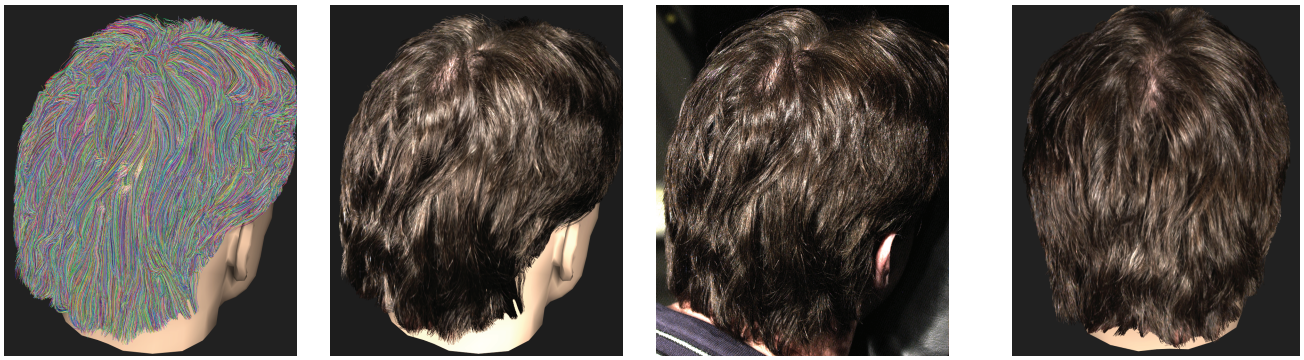
Hair Photobooth: Geometric and Photometric Acquisition of Real Hairstyles

Sylvain Paris[1] Will Chang[2] Oleg I. Kozhushnyan[3] Wojciech Jarosz[2]
Wojciech Matusik[1] Matthias Zwicker[2] Frédo Durand[3]
[1]Adobe Systems, Inc. [2]University of California, San Diego [3]MIT CSAIL



(a) reconstructed geometry (b) rendering (c) reference photograph (not in input data) (d) rendering (new viewpoint, new lighting)

Figure 1: Our method acquires photometric and geometric models of real hairstyles. It computes an accurate geometric model (a) which is used in combination with an image-based rendering technique to produce images (b). The results closely match an actual photograph (c) which is not part of the data set used by our algorithm. The hair can be rendered from arbitrary viewpoints and under arbitrary illumination (d).

Abstract

We accurately capture the shape and appearance of a person's hairstyle. We use triangulation and a sweep with planes of light for the geometry. Multiple projectors and cameras address the challenges raised by the reflectance and intricate geometry of hair. We introduce the use of structure tensors to infer the hidden geometry between the hair surface and the scalp. Our triangulation approach affords substantial accuracy improvement and we are able to measure elaborate hair geometry including complex curls and concavities. To reproduce the hair appearance, we capture a six-dimensional reflectance field. We introduce a new reflectance interpolation technique that leverages an analytical reflectance model to alleviate cross-fading artifacts caused by linear methods. Our results closely match the real hairstyles and can be used for animation.

Keywords: hair, active vision, image-based rendering

1 Introduction

Human hair is a significant challenge in image synthesis because of its geometric and photometric complexity. The problem becomes even harder when one wants to reproduce faithfully the hairstyle of a given person, because traditional 3D scanners usually fail with objects that do not correspond to a well-defined surface of Lambertian material. In this paper, we present a new scanning technique to reconstruct the geometry and reflectance of a person's hairstyle. We seek to capture complex geometric effects and individual hair strands for hairstyles with non-convex features, as well as complex appearance effects using image-based rendering.

We build on recent advances that have enabled the capture of hair geometry from image sequences [Paris et al. 2004; Wei et al. 2005]. Both rely on the extraction of hair orientation in images and chain these 3D orientations head-to-tail to build the geometry inside a visual hull. These methods were the first to enable hair capture, but ambiguities inherent to the visual hull make it difficult to capture complex features such as concavities and curls.

Our technique alleviates these limitations using triangulation between a plane of light and a camera. While the principle of triangulation is standard in 3D scanning, hair raises several challenges due to its intricate geometry and reflectance. The highly-specular BRDF rarely reflects the light towards the camera, and the filtering techniques implemented in scanning software heavily rely on the assumption that the object has a well-defined surface. Individual hair fibers are tiny and, while a hairstyle has a volumetric nature, it is hard to measure hair close to the scalp. A high-quality 3D model of hairstyle requires the reconstruction of individual hair fibers, especially when it is to be used for animation. We introduce a new hole-filling technique based on structure tensors to generate orientation inside the hairstyle volume.

While Marschner *et al.* [2003] have recently proposed a comprehensive analytical model of hair reflectance, acquiring a description of the BRDF of real hair remains unsolved and their paper only shows qualitative manual fits. Instead, we use a data-driven approach and acquire images under different lighting directions and from a number of cameras to recover a hair reflectance field. We introduce a new image-

based rendering technique to faithfully reproduce hair. In particular, it alleviates the cross-fading artifacts generated by traditional multi-linear interpolation by leveraging a simple hair BRDF as basis to guide the interpolation.

Contributions We introduce the following contributions:

▷ A multi-projector multi-camera approach that enables the triangulation of hair geometry.

▷ A new hole filling technique for hair orientation based on structure tensors.

▷ A new model-based interpolation for image-based rendering that yields faithful interpolation of specularities.

1.1 Related Work

A wealth of work deals with three-dimensional reconstruction of real-world objects [Blais 2004]. The following sections discuss the methods most related to our approach.

Triangulation Scanning sweeps the object with a plane of light (e.g. laser); and simple geometry deduces 3D locations from the image of the intersection of the light plane and the object [Curless 1997; Pulli 1997; Levoy et al. 2000; Davis et al. 2003]. Such techniques work best on diffuse objects that present a well-defined surface, the opposite of hair. Structured light scanning accelerates triangulation by covering a large part of space with each projected pattern [Rusinkiewicz et al. 2002; Zhang et al. 2003], but suffers from the same issues with hair. In comparison, single-plane setups have shown great ability to cope with difficult conditions such as smoke [Hawkins et al. 2005] and underwater diffusion [Narasimhan et al. 2005]. This paper tackles the difficulties linked to hair. In particular, hair fibers are too small to completely block the light ray coming from a single projector pixel. As a consequence, a single ray can hit several fibers at once, resulting in complex light patterns that would fool classical laser scanners.

Hair Modeling and Reconstruction Interactive tools have been developed to model hair geometry [Ward et al. 2006] and impressive results have been produced [Mihashi et al. 2003]. This provides the user with full control, but requires time and talent to match a given hairstyle.

