Accurate, Robust and Structure-Aware Hair Capture

Linjie Luo

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Computer Science
Adviser: Szymon Rusinkiewicz

September 2013

Abstract

Hair is one of human's most distinctive features and one important component in digital human models. However, capturing high quality hair models from real hairstyles remains difficult because of the challenges arising from hair's unique characteristics: the view-dependent specular appearance, the geometric complexity and the high variability of real hairstyles. In this thesis, we address these challenges towards the goal of accurate, robust and structure-aware hair capture.

We first propose an orientation-based matching metric to replace conventional color-based one for multi-view stereo reconstruction of hair. Our key insight is that while color appearance is view-dependent due to hair's specularity, orientation is more robust across views. Orientation similarity also identifies homogeneous hair structures that enable structure-aware aggregation along the structural continuities. Compared to color-based methods, our method minimizes the reconstruction artifacts due to specularity and faithfully recovers detailed hair structures in the reconstruction results.

Next, we introduce a system with more flexible capture setup that requires only 8 camera views to capture complete hairstyles. Our key insight is that strand is a better aggregation unit for robust stereo matching against ambiguities in wide-baseline setups because it models hair's characteristic strand-like structural continuity. The reconstruction is driven by the strand-based refinement that optimizes a set of 3D strands for cross-view orientation consistency and iteratively refines the reconstructed shape from the visual hull. We are able to reconstruct complete hair models for a variety of hairstyles with an accuracy about 3mm evaluated on synthetic datasets.

Finally, we propose a method that reconstructs coherent and plausible wisps aware of the underlying hair structures from a set of input images. The system first discovers locally coherent wisp structures and then uses a novel graph data structure to reason about both the connectivity and directions of the local wisp structures in a global optimization. The wisps are then completed and used to synthesize hair strands which are robust against occlusion and missing data and plausible for animation and simulation. We show reconstruction results for a variety of complex hairstyles including curly, wispy, and messy hair.

Acknowledgements

I would like to thank my advisor Szymon Rusinkiewicz, for his extraordinary patience and guidance from the very beginning of sketching the research ideas to the very end of completing this thesis. His inspiration, diligence and meticulousness are exemplary.

Throughout my graduate study, I have the privilege of working with many collaborators with unique insight and dedicated support. I wish to thank Hao Li, Sylvain Paris, Wojciech Matusik, Ilya Baran, Cha Zhang, Mark Pauly, Thibaut Weise, Michael Kazhdan and many others who inspired and supported my work. Without them, this dissertation could hardly be in its completion.

I must also thank all the faculty and students in Princeton Graphics Group, especially Tom Funkhouser and Adam Finkelstein who encouraged open and engaging discussions.

Finally, I wish to thank my wife Lingbo, for her love and persistent support; my parents, for their continuous love and care; and my friends, for their encouragement and companionship.

This thesis is supported by NSF grant CCF-1012147.

To my parents and my wife Lingbo

Contents

List of Figures

Chapter 1

Introduction

3D acquisition technology has been rapidly improved over past decades. The advancements in acquisition hardware and software, including 3D scanners, time-of-flight sensors, depth sensors, structured light systems and photometric stereo, have made 3D acquisition from the real world increasingly more accurate, efficient, flexible and robust. The current 3D acquisition systems work well for a wide range of real-world objects, from microscopic surface details to city-scale landscapes, from the surface of human face to the anatomical structures of full human body. This wide applicability to real-world objects combined with the improved ability to capture motions make 3D acquisition technology increasingly more popular in gaming, film production, medicine, military and many other areas.

Among all the real-world objects of interest, human hair possesses its unique expressive values. Hair is a vital component of a person's identity, and can provide strong cues about age, background and even personality. In games and films, successfully modeled and animated hair not only contributes to the richness of the virtual character, but also adds a unique language to express the character's experience, thoughts and emotions. Besides the main applications in game and film industries, digital hair models are also useful in cosmetics and advertisements.

This thesis focuses on the acquisition of hair geometry. To be specific, the goal is to reconstruct 3D hair models from real-world hairstyles with improved accuracy, robustness, flexbility and a representation aware of the underlying grouping structures (i.e., hair wisps) to ease hair editing and animation.

In this section, we first briefly review the acquisition methods for hair geometry, both general and specialized, as well as the challenges and limitations to apply these methods on hair (Sec. 1.1).

Then we outline the main contributions of this thesis to address these challenges and limitations (Sec. 1.2). Finally, the outline of the thesis is given in (Sec. 1.3).

1.1 Acquisition of Hair Geometry

1.1.1 General Methods

3D acquisition methods can be classified into two main categories: active and passive. The active methods project some form of energy (e.g., electomagnetic, sonic or mechanic) onto the object and detect the position of the object by measuring and accounting for the change of the energy by the object. The passive methods work directly from the observable features of the object without projecting additional energy.

Among the active methods, the structured light methods [92, 62, 30, 81, 13] promise versatile and efficient solutions to reconstruct a variety of real-world objects. The idea is to project a pattern of spatially and temporally multiplexed light onto the object and the points on the object can then be triangulated by corresponding the input pattern and the observed pattern in another view. However, finding the correct correspondences is difficult for hair because of hair's complex "strand-like" geometry and occlusion. This difficulty can be further emphasized by the multi-scattering light transport between the hair strands, resulting in attenuation and uncertainty of the emitting light observed.

Another type of active approach is photometric stereo [25, 83, 74], which integrates the surface geometry of the object from a set of normal maps estimated from varying directional lighting. These methods assume diffuse appearance on the surface of the object for reliable normal estimation. However, normal direction is not as well defined on hair as on the surface of other objects. Also, the specular appearance of hair violates the diffuse appearance assumption and renders the normal estimation ineffective.

Without probing the object with known energy patterns, all passive methods share two insights to address the ill-posed problem of multi-view stereo reconstruction: a consistency model to check hypothetical points on the object for cross-view consistency; and a smoothness model to regularize the hypothetical points to take place in proximity to one another. Two popular choices in conventional multi-view stereo for the consistency model and the smoothness model are the photo-consistency model and the surface patch model. The photo-consistency model assumes diffuse or view-independent appearance for the object to ease the consistency check across views. The surface

patch model assumes that a local neighborhood on the surface of the object resembles a smooth 2D patch. Hair, however, exhibits the non-diffuse appearance and the complex "strand-like" geometry which make the conventional photo-consistency model and surface patch model unsuitable for the passive methods.

1.1.2 Specialized Methods

To address the challenges arised from general acquisition methods, specialized acqusition methods for hair have been studied over the past decade.

A few active methods have been proposed to improve the acquisition of hair geometry. These methods typically employ special approaches to harness hair's complex geometry and appearance, including estimating 3D hair orientations from the hair's highlight variations under known illumination change [57], scanning hair volume with a single beam of projected light [58] and sweeping hair volume with a shallow depth-of-field [32]. In general, these active hair acquisition systems are capable of producing the most accurate hair reconstruction at the cost of complex acquisition setup and lengthy acquisition session.

For the passive methods, hair orientation consistency plays a key role to leverage the reconstruction quality of conventional multi-view stereo methods [80, 87]. It remains challenging to improve the reconstruction robustness and accuracy as well as to apply to all variety of hairstyles.

One limitation of most existing specialized methods is that the resulting hair model is unstrucutured, i.e., the strands are grown independently according to the computed 3D orientation field without higher level structures. This is in contrast to the hair models used in hair modeling, animation and simulation, which usually involve hierarchical structures such as guide strands and wisps to facilitate controlling, editing and animating. Capturing complex real hairstyles with a structured representation remains a challenging problem.

1.2 Contributions

This thesis addresses several of the main challenges in the acquisition of hair geometry mentioned above, including:

Accuracy. We propose an orientation-based matching metric to replace conventional color-based one for multi-view stereo reconstruction of hair [46]. Our key insight is that while color appearance is view-dependent due to hair's specularity, orientation is more robust across views. Orientation sim-

ilarity also identifies homogeneous hair structures that enable structure-aware aggregation along the structural continuities. Compared to color-based methods, our method minimizes the reconstruction artifacts due to specularity and faithfully recovers detailed hair structures in the reconstruction results.

Flexibility and robustness. We introduce a system with more flexible hair capture setup that requires only 8 camera views to capture complete hairstyles [48]. Our key insight is that strand is a better aggregation unit for robust stereo matching against ambiguities in wide-baseline setups because it models hair's characteristic strand-like structural continuity. The reconstruction is driven by the strand-based refinement that optimizes a set of 3D strands for cross-view orientation consistency and iteratively refines the reonstructed shape from the visual hull. We are able to reconstruct complete hair models for a variety of hairstyles with an accuracy about 3mm evaluated on synthetic datasets.

Plausible structures. We propose a method that reconstructs coherent and plausible wisps aware of the underlying hair structures from a set of input images [47]. The system first discovers locally coherent wisp structures in the reconstructed point cloud and the 3D orientation field, and then uses a novel graph data structure to reason about both the connectivity and directions of the local wisp structures in a global optimization. The wisps are then completed and used to synthesize hair strands which are robust against occlusion and missing data and plausible for animation and simulation. We show reconstruction results for a variety of complex hairstyles including curly, wispy, and messy hair.

1.3 Thesis Outline

The remainder of this thesis is organized as follows: Chapter 3 describes the orientation-based matching metric in a multi-view stereo system for accurate hair reconstruction. Chapter 4 presents a flexible hair capture system robust against wide-baseline setup using strand-based refinement. Chapter 5 describes the structure-aware hair capture system that reconstructs plausible wisp-based hair models from input images. Finally, Chapter 6 presents conclusions on the work in this thesis and suggests possible ideas for future work.

Chapter 2

Background and Related Work

2.1 3D Aquisition Systems

3D acquisition systems can be classified into three main categories by how the systems interact with the acquired objects. One category of systems measure real-world objects through physical *contact*, e.g. mechanical touch probes, accelerometers and various markers. Another category of systems use *transmissive* means such as X-rays (e.g. computed tomography), ultrasound and nuclear magnetic resonance (e.g. magnetic resonance imaging). These systems are capable of reconstructing internal structures and are widely used for industrial and medical applications with relative costly equipments. The last category of acquisition systems make use of *reflective* signals such as sound (sonar) and light to reconstruct the outer surfaces of the objects. In particular, we are interested in the optical systems using reflective light signal. These optical systems gain increasing popularity for improved flexibility, efficiency and accuracy. We can further classify the optical systems into *active* and *passive* systems depending on whether the system actively project light onto the acquired objects.

2.1.1 Active systems

One straightforward way to obtain the distance or depth to the object is measuring the round trip travel time of a reflective signal (e.g. light or sound) between the sensor and the object. The shape of the object can be obtained by measuring the distances to a set of sample points on the object surface. Such systems are called time-of-flight systems and two exemplary applications of this approach are radar and sonar. The advantages of time-of-flight systems are the large working

scale and the distance-insensitive measurement accuracy. However, the reconstruction accuracy is limited to several millimeters by the timing resolution.

Another approach to distance measurement is triangulation. Systems using this approach project light onto the object and measure the reflection by a camera observing at a different angle. The position of the reflection point on the object is then triangulated by the paths of projected light and the reflected light. To improve the reconstruction efficiency, multiple light rays are projected and measured at different points on the object at the same time. Special light patterns are needed to distinguish different projected rays for proper triangulation. One approach uses *structured light* patterns multiplexed in both space and time with color [92], binary gray codes [62], sinusoidal patterns [30, 81] and intensity ratios [13] for improved acquisition efficiency and robustness. Another approach uses random noise patterns, a.k.a. *unstructured light*, to facilitate robust correspondence matching between the projected and received light. Since this approach essentially performs stereo matching on the random noise pattern, it is also called *active stereo* due to its active nature. Active stereo method has become very popular since the introduction of the Kinect sensor [53].

2.1.2 Passive systems

Passive systems infer the shape of an object from the appearance without actively controlling the lighting on the object. There are a variety of clues from an object's appearance that allow inference of the object's shape which lead to a category of *shape from X* methods. These clues include shading [29], specularities [1], shadows [65], texture [24], contours [36], motion [72], focus and defocus [59, 38].

Shape from shading is of particular interest since it provides strong clues for accurate shape reconstruction [94]. Shape from shading typically assumes diffusive and view independent surface reflectance (Lambertian surface) under single light source. The algorithm jointly estimates the albedo or texture of the surface and the lighting based on global shading statistics. The surface normals are then reconstructed to best explain the shading followed by an integration to recover the shape of the surface. More accurate recontruction of surface normal and geometry is possible with *photometric stereo* methods if the shading is provided under known or unknown varying lighting conditions [25, 83]. One recent photometric stereo system using time-multiplexed directional lighting to capture full-body human performance is introduced in [74], which used numerous LED lights uniformly distributed on a dome to provide directional lighting.

Another popular passive approach is reconstructing the object from the images taken from different viewpoints. This approach is called *multi-view stereo* which is the subject of next section.

2.2 Multi-View Stereo

Multi-view stereo methods have been rapidly improved and widely used over the past decade thanks to the ease to acquire photos with more and more affordable digital cameras. Multi-view stereo reconstructs the visible surface of an object through finding the correspondences between the input images that determine the 3D points on the surface. To find the correspondences, a robust *matching metric* is needed to handle view-dependent appearance, illumination change and camera gain and bias. A common matching metric is *photo-consistency*, which measures the color consistency at the corresponding points on different photos. Also, the stereo problem is an ill-posed problem, which means there are many possible 3D surfaces to explain the input images. For example, one trivial solution to all stereo problems would be a set of different postcards placed in front of each camera position. In order to find reasonable solutions, a kind of *shape prior* is assumed to resolve the ambiguities during the matching process. The shape prior can be modeled explicitly (e.g. as the aggregation model) or implicitly (e.g., as the bias of the optimization process), which accounts for the bias of the stereo method. For the rest of the section, we will first review several general approaches to the multi-view stereo problem in the existing literature and then discuss the issues in matching metric and shape prior in more detail. Note that more thorough surveys on multi-view stereo can be found in [68, 67].

2.2.1 General approach

There are several general approaches to developing the reconstructed surface from the input images.

The first approach extracts a surface from a 3D volume where each voxel encodes the score of how consistent the projections of the voxel are on the input images. Higher consistency indicates higher confidence of an existent 3D point at the voxel. To extract the reconstructed surface from the 3D volume, an optimization is typically involved to maximize the consistency scores while maintaining the smoothness and continuity of the surface. Popular choices for optimization include Markov Random Field with max-flow [75] or multi-way graphcuts [37]. This approach yields globally optimal and consistent reconstructed surface but usually requires more memory and computation time for the optimization.

Another approach is to evolve or refine an initial surface to its final form by iteratively improving the consistency at the projections of each surface point among the input images. Space carving [39] and its variants implicitly evolve the reconstructed surface by removing the voxels from an initial volume. Other implicit representations can be used, such as level-set methods, to shrink or expand

an inital surface [21]. There are also methods that refine an explicit mesh from the initial visual hull [22, 20]. This type of approaches strongly rely on the initial shape for the reconstruction and thus typically exhibit certain biases in the final result. However, a strong prior can be helpful when the reconstruction is significantly ill-posed, such as a wide-baseline setup where cameras are posed far away from each other and the common regions for correspondence computation are much reduced.

Some other approach works by reconstructing separate depth maps from a few viewpoints and merging them in 3D space. Cross-view consistency is an issue for such systems. The consistency is either explicitly enforced as constraints in the reconstruction [12] or realized by volumetric merging methods such as [34, 18] as a post process. Each depth map is reconstructed from a local group of adjacent views (typically two) with small baseline which minimizes occlusions and maximizes overlapping region for correspondence matching. This type of systems prove to achieve high quality reconstruction results in terms of accuracy and robustness [2, 12].

The last approach finds a sparse set of feature points on the surface and then populate a denser set of points from these sparse feature points. The ideas are that the sparse salient features can usually be found robustly across views and that the rest of the points can be found following the continuity of the underlying surface. A exemplary method is Patch-based Multi-View Stereo (PMVS) [23]. PMVS first identifies corner-like and blob-like feature points and correspond them across the input images for the intial sparse points. Then an expansion scheme is performed iteratively to find the points adjacent to the exiting points. The result is a set of points on the object and the final surface can be obtained by applying the Poisson Surface Reconstruction [34]. This approach has great adaptivity to reconstruct arbitrarily complex surfaces at the cost of weak surface assumptions to regularize the reconstruction for less noisy results.

2.2.2 Matching metric

Matching metric lies at the core of the correspondence matching process of all multi-view stereo methods. A matching metric compares the pixel values from different views and aggregate the comparison results over local areas (typically square windows) to improve the matching robustness since certain shape continuity can always be assumed for the reconstructed object.

The most common metrics include the *squared intensity differences* and *absolute intensity differences* which directly measure the differences of pixel values. However, squared metrics are well-known for their intolerance for outliers or occlusions, a few more robust metrics have been proposed including truncated quadratics and contaminated Gaussians [7, 6] to limit the influence of outliers.

Another important goal to design a matching metric is its robustness against illumination change and camera gain and bias. Histogram equalization [17] is useful to neutralize the camera gain and bias as a preprocessing step. Some metrics are found insensitive to camera gain and bias such as gradient based metrics [66] and non-parametric measures like rank and census transforms [91]. A more popular approach is to normalize the pixel values within the matching windows such as normalized cross-correlation (NCC).

The change of viewpoints can dramatically change the appearance of the object and lead to serious occlusion problems. This is especially challenging for wide-baseline systems, where the cameras positions differ significantly from each other. A few feature descriptors are robust against the change of viewpoints and illuminations such as SIFT [45]. To extend the idea of SIFT for dense stereo matching, Tola et al. [71] proposed a matching metric *DAISY* for wide-baseline systems. View-invariant local regions have also been studied for reliable matching in wide-baseline systems such as affinely invariant regions [73] and maximally stable extremal regions [51].

Most multi-view stereo methods conveniently assume that the object surface is *lambertian*, meaning view-dependent reflectance and thus simpler matching metric between different views. However, if the specularity of the object's appearance is significant, the lambertian assumption no longer holds. One matching metric that accounts for specularity [88] is inspired by the fact that the reflected colors of each point on the surface under different lighting lie colinearly in RGB color space assuming Phong reflectance. This statistical property enables a maximum likelihood solution to estimate and match the surface albedo across views.

To aggregate the per-pixel matching results, a typical approach is to use square window of pre-defined size on the matched images. This scheme has the bias towards fronto-parallel surfaces [67] since it implicitly hypothesizes a fronto-parallel square surface patch to match across views. One solution to alleviate this problem is to use scaled window matching to better account for slanted surfaces [12]. Adaptive window size can also be estimated based on statistical model for the disparity variation within the window [56]. Locally adapted weighting can be applied within the window resemble the perception of human visual system [89]. Using edge-preserving filters such as bilateral filtering in aggregation allows reconstruction with sharp depth discontinuities [61]. Some other methods project each small patch of the hypothesized 3D surface onto the input images and aggregate the matching scores between the pixel values of the projected patches [23].

2.2.3 Shape prior

As suggested by the previous section, aggregations can imply underlying shape assumptions which result in biases in the reconstruction results, such as fronto-parallel bias. Other biases can be introduced by the reconstruction algorithm itself. Space carving [65] and its variants produce the largest photo-consistent scene reconstruction known as the "photo hull" while the level-set based methods typically converge to the minimal photo-consistent surface.

The assumption made about the underlying shape, implicitly or explicitly, is the shape prior that helps disambiguate the stereo matching. Explicit shape prior can be used as the reconstructed model such as the piecewise planar patch model [28], in which the input reference image is over-segmented into super-pixels based on color and each segment is reconstructed as a planar patch in the final model. Planar prior is simple but not the optimal model in terms of human perception. Ishikawa and Geiger [31] argued that second order priors that encourages the smoothness of second derivative on the surface is closer to human visual system. This idea proved to achieve better results than planar or first order priors [9, 82].

In general, shape priors are more important for plausible reconstruction if the input is insufficient and the problem is rather ill-posed. In contrast, with more views provided to the stereo reconstruction system, the reconstruction is more constrained and thus shape priors are less important.

In the next section, we will investigate into the reconstruction methods using stronger and more specific shape priors.

2.3 Model Fitting

The output from most of the 3D acquisition systems is usually noisy and incomplete due to reconstruction error and occlusion. Specific knowledge about the underlying model can help to *consolidate* the noisy and incomplete 3D data into complete, consistent and concise representations by model fitting.

Assuming that the reconstructed surface is watertight and smooth, deformable models can be used to consolidate the input data. In an early seminal work [33], Kass et al. proposed the *snake* model for robust image segmentation by minimizing a combination of data fitting and smoothness energies. This has been extended to model 3D objects by fitting a deformable mesh inside the matching volume during multi-view stereo reconstruction [20]. In a similar fashion, Duan and Qin

[19] introduced intelligent balloon to fit the input point cloud with an expanding deformable model from inside to yield watertight geometry of arbitrary topology.

Architecture exhibits regularity, symmetry and hierarchy which can be well formulated by generative models. "SmartBoxes" is introduced by [55] to interatively reconstruct urban architectures with regular box-like structures. The symmetric recurrences of planar elements in architectural structures can also be detected and consolidated using model fitting [98]. Using constructive solid geometry (CSG) with volumetric primitives, Xiao and Furukawa demonstrated a system [85] to reconstruct the indoor scenes in the museums from 3D point clouds. Wu et al. [84] introduced a system to infer the schemetic representation of swept surfaces for an architecture.

More specific models can be applied to fit correponding input data. Blanz et al. [8] proposed a morphable face model based on a database of captured face models to fit any input image of face. 3D Scan data of trees can also be fit with a tree skeletal model for consolidation [44]. Using a generalized cylinder model, "Arterial Snakes" is proposed to reconstruct delicate interleaving man-made structures [42].

2.4 Hair Capture

Human hair typically consists of hundreds of thousands of hair strands. The hair strands grow from the follicles on the scalp and form smooth, wavy or curly shapes depending on various physical properties and the growing dynamics of the hair. Hair can be categorized into three main classes based on ethnic groups: Asian hair, African hair and Caucasian hair. Asian hair strand has a circular cross-section and appears smooth, whereas African hair strand has elliptical cross-section and looks irregular. Caucasian hair ranges from smooth to highly curly hair with varying regularity in cross-sections. The high variability of color, curliness and physical properties among the hair of different ethnic groups of people, combined with countless processes to stylize hair through cutting, shearing, perming, combing and dyeing, creates the amazingly wide spectrum of hairstyles in the real world that contribute to the identification of each individual human being.

