Dynamic Hair Modeling from Monocular Videos using Deep Neural Networks

LINGCHEN YANG and ZEFENG SHI, State Key Lab of CAD&CG, Zhejiang University
YOUYI ZHENG and KUN ZHOU, Zhejiang University and ZJU-FaceUnity Joint Lab of Intelligent Graphics, China

Fig. 1. Given a video sequence captured with a handhold monocular camera (bottom row), our method constructs a dynamic hair model that closely matches hair shapes in individual frames and exhibits coherent motions.

We introduce a deep learning based framework for modeling dynamic hairs from monocular videos, which could be captured by a commodity video camera or downloaded from Internet. The framework mainly consists of two neural networks, i.e., *HairSpatNet* for inferring 3D spatial features of hair geometry from 2D image features, and *HairTempNet* for extracting temporal features of hair motions from video frames. The spatial features are represented as 3D occupancy fields depicting the hair volume shapes and 3D orientation fields indicating the hair growing directions. The temporal features are represented as bidirectional 3D warping fields, describing the forward and backward motions of hair strands cross adjacent frames. Both *HairSpatNet* and *HairTempNet* are trained with synthetic hair data. The spatial and temporal features predicted by the networks are subsequently used for growing hair strands with both spatial and temporal consistency. Experiments demonstrate that our method is capable of constructing plausible dynamic hair models that closely resemble the input video, and compares favorably to previous single-view techniques.

CCS Concepts: • **Computing methodologies → Motion capture**; **Neural networks**; **Animation**.

*Corresponding authors: Youyi Zheng (youyizheng@zju.edu.cn) and Kun Zhou (kunzhou@acm.org).
Authors' addresses: Lingchen Yang; Zefeng Shi, State Key Lab of CAD&CG, Zhejiang University; Youyi Zheng; Kun Zhou, Zhejiang University , ZJU-FaceUnity Joint Lab of Intelligent Graphics, Hangzhou, 310058, China.

0730-0301/2019/11-ART235 $15.00
https://doi.org/10.1145/3355089.3356511

Additional Key Words and Phrases: Dynamic Hair Modeling, Deep Convolutional Neural Networks

ACM Reference Format:
Lingchen Yang, Zefeng Shi, Youyi Zheng, and Kun Zhou. 2019. Dynamic Hair Modeling from Monocular Videos using Deep Neural Networks. *ACM Trans. Graph.* 38, 6, Article 235 (November 2019), 13 pages. https://doi.org/10.1145/3355089.3356511

1 INTRODUCTION

Modeling 3D hairs that closely match the sensing images has long been an important problem in computer graphics. Impressive reconstruction of high-fidelity, *static* hair models has been demonstrated with image-based techniques (e.g., [Chai et al. 2016; Hu et al. 2015; Luo et al. 2013; Paris et al. 2008]). Reconstructing *dynamic* hair models from video sequences, however, is a relatively underexplored problem.

The challenges of dynamic hair reconstruction mainly arise from the high complexity of hair motion and geometry. Dynamic hair often exhibits constantly changing occlusions among hair strands, making it extremely difficult to build reliable temporal correspondences across video frames. Furthermore, the thin features of hair strands may cause severe motion blur in videos, bringing additional troubles to existing feature-tracking algorithms such as optical-flow methods.

The state-of-the-art technique for dynamic hair capture requires multiple cameras and light arrays in a controlled environment [Xu et al. 2014]. It reconstructs static hair geometry at every video frame using a multi-view based method [Luo et al. 2013], and extracts hair motion paths to resolve hair temporal inconsistency. The dynamic

hair reconstruction is then formulated as a global spacetime optimization procedure. While this technique demonstrates impressive reconstruction results, the complex hardware setup (21 GoPro cameras and 6 LED arrays) restricts the modeling process to professional users. The lightweight technique introduced by Chai et al. [2013] is able to construct a dynamic hair model from a monocular video sequence. It constructs a static hair model for a reference video frame, and relies on optical flow methods to estimate a motion field from the video to deform the static model. Due to the instability of tracking hair strands in many situations, the technique is limited to simple hairstyles and modest motions.

