Eurographics Symposium on Rendering (DL-only Track) (2021)
A. Bousseau and M. McGuire (Editors)

Human Hair Inverse Rendering using Multi-View Photometric data

Tiancheng Sun[1], Giljoo Nam[2], Carlos Aliaga[2], Christophe Hery[2], Ravi Ramamoorthi[1]

[1]University of California, San Diego
[2]Facebook Reality Labs

Abstract

We introduce a hair inverse rendering framework to reconstruct high-fidelity 3D geometry of human hair, as well as its reflectance, which can be readily used for photorealistic rendering of hair. We take multi-view photometric data as input, i.e., a set of images taken from various viewpoints and different lighting conditions. Our method consists of two stages. First, we propose a novel solution for line-based multi-view stereo that yields accurate hair geometry from multi-view photometric data. Specifically, a per-pixel lightcode is proposed to efficiently solve the hair correspondence matching problem. Our new solution enables accurate and dense strand reconstruction from a smaller number of cameras compared to the state-of-the-art work. In the second stage, we estimate hair reflectance properties using multi-view photometric data. A simplified BSDF model of hair strands is used for realistic appearance reproduction. Based on the 3D geometry of hair strands, we fit the longitudinal roughness and find the single strand color. We show that our method can faithfully reproduce the appearance of human hair and provide realism for digital humans. We demonstrate the accuracy and efficiency of our method using photorealistic synthetic hair rendering data.

1. Introduction

With emerging technologies of augmented reality and virtual reality, creating photorealistic digital humans is gaining increasing attention in computer vision and graphics. Inverse rendering is a widely used technique that can alleviate 3D artists' labor-intensive tasks of modeling high-fidelity digital humans. For human inverse rendering, there have been several successful approaches to build capture systems that produce multi-view photometric data, which is a set of images taken from various viewpoints and under different lighting conditions. From the images, inverse rendering framworks are able to reconstruct detailed geometry as well as complex material appearance of human skin, e.g., specularity and subsurface scattering. While previous research on digital humans achieved great success on faces and bodies, inverse rendering for high-fidelity hair data is still an open problem due to the inherent characteristics of hair, i.e., the microscale geometry and the large number of hair strands. In this paper, we present a novel inverse rendering framework to reconstruct detailed hair geometry as well as its reflectance, which can be readily used for photorealistic rendering of digital humans.

Following recent successes of human inverse rendering research, we use multi-view photometric data. As a first stage, we reconstruct hair geometry with strand-level accuracy. Traditional multi-view stereo techniques fail in this stage as they are designed to reconstruct 3D surfaces, not 3D strands. The main challenge is to find pixel-wise correspondences across views. Nam et al. [NWKS19] proposed a line-based multi-view stereo algorithm that reconstructs a 3D line cloud from multi-view images. Their method, however, requires densely positioned cameras (70 cameras on a hemisphere) and relies on several heuristics for correspondence matching and view-selection. We propose a new solution for line-based multi-view stereo that gives us more densely reconstructed strands even from a smaller number of views compared to the previous work [NWKS19] (Section 4). We design a new matching cost function that fully utilizes the photometric data. We introduce lightcode,

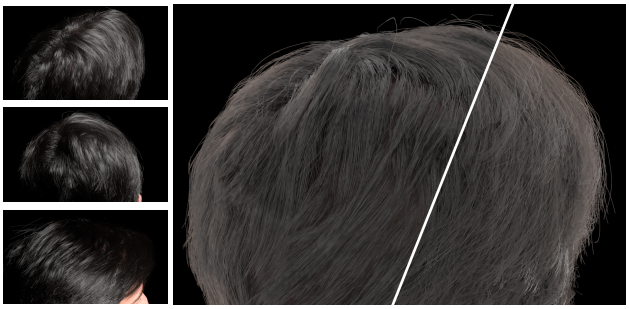

Figure 1: *Demonstration of our method on synthetic hair rendering data. (Left) Some of our input images. (Right) A rendering of our reconstructed hair model. Our hair inverse rendering algorithm can produce a full hair model with geometry and reflectance information, which is accurate and realistic.*

DOI: 10.2312/sr.20211301

a per-pixel light encoding that stores information about whether a small hair segment can be lit by each light or not. Using the lightcode, we efficiently solve the hair correspondence problem as well as the per-pixel view selection problem.

Using the reconstructed hair geometry, the second stage estimates reflectance properties using photometric data (Section 5). Estimating the reflectance of hair is challenging, since the overall appearance of hair is the result of the aggregated multiple interactions of light with each single hair fiber of about 80 microns thickness. Each of those fibers exhibits highly anisotropic, complex light scattering patterns. To estimate the full reflectance properties of hair strands, a highly complex and sophisticated capture system with a microscope is needed [KM17]. Therefore, we propose a practical solution to use a simple hair reflectance model and fit the parameters using our reconstructed geometry and illumination information.

We demonstrate our method using synthetic hair rendering data (Fig. 1). We use high-quality human head models and hair strand models that are created by digital artists and render photorealistic images using Blender. To mimic a real-world capture environment, we place multiple virtual cameras and light sources on a hemisphere pointing towards a subject, similar to the LightStage [GLD*19].

In summary, we introduce the following contributions:

- A hair inverse rendering framework using multi-view photometric data that yields high-fidelity hair geometry and its reflectance, which can be used for photorealistic rendering of hair on virtual characters.
- A novel solution for line-based multi-view stereo that yields accurate hair geometry from multi-view photometric data. By using lightcode, our method can reconstruct denser strands with sparser cameras compared to state-of-the-art work. (Section 4)
- Hair reflectance estimation using multi-view photometric data. Given the estimated fiber direction and a set of light and camera vectors, we leverage these sparse samples of the hair BSDF to estimate the multiple scattering albedo and the longitudinal roughness that define the overall color of the hair and the width of the highlights respectively. (Section 5)

2. Related Work

2.1. Multi-view Photometric Stereo

Multi-view stereo [HZ03] has been widely used to reconstruct 3D geometry from images taken from multiple viewpoints under a fixed lighting condition. On the other hand, photometric stereo [Woo80] reconstructs surface normals, as well as surface reflectance, using multiple images taken from a fixed viewpoint but under varying illumination. Multi-view photometric stereo combines the two techniques to get rich information of 3D geometry and surafce reflectance at the same time [SDR*20; LMC19; PSM*16; ZWT13; YY11; HVC08; TFG*13; DYW05; KN08]. The multi-view and multi-illumination setup has proven to be effective for creating photorealistic digital humans. However, most work has focused on faces [FJA*14; FGT*16; GFT*11] and bodies [ZFT*20; VPB*09; GLD*19], and is thus not applicable to hair.

2.2. Image-based Hair Modeling

There have been many efforts to model hair geometry using captured images. Please refer to [BQ18] for an overview. A common approach is to use a multi-view setup. Most existing methods first obtain a rough geometry from structured-light [PCK*08], visual hull [PBS04; WOQS05], multi-view stereo [LLP*12; LLR13; XWW*14; HMLL14; HML*14; HBLB17; EBGB14; BBN*12], depth-from-focus [JMM09], thermal video cameras [HZW12], or an RGB-D sensor [HML*14], and run an additional strand-fitting step to get 3D hair strands. Recently, Nam et al. [NWKS19] presented a line-based multi-view stereo method that directly reconstructs 3D hair strands from images, thus achieving high accuracy. Paris et al [PCK*08] showed the feasibility of a hair inverse rendering pipeline using multi-view and multi-light images. They first reconstruct 3D geometry of hair using structured light patterns and estimate its reflectance using controlled illumination. Different from [PCK*08], we reconstruct hair strands directly from multi-view photometric data using our novel solution.

Hair capture from a single image is another stream of research [HMLL15; HML*14; CWW*12; CWW*13; CLS*15; CSW*16; ZCW*17]. These are data-driven methods that utilize a large number of manually created 3D hair models. More recently, deep learning-based approaches were proposed. Pre-trained networks, such as convolutional neural networks [ZHX*18] or generative adversarial networks [ZZ19], were used to infer 3D hair geometry from a single image. While these methods have the benefit of easy capture, the reconstructed geometries are not suitable for photorealistic rendering.