We build on recent work on hairstyle acquisition from multiple views [Grabli et al. 2002; Paris et al. 2004; Wei et al. 2005]. Our approach uses the same signal processing approach as Paris *et al.* to analyze the hair orientation in input images. We convert these orientations into 3D with a multi-view method inspired from Wei *et al.*. However, we directly acquire 3D location using triangulation, while these techniques rely on a visual hull approximation. Our setup is made of static, carefully calibrated elements whereas Paris *et al.* and Wei *et al.* use handheld lights or cameras. We also minimize the subject motion with a comfortable headrest and compensate for the residual motion using tracking whereas Paris *et al.* require the subject to move. Moreover, we also capture reflectance data and introduce a new image-based rendering method to exploit these data whereas Paris *et al.* use manually adjusted parameters and Wei *et al.* compute view and light-independent colors, thereby precluding renderings under new lighting environments.

Hasinoff and Kutulakos [2006] have shown how to extract the depth map of a hairstyle from several photographs with varying focus and aperture settings. However, the extracted data lack depth accuracy and do not describe the strand structure, which precludes rendering and animation.

Scanning Human Faces Several capture setups have been proposed to acquire geometric and photometric data of human faces [Debevec et al. 2000; Zhang et al. 2004; Wenger et al. 2005; Weyrich et al. 2006]. They are complementary to our approach to generate a full virtual human head but are not suitable for hair since they are not adapted to the one-dimensional and highly-specular nature of hair fibers.

Surface Reflectance Fields Our image-based rendering approach builds on reflectance fields, which can be thought of as a large collection of images of an object under a number of viewing and lighting conditions [Debevec et al. 2000; Matusik et al. 2002]. They can represent intricate geometry with complex reflectance properties. When the surface is well defined, fitting analytical BRDFs alleviates the artifacts caused by sparse sampling [Lensch et al. 2001; Weyrich et al. 2006]. If the surface is not perfectly smooth, mesostructures can be captured by texture maps [Kautz et al. 2004]. Unfortunately, hair is far from being a smooth surface with well-defined normals and any pixel in an acquired image includes shadowing effects and captures contributions from several fibers with potentially varying orientations. In addition, the BRDF of hair [Marschner et al. 2003] is a complex function with many parameters that is hard to fit to a small number of samples. This is why we resort to direct interpolation of the acquired data. However, the high specularity of hair introduces artifacts in linear interpolation and we introduce a new data interpolation technique based on an analytical BRDF model.

Scanning Complex Objects Matusik *et al.* [2002] combine visual hull, reflectance fields and alpha matting to capture complex objects such as a Teddy bear. They achieve realistic reproductions but the acquired data are not suitable for editing nor animation since they ignore the intrinsic nature of the object. Reche *et al.* [2004] describe a volumetric method that captures and render faithful models of trees, but the fine fiber structure and the specular aspect of hair would not be addressed by this technique.

1.2 Method Overview

Capture Setup Our hair capture uses a set of cameras, video projectors and controllable LEDs arranged in a dome arround the hair and proceeds in two stages responsible for geometry and photometry respectively. The geometry pass relies on a sweep of the scene with planes of light (Fig. 4), while the reflectance phase captures the hair illuminated from 150 directions (Fig. 2). The duration of the acquisition makes motion compensation a necessity. Our capture setup is described in Section 2.

Geometric Reconstruction We use triangulation to retrieve the location of the strands, thereby recovering more intricate geometry than previous methods. In addition to position, orientation is a critical aspect of hair geometry. We use a traditional approach to orientation computation but introduce a new hole filling algorithm based on a quadratic representation of lines. Finally, we grow strands within this 3D orientation field to obtain three-dimensional curves using variations of known techniques. Geometric reconstruction is covered in Section 3.

Reflectance Acquisition To faithfully reproduce the complex reflectance of hair, we leverage the image data from multiple cameras and light directions. To render a new view, each 3D hair fiber location is projected onto the acquired images and we interpolate the values from the nearest neighbors. We introduce a simple technique to interpolate re-

flectance samples based on an analytical reflectance model. Our reflectance rendering is discussed in Section 4.

2 Acquisition Setup

Reconstructing the geometry and the reflectance requires a custom measurement system in order to make the acquisition robust and accurate. Our setup (Fig. 2) is inspired by light stages [Debevec et al. 2000; Matusik et al. 2002; Wenger et al. 2005; Weyrich et al. 2006] and structured light scanners [Rusinkiewicz et al. 2002; Zhang et al. 2003]. The subject sits in a chair with a head rest to keep the head still. It is surrounded by 16 Basler cameras (resolution 1300×1030), 3 DLP projectors (resolution 1024×768), and 150 LED lights mounted on a steel geodesic dome. They are regularly spaced to span the view and light directions. The projectors and cameras are geometrically calibrated [Zhang 2000] while the LEDs are mounted at known positions.

Our acquisition has two steps. In the geometry step, we capture hair images illuminated by a scanline of a single projector at a time. We simultaneously capture images with all 16 cameras. At the end, the hair volume has been swept by light planes coming from 3 directions. Every 10 frames, we insert a tracking frame with all projectors on to enable tracking for motion compensation. We capture 40 512 images in about 17 minutes. Our current setup is limited by network and storage bandwidth and does not allow more than 2 frames per second for each camera. More recent connections and drives should permit at least video frame rate and would reduce the acquisition time to about 2 minutes.

In the reflectance step, the system sequentially turns each LED light on in a light-stage fashion [Debevec et al. 2000; Matusik et al. 2002; Wenger et al. 2005; Weyrich et al. 2006]. All 16 cameras capture images simultaneously. The complete sequence takes about 25 seconds for all 150 light sources. We capture 4 800 images. In addition, for each camera, we sum the 150 LED images to obtain a uniformly lit image used for extracting hair orientations.

Specifics In our approach, light positions are combined with all camera positions. In contrast, sequential scanning usually only moves the objects, for instance using a turntable. As a result, our approach captures a wider variety of light-view configurations, which is critical to handle the complex specularities of hair. In addition, all the cameras record simultaneously, thereby reducing capture time through temporal multiplexing.

Scalp Geometry The scalp is important because this is where the hair fibers are rooted. We use a contact digitizer to measure a set of points on the scalp and fit a generic head mesh using Iterated Closest Point [Besl and McKay 1992].

3 Geometry Processing

Our geometry reconstruction (Figure 3) computes two main pieces of information: the complex volume occupied by hair,

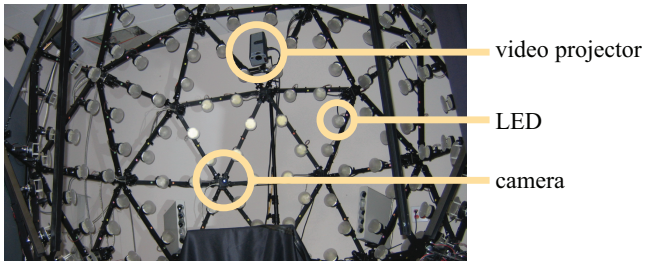
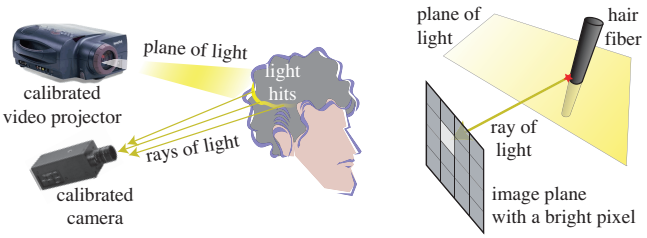


Figure 2: Our acquisition setup consists of 16 digital cameras, 150 LED light sources, and 3 DLP projectors.



