Shape Adaptation for 3D Hairstyle Retargeting

Lu Yu, Zhong Ren, Youyi Zheng, Xiang Chen*, and Kun Zhou, *Fellow, IEEE*

Abstract—It is demanding to author an existing hairstyle for novel characters in games and VR applications. However, it is a non-trivial task for artists due to the complicated hair geometries and spatial interactions to preserve. In this paper, we present an automatic shape adaptation method to retarget 3D hairstyles. We formulate the adaptation process as a constrained optimization problem, where all the shape properties and spatial relationships are converted into individual objectives and constraints. To make such an optimization on high-resolution hairstyles tractable, we adopt a multi-scale strategy to compute the target positions of the hair strands in a coarse-to-fine manner. The global solving for the inter-strands coupling is restricted to the coarse level, and the solving for fine details is made local and parallel. In addition, we present a novel hairline edit tool to allow for user customization during retargeting. We achieve it by solving physics-based deformations of an embedded membrane to redistribute the hair roots with minimal distortion. We demonstrate the efficacy of our method through quantitative and qualitative experiments on various hairstyles and characters.

Index Terms—3D hairstyle retargeting, shape adaptation, multi-scale solving, embedded membrane, constrained optimization.

---

1 INTRODUCTION

3D HAIRSTYLE is an essential asset for digital characters. In games and VR applications, it is often demanding to author an existing hairstyle for a novel character since the creation of high-quality 3D hairstyles requires laborious effort and expertise [1], [2], [3]. However, manual retargeting is a non-trivial task as a 3D hairstyle exhibits complicated geometries and spatial interactions, and the artist must ensure a faithful preservation of all these details during authoring (see Figure 2 b-c).

In this paper, we present an automatic shape adaptation method for 3D hairstyle retargeting. Given the hairstyle of a source character, our method can adapt it to diverse targets with distinct body shapes (see Figure 1). Although there are 2D hairstyle transfer methods like [4], [5], [6] investigated with the recent progress of StyleGAN [7], [8], the pixel-level manipulations cannot be applied to 3D hairstyles. As a 3D hairstyle possesses various per-strand shapes, complicated inter-strand relationships, and complex interactions with the character, a key challenge is how to preserve such geometric information in accordance with the shape variance between the source and target characters. Therefore, we formulate the adaptation process as a constrained optimization, where all the shape properties and spatial relationships are converted into individual objectives and constraints.

Shape adaptation can be computationally intractable for 3D hairstyles. In order to express sufficient fine details, a 3D hairstyle often possesses hundreds of thousands of strands and millions of particles. With such a high-resolution model, the global coupling between all hair strands leads to a huge system to solve. To tackle this problem, we present a multi-scale solving strategy to organize the hair strands into two levels of detail and restrict global solving to the coarse level. With the coarse adaptation ready, solving the fine details is made local and parallel.

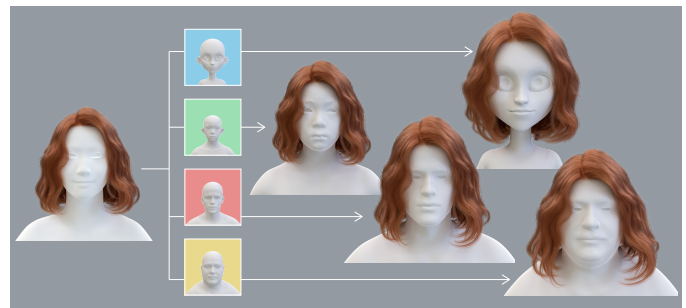We further allow the users to customize the position and

Fig. 1: Retargeting a 3D hairstyle to different characters.

shape of the hairline to obtain more variants. A critical step to support hairline edits is redistributing the hair roots by stretching a scalp membrane embedded in the head surface. To minimize the distortion of such a redistribution, we resort to a physics-based formulation, i.e., optimizing an elasticity energy parameterized intrinsically in the surface space.

We conduct experiments on various hairstyles and characters. The qualitative and quantitative results demonstrate the fidelity and diversity of our retargeting method. Furthermore, various hairline edits from users are supported, and the multi-scale solving achieves a two-orders-of-magnitude speedup, compared with the full solving of all strands.

Our main contributions are:

- The first shape adaptation framework based on constrained optimization for 3D hairstyle retargeting;
- A multi-scale solving method to enable the optimization of high-resolution hairstyles;
- A physics-based embedded membrane deformation method to support hairline edits.

2 RELATED WORK

2.1 3D Hair Modeling

With the increasing demand for realistic hairstyles in digital humans, 3D hair modeling techniques have been continuously improved [9], [10]. High-quality reconstruction of 3D

- *L. Yu, Z. Ren, Y. Zheng, X. Chen and K. Zhou are with the State Key Lab of CAD&CG, Zijingang Campus, Zhejiang University, Hangzhou, China 310058.*

* Corresponding author. E-mail: xchen.cs@gmail.com

hairstyles often requires a complex acquisition setup, e.g., the LED light sources, robotic arms, SLR camera arrays [1], [2], or even computed-tomography scanners [3]. The obtained hairstyles are of satisfying fidelity, but the expensive equipment or long capturing time restricts the acquisition scalability. Recently, data-driven and deep learning methods are adopted to generate 3D hairstyles from single-view [11], [12], [13], [14] or multi-view images [15], [16], and the depth information is also used [17], [18]. These image-based methods are straightforward to scale up. However, the produced 3D hairstyles cannot yet be compared with the acquisition results in terms of precision and quality, mainly due to the limited 3D hair models for training and the inherent domain gap between the real images and the synthetic ones.

Generative models for 3D hairstyles have been recently developed. The variational autoencoder is often employed to learn the representation model for individual strands and hairstyles. Hairstyle variants can be efficiently synthesized through a latent-space sampling or interpolation [19]. Text conditioning to a diffusion network is also enabled to help guide the hairstyle generation [20]. These works conduct the training on a canonical head shape that would be implicitly embedded within the network, and the synthesis is bound to this shape inherently. Our method complements them since we can effectively retarget the ones they produce to novel characters, vastly increasing the variants.

2.2 2D Hairstyle Transfer

Given the data scarcity of high-quality 3D hairstyles, reusing existing models has practical significance. Specifically, methods are investigated for hairstyle transfer in the image space [4], [5], [6], [21]. MichiGAN [4] designs individual condition modules to compose the shape, structure, appearance, and background attributes of different source images into a new one. Differently, LOHO [21] optimizes the latent space of StyleGAN2 [8] with a gradient orthogonalization strategy to disentangle the hair attributes. Based on the latent space of StyleGAN2, Barbershop [5] enhances its spatial encoding with a novel structure tensor and adopts a semantic alignment step to improve the blending coherence. StyleYourHair [6] employs facial-keypoints detection and a dedicated local-style-matching loss to deal with the pose difference between the reference images. Those methods do not apply directly to 3D hairstyle transfer. First, the latent-space hair manipulation depends on an underlying representation model. This necessitates a large training set that is publicly available to facial images but not 3D hair models. Second, 2D hairstyle transfer focuses on the visual plausibility of the pixel-space compositions under a given viewpoint, while a 3D method explicitly adapts the hair geometries. During the adaptation, we must ensure the full fidelity of hair shapes and preserve their spatial interactions with the characters. Moreover, our method does not rely on any 3D hair dataset. In fact, it could be leveraged to vary existing models for data augmentation.

2.3 3D Garment Retargeting

Although there has yet to be direct research on 3D hairstyle transfer, to the best of our knowledge, a few methods have been developed for resizing the garments to fit a target body. Earlier methods adopt skinning-type strategies to warp the garment surface mesh along with the character body [22], [23], [24]. The warping result is tightly bound to the shape variance of the body mesh, leading to undesirable artifacts for loose garments. Then, Meng et al. [25] propose manually adding user guidances to regularize the garment shape, but merely using the annotations on sparse views is insufficient to eliminate all the shape artifacts. Brouet et al. [26] presents a gradient-based objective and an operation for deformation projection to explicitly preserve the shape properties during garment transfer. The techniques from these methods could be borrowed to solve transfer problems for other assets. For example, most of them have defined the spatial relationship between the garment and body, though using diverse ways, and we modify the skeleton-based local positioning strategy presented in [26] to express the hair-body relationship. The methods are not fully applicable to hairstyle transfer due to the significant differences between the geometric representations of hairstyle and garment. The garment is modeled as a set of connected 2D surface meshes, while the hairstyle is composed of many individual 1D strands, all embedded in the 3D space. In this work, we design dedicated energies to model per-strand shapes and inter-strand interactions for hairstyle retargeting. Furthermore, we present a multi-scale solving strategy to deal with the high-resolution nature of hair models in a coarse-to-fine manner, as well as a hair-root relocation algorithm to support the user edits of hairline.

Recently, data-driven and learning-based methods have been utilized to retarget the pattern-based garment designs [27], cloth dynamics [28], [29] and wrinkles [30], [31]. These methods often require a training dataset with paired combinations of garment samples over body samples. Our method can be an optional tool to create such a dataset for hair.