Hair capture concerns about the reconstruction of hair geometry from the input hairstyles. Besides the high variability of real-world hairstyles, there are a few other challenges that make hair capture a difficult problem. The first is the sheer number of hair strands that need to be considered in the reconstruction of a complete hairstyle, typically in hundreds of thousands. Also, hair's strand-like geometry is in contrast to the surface-like geometry of most conventional objects that allows convenient patch-based continuity assumption in the reconstruction. Finally, hair's non-

diffuse appearance is challenging to many reconstruction methods that assume view independent appearance.

Many existing hair capture methods strive to recontruct individual hair strands based on the orientational and structural continuity along the strands. This type of methods usually work for hair with limited density in a limited working volume and require complex capture setups or lengthy capture sessions.

On the other hand, capturing individual hair strands is usually not necessary for many applications and infeasible in practice due to limited acquisition resolution and occlusion. As a result, many hair capture methods seek to reconstruct a hair volume in which hair strands are grown or synthesized to match the input hairstyles. This type of methods are more scalable to capture the complete hairstyle and more flexible on the acquisition setups.

In this section, we first elaborate the concepts and common techniques related to hair orientation as often used in the literature of hair capture. We then investigate into both types of hair capture methods that reconstruct hair geometry on the hair strand and hair volume levels.

2.4.1 Hair orientation

Hair orientation is an important concept for hair capture because it provides a distinctive feature for hair reconstruction and indicates the direction for hair's structural continuity. Hair orientations are usually first computed from the input hair images as 2D orientation maps and then extended to a 3D orientation field based on the reconstructed 3D hair geometry.

2D orientation map. A popular image-based method to compute 2D orientation map is proposed in [57] and has been since adopted by many following methods. The idea is to convolve the image with a bank of rotated anisotropic filters and the orientation is selected as the one with the maximum response. To be specific, K_θ is a filter generated by rotating an x-aligned anisotropic kernel K by θ. The 2D orientations $\ell(x, y)$ at each pixel (x, y) for the input hair image $I(x, y)$ is then computed as follows:

$$\ell(x, y) = \arg\max_{\theta} |K_\theta * I|(x, y). \tag{2.1}$$

An example of input image and the computed 2D orientation map is shown in Figure 2.1.

3D orientation field. If the 3D hair geometry is known, e.g. a point cloud, the 3D orientation ℓ_p at each point p can be computed by intersecting two projected planes π_1 and π_2 passing the 2D orientations ℓ_1 and ℓ_2 at the respective projected points as shown in Figure 2.1.

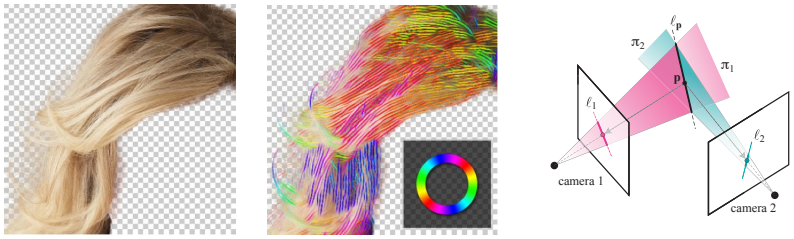


Figure 2.1: A hair image (left) is used to compute the 2D orientation map (middle). A 3D orientation can be determined by two 2D orientations (right). Figure reproduced from [14, 58].

A more robust method is to take into account all the visible views and compute the most probable 3D orientation based on the projected 2D orientations. To be specific, let n_i be the normal of the plane passing the camera center and the 2D orientation at the projected point on i-th visible view. Then the 3D orientation ℓ_p can be computed as:

$$\ell_p = \arg\max_{\ell} \sum_i (n_i \cdot \ell)^2, \quad \text{subject to } \|\ell\| = 1, \tag{2.2}$$

In most 3D acqusition system, occlusion causes incomplete reconstruction and missing geometry. As a result, only the 3D orientations on the outer visible hair can be estimated from the images and extrapolation is needed to handle the rest of the invisible hair volume. One popular method as proposed in [58] is to extrapolate the 3D orientations from the 3D orientations at the visible points by isotropic diffusion. To be specific, the orientations in the holes is the solution to the heat equation given the orientations at the visible points as the boundary condition:

$$\frac{\partial O}{\partial t} = \frac{\partial^2 O}{\partial x^2} + \frac{\partial^2 O}{\partial y^2} + \frac{\partial^2 O}{\partial z^2}, \tag{2.3}$$

where O is the structure tensor of the 3D orientation as will be detailed shortly. An simple iterative scheme can be applied to solve this equation by updating the orientation at each point in the hole by its laplacian. The structure tensor is useful to average or interpolate orientations with the $\pm\pi$ directional ambiguity. Formally, we define orientation ℓ as a unit vector and the interpolation between orientations $\ell_1, \cdots, \ell_n$ with weights $w_1, \cdots, w_n$ can be posed as a maximization problem:

$$\bar{\ell} = \arg\max_{\ell} \sum_i w_i (\ell_i^\top \ell)^2 = \arg\max_{\ell} \ell^\top (\sum_i w_i \ell_i \ell_i^\top) \ell, \quad \text{subject to } \|\ell\| = 1, \tag{2.4}$$

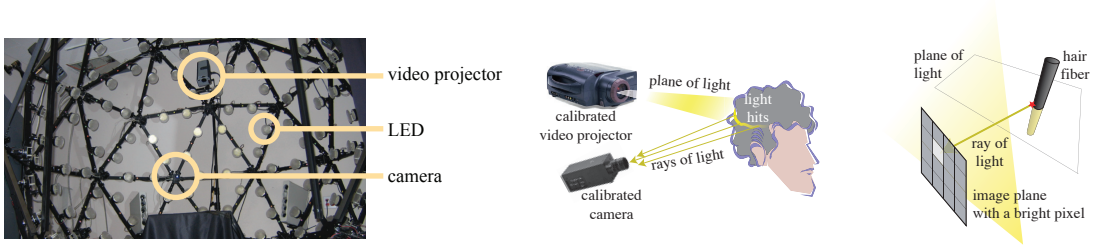


Figure 2.2: Scanning hair strands by Hair Photobooth [58]. The hair strands are triangulated by intersecting the projected plane of light and the rays of light from the camera. Figure reproduced from [58].

where we can define $O = \sum_i w_i \ell_i \ell_i^\top$ as the structure tensor, and the interpolating orientation $\bar{\ell}$ is the maximizer for $\ell^\top O \ell$. Note that the $\pm\pi$ directional ambiguity is eliminated by the squaring in Eq. 2.4.

2.4.2 Capturing hair strands

Structured light systems have been successfully applied to efficiently capture a variety of objects by corresponding and triangulating the time and space coded patterns on the object surface. This requires the reconstructed surface to be continuous and well-defined. However, hair geometry consists of a large set of thin and scattered strands and as a result the projected patterns on different strands can correspond to the same pixel on the triangulating camera, leading to ambiguity and failure of the triangulation.

Other active stereo methods, such as sweeping a plane of light, can work for hair without causing the correspondence ambiguity at the cost of lengthier capture process. One such system is introduced in [58] (Figure 2.2), which uses 3 projectors, 16 cameras and multiple LED lights mounted on a geodesic dome to capture a complete hairstyle on the strand level. The projectors and the cameras are fully calibrated and the positions of the hair strands can be uniquely triangulated by intersecting the ray of light from each bright pixel and the plane of light projected from the projectors. The result is a point cloud of the positions of the strands. The point cloud is then augmented with a volumetric 3D orientation field in order to generate the final hair strands. Finally, the algorithm synthesizes hair strands to match the input hairstyle with a straightforward growing scheme. To be exact, hair strands are originated from the sample root points on the scalp and then iteratively extend to the next point according to the 3D orientation ℓ at the current point by a certain step. The iteration stops if the predefined number of steps is reached followed by a check to retain the segment from the root point to the last visible point or to remove the entire strand if no visible

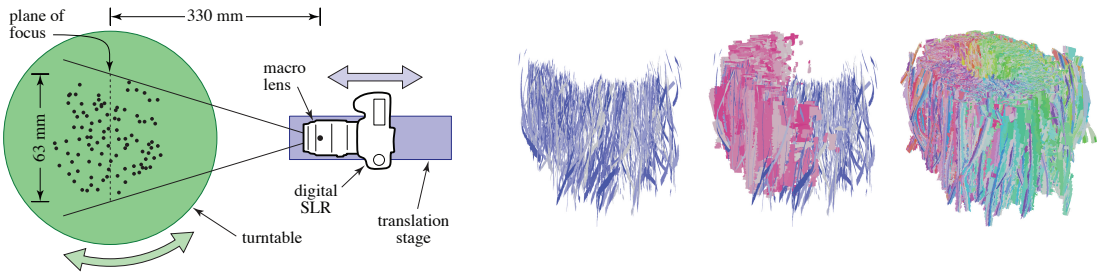


Figure 2.3: Scanning hair strands by sweeping the shallow depth-of-field of a macro lens [32]. In-focus strands form ribbons and intersect themselves from different views to reveal the paths of 3D strands. Figure reproduced from [32].

point is encountered. The final result is a set of hair strands grown from the scalp to match the 3D orientation field and the reconstructed point cloud.

Jakob et al. [32] proposed a passive system to scan the hair strand geometry by sweeping the shallow depth-of-field of a macro lens through the hair volume (Figure 2.3). More specificallly, a camera with a macro lens is mounted on a translation stage and takes photos of a hair assembly on a range of depths. The hair assembly is rotated around an axis vertical to the stage allowing the photos to be captured in different view angles (24 in this work). The method first detect in-focus hair strands at each depth plane by tracing the ridges using a scheme similar to Canny edge detector, the resulting ridges then span to form 3D ribbons within continuous range of in-focus depths. The ribbons from different views intersect each other and identify the paths of individual hair strands. The proposed system proves to be accurate enough to capture individual hair strands in a loose hair assembly. However, for hairstyle with high amount of strands packed densely, the system fails to reproduce the hair structure because individual hair strands are hard to be tracked and resolved from each other.

Another passive approach to capture facial hair geometry is introduced by Beeler et al. [3] (Figure 2.4). The proposed system uses a similar setup to [2] that employs 8 DSLR cameras to capture high resolution images revealing individual facial hair fibers. Because the system is single-shot, the acquisition process is very efficient since it only requires the time of one synthronized shot from all the cameras. The key idea is to perform stereo matching on the hair strands. The algorithm begins with a usual image-based hair strand detection on the reference image. The detected strands are then projected to visible camera views to search for matched strands along the epipolar lines. This leads to the search for a curve with the highest score in a matching matrix in which each entry is a matching score for each point on the detected strand with each depth in the search range. Note that the facial capture system in [2] is capable of reconstructing an approximate surface to the facial

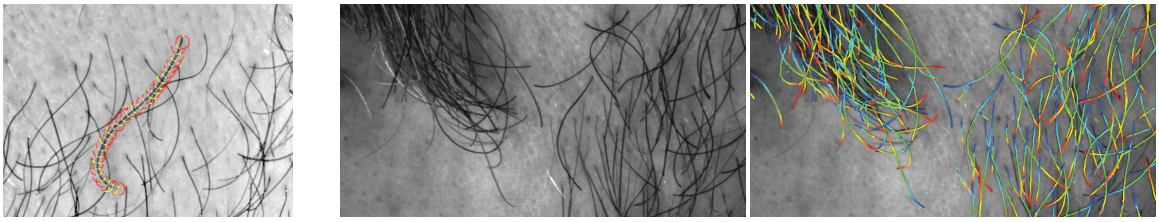


Figure 2.4: Capturing facial hair by matching detected 2D strands on adjacent views [3]. Figure reproduced from [3].

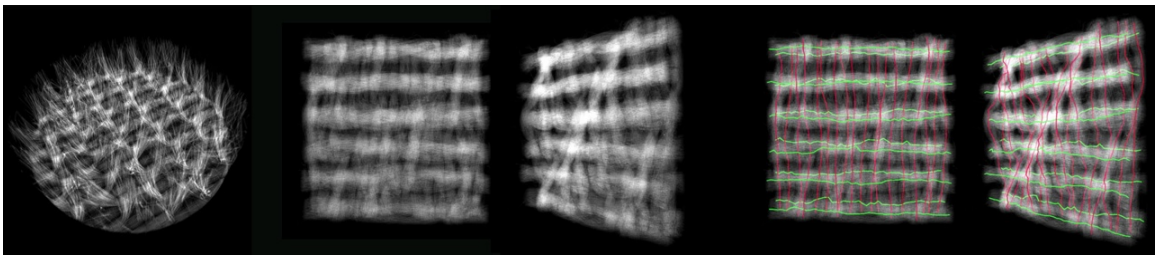


Figure 2.5: The density fields of fabric samples reconstructed by CT scan and the extracted weaving yarn structures [96, 97].

hair region, the search can be effectively limited to a depth range near this reconstructed surface. A refinement step is then performed to connect close segments, merge overlapping segments and remove duplicate segments. Although the system can reconstruct sparse facial hair accurately, it is difficult to scale the method to capture dense and full-head hairstyles limited by camera resolution and increased ambiguity in strand matching.

One promising approach to volumetric reconstruction of hair strand geometry is computer tomography (CT) scan (Figure 2.5). CT scan has been recently applied to capturing fabric samples for yarn-level volumetric apperance models in [96, 97]. To reconstruct the yarn-level structures, an 3D orientation field is first computed using a filtering scheme similar to [57]. The yarn structures are then tracked from the endpoints on the boundary following both the orientation field and the center of mass in the volumetric density field output from the CT scan. The endpoints can be automatically detected using a k-means clustering step. Although micro CT scanners are recently increasing in availability, the scanning volume is limited to small material samples. To scan a full-head hairstyle, full scale CT scan is required at the cost of increasing acquisition budget and time.

2.4.3 Capturing hair volume

The previous section describes various existing techniques to obtain strand-level hair geometry. For many practical applications, however, reconstructing hair geometry at strand-level is an over-kill.

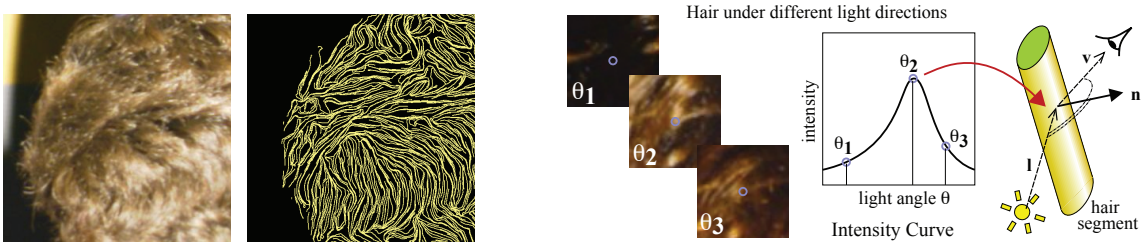


Figure 2.6: Estimating 3D hair orientations under varying lighting [57]. 2D orientations are first computed from the input image followed by a normal estimation on the strand's reflectance profile under varying lighting to determine the strand's 3D orientation. Figure reproduced from [57].

These applications may include gaming, teleconferencing and augmented reality that prioritize more on performance than model accuracy.

Many hair capture methods thus seek to reconstruct the hair volume that reflects the overall hair structures at a coarser level. The hair strands are synthesized inside the reconstructed hair volume following the orientation field to match the input hairstyle.

Paris et al. [57] proposed a method to estimate a 3D orientation field from the hair highlights under a moving light source with known trajectory (Figure 2.6). The orientation field is then used to grow the hair strands from a fit scalp inside the visual hull. The algorithm first compute the 2D hair orientation maps using the rotated filters. The key insight to extend these 2D orientation maps to 3D is to estimate the normal vectors of the hair strands by detecting the specular peak under the illumination of a moving light source with known trajectory. The specular peak of a hair strand occurs in the standard reflection direction with a slight bias toward the root based on the hair scattering model [50]. Constrained by the 2D orientation, a known normal vector then determines the 3D orientation of a hair strand. Note that to avoid the degenerate case where the light motion is perpendicular to the hair strand, two orthogonal light movements are performed to ensure that at least one light movement is usable.

Using more images can help to compute 3D orientation field and hair geometry without special illumination. Wei et al. [80] introduced a method to capture hairstyles using about 30 images under general lighting. The method employs a similar approach to [57] to grow hair strands inside the visual hull to develop the captured hair volume. However, their key insight is to use the cross-view *orientation consistency* to constrain the growing in order to improve the reconstruction accuracy. To be specific, if a 3D hair strand segment is on the captured hairstyle, its projected 2D strand

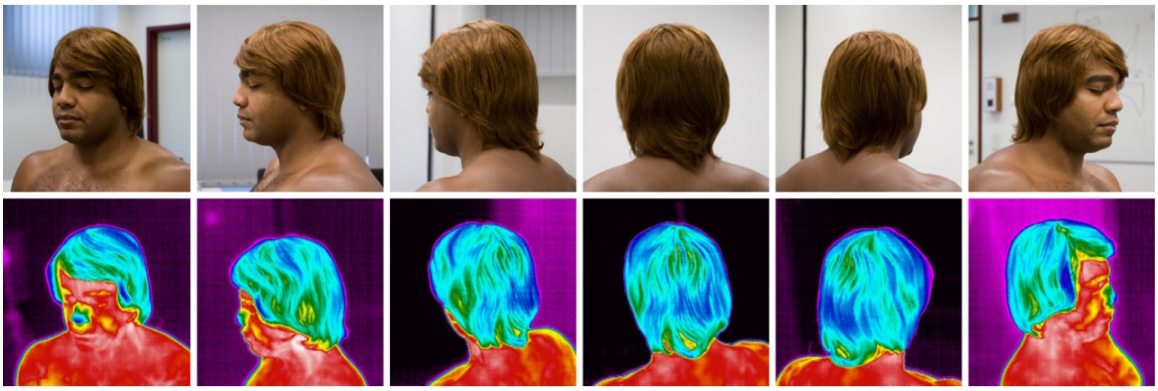


Figure 2.7: Thermal imaging reveals the relationship between the surface temperature and the distance to the scalp from the outer hair. Figure reproduced from [27].

segments on three views i, j, k should satisfy the following constraint:

$$\mathbf{A}_i^\top \mathbf{a}_i \cdot (\mathbf{A}_j^\top \mathbf{a}_j \times \mathbf{A}_k^\top \mathbf{a}_k) = 0. \tag{2.5}$$

where $\mathbf{A}_\star^\top$ is the 3×3 submatrix of the projection matrix $\mathbf{P}_\star = (\mathbf{A}_\star | \mathbf{t}_\star)$ and the projected segment lies on a line $\mathbf{a}_\star \mathbf{x} = 0$ on correponding view. This equation can be seen as the coplanarity of the normals $\mathbf{n}_\star$ of the projection plane passing the camera center and the 2D hair strand segment and $\mathbf{n}_\star = \mathbf{A}_\star^\top \mathbf{a}_\star$.

Beyond measurement using visible light, thermal imaging is also applied to reconstructing the hair volume in light of the infrared radiation from the human head [27] (Figure 2.7). The input to the system is a video taken around the hairstyle using a thermal camera. After the images are registered, the visual hull is reconstructed and refined using multi-view stereo on a coarse level. The method then takes the advantage of the relationship between the temperature and the distance to the scalp to further refine the hair volume on a fine level. This refinement is 10 times more efficient than full-resolution stereo reconstruction. In the end, the algorithm produces a hair model by growing the hair strands from the scalp following the hair boundary and the orientation field. The limitations of the method include that the thermal camera generally has lower resolution than normal camera and that the temperature feature becomes less distinctive for long and voluminous hairstyles.

Chai et al. [14] demonstrated a system to reconstruct a 3D hair model from a single portrait image using prior knowledge of generic human head model and strand-level smoothness. The system first performs user-assisted hair segmentation and fits a generic head model to the input image. 2D hair strands are then extracted and solved for their optimal depths in 3D space given the positional

constraint to the fit head model and the strand-level smoothness between the strands. The result is a 3D hair strand model that enables various applications for image manipulation such as hairstyle editing, transfer and novel view synthesis.

Chapter 3

Multi-View Hair Capture Using Orientation Fields

Orientation is a key feature for human hair. In this chapter, we investigate into the idea of using orientation as the matching metric in multi-view stereo for hair capture. Our key insight is that while color appearance is view-dependent due to hair's specularity, orientation is more robust across views. Orientation similarity also identifies homogeneous hair structures that enable structure-aware aggregation along the structural continuities, yielding hair reconstruction with detailed structures.

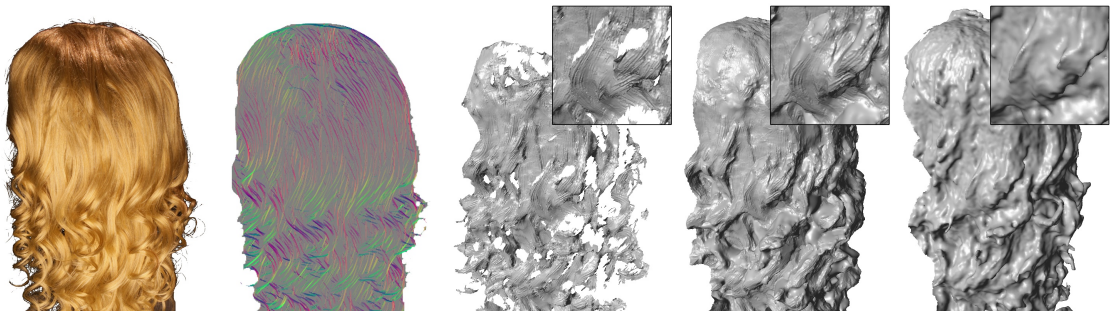


Figure 3.1: We begin with many high-resolution photographs (with unconstrained lighting), compute an orientation field for each, and perform multi-view stereo matching using a metric based on orientation similarity. The resulting depth maps show high-resolution details of hair strands and we integrate them into a single merged model. In contrast, conventional multi-view stereo algorithms and merging techniques [23, 34] fail at capturing the fine hair structures.