(a) triangulation setup (b) geometric configuration

Figure 4: Triangulation relies on a calibrated camera and a calibrated projector that generates a plane of light (a). This creates a light line on the hair volume visible from the camera (b). Since both the projector and the camera are calibrated, the geometric configuration is fully known. For each bright pixel in the camera image, the corresponding hair fiber is located at the intersection between the light plane from the projector and the light ray to the camera (b).

and the orientation of hair fibers. Our approach permits the extraction of complex volumetric data by enabling the use of triangulation with hair (Section 3.1). Our computation of orientation from images (Section 3.2) is similar to previous work [Paris et al. 2004; Wei et al. 2005] but we introduce a new hole filling algorithm (Section 3.3) that allows us to generate high-quality data for the full hair volume (Section 3.4). The input data of the geometry step are stripe images used for triangulation, uniformly lit images from which extract hair orientations, and tracking images for estimating the head motion.

3.1 Position Triangulation

Hair Location For each pair of camera and light plane, we obtain a stripe image similar to Figure 3. We extract the intersection of the light plane with the hair volume by thresholding the intensity. Since both the projector and the camera are calibrated, we can compute the intersection of the light plane and the ray originating from the camera center to a bright pixel to yield a 3D position (Fig. 4).

To improve accuracy, we use *space-time analysis* to account for the light plane thickness [Curless and Levoy 1995]. In practice, we consider only intensities corresponding to a temporal maximum. This discards spurious side responses which represent about 75% of the bright pixels.

We triangulate the data from all cameras separately. Since the cameras are registered in a common coordinate system, merging the 3D points is straightforward. To improve accuracy, we compensate for the subject motion as described below. Although we carefully acquire the data, a few outliers still appear due to reflections on surrounding objects and residual subject motion. We remove those errors by culling points outside the visual hull; these correspond to about 2% of the acquired data. The visual hull is obtained from approximate masks painted by hand. A faster acquisition would reduce the subject motion and alleviate the need for the visual hull.

An example of reconstructed data is shown in Figure 3. Observe the fine details and concavities of the acquired volume. A few regions in shadows are not triangulated and create small holes. Yet, the final model is complete since we are able to infer missing data (Section 3.3).

Motion Compensation Even with a head rest, small movements are unavoidable and produce projection errors up to 50 pixels in our tests. We reduce this error down to at most 3 pixels with the compensation procedure described below. In practice, we compute a rigid 3D transformation for each

stripe image and apply the inverse transform to the triangulated 3D points.

• *2D Tracking*: We first track the motion in the image planes of each camera. Similarly to Wenger *et al.* [2005], we insert a tracking frame with all projectors on every ten stripe frames. We compute the rigid transformations T^{2D} that maximize the correlation of each frame with the reference frame $\bar{F}$ (*e.g.* the last one). The user provides a mask M for the hair, which could be automated since only an approximate mask is required. To avoid aliasing, we convolve the images with a small Gaussian G. Formally, we solve the following optimization for each tracking frame F_i:

$$\text{argmin}_{T_i^{2D}} \sum_{\mathbf{p}\in M} \left\| T_i^{2D}(G \otimes F_i)_{\mathbf{p}} - (G \otimes \bar{F})_{\mathbf{p}} \right\|^2 \quad (1)$$

Since the movements are small, gradient descent converges efficiently. The accuracy is on the order of a pixel.

• *Recovering the 3D Motion*: For each time step, we combine the 2D motions computed for all cameras to deduce a 3D transformation. Using a non-compensated reconstruction, *i.e.* we merge the 3D points without applying any compensation, we estimate the hair surface location and name $\mathbf{c}_j$ the closest point to camera C_j. We search for the 3D rigid transformation T^{3D} that best approximates the 2D motions, in a least square sense. We denote P_j the projection matrix of camera C_j and $\mathbf{u}_i^j$ the 2D translation component of the transformation obtained from Equation 1 with the tracking frame F_i and the camera C_j. The 3D motion is obtained with gradient descent by optimizing:

$$\text{argmin}_{T_i^{3D}} \sum_j \left\| \left[P_j(T_i^{3D}(\mathbf{c}_j)) - P_j(\mathbf{c}_j) \right] - \mathbf{u}_i^j \right\|^2 \quad (2)$$

We linearly interpolate the compensation for the in-between frames [Wenger et al. 2005]. The overall remaining error is in the order of a couple of pixels.

• *Summary*: We compute 2D motion estimates at each tracking frame and combine them to evaluate the 3D motion at these frames. We use linear interpolation to obtain the motion at each time step.

The use of triangulation produces an volumetric occupancy map that is significantly more accurate than the visual hull used in previous work.

3.2 Volumetric Orientation Field

We now extract a dense 3D orientation field. For the visible portion of the hair volume, we mostly follow previous work [Paris et al. 2004; Wei et al. 2005]. We then introduce a new hole-filling algorithm for the hidden areas.

2D Orientations We evaluate the strand orientations in the image plane of each camera following Paris *et al.* [2004]. We apply several orientation detectors and select the most reliable result according to the variance of the response curves. We found that, with our setup, a single uniformly lit image

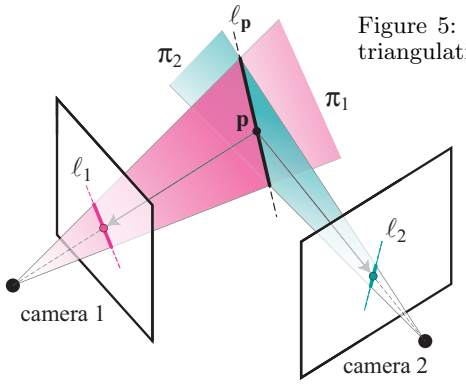


Figure 5: 3D orientation triangulation.

