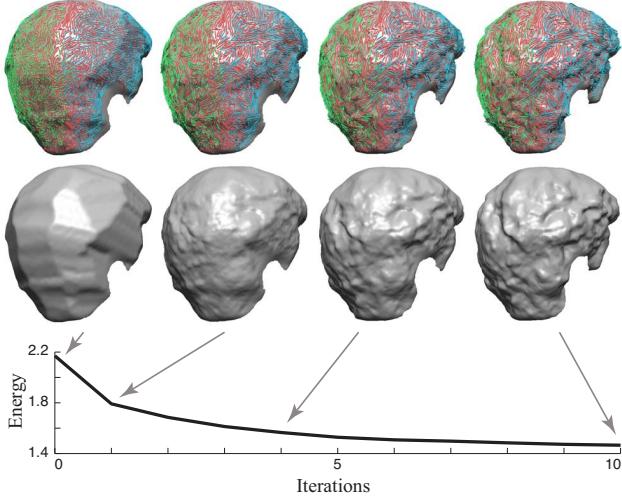


Figure 5: In strand-based refinement (Sec. 5), the strands (first row) are refined over the iterations with their reconstructed surfaces (second row) revealing more hair details.

5.1 Notations and Definitions

Let p denote a strand vertex on a 3D strand S . We use subscript to reference its successor p_{+1} and predecessor p_{-1} on S . Similarly, we can define $p_{+0.5}$ as the middle point between p and p_{+1} . The strand direction $d(p)$ at p is defined as $p_{+1} - p_{-1}$. The reference view of p is denoted as $R(p)$ and the visibility $\mathcal{V}(p)$ of p defines the set of views where p is visible. Since strand visibility is difficult to define exactly during strand refinement, we approximate $\mathcal{V}(p)$ by the visibility of its closest point $h(p)$ on the visual hull \mathcal{H} during the refinement. It is obvious to see that p 's reference view $R(p) \in \mathcal{V}(p)$.

We define two different neighborhoods for vertex p : the same-view neighborhood $\mathcal{N}^+(p)$ and the different-view neighborhood $\mathcal{N}^-(p)$, as shown in Fig. 6. $\mathcal{N}^+(p)$ is defined as the vertices from the *same* reference view as p and located within a certain 3D Euclidean distance from p . Vertices on the same strand as p are excluded from $\mathcal{N}^+(p)$. $\mathcal{N}^-(p)$ is defined similarly but the neighboring vertices are from *different* reference views.

Likewise, we define the same-view weight $w^+(p, q)$ between two vertices p and q if $q \in \mathcal{N}^+(p)$ and different-view weight $w^-(p, q)$ if $q \in \mathcal{N}^-(p)$. The different-view weight $w^-(p, q)$ is simply defined as the Gaussian weight:

$$w^-(p, q) = \exp\left(-\frac{\|p - q\|^2}{2\sigma_e^2}\right) \quad (2)$$

where σ_e controls the influence radius around the strand vertices and is set to 0.05 of the diagonal length D of the visual hull's bounding box. The same-view weight $w^+(p, q)$ is a bilateral weight that takes into account both the Euclidean

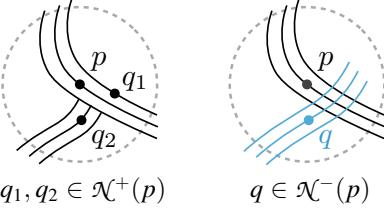


Figure 6: The illustrations of same-view neighborhood $\mathcal{N}^+(p)$ and different-view neighborhood $\mathcal{N}^-(p)$ for p . The neighbors are searched within the radius ($2.5\sigma_e$) indicated by the dashed circles. Same-view neighbors q_1 and q_2 can be weighted differently by how their orientations differ from p 's. The different-view neighbor q is located on the strands from a different reference view (in blue).

distance and the orientation difference between p and q :

$$w^+(p, q) = \exp\left(-\frac{1 - \langle d(p), d(q) \rangle^2}{2\sigma_o^2} - \frac{\|p - q\|^2}{2\sigma_e^2}\right) \quad (3)$$

where σ_o controls the influence between strand vertices with similar orientations and is set to 0.5. The notation $\langle A, B \rangle$ is defined as the cosine of the angle between two vectors A and B , i.e., $\langle A, B \rangle \triangleq A \cdot B / (\|A\| \|B\|)$. This applies to both 3D and 2D cases. If either A or B is zero, $\langle A, B \rangle = 1$.

