January 2024

HIGH-FIDELITY 3D HAIR MODELING USING COMPUTED TOMOGRAPHY

High-Fidelity 3D Hair Modeling using Computed Tomography

Fig. 1. Our framework produces high-fidelity 3D hair models using computed tomography and hair wigs. Ten samples of 3D hair models are produced by our framework, demonstrating the robustness of our method to various real-world hair styles. The 3D models are ready to use in various computer graphics applications such as physically-based simulation and rendering.

We introduce CT2Hair, a fully automatic framework for creating high-fidelity 3D hair models that are suitable for use in downstream graphics applications. Our approach utilizes real-world hair wigs as input, and is able to reconstruct hair strands for a wide range of hair styles. Our method leverages computed tomography (CT) to create density volumes of the hair regions, allowing us to see through the hair unlike image-based approaches which are limited to reconstructing the visible surface. To address the noise and limited resolution of the input density volumes, we employ a coarse-to-fine approach. This process first recovers guide strands with estimated 3D orientation fields, and then populates dense strands through a novel neural interpolation of the guide strands. The generated strands are then refined to conform to the input density volumes. We demonstrate the robustness of our approach by presenting results on a wide variety of hair styles and conducting thorough evaluations on both real-world and synthetic datasets.

CCS Concepts: • **Computing methodologies** → **Parametric curve and surface models**; **Volumetric models**.

Additional Key Words and Phrases: 3D modeling, hair modeling, computed tomography

1 INTRODUCTION

High-quality hair models are a crucial aspect of creating realistic digital humans. With the increasing popularity of games and social media, the demand for realistic and diverse hair styles in digital avatars has grown significantly. The diversity of hair styles is driven by variance in length, curliness, and topology. In computer graphics, 3D hair authoring tools have been continuously improved to support a wider range of hair styles [Fu et al. 2007; Kim and Neumann 2002; Xing et al. 2019; Yuksel et al. 2009]. However, the manual creation by artists is both time-consuming and difficult to scale, and can also be biased by the limitations of current 3D authoring tools. As a result, it remains a challenge to create a large dataset that accurately represents a wide range of real-world hair variations.

While image-based hair reconstruction methods have shown the promise of efficiently creating 3D hair models from real-world observations, these approaches suffer from occlusion, because a large portion of the hair interior is occluded from any viewpoint. Thus, existing approaches have to rely on either a simple heuristic such as diffusing flow fields from observed regions , or explicitly leveraging structural

priors to fill in missing regions such that each strand is attached to the scalp geometry. While these ad-hoc solutions work reasonably well for simple, relatively straight hair styles, they generally fail to model hair styles with high curvature. Furthermore, the accuracy of the inferred internal structure of hair, which is particularly important for hair animation, has never been validated with captured data.

To address this fundamental limitation of image-based methods, we explore 3D acquisition techniques that do not suffer from occlusion. In particular, we employ computed tomography (CT) using X-ray for its high resolution and large scan volumes. Because X-rays can pass through objects, we can obtain density volumes of the entire region of interest. While CT has been widely used for 3D reconstruction of human tissues or general objects [Rückert et al. 2022], recovering complete human hair strands from CT is a non-trivial task due to the thin structure of the strands and the limited resolution and noise inherent in CT imaging. In this paper, our goal is to enable accurate reconstruction of 3D hair strands from diverse real-world hair wigs[1] using industrial CT scanners[2].

To this end, we present *CT2Hair*, a fully automatic framework for creating high quality 3D hair models using computed tomography. Our method takes as input the 3D density volume of a hair wig, and generates a set of 3D curves representing individual hair strands. To handle the noise and limited resolution of density volumes, we employ a coarse-to-fine approach. We first estimate a 3D orientation field from a noisy density volume, and extract reliable sparse guide strands using the estimated 3D orientation field. Then, we populate the scalp with strands using a neural interpolation method inspired by Neural Strands . Finally, the generated strands are further refined with optimization such that they accurately conform to the input density volume. Importantly, because our framework does not have hand crafted priors for particular hair types, we can recover extremely diverse hair styles in a unified framework. Furthermore, our strand representation is ready to use in many downstream tasks in computer graphics, such as physically-based simulation and rendering.

In our experiments, we evaluate our design choices, and demonstrate the robustness and effectiveness of our approach over image-based solutions. We also show our reconstruction results from acquired density volumes of various hair styles, showcasing the robustness of our approach. Upon publication, we will publicly release both the raw density data and the reconstructed strands for at least ten hair styles.

Our contributions can be summarized as:

- The first work to use computed tomography to robustly reconstruct complete hair strands, without hallucinating the hair volume interior.
- A coarse-to-fine 3D curve reconstruction framework that handles noise and blurriness in 3D density volumes.
- Code and data to facilitate future research on 3D hair reconstruction from computed tomography.

[1]Many hair wigs are made of real human hair, and hairdressers can stylize wigs to create natural hair styles, making them virtually indistinguishable from real hair

[2]Industrial CT scanners are a better choice over medical scanners for our work. Medical scanners are optimized for X-ray dose efficiency and acquisition speed at the expense of losing geometric details. In contrast, industrial scanners are able to capture larger scanning volumes with higher resolutions at the cost of scanning time. Note that industrial CT scanners are unsuitable for human head for safety reasons.

2 RELATED WORK

In this section, we briefly review prior work in hair geometry modeling and editing including image-based techniques. We also describe prior uses of radiography in computer graphics.

2.1 Hair Geometry Modeling and Editing

The difficulty of 3D hair modeling mainly comes from the large number of hair strands (typically more than 100,000) and the wide variety of hair styles. Several user-friendly interactive tools have been developed for intuitive modeling and editing of 3D hair models. proposed a multiresolution modeling system. presented a sketching interface that takes user's strokes as input. introduced Hair Meshes, which enables intuitive strand-level 3D hair modeling by converting low-res polygonal meshes into hair strands. 3D haptic input interfaces or virtual reality display devices have also shown to be useful for interactive hair modeling. presented a deep learning based interactive tool for 3D hair modeling from 2D sketches.

Similarly, various representations have been proposed for modeling 3D hair. presented level-of-detail representations that can efficiently model individual strands, clusters, and strips of hair. proposed a Super-Helix strand representation for realistic and stable simulation.
proposed a hair style synthesis and editing method that uses exemplary hair models and 2D feature maps. A parametric 3D curve model for hair strands was proposed by The curve model can be used for several applications for hair modeling, such as hair synthesis, interpolation, and data fitting.
proposed a 3D vector field based hair modeling method that interactively models the 3D flows of hair strands and their curliness in a volumetric manner. We refer to and for a more exhaustive list of work in the area. While these approaches facilitate the modeling of diverse hair styles, manual creation by hair modelers is difficult to scale. In contrast, our work enables fully automatic reconstruction of 3D hair models with a wide range of hair styles.