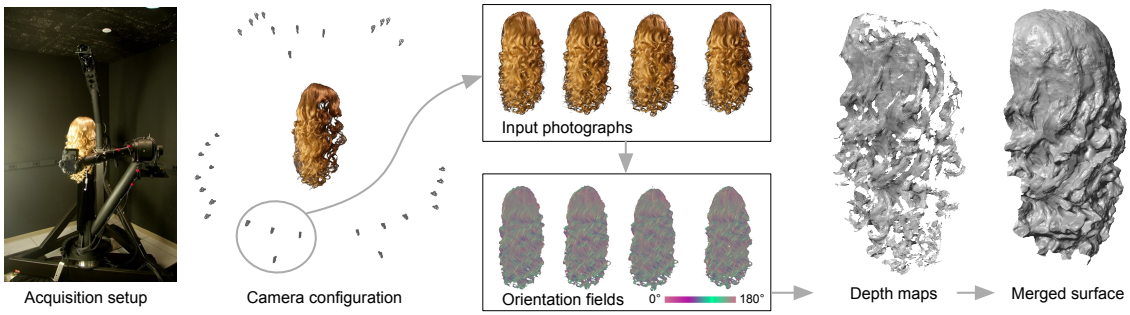


Figure 3.2: Our reconstruction pipeline. We use a robotic gantry to capture static images from different views. Each 4 views are grouped into a cluster to reconstruct a single accurate depth map based on orientation fields computed from input photographs. (Section 3.2 and 3.3) Finally, the depth maps are merged into a single hair surface (Section 3.4).

3.1 Introduction

Despite its aesthetic importance in defining a person's look, the reconstruction of realistic hair is often neglected in methods for 3D acquisition. Hair is one of the most challenging objects to capture using standard computer vision techniques. Occlusions are omnipresent, and hair's strand geometry precludes general surface-based smoothness priors for stereo (e.g., [82]). Besides, the highly specular nature of hair fibers [50] is not well modeled by standard appearance models. Even the latest facial reconstruction techniques (e.g., [2]) exclude hair from the region of interest, and so hair modeling largely remains a manual task in practice [54]. Dedicated acquisition methods have been proposed [58, 32], but they rely on scanning rigs that are costly and difficult to build. Furthermore, these methods require lengthy capture sessions that limit their suitability to treat hair in motion. While Hair Photobooth is perhaps the best example of this approach, we argue that in fact all active illumination methods will be challenging to apply to hair capture, especially in the dynamic setting. Single-frame methods such as noise patterns are ineffective because of the complex geometry and occlusion of hair, while multi-frame structured light methods such as Gray codes fail for moving hair.

To address these difficulties we investigate a single-shot passive multi-view stereo approach that requires only consumer-level hardware and produces results on par with or superior to existing techniques.

Naively applying existing stereo techniques fails because hair is specular, and hence, observed color varies quickly with changes in viewpoint. Our key insight is that hair orientation (which can be computed using a filter bank of many — e.g. 180 — oriented Difference-of-Gaussians (DoG) kernels) is a stable feature across nearby viewpoints and can be used as a reliable matching criterion. We

leverage this idea by defining a stereo matching metric based on *similarity of local orientation* that is insensitive to local changes in brightness. We further improve matching robustness by aggregating local evidence with a scheme that accounts for the local hair structure; i.e., we aggregate matching costs *along* strands but not perpendicular to them. We incorporate the resulting matching score into a coarse-to-fine multi-view stereo framework based on Markov Random Fields (MRF). We operate on small sets of nearby viewpoints at a time to minimize the effect of inter-view differences in foreshortening, then merge all the resulting depth maps into a globally consistent detail-preserving model of the whole head of hair.

We demonstrate that our approach can handle a wide variety of hairstyles, that strands can be grown within the reconstructed hair volume which is suitable for rendering, and that our approach is capable of capturing hair in motion when using video cameras. We quantitatively evaluate the precision of our approach using synthetic data.

3.1.1 Related Work

Multi-view stereo has received significant attention [68] but applying a generic algorithm to hair images yields unsatisfying results, as we illustrate in Figure 3.6.

A few dedicated techniques have been designed to capture hair. Paris et al. [57] introduced the idea of estimating the orientation of hair in images, coupled with an analysis of the highlights on the hair. This analysis requires a light source to move along predefined known trajectories. Paris et al. [58] later described *Hair Photobooth*, a complex system made of several light sources, projectors, and video cameras that captures a rich set of data to extract the hair geometry and appearance. Jakob et al. [32] showed how to capture individual hair strands using focal sweeps with a camera controlled by a robotic gantry. While accurate, these active techniques are expensive and the capture is inherently slow because of their design. For example, the method of Paris et al. [58] relies on time multiplexing and requires thousands of images for a single reconstruction. Similarly, Jakob et al. [32] uses many input photographs with a sweeping focus plane across the hair volume.

Wei et al. [80] proposed a purely passive technique based on several handheld photographs. Their approach also relies on hair orientation fields, but uses a coarse *visual hull* as the approximate bounding geometry for hair growing.

The numerical accuracy of existing hair acquisition techniques remains unexplored since only visual evaluations were conducted. We address this shortcoming by performing ground-truth analyses using synthetically generated data (see Section 3.5).

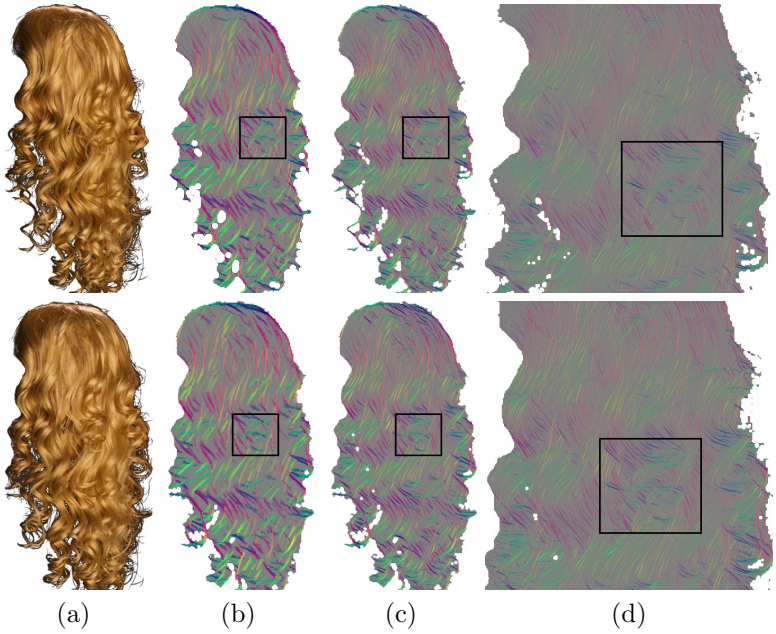


Figure 3.3: The multi-resolution orientation fields (b)-(d) computed from two nearby input images (a) shown in separate rows. Note that the local hair orientations highlighted by boxes correspond naturally between the two views. Finer levels of orientation field reveal high resolution hair details.

3.1.2 Contributions

Compared to previous work, we introduce the following contributions:

- a passive multi-view stereo approach to reconstruct finely detailed hair geometry;
- a robust matching criterion based on the local orientation of hair;
- an aggregation scheme to gather local evidence of hair structures;
- a progressive template fitting procedure to fuse multiple depth maps; and
- a quantitative evaluation of our acquisition system.

3.2 Local Hair Orientation

Because hair is highly specular, its color varies quickly when the viewpoint moves, making standard multi-view stereo fail, as shown in Figure 3.6. We address this issue by replacing colors with local orientations. In this section, we describe how we reliably estimate local directions of hair strands.

Paris et al. [57] first introduced dense orientation fields for hair modeling. Our orientation field computation differs from the prior one in that we only consider highlighted hair strands (i.e., positive filter response). We observe in our experiments that reliable orientations are difficult to obtain in

dark regions due to the poor signal-to-noise ratio of negative parts of the filter response. Instead, we recover the orientations in dark regions using available data obtained from the coarser resolution level where more data is aggregated and/or from a finer resolution level where finer and brighter structures are revealed.

Formally, given oriented filters K_θ generated by rotating an original x-aligned filter K_0 by angles $\theta \in [0, \pi)$, we define the orientation $\Theta(x, y)$ of image I at pixel (x, y) as $\Theta(x, y) = \arg\max_\theta |K_\theta * I(x, y)|$. To eliminate the $\pm\pi$ ambiguity of the orientation, we map Θ to the complex domain as in [57] by $\Phi(x, y) = \exp\big(2i\Theta(x, y)\big)$. We also use the (nonlinearly mapped) maximum response $F(x, y) = \max_\theta |K_\theta * I(x, y)|^\gamma$ as a confidence measure in our stereo algorithm: it captures both the strength of the image intensity and confidence in the orientation, at the filter's characteristic scale. The power-law mapping enhances weak responses and improves reconstruction quality. We use $\gamma = 0.5$ for all our datasets. Finally, our orientation field $O(x, y)$ is defined by taking the product of $\Phi(x, y)$ and $F(x, y)$, where the maximum filter response was positive:

$$O(x, y) = \begin{cases} F(x, y)\, \Phi(x, y), & \max_\theta\big(K_\theta * I(x, y)\big) > 0 \\ 0, & \max_\theta\big(K_\theta * I(x, y)\big) \leq 0 \end{cases} \tag{3.1}$$

We select a DoG filter for K_0. Specifically, $K_0(x, y) = \big(G_\sigma(x) - G_{\sigma'}(x)\big)\, G_{\sigma''}(y)$, where G_σ is a 1D zero-mean Gaussian with standard deviation σ. We use filters 1 degree apart, for a total of 180 filters.

Next, we describe a coarse-to-fine optimization strategy that requires orientation fields at multiple resolutions. We generate these fields with a pyramid structure to accelerate the computation: we recursively downsample the image for coarse levels in the pyramid and apply each oriented filter K_θ. We use a fixed sized K_θ with $\sigma = 0.5$, $\sigma' = 1$ and $\sigma'' = 4$ for all levels of the orientation field. The multi-resolution oriented pyramid is visualized in Figure 3.3.

3.3 Partial Geometry Reconstruction

In this section, we assume that we have a set of images of the hair from a few nearby viewpoints. We will discuss in Section 3.5 the specific setups we have used in our experiments. For now, we focus on reconstructing the partial geometry of the hair seen from these viewpoints. In Section 3.4 we will describe how to merge the pieces coming from several groups of cameras to form a full head of hair.

We formulate the partial reconstruction process as an MRF optimization based on the computed orientation fields, i.e., we seek to reconstruct the geometry that best explains the orientations observed in each view. To make this process robust, we use a coarse-to-fine strategy and locally aggregate evidence using a scheme inspired by [60].

We reconstruct a depth value $D(p)$ for each pixel p of the center reference view using orientation fields computed from all cameras. The reconstruction volume is bounded by the nearest and farthest depths, d_{near} and d_{far}.

3.3.1 Energy Formulation

We use an MRF energy minimization framework to optimize for D. The total MRF energy $E(D)$ with respect to D consists of a data term $E_d(D)$ and a smoothness term $E_s(D)$:

$$E(D) = E_d(D) + \lambda E_s(D), \tag{3.2}$$

where λ is the smoothness weight. The data energy is the sum of the per-pixel data cost $e_d(p, D)$ for each pixel p of the reference view while the smoothness energy is the weighted sum of the depth deviation between p and its 4-connected neighbors $\mathcal{N}(p)$:

$$\begin{aligned} E_d(D) &= \sum_{p \in pixels} e_d(p, D) \\ E_s(D) &= \sum_{p \in pixels} \sum_{p' \in \mathcal{N}(p)} w_s(p, p') \left| D(p) - D(p') \right|^2. \end{aligned} \tag{3.3}$$

The MRF cues $w_s(p, p')$ encode different depth continuity constraints between adjacent pixels p and p'. To enforce a strong depth continuity along the hair strands where orientations are similar, we define $w_s(p, p')$ as a Gaussian of the orientation distance in the reference image:

$$w_s(p, p') = \exp\left(-\frac{\left| O_{\text{ref}}(p) - O_{\text{ref}}(p') \right|^2}{2\sigma_o^2} \right). \tag{3.4}$$

The parameter σ_o controls the constraint sensitivity and is set to $\sigma_o = 0.15$ for all our datasets.

Similar to [63], we formulate the data term e_d based on the multi-resolution orientation field computed in Section 3.2. We define e_d as the sum of the matching costs $e_d^{(l)}$ of each level l from the

orientation field for all views:

$$
\begin{aligned}
e_d(p, D) &= \sum_{l \in levels} e_d^{(l)}(p, D) \\
e_d^{(l)}(p, D) &= \sum_{v \in views} c_v\Big(O_{\text{ref}}^{(l)}(p), O_v^{(l)}\big(P_v(p, D)\big)\Big),
\end{aligned}
\tag{3.5}
$$

where $O_{\text{ref}}^{(l)}$ and $O_v^{(l)}$ are the orientation fields at level l of the reference view and of adjacent view v, respectively. $P_v(p, D)$ is the projection of the 3D point defined by the depth map D at pixel p onto view v. The cost function c_v for adjacent view v is defined as:

$$
c_v(O, O') = -\Re\big\{O^* O' \exp\big(2i(\phi_{\text{ref}} - \phi_v)\big)\big\}, \tag{3.6}
$$

where $\Re(z)$ denotes the real part of a complex number z, ϕ_{ref} and ϕ_v are the angles between image x-axis and the vector from the image principal point to the epipole of the other view for reference view and adjacent view v. Intuitively, $c_v(O, O')$ measures the deviation of the orientation fields for corresponding pixels as the negative correlation of the two orientation vectors O and O', and the correction factor $\exp\big(2i(\phi_{ref} - \phi_v)\big)$ compensates for the influence of the camera pair's different tilting angles on the orientation field comparison.

The data term $e_d(p, D)$ is a function on the volume defined by the pixel image of the reference view and each possible depth value d in the interval $[d_{\text{near}}, d_{\text{far}}]$

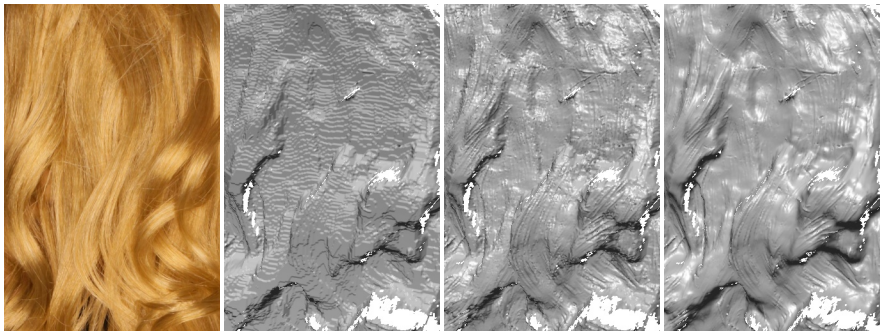
3.3.2 Structure-Aware Aggregation

To improve robustness and adaptivity of the data term energy to the local structure of the orientation field, we perform cross guided filtering [26] on each level l based on the orientation field of the reference view on that level. This process builds upon the idea of structure-aware aggregation introduced by Yoon and Kweong [89] for stereo and applied to other problems by Rhemann et al. [60]. However, none of these techniques can be directly applied to hair because they rely on color data. We address this issue in the rest of this section.

Before the data energy of each level $e_d^{(l)}(p, D)$ is summed in 3.5, each data energy at depth D of pixel p in the energy volume is aggregated as a weighted average of data energies of neighboring pixels p':

$$
e_d^{(l)}(p, D) \leftarrow \sum_{p' \in \omega_p} W^{(l)}(p, p')\, e_d^{(l)}(p', D), \tag{3.7}
$$

where $W^{(l)}(p, p')$ is the guided filter weight and ω_p is a local window centered at p.



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

Figure 3.4: The stages of depth map refinement improve reconstructed quality. The reference input image (a). The surface reconstructed from the initial MRF-optimized depth map (b) shows quantization artifacts that are removed by sub-pixel refinement (c). A post-reconstruction guided filtering step further improves quality (d).

Generalizing the expressions derived in [26], we define the weight $W^{(l)}(p, p')$ based on the orientation field:

$$W^{(l)}(p, p') = \frac{1}{|\omega|^2} \sum_{k:(p,p')\in\omega_k} \left(1 + \frac{\Re\{(O(p) - \mu_k)^*(O(p') - \mu_k)\}}{\sigma_k^2 + \epsilon}\right), \tag{3.8}$$

where the summation is over all pixels k such that p and p' are in a local window ω_k around k, $|\omega|$ is the number of pixels in the window, ϵ controls the structure-awareness based on the orientation, and μ_k and σ_k are the average and standard deviation of orientation, respectively, within ω_k.

Note that the computation in Equation (3.7) can be done efficiently regardless of the size of ω_k [26]. This enables efficient aggregation with large-kernel windows on high resolution datasets. We use 7×7, 11×11, and 15×15 kernel windows for the 3 levels of orientation fields, and we set $\epsilon = 0.1^2$ for all of our examples.

After aggregation, the resulting energy in Equation (3.2) can be efficiently minimized by graph cuts [11].

3.3.3 Depth Map Refinement

We employ a sub-pixel refinement technique similar to [2] to refine the integer depth map optimized by graph cuts. To be specific, for each pixel p on the reference view and its associated depth $D(p)$, we look up its data cost $e_0 = e_d\big(p, D(p)\big)$ and the data cost $e_{-1} = e_d\big(p, D(p) - 1\big)$ and $e_{+1} = e_d\big(p, D(p) + 1\big)$ for the adjacent depth values $D(p) - 1$ and $D(p) + 1$. The subpixel depth

$D'(p)$ is computed as:

$$D'(p) = \begin{cases} D(p) - 0.5, & e_{-1} < e_0, e_{+1} \\ D(p) + 0.5\frac{e_{-1}-e_{+1}}{e_{-1}-2e_0+e_{+1}}, & e_0 \ < e_{-1}, e_{+1} \\ D(p) + 0.5, & e_{+1} < e_0, e_{-1} \end{cases} \tag{3.9}$$

We then apply guided filtering once again on the depth map based on the finest orientation level to further reduce the stereo noise with the same weights as in Equation 3.8:

$$D'(p) \leftarrow \sum_{p' \in \omega_p} W(p, p')\, D'(p'). \tag{3.10}$$

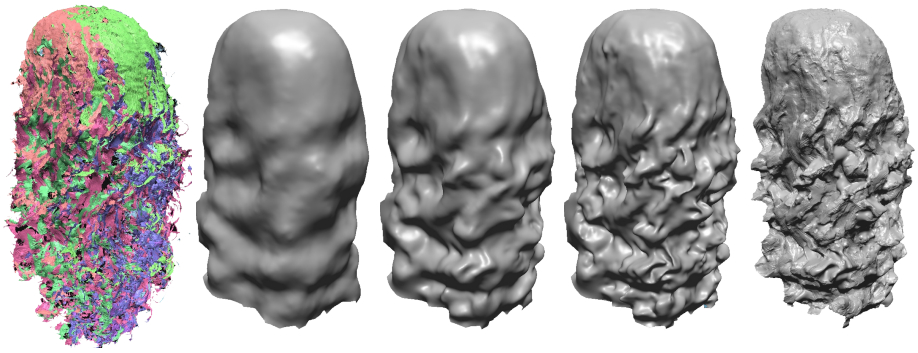
Figure 3.4 shows how the reconstructed surface evolves after applying each of the refinement steps discussed above. Note the importance of subpixel refinement, without which the features are overwhelmed by quantization artifacts. The post-reconstruction guided filtering step increases surface quality modestly, but is not a replacement for the structure-aware aggregation.

3.4 Final Geometry Reconstruction

The previous section described how we produce a set of partial reconstructions (depth maps) of the hair volume using small groups of nearby views. We combine these into a model of the full head of hair by aligning and merging these pieces.

We begin by forming a *coarse template* that establishes the topology and overall geometry of the merged reconstruction. This step accounts for the fact that different depth maps see different portions of the hair and, in overlapping regions, may have misalignment. We use *Poisson surface reconstruction* [34] on the union of all input points, but restrict the depth of the octree to level 6 (i.e., reconstructing the surface at a resolution of $2^6 \times 2^6 \times 2^6$ voxels) in order to capture only the coarsest geometry of the hair.

We then form a *refined template* by deforming the coarse template towards each of the original depth maps, moving along each depth map's line of sight. The warping is done using a graph-based non-rigid registration algorithm [43] that maximizes local rigidity, so that missing data in each depth map does not result in a bumpy surface, and so that the amount of deformation may be controlled (i.e., regularized). We repeat this deformation step by reinitializing its rest energy state 10 times,



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

Figure 3.5: The stages of final geometry reconstruction. (a) Partial depth maps from all views. (b) Coarse template. (c)-(d) Template refinement. (e) Re-introducing high frequency details.

reducing the amount of regularization for later iterations. The resulting template effectively averages the shape of all the input depth maps where they overlap, while remaining smooth.

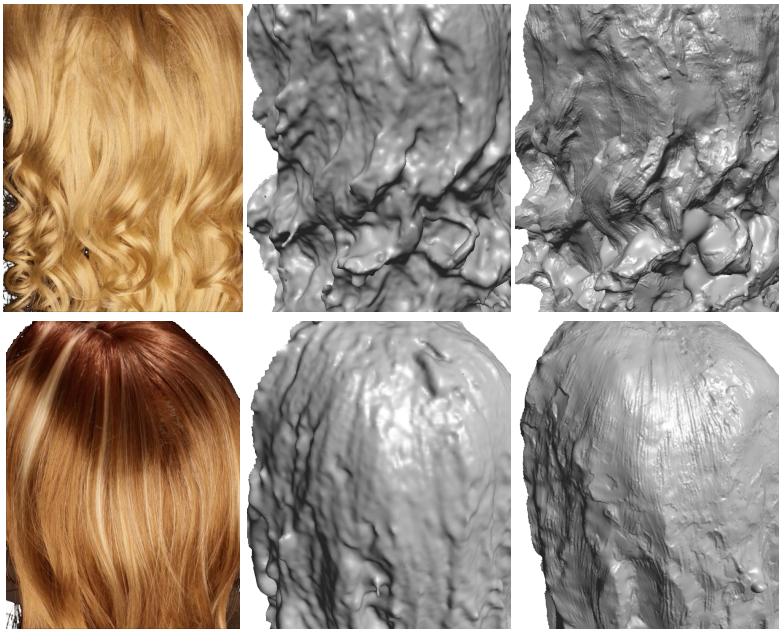
The *final mesh* is obtained by re-introducing high-frequency details from the depth maps onto the refined template. This is done using a more efficient linear deformation model based also described in [43] where the high-frequency details are represented as offsets along the mesh vertex normals.