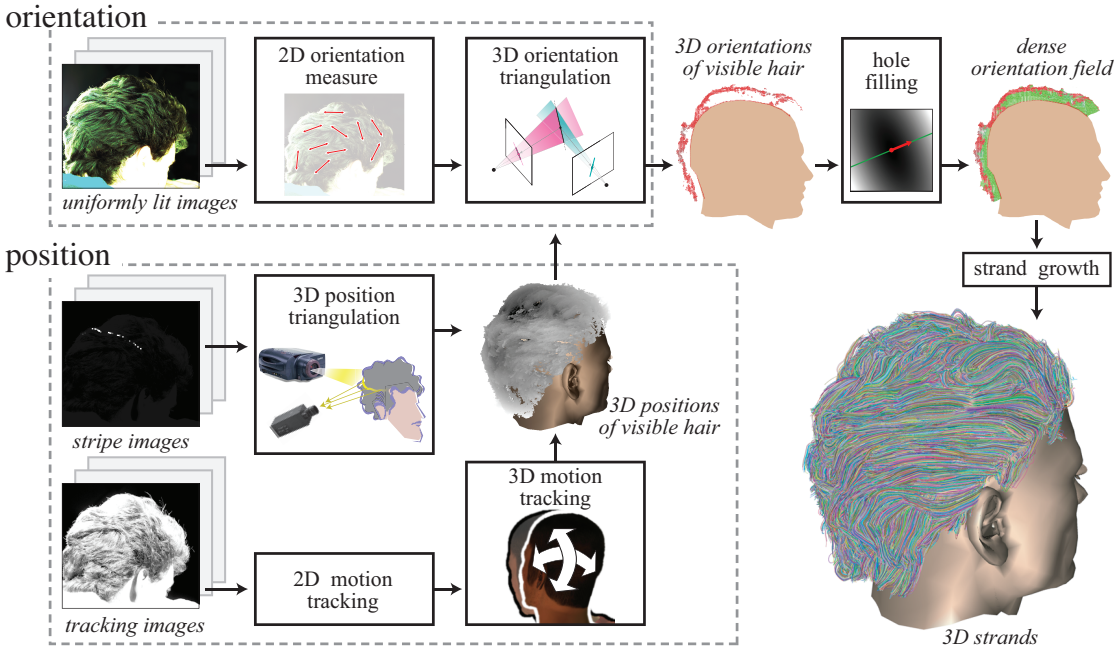


Figure 3: Our geometry reconstruction pipeline computes both a volumetric representation of hair position (bottom) and hair orientation (top). Our algorithm takes as input 16 uniformly images lit with all 150 LED lights turned on (one such image from each camera), 36 864 stripe images with a single line of light projected (768 lines per projector and 3 projectors), and 3 648 tracking images with all three projectors turned on (one such image every 10 stripe images). Hair fibers are then grown inside the volume and along the orientation, starting from the scalp.

was sufficient instead of several images with different lighting conditions as Paris *et al.* do. We capture these images with the 150 LED lights turned on. The result of this step is a 2D orientation map for each view; that is, for each pixel, we have a 2D line along the local hair orientation.

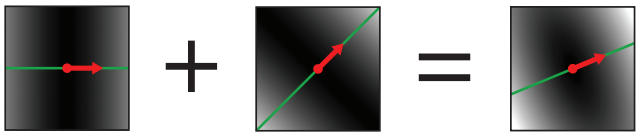
3D Orientations We compute the 3D orientation for each 3D point $\mathbf{p}$ using the cameras with the most frontal views. For this, we fit a plane to the 3D hair points surrounding $\mathbf{p}$ [Hoppe et al. 1992] and use the normal of this plane to determine the two most front-facing cameras C_1 and C_2. The projections of $\mathbf{p}$ in the images from C_1 and C_2 provides two 2D lines $\boldsymbol{\ell}_1$ and $\boldsymbol{\ell}_2$ computed during the 2D step. Back-projecting $\boldsymbol{\ell}_1$ and $\boldsymbol{\ell}_2$ in space forms two 3D planes π_1 and π_2. Following Wei *et al.* [2005], the 3D orientation $\boldsymbol{\ell}_\mathbf{p}$ of $\mathbf{p}$ is the intersection of π_1 and π_2: $\boldsymbol{\ell}_\mathbf{p} = \pi_1 \cap \pi_2$ (Fig. 5). Two cameras are sufficient since our cameras are well spaced. The result is a set of 3D orientations, that is, 3D lines locally parallel to the hair fibers.

3.3 Hole Filling

The previous step recovers the orientation of the visible fibers and to obtain a full head of hair, we need to infer the data for the hidden parts. This step also fills in small holes that may not have been triangulated. We seek to interpolate the existing orientation, but performing operations on orientations is not as simple as on scalars or vectors. In particular, we do not know the direction towards the root from the image data and cannot directly use unit vectors because two opposite vectors represent the same orientation. Our problem is related to quaternions but we only have two degrees of freedom (only pitch and yaw) and, as just mentioned, opposite directions represent the same orientation.

Structure Tensors For this purpose, we introduce the use of *structure tensors* [Granlund and Knutsson 1994] (p 301). We represent a 3D line $\boldsymbol{\ell}$ by a 3×3 symmetric matrix O such that $\boldsymbol{\ell}$ is the eigenspace corresponding to its largest eigenvalue. If we consider the quadratic function $q(\mathbf{x}) = \mathbf{x}^\mathsf{T} O\,\mathbf{x}$ where $\mathbf{x}$ is a 3D point, the line $\boldsymbol{\ell}$ is the direction along which q varies the fastest. See a 2D illustration in Fig. 6. To manipulate 3D lines, we operate on these quadratic functions. We sum their matrices, and at the end, the resulting line is the orientation of fastest variation of the modified function.

Given a line $\boldsymbol{\ell}$, we build a rank-1 matrix $O = \mathbf{v}_\boldsymbol{\ell}\,\mathbf{v}_\boldsymbol{\ell}^\mathsf{T} \,/\, \|\mathbf{v}_\boldsymbol{\ell}\|^2$ where $\mathbf{v}_\boldsymbol{\ell}$ is a direction vector of the line $\boldsymbol{\ell}$. Because of the product and normalization, all possible direction vectors give the same matrix. Conversely, given a matrix O, we compute an eigenvector corresponding to its largest eigenvalue. This



$\mathbf{v}_1 = \begin{pmatrix}1\\0\end{pmatrix}$ $\quad O_1 = \begin{pmatrix}1 & 0\\0 & 0\end{pmatrix}$

$\mathbf{v}_2 = \begin{pmatrix}0.7\\0.7\end{pmatrix}$ $\quad O_2 = \begin{pmatrix}0.5 & 0.5\\0.5 & 0.5\end{pmatrix}$

$\mathbf{v}_{12} = \begin{pmatrix}0.9\\0.4\end{pmatrix}$ $\quad O_{12} = \begin{pmatrix}1.5 & 0.5\\0.5 & 0.5\end{pmatrix}$

Figure 6: Addition of two 2D lines. We use direction vectors $\mathbf{v}$ to build structure tensors O. We visually represent a tensor O by the plot in gray level of the quadratic function $\mathbf{x}^\mathsf{T} O\,\mathbf{x}$. We sum these tensors and extract the eigenvector of the largest eigenvalue of the summed tensor. Our hole-filling method uses the same orientation representation in 3D; that is, lines and vectors are 3D and the matrices are 3×3.

defines a line $\boldsymbol{\ell}$ and all other eigenvectors which are scaled or flipped yield the same line. The mapping is ill-defined only if the eigenspace is more than 1D, which corresponds to situations that do not occur in practice, such as the addition of two perfectly orthogonal lines.