2.2 Image-based Hair Modeling

Image-based modeling is a promising alternative to manual modeling because it can achieve higher quality with lower effort. While RGB images are the most popular imaging modality, depth sensors or thermal imaging are also utilized for image-based hair modeling. Regardless of the imaging modality, the common approach is to use 2D orientation maps
2D orientation maps are computed from captured images using 2D kernels such as a Gabor filter. They are used as a geometric constraint such that the reconstructed 3D hair strands are consistent with the 2D orientation maps when they are projected to the view

It

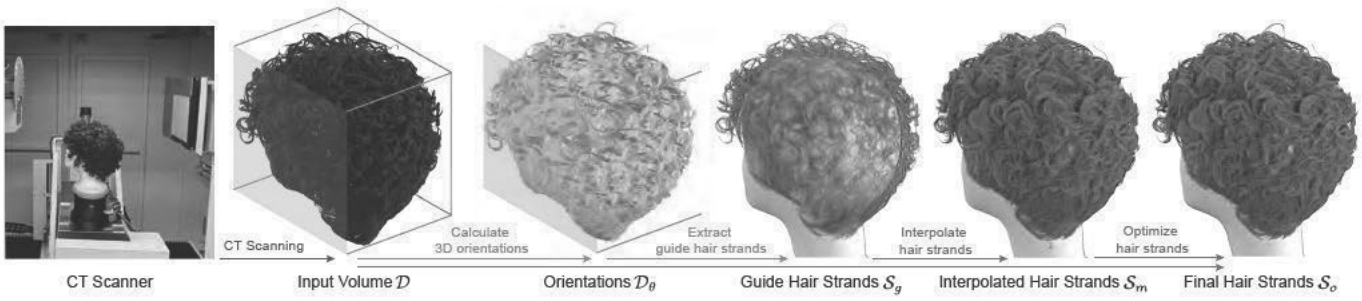

Fig. 2. Overview of our method. We use an industry CT scanner to obtain 3D density volumes. The first stage (Sec. 3.2) extracts reliable guide strands by estimating 3D hair orientation fields. The second stage (Sec. 3.3) refines the interpolated guide strands using the input density volume.

has also been shown that individual hair strands can be accurately reconstructed using multi-view stereo methods . However, 2D images, regardless of their modality, only see the outer surface of hair, and therefore do not have enough information to model complete hair strands, which is typically a prerequisite for physically-based simulation of hair. Inferring the occluded parts of hair have been handled using either simple heuristics or a database of 3D hair models . However, these solutions may fail at complex hair styles such as highly curly hair. As computed tomography allows us to see through the entire hair region, our approach does not suffer from occlusion.

Recently, data-driven methods using deep learning techniques have shown their effectiveness in both single view and multi-view setups. However, the quality of reconstructed hair models are often bounded by the synthetic 3D hair models they used to train the networks. Our work offers a solution for collecting diverse 3D hair models from real-world hair wigs to empower these learning-based methods.

2.3 Radiography for Computer Graphics

Radiography is an imaging technique that uses radiation, such as X-rays, to inspect the internal structure of an object. In the computer graphics literature, industrial CT scanning has been used as a non-destructive method to acquire the complete 3D volume of an object. proposed space-time tomography to model 4D volumes for deforming objects such as rising dough. Efficient and accurate 3D volume reconstruction is possible by using neural scene representations or super-resolution techniques . Micro CT scanning has been also used to capture the microscale geometric details of fabrics, which can be used for photorealistic material appearance modeling and rendering thousands of X-ray images of human body to train a neural network that infers the underlying skeleton from a 3D body shape. Magnetic resonance imaging (MRI) was used to collect a large dataset of human hands and to build parametric hand models with their bones and muscles To the best of our knowledge, our work is the first to utilize computed tomography for 3D hair modeling.

3 OUR METHOD

An overview of our pipeline is shown in Fig. 2. Scanning a hair wig with a CT scanner gives us a 3D volume, of which each voxel value represents the radiodensity at its 3D position. Given the 3D density volume as the input, our method reconstructs a set of 3D curves that conforms to the input. Our method takes a coarse-to-fine approach and consists of two stages: guide strands initialization (Sec. 3.2) and dense strands optimization (Sec. 3.3).

The first stage starts with computing a 3D orientation field from the input 3D density volume. We then treat each voxel as a 3D point with its position and orientation, and generate a point cloud of the hair wig. We then estimate hair segments, i.e., partially connected hair strands, by connecting neighboring points along their orientation. By connecting and extending hair segments, we obtain guide hair strands that are connected to the hair scalp. In the second stage, we interpolate the estimated guide strands so that they distribute uniformly on the scalp. Next, we optimize the interpolated hair strands using the source density volume as the target. The optimized hair strands are the final 3D hair model that are ready for down-stream applications.

3.1 Input Data

The input to our framework is a 3D density volume, $\mathcal{D} \in \mathbb{R}^{W \times H \times D}$, of the target hair wig. The resolution of $\mathcal{D}$ depends on the hardware specification of the scanner and the size of the wig. The voxel grid resolutions in this paper range from 1800^3 to 5000^3, and the voxel sizes are $0.08mm^3$ to $0.15mm^3$. The thickness of human hair strands ranges between $0.04mm$ to $0.12mm$. While the 3D volumes in this paper have the highest resolutions one can get from off-the-shelf CT scanners, they do not go beyond the Nyquist rate to reconstruct sharp edges of individual hair strands. Furthermore, the scanner has its own modulation transfer function (MTF) that blurs the output 3D volume to some degree . As a result, the density of each strand is not clearly isolated. In addition, the physical radiodensity of human hair is relatively low and does not show enough contrast against air. For all these reasons, the input 3D volume $\mathcal{D}$ is inevitably noisy and blurry. The goal of the first stage of our pipeline is to robustly extract guide hair strands $\{\mathcal{S}_g\}$ from $\mathcal{D}$.

Preprocessing. Before reconstruction, we crop wig regions out from our density volume using a density threshold range, and register a template head model to the mannequin head. More details on the preprocessing can be found in our supplementary document.