We have found that this three-stage process successfully combines the goals of establishing a consensus topology (rejecting spurious disconnected components), averaging geometry in overlapping regions, and maintaining the details present in the original depth maps to the maximum extent possible (Figure 3.5). In contrast, previous (single-step) surface merging algorithms average away details in the presence of misalignment in overlap regions, and are often unable to cope with spurious topology.

3.5 Evaluation

We demonstrate the performance of our approach using a variety of setups. First, we show results for a large (8×4 camera positions) set of high-resolution (21 Mpix) images of static subjects: wigs. We use a robotic camera gantry for our setup, fix a wig on the central turn table, and use a Canon EOS 5D Mark II SLR camera to capture images (Figure 3.2). The arm of the gantry is about 60cm long and can be rotated freely, horizontally and vertically, to arbitrary latitudes and longitudes around the hair.

The camera is positioned in groups to obtain aggregate views of the hair for the depth map computation in Section 3.3. Each group consists of four different camera positions in a T-pose: center, left, right and bottom. Each position is 10 degrees apart from the neighboring position in



(a) (b) (c)

Figure 3.6: Qualitative evaluation on the two real captured datasets of different hair styles (a) between the state-of-the-art multi-view stereo methods: PMVS + Poisson [23, 34] (b) and our method (c). Note that our method preserves hair strand details.

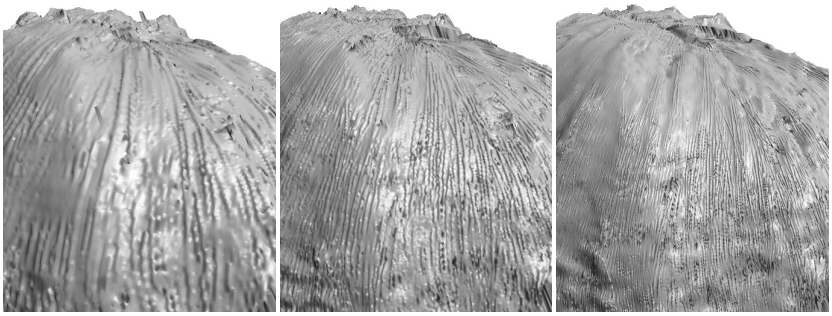


Figure 3.7: Reconstruction results on different levels. From left to right the resolution of the depth map increases from 0.4M to 1.5M and 6M pixels, respectively.

terms of gantry arm rotation. The left and right cameras in the T-pose provide balanced coverage with respect to the center reference camera. Since our system employs orientation-based stereo, matching will fail for horizontal hair strands (more specifically, strands parallel to epipolar lines). To address this problem, a bottom camera is added to extend the stereo baselines and prevent the "orientation blindness" for horizontal strands.

We use 8 groups of 32 views for all examples in this work. Three of these groups are in the upper hemisphere, while a further five are positioned in a ring configuration on the middle horizontal plane, as shown in Figure 3.2. We calibrate the camera positions with a checkerboard pattern [95], then

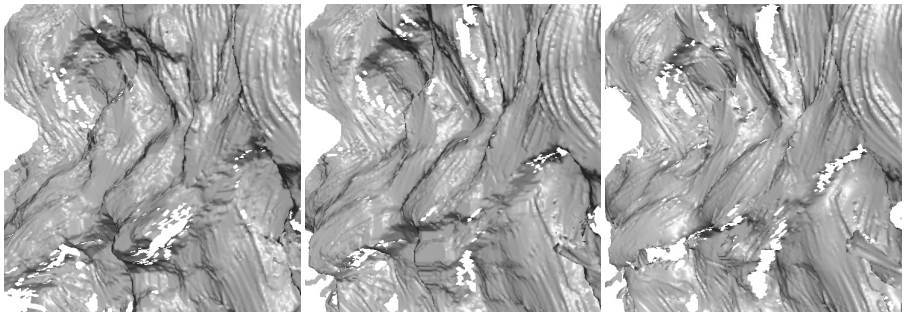


Figure 3.8: Comparison between the depth map reconstructed with 2, 3, 4 cameras.

perform foreground-background segmentation by background color thresholding combined with a small amount of additional manual keying. A large area light source was used for these datasets.

Qualitative Evaluation The top two rows of Figure 3.11 show reconstructions for two different hairstyles, demonstrating that our method can accommodate a variety of hairstyles — straight to curly — and handle various hair colors. We also compare our results on these datasets with [23] and [34] in Figure 3.6. Note the significant details present in our reconstructions: though we do not claim to perform reconstruction at the level of individual hair strands, small groups of hair are clearly visible thanks to our structure-aware aggregation and detail-preserving merging algorithms.

In Figure 3.7 and Figure 3.8, we show how our reconstruction algorithm scales with higher resolution input and more camera views. Higher resolution and more views greatly increase the detail revealed in the reconstructed results.

Quantitative Evaluation To evaluate our reconstruction accuracy quantitatively, we hired a 3D artist to manually create a highly detailed hair model as our ground truth model. We then rendered 8 groups of 32 images of this model with the same camera configuration as in the real capture session. We ran our algorithm on the images and compared the depth maps of our reconstruction and the ground truth model from the same viewpoints. The results are shown in Figure 3.9. On average, the distance between our result and the ground truth model is 5 mm, and the median distance is 3 mm. We also ran a state-of-the-art multi-view algorithm [23, 34, 2] on the synthetic dataset, and the statistics of its numerical accuracy are similar to ours. However, as shown in Figure 3.9, their visual appearance is a lot worse with the presence of blobs and spurious discontinuities.

Timings Our algorithm performs favorably in terms of efficiency. On a single thread of a Core i7 2.3GHz CPU, each full-resolution (1404×936) depth map reconstruction takes 4.5 minutes. Multiple depth map reconstructions can be easily parallelized using more cores. In the final geometry

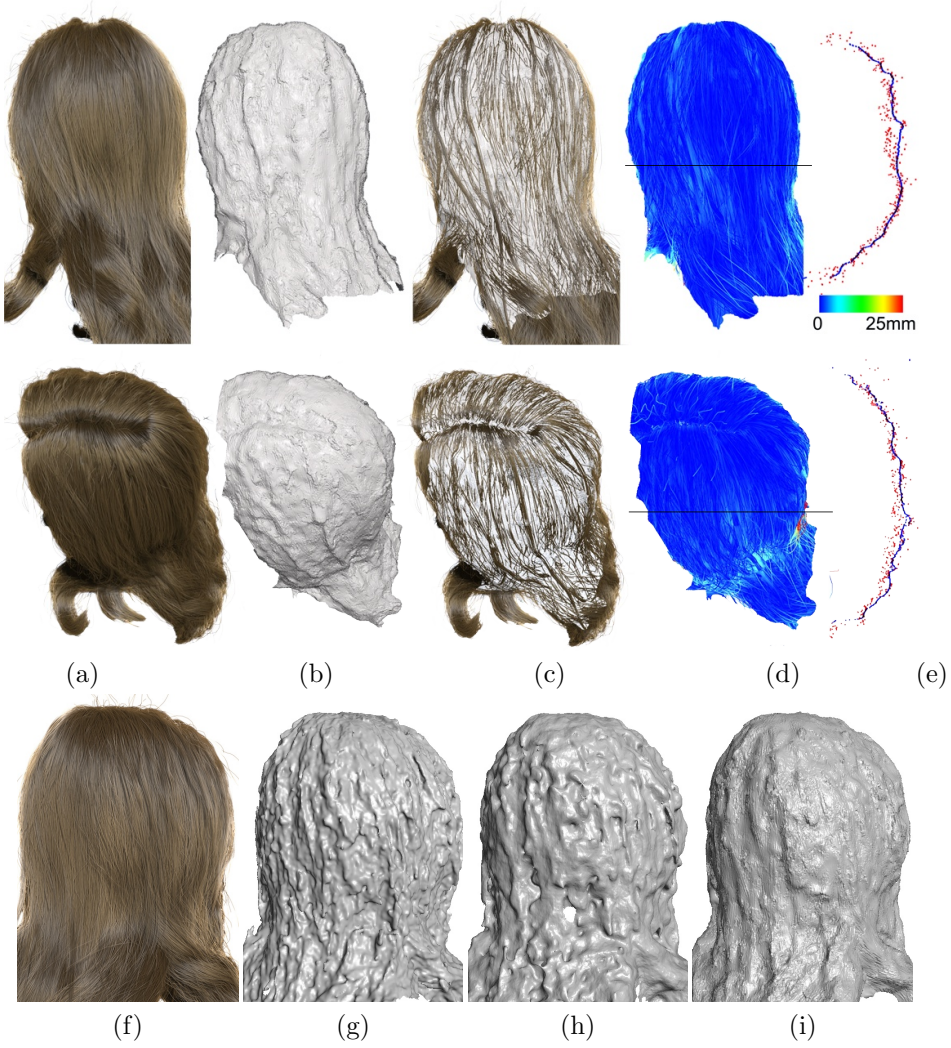


Figure 3.9: We evaluate the accuracy of our approach by running it on synthetic data (a), (f). The result is shown in (b), and is overlaid to the synthetic 3D model in (c). The difference between our reconstruction in the ground-truth 3D model is on the order of a few millimeters (d). We show a horizontal slice of the depth map in (e): the ground-truth strands are shown in red and our reconstruction result in blue. Compared to PMVS + Poisson [23, 34] (g) and [2] (h), our reconstruction result (i) is more stable and accurate.

reconstruction stage, the coarse template reconstruction (via Poisson) takes on average 30 seconds, template refinement 5 to 6 minutes, and final detail synthesis 20 seconds. In comparison, the Hair Photobooth [58] timings are on the order of several hours.

Dynamic Hair Capture A major advantage of our approach over previous work [58, 32] is that, being completely passive, it is amenable to simultaneous multi-view acquisition. This paves the way towards capturing hair in motion. As a proof of concept, we built a dynamic capture setup made of 4 high-speed video cameras capturing 640×480 pixels at 100 frames per second. Figure 3.10

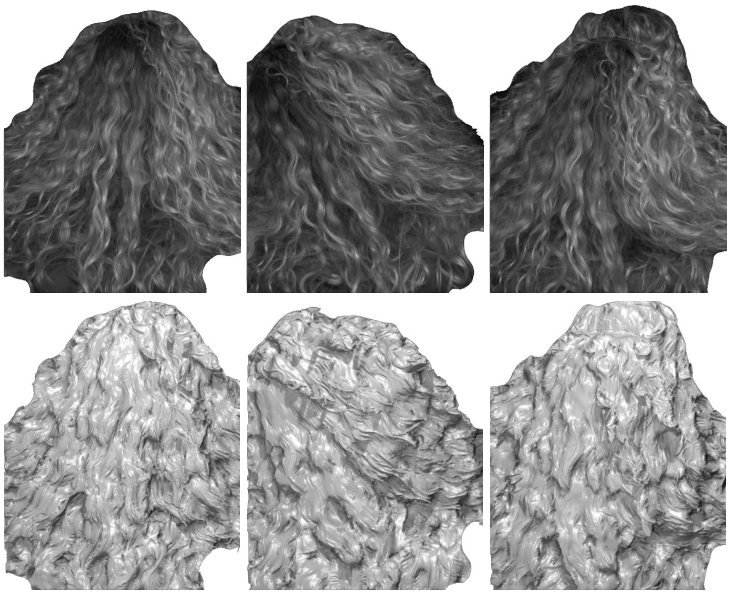


Figure 3.10: Sample frames (first row) and the reconstructed depth maps (second row) from our dynamic hair capture setup.

shows the reconstruction results of a few sample frames from 4 captured 200-frame videos (see accompanying video for full input videos and the reconstruction results). However, because of the limited resolution of the high-speed video cameras, we were not able to achieve similar quality to our static reconstructions.

3.6 Conclusion and Future Work

We have proposed a passive multi-view reconstruction algorithm based on multi-resolution orientation fields. We demonstrated quantitatively that accurate measurements can be achieved by using orientation-based stereo. Combined with structure-aware aggregation, our method faithfully recovers detailed hair structures that surpass the quality of previous state-of-the-art methods. We also demonstrate the capability of our method to capture hair geometry in motion.

However, there are several limitations in our current method that needs to be addressed in the future. The structure-aware aggregation step, performed on the matching energy volume, imposes a fronto-parallel bias on the reconstruction result. This bias becomes considerable towards the edges of the reconstructed depth map, resulting in spurious and misaligned geometry. A possible solution is to use slanted window matching similar to [12] that fuses information from different depth layers.

As mentioned in Section 3.5, our orientation-based matching metric is "blind" for strands parallel to the epipolar plane. This can be addressed by adding more cameras of different baselines. Although

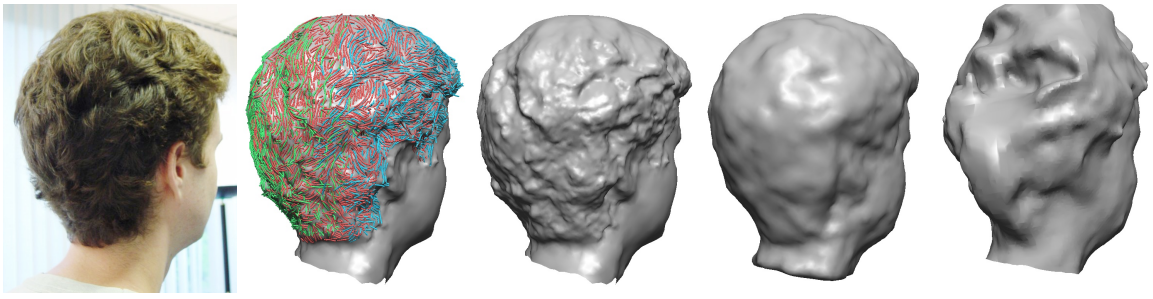
Figure 3.11: Final results for the three datasets (Alice, Lindsay and synthetic) reconstructed using our method. For each, we show one overview and two close-up views along with their reference images.

we have experimented on a few different camera configurations, it still requires further investigation to find out the best possible configuration for our orientation-based hair capture system for optimal coverage of the entire head.

Chapter 4

Wide-Baseline Hair Capture Using Strand-Based Refinement

Typical hair capture methods involves complex acquisition setups with densely sampled viewpoints. In this chapter, we introduced a wide-baseline hair capture system that requires only 8 camera views to reconstruct a complete full hairstyle using a robust shape refinement scheme called strand-based refinement. Our system gains the robustness against matching ambiguities in the wide-baseline setup by aggregating the matching cost along the long and continuous strands extracted from each input image.



Input image *Refined strands* *Our result* *Hernandez [20]* *PMVS [23]*

Figure 4.1: 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 [20] or unable to converge in the wide-baseline setup [23].

4.1 Introduction

Multi-view stereo methods have been widely used to reconstruct real world objects with ever improving quality [23, 2]. 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 [82] for hair reconstruction. The highly specular nature of hair [50] 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. [2]), or relied on manual input to achieve plausible results [54].

Researchers have explored specialized hardware to facilitate hair capture, such as a fixed camera with moving light sources [58], a stage-mounted camera with macro lens [32], thermal imaging [27], 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 [80, 46, 27]. 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 [46, 80] 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 cross-view reconstruction consistency.

We quantitatively evaluate our reconstruction method on synthetic hairstyles, and achieve an accuracy of $\sim$3mm. 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.

4.2 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.

4.2.1 Hair Capture

A few dedicated systems in the literature have been designed for hair capture. Paris et al. [57] 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. [58] 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. [32] 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 [27]. 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. [80] 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. [87] used an array of 12 cameras to capture partial geometry of straight hair in moderate motion. Guided by hair simulation, Zhang et al. [93] reconstructed smooth hair dynamics with 7 cameras. Using only a single view, Chai et al. [14] proposed a method to generate a depth map for convincing view interpolation of different hairstyles. Beeler at al. [3] 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.

4.2.2 Related Multi-view Stereo Methods

There have been many multi-view stereo methods presented in the literature [68]. 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.

Figure 4.2: 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.

Space carving [39] reconstructs objects by eliminating voxels in a volume with low photo-consistency across visible views. Inspired by the active contour method [33], 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 [20] proposed a visual hull refinement method by iteratively minimizing the texture, silhouette and surface smoothness energies. Furukawa and Ponce [22] 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 [45] is capable of finding feature points on images with significantly different viewpoints and illuminations. Tola et al. [71] extended the idea of SIFT to find dense correspondences across views for high quality reconstruction. Local regions with view-invariant properties have also been studied such as affinely invariant regions [73] and maximally stable extremal regions [51]. However, due to the lack of reliable texture and corner-like features on hair, it is difficult to apply these methods on hair reconstruction.

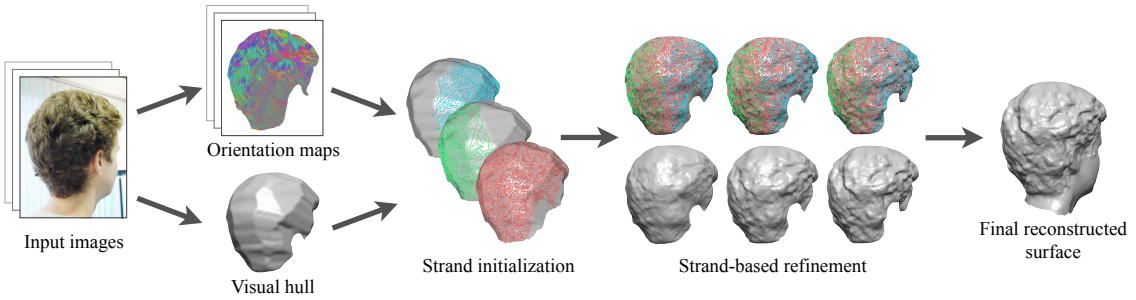


Figure 4.3: The overview of our reconstruction method. We take 8 input images from different views and compute the orientation map for each. Using the visual hull constructed from the segmented images, we extract strands on the orientation maps and project them from each view onto the visual hull for 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 [34].

4.3 Overview

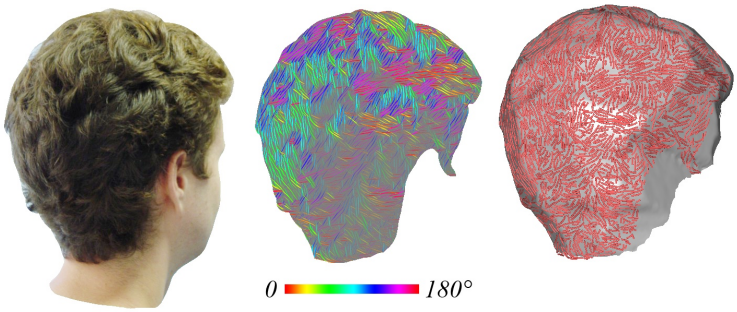
Given a set of wide-baseline images (see Fig. 4.2 for some sample images), our goal is to compute a shape that best approximates the captured hair volume. We achieve this by refining the positions of a dense set of representative 3D hair strands derived from each camera view.

Fig. 4.3 gives an overview of the various steps involved in our hair capture algorithm. We defer the description of our acquisition setup to Sec. 4.6. 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 [34] from the refined 3D strands.

In the rest of the work, we will describe strand initialization in Sec. 4.4, and present the novel strand-based refinement algorithm in Section 4.5. Experimental results and conclusions are given in Sections 4.6 and 4.7, respectively.

4.4 Strand initialization

We first compute an orientation map for each image using the method proposed in [46], 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 [14]. To



(a) (b) (c)

Figure 4.4: The steps of strand initialization in Sec. 4.4. 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).

further reduce noises in regions with low confidence, we apply the bilateral filtering method in [57] to diffuse the orientations of the high confidence regions.

We then track the confidence ridges of each orientation map (Fig. 4.4(b)) using hysteresis thresholding similar to [14]. 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. 4.4(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.

4.5 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. 4.5).

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:

$$E = \sum_{\star} \alpha_{\star} E_{\star} \tag{4.1}$$

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 [49].

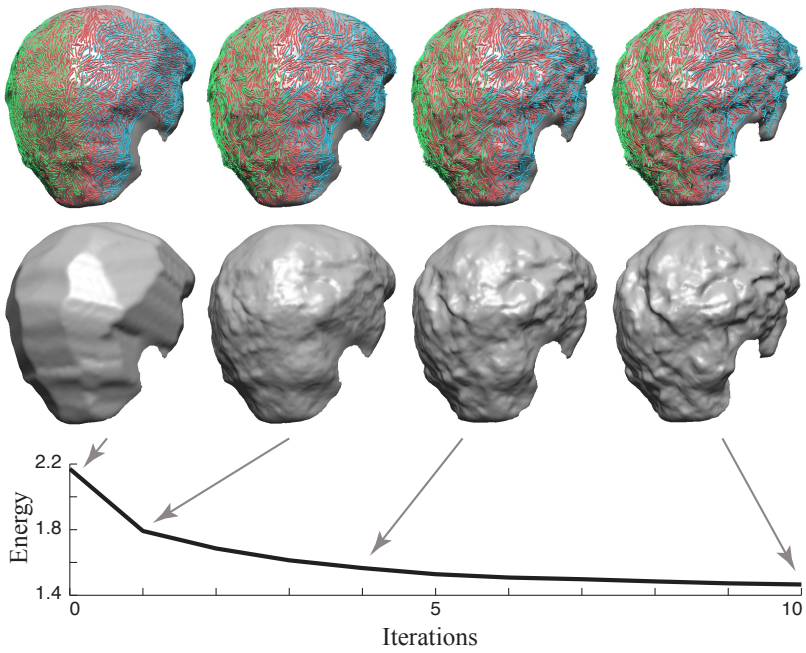


Figure 4.5: In strand-based refinement (Sec. 4.5), the strands (first row) are refined over the iterations with their reconstructed surfaces (second row) revealing more hair details.