Interpolation We interpolate the measured data with a simple diffusion process akin to the heat equation, *a.k.a.* isotropic diffusion. We build a regular 3D grid with a 1mm resolution. Each node stores an orientation tensor O. We compute the heat equation on this grid:

$$\frac{\partial O}{\partial t} \;=\; \frac{\partial^2 O}{\partial x^2} + \frac{\partial^2 O}{\partial y^2} + \frac{\partial^2 O}{\partial z^2} \qquad (3)$$

We iterate Equation 3 while using the measured orientations as hard constraints, *i.e.* we do not modify their values. We also use the scalp normals as constraints to "bend" the orientation field toward the scalp. In other words, grid nodes containing a measured orientation or on the scalp surface are Dirichlet boundary conditions. This process defines a new matrix field $\tilde{O}$, and the local fiber orientation at point $\mathbf{p}$ is obtained by extracting the largest eigenvector of $\tilde{O}_\mathbf{p}$.

On a typical model (Fig. 3 and 7), interpolation generates between 30% and 70% of the final geometry. This proportion underscores the importance of this step.

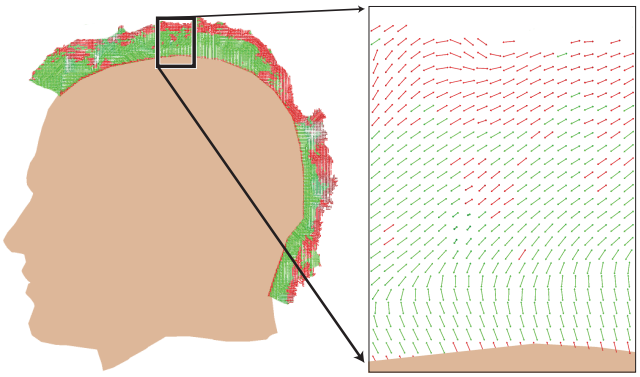


Figure 7: Hole filling demonstrated on a 2D slice. We use the measured points and the roots on the scalp as hard constraints (red) to infer the missing orientation data (green).

3.4 Strand Creation

From the previous sections, we have both location and orientation information, regularly sampled every 1mm in space. We construct 3D curves by following the orientation field. We start at root points randomly sampled over the scalp. The root density is set to produce approximately 100 000 strands, a typical number for humans. Let $\mathbf{p}_0$ be such a root with scalp normal $\mathbf{n}$. We grow a strand in the direction $\mathbf{d}_{\mathbf{p}_0} = \mathbf{n}$. Given a point $\mathbf{p}$ and a direction of growth $\mathbf{d}_\mathbf{p}$, we extend the strand as follows:

1. Compute the next point $\mathbf{q} = \mathbf{p} + \delta \frac{\mathbf{d}_\mathbf{p}}{\|\mathbf{d}_\mathbf{p}\|}$.
2. Given a vector $\boldsymbol{\ell}_\mathbf{q}$ along the orientation at $\mathbf{q}$. The growth direction $\mathbf{d}_\mathbf{q}$ can be $+\boldsymbol{\ell}_\mathbf{q}$ or $-\boldsymbol{\ell}_\mathbf{q}$. We select the option that induces the smaller rotation angle: $\mathbf{d}_\mathbf{q} = \text{sign}(\boldsymbol{\ell}_\mathbf{q} \cdot \mathbf{d}_\mathbf{p})\, \boldsymbol{\ell}_\mathbf{q}$.
3. Iterate until a predefined maximum length is reached.
4. Create a strand starting from the root to the last measure point encountered. If no measure point encountered, delete the strand.

The integration step δ in Step 1 is set to 0.5mm. Step 2 prevents sharp turns. Step 3 enforces a maximum length set by the user depending on the length of the subject's hair. Step 4 ensures that the strands remain within the hair volume and ends at a measured point. This last property, ending at a measured point, is a major difference with existing methods that rely on the visual hull. Our approach produces a detailed hair volume measured by triangulation. Figure 3-right shows an example of the reconstructed hair strands.

4 Model-based Interpolation of Reflectance Fields

The second stage of our capture generates reflectance field data corresponding to a large number of camera-light pairs. We leverage this data to faithfully reproduce the appearance of the subject's hair, including highlights and color variation. The quality we obtain is superior to previous image-based approaches because we do not use a simplified geometric surrogate such as a visual hull [Gortler et al. 1996; Matusik et al. 2002] but render actual hair geometry. Furthermore, we introduce a new model-based interpolation scheme based on a simple parametric BRDF [Kajiya and Kay 1989] that allows us to reproduce sharp highlights. We expose a classical data-driven rendering method before describing our model-based approach. We then extend it to animation.

4.1 Standard Interpolation

We acquire a six-dimensional reflectance field that contains samples of the hair appearance from 16 view and 150 light directions. To reproduce the hair appearance from a new view and light direction, we interpolate samples from the three closest cameras and a small number of closest light sources. For each 3D location on a hair strand, we first retrieve the corresponding reflectance samples by projecting the 3D location into the acquired images. Each camera and light pair (i, j) contributes one reflectance sample I_{ij}. For each sample, we compute a weight w_{ij} that is the product of a barycentric weight for the camera and a radially symmetric weight function for the light. We prefer smooth radial functions over barycentric interpolation for the lights because it leads to smoother animations. Finally, the interpolated reflectance is the weighted sum $\sum_{ij} w_{ij} I_{ij}$.

This interpolation is essentially the same as most image-based techniques, with the exception that it uses our high-quality geometry. It achieves good quality for still images, but results in loss of highlight sharpness and in highlight ghosting, which is objectionable for animation.

4.2 Model-based Interpolation

To introduce our new model-based interpolation, we take a somewhat counterintuitive perspective on the above interpolation. Consider how the standard interpolation predicts the hair appearance over the space of view and light directions. We simply compute a weighted sum of samples regardless of the view and light direction. We can understand this as the weighted average of constant functions (Fig. 8 left). Our key idea is to replace these constant functions with a hair BRDF (Fig. 8 right) that more faithfully captures hair reflectance over changing view and light directions.

Reflectance Prediction We assume that the hair reflectance follows the Kajiya-Kay model [1989], chosen for its simplicity:

$$I = L\, K(\mathbf{t}, \mathbf{h})\, \cos(\theta) \qquad (4)$$
$$\text{with} \quad K(\mathbf{t}, \mathbf{h}) = \rho_s\, \|\sin(\mathbf{t}, \mathbf{h})\|^{\alpha} + \rho_d$$

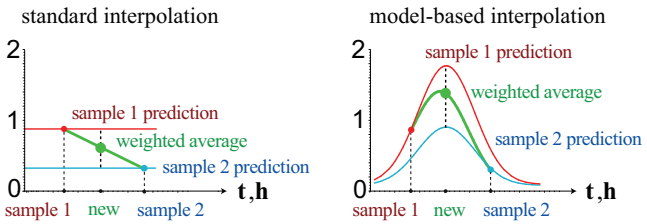


Figure 8: On the left, we see that standard interpolation predicts constant reflectance everywhere. In comparison, our model-based interpolation uses the hair BRDF to predict more accurate reflectance curves.

where I is the intensity of the reflected light, L the intensity of the light source, $\mathbf{t}$ the hair direction, $\mathbf{h}$ the half vector between the view and light directions, θ the angle between the light and the plane perpendicular to the hair direction, ρ_s and ρ_d are the specular and diffuse coefficients, and α is the shininess exponent. We postpone the discussion how to determine the parameters ρ_s, ρ_d, and α and assume they are known for now.