3.2 3D Volume to Guide Strands

3.2.1 3D Orientation Estimation. A 3D orientation field conveys important information about the hair structures inside the volume, and thus has been widely used in hair modeling

A straightforward method to compute 3D orientation from 3D densities is kernel filtering, which has been demonstrated in CT scanned volume data

However, we find such a kernel filtering approach performs poorly with our noisy and blurred input. In addition, it is non-trivial to find an optimal set of parameters including kernel size and frequency, especially with varying resolutions. Therefore, we present a new 3D orientation estimation method based on vector calculus, significantly improving robustness and accuracy.

We first compute the gradient field of the input volume:

$$\nabla \mathcal{D} = \left(\frac{\partial \mathcal{D}}{\partial x}, \frac{\partial \mathcal{D}}{\partial y}, \frac{\partial \mathcal{D}}{\partial z} \right). \quad (1)$$

Because the hair orientation is orthogonal to its density gradient, each voxel orientation $o \in \mathbb{R}^3$ can be estimated based on gradients in adjacent voxels as follows:

$$\min_{o_i} ||\mathrm{G}_i o_i||_2 \quad s.t. ||o_i||_2 = 1, \quad (2)$$

where $\mathrm{G}_i \in \mathbb{R}^{n_g \times 3}$ is a matrix consisting of n_g gradients and each row $\mathrm{G}_i(j) \in \mathbb{R}^3$ represents a neighbor gradient of the i-th voxel. We perform SVD on $\mathrm{G} = U\Sigma V^T$. Solving Eq. 2 is equivalent to finding the eigenvector with the smallest eigenvalue in Σ. Following [

we also take an iteratively reweighted least squares approach to reduce any bias from outliers.

Noise Removal. Once we have per-voxel orientation o, we use $||\mathrm{G}o||_2$ as a scalar confidence value for every voxel, and exclude voxels with low confidence to further remove outliers. We define the calculated 3D orientation volume as $\mathcal{D}_\theta \in \mathbb{R}^{3 \times W \times H \times D}$ where the invalid voxels only have empty values. Note that we do not resolve directional ambiguity in this step. Namely, connected voxels may have opposite directions. We resolve this ambiguity in the following steps.

3.2.2 Guide Strands Generation. This subsection describes how to obtain a set of guide strands $\{\mathcal{S}_g\}$ from 3D orientation volume $\mathcal{D}_\theta$. Previous work extracted hair strands from 3D orientation fields by directly tracing 3D curves from the head scalp [Saito et al. 2018; Yang et al. 2019; Zhang and Zheng 2019] which is a noise-sensitive solution unsuitable for our noisy input volume with high-frequency details. To effectively extract hair strands from our high-resolution 3D orientation volume, we first generate short hair segments and then connect them to get fully grown hair strands. To this end, we convert our 3D orientation volume $\mathcal{D}_\theta$ into a dense point cloud with orientations $\mathcal{P}_\theta \in \mathbb{R}^{n_p \times 6}$ where n_p is the number of valid voxels. Each point p in $\mathcal{P}_\theta$ stores its 3D position p^p and 3D orientation o^p.

Hair Segments from Point Cloud. Given a dense point cloud with orientations $\mathcal{P}_\theta$, we follow [Nam et al. 2019] and get a set of partitioned hair segments $\{s\}$. We first perform the mean-shift point cloud filtering described in [Nam et al. 2019] (Sec. 5.1), and get a clean, filtered point cloud $\mathcal{P}'_\theta$ (See Fig. 3b). We then use the forward Euler method on $\mathcal{P}'_\theta$ for generating hair segments $\{s\}$. A hair segment s is a sequence of connected 3D points, i.e., $s = \{p_0^p, p_1^p, ...\}$.

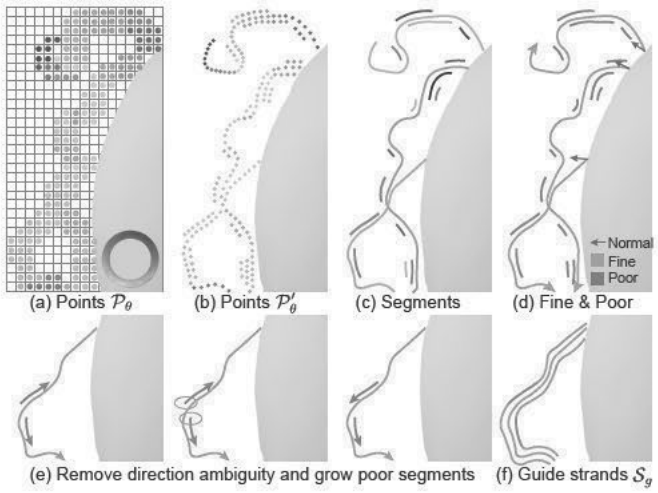

Fig. 3. The processing of guide hair strands generation. (a)-(c) presents hair segments generation. (b) shows segmented fine strands and poor segments. (e) presents that we compute the nearest fine strand, remove direction ambiguity by using the fine direction for poor segments, and grow poor segments using fine directions. (f) shows the guide hair strands after growing.

Guide Hair Growing. To obtain complete strands with a coherent direction, we connect hair segments to the hair scalp and also extend hair segments to the hair tip. As illustrated in Fig. 3 (c), the qualities are irregular due to the noise of the input volume. The majority of hair segments are incomplete and far away from the hair scalp because the region close to the hair scalp is noisier due to the interference of the underlying hairnet of the wig. Therefore, we introduce a guide hair growing method from hair segments. More specifically, we first cluster hair segments into complete fine strands $\{\mathcal{S}^f\}$ and isolated poor hair segments $\{s^p\}$. We provide detailed procedures in Algorithm 1. Given the hair scalp surface Ω from the registered head model, if a sufficiently long segment touches the scalp with coherent orientation, we label it as a fine hair strand $\mathcal{S}^f$. We can also remove the directional ambiguity of $\{\mathcal{S}^f\}$ by testing their connectivity to the scalp using the scalp normal. Once we get the fine hair strands set $\{\mathcal{S}^f\}$, we re-direct and grow the remaining poor segments $\{s^p\}$ alongside their nearest fine hair strands, as illustrated in Fig. 3 (e) and (f). More details about the poor segments growing method can be found in our supplementary material. The union of the grown poor segments set and the fine hair strands set $\{\mathcal{S}^f\}$ is now our guide hair strands set $\{\mathcal{S}_g\}$.

3.3 Hair Strands Reconstruction

While our reconstructed guide hair strands $\{\mathcal{S}_g\}$ preserve the original hair style, they have two major problems; the number of guide strands is much less than the original hair, and the root positions are not uniformly distributed on the scalp (See Fig. 2). In this section, we describe how we naturally densify the guide hair strands using learned 3D curve prior. In addition, we propose an optimization stage to refine our dense strands to better align them with the input volume.