4.5.1 Notations and Definitions

Let p denote a strand vertex on a 3D strand S. We use subscript to reference its successor p_{+1} and 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. 4.6. $\mathcal{N}^+(p)$ 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:

$$w^-(p, q) = \exp\left(-\frac{\|p - q\|^2}{2\sigma_e^2}\right) \tag{4.2}$$

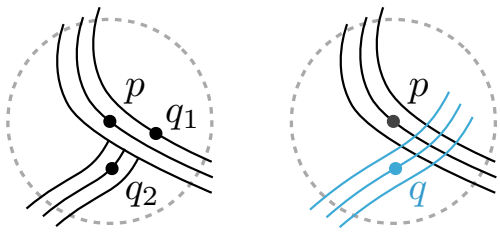


Figure 4.6: The illustrations of same-view neighborhood $\mathcal{N}^+(p)$ and different-view neighborhood $\mathcal{N}^-(p)$ for p. The neighbors are searched within the radius ($2.5\sigma_e$) indicated by the dashed circles. Same-view neighbors q_1 and q_2 can be weighted differently by how their orientations differ from p's. The different-view neighbor q is located on the strands from a different reference view (in blue).

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 distance and the orientation difference between p and q:

$$w^+(p,q) = \exp\left(-\frac{1-\langle d(p), d(q)\rangle^2}{2\sigma_o^2} - \frac{\|p-q\|^2}{2\sigma_e^2} \right) \tag{4.3}$$

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 p^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.

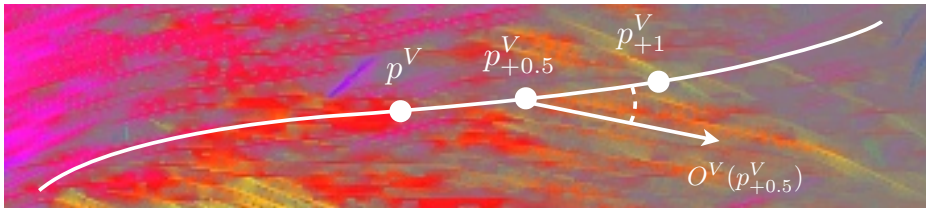


Figure 4.7: The illustration of orientation energy. A strand is projected on the orientation map in similar color coding as in Fig. 4.4. The orientation energy term $e^V_{orient}(p^V)$ is determined by the angle between $O^V(p^V_{+0.5})$ and $p^V_{+1} - p^V$.

4.5.2 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. 4.4 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. 4.7). We then define an orientation energy term $e^V_{orient}(p^V)$ for each segment (p, p_{+1}) on S as follows:

$$e^V_{orient}(p^V) = \min\left\{1 - \langle p^V_{+1} - p^V, O^V(p^V_{+0.5})\rangle^2, T_{orient}\right\} \tag{4.4}$$

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 $\pm 180°$ directional ambiguity.

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

$$E_{orient} = \sum_{p} \sum_{V \in \mathcal{V}(p)} w^V(p) e^V_{orient}(p^V), \tag{4.5}$$

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.

4.5.3 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. 4.8, given p's closest point $h(p)$ on $\mathcal{H}$ and $h(p)$'s normal n_h, we can define

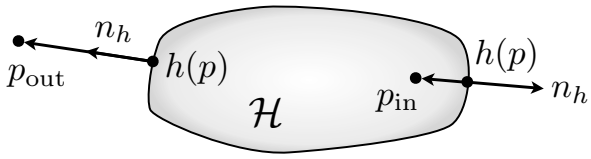


Figure 4.8: The illustration of silhouette energy. The sign of $(p - h(p)) \cdot n_h$ determines if a point is inside $\mathcal{H}$ and thus the value of β in silhouette energy for p_{in} and p_{out}.

silhouette energy E_{silh} as:

$$E_{silh} = \frac{1}{D^2} \sum_p \beta \Big(\big(p - h(p)\big) \cdot n_h \Big)^2 \tag{4.6}$$

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

$$\beta = \begin{cases} 1 & (p - h(p)) \cdot n_h \leq 0 \\ w_{out} & (p - h(p)) \cdot n_h > 0 \end{cases}, \tag{4.7}$$

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 used for unit-less energy formulation in the following sections.

4.5.4 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 [4] often used in hair simulation that minimizes the squared curvature along hair strands. Further inspired by [14], we take into account the orientation similarity in the bilateral same-view weight w^+ so that the 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 [4], we define the strand smoothness energy as the summation of squared curvature for each vertex along all the strands:

$$E_{strand} = D^2 \sum_p \text{curv}^2(p) \tag{4.8}$$

where curvature is computed as:

$$\text{curv}(p) = \frac{2}{l_{+1} + l_{-1}} \left\| \frac{p_{+1} - p}{l_{+1}} - \frac{p - p_{-1}}{l_{-1}} \right\| \tag{4.9}$$

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:

$$E_{wisp} = \frac{1}{D^2} \sum_p \left(\left(p - \sum_{q \in \mathcal{N}^+(p)} \bar{w}^+(p,q) q \right) \cdot n(p) \right)^2 \tag{4.10}$$

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:

$$E_{global} = \frac{1}{D^2} \sum_p \left(\left(p - \sum_{q \in \mathcal{N}^-(p)} \bar{w}^-(p,q) q \right) \cdot n(p) \right)^2 \tag{4.11}$$

4.6 Results

We use a camera rig that contains 8 cameras around the subject powered by a single workstation, as shown in Fig. 4.2. 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. 4.2. 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 [41], but 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:

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

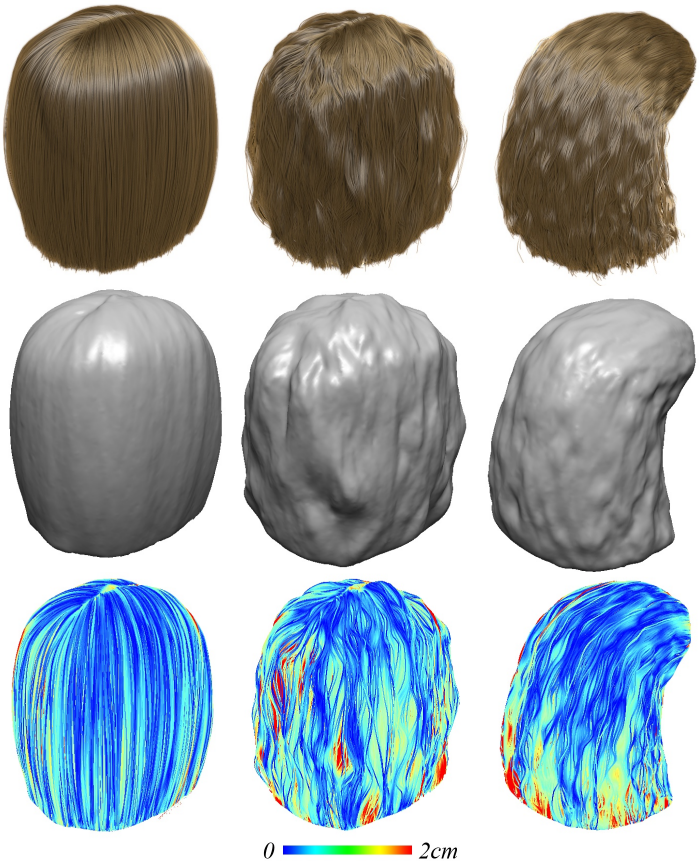


Figure 4.9: We evaluate the reconstruction accuracy on three synthetic hair styles (straight, wavy and wavythin, first row). We compute the depth maps of our reconstruction (second row) and compare them with the hair's. The depth map differences are visualized in coded color (third row). The average reconstruction error is around 3 millimeters.

The reconstruction results for all the examples are shown in Fig. 4.11. Note that we use [23] to reconstruct each subject's facial area and then merge our hair reconstruction using Poisson surface reconstruction [34]. 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 [20] loses details (Fig. 4.1). Multi-view stereo methods with weak regularization, such as [23], fail to converge to the correct shape (Fig. 4.1) due to the challenging wide-baseline setup.

Quantitative evaluation To quantitatively evaluate our method, we use the hair models by [90] and render them using the hair appearance model in [50], as shown in Fig. 4.9. 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

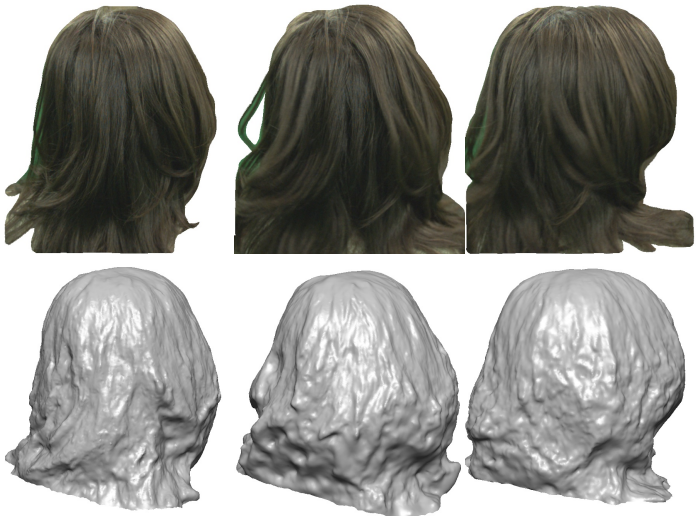


Figure 4.10: Sample frames (first row) and the reconstructed surfaces (second row) from the dynamic hair capture setup.

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. 4.9). 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 [58, 46], 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. 4.10 (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.

4.7 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 ∼3mm 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 compute 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.

Figure 4.11: Reconstruction results of all real examples. For each, we show three views with the reference input images.

Chapter 5

Structure-Aware Hair Capture

Most existing hair capture methods use unstructured hair models, i.e., the hair strands in the models are generated independently from the scalp according to the captured 3D orientation field and geometry. This proves impausible for subsequent hair editing and animation which usually involves structured models for high level control. In this chapter, we propose structure-aware hair capture, a method to reconstruct and infer underlying hair wisp structures from the unstructured and noisy point cloud input. The resulting model are robust against occlusion and missing data and plausible for animation and simulation.

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

Figure 5.1: Our system takes a collection of images as input (a) and reconstructs a point cloud with a 3D orientation field (b). In contrast to previous methods (e.g. [58]) that straightforwardly grow hair strands from the scalp following the orientation field and hence cannot reconstruct complex hairstyles with convoluted curl structures (c), we reconstruct complete, coherent and plausible wisps (d) aware of the underlying hair structures. The wisps can be used to synthesize hair strands (e) that are plausible for simulation (f).

5.1 Introduction

Shared between culture, nature, and sculpture, the hairstyle is a medium that creates a unique expression of self. A person's hairstyle is a vital component of his or her identity, and can provide strong cues about age, background, and even personality. The same is true of virtual characters, and the modeling and animation of hair occupy a large portion of the efforts of digital artists. Driven by increased expectations for the quality of lead and secondary characters, as well as digital doubles, this effort has only been increasing.

Acquiring complex hairstyles from the real world holds the promise of achieving higher quality with lower effort, much as 3D scanning has revolutionized the modeling of faces and other objects of high geometric complexity. This is especially true for digital doubles, which require high fidelity, and secondary animated characters, which may appear by the dozens and must receive less personalized attention from 3D modeling artists. However, even for lead characters that are modeled largely by hand, it is frequently easier to start with scanned data than to begin modeling from a blank canvas. This allows the digital hair to exploit the full talents of real-world stylists, who express their creativity through cutting, shearing, perming, combing, and waxing.

Despite the potential benefits of capturing real-world hairstyles, there is a large gap between the data produced by existing acquisition techniques and the form in which a hairstyle must end up to be incorporated into a production pipeline. Hair animation, whether done by hand or via physical simulation, typically operates on a collection of *guide strands*. Each of these is a curve through space, starting from a point on the scalp and going to the tip of the hair. The guide hairs must not intersect, and the entire collection must not be overly tangled. The hair model consists of tens of thousands of strands, whose motion is interpolated from the guide strands.

How close are existing hair capture systems to this representation? The raw output of 3D acquisition devices is typically an unstructured point cloud or a partial surface. Prior research on hair capture has typically augmented this geometry with a 3D orientation field, computed from orientations observed in a number of images. Given these two types of data, it is possible to grow a set of strands from the scalp whose position is constrained to the reconstructed geometry and whose direction follows the orientation field.

There are two major difficulties with this approach. First, the geometry will always suffer from missing regions and noise because of the complex occlusion patterns of hair. Second, the orientation field is necessarily extrapolated from what was observed on the outermost layer of hair. As a result, the grown hair strands exhibit a variety of undesirable artifacts: they may suddenly reverse direction

and form implausible U-shapes, diverge wildly from their neighbors, exhibit variations in density, or simply fail to reach all parts of the visible hair volume (see, for example, Figure 5.1c). Moreover, these independently-grown strands have no natural and coherent grouping structure that allows them to be controlled by guide hairs. As a result, it is impossible to incorporate these systems, based on naive strand-growing, into animation pipelines.

We describe a system that reconstructs structured hair models plausible for hair animation and simulation (Figure 5.1) by performing a higher-level analysis of the hair's structure. The system incorporates four key insights. First, it grows groups or *wisps* of hair coherently. This not only matches the structure of real hairstyles, which frequently contain strands that run nearly parallel to many of their neighbors, but also allows each wisp to be associated with a guide strand for animation. Second, it builds up a *connection graph*, allowing strands to span regions of missing geometric data. As long as two curvature-compatible portions of hair have been acquired, our system is able to connect across the gap between them. Third, it explicitly resolves the 180-degree ambiguity of orientation fields by performing a global Markov Random Field (MRF) optimization on the directions associated with nodes in the graph. This keeps strands from performing sudden U-turns. Finally, it ensures that each portion of visible surface is connected back to the scalp along some path. This allows our system to work for arbitrarily complex hairstyles, even if a naive strand-growing method would have difficulty reaching the end of, e.g., a complicated curl, by strictly following the orientation field.

Our algorithm begins with a point cloud and orientation field obtained from a collection of still images (Sec. 5.4) and extracts a set of local strand segments. These are clustered into "ribbons" that cover sets of parallel strands (Sec. 5.5), and connected across gaps in the point cloud (Sec. 5.6.1). Following a global direction analysis (Sec. 5.6.2), we connect up the ribbons to each other, and to the scalp (Sec. 5.7.1), to obtain complete wisps with plausible topology. These not only drive the final strand synthesis (Sec. 5.7.3), but also can be used as guide strands for hair simulation (Sec. 5.8).

5.2 Related Work

Hair capture. Most existing hair capture methods generate a strand set model by growing hair strands independently, constrained by the captured hair orientations and geometry. Paris et al. [57] proposed a method to estimate 3D hair orientations from highlights under a moving light source with known trajectory. They grow strands along the estimated orientations, starting at the scalp. Wei et al. [80] introduced a technique to create a strand model from the visual hull constructed from

many views. The strands are grown constrained by the orientation consistency across the views. Paris et al. [58] introduced an active acquisition system capable of accurately capturing the positions of the exterior hair strands. A strand model is generated by growing the strands within the diffused orientation field from the scalp to the captured exterior hair layer. Jakob et al. [32] proposed a system to capture fiber-by-fiber hair assemblies by growing hairs through the intersecting ribbons created by back-projecting the 2D strands with shallow depth-of-fields. Chai et al. [14] showed how to create an approximate strand model from a single image using the inter-strand occlusion relationships and the head model fit to match the face in the image. Beeler et al. [3] introduces a system to capture facial hairs using multi-view stereo matching. Their method employs a refinement method to improve the connections between the captured strand segments and remove outlier hairs. Luo et al. [46] proposed a method to grow a strand model in the orientation field constrained by the geometry constructed from hair orientation fields. Herrera et al. [27] applied thermal imaging to generate a strand model by growing strands on the boundary of the captured hairstyle. Their method joins the loose ends of nearby segments with smooth curvature and connects the strands to a user-defined ellipsoid as the scalp.

Geometry-based hair modeling. Structured models are commonly used in geometry-based hair modeling. A more complete survey on hair modeling techniques can be found in [78], here we only enumerate a few recent work relevant to ours. Ward et al. [79] used a high level skeleton representation to control various low level representations for level-of-detail hair modeling. Kim and Neumann [35] employed a hierarchy of generalized cylinders to model and edit hair in multi-resolution. Yuksel et al. [90] introduced "Hair Meshes" to model hair using polygonal meshes with various topological operations. Wang et al. [77] introduced a method to synthesize new hair styles from guide strands based on texture synthesis techniques.

Model-based reconstruction. Model fitting methods have been applied to the reconstruction of a variety of objects, e.g., architecture, furniture and trees. Livny et al. [44] uses a tree skeletal model to reconstruct tree structures from point cloud input. Nan et al. [55] introduced "SmartBoxes" to interactively reconstruct urban architectures with regular box-like structures. Li et al. [42] uses "Arterial Snakes" to reconstruct delicate interleaving man-made structures.

Comparison to our method. We build upon many of the ideas pioneered by the work above, including matching hair strands to a reconstructed orientation field, exploiting a constant-curvature prior to hypothesize connections between partial strands, and connecting reconstructed strands to

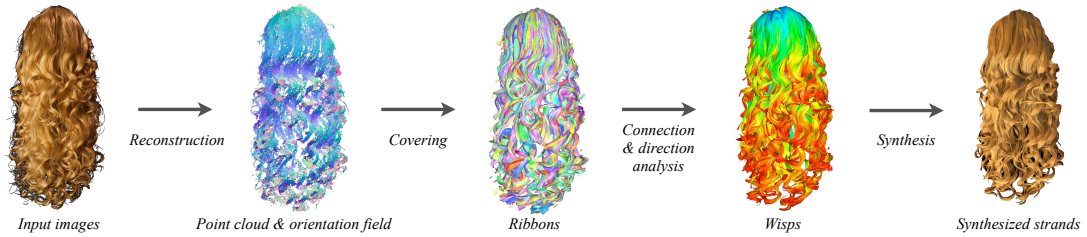


Figure 5.2: Overview of our method. We start with a collection of input images, reconstruct a point cloud with 3D orientation field, and cover the point cloud with ribbons that reveals the locally coherent wisp structures. A connection and direction analysis is then performed on the ribbons to determine their directions and connect them up into long wisps. Finally the wisps are used to synthesize the strands.

the scalp. However, we achieve a more plausible reconstruction that better matches the structures of real hair by focusing on *wisp* reconstruction, performing a global direction analysis, and using a connection graph data structure to find long connected paths between visible hair strands and points on the scalp. As a result, we are able to reconstruct complex hairstyles including curly and messy hair, and produce hair models that are plausible for animation and simulation.

5.3 Overview

Our method is shown in Figure 5.2. Its input is a set of images captured from multiple views for a real hairstyle (or, for some of our examples, a wig). We key the images to separate foreground and background, and use the Patch-based Multi-View Stereo (PMVS) [23] algorithm to reconstruct a raw point cloud with normals. We perform Moving Least Squares (MLS) fitting to filter the points and normals and compute the 3D orientation field on the points according to the 2D orientation maps (Sec. 5.4).

We then identify locally coherent wisp structures and group them into *ribbons* that cover the input point cloud. To achieve this goal, we first grow *strand segments* on the point cloud, following the 3D orientation field and stopping at discontinuities in orientation. (Sec. 5.5.1). Then we cover the grown strand segments with ribbons to expand the local regions into coherent wisp structures (Sec. 5.5.2).

Because of occlusion and missing data, the ribbons are disconnected from each other and do not form complete wisps. We discover missing connections between adjacent ribbons by trying to fit circular arcs to the covered strand segments of the ribbons (Sec. 5.6.1). Good fits indicate plausible connections between ribbons, which we encode in a *connection graph*. We also associate a growth direction with each ribbon by globally optimizing for compatible connections and local

hints of ribbon direction using a Markov Random Field (MRF) formulation (Sec. 5.6.2). Following the connections and optimized directions, the ribbons can be connected up to form complete *wisps* (Sec. 5.6.3).

We attach the wisps to the scalp of a manually fit head model (Sec. 5.7.1), then generate interior wisps to fill in the empty regions inside the hair volume, which had been occluded in the input (Sec. 5.7.2). Finally we synthesize strands using the complete attached wisps and the interior wisps (Sec. 5.7.3).

5.4 Reconstruction

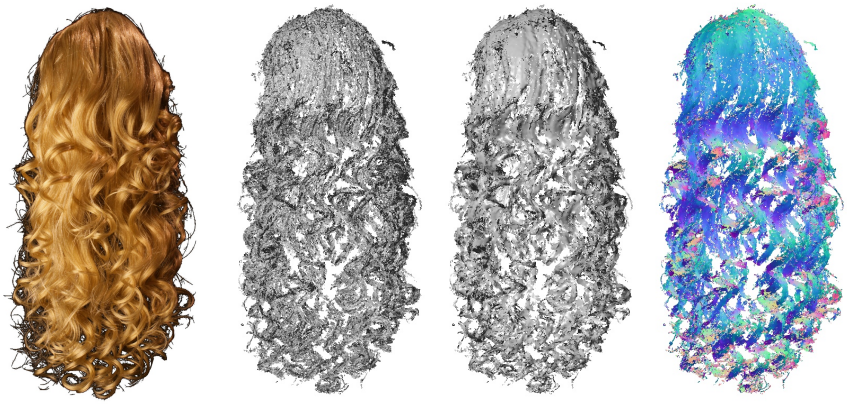
We begin by reconstructing an initial point cloud and 3D orientation field from a set of input images, acquired under unconstrained lighting. We semi-automatically segment out the hair from the background, then use PMVS [23], a state-of-the-art multi-view stereo algorithm, to reconstruct a point cloud. We use around $30 \sim 50$ input images for the reconstruction.

Filtering. The initial point cloud and the estimated normals from PMVS can be noisy, so we smooth them with Moving Least Squares (MLS) [40]: for each point, we fit an optimal plane to the locally-weighted neighbors near that point. The normal of the plane and the point's projection on the plane are then used to update the point's original normal and position. We use a sigma of 2mm to achieve plausible filtering results.

2D orientation maps. We compute an orientation map for each input image using the method of Luo et al. [46], which uses a bank of rotated filters to detect the dominant orientation at each pixel. The orientation map is then enhanced with 3 passes of iterative refinement for better signal-to-noise ratio, as proposed by Chai et al. [14]. To further reduce noise in regions with low confidence, we apply the bilateral filtering method of Paris et al. [57] to diffuse orientations from high-confidence regions.