2.3. Hair Reflectance Model

The seminal work of Marschner et al. [MJC*03] described the complex light transport within each human hair fiber, representing it as a rough dielectric cylinder. This work resulted in the classification of different scattering modes, namely **R** for the primary reflectance, **TRT** for a secondary colored and attenuated reflection offset from the primary, and **TT** for a transmission term, responsible for most of the observed hue of the hair. D'Eon et al. [dFH*11] refined this far field model into a more principled spherical Gaussian for the longitudinal (along the fiber) component, and supplied a separate roughness control on the azimuthal (in the cross sectional plane of the cylinder) term. More recently, Chiang et al. [CBTB15] returned to a near field solution, and provided a fourth **TRRT** lobe to cover the missing energy from the additional internal bounces not considered previously. We start from a base implementation of this model in Blender [Ble20] and add explicit artistic and debugging factors for each term, also improving the convergence via fits similar to the approaches from Pekelis et al. [PHVL15].

Previous work has tried to match the appearance of hair from a single image [BPV*09], but accuracy is limited because they do not utilize accurate strand level hair geometry. Zinke et al. [ZRL*09] proposed to measure the reflectance properties of hair fibers coiled around a cylinder. Progress has also been made in the case of textile fibers [KSZ*16] based on volumetric representations of fiber assemblies. Different from previous work, we rely on explicit fiber level estimation of hair geometry and do not depend on the usual

limitations of traditional hair reflectance models, being able to model any kind of color coming from dyes, or extra coating or wetness layers.

3. Overview

An overview of our hair inverse rendering pipeline is shown in Fig. 2. The input of our framework is multi-view photometric data, i.e., a set of images taken from various view-points and different lighting conditions. We assume that N_c cameras are pointing towards the subject and there are N_l point light sources ($N_l \geq N_c$). We also assume that each camera has a co-located light source. Given $N_c \times (N_l + 1)$ images captured from all cameras under each light source (and additional uniform lighting), we first reconstruct a dense set of hair strands using the theory of multi-view photometric stereo (Section 4), and then estimate the reflectance properties of the hair strands using the inferred geometry (Section 5).

We develop our algorithm based on synthetically "captured" data. We use high-quality human head models and hair strand models created by digital artists, and use Blender to render the input images. Specifically, we place $N_c = 24$ cameras on the upper hemisphere, aiming at the hair region, and then distribute $N_l = 36$ point lights uniformly on the sphere. We make sure that each camera has a co-located point light source. Using synthetic data has the following advantages. First, we have access to the ground-truth hair strands which is impossible to obtain in real captured data. Second, we can efficiently control the dataset. For example, we can easily change the hair reflectance parameters, the number of cameras and lights, exposure settings, etc. We render the images using the default setting of the Blender Cycles renderer [Ble20], which uses path tracing with 1024 samples per pixel. All the images are rendered with the resolution 2048×2048, where a single hair strand is roughly 3 pixels wide.

4. Hair Geometry Reconstruction

Our geometry reconstruction algorithm is inspired by the recent success of Nam *et al.* [NWKS19]. They proposed the line-based multi-view stereo (LMVS) to reconstruct strand-accurate hair geometry using multi-view captures. However, due to multiple scattering effects inside the hair, strands cannot be easily distinguished from captured images. As a result, a large number of cameras (70 views in [NWKS19]) are needed for satisfactory output, and only the outer surface of the hair is recovered.

We therefore use a multi-view photometric stereo (MVPS) setup to solve the problem and propose a new solution for LMVS that fully exploits the photometric data. In particular, we introduce *lightcode*, a novel per-pixel light encoding structure. *lightcode* enables efficient neighbor view selection for multi-view stereo, and also provides a strong signal for robust correspondence matching across views. The remainder of this section describes our new solution for LMVS using MVPS data. We kindly refer readers to [NWKS19] for more details about basic LMVS and to [CLK09; BS05; BFV05] for more traditional computer vision approaches that use lines.

4.1. 2D Orientation Map

4.1.1. Per Light Orientation Extraction

Similar to previous work [PBS04], we filter each photometric image $\mathbf{I}_c(l,x,y)$ under a point light l and camera c with a set of convolutional kernels, and find the per pixel 2D orientation from the maximum response. We use log-gabor filters [FŠP*07] rather than gabor filters, because they have finer frequency supports:

$$\begin{aligned} \mathbf{R}_\theta(l,x,y) &= \sigma_\theta * \mathbf{I}_c(l,x,y) \\ \Theta_c(l,x,y) &= \operatorname*{argmax}_\theta \mathbf{R}_\theta(l,x,y) \\ \mathbf{W}_c(l,x,y) &= \frac{\sum_\theta \mathbf{R}_\theta(l,x,y) \cdot \cos(\theta - \Theta_c(l,x,y))}{\max_\theta \mathbf{R}_\theta(l,x,y)}, \end{aligned} \tag{1}$$

where $\mathbf{R}_\theta(l,x,y)$ is the convolutional response at pixel (x,y) with a θ-rotated log-Gabor filter σ_θ. $\Theta_c(l,x,y)$ and $\mathbf{W}_c(l,x,y)$ are the 2D orientation map and the confidence map for image $\mathbf{I}_c(l,x,y)$.

4.1.2. Per View Orientation Merge

After we extract the 2D orientation and its confidence from each photometric image under a specific camera c, we apply a weighted average on the per light 2D orientation and confidence:

$$\begin{aligned} \Theta_c(x,y) &= \sum_l \mathbf{W}_c(l,x,y) \cdot \Theta_c(l,x,y) \\ \mathbf{W}_c(x,y) &= \sum_l \mathbf{W}_c(l,x,y) \cdot \cos(\Theta_c(l,x,y) - \Theta_c(x,y)), \end{aligned} \tag{2}$$

where the output is per-view 2D orientation map $\Theta_c(x,y)$ and confidence map $\mathbf{W}_c(x,y)$. Please refer to the supplemental material for more details about the effect of the log-Gabor filter compared to the original Gabor filter.

4.1.3. Hair Region Masking

We also use a pre-trained deep neural network [YWP*18; LLWL20] to extract the hair region from the uniformly-lit images. We compute 2D orientation maps only for the valid hair region, which has great benefits in fast computation and efficient outlier removal.

4.2. 3D Line Reconstruction

4.2.1. Lightcode

In multi-view stereo (MVS), finding correct correspondences across views is crucial for accurate geometry reconstruction. Traditional methods fail in hair because first, the intensities between neighboring pixels are highly similar, and second, due to the microstructures of hair strands, the pixel values change rapidly with different viewpoints. 2D orientation maps $\Theta(x,y)$ can be used for correspondence matching [NWKS19], but it is inefficient as it involves multiple projections between cameras.

We therefore propose to use *lightcode* $\mathbf{L}_c(x,y)$ as a matching feature to find correspondences between multiple views:

$$\mathbf{L}_c(x,y) = b_1 b_2 \cdots b_{N_l}, \tag{3}$$

where b_i is a binary bit which represents the visibility of the

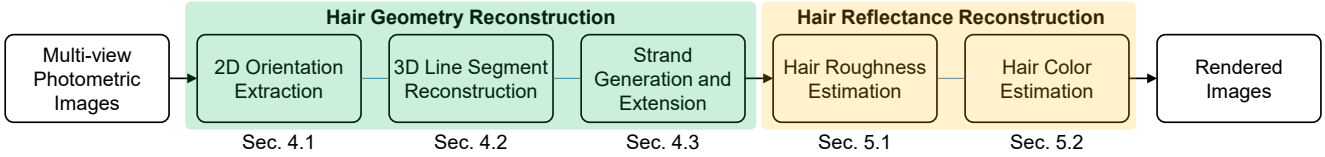

Figure 2: *An overview of our hair inverse rendering pipeline.*

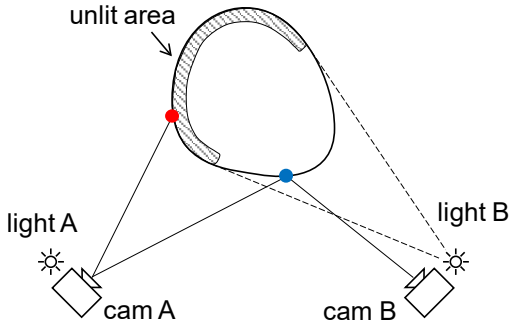

Figure 3: *This figure illustrates how lightcode can be used for view selection. We use the interchangeability of the camera rays and light rays. If one pixel of camera A under light B is bright, it means that pixel can also be observed by camera B (blue point). If another pixel of camera A under light B is dark, it indicates that camera B cannot see the pixel (red point).*

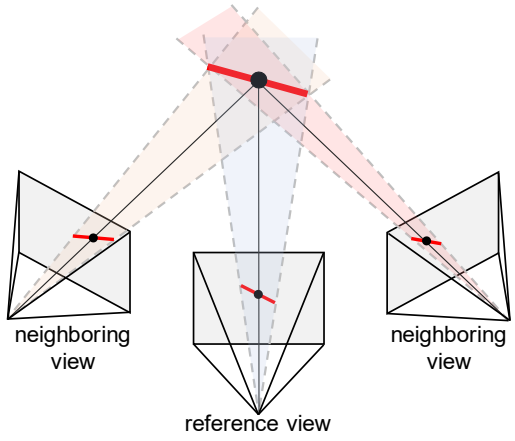

Figure 4: *An illustration of how we analytically derive the 3D line segment S from a given depth d. For a pixel in the reference view, we first get S_{pos} (the position of S) by shooting a ray toward the pixel. Then we get S_{dir} (the 3D direction of S) using Eq. 4.*

pixel under light i. The lightcode shares a similar idea as Space-time Stereo [DRR03], where they use pixel values from different time slices for matching between cameras. Here, we assume that whether a strand is bright or not is independent of camera views, though its intensity is view-dependent. We use the percentage of matching bits between the lightcodes to evaluate the similarity between cameras. We found that the lightcode could find the corresponding pixels in the neighboring cameras efficiently and accurately. In practice, we use the median pixel value of the image under the furthest light as the shadow threshold of the camera.