ALGORITHM 1: Seperate hair segments into fine and poor sets

Input: Hair segments set $\{s\}$, hair scalp surface Ω
Output: Fine hair strands set $\{\mathcal{S}^f\}$, poor hair segments set $\{s^p\}$
Params: $\tau_c = 8$mm, $\tau_l = 36$mm
Function ConnectedRootsSearch(s):
  $s^r = s[0]$; $s^o = \text{normalize}(s[1] - s[0])$;
  find the nearest point r to s^r on Ω,
  r^p is the position, and r^n is the normal;
  if $||r^p - s^r||_2 <= \tau_c$ & $r^n \cdot s^o >= cos(\tau_a)$ **then**
    s.push_forward(r_n^p); **return** true;
  end
  return false;
End Function
foreach $s \in \{s\}$ **do**
  if *length*(s) < τ_l **then**
    $\{s^p\}$.add(s); continue;
  end
  $s^{-1} = s$.reverse;
  if ConnectedRootsSearch(s) **then**
    $\{\mathcal{S}^f\}$.add(s); continue;
  else
    if ConnectedRootsSearch(s^{-1}) **then**
      $\{\mathcal{S}^f\}$.add(s^{-1}); continue;
    end
  end
  $\{s^p\}$.add(s);
end

3.3.1 Hair Interpolation. The goal of the interpolation step is to get dense hair strands $\{\mathcal{S}_m\}$ from the guide hair strands $\{\mathcal{S}_g\}$. To ensure the stability of 3D curve interpolation process, we utilize a learned 3D curve shape prior.

Similar to , we pre-train a variational autoencoder (VAE) consisting of an encoder **Enc** and decoder **Dec** to learn a latent space of the natural shapes of hair strands. More details of the training can be found in the supplementary document. We first transform $\{\mathcal{S}_g\}$ into TBN space $T(\{\mathcal{S}_g\})$ and then apply an encoder **Enc** on them to get the latent representation $Z_g = \text{Enc}(T(\{\mathcal{S}_g\}))$, $Z_g \in \mathbb{R}^{N_g \times C}$, where N_g is the number of guide hair strands and C is the size of latent code. Given a tiled texture map of the scalp shown in Fig. 4, we convert Z_g into a 2D feature map and then perform interpolation. Specifically, we uniformly sample N_i texels in the scalp texture region, and for each sampled texel, we blend k_i nearest neighbors with existing latent representations into $z_n \in \mathbb{R}^{k_i \times C}$. Then for each sampled texel $i, i \in [0, N_i - 1]$, the interpolated strand latent representation $z_i \in \mathbb{R}^C$ is calculated following the weighted sum of z_n:

$$z_i = z_n^T w, \quad w = \text{normalize}\left(\frac{P_d(z_n, z_i)}{||P_d(z_n, z_i)||_2} \cdot \frac{z_n z_0^T}{||z_n z_0^T||_2}\right), \tag{3}$$

where $P_d(a, b)$ is the pixel distance between a and b, and z_0 is the nearest latent representation to z_i in UV space. In our implementation, we discard sampled pixels when $||P_d(z_n, z_i)||$ is out of the range of $[\sigma_{min}, \sigma_{max}]$. Therefore, we do not interpolate hair strands if the region has high density of existing strands. σ_{max} is a hyper parameter to avoid interpolating hair strands in empty regions, e.g., outside the scalp boundary.

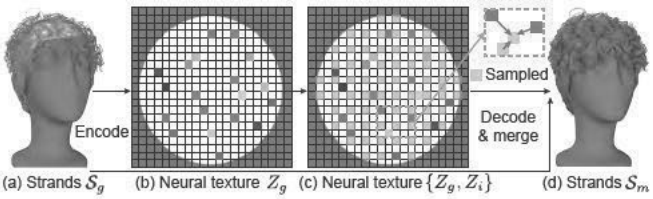

Fig. 4. The processing of our neural strands interpolation. We use the network to encode strands (a) into a neural texture map (b). And then, we sample pixels in (c) for interpolation. (d) shows the merged strands.

After interpolation in 2D space, we use a decoder **Dec** to recover the strand geometry, and convert them into the original coordinates. Finally we merge them with the input guide strands as a set of strands as: $\{\mathcal{S}_m\} = \{\mathcal{S}_g\} \cup T^{-1}(\text{Dec}(Z_i))$. In our experiments, we set $k_i = 3, \sigma_{min} = 0.6, \sigma_{max} = 6$, and N_i =100K so that the number of merged hair strands matches an average number of real human hair strands.

3.3.2 Hair Optimization. Our optimization step aims to align our reconstructed hair strands with the input volume better. While an intuitive solution is to minimize the error between strands and the volume, this is non-trivial as there is no differentiable error metrics between these two heterogeneous data types, strands and volume. To address this, we propose to convert these two representations into point clouds. Unlike the naive conversion used in Sec. 3.2.2, we incorporate density information. Because the original density is not fully reliable due to noise, we discretize the wig density range and separate wig voxels into n_d bins using a constant interval and then sample i points from i-th bin, which means for higher density voxels, the sampled target points are denser. As a result, we obtain the point cloud $\mathcal{P}_d \in \mathbb{R}^{n'_p \times 3}$, where $n'_p > n_p$, from the input density volumes.

To minimize the distance between $\mathcal{P}_d$ and ordered points $\mathcal{P}_s$ from the interpolated strands, we employ the Chamfer Distance for differentiability. To avoid strand geometry stretches caused by the movement of the points, we first increase the number of points representing each strand using differentiable curve cubic interpolation $\mathcal{P}_c = \text{Cu}(\mathcal{P}_s)$. Note that we still optimize with respect to $\mathcal{P}_s$ instead of $\mathcal{P}_c$ to reduce degrees of freedom.

Similar to for each strand after cubic interpolation S, we also add the following regularization terms:

$$\mathcal{L}_{len} = \sum_i^{n-1} (||dir(S_i^t)||_2 - ||dir(S_i^0)||_2)^2, \tag{4}$$

$$\begin{aligned}\mathcal{L}_{tan} = &\sum_i^{n-1} ((dir(S_i^t) - dir(S_i^{t-1})) \cdot tan(S_i^{t-1}))^2 \\ &+ \sum_i^{n-1} ((dir(S_i^{t-1}) - dir(S_i^t)) \cdot tan(S_i^t))^2,\end{aligned} \tag{5}$$

where t means the iteration of the optimization process, $dir(S_i) = S_{i+1} - S_i$ is the first order approximation of the strand direction at point S_i and $tan(S_i) = normalize(dir(S_i))$ is the approximation of the tangent.