3 OVERVIEW

We aim to retarget strand-based 3D hairstyles. Specifically, given two characters with similar poses, our method takes an arbitrary hairstyle customized for the source and adapts its shape to the target. The produced hairstyle preserves a consistent look with the original hair while being compatible with the target character.

Due to the inherent shape differences between the source and the target character, naively transferred hairstyles often possess one or many of the following issues.

The *single-strand shape* is visually crucial to 3D hairstyles. Without any constraints for it, a position-based optimization on the strand particles would easily damage the per-strand shape properties, e.g., curliness.

The *inter-strand relationship* represents the spatial interactions between different strands in a hairstyle. Hair strands are topologically independent, but their relative locations in 3D are visually salient. Thus, processing each strand without any mutual constraints would quickly lose this global shape information (see insets of Figure 2c).

The *hair-body relationship* describes the relative positions between the hair strands and the character model, especially the head, neck, shoulder, and upper arms. Such spatial relationships could be inherently semantic, e.g., a side fringe beside the cheek. To faithfully capture these abundant mutual semantics, solely considering the hair-roots positioning on the scalp during transfer is far from satisfactory (Figure 2b).
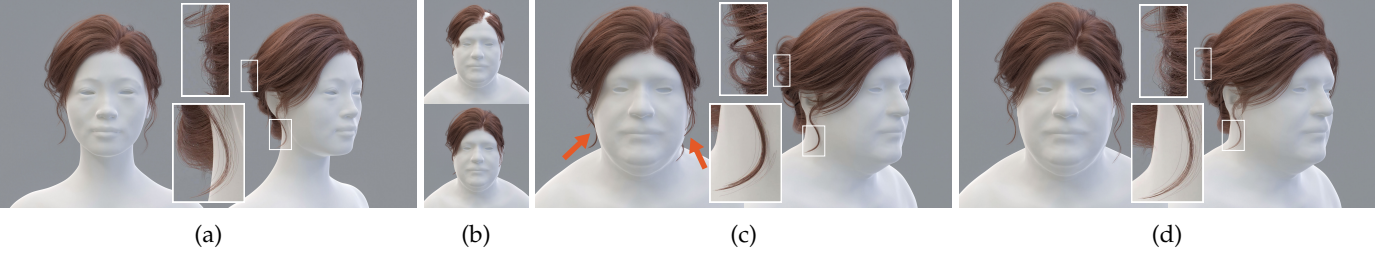
Fig. 2: (a) The source character and hairstyle. (b) A naive retargeting, using either a global translation (up) or per-strand translations based on the barycentric coordinates of hair roots on the scalp (bottom), produces wrong proportions between the hair and the character. (c) With a global affine transformation and per-strand translations, the large-scale proportion improves. However, the retargeted hairstyle still has the wrong hair-body relationship (e.g., see the arrows: the tight attaching and interpenetration artifacts between the side fringes and the cheek) and inter-strand relationship (e.g., see the zoom-ins: the hallucinated and exaggerated bundle concentrations). (d) Our method produces a faithful retargeting result exhibiting, e.g., side fringes with reasonable gaps to the cheek and bundles with accurate, fluffy shapes.

Things get even worse when the transferred hairs *penetrate* the body, which leads to severe artifacts (Figure 2c).

The above shape factors could conflict; for example, strict preservation of the hair-body relationships could harm the single-strand shapes. Thus, the retargeting method should sensibly balance those factors and produce a hairstyle that is visually faithful to the source models and spatially adaptive to the target character, e.g., penetration-free. Meanwhile, the method must be fast to compute and flexible for user edits.

To this end, we carefully design a two-stage method for 3D hairstyle retargeting (Section 4). First, we compute an initial hairstyle that preserves the hair-body relationship as much as possible but ignores all the other shape factors. Second, starting from the initial transfer result, we simultaneously optimize all the mentioned shape factors by solving an iterative quadratic programming problem to obtain the final hairstyle.

To accelerate the optimization, we build a hair hierarchy to decouple the inter-strand relationships, enabling approximate but efficient multi-scale solving (Section 5). To support user edits of the hairline, we present a physics-based embedded membrane deformation to redistribute the hair roots on the target scalp with minimal distortion (Section 6).

4 3D HAIRSTYLE RETARGETING

We represent a 3D hairstyle as independent strands, where each strand is composed of sequentially connected particles. We stack per-particle 3D positions of the original and the transformed hairstyle to vectors $\bar{\mathbf{p}}, \mathbf{p} \in \mathbb{R}^{3 \times N}$, where N is the total number of particles. We denote the source and target character mesh by M_s and M_t, and their 3D positions by $\bar{\mathbf{q}}$ and $\mathbf{q}$, respectively. Moreover, we assume that M_s and M_t have the same topology.

For the 3D hairstyle retargeting, we seek to find a good $\mathbf{p}$ given $\bar{\mathbf{p}}, \bar{\mathbf{q}}$ and $\mathbf{q}$. Mathematically, we formulate this *shape adaptation* as a constrained nonlinear optimization problem.

$$
\begin{aligned}
\min_{\mathbf{p}} \quad & E_\text{strand-shape} + \alpha E_\text{inter-strand} + \beta E_\text{hair-body} \\
\text{s.t.} \quad & C_\text{root}, C_\text{penetration}
\end{aligned}
\tag{1}
$$

4.1 Initial Transfer

The nonlinear optimization necessitates an initial value of $\mathbf{p}$. In order to stabilize and accelerate the convergence, a good initial hairstyle should possess a smooth shape and correct proportions. Thus, we compute such a value by conforming to the source hair-body relationship.

We first generate the source and target skeleton K_s and K_t, with the same topology, by fitting the SMPL [32] to the character models M_s and M_t, respectively.

Next, we adapt the cloth positioning strategy [26] to localize hairs. For each particle $\bar{\mathbf{p}}_i$ of the source hairstyle, we choose a bone from K_s as the anchor for local positioning:

$$
\min_{b \in \text{bones}} \mathbf{1}_{\mathbf{q}_b \in \Lambda_b} \cdot \|\bar{\mathbf{p}}_i - \bar{\mathbf{q}}_b\| \cdot e^{\sigma \langle \bar{\mathbf{r}}_b, \bar{\mathbf{v}}_b \rangle^2}.
\tag{2}
$$

As shown in Figure 3a, on each bone we find the point $\bar{\mathbf{o}}_b$ that is closest to $\bar{\mathbf{p}}_i$, and we find the intersection between the ray $\overrightarrow{\bar{\mathbf{o}}_b \bar{\mathbf{p}}_i}$ and the source character M_s as the point $\bar{\mathbf{q}}_b$. In the above exponentiation, $\bar{\mathbf{r}}_b = \frac{\bar{\mathbf{q}}_b - \bar{\mathbf{o}}_b}{\|\bar{\mathbf{q}}_b - \bar{\mathbf{o}}_b\|}$ is the direction of the ray, $\bar{\mathbf{v}}_b$ is the direction of the bone, and σ is a constant that we set to 100. In other words, we tend to choose a bone orthogonal to the ray with an intersection point near the hair particle. As this simple metric [26] may choose a faraway bone with false intersection, e.g., a ray starting from clavicle intersects the scalp, we add an indicator function $\mathbf{1}_{\bar{\mathbf{q}}_b \in \Lambda_b}$ to ensure that the intersection point is within a valid surface region. We clip the skinning weights of SMPL for bone b to determine this valid region Λ_b.

For each $\bar{\mathbf{p}}_i$, we record the length $\bar{\eta}_i = \|\bar{\mathbf{p}}_i - \bar{\mathbf{q}}_b\|$ as the local coordinates. We also use the barycentric coordinates of $\bar{\mathbf{q}}_b$ and $\bar{\mathbf{o}}_b$ to find the reference points $\mathbf{q}_b$ and $\mathbf{o}_b$ on target models K_t and M_t. By applying the local coordinates to the reference points, we obtain the target position of particle $\bar{\mathbf{p}}_i$:

$$
\tilde{\mathbf{p}}_i = \mathbf{q}_b + \bar{\eta}_i \cdot \frac{\mathbf{q}_b - \mathbf{o}_b}{\|\mathbf{q}_b - \mathbf{o}_b\|}.
\tag{3}
$$

The position $\tilde{\mathbf{p}}_i$ is a local retargeting due to its individual computation on each particle. This leads to stairstep strands, which hinders the optimization in Equation 1. Therefore, we smooth each strand by solving a discrete Poisson's equation for $\hat{\mathbf{p}}$ with the Dirichlet boundary condition:

$$
\begin{aligned}
& \mathbf{L}_i[\hat{\mathbf{p}}] = \mathbf{L}_i[\bar{\mathbf{p}}], \quad \forall i \in \mathcal{I}_\mathrm{u} \\
\text{s.t.} \quad & \hat{\mathbf{p}}_j = \tilde{\mathbf{p}}_j, \quad \forall j \in \mathcal{I} - \mathcal{I}_\mathrm{u},
\end{aligned}
\tag{4}
$$

where $\mathbf{L}[*]$ is a discrete Laplace operator defined on $\bar{\mathbf{p}}$:

$$
\mathbf{L}_i[f] = \frac{f_{i+1} - f_i}{\|\bar{\mathbf{p}}_{i+1} - \bar{\mathbf{p}}_i\|} - \frac{f_i - f_{i-1}}{\|\bar{\mathbf{p}}_i - \bar{\mathbf{p}}_{i-1}\|},
\tag{5}
$$

and $\mathcal{I}_{\mathrm{u}}$ is the index set of the particles with large Laplacian feature discrepancies from the source:

$$\mathcal{I}_{\mathrm{u}} = \{\, i \mid \|\mathbf{L}_i[\tilde{\mathbf{p}}] - \mathbf{L}_i[\bar{\mathbf{p}}]\|_2 > \epsilon_{\mathrm{s}} \,\}. \tag{6}$$