Note that we often use the normalized weights for all the neighbors. We define the normalized same-view weight $\bar{w}^+(p, q)$ and different-view weight $\bar{w}^-(p, q)$ as:

$$\bar{w}^+(p, q) = \frac{w^+(p, q)}{\sum_{q \in \mathcal{N}^+(p)} w^+(p, q)}, \bar{w}^-(p, q) = \frac{w^-(p, q)}{\sum_{q \in \mathcal{N}^-(p)} w^-(p, q)}.$$

We also define a “surface” normal $n(p)$ at each strand vertex p , which can be computed by finding the eigenvector with the smallest eigenvalue of the covariance matrix $\sum_{q \in \mathcal{N}^+(p)} w^+(p, q)(q - p)(q - p)^\top$.

We use superscript V to define the projected 2D point of p on one of the visible views $V \in \mathcal{V}(p)$. During the refinement, the position of p in 3D space is restricted along the ray shooting from the optical center of the reference view $R(p)$ to its projected point $p^{R(p)}$ on the reference view. This ensures that the vertex has the same projection on the reference view, and saves computation cost thanks to the reduced degrees-of-freedom.

5.2 Orientation Energy

The orientation energy E_{orient} is designed to make sure that when a 3D strand is projected onto its visible views, the projected orientations are consistent with those indicated by the orientation maps of those views.

Once we apply the diffusion scheme described in Sec. 4 to the orientation map O^V of view V , an orientation vector $O^V(p^V)$ is defined at any point p^V in the hair region, otherwise we set $O^V(p^V) = 0$ (Fig. 7). We then define an orientation energy term $e_{orient}^V(p^V)$ for each segment (p, p_{+1}) on

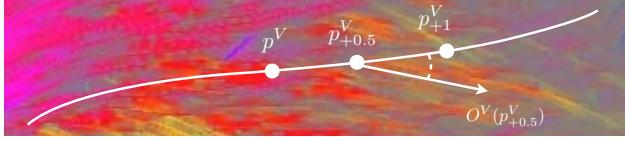


Figure 7: The illustration of orientation energy. A strand is projected on the orientation map in similar color coding as in Fig. 4. The orientation energy term $e_{orient}^V(p^V)$ is determined by the angle between $O^V(p_{+0.5}^V)$ and $p_{+1}^V - p^V$.

S as follows:

$$e_{orient}^V(p^V) = \min \{1 - \langle p_{+1}^V - p^V, O^V(p_{+0.5}^V) \rangle^2, T_{orient}\} \quad (4)$$

where $T_{orient} = 0.5$ is a threshold to make the energy robust to outliers with large projected orientation inconsistency. Note that the square in the definition makes it invariant to $\pm 180^\circ$ directional ambiguity.

Finally, we define the orientation energy E_{orient} as:

$$E_{orient} = \sum_p \sum_{V \in \mathcal{V}(p)} w^V(p) e_{orient}^V(p^V), \quad (5)$$

where $w^V(p) = \max(\langle n(p), v(p) \rangle, 0)$ is the visibility weight of p with respect to view V , and $v(p)$ is the direction from p to the optical center of view V .

5.3 Silhouette Energy

We also enforce the 3D strands to be within and near the visual hull \mathcal{H} using silhouette energy. As illustrated in Fig. 8, given p 's closest point $h(p)$ on \mathcal{H} and $h(p)$'s normal n_h , we can define silhouette energy E_{silh} as:

$$E_{silh} = \frac{1}{D^2} \sum_p \beta \left((p - h(p)) \cdot n_h \right)^2 \quad (6)$$

where \cdot represents inner product, and β is used to discriminate the inside and outside cases for p with respect to \mathcal{H} :

$$\beta = \begin{cases} 1 & (p - h(p)) \cdot n_h \leq 0 \\ w_{out} & (p - h(p)) \cdot n_h > 0 \end{cases}, \quad (7)$$

where w_{out} is a large penalty (10^4) against the case where the vertex is outside the visual hull \mathcal{H} . Note that the diagonal length D of \mathcal{H} 's bounding box is used to make the energy unit-less. Similar approach is applied for unit-less energy formulation in the following sections.