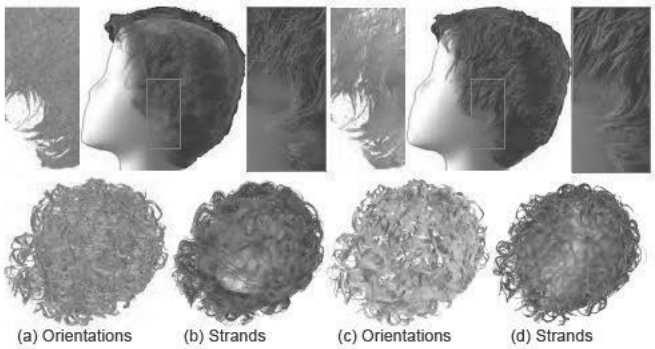

Fig. 5. Visual comparisons of orientations and guide hair strands using Gabor filters (a-b) and our method (c-d).

We denote the Chamfer Distance as $\mathcal{L}_{cham} = Cham(\mathcal{P}_d, \mathcal{P}_s)$, and the total loss of our optimization step is

$$\mathcal{L} = \omega_c \mathcal{L}_{cham} + \omega_l \sum^{S_m} \mathcal{L}_{len} + \omega_t \sum^{S_m} \mathcal{L}_{tan} \quad (6)$$

We minimize the loss $\mathcal{L}$ using gradient descent. Please refer to the supplementary material for more details about our gradient descent optimization.

In our implementation, due to the limited memory size, we optimize strands locally by selecting and optimizing local strands in each batch. We set $\omega_c = 1, \omega_l = \omega_t = 100$ in our experiments. After optimization, we convert the interpolated point cloud $\mathcal{P}'_c$ back to the strand format and consider them as our final output $\{\mathcal{S}_o\}$.

4 EXPERIMENTS

To demonstrate our method, we conduct experiments on real data collected from ten different wigs, i.e., 3D density volumes acquired by industrial CT scanners. The collected ten hair styles have diverse shapes, including curly, straight, short, and long. Also, we evaluate our method on several synthetic volumes for quantitative analysis. In addition, we show the superiority of our method over state-of-the-art image-based hair modeling methods. All of our experiments are conducted on a PC with 64GB memory and an NVIDIA GeForce RTX 3090 GPU.

4.1 Ablation Study

4.1.1 3D Orientation Estimation. As mentioned in Sec. 3.2.1, previous methods use filters to extract 2D/3D orientations from the input signal, but such filtering approaches do not perform well on our real-scanned noisy volume data. Here we show our improvements by using gradient-based calculation on two input volumes of two hair styles (short and curly). In Fig. 5, we show both the differences between orientations and extracted guide hair strands. Our gradient-based method significantly improves the quality of 3D orientations and yields more continuous and less noisy orientations, as shown in Fig. 5c. As a consequence, the extracted hair strands in Fig. 5d match the original volume much better, have more detail with less noisy strands. Note that, we carefully tune the parameters of the Gabor filter for a fair comparison, e.g., the size and frequency of the kernel.

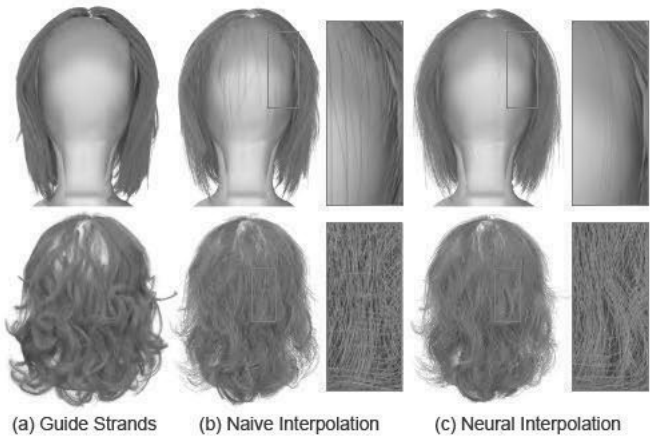

Fig. 6. Visual comparison of interpolation methods. We increase the number of strands from our estimated guide strands (a) by the weighted blending of neighboring strands in the Euclidean space (b) and our latent-space interpolation (c). Notice how our neural interpolation reduces implausible strands compared to the naive interpolation method.

4.1.2 Hair interpolation. Fig. 2 and Fig. 4 show the visualization of extracted guide hair strands from density volumes, and show that the guide strands are incomplete and non-uniformly distributed. There are two reasons for this problem. One is that the noise of the original volume makes us only extract limited guide hair strands. In addition, our pre-processing stage to remove the underlying hairnet may also remove some hair voxels near the scalp, and we cannot compute accurate and complete 3D orientations around hair root positions.

A straightforward solution to interpolate 3D curves is to interpolate all the points on the curves in the Euclidean space. However, because our input strands are non-uniform and have complex shape, such a naive interpolation produces severe artifacts, as shown in Fig. 6 (b). There are lots of interpolated strand directions that deviate from the original shape and penetrate the head. Naive interpolation creates too many parallel strands that do not match the input volume and also look unnatural. Benefiting from the learning-based method, our neural interpolation generates strands with higher similarity to the source.

4.1.3 Hair Optimization. Fig. 7 demonstrates the efficacy of our hair optimization stage (Sec. 3.3.2). We compare (a) the input density volumes against reconstructed strands (b) with and (c) without the optimization step. The result with optimization better models the original shape in the volume, and therefore is more realistic than the one before optimization because optimization allows the hair strands to have the same natural splits as the input volume, with more natural diversity. Therefore, our optimization helps the reconstructed strands form natural wisps with gathered parts. We also show quantitative comparisons of the proposed optimization step in the next subsection.

4.2 Evaluation with Synthetic Data

Because we do not have the ground truth hair strands of the real-world scanned volume, we evaluate our method on synthetic data

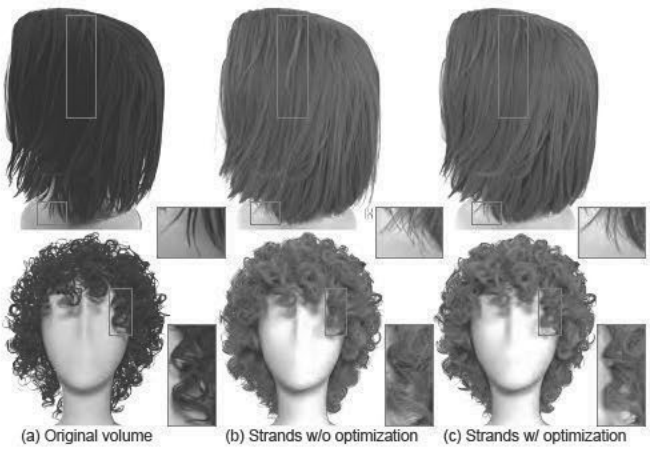

Fig. 7. Visual comparison of our optimization stage. The proposed optimization approach significantly reduces misalignment between the original volume (a) and our final strands (c) from our interpolation results (b).

to show that we can recover the original hair strands well and to demonstrate the improvements of our optimization quantitatively.