3D orientation field. We use the 2D orientation maps to compute a 3D orientation field on the technique of Wei et al. [80]. However, it is challenging to determine the visibility of each point in a point cloud to all the input views. We find the following scheme effective in our situation without having to apply the more sophisticated methods such as [52].

For each point p in the point cloud with normal n, we find a reference view Y among all the views which has the minimum angle between n and the line-of-sight vector $v = \langle c - p \rangle$, where c is the view's



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

Figure 5.3: Point cloud and orientation field generation. From a set of input images (a) we compute a point cloud (b) using PMVS [23]. We then apply moving least squares fitting to smooth out the reconstruction noise (c) followed by computing a 3D orientation field (d) based on the point cloud and the input images.

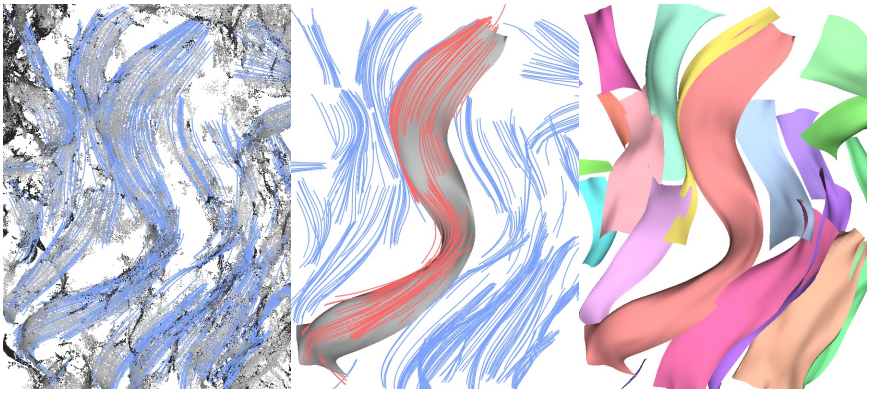
projection center and $\langle\cdot\rangle$ is the normalization operator (i.e., $\langle A\rangle = A/\|A\|$ for any vector $A \neq 0$). We then compute a reference 3D orientation $o_Y = \langle v_Y \times d_Y \times n\rangle$, where d_Y is the orientation at p's projection on Y's orientation map and v_Y is Y's line-of-sight vector. We can determine a view V as one of the visible views $\mathcal{V}$ to p if V's derived 3D orientation $o_V = \langle v_V \times d_V \times n\rangle$ at p is compatible with o_Y, i.e., $|o_V \cdot o_Y| > T_o$, where v_V is V's line-of-sight vector and d_V is the orientation at p's projection on V's orientation map. $T_o = 0.5$ is a threshold to reject outlier views invisible to p. The final 3D orientation o at p can be computed by maximizing

$$\rho = \max_o \frac{\sum_{v\in\mathcal{V}} w_V (o \cdot o_V)^2}{\sum_{v\in\mathcal{V}} w_V}, \quad \text{subject to } \|o\| = 1, \tag{5.1}$$

where $w_V = \max(n \cdot v_V, 0)$. This can be solved efficiently by singular value decomposition. ρ is defined as the confidence of the 3D orientation o. Note that our formulation ensures that the 3D orientation o for each point p is normal to the point's normal n.

With the 3D orientation defined at each point in the point cloud, the 3D orientation $o(p)$ at any point p can be computed as:

$$o(p) = \arg\max_o \sum_{q\in\mathcal{N}(p)} \exp\left(\frac{\|p-q\|^2}{2\sigma_d}\right) \rho_q (o \cdot o_q)^2, \quad \text{subject to } \|o\| = 1, \tag{5.2}$$



(a) (b) (c)

Figure 5.4: The steps of covering. Strand segments are first grown to cover the input point cloud (a). A ribbon is then expanded to cover adjacent strand segments (b). The covering continues until all the strand segments are covered by ribbons and thus reveals the locally coherent wisp structures (c).

where $\mathcal{N}(p)$ is the set of neighboring points around p and σ_d the parameter to control interpolation smoothness (2mm in our experiments). o_q and ρ_q are the orientation and confidence of a neighboring point q respectively. See Figure 5.3.

5.5 Covering

We begin the process of analyzing the hair's structure by proceeding bottom up: we first grow local *strands*, and then group coherent groups of strands into *ribbons*. In this way, we cover most of the point cloud with ribbons, omitting only those portions where we did not find coherent structures (Figure 5.4).

5.5.1 Covering by Strand Segments

We grow a number of strand segments $\mathcal{S}$ from a set of seed points. Typically we use all the points in the point cloud as seed points. A strand segment $S \in \mathcal{S}$ is a chain of vertices $(p_1, \ldots, p_N)$ connected by line segments in 3D space.

Growing. Starting from a seed point q we grow S in both directions $t(q) = \pm o(q)$, constrained by the 3D orientation field as well as the point cloud. For each growing direction, we repeatedly perform forward Euler steps to extend S from the current growing vertex p_i: $p'_i = p_i + t(p_i)\delta$, where δ is a small increment step (2mm) and $t(p_i)$ the current growing direction. To avoid drifting from the point cloud during the integration, we apply moving least squares to project p'_i onto some point

p_i'' within the point set surface defined by the point cloud. The growing is terminated, in either growing direction, if any of the following applies:

1. Incompatible growing direction: $|t(p_i'') \cdot t(p_i)| < T_t$, where $t(p_i'') = \mathrm{sgn}\big(t(p_i) \cdot o(p_i'')\big)\, o(p_i'')$ is the next growing direction from p_i'' consistent with $t(p_i)$.
2. Being in holes or out of boundary: $|\mathcal{N}(p_i'')| < T_{\mathcal{N}}$.
3. Unreliable MLS projection: $|\langle p_i'' - p_i' \rangle \cdot t(p_i)| > T_{\mathrm{MLS}}$. This happens where the estimated normals are unreliable.

We set $T_t = 0.9$, $T_{\mathcal{N}} = 5$ and $T_{\mathrm{MLS}} = 0.5$ in all our examples. The next growing vertex is obtained by $p_{i+1} = p_i''$ if none of these termination conditions apply.

Note that in the computation of the next orientation $o(p_i'')$ (Equation 5.2), we on-the-fly select $\mathcal{N}(p_i'')$ with orientations compatible with the current growing direction $t(p_i)$, i.e., $q \in \mathcal{N}(p_i'')$ only if $|o_q \cdot t(p_i)| > T_o$. This scheme avoids the orientational ambiguities at the crossings of multiple hair wisps with different orientations during strand growing. In contrast, many previous methods [58, 14] precompute a diffused 3D orientation field, and the orientations at wisp crossings are problematic.

Smoothing. The initial grown strand may be noisy because of the MLS projection step and the noise in the input point cloud. We therefore smooth the strand by minimizing the following energy:

$$\mathcal{E} = \sum_i \alpha_0 \big\| p_i - p_i^{(0)} \big\|^2 + \alpha_1 \big\| p_{i+1} - p_i - t(p_i^{(0)})\delta \big\|^2 + \alpha_2 \big\| p_{i-1} - 2p_i + p_{i+1} \big\|^2, \tag{5.3}$$

where p_i is a vertex on a strand, $p_i^{(0)}$ is the initial position of p_i before optimization, p_{i-1} and p_{i+1} are the predecessor and successor of p_i on the strand and α_0, α_1 and α_2 are weights that control positional, orientational, and curvature energy terms. We set $\alpha_0 = 0.1$, $\alpha_1 = 1$ and $\alpha_2 = 5$. (assuming the point cloud is in millimeters).

Covering. After a strand S is grown and smoothed, we remove the seed points that S covers from the original seed points to avoid repeated growing. To find if a seed point q is covered by a strand, we traverse every line segment (p_i, p_{i+1}) in the strand and compute the distance between q and (p_i, p_{i+1}) to find the minimum distance. If the minimum distance is smaller than a designated threshold Δ (set to 1.5mm), q is covered by the strand.

We repeat the growing, smoothing and covering steps above until all the seed points are covered.

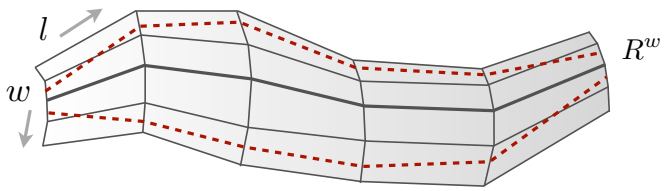


Figure 5.5: A ribbon $\mathcal{R}$ is a grid of 3D vertices. The two parametric directions $w \in [0, W]$ and $l \in [0, L]$ are shown with arrows. Isocurve R^w goes along the l direction (bold line). The ranges $g^-(l), g^+(l)$ of the ribbon is shown in red dashed lines (Sec. 5.7.3).

Plausibility check. The strand growing may cause implausible strand segments, such as U-shaped strands with low turning points. To avoid this, we compute a height value $H(p) = p \cdot d_{down}$ for each point p in S, where d_{down} is a reference down direction, and check each height difference ΔH of every pair of consecutive local extrema. If the height difference $\Delta H > T_H$ (T_H is set to 50mm), then we split S at the extrema points. Note that the connections between these split strand segments will be re-discovered in the connection analysis (Sec. 5.6.1) for the subsequent connection graph (Sec. 5.6.2).

5.5.2 Covering by Ribbons

Intuitively, our goal is to group nearly-parallel strand segments into ribbons, which will later be connected into complete wisps. Thus, the coherence present in our final output is a function of the coherence we are able to find in the strand-to-ribbon grouping stage. We proceed greedily, by always working with the currently longest strand segment that has not been covered by a ribbon.

Ribbon. Formally, a ribbon $\mathcal{R}$ is a 2D grid of 3D vertices $\{R_i^j\}$ where $i = 0, 1, \ldots, L$ and $j = 0, 1, \ldots, W$. L is the length of the ribbon and W the width. The isocurves R^j along the length define the orientation of the ribbon. Isocurve $R^{W/2}$ is defined as the center isocurve of the ribbon (Figure 5.5).

After tessellating the grid of the ribbon, we can use $\mathcal{R}$ to define a local parameterization $R(w, l) : [0, W] \times [0, L] \mapsto \mathbb{R}^3$, where $w \in [0, W]$ and $l \in [0, L]$. Also, the inverse projection operator $R^{-1}(p) : \mathbb{R}^3 \mapsto [0, W] \times [0, L]$ can be defined for any point p by finding the closest point q on the tessellated ribbon and mapping back to the parametric domain.

We can define the orientation o_R of each vertex on the ribbon as $o_R(R_i^j) = \langle R_{i+1}^j - R_{i-1}^j \rangle$ and the normal as $n_R(R_i^j) = \langle (R_i^{j+1} - R_i^{j-1}) \times (R_{i+1}^j - R_{i-1}^j) \rangle$. These definitions can be extended to all the points on the ribbon by linear interpolation.

Expansion. Starting from a strand segment S, we add S to $\mathcal{R}$ as the first isocurve R^0, then we expand the width of $\mathcal{R}$ on both sides and fit to the input point cloud. The initial expansion offsets are computed as $b(R_i^0) = \pm o_R(R_i^0) \times n(R_i^0)$, where $n(R_i^0)$ is the normal at point R_i^0. $n(R_i^0)$ can be computed as the weighted average of the normals of the neighboring points $\mathcal{N}(R_i^0)$. We then initialize a new isocurve R' from R^0 by: $R'_i = R_i^0 + b(S_i)\Delta$. We apply MLS projection to every point of R' on the input point cloud and smooth R' as in Sec. 5.5.1.

Covering. Now we try to cover more adjacent strand segments with the expanded ribbon (Figure 5.4). For each adjacent strand segment S, we project each point p_i of S onto $\mathcal{R}$ as $q = R^{-1}(p_i)$ and classify p_i as a bad point if any of following applies:

1. Incompatible orientation: $|o_S(p_i) \cdot o_R(q)| < T_t$, where $o_S(p_i) = \langle p_{i+1} - p_{i-1} \rangle$ is the strand orientation at p_i.
2. Too far away: $\|p_i - q\| > \Delta$.

S is covered by $\mathcal{R}$ if and only if the number of bad points is less than T_{bad} (set to 10).

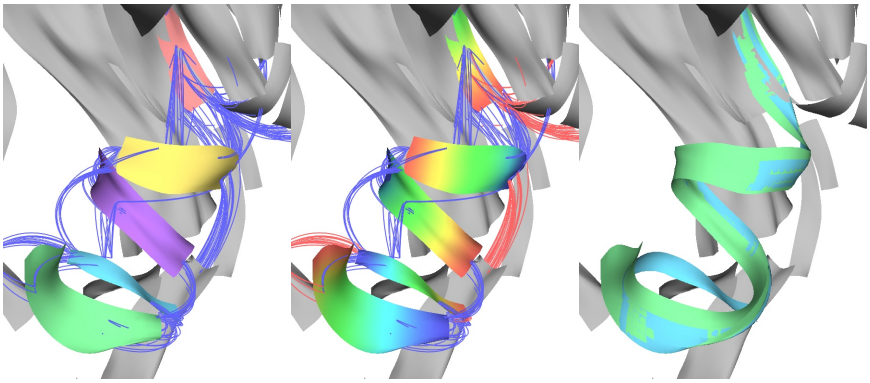
We continue to expand $\mathcal{R}$ if there are new strand segments covered by $\mathcal{R}$. Otherwise, we mark this expansion as failed. If we have 2 consecutive failed expansions, we terminate the expansion on the current side and try the other side if it has not been expanded.

Note that when the expansion on one side is terminated, we may have excessive expanded isocurves for covering the strand segments. We then repeatedly remove the outermost isocurves on both sides from $\mathcal{R}$ given that the set of covered strand segments are unaffected by the removal. Also, after the first expansion, we can replace $n(R_i^j)$ with $n_R(R_i^j)$ to evaluate the expansion offset $b(R_i^j)$.

5.6 Connection and Direction Analysis

So far, our bottom-up analysis of the hairstyle has proceeded by local agglomeration. However, occlusions and missing data force us to look for more distant connections between ribbons (Sec. 5.6.1). These connections, however, may be less reliable, and in fact may be inconsistent with each other.

This inconsistency becomes apparent when we attempt to assign a *direction* to each ribbon: which way strands should traverse the ribbon as they go from scalp to tip. This direction will be necessary for final strand synthesis (Sec. 5.7.3), yet is difficult or impossible to infer locally from the input, especially for challenging hairstyles in which hair strands can go in all directions such



(a) (b) (c)

Figure 5.6: Connection and direction analysis on curly ribbons. We find possible missing connections between the ribbons by fitting circular arcs to the covered strands as shown in blue arcs (a). The direction analysis is performed to determine the directions of the ribbons in (b) (from blue to red). Notice that the resulting incompatible links are colored in red. Finally ribbons are connected up to form wisps (c). Note that the upper overlapped wisps are removed.

as the messy hairstyle illustrated in Figure 5.16 and the curly hairstyle with helical wisp structures illustrated in Figure 5.6.

Therefore, we encode the hypothesized ribbon connections in a graph, and perform a global direction analysis (Sec. 5.6.2) to discover the most consistent direction assignments. We drop any connections that are inconsistent with the assigned directions, and connect the ribbons into our final wisps (Sec. 5.6.3) as illustrated in Figure 5.6.

5.6.1 Connection Analysis

We analyze the possible connections between two ribbons by fitting circular arcs between the strand segments covered by these ribbons. We use circular arcs as the fitting model because hair strands exhibit low variation in curvature, as suggested by the hair simulation model [5]. Also, efficient methods [15] exist to fit circles robustly in the presence of noise and missing data. Note that one could use helix fitting [64] to explicitly account for hair torsion in the fitting model, however we found that helix fitting is much more expensive to compute and tends to overfit for noisy data.

Thus, we test each pair of adjacent strand segments S and S' covered by $\mathcal{R}$ and $\mathcal{R}'$, respectively, and having two end vertices p and p' within a distance of 30mm. We extract K (set to 10) closest vertices on S and S' to p and p' denoted as $\mathcal{P} = \{p_1, \ldots, p_K\}$ and $\mathcal{P}' = \{p'_1, \ldots, p'_K\}$ ($p = p_1$ and $p' = p'_1$), from the closest to the farthest. We also denote $\mathcal{Q} = \{q_i\} = \{p_K, \ldots, p_1, p'_1, \ldots, p'_K\}$ as the concatenation of the input points. We adopt a method in [15] to fit a 3D circle to $\mathcal{Q}$. First, we fit a plane to these points so that we can project the points on the plane and apply robust 2D circle

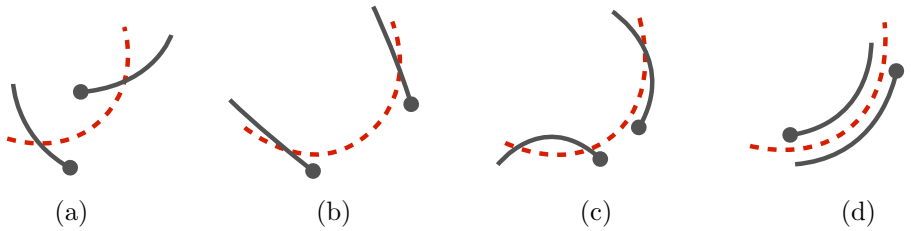


Figure 5.7: Rejection criteria in connection analysis. Adjacent strand segments (solid curves) with end vertices (dots) are fit with circular arcs (dashed curves). (a) Large fitting error. (b) Incompatible curvatures. (c) Large torsion. (d) Large overlap.

fitting methods to initialize the fit. We find that Taubin's algebraic fit [70] works well even in the cases with very small curvature. We then refine the circle center $\hat{c}$, normal $\hat{n}$ and radius $\hat{r}$ jointly using Levenberg-Marquardt.

We use a set of criteria to reject bad fits (Figure 5.7):

(a) Large fitting error. We reject the fit if the Root Mean Square (RMS) fitting error is larger than 2mm.

(b) Incompatible curvatures. We also compute the curvatures κ and κ' of $\mathcal{P}$ and $\mathcal{P}'$ by fitting circles and compare with the curvature $\hat{\kappa}$ of the fit circle. We reject the fitting if $|\kappa-\kappa'|/|\kappa+\kappa'| > T_{curv}$ or $|2\hat{\kappa}-\kappa-\kappa'|/|\kappa+\kappa'| > T_{curv}$, where T_{curv} is set to 0.4.

(c) Large torsion. We reject the fitting if the angle between any two of the normals n_1, n_2 and $\hat{n}$ is larger than 90 degrees, where n_1 and n_2 are the normals of the fit circles to $\mathcal{P}$ and $\mathcal{P}'$.

(d) Large overlap of the input points. We compute the overlap of the input points as $\eta = 1-|V/V_0|$, where the signed sweeping volume $V = \sum_i (q_{i+1} - \hat{c}) \times (q_i - \hat{c}) \cdot \hat{n}$ and the unsigned sweeping volume $V_0 = \sum_i |(q_{i+1} - \hat{c}) \times (q_i - \hat{c}) \cdot \hat{n}|$. We reject the fitting if $\eta > 0.1$.

If we accept the fitting, we project p and p' onto $\mathcal{R}$ and $\mathcal{R}'$ and define the pair of projected points $(R^{-1}(p), R'^{-1}(p'))$ as the *link points* of a *link* ℓ. We refer to the link points $R^{-1}(p)$ and $R'^{-1}(p')$ as $\ell(\mathcal{R})$ and $\ell(\mathcal{R}')$. See example fitting results in Figure 5.6.

5.6.2 Direction Analysis

Connection graph. The major data structure used in direction analysis is the connection graph (Figure 5.8). The connection graph consists of ribbons as vertices and connections between ribbons as edges. Each connection contains one or more links derived from the connection analysis (Sec. 5.6.1).

The direction $D(\mathcal{R})$ of a ribbon $\mathcal{R}$ is either consistent with the parametric direction along the length $0 \to L$ or the opposite $L \to 0$. We denote that a strand segment $S = \{p_1, \ldots, p_n\}$ is

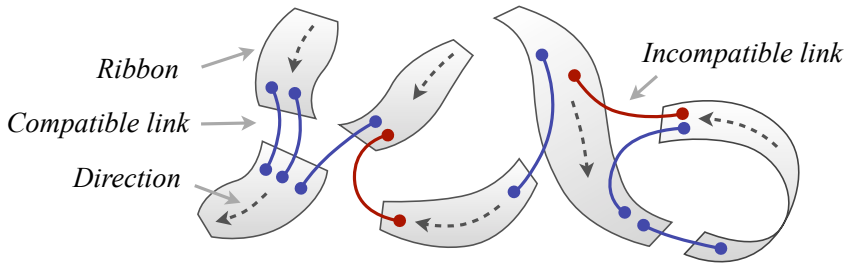


Figure 5.8: A connection graph consists of ribbons connected by links. The ribbons are assigned with directions (dashed line). The links can be compatible (blue) or incompatible (red) with the directions of the incident ribbons.

compatible with the ribbon's direction $D(\mathcal{R})$ as $S \sim D(\mathcal{R})$, if the following expression is *false* for the majority of the vertices in S: $o_S(p) \cdot o_R(R^{-1}(p)) > 0 \oplus D(\mathcal{R}) = 0 \rightarrow L$, where p is a vertex of S and $\oplus$ the *exclusive or* operator.

We define that a link ℓ between $\mathcal{R}$ and $\mathcal{R}'$ is compatible with $D(\mathcal{R})$ and $D(\mathcal{R}')$ if the following expression is *true* for strand segments $\mathcal{P} = \{p_1, ..., p_K\}$ in $\mathcal{R}$ and $\mathcal{P}' = \{p'_1, ..., p'_K\}$ in $\mathcal{R}'$ (Sec. 5.6.1) used to fit ℓ: $\mathcal{P} \sim D(\mathcal{R}) \oplus \mathcal{P}' \sim D(\mathcal{R}')$. Intuitively, a link is compatible with the directions of two ribbons if a strand can grow from one ribbon to the other through the link without having an incompatible direction at the other, as illustrated in Figure 5.8.