4.2.2. View Selection using Lightcode

In MVS, neighboring view selection also affects efficiency and accuracy of the algorithm. Due to the lack of visibility information, traditional MVS algorithms, including [NWKS19], have to search for matching pixels in all the neighboring views. Using *lightcode*, we can extract the visibility information since a point light is placed right next to each camera. As shown in Fig 3, if one pixel of camera A under light B is bright, then it means that specific pixel can also be observed by camera B, and vice versa. Here, we use the interchangeability of the camera rays and light rays. The shadow casted by the light next to a camera represents the visibility information when observed from other views. We use this visibility information to select per-pixel valid neighboring views. For the remaining parts of this section, 'neighbor view' always refers to 'per-pixel valid neighbor view'.

Essentially, the lightcode encodes the visibility information at each 3D point. Since the human head is roughly a sphere, about half of the bits in each lightcode are 0. This means that for each point associated with a lightcode, a large number of cameras cannot see this point. In order to prune these invalid observations, we only use the nearest 16 cameras when we are doing 3D reconstruction.

4.2.3. Analytic Derivation of 3D Line Segment from Pixel Depth

Given the 2D information (orientation map $\Theta(x,y)$, confidence $\mathbf{W}(x,y)$, and lightcode $\mathbf{L}(x,y)$) of all cameras, our goal is to find the per-pixel optimal line segment S in the 3D space, represented by a 3D position S_{pos} and a 3D line direction S_{dir}. We observe that both S_{pos} and S_{dir} can be analytically derived when the depth d of the pixel is given. For a single pixel (x,y) in camera c, assuming we know its depth d, we can find the 3D location of its corresponding segment S_{pos} and project it to its visible neighbors. For each n-th neighboring camera, if the 3D line direction S_{dir} is consistent with the 2D line direction $\Theta_n(x_{proj},y_{proj})$, the 3D line should lie in the plane formed by the projection ray and the 2D line on the image plane. In other words, we will get the 3D line direction S_{dir} by solving the equation below:

$$\begin{bmatrix} \text{Ray}_0(x_{proj},y_{proj}) \times \Theta_0(x_{proj},y_{proj}) \\ \vdots \\ \text{Ray}_{N_n}(x_{proj},y_{proj}) \times \Theta_{N_n}(x_{proj},y_{proj}) \end{bmatrix} \cdot S_{dir} = 0, \quad (4)$$

where N_n is the number of neighbor views and $n = 0$ indicates the reference view. Here, we use Θ_n to represent the 2D line vector on

the image plane of the n-th neighboring camera. The equation can be efficiently solved for S_{dir} by finding the singular vector corresponding to the smallest singular value. Figure 4 illustrates this procedure. In this way, we reduce the whole problem to a 1D searching problem w.r.t. the pixel depth d.

4.2.4. Objective Function

Now, the problem is reduced to a 1D search over the depth d for each pixel. To formulate our objective function, we sample $N_s = 25$ points along the 3D direction S_{dir} of the segment S. The sampled points are represented by the superscript s. Our objective function is a multiplication of two sub functions, the lightcode correlation term $\mathcal{O}_{lightcode}$ and the geometric correlation term $\mathcal{O}_{\Theta}$:

$$\begin{aligned} \mathcal{O}_{lightcode} &= m_{\mathbf{L}}(\mathbf{L}_c(x,y), \mathbf{L}_n(x^s_{proj}, y^s_{proj})), \\ \mathcal{O}_{\Theta} &= \cos(\mathrm{diff}(\Theta_n(S^s_{dir}), \Theta_n(x^s_{proj}, y^s_{proj}))) \\ &\quad \cdot \mathbf{W}_n(x^s_{proj}, y^s_{proj}), \end{aligned} \tag{5}$$

where s indicates the sampled 3D points, n refers to n-th view, $m_{\mathbf{L}}()$ evaluates the similarity between two lightcodes, (x,y) is the target pixel in the reference view, (x^s_{proj}, y^s_{proj}) is the pixel coordinate where the sample s is projected to the n-th view, diff returns the 2D angle difference, and $\Theta_n(S_{dir})$ is the 2D line direction of the 3D line segment when projected to the n-th view.

Finally, we find optimal depth d per pixel, which maximizes the following objective:

$$\mathcal{O} = \sum_{s=1}^{N_s} \sum_{n=0}^{N_n} \mathcal{O}_{lightcode} \cdot \mathcal{O}_{\Theta}, \tag{6}$$

where N_s is the number of samples, and N_n is the number of neighbor views. Note that $n = 0$ indicates the reference view. This objective function represents the overall correlation across views. We find depth d that maximizes the correlation.

4.2.5. Optimization

Optimizing Eq. 6 is straightforward. For each pixel on the reference view, we enumerate possible depth candidates and calculate $\mathcal{O}$ using Eq. 4 and Eq. 5. Then we select the depth d that returns maximum $\mathcal{O}$. This brute force search guarantees to find the global optimum, but it can be slow. We therefore run a two-stage optimization. First, for each depth candidate, we only evaluate $\mathcal{O}_{lightcode}$ in Eq. 6 and collect the depth candidates that produce high lightcode matching. Then we evaluate the full objective $\mathcal{O}$ only for the selected candidates. We find that this two-stage optimization enables faster correspondence matching while preserving the output quality.

4.2.6. Comparison to the Previous Work

Previous work by Nam et al. [NWKS19] treats the optimization as a multi-dimensional problem, as both the depth d and the 3D line direction S_{dir} are unknown. In contrast, we treat it as a 1D problem with a single unknown variable d. This makes our algorithm faster and more stable compared to [NWKS19].

Another key difference between our objective function and the one from Nam et al. [NWKS19] is that we are solving a maximization problem rather than a minimization problem. This has the following two advantages. First, during optimization, we are not only finding the best matching pixels, but also maximizing the number of pixels that correspond to a strand (large $\mathbf{W}$ value). Second, by formulating the problem as maximization, we implicitly set the value of "bad matching" and "non-valid matching" to be zero, which leads to more stable optimization.

4.3. Strand Generation and Extension

After we generate the line segments from each pixel in each camera, we follow the approach of [NWKS19] to connect them into strands. We first select the small line segments S that are consistent in at least 3 neighboring cameras, and then apply the mean shift algorithm on the line cloud. The mean shift algorithm efficiently collects the small line segments into a long strand in the 3D space. After that, we cluster the neighboring line segments into strands.

At the current stage, we arrive at a set of hair strands. However, the connected strands do not represent the hair geometry very well. The average strand length is usually under 1 cm. Thus, we further extend each hair strand from its tips to get longer and more reasonable hair strands. For each tip P, we project the 3D point X_P back to each camera, and select the cameras where the projected 3D direction of P aligns with the 2D direction at the projected point $\Theta_i(x_{proj}, y_{proj})$ (angle different less that 5°). Using the aligned 2D directions from valid cameras, we can compute the possible growing direction S_{dir} by following Eq. 4. After we solve for the correct growing direction S_{dir}, we extend the strand for a certain step $X_P = X_P + S_{dir} \cdot d_{step}$, where $d_{step} = 0.05cm$. We repeat this process until the new tip point can find no more than 5 cameras that align with our 3D direction S_{dir}.

5. Hair Reflectance Estimation

Hair reflectance is difficult to estimate due to the aggregated nature of hair appearance. Light may bounce many times inside the hair volume, especially in the case of blonde or other kinds of lightly colored hair. In order to match the look of the captured image, we need to accurately estimate the reflectance properties of the single fiber. There are two properties that affect the visual appearance the most: the longitudinal roughness of the hair and the absorption coefficient. The longitudinal roughness β_l controls the size of the highlight on the hair strands, and the absorption coefficient σ determines how much light is absorbed by each single fiber, thus controlling the color of the hair strands.