Given synthetic hair strands created by 3D artists, we first convert them to high-resolution density volumes by accumulating the number of strands in voxels (please refer to the supplementary document for more details). Then we apply our method for reconstructing strands to the synthesized volumes. We show the reconstructed results in Fig. 9. The results are consistent with the input, indicating the accuracy of our reconstruction.

As a further test, we also convert our reconstructed hair strands to volumes again and compare them with the input synthetic volumes. We compute IoU (Intersection over Union), precision, recall and MSE (Mean squared error) between the reconstructed volumes and the input volumes. As shown in Tab. 1, all of the above metrics show that our optimization step can significantly improve our reconstructed hair strands to match the input better. Note that the synthetic density volume is not an exact match to the real scanned volumes and is only suitable for evaluation purposes. Despite the domain gap between the synthesized volumes and the real ones, our method demonstrates its robustness.

4.3 Comparisons with Image-based Methods

We compare our methods with three state-of-the-art image-based hair reconstruction algorithms including single-view-based | sparse-view-based and dense-view-based method. We use eight images for DeepMVSHair [ ] and 134 images for Line-based Multi-view Stereo (LMVS) .

Fig. 8 shows the comparison. While learning-based methods (NeuralHDHair nd DeepMVSHair) can produce reasonable results for relatively simple hair styles (bottom two rows), they fail to accurately reconstruct the curly hair shown in the first row. This failure is largely due to the lack of such curly hair data in their training dataset. On the other hand, LMVS can accurately reconstruct 3D hair segments that are visible from multi-view images. However, LMVS does not infer the interior geometry and thus produces incomplete geometry.

Table 1. Quantitative Comparisons for our optimization stage of four cases. Short and Ponytail hairstyles are two cases shown in Fig. 9. Prec. means the precision. The numbers show differences between reconstructed volumes and the input volumes.

		IoU ↑	Prec. ↑	Recall ↑	MSE(10^{-2}) ↓
Short	w/o opt.	0.538	0.713	0.686	3.5944
	w/ opt.	**0.609**	**0.833**	**0.694**	**3.4747**
Ponytail	w/o opt.	0.599	0.722	0.779	3.8924
	w/ opt.	**0.666**	**0.799**	**0.800**	**3.8033**
Complex	w/o opt.	0.463	0.588	0.686	6.7203
	w/ opt.	**0.533**	**0.676**	**0.715**	**6.6963**
Curly	w/o opt.	0.392	0.502	0.641	1.9718
	w/ opt.	**0.458**	**0.596**	**0.664**	**1.9181**

Lastly, our reconstructed hair models show faithful geometries and have more detail which makes them look significantly more realistic.

4.4 Applications

Because our approach ensures the resulting strands are attached to the scalp, we can apply physically-based simulation *w*ithout any modifications to the reconstruction as shown in Fig. 10. Similarly, accurate reconstruction of hair interior enables more photorealistic rendering with physically-based rendering techniques The comparison with LMVS in Fig. 8 shows a clear difference in the final rendering quality.

5 CONCLUSION

We introduced CT2Hair, a fully automatic framework to robustly reconstruct strands with diverse hairstyles from 3D density volumes obtained from computed tomography. By seeing through the entire hair volume, our approach achieves, for the first time, the accurate recovery of the occluded hair interior, which has been a fundamental challenge for image-based approaches. In our experiments, we show that recovering detailed strands from noisy and blurry density volumes is now possible by carefully designing a coarse-to-fine approach consisting of guide strands generation and dense strands optimization. We also examined the effectiveness of the proposed learning-based interpolation scheme over a naive interpolation in the Euclidean space. Because the resulting hair models accurately retain connectivity to the scalp, we can effortlessly use them for physically-based simulation and rendering. Lastly, we believe releasing the data and code will further facilitate future research on high-quality 3D hair modeling using computed tomography.

Limitations and Future Work. We demonstrated our approach using real-world hair wigs with industry CT scanners. However, extending this to capturing real human heads remains challenging. To achieve sufficient resolution and capture volume, industry CT scanners use the large exposure of X-rays that exceeds the safety limit for living tissues. In addition, even subtle motions during capture lead to substantial blurriness in the density volume. Leveraging machine learning with a large corpus of high-quality 3D hair data obtained by our approach could enable plausible inference of 3D hair models even from low-resolution density volumes using medical CT scanners. Future work also includes co-registration of the reconstructed strands and multi-view images to learn photorealistic appearance of the hair models.

Fig. 8. Visual comparisons with image-based methods (NeuralHDHair , DeepMVSNet and LMVS). Note that geometry shape differences are caused by different deployments of wigs in different capture systems.

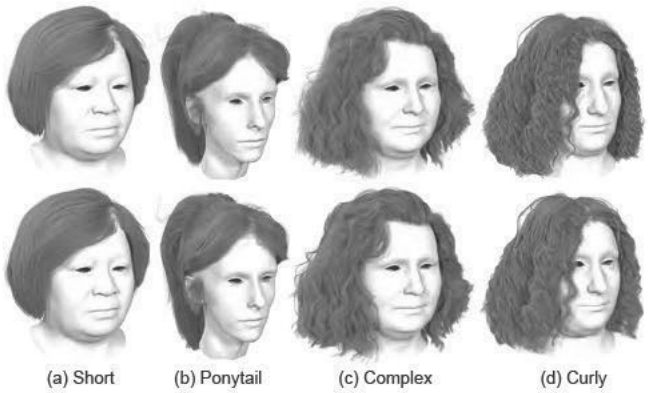

Fig. 9. Visualizations of synthetic evaluation. The 1st row is the input synthetic strands, and the 2nd row is the reconstructed strands. Our pipeline faithfully reconstructs the original hair styles.

Fig. 10. We run physically-based simulation with our reconstruction results. We show that our reconstructed hair strands can be easily integrated into the existing simulation pipeline and yield natural and realistic hair dynamics. Please refer to the supplemental video.

Fig. 11. Side-by-side comparison of input density volume (1st, 3rd, and 5th rows) and output strands (2nd, 4th, and 6th rows). Please refer to the supplemental video and document for more hair styles and their visualizations.