4.2 Single-Strand Shape

In Equation 1, the first energy term measures how large the strand shapes change during retargeting. At first glance, it is appealing to define the local shape change as the variation of individual strand segments, where each segment connects two adjacent particles in a strand. However, the retargeting often necessitates a proportion change due to the significant shape difference between the source and target character. To this end, we allow a segment to change length but penalize its change in direction:

$$E_{\text{strand-shape}} = \sum_{a,b \in \text{segments}} \left\| \frac{\mathbf{p}_a - \mathbf{p}_b}{\|\mathbf{p}_a - \mathbf{p}_b\|_2} - \frac{\bar{\mathbf{p}}_a - \bar{\mathbf{p}}_b}{\|\bar{\mathbf{p}}_a - \bar{\mathbf{p}}_b\|_2} \right\|_2^2. \tag{7}$$

4.3 Inter-Strand Relationship

The second energy term in Equation 1 measures the change of spatial relationships between different strands. For each hair particle in the source hairstyle, we find the k particles nearest to it in different strands and then build a distance-weighted Laplacian feature to encode the relative positions between them (see Figure 3b):

$$\bar{\mathbf{L}}_i = \sum_{j \in \text{knn}(i)} \omega_j \left(\bar{\mathbf{p}}_i - \bar{\mathbf{p}}_j \right). \tag{8}$$

Considering that closer points imply stronger bindings, we compute the inverse of the distances and normalize them to obtain the Laplacian weights:

$$\sigma_j = \frac{1}{\|\bar{\mathbf{p}}_i - \bar{\mathbf{p}}_j\|}, \quad \omega_j = \frac{\sigma_j}{\sum_{m \in \text{knn}(i)} \sigma_m}. \tag{9}$$

Using the nearest neighbors and Laplacian weights computed above, we generate the same feature for the adapted hairstyle and penalize its change during retargeting:

$$\mathbf{L}_i = \sum_{j \in \text{knn}(i)} \omega_j \left(\mathbf{p}_i - \mathbf{p}_j \right), \quad E_{\text{inter-strand}} = \sum_{i=0}^{N} \|\mathbf{L}_i - \bar{\mathbf{L}}_i\|_2^2. \tag{10}$$

4.4 Hair-Body Relationship

In the first stage of our method (see Section 4.1), we produce an initial transfer result $\hat{\mathbf{p}}$ according to the spatial relationship between the hairstyle and the character model. Besides serving as the initial value for optimization, we also use $\hat{\mathbf{p}}$ to regularize the optimization variable $\mathbf{p}$ in the third energy term of Equation 1:

$$E_{\text{hair-body}} = \sum_{i=0}^{N} \|\mathbf{p}_i - \hat{\mathbf{p}}_i\|_2^2 \tag{11}$$
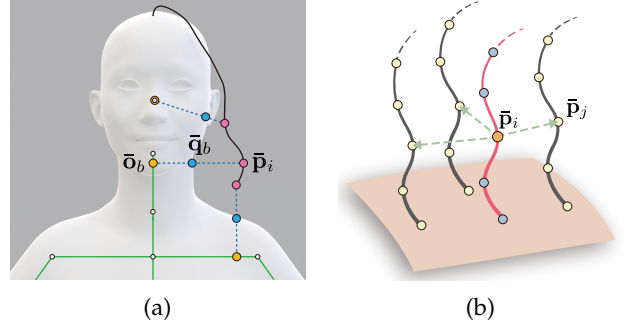

(a)   (b)

Fig. 3: (a) The geometric references for computing the local positioning coordinates of the hair-body relationship. (b) The nearest particles for computing the Laplacian features of the inter-strand relationship.

4.5 Constraints

After the initial transfer, we obtain the position on the target scalp where each strand should grow from, and we fix each root particle to stay at this position:

$$C_{\text{root}}: \quad \mathbf{p}_i = \hat{\mathbf{p}}_i, \quad \forall i \in \mathcal{I}_{\mathrm{r}}, \tag{12}$$

where $\mathcal{I}_{\mathrm{r}}$ is the index set of root particles.

To ensure the transfer is penetration-free, we project each non-root particle $\mathbf{p}_i$ onto the target character model M_{t} to get the projection point $\mathbf{q}_i$ and the surface normal $\mathbf{n}_i$ at this point. Then we constrain the particle $\mathbf{p}_i$ to stay in the upper half-space defined by $\mathbf{q}_i$ and $\mathbf{n}_i$:

$$C_{\text{penetration}}: \quad \langle \mathbf{p}_i - \mathbf{q}_i, \, \mathbf{n}_i \rangle \geq \epsilon_{\mathrm{c}}, \quad \forall i \in \mathcal{I}_{\mathrm{p}}, \tag{13}$$

where ϵ_{c} is a safe-guard for clearance, and $\mathcal{I}_{\mathrm{p}} = \mathcal{I} - \mathcal{I}_{\mathrm{r}}$ is the index set of non-root particles.

4.6 Optimization

The complexity of Equation 1 stems from two places. One is the nonlinear denominator $\|\mathbf{p}_a - \mathbf{p}_b\|_2$ in $E_{\text{strand-shape}}$. The other is the dependency of $\mathbf{q}_i$ and $\mathbf{n}_i$ on $\mathbf{p}_i$ in $C_{\text{penetration}}$. To solve the optimization in a robust and tractable manner, we adopt an iterative update strategy.

In each iteration, we first use the current values of the optimization variable $\mathbf{p}$ to compute each value of $\|\mathbf{p}_a - \mathbf{p}_b\|_2$, $\mathbf{q}_i$, and $\mathbf{n}_i$. Then we fix these values to transform the original formulation of Equation 1 into a classic quadratic programming problem, since now all the objectives are quadratic and all the constraints are linear, with respect to $\mathbf{p}$. Specifically, we use the ADMM algorithm [33] to solve this QP problem and update $\mathbf{p}$. We end the iterations when $\mathbf{p}$ converges.

5 MULTI-SCALE SOLVING

For a typical 3D hairstyle with hundreds of thousands of strands, the Laplacian energy in Equation 10 has to deal with the global coupling between millions of unknown variables. This requires solving a large sparse linear system, which is the bottleneck of the optimization in Section 4.6.

To accelerate the transfer optimization, we build a hierarchy for the hair strands and separate the computation into two scales. A similar strategy is also used in recent studies for real-time hair simulations [34], [35], [36]. Specifically, we
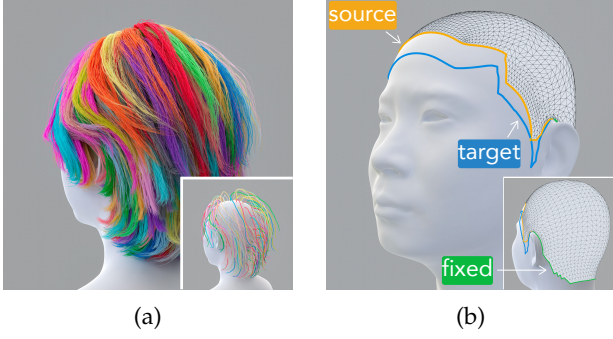
Fig. 4: (a) Hair clustering for multi-scale solving. The inset shows the guide hairs. (b) The front part of the source and target hairlines. The inset shows the fixed part.

apply the clustering method in [37] to the original hairstyle to choose a small set of representative hair strands, i.e., the *guide* hairs (Figure 4a). The rest of the hair strands are called *normal* hairs.

In the coarse scale, we adapt the guide hairs altogether to establish the global shape structure of the retargeting. We use the same strategy introduced in Section 4.6 to optimize Equation 1, but only for the guide hairs. Since their number is quite low, the sparse linear system from the inter-strands coupling becomes much smaller. After this global solving, all the guide hairs have already been adapted to the target character while preserving their shape features and spatial interactions.