5.4 Smoothness Energy

Smoothness energy is formulated at three different levels to better control the smoothness granularity: the strand level, the wisp level and the global level. The formulation for strand level smoothness E_{strand} stems from the discrete elastic rod model [3] often used in hair simulation that minimizes the squared curvature along hair strands. Further inspired by [4], we take into account the orientation similarity in the bilateral same-view weight w^+ so that the

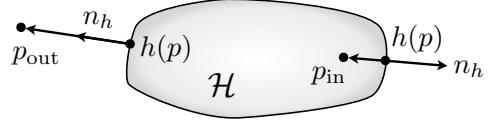


Figure 8: The illustration of silhouette energy. The sign of $(p - h(p)) \cdot n_h$ determines if a point is inside \mathcal{H} and thus the value of β in silhouette energy for p_{in} and p_{out} .

wisp smoothness energy E_{wisp} can better adapt to the local wisp structures and hair's depth discontinuities. Finally, the global smoothness energy E_{global} ensures the global consistency of strand geometry across different views.

Strand smoothness energy Inspired by [3], we define the strand smoothness energy as the summation of squared curvature for each vertex along all the strands:

$$E_{strand} = D^2 \sum_p \text{curv}^2(p) \quad (8)$$

where curvature is computed as:

$$\text{curv}(p) = \frac{2}{l_{+1} + l_{-1}} \left\| \frac{p_{+1} - p}{l_{+1}} - \frac{p - p_{-1}}{l_{-1}} \right\| \quad (9)$$

where $l_{+1} = \|p_{+1} - p\|$ and $l_{-1} = \|p - p_{-1}\|$.

Wisp smoothness energy We use wisp smoothness energy to enforce a strand vertex and its small same-view neighborhood $\mathcal{N}^+(p)$ within the same wisp to lie on a local plane. We use the orientation similarity to estimate the likelihood of being in the same wisp and encode it in the same-view weight w^+ . The wisp smoothness energy is thus defined as:

$$E_{wisp} = \frac{1}{D^2} \sum_p \left(\left(p - \sum_{q \in \mathcal{N}^+(p)} \bar{w}^+(p, q) q \right) \cdot n(p) \right)^2 \quad (10)$$

Global smoothness energy Finally, the global smoothness energy is defined similarly to the wisp smoothness energy to enforce global refinement consistency through local planar resemblance across different views:

$$E_{global} = \frac{1}{D^2} \sum_p \left(\left(p - \sum_{q \in \mathcal{N}^-(p)} \bar{w}^-(p, q) q \right) \cdot n(p) \right)^2 \quad (11)$$

6 Results

We use a camera rig that contains 8 cameras around the subject powered by a single workstation, as shown in Fig. 3. The cameras are Point Grey Flea2 FireWire cameras, operating at 600×800 pixel resolution and 30 frames per second. A few example images are also shown in Fig. 3. We find the current 8-camera setup a good trade-off between reconstruction quality and acquisition complexity for full hairstyle capture. Fewer camera views will push the reconstruction quality towards visual hull due to smaller overlap between views. We also use the hair datasets from [13], but

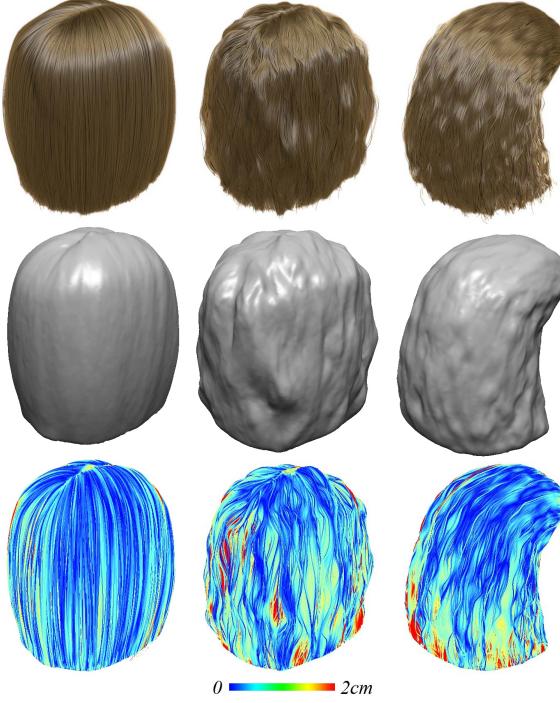


Figure 9: We evaluate the reconstruction accuracy on three synthetic hair styles (straight, wavy and wavythin, first row). We compute the depth maps of our reconstruction (second row) and compare them with the hair’s. The depth map differences are visualized in coded color (third row). The average reconstruction error is around 3 millimeters.

only select 8 images with similar views as our setup from each original dataset for the reconstruction tests.