Supplementary Material for High-Fidelity 3D Hair Modeling using Computed Tomography

1 VOLUME PREPROCESSING

CT scanning records all the physical density in the 3D region, even for the air, but different physical materials have different density value ranges. Therefore, we can easily use several thresholds to crop wig regions out. Fig. 1 (a) shows the volume shape segmented by thresholding.

However, wigs usually have an underlying hairnet to pin strands onto the scalp, and it has a very similar density range to hair strands, so we cannot wholly split strands and the underlying hairnet using thresholds. Also, as we can see in the highlighted part in Fig. 1 (a), underlying hairnets have very complex topology, and it is difficult to calculate accurate orientations for them, i.e., without removing underlying hairnet, we will reconstruct lots of noisy strands. To remove underlying hairnet regions in the density volume, we first segment strand roots using a high-density threshold range because roots usually have higher density. As shown in Fig. 1 (b), the extracted points have a good shape of the head scalp. Note that the extracted points are not exact strand roots, many strand roots cannot be segmented using thresholds and there exist many underlying hairnet points in the extracted points. Next, we estimate point normals for extracted roots using the directions from their bounding sphere center and reconstruct the surface using Poisson surface reconstruction

Then, we can remove the underlying hairnet parts by deleting voxels close to the reconstructed surface.

We also use the extracted roots to register a template head model to fit the wig strands, and the registered head model will be used for our interpolation step (Sec. 4.3.1). To solve the symmetric ambiguity in the registration step, we first rotate the template head model to the known face direction, the same as the scanned volume. And then, we align the bounding spheres of the sampled scalp points from the template head model and the extracted roots. Next, we use rigid ICP for further alignment. Besides, we also deform the template scalp to the extracted roots for a better fitting. Fig. 1 (d) shows a fitted head model.

2 POOR SEGMENTS GROWING

Given separated poor segments and fine strands, we grow poor segments using the information from fine strands to preserve features extracted from the density volume as much as possible. Firstly, for each poor segment, we compute its nearest fine strands for both ends. Then, we use the growing direction of the nearest fine strands to correct the segment direction. To grow short, poor segments into strands, we refer to the tangents of their nearest fine strands for connecting to the scalp and extending to the tip. Algorithm 1 presents our re-directing and growing steps in detail. In our implementation, we search the nearest fine strands in a limited radius $\tau_s = 3mm$, so $\mathcal{S}^0$ or $\mathcal{S}^1$ could be empty, which means some segments can only grow backward to the scalp (no forward growing). Also, segments that can only grow forward are supposed to be deleted.

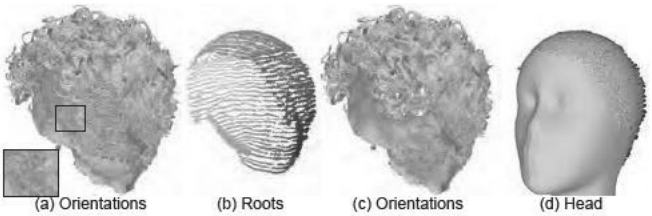

Fig. 1. The pre-processing step includes the underlying hairnet (highlight parts in (a)) removal and head model registration (d). (b) shows the extracted points with higher densities. (c) shows the orientations after removal.

ALGORITHM 1: Grow poor segments

Input: Poor segments set $\{s^p\}$, fine hair strands set $\{\mathcal{S}^f\}$, hair scalp surface Ω
Output: Guide hair strands set $\{\mathcal{S}^g\}$
$\{\mathcal{S}^g\}$.extend($\{\mathcal{S}^f\}$)
foreach $s \in \{s^p\}$ **do**
  $\mathcal{S}^0 \leftarrow$ Nearest fine strand to s[0]; $\mathcal{S}^0[p^0] \leftarrow$ Nearest point;
  $\mathcal{S}^1 \leftarrow$ Nearest fine strand to s[-1]; $\mathcal{S}^1[p^1] \leftarrow$ Nearest point;
  // Remove direction ambiguity
  $s^0 = normalize(s[1] - s[0])$;
  $s^{-1} = normalize(s[-2] - s[-1])$;
  if $s^0 \cdot direction(p^0) < 0$ & $s^{-1} \cdot direction(p^1) < 0$ **then**
    continue;
  end
  if $s^0 \cdot direction(p^0) < 0$ **then**
    $s = s$.reverse; Swap($\mathcal{S}^0$, $\mathcal{S}^1$);
  end
  // Grow segment
  while $distance(s[0], \Omega) > \tau_c$ & $p^0 > 0$ **do**
    s.push_forward($s[0] + (\mathcal{S}^0[p^0 - 1] - \mathcal{S}^0[p^0])$);
    $p^0 = p^0 - 1$;
  end
  while $p^1 < size(\mathcal{S}^1) - 1$ **do**
    s.push_back($s[-1] + (\mathcal{S}^1[p^1 + 1] - \mathcal{S}^1[p^1])$);
    $p^1 = p^1 + 1$;
  end
  $\{\mathcal{S}^g\}$.add(s);
end

3 VOLUME SYNTHESIS

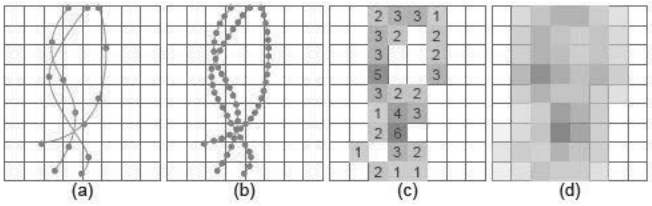

Fig. 2. The processing of volume synthesis. (a) shows synthetic strands with sparse control points. (b) shows interpolated dense points. (c) accumulates counts of points in the voxels. (d) shows the volume after the Gaussian blur.

We synthesize volumes from synthetic hair strands created by artists to evaluate our method. Starting from each synthetic strand, we first densify strand points using Hermit interpolation and make sure the distances between two connected points are less than the half size of our synthesized voxel, as shown in Fig. 2 (b). Then, for each voxel, we count the number of strand points in it as the initial density value. Since the motion blur and scanning noise happens during actual CT scanning, we also add Gaussian blur to our synthesized volume. Fig. 2 (c) and (d) show the volume before and after Gaussian blur, respectively.

We also show visualizations of our synthesized volumes in Fig. 3. The results show that our synthesized volumes from the input strands are very similar to the synthesized volumes from our reconstructed strands and prove our method can recover strands and synthesis volumes correctly.