5.1. Hair roughness estimation

The longitudinal roughness is one of the parameters of the hair reflectance model, which is usually expressed as a BSDF function. We use the BSDF model of Chiang et al. [CBTB15] in our paper. The general form of the BSDF model can be written as:

$$\rho(\theta_i, \theta_r, \phi_i, \phi_r) = \rho_l(\theta_i, \theta_r, \beta_l) \cdot \rho_a(\theta_i, \theta_r, \phi_i, \phi_r, \beta_a), \tag{7}$$

where ρ_l is the longitudinal component and ρ_a is the azimuthal component. The meanings of angles are shown in Fig. 5. Longitudinal roughness β_l determines the width of white (R) and all the subsequent colored lobes (TT, TRT, TRRT). In contrast, the azimuthal roughness β_a affects the overall translucency of the hair

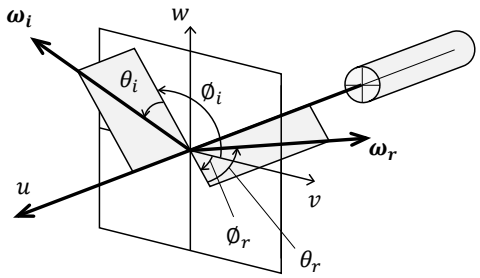

Figure 5: *Parametrization of the hair BSDF. θ_i and θ_r are the longitudinal angles, and ϕ_i and ϕ_r are the azimuthal angles.*

volume. In order to keep the problem tractable, we decide to only optimize the longitudinal roughness β_l since it is ultimately what controls the most prominent visual features of hair: the length of the highlights on the strands. For this, we fix the azimuthal roughness β_a to common plausible values for human hair $\beta_a = 0.2$ [MJC*03; YY97; Bhu08], and assume that the azimuthal component ρ_a is a constant. The analytical BSDF function of the longitudinal component is expressed as:

$$\begin{aligned} \rho_l(\theta_i, \theta_r, \beta_l) &= \frac{\exp(-S - 1/\beta_l) \cdot I}{\beta_l \cdot (1 - \exp(-2/\beta_l))} \\ \text{where } I &= I_0(\cos(\theta_i) \cdot \cos(\theta_r)/\beta_l) \\ S &= \sin(\theta_i) \cdot \sin(\theta_r)/\beta_l \end{aligned} \tag{8}$$

I_0 is the modified Bessel function of the first kind, order 0.

In order to fit to the BSDF model and solve for the longitudinal roughness, we collect the BSDF samples of the hair reflectance by collecting the pixel values whose viewing and lighting directions are close to the mirror reflection:

$$\begin{aligned} \Delta\theta &= |\theta_i - \theta_r| < \frac{\pi}{6}, \\ \Delta\phi &= |\phi_i - \phi_r| < \frac{\pi}{6}. \end{aligned} \tag{9}$$

We divide the range of $\Delta\theta \in [0, \frac{\pi}{6}]$ into 32 bins, and average the BSDF samples in each bin. Suppose the measured BSDF is $\rho_m(\Delta\theta)$, we find the longitudinal roughness of the hair strands by solving the following optimization:

$$\max_{\beta_l} \frac{\sum_{\Delta\theta} \rho_m(\Delta\theta) \cdot \rho_l(\Delta\theta, \beta_l)}{\sqrt{\sum_{\Delta\theta} \rho_l^2(\Delta\theta, \beta_l)}}. \tag{10}$$

Figure 6 shows a measured BSDF and the fitted analytical BSDF of the longitudinal components. The shape of the measured BSDF follows the analytical one when $\Delta\theta$ is small, as the BSDF is dominated by the direct reflection (R component). As $\Delta\theta$ becomes larger, the measured BSDF becomes flat, due to the multiple scattering between hair strands. For this reason, we only do our fitting on $\Delta\theta \in [0, \frac{\pi}{6}]$.

5.2. Hair color optimization

The overall perceived color of the hair, often called multiple scattering albedo, not only depends on the absorption coefficient σ of each

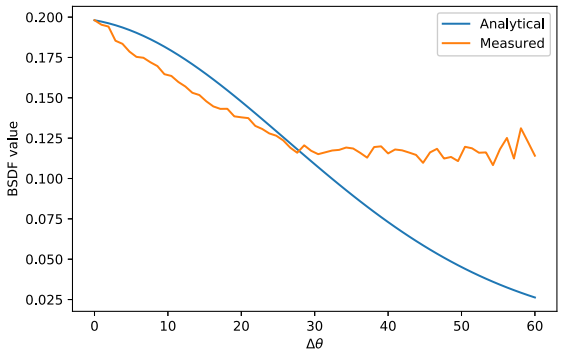

Figure 6: *The measured and the corresponding fitted analytical BSDF of the longitudinal components. Our measured BSDF follows the analytical BSDF when $\Delta\theta$ is small, and becomes flat when $\Delta\theta$ is large, due to multiple scattering in the hair strands.*

Algorithm 1: Hair Color Optimization

Function `HairColorOpt` (*Captured Hair Image I under uniform lighting, IterNum=5*) **:**

- $R_{rgb} \leftarrow$ Ave color of the hair region in I;
- $C^0_{rgb} \leftarrow R_{rgb}$;
- **for** $s = 0 \cdots IterNum - 1$ **do**
  - Rendering the hair image $I^{(s)}$ using hair color parameter $C^{(s)}_{rgb}$;
  - $R^{(s)}_{rgb} \leftarrow$ Ave color of the hair region in $I^{(s)}$;
  - **if** $s == 0$ **then**
    - $C^{(s+1)}_{rgb} \leftarrow \frac{R_{rgb}}{R^{(s)}_{rgb}} \cdot C^{(s)}_{rgb}$;
  - **else**
    - Perform linear fitting $R^{(s)}_{rgb} = a \cdot C^{(s)}_{rgb} + b$ using $R^{(s)}_{rgb}$ and $C^{(s)}_{rgb}$ from 0 to s;
    - $C^{(s+1)}_{rgb} \leftarrow \frac{R_{rgb} - b}{a}$;
- **return** $C^{(IterNum)}_{rgb}$;

of the single fibers, but also on the azimuthal roughness β_a, that determines the translucency of the hair volume. However, the multiple scattering albedo is invariant to hair density[CBTB15; ZW07], given a dense enough volume of strands. Chiang *et al.* [CBTB15] empirically linked the multiple scattering hair albedo C_{rgb} to the single fiber absorption coefficient σ and the azimuthal roughness β_a using the following formula:

$$\begin{aligned} \sigma = &(\log C_{rgb}/(5.969 - 0.215\beta_a + 2.532\beta_a^2 \\ &- 10.73\beta_a^3 + 5.574\beta_a^4 + 0.245\beta_a^5))^2. \end{aligned} \tag{11}$$

This equation gives us a good approximation of the absorption coefficient σ given a fixed azimuthal roughness β_a and an RGB hair color C_{rgb}. This enables us to optimize for an RGB hair color C_{rgb} instead of the absorption coefficient σ.

We develop an iterative algorithm to find the color C_{rgb} that makes the color of the rendered hair match the color of the captured hair images. We present the pseudo code of our hair color optimization in Algo. 1. Given the captured hair image I under uniform lighting, we first extract the average color R_{rgb} of its hair region, and initialize the color parameter C^0_{rgb} as R_{rgb}. Then, we compute the absorption coefficient using the color parameter C^0_{rgb}, render the image of the hair $I^{(0)}$, and compute the average color of the hair region R^0_{rgb} on the rendered image. We iteratively update the color parameter $C^{(s)}_{rgb}$ in order to close the gap between the rendered hair color $R^{(s)}_{rgb}$ and the groundtruth hair color R_{rgb}.

We assume a simple linear model on the relation between the color of the rendered hair $R^{(s)}_{rgb}$ and the hair color parameter $C^{(s)}_{rgb}$:

$$R^{(s)}_{rgb} = a \cdot C^{(s)}_{rgb} + b, \tag{12}$$

where s is the iteration step number, and $R^{(s)}_{rgb}$ is the hair color of the rendered image. This assumption is based on the observation that the pixel colors of the hair region consist of the constant color part that does not change with the hair color parameter (mainly the scalp color), and the global hair albedo that is mostly linear in the color parameter $C^{(s)}_{rgb}$ thanks to Eq. 11. As we observe more $R^{(s)}_{rgb}$ and $C^{(s)}_{rgb}$ pairs, we can perform fitting based on Equ. 12, and solve the optimal C_{rgb} using the groundtruth hair color R_{rgb}. Our algorithm can find the correct color parameter within $3 \sim 5$ iterations.