The same set of energy weights are used for all the results in this work. Note that a relatively small α_{silh} is used to de-emphasize the importance of the visual hull on the reconstruction once the shape is inside the visual hull:

$$\begin{aligned} \alpha_{orient} &= 2 \times 10^{-2} & \alpha_{strand} &= 10^{-4} & \alpha_{global} &= 0.5 \\ \alpha_{silh} &= 3 \times 10^{-5} & \alpha_{wisp} &= 0.5 \end{aligned}$$

The reconstruction results for all the examples are shown in Fig. 11. Note that we use [7] to reconstruct each subject’s facial area and then merge our hair reconstruction using Poisson surface reconstruction [10]. Our method can accurately reconstruct a variety of hair styles from short to long, from smooth to messy and from unconstrained to constrained. Also, our method is able to faithfully reveal interesting hair structures like wisps and curls. In contrast, general visual hull refinement on color texture [5] loses details (Fig. 1). Multi-view stereo methods with weak regularization, such as [7], fail to converge to the correct shape (Fig. 1) due to the challenging wide-baseline setup.

Quantitative evaluation To quantitatively evaluate our method, we use the hair models by [28] and render them using the hair appearance model in [17], as shown in Fig. 9.

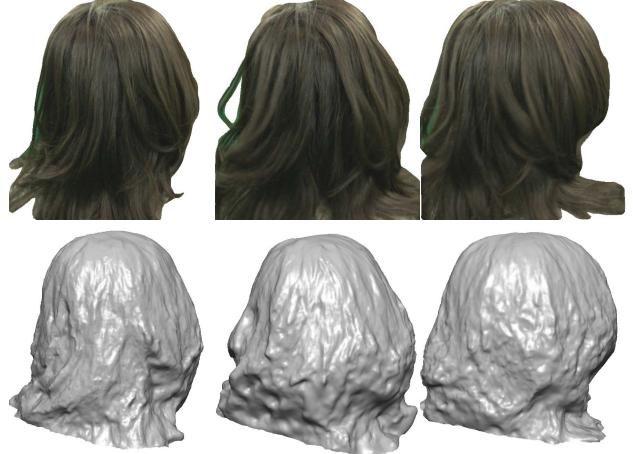


Figure 10: Sample frames (first row) and the reconstructed surfaces (second row) from the dynamic hair capture setup.

The three hair models (straight, wavy, wavythin) in the evaluation are representative for a variety of common hair types. Using the rendered images from viewpoints similar to our real capture setup, we are able to reconstruct the surface for the synthetic hair models. Since hair is volumetric, average closest point distance is not a good error measure. We therefore evaluate the reconstruction accuracy by comparing the depth maps of the hair model and the reconstructed surface on a specific view and visualize the differences in coded color (Fig. 9). The average reconstruction error is around 3mm. Larger errors can be observed in deep concave regions and regions at grazing angles to the cameras.

Dynamic hair capture Compared to previous methods [21, 15], our method is able to capture complete moving hair with only 8 cameras. Three sample reconstructed frames for a hair-shaking performance are shown in Fig. 10 (please see the accompanying video for more results).

Computation time The algorithm is implemented in C++. All the reconstruction tests are performed on a Core i7 2.3 GHz machine with 4GB memory. It takes 10 seconds to compute the orientation map for each 600×800 input image and 1 second to compute the visual hull from 8 segmented images. The strand-based refinement takes 2 minutes.

7 Conclusion and Future Work

We have proposed a novel algorithm to reconstruct complete full-head hair styles with strand-based refinement using only 8 views. Compared to previous methods, our method is able to capture hair accurately with faithful hair structures even with a wide baseline setup. The reconstruction results are evaluated on a set of synthetic hair models and achieve ~ 3 mm reconstruction error on average. The flexible requirement for input allows us to capture complete hair in motion with an inexpensive camera rig.

However, our method does have a few limitations that need to be addressed in the future. The strand-based refinement relies on reasonably long strands to provide good regularization in the optimization. For certain extreme hair styles, like very short hair and fluffy hair, long continuous strands are scarce, which can adversely affect our reconstruction result. Also, because segmentation of hairy objects is still a very challenging problem in computer vision, the visual hull we used to reconstruct the hair is often too smooth, which causes our method to easily miss interesting stray hairs in the reconstruction. For dynamic capture, the motion blur can introduce “artificial strands” along the moving direction that undermines the reconstruction accuracy. The temporal coherence issue also needs to be addressed in the future by imposing temporal constraints.