In the fine scale, we fix all the adapted guide hairs and use them to constrain the adaptation of normal hairs. To this end, we adjust the Laplacian defined in Section 4.3 to:

$$\bar{\mathbf{L}}_i = \sum_{j \in \text{knn}(i, \mathcal{I}_{\text{g}})} \omega_j \left(\bar{\mathbf{p}}_i - \bar{\mathbf{p}}_j \right), \quad \forall i \in \mathcal{I}_{\text{n}}, \quad (14)$$

where $\mathcal{I}_{\text{g}}$ is the index set of the guide hair particles and $\mathcal{I}_{\text{n}} = \mathcal{I} - \mathcal{I}_{\text{g}}$ is of the normal hair particles. In other words, we find the nearest particles of each normal hair particle *only from* the guide hairs, and use these neighbors to compute the Laplacian. This adjustment breaks the global coupling between the normal strands, and each strand only depends on its nearby guide strands. As a result, we separate a large system into independent parts for individual strands, which can be readily solved in parallel. This decoupling is based on the local coherency of hairstyles and is a local approximation in essence, which is the key to a significant acceleration. The users could balance the transfer speed and fidelity by tuning the number of guide hairs. We optimize Equation 1 with this decoupled Laplacian to achieve the normal hairs adaptation.

6 HAIRLINE EDIT

During hairstyle transfer, users often require design changes for customization. We introduce a method to support user-edits of the hairline, as it is one of the salient features of a hairstyle. To enable this, we first relocate all the hair roots on the target head to obey the user-input hairline by solving a physics-based deformation model, and then we adjust the objectives in Equation 1 to accommodate the relocated hair roots and fine-tune the optimization.

6.1 Hair Roots Relocation

After the initial transfer, we obtain the positions of the hair roots on the target head. Then, we extract its scalp region by collecting all the head mesh triangles with at least one root particle inside. The boundary of the scalp, i.e., the *hairline*, is separated into front/back parts at the ears. We fix the back part and let users draw a curve to specify the desired look of the front part (Figure 4b). We also extract the turning points of the hairline and let the user specify their corresponding positions on the drawing curve. The dense correspondence for each segment between adjacent turning points is found by mapping the normalized arc-length parameters.

With the current shape of the scalp $\mathcal{S}$ as the undeformed configuration and the hairline correspondence $h(*)$ as the boundary condition, we compute the target configuration of $\mathcal{S}$ by solving a deformation problem. As a key insight, the stretching and compression of the scalp should be restricted to the surface space of the head. Therefore, we treat $\mathcal{S}$ as a membrane model embedded in the head surface and minimize a hyperelastic strain energy under Dirichlet boundary condition:

$$\min_{\mathbf{u}} \quad \sum_{t \in \mathcal{T}_{\mathcal{S}}} A_t \cdot \psi \left(\mathbf{F}_t(\mathbf{x}(\mathbf{u}), \mathbf{X}) \right)$$
$$\text{s.t.} \quad \mathbf{x}_i = h(\mathbf{X}_i), \quad \forall i \in \mathcal{I}_{\partial \mathcal{S}}. \quad (15)$$

Here $\mathcal{T}_{\mathcal{S}}$ is the index set of the triangles in $\mathcal{S}$, and A_t is the area of triangle t, whose deformation gradient $\mathbf{F}_t$ depends on the undeformed positions $\mathbf{X}$ and deformed positions $\mathbf{x}$ of the vertices in $\mathcal{S}$. To enable the head surface embedding, we rely on an arbitrary parameterization space, e.g., [38], of the head surface $\mathcal{H}$ to conduct the optimization. The variable to optimize is $\mathbf{u}$, i.e., $\mathbf{x}$'s counterpart in parameterization space. We choose the neo-Hookean membrane model as the strain energy density function ψ, which takes $\mathbf{F}$ as the input. The index set $\mathcal{I}_{\partial \mathcal{S}}$ contains all the hairline vertices.

The deformation gradient $\mathbf{F}_t$ is a constant 3×2 matrix for each scalp triangle t. Now we omit the subscript t and represent a deformation gradient as $\mathbf{F} = \mathbf{d} \cdot \mathbf{D}^{-1}$, where $\mathbf{D}$ is a 2×2 matrix stacking two adjacent edge vectors of the undeformed triangle in a reparameterized 2D space:

$$\mathbf{D} = \left[\hat{\mathbf{X}}_1 - \hat{\mathbf{X}}_0 \mid \hat{\mathbf{X}}_2 - \hat{\mathbf{X}}_0 \right]$$
$$= \begin{bmatrix} \|\mathbf{X}_1 - \mathbf{X}_0\| & \frac{(\mathbf{X}_2 - \mathbf{X}_0) \cdot (\mathbf{X}_1 - \mathbf{X}_0)}{\|\mathbf{X}_1 - \mathbf{X}_0\|} \\ 0 & \frac{\|(\mathbf{X}_2 - \mathbf{X}_0) \times (\mathbf{X}_1 - \mathbf{X}_0)\|}{\|\mathbf{X}_1 - \mathbf{X}_0\|} \end{bmatrix}, \quad (16)$$

and $\mathbf{d}$ is a 3×2 matrix stacking the two edge vectors of the deformed triangle in the 3D space:

$$\mathbf{d} = \left[\mathbf{x}_1 - \mathbf{x}_0 \mid \mathbf{x}_2 - \mathbf{x}_0 \right]. \quad (17)$$

Since the scalp is required to deform on the head surface, a single deformed vertex of $\mathcal{S}$ moves locally inside a triangle τ of $\mathcal{H}$, whose 3D coordinates $\mathbf{x}$ is a barycentric interpolation on vertices of triangle τ (see Figure 5):

$$\mathbf{x} = \left[\mathbf{X}_1^\tau - \mathbf{X}_0^\tau \mid \mathbf{X}_2^\tau - \mathbf{X}_0^\tau \right] \lambda + \mathbf{X}_0^\tau, \quad (18)$$

where λ is the 2D barycentric coordinates computed in the parameterization space of $\mathcal{H}$ for the intrinsic parameters $\mathbf{u}$ corresponding to $\mathbf{x}$:

$$\lambda = \left[\mathbf{u}_1^\tau - \mathbf{u}_0^\tau \mid \mathbf{u}_2^\tau - \mathbf{u}_0^\tau \right]^{-1} \left(\mathbf{u} - \mathbf{u}_0^\tau \right). \quad (19)$$

According to Equations 16-19, $\mathbf{F}$ is a linear function of $\mathbf{u}$.

We solve for the 2D parameters $\mathbf{u}$ of each deformed scalp vertex by minimizing the nonlinear objective in Equation 15 under the constraints from hairline correspondence $h(*)$. In practice, we conduct this optimization using the Projected Newton method [39] with a backtracking line search. The host triangle indices τ must be updated after each Newton iteration according to the current values of $\mathbf{u}$, i.e., the vertex positions in the parameterization space. We use a harmonic map [38] to parameterize $\mathcal{H}$, though the converged result is insensitive to the choice of parameterization (Figure 17). Besides, our embedded membrane resembles the Lagrangian-on-Lagrangian approach [40] that employs the limit geometry of the Loop subdivision to eliminate the discontinuities across triangles. For our problem, the iterative solving in a single parameterization space of the surface is sufficient.

After the scalp deformation, we first move the hair roots by applying their barycentric coordinates to the deformed scalp vertices, and then project them onto the head surface $\mathcal{H}$ to eliminate the discretization error left. By leveraging the physics-based hyperelasticity energy, our relocation strategy enables a natural and smooth diffusion of the given hairline displacements over the whole scalp, while bringing minimal distortions to the original distribution of the hair roots.
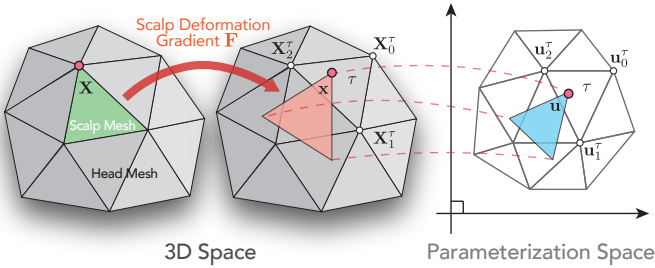

Fig. 5: The scalp region is a subset of the head mesh. We copy it as the undeformed configuration and constrain the scalp mesh to deform *only on* the head mesh. The energy in Equation 15 measures how large the scalp deformation is in the 3D space. Meanwhile, the deformed 3D position $\mathbf{x}$ of a scalp vertex is determined by its 2D coordinates $\mathbf{u}$, which we optimize in Equation 15, in the head parameterization space. In this way, we model the scalp as an embedded membrane.

6.2 Shape Adaptation Tuning

With the relocation of the hair roots ready, we still have to tune the shape adaptation in Section 4 for it. First, we change the right-hand side of the constraints in Equation 12 to the relocated positions of hair roots. Second, the redistributed hair roots make it harder to balance the preservation of individual strand shapes and spatial relationships. Therefore, we introduce an adaptive weight for each non-root particle according to its curve-space position in the strand:

$$\gamma_i = 1 - e^{-\sigma(s_i/r_i)^2}, \tag{20}$$

where s_i is the arc length from particle i to its hair root, r_i is the traveling distance of the hair root during relocation, and σ is a constant that we set to 0.2. The closer a particle is to its hair root, the larger this weight is. Accordingly, we

fine-tune the energy terms for the inter-strand and hair-body relationships:

$$E_{\text{inter-strand}} = \sum_{i=0}^{N} \gamma_i \cdot \left\| \mathbf{L}_i - \bar{\mathbf{L}}_i \right\|_2^2$$

$$E_{\text{hair-body}} = \sum_{i=0}^{N} \gamma_i \cdot \left\| \mathbf{p}_i - \hat{\mathbf{p}}_i \right\|_2^2. \tag{21}$$

Harnessing such tuning to the optimization, we can produce a transfer result with a visually pleasant trade-off between the preservation of different shape factors, under a moderate hairline adjustment.