We can then define the set of compatible links between $\mathcal{R}$ and $\mathcal{R}'$ as $\mathcal{C}(\mathcal{R}, \mathcal{R}')$ and the set of incompatible links as $\overline{\mathcal{C}}(\mathcal{R}, \mathcal{R}')$.

The strength $\Psi(\mathcal{L})$ of a set of links $\mathcal{L}$ between two ribbons $\mathcal{R}$ and $\mathcal{R}'$ is defined as the smaller one of the numbers of different strand segments in $\mathcal{R}$ and $\mathcal{R}'$ used to fit the links in $\mathcal{L}$. Note that this definition down-weights the outlier case where only one or a few strand segments in a ribbon are connected by many strand segments in another ribbon, as often occurs in practice.

For ribbons that overlap each other, we define the overlap $\eta(\mathcal{R}, \mathcal{R}')$ of $\mathcal{R}$ and $\mathcal{R}'$ as the ratio of the number of overlapped vertices over the total number of vertices in $\mathcal{R}$ and $\mathcal{R}'$. A ribbon vertex p is overlapped with a vertex p' in another ribbon if $\|p - p'\| < \Delta$. The directions of $\mathcal{R}$ and $\mathcal{R}'$ at p and p' are compatible if the following expression is *false*: $o_R(p) \cdot o_{R'}(p') > 0 \oplus D(\mathcal{R}) = D(\mathcal{R}')$. We denote that $D(\mathcal{R})$ is compatible with $D(\mathcal{R}')$ as $D(\mathcal{R}) \sim D(\mathcal{R}')$ if the directions are compatible at the majority of the overlapped vertices or as incompatible $D(\mathcal{R}) \nsim D(\mathcal{R}')$ otherwise.

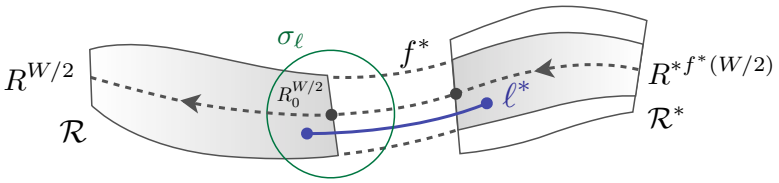


Figure 5.9: A ribbon $\mathcal{R}$ is connected up to a predecessor ribbon $\mathcal{R}^*$. The feasible link ℓ^* that minimizes the mapping distortion in Equation (5.10) within the distance σ_ℓ of vertex $R_0^{W/2}$ defines a stitching mapping f^* to map each isocurve of $\mathcal{R}$ to that of $\mathcal{R}^*$. In particular, $\mathcal{R}$'s center isocurve $R^{W/2}$ is mapped to $R^{*f^*(W/2)}$.

MRF formulation. We use a Markov Random Field (MRF) to optimize the set of directions $\mathcal{D}$ for all the ribbons by minimizing the following energy:

$$E = \sum_{\mathcal{R}} E_{\text{ribb}}(\mathcal{R}) + \sum_{\mathcal{R},\mathcal{R}'} \alpha_{\text{link}} E_{\text{link}}(\mathcal{R},\mathcal{R}') + \alpha_{\text{close}} E_{\text{close}}(\mathcal{R},\mathcal{R}'). \tag{5.4}$$

The ribbon energy E_{ribb} accounts for the fact that the ribbon's direction is more likely to be falling down due to gravity or going farther away from scalp since it originates from the scalp.

$$E_{\text{ribb}}(\mathcal{R}) = \frac{1}{2h}\Big(H(R_{l_1}^{W/2}) - H(R_{l_2}^{W/2}) + \zeta(R_{l_1}^{W/2}) - \zeta(R_{l_2}^{W/2})\Big), \tag{5.5}$$

where H is the height function defined in Sec. 5.5.1, $\zeta(p)$ is the distance from a point p to the closest point on the scalp (please refer to Sec. 5.7.1), $l_1 = 0$ and $l_2 = L$ if $D(\mathcal{R}) = 0 \to L$ and $l_1 = L$ and $l_2 = 0$ if $D(\mathcal{R}) = L \to 0$. h is a height threshold (set to 100mm) to adjust the sensitivity of ribbon's direction to its height difference or the scalp distance difference.

The link energy E_{link} minimizes incompatible links:

$$E_{\text{link}}(\mathcal{R},\mathcal{R}') = \frac{\Psi\big(\overline{\mathcal{C}}(\mathcal{R},\mathcal{R}')\big)}{\max_{\mathcal{R},\mathcal{R}'} \Psi\big(\overline{\mathcal{C}}(\mathcal{R},\mathcal{R}')\big)}. \tag{5.6}$$

Finally, the closeness energy E_{close} penalizes incompatible directions between overlapped ribbons:

$$E_{\text{close}}(\mathcal{R},\mathcal{R}') = \begin{cases} 0, & D(\mathcal{R}) \sim D(\mathcal{R}') \\ \eta(\mathcal{R},\mathcal{R}'), & D(\mathcal{R}) \nsim D(\mathcal{R}') \end{cases}. \tag{5.7}$$

The total energy E can be effectively minimized using the method in [11].

5.6.3 Connecting Ribbons into Wisps

After we determine the ribbon directions, we connect the ribbons into wisps following the compatible links. For the sake of brevity, we now assume that each ribbon $\mathcal{R}$ is consistent to its optimized direction, i.e., $D(\mathcal{R}) = 0 \rightarrow L$.

For each ribbon $\mathcal{R}$, we connect "up" from the beginning ($l = 0$) of the ribbon to the end ($l = L$) of a *predecessor* ribbon $\mathcal{R}^*$. To simplify the problem, we use the center isocurve $R^{W/2}$ as the delegate to the ribbon when we are looking for the predecessor ribbon to connect up (Figure 5.9).

We first define a *feasible link* that can be used to connect the beginning of $R^{W/2}$ to another ribbon. We require the feasible link to be close enough to $R_0^{W/2}$. To be specific, a link ℓ is *feasible* for $\mathcal{R}$ if $\|\ell(\mathcal{R}) - R_0^{W/2}\| < \sigma_\ell$, where σ_ℓ is a distance threshold (10mm) to define closeness between the link point and the center isocurve. We can then define the set of feasible links between $\mathcal{R}$ and $\mathcal{R}'$ as $\mathcal{F}(\mathcal{R}, \mathcal{R}')$. The predecessor ribbon $\mathcal{R}^*$ is the ribbon that maximizes the strength of compatible and feasible links between $\mathcal{R}$ and $\mathcal{R}^*$:

$$\mathcal{R}^* = \arg\max_{\mathcal{R}'} \Psi\big(\mathcal{F}(\mathcal{R}, \mathcal{R}') \cap \mathcal{C}(\mathcal{R}, \mathcal{R}')\big). \tag{5.8}$$

Given a feasible link $\ell \in \mathcal{F}(\mathcal{R}, \mathcal{R}^*) \cap \mathcal{C}(\mathcal{R}, \mathcal{R}^*)$, we can define a stitching mapping $f^\ell : [0, W] \mapsto [0, W^*]$, where $[0, W]$ is the width domain of $\mathcal{R}$ and $[0, W^*]$ of $\mathcal{R}^*$ as follows:

$$f^\ell(w) = \min(\max(w - \ell(\mathcal{R}).w + \ell(\mathcal{R}^*).w, 0), W^*), \tag{5.9}$$

where $\ell(\mathcal{R}).w$ is the w parameter of $\ell(\mathcal{R})$ and $\ell(\mathcal{R}^*).w$ of $\ell(\mathcal{R}^*)$.

Finally, we choose the link $\ell^* \in \mathcal{F}(\mathcal{R}, \mathcal{R}^*) \cap \mathcal{C}(\mathcal{R}, \mathcal{R}^*)$ for the stitching mapping f^* that minimizes the mapping distortion:

$$\ell^* = \arg\min_{\ell} \left(1 - \frac{f^\ell(W) - f^\ell(0)}{W}\right). \tag{5.10}$$

Using f^* we can extend every isocurve $R^w \in \mathcal{R}$ to $R^{*f^*(w)} \in \mathcal{R}^*$. To generate a smooth transition between the two isocurves, we first connect them with a straight line using a discretization step of δ, then we perform the smoothing step described in Sec. 5.5.1 according to the orientation field. Note that during the smoothing, we fix the two end points of the line and use a larger σ_d to compute the orientation since the transition region lacks data points.

We replace the center isocurve $R^{W/2}$ with $R^{*f^*(W/2)}$ and repeat the connection process above until we cannot find any predecessor ribbons to connect.

However, in order to maximize the chance of finding a predecessor ribbon and connecting up, when no predecessor ribbons can be found directly from $\mathcal{R}$, we also look for feasible links in the adjacent overlapped ribbons $\mathcal{R}'$ with $\eta(\mathcal{R}, \mathcal{R}') > 0$. We compute the R^* and f^* with respect to $\mathcal{R}'$ but change the definition of a feasible link to any link ℓ that $\|\ell(\mathcal{R}') - R_0^{W/2}\| < \sigma_\ell$ as well as replacing the role of $\mathcal{R}'$ with $\mathcal{R}$ in Equations 5.9 and 5.10.

To avoid cycles when connecting up, we store all the ribbons connected and avoid connecting to them again. Also, we try to avoid local cycles by rejecting connections to a overlapped ribbon $\mathcal{R}'$ with $\eta(\mathcal{R}, \mathcal{R}') > 0.1$

After all the ribbons are connected up into wisps, we remove all wisps that are covered by others. To see if one ribbon is covered, we check if every vertex of the wisp is overlapped by the vertices of other wisps using the method in Sec. 5.6.2. We also remove outlier ribbons that fail to connect to at least one other ribbon, which often arise from outlier points of the initial point cloud.

5.7 Synthesis

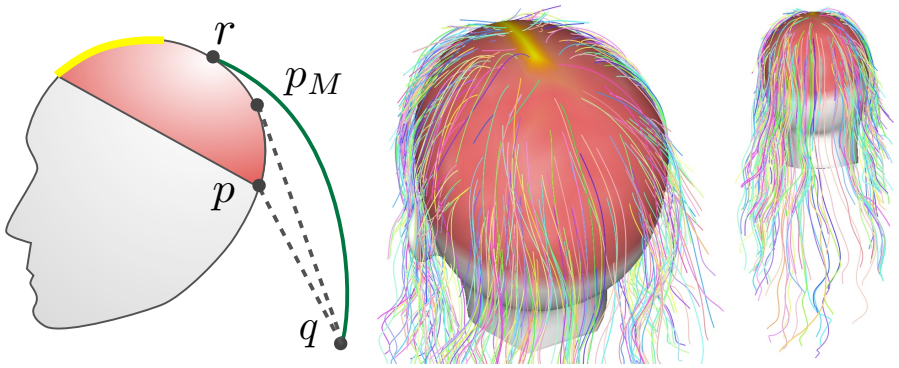
Once we have found the wisps, we attach them to the scalp. Since the interior of the hair is occluded, and hence we have no data there, we need to generate interior wisps to fill in the empty region. Finally, we use the attached exterior wisps and interior wisps to generate plausible strands according to the input point cloud and orientation field.

5.7.1 Attaching Wisps to the Scalp

Head model and scalp. We manually fit a head model to each dataset and paint on the model to indicate the possible scalp region. We also paint a parting line to specify the vertices on the scalp that parts the hair into different directions according to the captured hairstyle (Figure 5.10). Painting on the model can be done in minutes using any 3D mesh texture painting software and the scalp region is reused for all the examples.

Hair growing directions. Human hair has natural growing directions perpendicular to the radial line from the hair whorl on the scalp. However, in practice it is very difficult to determine the growing directions for non-trivial hairstyles. Existing methods either simplify the growing directions to normal directions on the scalp [86] or specify the growing directions manually [69].

We notice the fact that most common hairstyles have a distinctive parting line to part the hair into different growing directions. We compute the reference growing directions with the drawn



(a) (b) (c)

Figure 5.10: (a) Attaching a point q to the scalp (the red area). p is the closest point to q on the scalp and p_M is a point on p's closest path to the parting line (the yellow line) that maximizes the directional compatibility. r is chosen as the root point to attach and generate the interior strand (green curve). A real attaching example with internal strands is shown in (b) with a zoom-out view in (c).

parting line on the scalp. We first compute shortest paths from the vertices on the parting line to every vertex p on the scalp. For each p we denote the shortest path from the parting line to p as $\Gamma(p) = \{p_1, p_2, ..., p_N\}$, where p_1 is the vertex on the parting line and $p_N = p$. Then the reference growing direction $d_s(p)$ at p is computed as $d_s = p_N - p_{N-1}$. We can compute the reference growing directions at every point on the scalp by interpolation.

Root point assignment. To attach a wisp $\mathcal{R}$ to the scalp, we first need to find a root point r and a growing direction d_g on the scalp for $\mathcal{R}$. Here we again use the center isocurve $R^{W/2}$ as the delegate for $\mathcal{R}$ to simplify the problem. For notational simplicity, we denote the vertex $R_1^{W/2}$ at the beginning of $\mathcal{R}$ as the attaching point q. We find the closest scalp vertex p for q and search along p's shortest path from the parting line $\Gamma(p) = \{p_1, p_2, ..., p_N\}$ and find p_M with the maximum directional compatibility:

$$p_M = \arg \max_{p_i \in \Gamma(p)} \langle q - p_i \rangle \cdot d_s(p_i). \tag{5.11}$$

To avoid cluttering of the root points we randomly select a point p_k from $\{p_1, p_2, ...p_M\}$ and sample the root point r around a neighborhood of p_k on the scalp. The final growing direction d_g is computed as:

$$d_g = \begin{cases} d_s(r), & n(r) \cdot d_s(r) > n(r) \cdot \langle q - r \rangle \\ \langle q - r \rangle, & n(r) \cdot d_s(r) < n(r) \cdot \langle q - r \rangle, \end{cases} \tag{5.12}$$

where $n(r)$ is the normal at root point r on the scalp. The second case adjusts d_g when the attaching points are above the root points on the scalp.

Interior strand generation. We use a technique similar to that in Sec. 5.6.3 to generate the interior strand between the attaching point $R_0^{W/2}$ at the wisp and the root point r. That is, we first initialize the strand as a straight line and then smooth it. However, it is desirable to ensure that interior strands are hidden in the interior region, so that they have minimum impact on the exterior appearance. To this end, we use the *interior field* to regularize the interior strands in the interior region.

The interior field I is a level set field that smoothly transitions from 0 to 1 from the scalp points to the input data points. Using a regular grid, we first set the voxels containing input points to 1 and the voxels on the scalp to 0. Then we apply diffusion constrained by the values of these set voxels to smoothly generate the interior values between the input points and the scalp. We add the following interior energy term E_{int} to Equation 5.3 for the interior field:

$$E_{\text{int}} = \sum_i \alpha_{\text{int}} B \left(\frac{i}{N} - I(p_i) \right)^2, \tag{5.13}$$

where N is the number of vertices of the interior strand and B the diagonal length of the bounding box of the input point cloud. The weight for interior field α_{int} is set to 0.005. We assume the direction of the interior strand is from the scalp to the input data points.

To keep the growing direction on the root point of the interior strand during smoothing with Equation 5.3, we fix two points nearest to the root point, to align the strand with the growing direction.

After we compute and attach the interior strand for the center isocurve $R^{W/2}$, we offset and attach the interior strand to all the other isocurves in $\mathcal{R}$ to form a complete wisp from the scalp.

5.7.2 Interior Wisp Generation

Interior wisps are generated in a similar way as the interior strands in Sec. 5.7.1. Specifically, we sample the points from the input point cloud that have greater distances to the scalp than a specified threshold. For each, we generate an interior strand as in Sec. 5.7.1. To make the strand completely hidden in the hair volume, we truncate the farther end of the curve.

We then expand the strand into a ribbon with a designated width. Note that the initial normals of the ribbon can be derived from the gradient of the interior field ∇I.

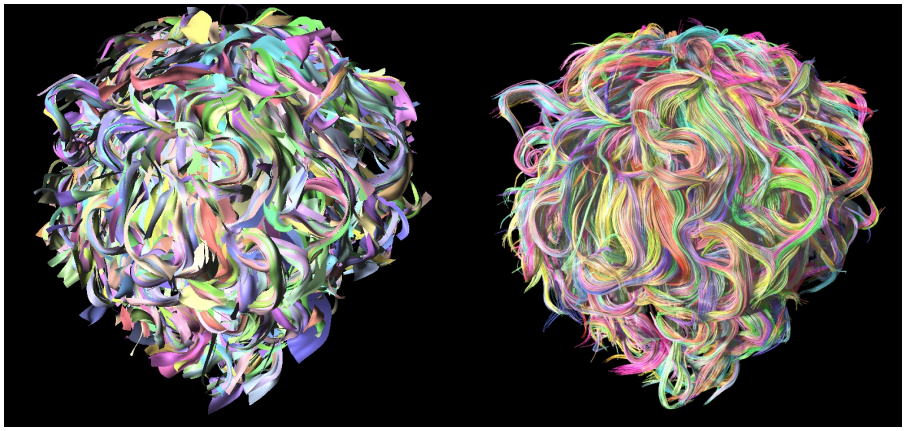


Figure 5.11: The synthesized strands (right) with the wisps (left).

5.7.3 Strand Synthesis

We introduce the *ranges* of a wisp $\mathcal{R}$ to fit the input data more accurately when we synthesize the strands. To be specific, the ranges of $\mathcal{R}$ are two functions $g^-, g^+ : [0, L] \mapsto [0, W]$. g^-, g^+ are the lower and upper bound of the width parameter defined on $[0, L]$.

To compute the ranges of each wisp $\mathcal{R}$ for the input data points, we project each data point p to $\mathcal{R}$ and check if p is covered by $\mathcal{R}$. Uncovered points are not used to compute the ranges of $\mathcal{R}$. We then update the ranges with the covered point p as follows:

$$\begin{aligned} g^-(l_p) &= \min(g^-(l_p), R^{-1}(p).w) \\ g^+(l_p) &= \max(g^+(l_p), R^{-1}(p).w) \end{aligned} \tag{5.14}$$

where $l_p = \lfloor R^{-1}(p).l + 0.5 \rfloor$ is rounded to the nearest integer length parameter. We then smooth $g^-(l), g^+(l)$ by fitting smooth 1D curves to them on $[0, L]$. To improve realism, we also taper the ranges towards the strand tips using a quadratically decreasing function as the tapering factor.

For the interior wisps where no data points can be used to compute the ranges, we simply set the ranges to the maximum: $[0, W]$.

We can interpolate the ranges by defining the range interpolation function $g(l, t) = g^-(l) \cdot (1 - t) + g^+(l) \cdot t$ and a strand S can be synthesized from $\mathcal{R}$ by $R_l^{g(l,t)}$ for all $l \in [0, L]$.

Now S is synthesized on the surface of the wisp (Figure 5.11). To add thickness to the wisps, we offset S in $\mathcal{R}$'s inverse normal direction by a random amount between 0 and T_{thick}. We find that $T_{thick} = 3$mm works well for all the examples in this work. Finally, we perform smoothing on S using Equation 5.3.

Figure 5.12: We use a robotic gantry to position an SLR camera at 50 views to capture the images for the wigs (left). For real hairstyles, we use a camera array of 30 SLR cameras (right).

To control the density of the synthesized strands, we record the number of strands covering the vertices within the ranges of the wisps during strand synthesis. If the number of covering strands for a vertex is smaller than a preset threshold $T_{density}$, the vertex can initiate the synthesis of one new strand covering it from its wisp, otherwise the vertex is skipped. We iterate on each vertex until no vertex can initiate the synthesis of new strands.

5.8 Results

We present results of our pipeline on five datasets. Three of these are of wigs, and were captured using a single digital SLR camera mounted on a motorized spherical gantry and moved to 50 positions (Figure 5.12 left). These results are shown in Figure 5.1 and the first two rows of Figure 5.16. Two other results (Figure 5.16, last two rows) are of real hair, and were captured using a rig containing 30 cameras (Figure 5.12 right).

As shown in Figure 5.1, our system is capable of reconstructing challenging hair styles. Our connection graph is able to establish correct correspondences among the partially-visible portions of curls, and hence our reconstructed wisps are, in most cases, able to follow the curls all the way from the scalp to the tips. Figure 5.13 compares the reconstruction details to the input hairstyle and shows that our reconstructed hair model can faithfully reveal the intricate hair structures in the input hairstyle. Figure 5.16, first row, contains many crossing wisps that present a challenge to competing algorithms, while the second row illustrates a complicated and disordered hairstyle. Though we are not able to reconstruct each curl perfectly, many specific curls are captured, and the general impression of the hairstyle makes our reconstructed data suitable for virtual characters.

Figure 5.13: Close-up comparison of the hair details between the reference hairstyle and our reconstructed hairstyle.

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

Figure 5.14: We evaluate the robustness of our method by reconstructing hairstyles (b) and (d) from the input point clouds of 2M points (a) and 230K points (c), respectively. Notice the little visual difference of our results from the inputs of very different quality.

The third and fourth rows of Figure 5.16 illustrate hair capture for a "digital double" scenario. Though the required acquisition apparatus is nontrivial, it consists solely of digital (still) cameras. Acquisition is thus completely passive and one-shot, meaning that it would be practical to incorporate a hair capture session into a movie special-effects workflow (and budget). These hair styles are simpler than the wigs (which were chosen to illustrate challenging cases), and our reconstruction is relatively accurate. Though we have chosen rendering parameters by hand (for the Marschner et al. [50] hair scattering model) to roughly match the input images, a closer match might be obtained using a method such as that of Bonneel et al. [10]. Of course, situations in which hair is colored or highlighted, such as the fourth row of Figure 5.16, would require even more sophisticated estimation

Figure 5.15: Three frames from a physical simulation in which hair is tousled through rapid head motion. As with most hair simulations, this one uses sparse guide strands (About 1K in this simulation, shown at top), which naturally arise from the reconstructed wisps. The motion of the guide strands is interpolated onto the full set of strands (about 30K shown at bottom).

of hair appearance; we believe that this is an interesting and necessary direction of future work to allow hair capture for digital doubles to become practical.

To evaluate the plausibility of our reconstruction results, we created animations of growing the synthesized strands from the scalp to the tip to visualize the internal coherent topology of the hair model. Please see the accompanying video for the results.