The goal of our work is to develop a lightweight dynamic hair modeling method that can construct plausible dynamic hair models from monocular video sequences. We aim at recovering both the spatial and temporal features of dynamic hairs – the constructed hair models not only have the spatial details of hair strands revealed in the input frames, but also exhibit the temporal coherence of hair motions across the video sequence.

To this end, we introduce a deep learning based framework for modeling dynamic hairs from monocular videos. The framework mainly consists of two network structures, i.e., *HairSpatNet* for inferring 3D spatial features of hair geometry from 2D image features, and *HairTempNet* for extracting temporal features of hair motions from video frames. The spatial features are represented as 3D occupancy fields depicting the hair shapes and 3D orientation fields indicating the hair strand directions. The temporal features are represented as bidirectional 3D warping fields, describing the forward and backward motions of hair strands cross adjacent frames. An important benefit of our deep learning based framework is that it avoids the unstable feature tracking which is commonly employed by previous dynamic hair capture techniques (e.g., [Chai et al. 2013]), making it robust to real-world hair motions. Both *HairSpatNet* and *HairTempNet* are trained with synthetic hairstyles in a dataset and synthetic hair motions from hair simulation. The spatial and temporal features predicted by the networks are subsequently used for growing hair strands and spacetime optimization to generate the final dynamic hair models.

We tested our method with video clips of several real-world hairstyles driven by head movements or external forces, which are captured by a commodity video camera or downloaded from Internet. Experimental results show that our method is capable of constructing plausible dynamic hair models that closely resemble the input video, and compares favorably to previous single-view techniques.

In summary, our paper makes the following contributions:

- We introduce the first deep learning based approach for constructing high-quality dynamic hairs from a monocular video sequence;
- We introduce *HairSpatNet* and *HairTempNet* to synthesize the spatial and temporal features from the video streams, which are trained with synthetic hair data;
- We introduce a novel hair-strand-growing algorithm that accounts for the spatial and temporal consistency of hair strands.

2 RELATED WORK

We review the literature over the field of image-based hair modeling. As one of the most vital parts of digital characters, hair modeling has received extensive attention in computer graphics in the past decades. For a seminal introduction of this prolific field, we refer to the survey of [Ward et al. 2007].

Static hair modeling. With the development of image sensing and computer vision techniques, it becomes possible to create high-quality complex hairstyles from captured images. Among these works, the pioneering multi-view based methods [Echevarria et al. 2014; Herrera et al. 2012; Hu et al. 2014a; Jakob et al. 2009; Luo et al. 2013; Nam et al. 2019; Paris et al. 2008] usually require controlled environments, complex hardware setup, and long processing cycles, which make them not accessible to average users. Single-view based methods [Chai et al. 2015, 2016, 2013, 2012; Hu et al. 2015, 2014b, 2017b; Saito et al. 2018] can produce high-fidelity hair reconstructions, but could fall short in creating realistic results at views distant from the input image. To fill in the gap between multi-view based methods and single-view based methods, later work introduces hair modeling methods based on sparse (three or four) views [Zhang et al. 2017], video streams [Liang et al. 2018], or RGBD inputs [Zhang et al. 2018] to generate realistic hair models. While it is possible to apply such static hair modeling techniques to generate a separate hair model for each video frame, ignoring the temporal correspondences between hair strands across frames would result in flickering artifacts in the reconstructed hair models.

Dynamic hair modeling. Dynamic hair modeling is much less explored in the literature compared to static hair modeling. The pioneering work of [Ishikawa et al. 2007] uses motion capture systems to track the motions of a set of guiding hair strands and subsequently interpolates the motions to generate a full dynamic hair model. Follow-up works exploit multi-view correspondences [Luo et al. 2011; Yamaguchi et al. 2009] or physical simulation [Hu et al. 2017a; Zhang et al. 2012] for dynamic hair modeling. However, these methods are either restricted to coarse and limited hair motions, simple hairstyles, or prone to over-smoothed geometry and artifacts of temporal incoherence. Luo et al. [2013] proposed an algorithm to reconstruct a 3D hair surface of high accuracy by utilizing cross-view consistency of hair orientations. The most recent method of [Xu et al. 2014] introduces motion paths to account for temporal coherence and formulates the hair reconstruction as a global spacetime optimization problem. Their method generates compelling results compared with prior arts. However, the requirements of controlled environments and specialized hardware such as calibrated high-speed cameras and light arrays make these method not accessible to average users.