7 EXPERIMENTAL RESULTS

7.1 Implementation Details

We build our system on a desktop PC with an Intel Core i9-13900K CPU and 128 GB memory. We implement the optimization module with the auto-differentiation tool Chumpy [41] and the OSQP solver [42]. Moreover, we use ANN [43] to find the nearest neighbors and libigl [44] to parameterize the head surface mesh. In Table 1, we list all the hyperparameters for our method, and we set their values by using meters as the unit of model size.

TABLE 1: The hyper-parameters used in our algorithm.

Symbol	Description	Value
α	the weight of $E_{\text{inter-strand}}$ in Eqn.(1)	3e3
β	the weight of $E_{\text{hair-body}}$ in Eqn.(1)	1e3
k	the #neighbors for knn in Eqn.(8)	5
ϵ_{c}	the clearance safeguard in Eqn.(13)	5e-4
ϵ_{s}	the discrepancy threshold in Eqn.(6)	0.3

7.2 Diversity and Consistency

Our method is capable of retargeting diverse 3D hairstyles to characters with distinct body shapes. We exemplify some results in Figure 7, which contains 8 hairstyles with different lengths, curliness, growth directions, split styles, and braid styles, and 3 target characters with different ages, genders, and races. The transferred hairstyles faithfully preserve the shapes and spatial interactions of the sources.

We execute two regression experiments, i.e., the *reflexive test* and *cycle test*, to demonstrate the inherent consistency of our retargeting method. We show these regression results in Figure 6. Visually, they are almost indistinguishable from the original hairstyle. In Table 2, we report their corresponding per-particle distances and per-segment angle differences.

(a) source A   (b) A → A   (c) A → B   (d) A → B → A

Fig. 6: (a) The character A and its hairstyle. (b) *reflexive test*: Retargeting the hairstyle from A to itself. (c) Retargeting the hairstyle from A to B. (d) *cycle test*: Retargeting the hairstyle from A to B and then back to A.

Fig. 7: Our method is capable of retargeting diverse hairstyles. The first row shows 8 source hairstyles, and the next rows show their adaptation variants by retargeting to 3 different target characters. See the supplemental for more results.

TABLE 2: Quantitative errors of the regression experiments.

	Mean Distance	Mean Angle
Regression 1 (reflexive)	7.85×10^{-8} m	3.75×10^{-6} rad
Regression 2 (cycle)	1.97×10^{-3} m	4.64×10^{-3} rad

7.3 Ablation Study

7.3.1 Constrained Optimization

We conduct qualitative experiments to justify the necessity of individual energy terms and constraints in Equation 1.

Figure 8a demonstrates the efficacy of the energy term $E_{\text{strand-shape}}$ in Equation 7. Without it, the strand curvatures quickly distort during the optimization due to the preservation of other shape factors.

Besides the shapes of single strands, the spatial relationships between adjacent strands are significant visual clues to a hairstyle. Accordingly, Figure 8b shows the efficacy of $E_{\text{inter-strand}}$ in Equation 10. Without this term, the clustering and spread-out effects are easily damaged during the optimization.

Different characters have distinct shapes of head, face, neck, and shoulder. The spatial positionings of the strands relative to the body would quickly become inaccurate after retargeting if the energy term $E_{\text{hair-body}}$ in Equation 11 is not incorporated during the optimization (see Figure 8c).

Correct collision handling is crucial for successful retargeting. As shown in Figure 8d, the interpenetration artifacts emerge soon if the hard constraint $C_{\text{penetration}}$ in Equation 13 is not incorporated.

In contrast, with our complete formulation (Equation 1), all the energy terms and constraints mutually confine each other to achieve a balanced shape preservation. To provide a quantitative reference, Figure 9 compares the per-particle and total objective values computed on results of Figure 8.

7.3.2 Local Positioning Strategies

The local positioning of hair particles relative to the character body determines the quality of initial transfer (Section 4.1). In Figure 10, we compare three local positioning strategies, i.e., [25], [45], and our modified version of [26]. Specifically, [25] finds the closest triangle on the body to each hair particle as its anchor, while [45] builds this correspondence by searching the coverage volume defined on each triangle, and the modified [26] selects the best bones to compute the positioning rays. We find that the modified [26] provides the most consistent local-positioning across all the hair particles, due to the fact that the body shapes usually differ much but the reference bones are rather stable.

Moreover, all the above local positioning strategies produce sharp turning points in the initial transfer result (Figure 11), whose smoothness is crucial for ensuring the convergence of the following optimizations. This necessitates an extra step for adaptive smoothing (Equation 4) right after recovering the local positionings on the target character.

7.4 Multi-scale Transfer

The ablation study shows that preserving the inter-strands relationship is crucial for retargeting fidelity. However, the corresponding Laplacian energy requires solving a large linear system from the global coupling, which makes the whole optimization intractable for high-resolution hairstyles. For a typical curly hairstyle with 113K strands and 2.9M particles, the global solver consumes up to 86GB of memory during optimization and requires over 1 hour to complete the entire retargeting. Fortunately, our multi-scale solver introduced
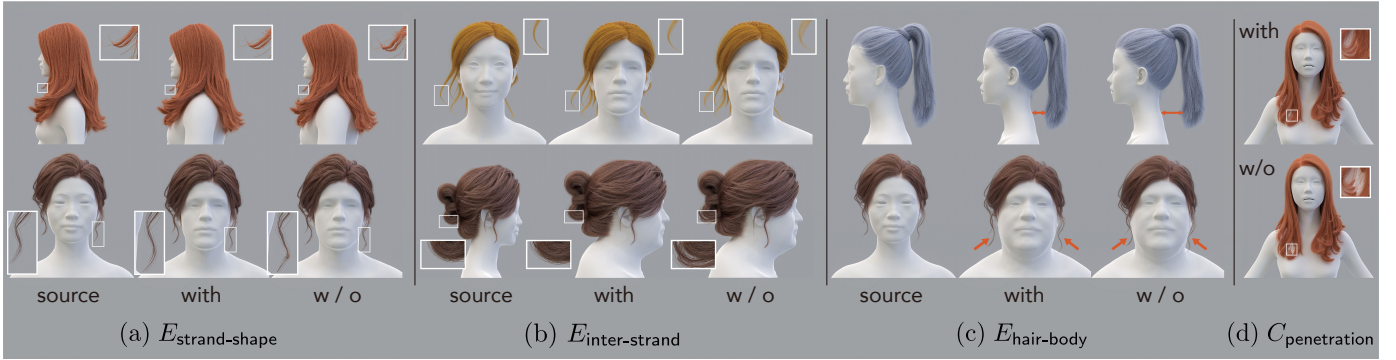
Fig. 8: Ablation study for the objectives and constraints used in the shape adaptation optimization (Equation 1).
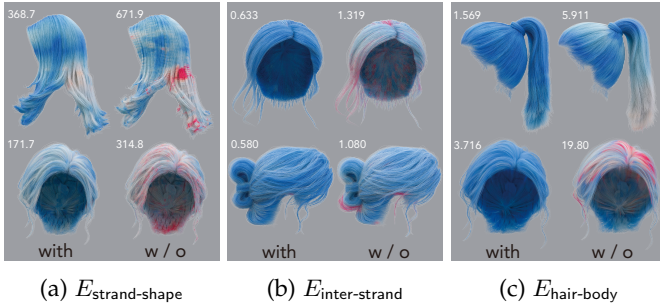

(a) $E_{\text{strand-shape}}$ (b) $E_{\text{inter-strand}}$ (c) $E_{\text{hair-body}}$

Fig. 9: The objectives computed on results of Figure 8.
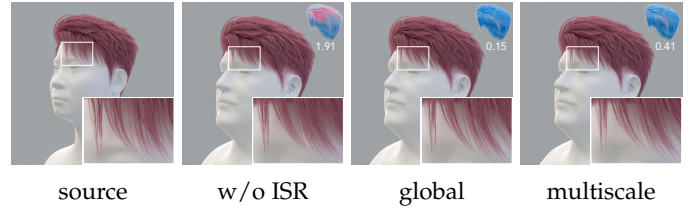

source   w/o ISR   global   multiscale

Fig. 12: The adaptation without the inter-strand relationship (ISR) is over-smoothing. With ISR enabled, both the global and multi-scale solvers produce good fidelities. At the upper right, we show the per-particle and total value of $E_{\text{inter-strand}}$. Though the multi-scale solver resorts to the guide hairs and optimizes a decoupled version of the inter-strand energy, a sufficient decrease is still obtained.

source   modified [26]   [45]   [25]

Fig. 10: Comparison of different local-positioning strategies.