Given an observed reflectance sample I_{ij} we invert Equation (4) to estimate the light intensity that reaches this sample, $L_{ij} = I_{ij} \,/\, [K(\mathbf{t}, \mathbf{h}_{ij})\, \cos(\theta_{ij})]$. Here $\mathbf{h}_{ij}$ and θ_{ij} are the parameters of the observed view i and light j. Using the estimated incoming light, we apply the Kajiya-Kay model to predict the reflected light $L_{ij}\, K(\mathbf{t}, \mathbf{h})\, \cos(\theta)$ for new light and view directions. Therefore, we predict the reflectance at the desired directions as:

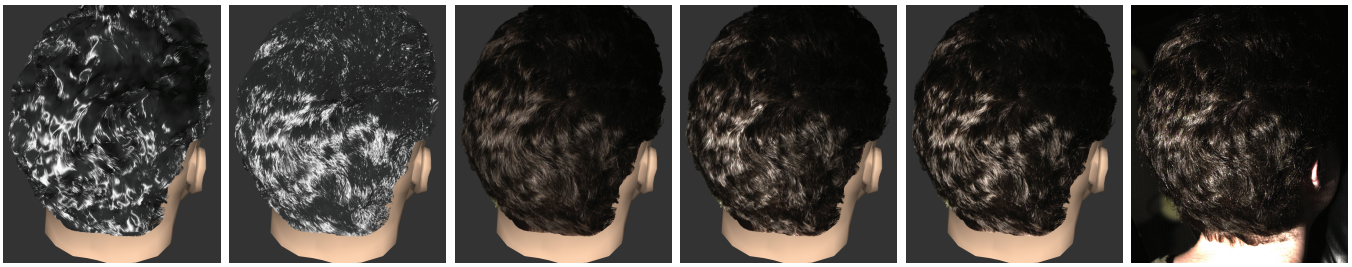
$$\hat{I}_{ij}(\mathbf{h}, \theta) = I_{ij} \frac{K(\mathbf{t}, \mathbf{h})\, \cos(\theta)}{K(\mathbf{t}, \mathbf{h}_{ij})\, \cos(\theta_{ij})} \qquad (5)$$

where $\mathbf{h}$ and θ are parameters of the desired light and viewing directions, and I_{ij}, $\mathbf{t}$, $\mathbf{h}_{ij}$, and θ_{ij} are constant since they correspond to measured data.

We interpolate these predicted values $\hat{I}_{ij}(\mathbf{h}, \theta)$ as before using the weighted sum $\sum_{ij} w_{ij}\, \hat{I}_{ij}$. In summary, for each reflectance data point, we scale a Kajiya-Kay BRDF to predict hair reflectance in its neighborhood. Figure 8 illustrates the benefits of our model-based prediction scheme. Whereas a standard approach systematically "cuts" highlights, our approach is able to predict them and yields significantly more accurate results. The limitation of our approach is that it does not take into account changes in light source visibility. Since we only extrapolate locally, however, this is not noticeable with our data.

Parameters The parameters ρ_s, ρ_d, α of the Kajiya-Kay model are scalars in our technique because the color information comes from the sampled reflectance I_{ij}. We assume constant parameters over the whole hair; spatial variations are captured by the image data. Since Equation 5 is invariant under uniform scaling of ρ_s and ρ_d, we set $\rho_s = 1$ without loss of generality.

We determine the two other parameters ρ_d and α by removing one light source from our image set and searching for the values of ρ_d and α that "best" approximate this data using only the remaining images. We perform an exhaustive search over a small range of possible values. We use the maximum L_2 error over small image areas to evaluate the accuracy of the approximated data because it yields good reproduction of the highlights. Since highlights cover only a small fraction of the data, other criteria such as the average pixel error or the L_2 error over the whole image are dominated by large dark regions and produce less pleasing results.



(a) Kajiya-Kay shading without per-camera normals | (b) Kajiya-Kay shading with per-camera normals | (c) naive linear interpolation | (d) our model-based interpolation without per-camera normals | (e) our model-based interpolation with per-camera normals | (f) reference photograph (not in input data)

Figure 9: Our model-based view interpolation relies on the Kajiya-Kay BRDF to predict highlight location. First, we show the hair model rendered with this model only (a,b). Our per-camera normals (b) successfully capture the fine details of the hair reflection. We then validate our view-interpolation method. Traditional linear view interpolation (c) fails to render shiny highlights. In comparison, our approach (d,e) matches the hair shininess observed in the photograph (f). However, without per-camera normals, the details of the reflection patterns (d) differ from the original ones (f). Our per-camera normals (e) yield a faithful match to the person's hair (f).

Hair Orientation Refinement for Shading Rendering is very sensitive to hair orientation because hair is highly specular. Therefore, we refine the 3D orientations on a per-camera basis to optimize rendering quality. For each camera, we adjust strand orientations so that the Kajiya-Kay model best agrees with the variation over light directions. Using per-camera orientations allows us to approximate the highlight shift due to the cuticle angle [Marschner et al. 2003].

Given a strand with an original orientation $\mathbf{t}$, we generate candidate orientations $\mathbf{t}_k$ that uniformly cover the neighborhood of $\mathbf{t}$ defined by a maximum angle. For each strand segment, we collect reflectance samples $\{I_{kj}\}$ from a single camera k and over all lights j. We use z-buffering to discard the samples that are on the backside of the head. With the remaining samples, we select the orientation $\mathbf{t}_k$ that maximizes the correlation between the predicted and measured reflectance:

$$\arg\max_{\mathbf{t}_k} \sum_j I_{kj}\, K(\mathbf{t}_k, \mathbf{h}_{kj}) \cos(\theta_{kj})$$

This gives us a corrected orientation for each camera that is more faithful to the reflectance model. Figure 9 illustrates the gain brought by this correction.

4.3 Rendering of Animated Geometry

We extend our model-based interpolation to animation. The challenge is that the animated strands are not located where we measured their reflectance. Our strategy is to use the data from the rest pose to render the animated geometry. For each location on a moving hair strand, we look up its reflectance sample by projecting the corresponding rest position into the acquired images. To capture the variation in shading due to the animation, we apply Equation 5. This predicts the reflectance of the deformed geometry using new vectors $\mathbf{t}$, $\mathbf{h}$ relative to the new hair orientation.

Figure 10: Rendering of animated geometry. The result on animated data is better seen in the companion video.