Chai et al. [2013] introduce a lightweight technique for constructing dynamic hair models from single-view video clips. The main idea is to use a single-view static hair modeling technique to create a static hair model for a reference frame of the video, and then use a motion field estimated by optical flow methods to deform the static model to match the other frames. Due to the complexity of hair geometry and motion, optical flow methods often fail to track hair strands in many situations, restricting the technique to simple

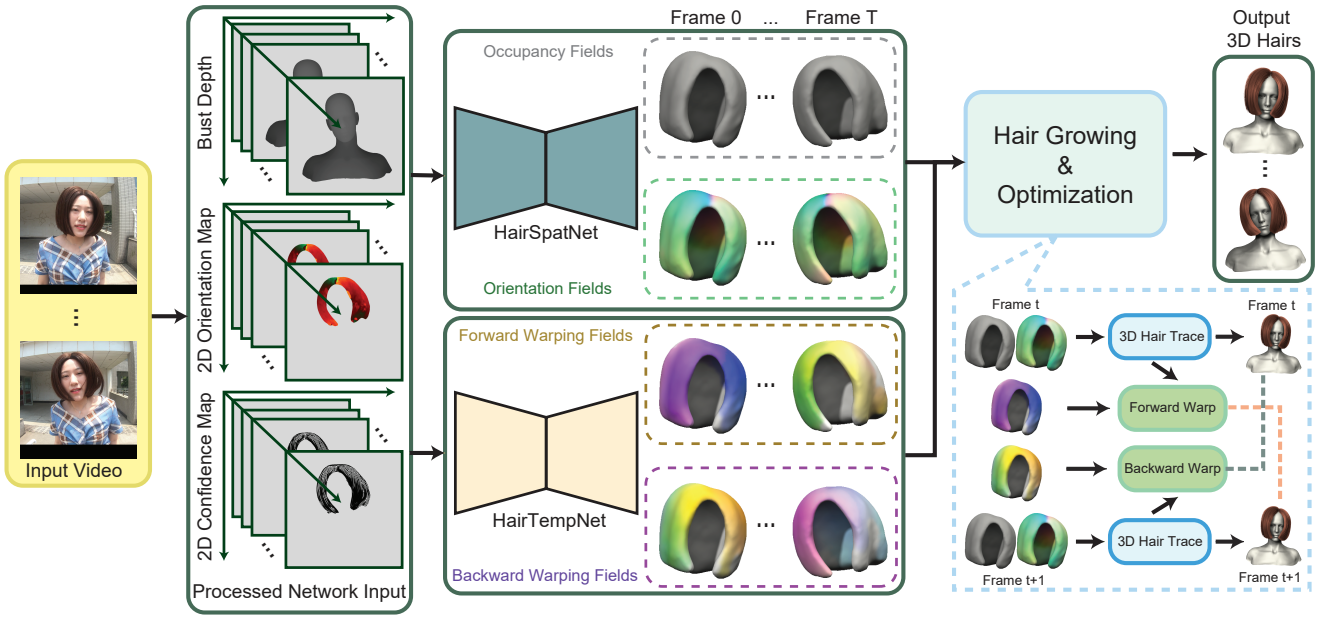


Fig. 2. The pipeline of our method. The input video frames (left) are first preprocessed to generate 2D image features with a bust model (middle left), which are then fed into our *HairSpatNet* and *HairTempNet* to compute 3D hair spatial and temporal features (middle right). The output 3D fields serve as guidance for consistent hair growth and spacetime strands optimization to generate the final dynamic hair models (right).

hairstyles and modest motions. Our method avoids feature tracking by using deep neural networks to calculate 3D bidirectional warping fields between adjacent frames.