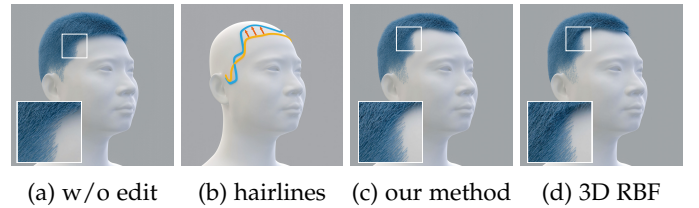(a) w/o edit   (b) hairlines   (c) our method   (d) 3D RBF

Fig. 13: (a) The transfer result without hairline edits. (b) The source/target hairlines. (c-d) Hairline edits with our embedded membrane and 3D RBF interpolation, respectively.

7.5 Transfer with Hairline Edits

Our method supports diverse hairline edits. Figure 14 shows the retargeting results of 3 distinct hairstyles for 6 common hairline types, and Figure 15 shows more edits.

The hair roots relocation step introduced in Section 6.1 is the key to retargeting with hairline edits. We thus compare our embedded membrane formulation in Equation 15 with several alternative models to show its superiority.

In Figure 13, we compare to an RBF-based 3D interpolation method, choosing the thin plat spline, i.e., TPS, as the radial basis function. Specifically, we sample a sparse set of vertices inside the scalp to fix, i.e., zero displacements. Then, we combine them with the hairline vertices as the seeds and use their prescribed 3D displacements as the known data to fit an 3D displacement field based on RBF. Finally, we apply this 3D displacement field to all the free vertices of the scalp and reproject the displaced vertices onto the head to obtain

[45]   [25]   [26]   [26] + smooth

Fig. 11: Sharp turnings necessitates an adaptive smoothing.

in Section 5 consumes only 1.2GB of memory and takes less than 2 minutes to optimize, which is *two orders of magnitude* faster. Figure 12 shows that the retargeting results produced by the global and multi-scale solver do not have significant visual differences. See the supplemental for more results.

Fig. 14: Our method supports diverse hairline edits. The first column shows the source hairstyles. The following columns show their retargeting results to a new character with 6 common hairline types. See the supplemental for more results.

Fig. 15: The sideburns (left) and baldness (right) edits.

TABLE 3: Density changes of the hair roots. For 10 hairstyles under 2 hairline edits (S for straight, W for widow's peak), we compute the L_1 and L_∞ norms of the density change vectors and report the *min*, *max*, and *avg* values across all the combinations. Compared with the alternatives (Figure 16), our method produces much smaller distortions consistently. LSCM [46] is used for parameterization. See the supplemental for per-hairstyle data.

| | $\|\cdot\|_1$ of density change | | | $\|\cdot\|_\infty$ of density change | | |
	Our	Harmonic	RBF(2D)	Our	Harmonic	RBF(2D)
min(S)	**0.015**	0.034	0.041	**0.279**	0.369	0.322
max(S)	**0.048**	0.122	0.152	**1.023**	1.706	2.886
avg.(S)	**0.035**	0.073	0.081	**0.457**	0.672	0.719
min(W)	**0.019**	0.034	0.047	**0.280**	0.435	0.372
max(W)	**0.052**	0.125	0.126	**0.991**	1.706	2.742
avg.(W)	**0.035**	0.071	0.080	**0.447**	0.816	0.755

the relocated hair roots. Figure 13d shows that this method produces unnatural compression and distortion of the hair distribution.

The reprojection mentioned above is necessary because the interpolated vertex displacement is free in 3D, while the head surface is intrinsically 2D. To avoid such reprojection, we instead build a 2D displacement field in the parameterization space of the head. Specifically, we experiment with two different methods. One is to use an RBF interpolation as above but in 2D. The other is to solve a harmonic function on the parameterized scalp mesh with the same boundary conditions as our embedded membrane formulation. Both methods produce a 2D displacement field to move the scalp vertices in the parameterization space and recover their 3D positions. Since the hair-root relocation effects of these two methods depend on the parameterization, we compare two choices in Figure 16, i.e., LSCM [46] and ARAP [47].

The uniformity of the scalp deformation determines the quality of the hair-root relocation. Figure 16 compares the scalp mesh deformation generated by our embedded membrane and all the alternatives. In this figure, we also show the density change of the hair roots over the scalp region. Specifically, for each triangle, we define the average density as the number of hair roots in this triangle divided by its area. We compute per-triangle density values on the initial and deformed mesh, respectively, and stack all their relative changes into a vector. Table 3 sums up the L_1 and L_∞ norms of the density change vectors across 10 hairstyles under 2 different hairline edits. The smaller the relative change, the less distortion the deformation introduces, and the smoother the hair-root relocation is.

Though our embedded membrane also requires a surface parameterization to optimize the scalp vertex, the deformation energy itself is defined in the 3D space. Therefore, our method always converges to similar results no matter which parameterization algorithm is used. Figure 17 compares the scalp deformation with four popular parameterizations, and the results are consistent.

7.6 Performance

In practice, given the source character and the 3D hairstyle customized for it, we first extract the necessary information in advance for subsequent retargeting tasks. Such preprocessing depends on these source models only and is agnostic of any other characters. Specifically, we compute the local references and coordinates for hair particle positioning with respect to the body, the guide hairs for multi-scale solving, and the knn sets for hair Laplacian. Harnessing the extracted
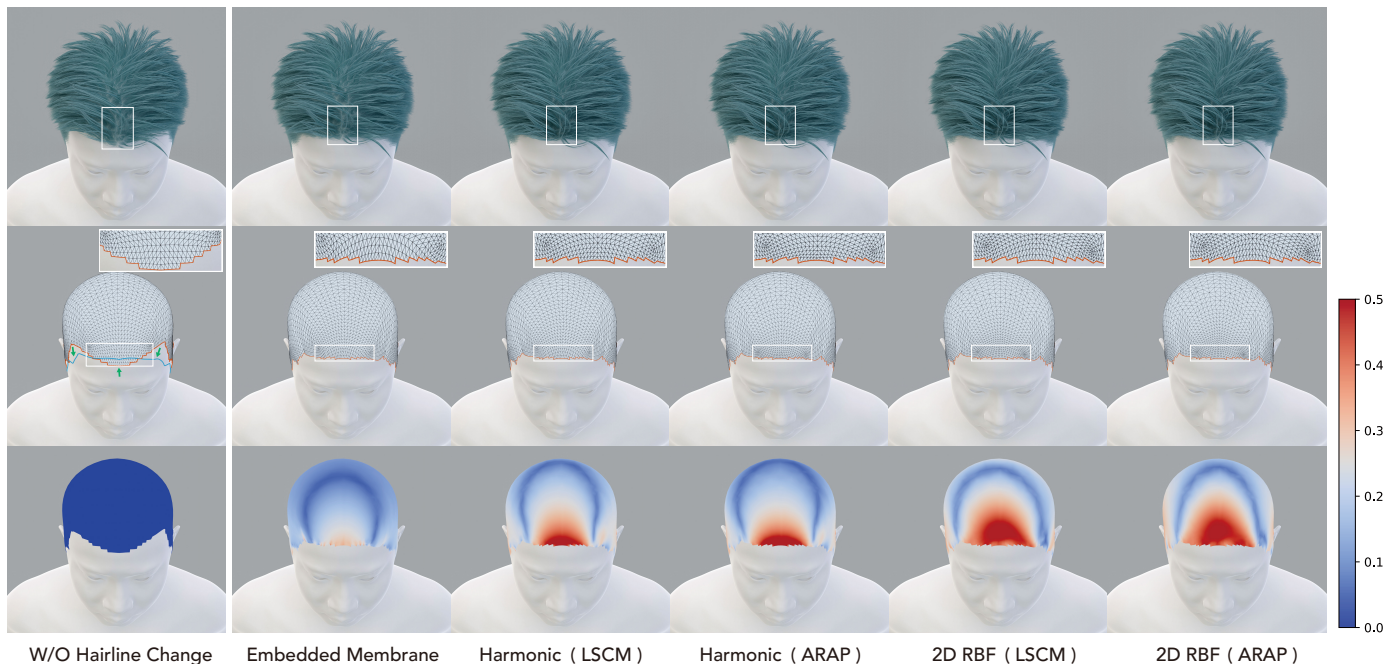
Fig. 16: From up to bottom: the produced 3D hairstyles, the deformed scalp mesh, and the hair-root density changes. The first column is the ground truth, i.e., retargeting without hairline edits. The following columns show the retargeting results with the same hairline edit (see row 2, column 1), using different hair-root relocation methods. The name in the parenthesis represents the surface parameterization method used. Compared with our embedded membrane method, the alternatives produce much more compressions and distortions in the scalp (see both the meshes and the density errors).
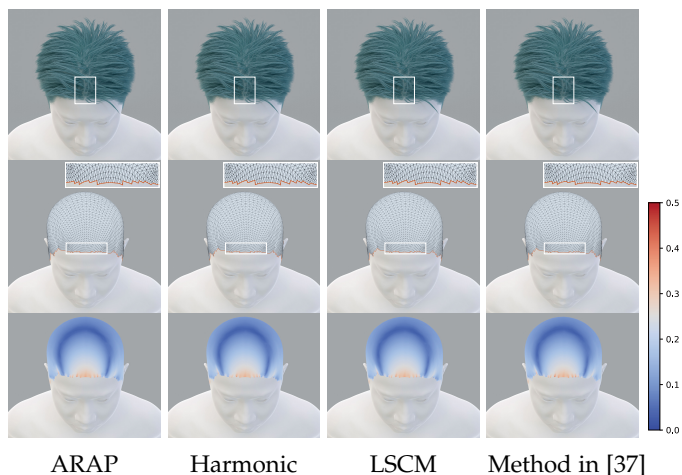

Fig. 17: Retargeting results under the same hairline edit by using our embedded membrane with four different parameterizations, i.e., ARAP [47], Harmonic map [38], LSCM [46], and the one introduced in [37].

information, we obtain an *adaptable* hairstyle ready for retargeting. At runtime, we adapt this hairstyle to an arbitrary target character, with an optional user edit of the hairline, by solving the constrained optimizations (Equation 1, 15).