6. Evaluation

6.1. Previous Work Comparison

We compare our algorithm with the current state-of-the-art work on hair geometry reconstruction [NWKS19]. [NWKS19] can reconstruct strand-level accurate geometry of the hair given a dense set of cameras ($N_c = 70$), while our algorithm is designed to use additional lighting for recovering both the geometry and the reflectance from a sparse set of cameras ($N_c = 24$). We run both of the algorithms on our synthetic dataset, and we use the uniformly lit images from our cameras ($N_c = 24$) as the input to [NWKS19]. Figure 7 shows visual comparison of the reconstructed strands. Our algorithm can reconstruct denser and more complete hair strands compared to [NWKS19]. This is because [NWKS19] relies on a large number of camera views to eliminate outliers (non-visible camera), while our algorithm utilizes the cues from lightings to select valid camera candidates for 3D reconstruction.

Table 1 shows a quantitative comparison on geometry reconstruction errors following the metric in [NWKS19]. Precision (a.k.a. accuracy) and recall (a.k.a. completeness) values are computed using the groundtruth hair geometry. τ_p and τ_d are thresholds for estimated position and direction of 3D points. We validate 3D points if they satisfy both τ_p and τ_d. F-score is defined as harmonic mean of precision and recall. As shown in Table 1, our method can reconstruct hair strands more accurately and completely compared to the previous work.

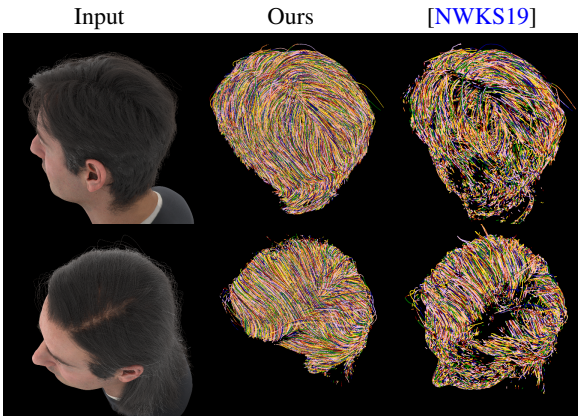

Figure 7: *Compared to Nam et al. [NWKS19] which uses $N_c = 70$ cameras in the original paper, our algorithm better reconstructs the hair geometry under sparse camera setups ($N_c = 24$).*

6.2. Results

We now show our full results of hair inverse rendering. Given a set of photometric hair images, we run our algorithm to get the geometry of hair strands, the hair roughness parameter β_l, and the hair color C_{rgb}. We then render the hair overlaid on a predefined human head model using the reconstructed geometry and reflectance information. Rendering results of various hairstyles under uniform light and point lights are shown in Fig. 10. Our algorithm can handle different hair styles, and the re-rendered results match with the groundtruth on hair color, highlight shape, as well as the overall hair styles.

6.2.1. Hair Roughness

We run our algorithm on a set of hair examples which only differ in their longitudinal roughness, and the reconstructed results are shown in Fig. 8. As we decrease the roughness, it becomes harder to reconstruct the hair geometry since the highlights on the hair strands becomes sharper and shorter (bottom-left corner). However, in all three cases, our algorithm could estimate the longitudinal roughness of the hair strands fairly well, and the highlight shape of the reconstruction matches the groundtruth image in its overall appearance.

6.2.2. Hair Color

We additionally test our hair color optimization component by reconstruction of hair with different colors. As shown in Fig. 9, our algorithm could reconstruct the fine geometry of the hair consistently well, regardless of the hair color. Moreover, our algorithm could accurately recover the original hair color well.

6.3. Runtime

Our algorithm is implemented in Python and CUDA, and the experiments are all tested using a single NVIDIA Quadro P6000 GPU. It takes around 30 minutes to apply our inverse rendering algorithm on a set of synthetic hair images. For hair geometry reconstruction,

	Short hair						Long hair					
τ_p/τ_d	1mm / 10°		2mm / 20°		3mm / 30°		1mm / 10°		2mm / 20°		3mm / 30°	
Method	Nam'19	Ours	Nam'19	Ours	Nam'19	Ours	Nam'19	Ours	Nam'19	Ours	Nam'19	Ours
Precision	**46.36**	42.33	78.80	**86.60**	85.71	**96.19**	**33.42**	32.36	69.88	**75.70**	85.85	**88.56**
Recall	3.53	**15.25**	12.07	**44.99**	23.86	**67.52**	3.83	**25.03**	9.48	**57.75**	16.69	**73.96**
F-score	6.56	**22.42**	20.94	**59.22**	37.32	**79.35**	6.87	**28.22**	16.70	**65.52**	27.95	**80.60**

Table 1: *Precision and recall of the two datasets in Figure 7 with various threshold values. Our method outperforms the previous work [NWKS19] in most threshold values.*

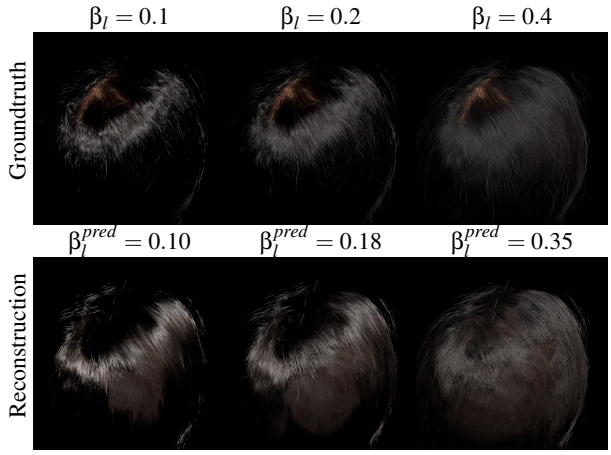

Figure 8: *Our inverse rendering results on hair with different longitudinal roughness values.*

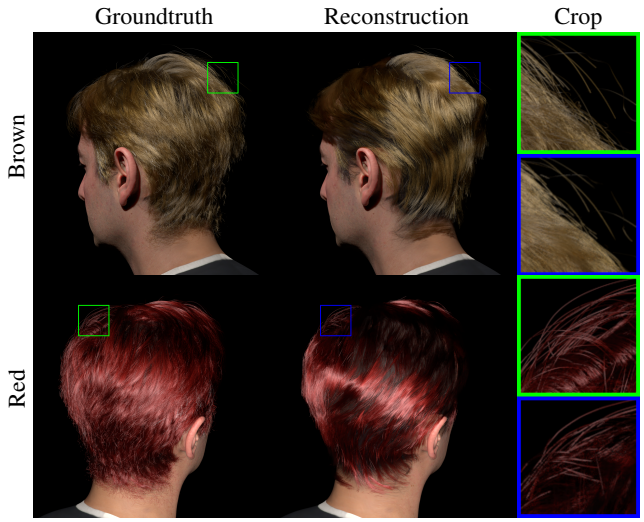

Figure 9: *Our inverse rendering results on hair with different colors.*

2D orientation extraction takes 700 seconds (30 seconds per camera), 3D segment generation needs 340 seconds, and strand generation and extension takes 150 seconds. For hair reflectance estimation, it takes 300 seconds to estimate the roughness, and around 800 seconds for hair color estimation.

7. Conclusions and Future Work

In conclusion, we have proposed a full pipeline for hair inverse rendering given a set of photometric images captured from multiple cameras. This is to our knowledge the first algorithm that both reconstructs the hair geometry and the fine-details of hair reflectance information from only images. We demonstrated that our algorithm can reconstruct the hair geometry better compared to previous works under a sparse camera setup, and also recover the hair reflectance properties from only a sparse set of input images.

Our hair inverse rendering algorithm is not free from limitations. As shown in Fig. 7 and 10, our algorithm yields sub-optimal output for the hair that is curly or long. In addition, as shown in Fig. 9, we tend to underestimate the roughness value when hair strands have saturated color. Another limitation is that we can only reconstruct the outer hair strands that are visible from multiple cameras. A possible future work is to estimate the 3D flow field of the inner region based on priors, and grow the inner hair strands to make the hair style more complete. For reflectance recovery, we only estimate the longitudinal roughness and assume that the azimuthal roughness is constant. A direct next step is to recover the full roughness parameters using densely captured images.