Deep learning for hair modeling. With the recent success of deep neural networks in various fields, deep learning based algorithms have also been employed in hair modeling. Early work of Chai et al. [2016] exploits convolutional neural networks for fully automatic hair segmentation and hair orientation field estimation. Hu et al. [2017b] use a deep learning method for hair attribute classification to improve the retrieval performance. Zhou et al. [2018] introduce an encoder-decoder network to directly generate parametric strand point positions, while Saito et al. [2018] leverage a 3D latent space of volumetric hair representations and subsequently train a network to transform dense image features to the latent vector to synthesize 3D hair volume. In the latest work of Zhang et al. [2019], they introduce generative adversarial networks to directly transfer from 2.5D inputs extracted from a single image to a 3D hair orientation field. Our *HairSpatNet* is inspired by their work. Unlike all these approaches, our method is designed for reconstructing both hair geometry and motion across frames rather than reconstructing static hair geometry only in a single frame. We introduce *HairTempNet* that seamlessly integrates with *HairSpatNet* to produce reliable hair temporal correspondences.

3 OVERVIEW

Fig. 2 shows the pipeline of our dynamic hair modeling framework. Our method takes as input a monocular video sequence with hair motions induced by head movements or external forces. The output of our method is a strand-level 3D hair model with vivid hair motions matching the input video. We assume that the face is visible in time so that state-of-the-art head tracking algorithms (e.g., [Cao et al. 2014]) are sufficiently robust. With the video sequence, we first extract the 2D image features (2D hair mask, 2D hair orientation map), and a 3D bust model capturing head motion in the input frames. The extracted 2D image features and the bust model are subsequently fed into our *HairSpatNet* and *HairTempNet* to estimate the 3D hair features (occupancy, 3D hair orientation, and 3D hair warping fields), which are encoded as 3D volumetric tensors (Fig. 2, middle right). Then in a key stage, we grow the hair strands from the 3D occupancy and orientation fields under the guidance of the warping fields to enforce temporal coherence. The generated hair strands in individual frames are further refined with a spacetime optimization.

To train our networks, we use both static and dynamic synthetic hairs. The static hairs are collected from a hair database while the dynamic hairs are derived from a hair simulation system. We train our *HairSpatNet* using both the static and dynamic hair examples while the *HairTempNet* using only the dynamic hair examples. This strategy is based on the key observation that the space of hair motion is rather constrained and in general piece-wise smooth, thus a few representative hairstyles in the simulation system are sufficient to generate basis for various types of hair motions.

4 THE METHOD

In this section, we first explain our image-based hair input and the volumetric output which consists of 3D hair spatial features (occupancy and orientation fields) and 3D hair temporal features (3D forward and backward warping fields). Then, we introduce the two

networks, *HairSpatNet* and *HairTempNet*, to separately predict the hair spatial and temporal details, followed by the descriptions of our novel hair growing algorithm and the global spacetime optimization.

4.1 Data Representation

Our data representation should satisfy two requirements. The first is that we should bridge the input gap between the synthetic data (synthetic images generated from hair databases and the simulation system) and the real data (real video images). The second is that it should be easy for CNNs to handle. Inspired by the previous works, we use 2D orientation map and confidence as input hair information to meet the first requirement. For the latter one, we propose to represent both 3D hair geometry and 3D hair motion as volumetric fields, all of which are defined on evenly-distributed grid of resolution $128 \times 128 \times 96$.

2D Hair Image Feature. We use the method described in [Zhang and Zheng 2019] to generate 2D dense hair orientation and confidence maps as 2D hair features for both synthetic data and real images. The 2D dense hair orientation and confidence maps filter out original hair texture, and thus avoid the appearance discrepancy between synthetic and real hairs, making it credible to train our networks on synthetic data and execute inference on the real data. We also utilize the fitted bust's depth map as a 2.5D input to help our networks generate 3D voxels. All of the input images have the resolution of 1024×1024.