Table 4 shows the model complexities of all the experimented 3D hairstyles and their transfer time. The QP solving dominates the runtime computation. Although the most complex hairstyle contains more than $110K$ strands and $8M$ particles, our method produces a high-fidelity transfer result in 5 minutes at runtime, ensuring its practical use in digital

human applications.

8 CONCLUSION, LIMITATION, AND FUTURE WORK

We present a method to automatically adapt a 3D hairstyle from the source character to a distinct target, and we design a constrained optimization framework to achieve this. With dedicated energies and constraints, the adapted 3D hairstyle can faithfully preserve the shapes and spatial interactions of the source models. Considering the high-resolution nature of 3D hairstyles, we build a two-level hierarchy for strands and employ a multi-scale solving strategy that substantially accelerates the nonlinear optimization. Moreover, to satisfy the user requirements for hairline edits, we apply an embedded membrane formulation to redistribute all the hair roots uniformly. The experiments demonstrate the effectiveness of our method.

Our method requires the same topology for the source and target characters, which is necessary for transferring the spatial positioning conditions of the hair-body relationship. However, algorithms like [48], [49] could be conducted first
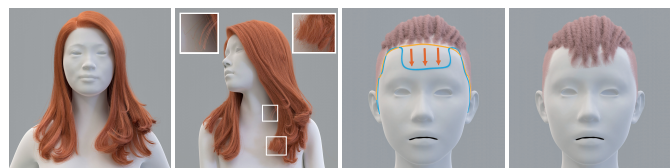

Fig. 18: (left) The source hairstyle and its transfer to another character with a large variance of pose, where artifacts arise. (right) Drastic hairline edits lead to unnatural distortions.

TABLE 4: **Runtime Statistics.** Columns from left to right: the model name, the number of hair strands/particles, the number of guide strands, the preprocessing time, the initial transfer time, the hair-root relocation time, the multi-scale solving time, and the total runtime (in seconds). We also list the full global solving time and the speedup of the multi-scale solving.

Hairstyle	#Strands / #Particles	#Guides	Preprocess	Runtime				Full Solving	Speedup
				Initial Transfer	Hair-root Relocation	Multi-scale Solving	Total		
Short hair (Fig.7, col 1)	65,819 / 460,376	302	6.66s	6.35s	N/A	14.20s	20.55s	440.58s	31×
Dreadlocks(Fig.7, col 7)	101,763 / 4,986,476	309	74.37s	21.41s	N/A	113.73s	135.14s	6,960.05s	61×
Spiky (Fig.7, col 2)	109,097 / 2,218,745	318	30.30s	11.63s	N/A	52.12s	63.75s	3,061.56s	59×
Side fringe (Fig.7, col 3)	112,962 / 2,961,992	302	44.0s	14.67s	N/A	73.65s	88.32s	3,940.99s	54×
Ponytail (Fig.8, row 1, col 7)	85,259 / 2,837,372	297	61.98s	12.90s	N/A	51.59s	64.49s	5,726.29s	111×
Long hair 1 (Fig.7, col 6)	49,991 / 7,229,645	312	165.66s	38.41s	21.23s	148.70s	208.34s	13,810.93s	93×
Long hair 2 (Fig.7, col 8)	96,089 / 8,473,200	316	105.52s	34.47s	25.92s	207.79s	268.18s	8,021.95s	39×
Updo 1 (Fig.2)	86,847 / 5,542,057	308	122.62s	21.30s	26.71s	177.80s	225.81s	10,337.34s	58×
Curly (Fig.7, col 4)	117,364 / 4,456,984	316	121.66s	15.24s	22.67s	130.31s	168.22s	9,643.62s	57×
Updo 2 (Fig.7, col 5)	111,042 / 6,769,042	302	167.06s	26.27s	23.34s	205.53s	255.14s	18,245.06s	89×
Med-Len (Fig.1)	84,268 / 6,207,432	329	113.78s	23.65s	20.03s	120.52s	164.20s	19,890.22s	165×

to build a dense correspondence when the character models have different topologies.

We also require the source and target characters to have similar poses. Although the optimization objective is pose-variant, our method is insensitive to a minor difference in pose. However, when significant pose differences arise (see Figure 18 left), the physics-based hair simulation might be feasible to produce visually plausible results [50], [51].

The quality of hair-root redistribution could deteriorate under a drastic hairline edit (see Figure 18 right). In such a case, the texture synthesis technique for hair, e.g., [37], is a better choice.

Due to its robust and effective retargeting capability, our method can be regarded as a data augmentation tool to increase the variances and capacities of existing hair datasets, e.g., [11]. Training neural networks on such augmented data is beneficial to advance interactive VR techniques for digital humans. Moreover, making our method differentiable could provide a semantic loss to enable self-supervised learning.

ACKNOWLEDGMENTS

REFERENCES

[1] S. Paris, W. Chang, O. I. Kozhushnyan, W. Jarosz, W. Matusik, M. Zwicker, and F. Durand, "Hair photobooth: geometric and photometric acquisition of real hairstyles." ACM Trans. Graph., vol. 27, no. 3, p. 30, 2008.

[2] L. Luo, H. Li, and S. Rusinkiewicz, "Structure-aware hair capture," ACM Transactions on Graphics (TOG), vol. 32, no. 4, pp. 1–12, 2013.

[3] Y. Shen, S. Saito, Z. Wang, O. Maury, C. Wu, J. Hodgins, Y. Zheng, and G. Nam, "Ct2hair: High-fidelity 3d hair modeling using computed tomography," ACM Transactions on Graphics (TOG), vol. 42, no. 4, pp. 1–13, 2023.

[4] Z. Tan, M. Chai, D. Chen, J. Liao, Q. Chu, L. Yuan, S. Tulyakov, and N. Yu, "Michigan: multi-input-conditioned hair image generation for portrait editing," arXiv preprint arXiv:2010.16417, 2020.

[5] P. Zhu, R. Abdal, J. Femiani, and P. Wonka, "Barbershop: Gan-based image compositing using segmentation masks," arXiv preprint arXiv:2106.01505, 2021.

[6] T. Kim, C. Chung, Y. Kim, S. Park, K. Kim, and J. Choo, "Style your hair: Latent optimization for pose-invariant hairstyle transfer via local-style-aware hair alignment," in European Conference on Computer Vision. Springer, 2022, pp. 188–203.

[7] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 4401–4410.

[8] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 8110–8119.

[9] K. Ward, F. Bertails, T.-Y. Kim, S. R. Marschner, M.-P. Cani, and M. C. Lin, "A survey on hair modeling: Styling, simulation, and rendering," IEEE transactions on visualization and computer graphics, vol. 13, no. 2, pp. 213–234, 2007.

[10] Y. Bao and Y. Qi, "A survey of image-based techniques for hair modeling," IEEE Access, vol. 6, pp. 18 670–18 684, 2018.

[11] L. Hu, C. Ma, L. Luo, and H. Li, "Single-view hair modeling using a hairstyle database," ACM Transactions on Graphics (ToG), vol. 34, no. 4, pp. 1–9, 2015.

[12] M. Chai, T. Shao, H. Wu, Y. Weng, and K. Zhou, "Autohair: Fully automatic hair modeling from a single image," ACM Transactions on Graphics, vol. 35, no. 4, 2016.

[13] S. Saito, L. Hu, C. Ma, H. Ibayashi, L. Luo, and H. Li, "3d hair synthesis using volumetric variational autoencoders," ACM Transactions on Graphics (TOG), vol. 37, no. 6, pp. 1–12, 2018.

[14] K. Wu, Y. Ye, L. Yang, H. Fu, K. Zhou, and Y. Zheng, "Neuralhdhair: Automatic high-fidelity hair modeling from a single image using implicit neural representations," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 1526–1535.