We used a procedurally animated geometry to validate our approach. The generated translations are on the order of a few centimeters and the rotations can be up to 180°. Although this strategy ignores the occlusions and disocclusions that can occur due to the changes of geometry, it generates plausible results as demonstrated in Figure 10 and in the companion video. In particular, the renderings still exhibit the visual complexity characteristic of hair.

5 Results

We validate our method with side-by-side comparisons with real photographs for a variety of hair styles: straight (Fig. 1), wavy, tangled, and puffy (Fig. 11). For these comparisons, we have excluded the reference photographs from the data set used for image-based rendering. In terms of appearance, our result yields close visual matches with photographs of real hairstyles. It also renders new views under new lighting conditions (Fig. 1 and video), which is not possible with existing hair capture techniques [Paris et al. 2004; Wei et al. 2005]. We also provide videos with moving cameras and lights in supplemental material. Due to fundamentally different input data, it was not possible to directly compare to the methods of Paris *et al.* and Wei *et al.*.

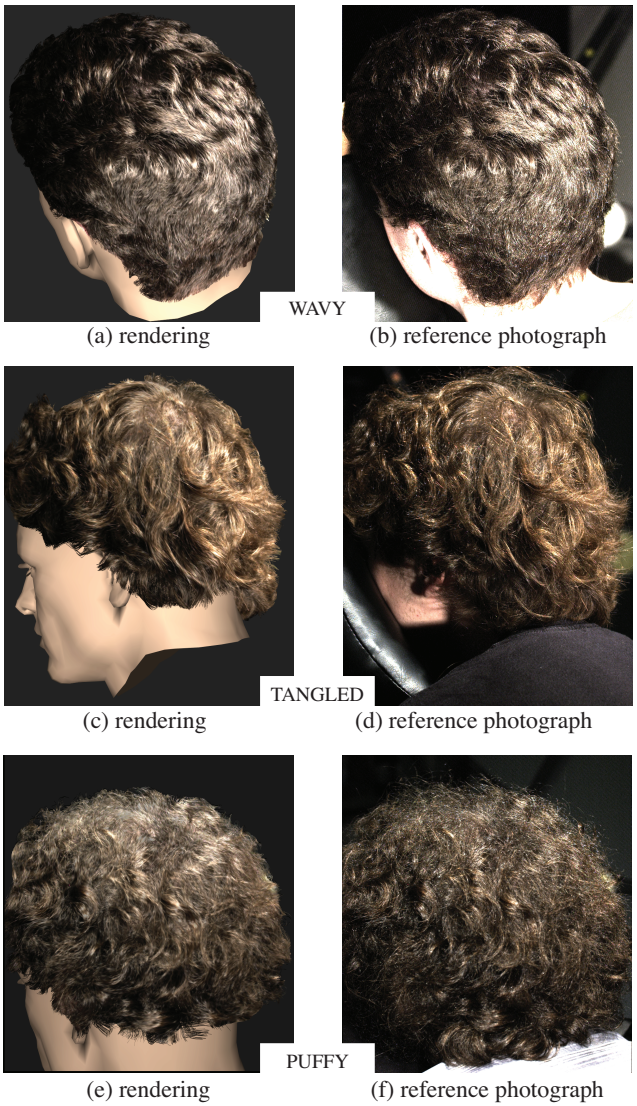
As shown in Figure 12, our algorithm is able to generate complex hair models, producing on the order of 100,000 hair strands. A distinctive property of our geometric models is their truly volumetric nature with concavities and intricate strands that faithfully capture the original hairstyles. In particular, the tangled and puffy hairstyles (Fig. 11-c,d,e,f) would be challenging for visual-hull approaches. In addition, our method produces strands attached to the scalp that enable animation. The video shows an example that uses a procedural motion field to animate the hair. In contrast, existing approaches retrieve only the visible hair layer.

The geometric reconstruction runs in 10 hours on a single core, most of it being due to 2D motion tracking that involves repeated transformations and comparisons of megapixel images. Since most computations are independent, they could be distributed on a cluster. On a single core of a Core2Duo 2.66GHz, rendering runs between 90s and 140s depending on the model and requires about 700MB of memory. Currently the bottleneck in our rendering is the memory access to read the reflectance data. During rendering, we access on the order of 4.3GB of reflectance data. Each shading operation interpolates 3 camera views and about 6 light directions. In total, accessing reflectance

samples accounts for approximately 50% of the rendering time. The next most expensive step is the model-based interpolation at about 25% of the total time. The rest is spent in image operations and disk and memory accesses.

Priorities Our approach has two main components, the geometry and the image-based reflectance. Which of those two aspects is most important depends on one's application and existing infrastructure. If one has a basic system to create some rough geometry, one could just implement the image-based component to render high-quality static images. To create a movie with changing viewpoints, the geometric part should be implemented first to correctly render occlusions and disocclusions. The companion video shows that our geometry yields superior results compared to a visual hull, especially with puffy hairstyles. The geometric part should also come first if the goal is to animate the geometry.

Limitations and Possible Extensions Our method achieves results that significantly improve the state of the art. However, this comes at the price of a dedicated acquisition setup.



(a) rendering (b) reference photograph

(c) rendering (d) reference photograph

(e) rendering (f) reference photograph

Figure 11: Side-by-side comparisons for various hairstyles. For fair comparison, the reference photographs were removed from the data set used by our image-based rendering method.

	Straight	Tangled	Puffy	Wavy
Strands	103,364	114,269	145,787	107,219
Vertices	3,703,570	3,528,195	4,568,003	3,287,934

Figure 12: Hair geometry statistics.

The current version of our setup also has a few low-level, technical limitations. First, the resolution of the camera (1.3 megapixels) does not allow us to capture stray hairs as can be seen at the silhouettes of the puffy hairstyle (Fig. 11). An intriguing avenue for the future is using high-resolution low-noise SLR cameras which can image individual fibers. We believe that it can lead to more accurate and lightweight acquisition and also produce higher dynamic range data which would better capture hair shininess. Cameras with wider fields of view would also let us capture longer hairstyles. Another limitation of our setup is that we do not have a precise model of the neck and back of the subject, which results in inaccuracies at the lower part of the hairstyle, *e.g.* the puffy model. Finally, higher bandwidth for network connections and storage devices should drastically reduce the acquisition time. This would reduce the errors remaining after motion compensation that cause small cross-fading artifacts. It would also alleviate the need for the visual hull used to remove strands out of the hair volume.

We also made a number of choices that may impact our results. Our method to fill the hidden part of the hair volume produces a smooth orientation field. Although this performed well, even on tangled hair, some hairstyles may benefit from a more sophisticated approach that takes into account information such as curvature statistics.

Our model-based interpolation scheme assumes that only hair is visible at a given image pixel, which may not be true at silhouettes where hair is thin and may reveal skin or background objects. This induces erroneous colors at a few spots near silhouettes. Matting the hair volume in the input images would address this color mixture issue.