Acknowledgement

This work was supported in part by NSF Chase-CI grant 1730158 and ONR DURIP grant N000141912293 to UC San Diego. This work was also supported by a Facebook Distinguished Faculty Award, A Google PhD Fellowship, the Ronald L. Graham Chair, and the UC San Diego Center for Visual Computing. Thanks to the reviewers for the valuable feedback, to Akira Orikasa and Cyrus Jam for the help on hair modeling, to Olivier Maury for the fruitful discussion, to Ronald Mallet for being the captured subject, and to Tomas Simon for the help on data capture.

References

[BBN*12] BEELER, THABO, BICKEL, BERND, NORIS, GIOACCHINO, et al. "Coupled 3D reconstruction of sparse facial hair and skin". *ACM Transactions on Graphics (ToG)* 31.4 (2012), 117 2.

[BFV05] BAY, HERBERT, FERRARIS, VITTORIO, and VAN GOOL, LUC. "Wide-baseline stereo matching with line segments". *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE. 2005, 329–336 3.

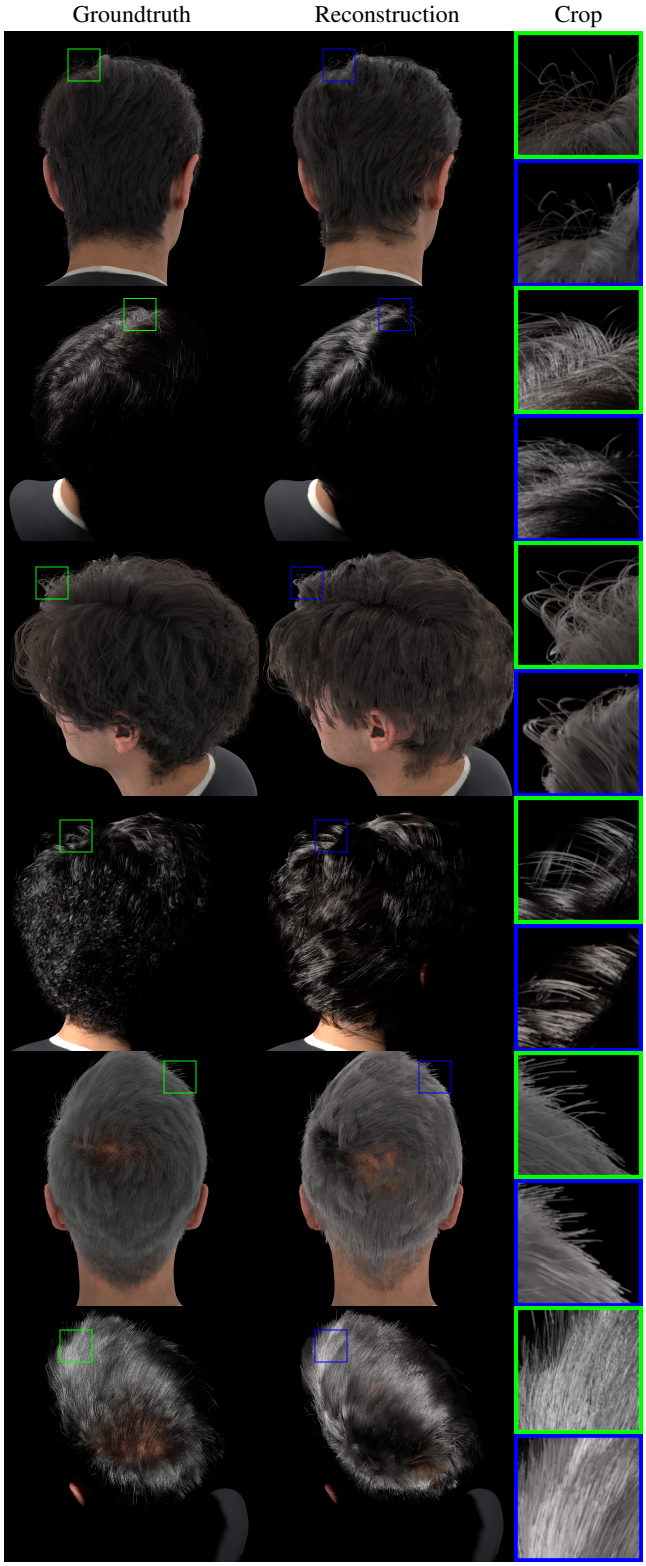

Figure 10: *We apply our hair inverse rendering algorithm on various hairstyles. Our algorithm can faithfully reproduce the appearance of the captured hair.*

[Bhu08] BHUSHAN, BHARAT. "Nanoscale characterization of human hair and hair conditioners". *Progress in Materials Science* 53.4 (2008), 585–710 6.

[Ble20] BLENDER ONLINE COMMUNITY. *Blender - a 3D modelling and rendering package*. Blender Foundation. Blender Institute, Amsterdam, 2020. URL: http://www.blender.org 2, 3.

[BPV*09] BONNEEL, NICOLAS, PARIS, SYLVAIN, VAN DE PANNE, MICHIEL, et al. "Single photo estimation of hair appearance". *Computer Graphics Forum*. Vol. 28. 4. Wiley Online Library. 2009, 1171–1180 2.

[BQ18] BAO, YONGTANG and QI, YUE. "A survey of image-based techniques for hair modeling". *IEEE Access* 6 (2018), 18670–18684 2.

[BS05] BARTOLI, ADRIEN and STURM, PETER. "Structure-from-motion using lines: Representation, triangulation, and bundle adjustment". *Computer vision and image understanding* 100.3 (2005), 416–441 3.

[CBTB15] CHIANG, MATT JEN-YUAN, BITTERLI, BENEDIKT, TAPPAN, CHUCK, and BURLEY, BRENT. "A Practical and Controllable Hair and Fur Model for Production Path Tracing". *ACM SIGGRAPH 2015 Talks*. SIGGRAPH '15. Los Angeles, California: ACM, 2015, 23:1–23:1. ISBN: 978-1-4503-3636-9. DOI: 10.1145/2775280.2792559. URL: http://doi.acm.org/10.1145/2775280.2792559 2, 5, 6.

[CLK09] CHANDRAKER, MANMOHAN, LIM, JONGWOO, and KRIEGMAN, DAVID. "Moving in stereo: Efficient structure and motion using lines". *2009 IEEE 12th International Conference on Computer Vision*. IEEE. 2009, 1741–1748 3.

[CLS*15] CHAI, MENGLEI, LUO, LINJIE, SUNKAVALLI, KALYAN, et al. "High-quality hair modeling from a single portrait photo". *ACM Transactions on Graphics (TOG)* 34.6 (2015), 204 2.

[CSW*16] CHAI, MENGLEI, SHAO, TIANJIA, WU, HONGZHI, et al. "AutoHair: fully automatic hair modeling from a single image". *ACM Transactions on Graphics* 35.4 (2016) 2.

[CWW*12] CHAI, MENGLEI, WANG, LVDI, WENG, YANLIN, et al. "Single-view hair modeling for portrait manipulation". *ACM Transactions on Graphics (TOG)* 31.4 (2012), 116 2.

[CWW*13] CHAI, MENGLEI, WANG, LVDI, WENG, YANLIN, et al. "Dynamic hair manipulation in images and videos". *ACM Transactions on Graphics (TOG)* 32.4 (2013), 75 2.

[dFH*11] D'EON, EUGENE, FRANCOIS, GUILLAUME, HILL, MARTIN, et al. "An Energy-Conserving Hair Reflectance Model". *Computer Graphics Forum*. Vol. 30. Wiley Online Library. 2011, 1181–1187 2.

[DRR03] DAVIS, JAMES, RAMAMOORTHI, RAVI, and RUSINKIEWICZ, SZYMON. "Spacetime stereo: A unifying framework for depth from triangulation". *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* Vol. 2. IEEE. 2003, II–359 4.

[DYW05] DAVIS, JAMES E, YANG, RUIGANG, and WANG, LIANG. "BRDF invariant stereo using light transport constancy". *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. Vol. 1. IEEE. 2005, 436–443 2.

[EBGB14] ECHEVARRIA, JOSE I, BRADLEY, DEREK, GUTIERREZ, DIEGO, and BEELER, THABO. "Capturing and stylizing hair for 3D fabrication". *ACM Transactions on Graphics (ToG)* 33.4 (2014), 125 2.

[FGT*16] FYFFE, GRAHAM, GRAHAM, PAUL, TUNWATTANAPONG, BOROM, et al. "Near-Instant Capture of High-Resolution Facial Geometry and Reflectance". *Computer Graphics Forum*. Vol. 35. 2. Wiley Online Library. 2016, 353–363 2.

[FJA*14] FYFFE, GRAHAM, JONES, ANDREW, ALEXANDER, OLEG, et al. "Driving high-resolution facial scans with video performance capture". *ACM Transactions on Graphics (TOG)* 34.1 (2014), 1–14 2.