[15] M. Zhang, M. Chai, H. Wu, H. Yang, and K. Zhou, "A data-driven approach to four-view image-based hair modeling." ACM Trans. Graph., vol. 36, no. 4, pp. 156–1, 2017.

[16] Z. Kuang, Y. Chen, H. Fu, K. Zhou, and Y. Zheng, "Deepmvshair: Deep hair modeling from sparse views," in SIGGRAPH Asia 2022 Conference Papers, 2022, pp. 1–8.

[17] L. Hu, C. Ma, L. Luo, L.-Y. Wei, and H. Li, "Capturing braided hairstyles," ACM Transactions on Graphics (TOG), vol. 33, no. 6, pp. 1–9, 2014.

[18] M. Zhang, P. Wu, H. Wu, Y. Weng, Y. Zheng, and K. Zhou, "Modeling hair from an rgb-d camera," ACM Transactions on Graphics (TOG), vol. 37, no. 6, pp. 1–10, 2018.

[19] Y. Zhou, M. Chai, A. Pepe, M. Gross, and T. Beeler, "Groomgen: A high-quality generative hair model using hierarchical latent representations," ACM Transactions on Graphics (TOG), vol. 42, no. 6, pp. 1–16, 2023.

[20] V. Sklyarova, E. Zakharov, O. Hilliges, M. J. Black, and J. Thies, "Text-conditioned generative model of 3d strand-based human hairstyles," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 4703–4712.

[21] R. Saha, B. Duke, F. Shkurti, G. W. Taylor, and P. Aarabi, "Loho: Latent optimization of hairstyles via orthogonalization," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1984–1993.

[22] F. Cordier, H. Seo, and N. Magnenat-Thalmann, "Made-to-measure technologies for an online clothing store," IEEE

Computer graphics and applications, vol. 23, no. 1, pp. 38–48, 2003.

[23] C. C. Wang, Y. Wang, and M. M. Yuen, "Design automation for customized apparel products," Computer-aided design, vol. 37, no. 7, pp. 675–691, 2005.

[24] C. C. Wang, K.-C. Hui, and K.-M. Tong, "Volume parameterization for design automation of customized free-form products," IEEE transactions on automation science and engineering, vol. 4, no. 1, pp. 11–21, 2007.

[25] Y. Meng, C. C. Wang, and X. Jin, "Flexible shape control for automatic resizing of apparel products," Computer-aided design, vol. 44, no. 1, pp. 68–76, 2012.

[26] R. Brouet, A. Sheffer, L. Boissieux, and M.-P. Cani, "Design preserving garment transfer," ACM Transactions on Graphics, vol. 31, no. 4, pp. Article–No, 2012.

[27] T. Y. Wang, D. Ceylan, J. Popovic, and N. J. Mitra, "Learning a shared shape space for multimodal garment design," arXiv preprint arXiv:1806.11335, 2018.

[28] P. Guan, L. Reiss, D. A. Hirshberg, A. Weiss, and M. J. Black, "Drape: Dressing any person," ACM Transactions on Graphics (ToG), vol. 31, no. 4, pp. 1–10, 2012.

[29] I. Santesteban, M. A. Otaduy, and D. Casas, "Learning-based animation of clothing for virtual try-on," in Computer Graphics Forum, vol. 38, no. 2. Wiley Online Library, 2019, pp. 355–366.

[30] G. Pons-Moll, S. Pujades, S. Hu, and M. J. Black, "Clothcap: Seamless 4d clothing capture and retargeting," ACM Transactions on Graphics (ToG), vol. 36, no. 4, pp. 1–15, 2017.

[31] Z. Lahner, D. Cremers, and T. Tung, "Deepwrinkles: Accurate and realistic clothing modeling," in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 667–684.

[32] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: a skinned multi-person linear model," ACM Transactions on Graphics (TOG), vol. 34, no. 6, pp. 1–16, 2015.

[33] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers," Foundations and Trends® in Machine learning, vol. 3, no. 1, pp. 1–122, 2011.

[34] M. Chai, C. Zheng, and K. Zhou, "A reduced model for interactive hairs," ACM Transactions on Graphics (TOG), vol. 33, no. 4, pp. 1–11, 2014.

[35] Q. Lyu, M. Chai, X. Chen, and K. Zhou, "Real-time hair simulation with neural interpolation," IEEE Transactions on Visualization and Computer Graphics, vol. 28, no. 4, pp. 1894–1905, 2020.

[36] J. Hsu, T. Wang, Z. Pan, X. Gao, C. Yuksel, and K. Wu, "Real-time physically guided hair interpolation," ACM Transactions on Graphics (TOG), vol. 43, no. 4, pp. 1–11, 2024.

[37] L. Wang, Y. Yu, K. Zhou, and B. Guo, "Example-based hair geometry synthesis," in ACM SIGGRAPH 2009 papers, 2009, pp. 1–9.

[38] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle, "Multiresolution analysis of arbitrary meshes," in Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, 1995, pp. 173–182.

[39] J. Teran, E. Sifakis, G. Irving, and R. Fedkiw, "Robust quasistatic finite elements and flesh simulation," in Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation, 2005, pp. 181–190.

[40] J. Montes, B. Thomaszewski, S. Mudur, and T. Popa, "Computational design of skintight clothing," ACM Transactions on Graphics (TOG), vol. 39, no. 4, pp. 105–1, 2020.

[41] M. Loper, "Chumpy autodifferentiation library," 2014. [Online]. Available: https://github.com/mattloper/chumpy

[42] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," Mathematical Programming Computation, vol. 12, no. 4, pp. 637–672, 2020. [Online]. Available: https://doi.org/10.1007/s12532-020-00179-2

[43] D. M. Mount, "Ann: A library for approximate nearest neighbor searching," http://www.cs.umd.edu/~mount/ANN/, 2010.

[44] A. Jacobson, D. Panozzo, C. Schüller, O. Diamanti, Q. Zhou, N. Pietroni et al., "libigl: A simple c++ geometry processing library," Google Scholar, 2013.

[45] R. A. Al-Asqhar, T. Komura, and M. G. Choi, "Relationship descriptors for interactive motion adaptation," in Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2013, pp. 45–53.

[46] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, "Least squares conformal maps for automatic texture atlas generation," in Seminal Graphics Papers: Pushing the Boundaries, Volume 2, 2023, pp. 193–202.

[47] L. Liu, L. Zhang, Y. Xu, C. Gotsman, and S. J. Gortler, "A local/global approach to mesh parameterization," in Computer graphics forum, vol. 27, no. 5. Wiley Online Library, 2008, pp. 1495–1504.

[48] F. Bogo, J. Romero, M. Loper, and M. J. Black, "Faust: Dataset and evaluation for 3d mesh registration," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 3794–3801.

[49] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas, "Functional maps: a flexible representation of maps between shapes," ACM Transactions on Graphics (ToG), vol. 31, no. 4, pp. 1–11, 2012.

[50] G. Daviet, F. Bertails-Descoubes, and L. Boissieux, "A hybrid iterative solver for robustly capturing coulomb friction in hair dynamics," in Proceedings of the 2011 SIGGRAPH Asia Conference, 2011, pp. 1–12.

[51] D. M. Kaufman, R. Tamstorf, B. Smith, J.-M. Aubry, and E. Grinspun, "Adaptive nonlinearity for collisions in complex rod assemblies," ACM Transactions on Graphics (TOG), vol. 33, no. 4, pp. 1–12, 2014.

Lu Yu received his bachelor's degree in Applied Mathematics from Zhejiang University in 2022. Currently, he is working toward a PhD degree at the State Key Lab of CAD&CG, Zhejiang University. His research interests include geometry processing and physics-based simulation.

Zhong Ren received his bachelor's degree in Information Engineering, his master's degree in Mechanical Engineering, and his PhD in Computer Science, all in 2007 from Zhejiang University. He is currently an associate professor at Zhejiang University and a member of the State Key Laboratory of CAD&CG. Before returning to Zhejiang University in 2010, he spent three years at Microsoft Research Asia, where he served as an associate researcher in the Internet Graphics Group. His research interests include real-time rendering and GPU-based photorealistic rendering.

Youyi Zheng is a ZJU100 Young Professor at the State Key Lab of CAD&CG, College of Computer Science, Zhejiang University. He obtained his Ph.D. from the Department of Computer Science and Engineering at Hong Kong University of Science and Technology, and his M.Sc. and B.Sc. degrees in Mathematics, both from Zhejiang University. His research interests include geometric modeling, imaging, and human-computer interaction.

Xiang Chen is an Associate Professor at the State Key Lab of CAD&CG with Zhejiang University. He received his BSc and PhD both from the department of Computer Science at Zhejiang University. His research interests include geometry processing, physics-based simulation, data analysis, and computer-aided design.

Kun Zhou is a Cheung Kong Professor in the Computer Science Department of Zhejiang University and the Director of the State Key Lab of CAD&CG. He received his BS degree and PhD degree in computer science, both from Zhejiang University. After graduation, he spent six years with Microsoft Research Asia and was a lead researcher of the graphics group before moving back to Zhejiang University. He was elected as an IEEE Fellow in 2015 and an ACM Fellow in 2020.