The use of the Kajiya-Kay BRDF to guide our view-interpolation scheme yields visually convincing results. Yet, this reflectance model has its shortcomings; in particular, it tends to overestimate reflections at grazing angles, which produces too bright highlights at some places in our results. Switching to the more accurate model of Marschner and colleagues [2003] would reduce this effect at the cost of a higher-dimensional, nonlinear parameter fitting problem.

6 Conclusions

We have described a method to capture both the geometry and the appearance of a person's hairstyle. Our capture setup and analysis algorithm significantly improve the accuracy of the acquired geometry. We have complemented the acquisition with a method to infer the hidden geometry of hair. We have also shown the gain brought by incorporating a priori knowledge in the image-based rendering. As a result, we are able to capture real hairstyles with their full complexity.

Acknowledgments We thank Janet McAndless for her help during the acquisition sessions, all the subjects who participated in the project, Peter Sand for the 2D tracking software, Tim Weyrich for helping with the calibration, John Barnwell for his discussions on hardware issues, MERL for providing the acquisition dome, the MIT pre-reviewers, and the anonymous SIGGRAPH reviewers. This work was supported by an NSF CAREER award 0447561. Frédo Durand acknowledges a Microsoft Research New Faculty Fellowship, a Sloan Fellowship, and a generous gift from Adobe.

References

Besl, P. J., and McKay, N. D. 1992. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 14*, 2.

Blais, F. 2004. Review of 20 years of range sensor development. *Journal of Electronic Imaging 13*, 1.

Curless, B., and Levoy, M. 1995. Better optical triangulation through spacetime analysis. In *Proc. of the International Conference on Computer Vision*.

Curless, B. 1997. *New methods for surface reconstruction from range images*. PhD thesis, Stanford University.

Davis, J., Ramamoorthi, R., and Rusinkiewicz, S. 2003. Spacetime stereo: A unifying framework for depth from triangulation. In *Proc. of the Conference on Computer Vision and Pattern Recognition*.

Debevec, P., Hawkins, T., Tchou, C., Duiker, H.-P., Sarokin, W., and Sagar, M. 2000. Acquiring the reflectance field of a human face. In *Proc. of the ACM SIGGRAPH conference*.

Gortler, S. J., Grzeszczuk, R., Szeliski, R., and Cohen, M. F. 1996. The lumigraph. In *Proc. of the ACM SIGGRAPH conference*.

Grabli, S., Sillion, F., Marschner, S. R., and Lengyel, J. E. 2002. Image-based hair capture by inverse lighting. In *Proc. of the Graphics Interface conference*.

Granlund, G. H., and Knutsson, H. 1994. *Signal Processing for Computer Vision*. Springer. ISBN 978-0-7923-9530-0.

Hasinoff, S. W., and Kutulakos, K. N. 2006. Confocal stereo. In *Proc. of the European Conference on Computer Vision*.

Hawkins, T., Einarsson, P., , and Debevec, P. 2005. Acquisition of time-varying participating media. *ACM Transactions on Graphics 24*, 3. Proc. of the ACM SIGGRAPH conference.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W. 1992. Surface reconstruction from unorganized points. In *Proc. of the ACM SIGGRAPH conference*.

Kajiya, J. T., and Kay, T. L. 1989. Rendering fur with three dimensional textures. In *Proc. of the ACM SIGGRAPH conference*.

Kautz, J., Sattler, M., Sarlette, R., Klein, R., and Seidel, H.-P. 2004. Decoupling BRDFs from surface mesostructures. In *Proc. of the Graphics Interface conf.*

Lensch, H., Kautz, J., Goesele, M., Heidrich, W., and Seidel, H.-P. 2001. Image-based reconstruction of spatially varying materials. In *Proc. of the Eurographics Workshop on Rendering*.

Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., Ginzton, M., Anderson, S., Davis, J., Ginsberg, J., Shade, J., and Fulk, D. 2000. The digital Michelangelo project: 3D scanning of large statues. In *Proc. of the ACM SIGGRAPH conference*.

Marschner, S. R., Jensen, H. W., Cammarano, M., Worley, S., and Hanrahan, P. 2003. Light scattering from human hair fibers. *ACM Transactions on Graphics 22*, 3. Proc. of the ACM SIGGRAPH conference.

Matusik, W., Pfister, H., Ngan, A., Beardsley, P., Ziegler, R., and McMillan, L. 2002. Image-based 3D photography using opacity hulls. *ACM Transactions on Graphics 21*, 3. Proc. of the ACM SIGGRAPH conference.

Mihashi, T., Tempelaar-Lietz, C., and Borshukov, G. 2003. Generating realistic humain hair for "the matrix reloaded". In *Proc. of the ACM SIGGRAPH conference*. Technical Sketch.

Narasimhan, S. G., Nayar, S. K., Sun, B., and Koppal, S. J. 2005. Structured light in scattering media. In *Proc. of the International Conference on Computer Vision*.

Paris, S., Briceño, H., and Sillion, F. 2004. Capture of hair geometry from multiple images. *ACM Transactions on Graphics 23*, 3. Proc. of the ACM SIGGRAPH conference.

Pulli, K. 1997. *Surface Reconstruction and Display from Range and Color Data*. PhD thesis, University of Washington.

Reche, A., Martin, I., and Drettakis, G. 2004. Volumetric reconstruction and interactive rendering of trees from photographs. *ACM Transactions on Graphics 23*, 3. Proc. of the ACM SIGGRAPH conference.

Rusinkiewicz, S., Hall-Holt, O., and Levoy, M. 2002. Real-time 3D model acquisition. *ACM Transactions on Graphics 21*, 3 (July), 438–446.

Ward, K., Bertails, F., Kim, T., Marschner, S. R., Cani, M.-P., and Lin, M. 2006. A survey on hair modeling: Styling, simulation, and rendering. *IEEE Transactions on Visualization and Computer Graphics*.

Wei, Y., Ofek, E., Quan, L., and Shum, H.-Y. 2005. Modeling hair from multiple views. *ACM Transactions on Graphics 24*, 3. Proc. of the ACM SIGGRAPH conference.

Wenger, A., Gardner, A., Tchou, C., Unger, J., Hawkins, T., and Debevec, P. 2005. Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM Transactions on Graphics 24*, 3. Proc. of the ACM SIGGRAPH conf.

Weyrich, T., Matusik, W., Pfister, H., Bickel, B., Donner, C., Tu, C., McAndless, J., Lee, J., Ngan, A., Jensen, H. W., and Gross, M. 2006. Analysis of human faces using a measurement-based skin reflectance model. *ACM Transactions on Graphics 25*, 3. Proc. of the ACM SIGGRAPH conference.

Zhang, L., Curless, B., and Seitz, S. M. 2003. Spacetime stereo: shape recovery for dynamic scenes. In *Proc. of the conf. on Computer Vision and Pattern Recognition*.

Zhang, L., Snavely, N., Curless, B., and Seitz, S. M. 2004. Spacetime faces: High-resolution capture for modeling and animation. *ACM Transactions on Graphics 23*, 3. Proc. of the ACM SIGGRAPH conference.

Zhang, Z. 2000. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 11.