[FŠP*07] FISCHER, SYLVAIN, ŠROUBEK, FILIP, PERRINET, LAURENT, et al. "Self-invertible 2D log-Gabor wavelets". *International Journal of Computer Vision* 75.2 (2007), 231–246 3, 11.

[GFT*11] GHOSH, ABHIJEET, FYFFE, GRAHAM, TUNWATTANAPONG, BOROM, et al. "Multiview face capture using polarized spherical gradient illumination". *Proceedings of the 2011 SIGGRAPH Asia Conference*. 2011, 1–10 2.

[GLD*19] GUO, KAIWEN, LINCOLN, PETER, DAVIDSON, PHILIP, et al. "The relightables: Volumetric performance capture of humans with realistic relighting". *ACM Transactions on Graphics (TOG)* 38.6 (2019), 1–19 2.

[HBLB17] HU, LIWEN, BRADLEY, DEREK, LI, HAO, and BEELER, THABO. "Simulation-ready hair capture". *Computer Graphics Forum*. Vol. 36. 2. Wiley Online Library. 2017, 281–294 2.

[HML*14] HU, LIWEN, MA, CHONGYANG, LUO, LINJIE, et al. "Capturing braided hairstyles". *ACM Transactions on Graphics (TOG)* 33.6 (2014), 225 2.

[HMLL14] HU, LIWEN, MA, CHONGYANG, LUO, LINJIE, and LI, HAO. "Robust hair capture using simulated examples". *ACM Transactions on Graphics (TOG)* 33.4 (2014), 126 2.

[HMLL15] HU, LIWEN, MA, CHONGYANG, LUO, LINJIE, and LI, HAO. "Single-view hair modeling using a hairstyle database". *ACM Transactions on Graphics (TOG)* 34.4 (2015), 125 2.

[HVC08] HERNANDEZ, CARLOS, VOGIATZIS, GEORGE, and CIPOLLA, ROBERTO. "Multiview photometric stereo". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.3 (2008), 548–554 2.

[HZ03] HARTLEY, RICHARD and ZISSERMAN, ANDREW. *Multiple view geometry in computer vision*. Cambridge university press, 2003 2.

[HZW12] HERRERA, TOMAS LAY, ZINKE, ARNO, and WEBER, ANDREAS. "Lighting hair from the inside: A thermal approach to hair reconstruction". *ACM Transactions on Graphics (TOG)* 31.6 (2012), 1–9 2.

[JMM09] JAKOB, WENZEL, MOON, JONATHAN T, and MARSCHNER, STEVE. "Capturing hair assemblies fiber by fiber". *ACM Transactions on Graphics (TOG)* 28.5 (2009), 164 2.

[KM17] KHUNGURN, PRAMOOK and MARSCHNER, STEVE. "Azimuthal scattering from elliptical hair fibers". *ACM Transactions on Graphics (TOG)* 36.2 (2017), 1–23 2.

[KN08] KOPPAL, SANJEEV J and NARASIMHAN, SRINIVASA G. "Appearance derivatives for isonormal clustering of scenes". *IEEE transactions on pattern analysis and machine intelligence* 31.8 (2008), 1375–1385 2.

[KSZ*16] KHUNGURN, PRAMOOK, SCHROEDER, DANIEL, ZHAO, SHUANG, et al. "Matching Real Fabrics with Micro-Appearance Models". *ACM Trans. Graph.* 35.1 (Dec. 2016). ISSN: 0730-0301. DOI: 10.1145/2818648. URL: https://doi.org/10.1145/2818648 2.

[LLP*12] LUO, LINJIE, LI, HAO, PARIS, SYLVAIN, et al. "Multi-view hair capture using orientation fields". *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, 1490–1497 2.

[LLR13] LUO, LINJIE, LI, HAO, and RUSINKIEWICZ, SZYMON. "Structure-aware hair capture". *ACM Transactions on Graphics (TOG)* 32.4 (2013), 76 2.

[LLWL20] LEE, CHENG-HAN, LIU, ZIWEI, WU, LINGYUN, and LUO, PING. "Maskgan: Towards diverse and interactive facial image manipulation". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, 5549–5558 3.

[LMC19] LOGOTHETIS, FOTIOS, MECCA, ROBERTO, and CIPOLLA, ROBERTO. "A differential volumetric approach to multi-view photometric stereo". *Proceedings of the IEEE International Conference on Computer Vision*. 2019, 1052–1061 2.

[MJC*03] MARSCHNER, STEPHEN R., JENSEN, HENRIK WANN, CAMMARANO, MIKE, et al. "Light Scattering from Human Hair Fibers". *ACM Trans. Graph.* 22.3 (July 2003), 780–791. ISSN: 0730-0301. DOI: 10.1145/882262.882345. URL: http://doi.acm.org/10.1145/882262.882345 2, 6.

[NWKS19] NAM, GILJOO, WU, CHENGLEI, KIM, MIN H, and SHEIKH, YASER. "Strand-accurate multi-view hair capture". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, 155–164 1–5, 7, 8.

[PBS04] PARIS, SYLVAIN, BRICEÑO, HECTOR M, and SILLION, FRANÇOIS X. "Capture of hair geometry from multiple images". *ACM transactions on graphics (TOG)* 23.3 (2004), 712–719 2, 3.

[PCK*08] PARIS, SYLVAIN, CHANG, WILL, KOZHUSHNYAN, OLEG I, et al. "Hair photobooth: geometric and photometric acquisition of real hairstyles". *ACM Transactions on Graphics (TOG)*. Vol. 27. 3. ACM. 2008, 30 2.

[PHVL15] PEKELIS, LEONID, HERY, CHRISTOPHE, VILLEMIN, RYUSUKE, and LING, JUNYI. *A Data-Driven Light Scattering Model for Hair*. https://graphics.pixar.com/library/DataDrivenHairScattering/. Feb. 2015 2.

[PSM*16] PARK, JAESIK, SINHA, SUDIPTA N, MATSUSHITA, YASUYUKI, et al. "Robust multiview photometric stereo using planar mesh parameterization". *IEEE transactions on pattern analysis and machine intelligence* 39.8 (2016), 1591–1604 2.

[SDR*20] SCHMITT, CAROLIN, DONNE, SIMON, RIEGLER, GERNOT, et al. "On Joint Estimation of Pose, Geometry and svBRDF From a Handheld Scanner". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, 3493–3503 2.

[TFG*13] TUNWATTANAPONG, BOROM, FYFFE, GRAHAM, GRAHAM, PAUL, et al. "Acquiring reflectance and shape from continuous spherical harmonic illumination". *ACM Transactions on graphics (TOG)* 32.4 (2013), 1–12 2.

[VPB*09] VLASIC, DANIEL, PEERS, PIETER, BARAN, ILYA, et al. "Dynamic shape capture using multi-view photometric stereo". *ACM SIGGRAPH Asia 2009 papers*. 2009, 1–11 2.

[Woo80] WOODHAM, ROBERT J. "Photometric method for determining surface orientation from multiple images". *Optical engineering* 19.1 (1980), 191139 2.

[WOQS05] WEI, YICHEN, OFEK, EYAL, QUAN, LONG, and SHUM, HEUNG-YEUNG. "Modeling hair from multiple views". *ACM Transactions on Graphics (ToG)*. Vol. 24. 3. ACM. 2005, 816–820 2.

[XWW*14] XU, ZEXIANG, WU, HSIANG-TAO, WANG, LVDI, et al. "Dynamic hair capture using spacetime optimization". *To appear in ACM TOG* 33 (2014), 6 2.

[YWP*18] YU, CHANGQIAN, WANG, JINGBO, PENG, CHAO, et al. "Bisenet: Bilateral segmentation network for real-time semantic segmentation". *Proceedings of the European conference on computer vision (ECCV)*. 2018, 325–341 3.

[YY11] YOSHIYASU, YUSUKE and YAMAZAKI, NOBUTOSHI. "Topology-adaptive multi-view photometric stereo". *CVPR 2011*. IEEE. 2011, 1001–1008 2.

[YY97] YOU, H and YU, L. "Atomic force microscopy as a tool for study of human hair". *Scanning* 19.6 (1997), 431–437 6.

[ZCW*17] ZHANG, MENG, CHAI, MENGLEI, WU, HONGZHI, et al. "A data-driven approach to four-view image-based hair modeling". *ACM Transactions on Graphics* 36.4 (July 2017), 156:1–156:11. ISSN: 0730-0301 (print), 1557-7368 (electronic). DOI: https://doi.org/10.1145/3072959.3073627 2.