References

- [1] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. Gross. High-quality single-shot capture of facial geometry. *ACM Trans. Graph.*, 29, 2010. 1
- [2] T. Beeler, B. Bickel, G. Noris, S. Marschner, P. Beardsley, R. W. Sumner, and M. Gross. Coupled 3d reconstruction of sparse facial hair and skin. *ACM Trans. Graph.*, 31:117:1–117:10, August 2012. 2
- [3] M. Bergou, M. Wardetzky, S. Robinson, B. Audoly, and E. Grinspan. Discrete elastic rods. In *SIGGRAPH*, 2008. 5
- [4] M. Chai, L. Wang, Y. Weng, Y. Yu, B. Guo, and K. Zhou. Single-view hair modeling for portrait manipulation. *ACM Trans. Graph.*, 31(4):116:1–116:8, July 2012. 2, 3, 5
- [5] C. H. Esteban and F. Schmitt. Silhouette and stereo fusion for 3d object modeling. *Comput. Vis. Image Underst.*, 96(3):367–392, Dec. 2004. 1, 2, 6
- [6] Y. Furukawa and J. Ponce. Carved visual hulls for image-based modeling. *Int. J. Comput. Vision*, 81(1):53–67, Jan 2009. 2
- [7] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. on PAMI*, 32:1362–1376, 2010. 1, 6
- [8] W. Jakob, J. T. Moon, and S. Marschner. Capturing hair assemblies fiber by fiber. *ACM Trans. Graph.*, 28(5):164:1–164:9, December 2009. 1, 2
- [9] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988. 2
- [10] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proc. SGP*, 2006. 2, 3, 6
- [11] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000. 2
- [12] T. Lay Herrera, A. Zinke, and A. Weber. Lighting hair from the inside: A thermal approach to hair reconstruction. In *ACM Transactions on Graphics*, volume 31, 2012. 1, 2
- [13] M. Lhuillier and L. Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Trans. PAMI*, 27(3):418–433, Mar 2005. 5
- [14] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, 1999. 2
- [15] L. Luo, H. Li, S. Paris, T. Weise, M. Pauly, and S. Rusinkiewicz. Multi-view hair capture using orientation fields. In *Proc. CVPR*, 2012. 1, 3, 6
- [16] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2). 3
- [17] S. Marschner, H. W. Jensen, M. C. and S. Worley, and P. Hanrahan. Light scattering from human hair fibers. *ACM Trans. Graph.*, 22(3):780–791, July 2003. 1, 6
- [18] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761 – 767, 2004. 3
- [19] T. Mihashi, C. Tempelaar-Lietz, and G. Borshukov. Generating realistic human hair for *The Matrix Reloaded*. In *ACM SIGGRAPH Sketches and Applications Program*, 2003. 1
- [20] S. Paris, H. Briceño, and F. Sillion. Capture of hair geometry from multiple images. *ACM Trans. Graph.*, 23(3):712–719, August 2004. 2, 3
- [21] S. Paris, W. Chang, O. I. Kozhushnyan, W. Jarosz, W. Matusik, M. Zwicker, and F. Durand. Hair Photobooth: Geometric and photometric acquisition of real hairstyles. *ACM Trans. Graph.*, 27(3):30:1–30:9, August 2008. 1, 2, 6
- [22] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings of CVPR*, 2006. 2
- [23] E. Tola, V. Lepetit, and P. Fua. A Fast Local Descriptor for Dense Matching. In *Proc. CVPR*, 2008. 2
- [24] T. Tuytelaars and L. V. Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *BMVC*, Sep 2000. 3
- [25] Y. Wei, E. Ofek, L. Quan, and H.-Y. Shum. Modeling hair from multiple views. *ACM Trans. Graph.*, 24(3):816–820, July 2005. 1, 2
- [26] O. Woodford, P. Torr, I. Reid, and A. Fitzgibbon. Global stereo reconstruction under second order smoothness priors. In *Proc. CVPR*, 2008. 1
- [27] T. Yamaguchi, B. Wilburn, and E. Ofek. Video-based modeling of dynamic hair. In *Proc. PSIVT*, 2008. 2
- [28] C. Yuksel, S. Schaefer, and J. Keyser. Hair meshes. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2009)*, 28(5):166:1–166:7, 2009. 6
- [29] Q. Zhang, J. Tong, H. Wang, Z. Pan, and R. Yang. Simulation guided hair dynamics modeling from video. *Computer Graphics Forum*, 31(7pt1). 2



Figure 11: Reconstruction results of all real examples. For each, we show three views with the reference input images.