4 IMPLEMENTATION DETAILS

Neural Interpolation. Our network includes an Encoder network **Enc** and a Decoder network **Dec**. **Enc** consists of 8 layers of 1D CNN and outputs a latent code with the size of 128. Similar to NeuralStrands we implement **Dec** as a modulated SIREN **Dec** takes a latent code and a parameter $t \in [0, 1]$ as inputs and outputs a direction vector with the step size. Same with we integrate the output directions with sorted uniformly sampled t to recover the strand geometry.

For fitting curly strands better, we increase the point resolution of the dataset to 256 and use realistic and complex synthetic hairstyles created from *datagen*[1] as our training dataset. We use the same training loss and training strategy with .

In our interpolation stage, we set the texture resolutions to 2048^2.

Optimization. We set the target number of points after differentiable interpolation as 64. The size of the local batch is 1600 neighbor strands. For each batch, we build a partial target point cloud by selecting $k_o = 16$ nearest points from the original target points set $\mathcal{P}_d$, for every point in the local strands batch. After each iteration, we update the partial target point via re-computing the k nearest points. We stop the optimization iteration when the total loss $\mathcal{L}$ does not decrease more than $\epsilon = 0.05$. We use Adam (betas(0.9,0.999)) as our optimizer with the learning rate 0.1, and the optimization usually converges in 80 iterations.

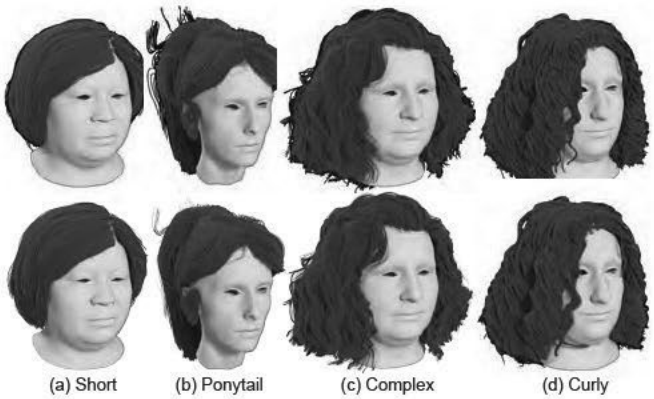

Fig. 3. The visualizations of synthesized volumes. The first row shows the synthesized volumes from synthetic hair strands, and the second row shows the synthesized volumes from our reconstructed strands.

5 MORE VISUAL RESULTS

We show side-by-side comparisons between the volume rendering and our reconstructed strands, of all our ten scanned wigs, in Fig. 4. Our results match the input volumes well in both shape and geometry details. By using physics-based rendering, our rendered results look very realistic.

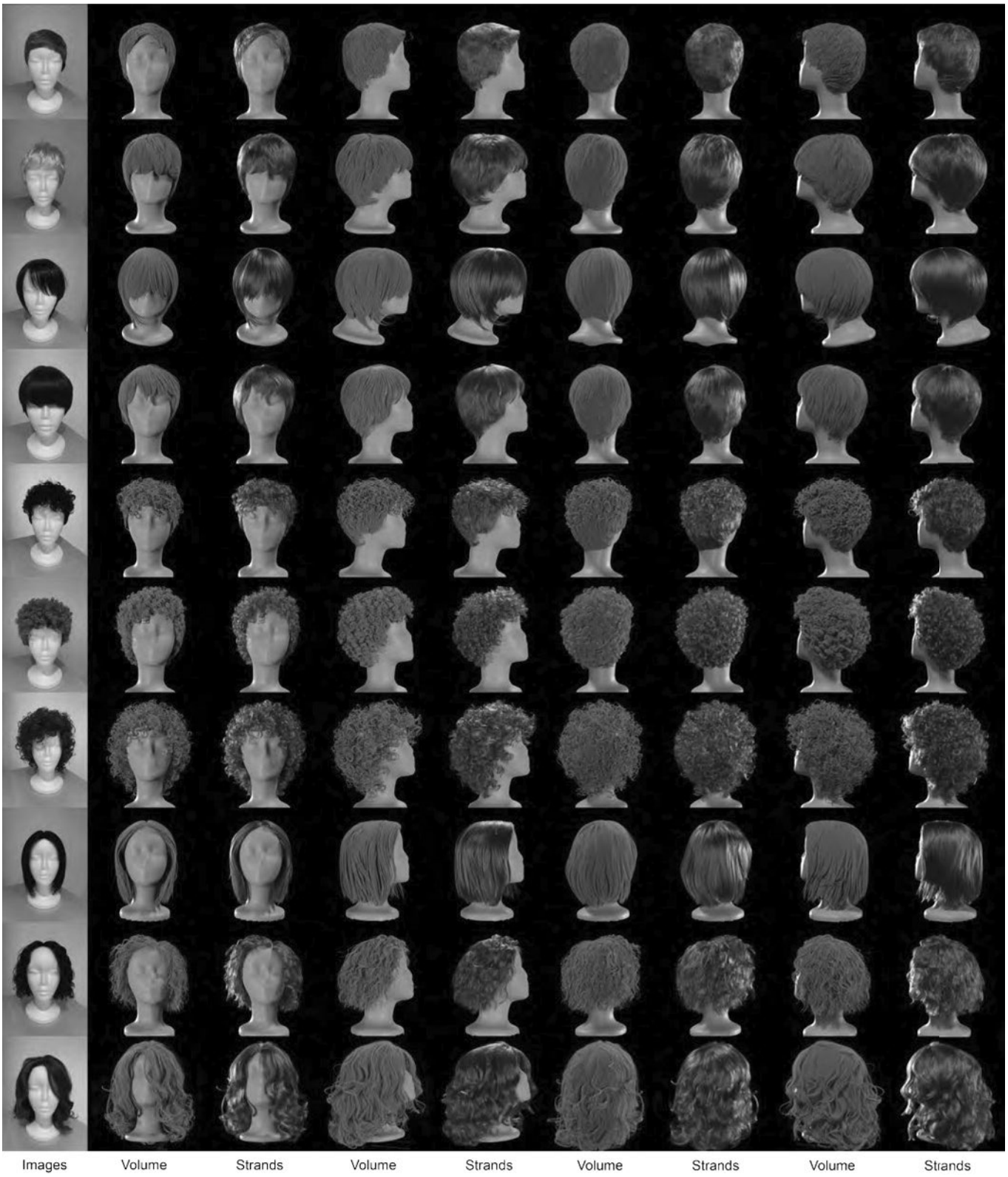

Fig. 4. Visual comparisons between the volume rendering and our reconstructed strands. We also show captured images of our used wigs in the first column.