[ZFT*20] ZHANG, XIUMING, FANELLO, SEAN, TSAI, YUN-TA, et al. "Neural Light Transport for Relighting and View Synthesis". *arXiv preprint arXiv:2008.03806* (2020) 2.

[ZHX*18] ZHOU, YI, HU, LIWEN, XING, JUN, et al. "HairNet: Single-View Hair Reconstruction using Convolutional Neural Networks". *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, 235–251 2.

[ZRL*09] ZINKE, ARNO, RUMP, MARTIN, LAY, TOMÁS, et al. "A practical approach for photometric acquisition of hair color". *ACM Transactions on Graphics (TOG)* 28.5 (2009), 1–9 2.

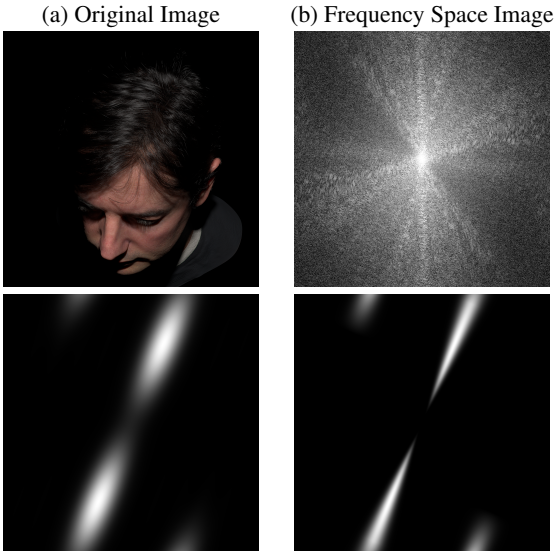

Figure 11: *As visualized in frequency space, the log-gabor filter contains sharper and more precise frequency support, which can better capture the 2D orientation of the hair strands.*

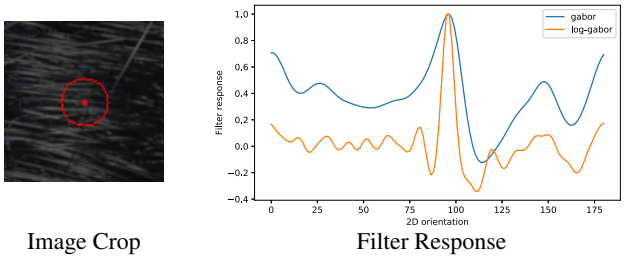

Figure 12: *Filter responses of gabor and log-gabor filters applied on the same pixel. Log gabor filters produce sharper peaks in the filter response, thus can detect the hair strand direction better.*

[ZW07] ZINKE, ARNO and WEBER, ANDREAS. "Light scattering from filaments". *IEEE Transactions on Visualization and Computer Graphics* 13.2 (2007), 342–356 6.

[ZWT13] ZHOU, ZHENGLONG, WU, ZHE, and TAN, PING. "Multi-view photometric stereo with spatially varying isotropic materials". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, 1482–1489 2.

[ZZ19] ZHANG, MENG and ZHENG, YOUYI. "Hair-GAN: Recovering 3D hair structure from a single image using generative adversarial networks". *Visual Informatics* 3.2 (2019), 102–112 2.

Appendix A: Log-Gabor Filter

We use log-gabor filter [FŠP*07] to extract the 2D orientation map from the images. The frequency representation of a log-gabor filter which captures the 2D line orientation θ is expressed as:

$$\mathcal{F}_\theta(\rho,\phi) = \exp\left(-\frac{\log(\rho\cdot\lambda)}{2\sigma_\lambda^2}\right)\cdot\exp\left(-\frac{(\phi-\theta)^2}{2\sigma_\phi^2}\right), \qquad (13)$$

where ρ and ϕ are the polar coordinates in frequency space. We use $\lambda = 3$ because the average width of the hair strands we observe is roughly 3 pixels. We set $\sigma_\lambda = log(2)$ to tolerate the variation in the hair width, and $\sigma_\phi = 2*180^\circ/128$ since we have divided 180° into 128 bins. In other words, we have 128 log-gabor filters in our filter bank.

Figure 11(b) shows the frequency representation of a photometric image which contains many hair strands. Since there are many hair strands that are following the same 2D line orientation in the image, we can observe many line-like structures emitting from the image center. Compared to the frequency support of gabor filter in Fig 11(c), the frequency support of log-gabor filter is more concentrated to a specific direction, and its shape has more resemblance to the pattern observed in the frequency space image.

Figure 12 shows an example of applying the gabor and the log-gabor filters on a specific pixel. As shown in the plots, the filter response from the log-gabor filters contains much sharper peaks than the response from the gabor filter.

Appendix B: 2D orientation results

In Fig. 13, we show our procedure of extracting 2D orientation maps and confidence maps from photometric images. We first apply log-gabor filter on each photometric image to get the per-image 2D orientation map and the confidence map. Notice that these per-image 2D maps are sometimes inaccurate, since there are many dark regions in the photometric images. We then collect the 2D maps from the images taken from the same camera, and merge them into a single 2D orientation map and a confidence map. These two maps are later used for 3D line reconstruction.

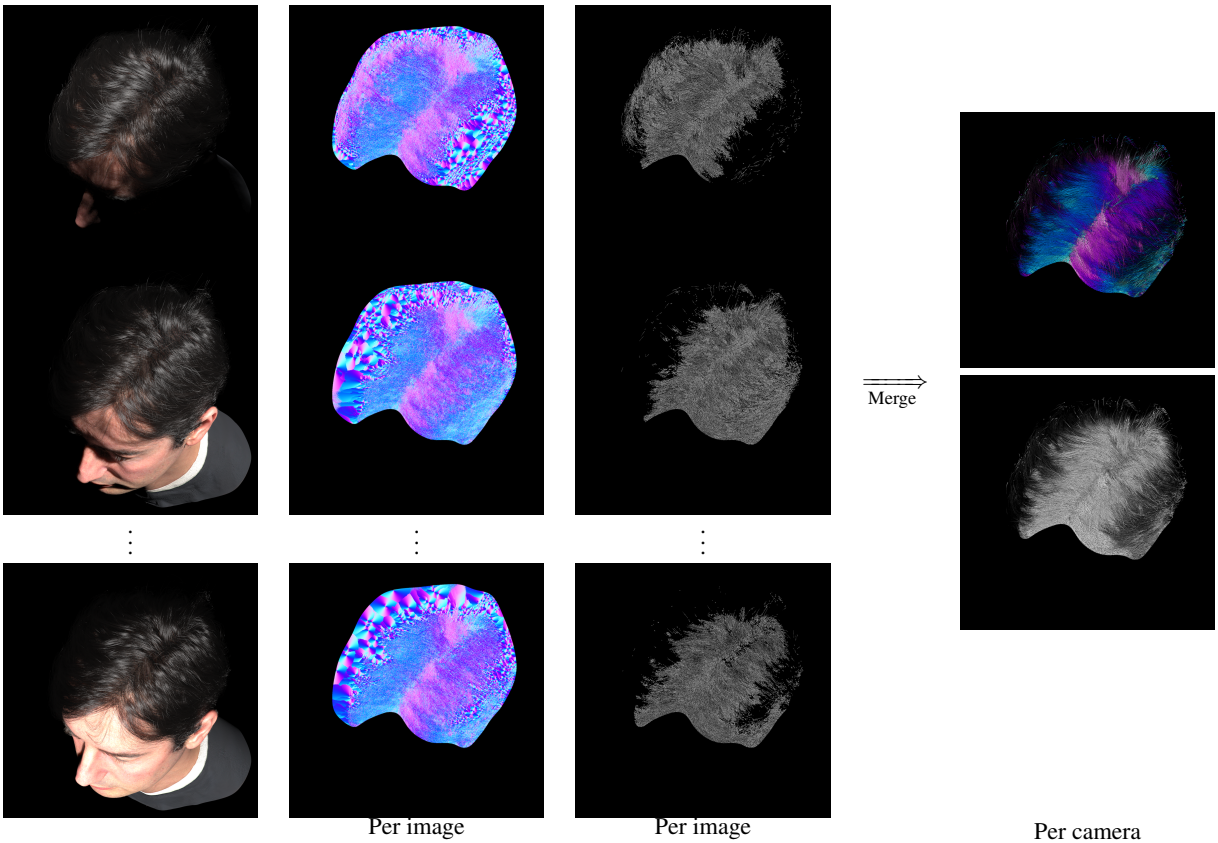

Figure 13: *We first extract the 2D orientation map and the confidence map from each photometric image. Then we merge the 2D maps from the same camera into a single 2D orientation map and a confidence map.*