Robustness. We evaluate the robustness of our method using inputs of varying quality. We apply our pipeline on an input point cloud of 230K points, about 10 times less than the full resolution 2M points which we used to reconstruct the hair model shown in Figure 5.1. We compare our reconstructed hair models for two different inputs in Figure 5.14. Note that the result from low quality input faithfully recovers all key hair wisp structures and has little difference on the visual quality compared to the full resolution result.

Simulation. Our wisp-based hair models are plausible for hair simulations. We demonstrate a simulation setup using the center isocurve of each wisp as the guide strands in a hair simulator based on articulated rigid curves. The hair motion is driven by a pre-defined scalp movement and the full motion of the synthesized strands is then interpolated from the guide strands. We use about 1K guide strands to drive around 30K synthesized strands. Three frames of the simulation result are shown in Figure 5.15, and more results can be found in the accompanying video. More advanced simulation-specific processing and computation are important to improve the realism of the simulation result, including pre-tensioning for inverse statics and collision correction for complex hair structures.

Parameter choice. Although our method involves a large number of parameters, we find most of the parameters insensitive to the input datasets and we use the provided fixed values throughout our experiments. One parameter we do find useful to improve the results for specific hairstyles is the curvature weight α_2 for strand smoothing in Equation (5.3). For straight hairstyles, we set $\alpha_2 = 30$ to provide stronger regularization against the stereo noise in the reconstructed point cloud to compensate for the weak regularization nature of PMVS.

Timings. All the examples are computed on a quad-core Intel i7 machine with 16GB memory. For datasets with 50 views, the reconstruction of initial point cloud takes about one hour using PMVS. The computation for 3D orientation field takes 3 minutes. All the subsequent processing steps are implemented single threaded. For an input point cloud with 2M points, the covering step takes 3 minutes. The connection analysis takes 5 minutes and direction analysis 1 minute. The final strand synthesis takes 2 minutes.

5.9 Limitations, Future Work and Conclusion

Although our method can be successfully applied to a variety of challenging hairstyles, the ribbon-based representation and certain smoothness constraints prevent our method from capturing very fine-scale stray hairs or extremely disordered hairstyles (This becomes clear by looking at the result of messy hairstyle in Figure 5.16). The effective amount of regularization, however, could be reduced by starting with a more accurate initial point cloud obtained using one of the recent methods such as [32] or [27]. These more accurate hair reconstruction methods can also help the cases with smooth hairstyles, for which we find larger perceivable reconstruction errors in the point cloud due

Figure 5.16: Examples of our pipeline applied to four datasets. For each, we show two views of the reference input images and the synthesized hairs as well as a color-coded visualization of the reconstructed wisps, where synthesized strands in the same wisp are in the same color.

to increased hair specularity and the weak regularization nature of PMVS. Other improvements to our method might include making our connection and direction analysis more physically based, by including gravity, contact forces, and hair growth models (with estimated stiffness and "curliness" parameters) as priors. Further image-based analysis can also be done to estimate the thickness of the wisps automatically.

Another especially difficult class of hairstyles is those in which the hair does not hang freely but is constrained by braids, dreadlocks, clips, ties, or support against the body. We believe that most existing methods, including the one presented here, would have difficulty in generating topologically-

correct hair strands in these cases. Handling these styles may require coupling the hair acquisition process with physical simulation, and possibly matching to a database of exemplars.

Looking beyond geometry, a full system for hair capture should also include measurement of appearance and motion. As mentioned above, researchers have already investigated the problem of estimating the parameters of hair appearance models, but handling variation from wisp to wisp is likely to require a combination of inverse rendering and data-driven techniques.

Because our system is one-shot, it could be generalized to video input just by running independently on each frame. This is likely to result in flicker, so some method of ensuring temporal coherence would be needed. This may require coupling a hair simulator with the reconstruction system, simultaneously using the data to constrain the simulation and using the simulation to provide temporal coherence and fill in parts of the data that could not be observed.

Overall, the results in Figure 5.16 suggest that our system can generate hair models of the sort needed for current production pipelines. For digital doubles and secondary animated characters, only modest manual editing might be necessary to achieve the required quality. For primary characters, of course, there are considerably greater requirements on quality and controllability, but the captured results may still serve as a reference for hand-modeling by skilled artists.

Chapter 6

Conclusion and Future Work

Hair is a vital component for human appearance in both the real and virtual world. Hair capture is an important approach to acquire 3D hair models from real hairstyles and to significantly reduce the modeling and animating effort of the artists. However, hair capture is unusually challenging due to hair's unconventional characteristics: the view-dependent specular appearance, the geometric complexity and the high variability of real hairstyles.

In this thesis, we first investigate into the idea of using characteristic hair orientations as an matching metric or feature in conventional multi-view stereo systems in Chapter 3 and 4. The orientation proves to be robust to hair's specular highlights compared to color-based metric. More importantly, the unique geometric continuity of hair strands enables effective shape priors for structure-aware aggregation and strand-based refinement that improved the overall reconstruction accuracy and robustness. Using these methods, we are able to reconstruct accurate and detailed surface geometry approximating the captured hair volumes for digital human models in various applications, e.g. 3D printing, gaming and teleconferencing. However, for some other applications such as character animation in film production, much structured and physically plausible hair models are expected. We introduce structure-aware hair capture in Chapter 5 to address this need and reconstructs plausible hair structures on top of the incomplete and unstructured hair geometry (i.e. point cloud) using existing methods. The key idea is that hair is typically grouped into structured wisps, which can be locally discovered and globally completed using a novel connection graph data structure to encode partial wisp directions and their inter-connectivity information.

Nevertheless, there are still much room to improve and extend the current work in the future. The following are a few directions.

Constrained hairstyles. One assumption made in Chapter 5 is unconstrained hairstyles hanging freely from the scalp. This allows the heuristics for connection analysis to work. To extend the current method for constrained hairstyles, physical simulation or pseudo-physics [16] may be used to resolve the direction and connection ambiguities.

Dynamic hair capture. Dynamic hair capture is a very challenging problem merely considering the number of hair strands and the degrees of freedom of hair motion. One promissing approach is still to couple hair dynamics to resolve the ambiguities arising from the occlusion in a similar way to [93, 76]. Also, wisp structures are useful to constrain the motion and control the complexity of the captured model.

Unified capture system. Hair geometry is just one side of the coin. A complete hair capture system should also include the acquisition of hair appearance as done by [10]. Another possible direction is the coupled acquisition of human face and hair. One inspiring work towards such direction can be found in [3].

Bibliography

[1] Yair Adato, Yuriy Vasilyev, Ohad Ben-Shahar, and Todd Zickler. Toward a theory of shape from specular flow. In *ICCV*, pages 1–8, 2007.

[2] Thabo Beeler, Bernd Bickel, Paul Beardsley, Bob Sumner, and Markus Gross. High-quality single-shot capture of facial geometry. *ACM Trans. Graph.*, 29(4), 2010.

[3] Thabo Beeler, Bernd Bickel, Gioacchino Noris, Steve Marschner, Paul Beardsley, Robert W. Sumner, and Markus Gross. Coupled 3d reconstruction of sparse facial hair and skin. *ACM Trans. Graph.*, 31:117:1–117:10, 2012.

[4] Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. Discrete elastic rods. *ACM Trans. Graph.*, 27(3):63:1–63:12, 2008.

[5] Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. Discrete elastic rods. *ACM Trans. Graph.*, 27(3):63:1–63:12, 2008.

[6] Michael J. Black and Anand Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *Int. J. Comput. Vision*, 19(1):57–91, July 1996.

[7] M.J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *Computer Vision, 1993. Proceedings., Fourth International Conference on*, pages 231–236, 1993.

[8] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[9] M. Bleyer, C. Rother, and P. Kohli. Surface stereo with soft segmentation. In *Proceedings of CVPR*, 2010.

[10] Nicolas Bonneel, Sylvain Paris, Michiel Van De Panne, Frédo Durand, and George Drettakis. Single photo estimation of hair appearance. *Computer Graphics Forum (Proc. EGSR)*, 28(4), 2009.

[11] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. PAMI*, 23(11):1222–1239, 2001.

[12] D. Bradley, T. Boubekeur, and W. Heidrich. Accurate multi-view reconstruction using robust binocular stereo and surface meshing. In *Proceedings of CVPR*, 2008.

[13] Brian Carrihill and Robert Hummel. Experiments with the intensity ratio depth sensor. *Computer Vision, Graphics, and Image Processing*, 32(3):337 – 358, 1985.

[14] Menglei Chai, Lvdi Wang, Yanlin Weng, Yizhou Yu, Baining Guo, and Kun Zhou. Single-view hair modeling for portrait manipulation. *ACM Trans. Graph.*, 31(4):116:1–116:8, 2012.

[15] Nikolai Chernov. *Circular and linear regression : fitting circles and lines by least squares*. Monographs on statistics and applied probability. CRC Press/Taylor & Francis, Boca Raton, 2011.

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

[17] I. J. Cox, S. Roy, and S. L. Hingorani. Dynamic histogram warping of image pairs for constant image brightness. In *Proceedings of the 1995 International Conference on Image Processing (Vol.2)-Volume 2 - Volume 2*, ICIP '95, pages 2366–, Washington, DC, USA, 1995. IEEE Computer Society.

[18] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 303–312, New York, NY, USA, 1996. ACM.

[19] Ye Duan and Hong Qin. Intelligent balloon: a subdivision-based deformable model for surface reconstruction of arbitrary topology. In *Proceedings of the sixth ACM symposium on Solid modeling and applications*, SMA '01, pages 47–58, New York, NY, USA, 2001. ACM.

[20] Carlos Hernández Esteban and Francis Schmitt. Silhouette and stereo fusion for 3d object modeling. *Comput. Vis. Image Underst.*, 96(3):367–392, 2004.

[21] Olivier D. Faugeras and Renaud Keriven. Complete dense stereovision using level set methods. In *Proceedings of the 5th European Conference on Computer Vision-Volume I - Volume I*, ECCV '98, pages 379–393, London, UK, UK, 1998. Springer-Verlag.

[22] Yasutaka Furukawa and Jean Ponce. Carved visual hulls for image-based modeling. *Int. J. Comput. Vision*, 81(1):53–67, 2009.

[23] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. PAMI*, 32:1362–1376, 2010.

[24] J. Garding. Direct estimation of shape from texture. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(11):1202–1208, 1993.

[25] D.B. Goldman, B. Curless, A. Hertzmann, and S.M. Seitz. Shape and spatially-varying brdfs from photometric stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(6):1060–1071, 2010.

[26] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. In *ECCV*, 2010.

[27] Tomas Lay Herrera, Arno Zinke, and Andreas Weber. Lighting hair from the inside: A thermal approach to hair reconstruction. *ACM Trans. Graph.*, 31(6):146:1–146:9, 2012.

[28] Li Hong and G. Chen. Segment-based stereo matching using graph cuts. In *Proceedings of CVPR*, 2004.

[29] Berthold K. P. Horn and Michael J. Brooks. The variational approach to shape from shading. *Comput. Vision Graph. Image Process.*, 33(2):174–208, February 1986.

[30] Peisen S. Huang, Chengping Zhang, and Fu-Pen Chiang. High-speed 3-d shape measurement based on digital fringe projection. *Optical Engineering*, 42(1):163–168, 2003.

[31] Hiroshi Ishikawa and Davi Geiger. Rethinking the prior model for stereo. In *Proceedings of the 9th European conference on Computer Vision - Volume Part III*, ECCV'06, pages 526–537, Berlin, Heidelberg, 2006. Springer-Verlag.

[32] Wenzel Jakob, Jonathan T. Moon, and Steve Marschner. Capturing hair assemblies fiber by fiber. *ACM Trans. Graph.*, 28(5):164:1–164:9, 2009.

[33] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[34] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proc. SGP*, 2006.

[35] Tae-Yong Kim and Ulrich Neumann. Interactive multiresolution hair modeling and editing. *ACM Trans. Graph.*, 21(3):620–629, 2002.

[36] J. J. Koenderink. What does the occluding contour tell us about solid shape. *Perception*, 13, 1987.

[37] Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. In *Proceedings of the 7th European Conference on Computer Vision-Part III*, ECCV '02, pages 82–96, London, UK, UK, 2002. Springer-Verlag.

[38] Eric Krotkov. Focusing. *International Journal of Computer Vision*, 1(3):223–237, 1988.

[39] Kiriakos N. Kutulakos and Steven M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000.

[40] David Levin. The approximation power of moving least-squares. *Mathematics of Computation*, 67(224):1517–1531, 1998.

[41] Maxime Lhuillier and Long Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Trans. PAMI*, 27(3):418–433, 2005.

[42] Guo Li, Ligang Liu, Hanlin Zheng, and Niloy J. Mitra. Analysis, reconstruction and manipulation using arterial snakes. *ACM Trans. Graph.*, 29(5):152:1–152:10, 2010.

[43] Hao Li, Bart Adams, Leonidas J. Guibas, and Mark Pauly. Robust single-view geometry and motion reconstruction. *ACM Trans. Graph.*, 28(5):175:1–175:10, 2009.

[44] Yotam Livny, Feilong Yan, Matt Olson, Baoquan Chen, Hao Zhang, and Jihad El-sana. Automatic reconstruction of tree skeletal structures from point clouds. *ACM Trans. Graph.*, 29:151:1–151:8, 2010.

[45] David G. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, 1999.

[46] Linjie Luo, Hao Li, Sylvain Paris, Thibaut Weise, Mark Pauly, and Szymon Rusinkiewicz. Multi-view hair capture using orientation fields. In *Proc. CVPR*, 2012.

[47] Linjie Luo, Hao Li, and Szymon Rusinkiewicz. Structure-aware hair capture. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 32(4), July 2013.

[48] Linjie Luo, Cha Zhang, Zhengyou Zhang, and Szymon Rusinkiewicz. Wide-baseline hair capture using strand-based refinement. In *Proc. CVPR*, 2013.

[49] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2), 1963.

[50] S. Marschner, H. Wann Jensen, M. Cammarano andS. Worley, and P. Hanrahan. Light scattering from human hair fibers. *ACM Trans. Graph.*, 22(3):780–791, 2003.

[51] 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.

[52] Ravish Mehra, Pushkar Tripathi, Alla Sheffer, and Niloy J. Mitra. Visibility of noisy point cloud data. *Computers and Graphics*, 34(3):219–230, 2010.

[53] Microsoft. Kinect, 2010.

[54] Tadao Mihashi, Christina Tempelaar-Lietz, and George Borshukov. Generating realistic human hair for *The Matrix Reloaded*. In *ACM SIGGRAPH Sketches and Applications Program*, 2003.

[55] Liangliang Nan, Andrei Sharf, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. SmartBoxes for interactive urban reconstruction. *ACM Trans. Graph.*, 29(4):93:1–93:10, 2010.

[56] M. Okutomi and T. Kanade. A locally adaptive window for signal matching. In *Computer Vision, 1990. Proceedings, Third International Conference on*, pages 190–199, 1990.

[57] Sylvain Paris, Hector Briceño, and François Sillion. Capture of hair geometry from multiple images. *ACM Trans. Graph.*, 23(3):712–719, 2004.

[58] Sylvain Paris, Will Chang, Oleg I. Kozhushnyan, Wojciech Jarosz, Wojciech Matusik, Matthias Zwicker, and Frédo Durand. Hair Photobooth: Geometric and photometric acquisition of real hairstyles. *ACM Trans. Graph.*, 27(3):30:1–30:9, 2008.

[59] Alex Paul Pentland. A new sense for depth of field. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-9(4):523–531, 1987.

[60] Christoph Rhemann, Asmaa Hosni, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *CVPR*, 2011.

[61] Christian Richardt, Douglas Orr, Ian Davies, Antonio Criminisi, and Neil A. Dodgson. Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid. In *Proceedings of the 11th European conference on computer vision conference on Computer vision: Part III*, ECCV'10, pages 510–523, Berlin, Heidelberg, 2010. Springer-Verlag.

[62] Szymon Rusinkiewicz, Olaf Hall-Holt, and Marc Levoy. Real-time 3d model acquisition. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '02, pages 438–446, New York, NY, USA, 2002. ACM.

[63] Kan'ya Sasaki, Seiji Kameda, Hiroshi Ando, Mamoru Sasaki, and Atsushi Iwata. Stereo matching algorithm using a weighted average of costs aggregated by various window sizes. In *Proceedings of ACCV*, 2006.

[64] Peter Savadjiev, Jennifer S.W. Campbell, G. Bruce Pike, and Kaleem Siddiqi. 3d curve inference for diffusion mri regularization and fibre tractography. *Medical Image Analysis*, 10(5):799–813, 2006.

[65] Silvio Savarese, Marco Andreetto, Holly Rushmeier, Fausto Bernardini, and Pietro Perona. 3d reconstruction by shadow carving: Theory and practical evaluation. *Int. J. Comput. Vision*, 71(3):305–336, March 2007.

[66] D. Scharstein. Matching images by comparing their gradient fields. In *Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision amp; Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, pages 572–575 vol.1, 1994.

[67] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3):7–42, 2002.

[68] Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings of CVPR*, 2006.

[69] G. Sobottka, M. Kusak, and A. Weber. In *Proc. CGIV*, pages 365–371, july 2006.

[70] G. Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Trans. PAMI*, 13(11):1115–1138, 1991.

[71] E. Tola, V.Lepetit, and P. Fua. A Fast Local Descriptor for Dense Matching. In *Proc. CVPR*, 2008.

[72] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[73] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *BMVC*, 2000.

[74] Daniel Vlasic, Pieter Peers, Ilya Baran, Paul Debevec, Jovan Popović, Szymon Rusinkiewicz, and Wojciech Matusik. Dynamic shape capture using multi-view photometric stereo. *ACM Trans. Graph.*, 28(5):174:1–174:11, 2009.

[75] G. Vogiatzis, P. H S Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 391–398 vol. 2, 2005.

[76] Huamin Wang, Miao Liao, Qing Zhang, Ruigang Yang, and Greg Turk. Physically guided liquid surface modeling from videos. *ACM Trans. Graph.*, 28(3):90:1–90:11, 2009.

[77] Lvdi Wang, Yizhou Yu, Kun Zhou, and Baining Guo. Example-based hair geometry synthesis. *ACM Trans. Graph.*, 28(3):56:1–56:9, 2009.

[78] Kelly Ward, Florence Bertails, Tae yong Kim, Stephen R. Marschner, Marie paule Cani, and Ming C. Lin. A survey on hair modeling: Styling, simulation, and rendering. *TVCG*, 13(2):213–234, 2006.

[79] Kelly Ward, Ming C. Lin, Joohi Lee, Susan Fisher, and Dean Macri. Modeling hair using level-of-detail representations. In *Proc. CASA*, page p. 41.

[80] Yichen Wei, Eyal Ofek, Long Quan, and Heung-Yeung Shum. Modeling hair from multiple views. *ACM Trans. Graph.*, 24(3):816–820, 2005.

[81] T. Weise, B. Leibe, and L. Van Gool. Fast 3d scanning with automatic motion compensation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*, June 2007.

[82] O.J. Woodford, P.H.S. Torr, I.D. Reid, and A.W. Fitzgibbon. Global stereo reconstruction under second order smoothness priors. In *Proc. CVPR*, 2008.

[83] Robert J. Woodham. Photometric method for determining surface orientation from multiple images. In Berthold K. P. Horn and Michael J. Brooks, editors, *Shape from shading*, pages 513–531. MIT Press, Cambridge, MA, USA, 1989.

[84] Changchang Wu, Sameer Agarwal, Brian Curless, and Steven M. Seitz. Schematic surface reconstruction. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 1498–1505, Washington, DC, USA, 2012. IEEE Computer Society.

[85] Jianxiong Xiao and Yasutaka Furukawa. Reconstructing the world's museums. In *Proceedings of the 12th European conference on Computer Vision - Volume Part I*, ECCV'12, pages 668–681, Berlin, Heidelberg, 2012. Springer-Verlag.

[86] K. Yagyu, K. Hayashi, and SC Chang. Orientation of multi-hair follicles in nonbald men: perpendicular versus parallel. *Dermatologic Surgery*, 32(5):651–660, 2006.

[87] Tatsuhisa Yamaguchi, Bennett Wilburn, and Eyal Ofek. Video-based modeling of dynamic hair. In *Proc. PSIVT*, 2008.

[88] Ruigang Yang, Marc Pollefeys, and Greg Welch. Dealing with textureless regions and specular highlights-a progressive space carving scheme using a novel photo-consistency measure. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ICCV '03, pages 576–, Washington, DC, USA, 2003. IEEE Computer Society.

[89] Kuk-Jin Yoon and In So Kweon. Adaptive support-weight approach for correspondence search. *IEEE Trans. PAMI*, 28(4):650–656, 2006.

[90] Cem Yuksel, Scott Schaefer, and John Keyser. Hair meshes. *ACM Trans. Graph.*, 28(5):166:1–166:7, 2009.

[91] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *Proceedings of the third European conference on Computer Vision (Vol. II)*, ECCV '94, pages 151–158, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc.

[92] Li Zhang, Brian Curless, and Steven M. Seitz. Rapid shape acquisition using color structured light and multi-pass dynamic programming. In *The 1st IEEE International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 24–36, June 2002.

[93] Qing Zhang, Jing Tong, Huamin Wang, Zhigeng Pan, and Ruigang Yang. Simulation guided hair dynamics modeling from video. *Computer Graphics Forum*, 31(7pt1):2003–2010, 2012.

[94] Ruo Zhang, P.-S. Tsai, J.E. Cryer, and M. Shah. Shape-from-shading: a survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(8):690–706, 1999.

[95] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Trans. PAMI*, 22(11):1330–1334, 2000.

[96] Shuang Zhao, Wenzel Jakob, Steve Marschner, and Kavita Bala. Building volumetric appearance models of fabric using micro ct imaging. *ACM Trans. Graph.*, 30(4):44:1–44:10, July 2011.

[97] Shuang Zhao, Wenzel Jakob, Steve Marschner, and Kavita Bala. Structure-aware synthesis for predictive woven fabric appearance. *ACM Trans. Graph.*, 31(4):75:1–75:10, July 2012.

[98] Qian Zheng, Andrei Sharf, Guowei Wan, Yangyan Li, Niloy J. Mitra, Baoquan Chen, and Daniel Cohen-Or. Non-local scan consolidation for 3d urban scene. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2010)*, 29:Article 94, 2010.