3D Hair Spatial Feature. The occupancy field and orientation field together define the hair geometry. Specifically, the occupancy field is a 3D binary mask where each cell is set to 1 if its center is inside the hair volume or 0 otherwise. The orientation field is a dense 3D direction field where each cell in the hair volume contains a unit vector indicating the local hair growing direction. The orientations for the cells out of the hair volume are undefined and set to zero vectors.

Given a sequence of 3D strand-based hairs of the same style $\{\mathcal{S}^t | t = 0, ..., T-1\}$, we can define occupancy field O^t_{occ} by extracting the outer surface of the hair vertices [Saito et al. 2018] for each frame t. We can also generate the corresponding orientation field O^t_{ori} by iteratively averaging the directions of inner hair segments for each cell in the hair volume [Wang et al. 2009]. Then we diffuse the orientations into entire hair volume as proposed by [Paris et al. 2008]. On the other hand, given the sequence of geometry fields $\{(O^t_{occ}, O^t_{ori}) | t = 0, ..., T-1\}$, we can easily generate 3D hair strands $\{\mathcal{S}'^t | t = 0, ..., T-1\}$ using the hair growing algorithm [Zhang et al. 2018] (S is not necessarily equal to S'). However, the temporal coherence is lost across adjacent frames.

3D Hair Temporal Feature. In video processing, optical flow is commonly used to build the correspondence between two neighboring frames [Chen et al. 2017; Wang et al. 2018a]. By analogy, we build the 3D motion of hair strands as a warping field defined on the occupancy field where each cell in the hair volume contains the local 3D hair motion vector. If the 3D warping field between two consecutive frames is known, we can estimate the hair of the next frame by warping the current frame (see details in the paragraph below). However, only warping hair forwards will accumulate errors in one direction, as shown in Fig. 3. Thus, we adopt the similar concept in [Xu et al. 2014], namely to take both forward and backward directions into consideration and jointly optimize the hairs (Section 4.5).

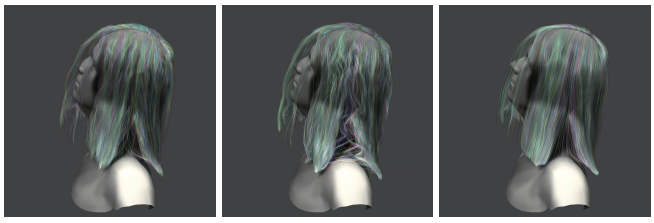
Given $\{\mathcal{S}^t | t = 0, ..., T-1\}$, we denote the corresponding 3D forward warping field as $\mathcal{D}^t_f$ (between $\mathcal{S}^t$ and $\mathcal{S}^{t+1}$) and 3D backward warping field as $\mathcal{D}^t_b$ (between $\mathcal{S}^t$ and $\mathcal{S}^{t-1}$). Due to the noises of both motion direction and magnitude, it is unreliable to estimate the motion for one cell by simply averaging the motions of hair vertices inside it. Inspired by [Wang et al. 2009], we use iteratively re-weighted least squares to reduce the correction by outliers. Different from [Wang et al. 2009] where they only take growing direction as objective, we divide motion into its magnitude and unit direction, and separately regress them. Specifically, let (x, y, z) denote the indices of a cell; $\boldsymbol{dv}$ denote the motion of one hair vertex $\boldsymbol{v}$; $\mathcal{A}(\boldsymbol{v}) = \|\boldsymbol{dv}\|$ denote the magnitude of $\boldsymbol{dv}$ while $\mathcal{T}(\boldsymbol{v}) = \frac{\boldsymbol{dv}}{\|\boldsymbol{dv}\|}$ its unit direction; $C(x, y, z)$ denote the collection of vertices passing through the cell (x, y, z), where vertices with small motions ($\mathcal{A}(\boldsymbol{v}) <$ 1e-6) are removed. The magnitude $\mathcal{A}_c$ and the unit direction $\mathcal{T}_c$ of this cell can be computed iteratively as follows:

$$\mathcal{T}^i_c = \frac{\sum_{\boldsymbol{v} \in C(x,y,z)} \omega^i(\boldsymbol{v}) \mathcal{T}(\boldsymbol{v})}{\| \sum_{\boldsymbol{v} \in C(x,y,z)} \omega^i(\boldsymbol{v}) \mathcal{T}(\boldsymbol{v}) \|}, \tag{1}$$

$$\mathcal{A}^i_c = \sum\nolimits_{\boldsymbol{v} \in C(x,y,z)} \omega^i(\boldsymbol{v}) \mathcal{A}(\boldsymbol{v}) \tag{2}$$

where i denotes the iteration, $\omega^i(\boldsymbol{v}) = (\mathcal{T}(\boldsymbol{v}) \cdot \mathcal{T}^{i-1}_c + 1)/2$ and $\mathcal{T}^0_c = \mathbf{0}$. We use 3 iterations in all our cases. Then we smoothly diffuse the motions into entire hair volume to get our warping fields.

Hair Prediction using Warping Field. The derived warping fields can be used to build the correspondence between the strands of the same root across adjacent frames, which is the key in our final spacetime optimization. Given the 3D warping field $\mathcal{D}^t$ of current hair strands $\mathcal{S}^t$, we can warp each vertex $\boldsymbol{v}$ on one strand $s \in \mathcal{S}^t$ to get corresponding warped strand s' by simply querying which volumertic cell contains $\boldsymbol{v}$: $\boldsymbol{v}' = \boldsymbol{v} + \mathcal{D}^t(\boldsymbol{v})$, $\boldsymbol{v}'$ being the warped vertex. For the root of s, we directly use the rigid head transform to get the warped position, which serves as a hard constraint to fix the roots on the scalp across the whole video. Otherwise, the strand may appear to deviate from the scalp because of the tiny errors of the warping fields. The above method works if all the vertices lie in current hair volume. However, several vertices may



(a) original strands (b) only forward (c) forward + backward

Fig. 3. Comparing our bidirectional hair optimization with only forwards warping hair of initial frame to the current frame.

be out of it when we consecutively warp the strand to build longer correspondence across multiple frames, which will render them the dead vertices, as shown in Fig. 4.

Consequently, we regard it as an energy minimization problem for s' instead. We first get the known warped positions Φ^V for the strand vertices inside the hair volume. Then, we use Φ^V as sparse constraints and regress the vertex positions on s' using least squares, as in [Xu et al. 2014]. More concretely, the least-squares energy combines terms for sparse position alignment E_{pos}, shape preserving E_{lap} and length regularization E_{reg}:

$$E(s') = \omega_{pos}E_{pos}(s', \Phi^v) + \omega_{lap}E_{lap}(s') + \omega_{reg}E_{reg}(s'), \quad (3)$$

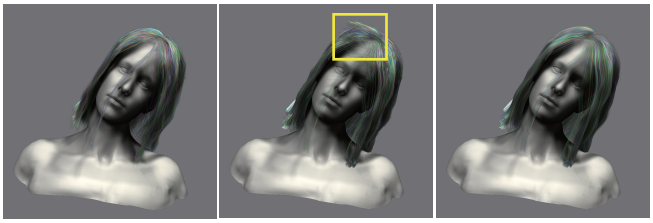
where ω_{pos}, ω_{lap}, ω_{reg} are weights and set to 300, 1 and 1 respectively. The later two terms are used to help extrapolate the dense motions and smooth the result. We refer to [Xu et al. 2014] for more details of the two terms. Note that, since our warping field is a dense 3D field, there is no "Barberpole Illusion", which means one view is enough and directly solving the linear system only once is robust to get s'. This is more stable and faster than in [Xu et al. 2014] where multiple views and more optimization iterations are needed.

Unified Space. In order to generate these paired training data, as in [Zhang and Zheng 2019], we define a bounding box of shape $H \times W \times D$ whose center intersects with that of our bust CG model. Then the hair input information, the bust depth map, and the corresponding fields can be obtained by the method described in [Zhang and Zheng 2019]. Fig. 2 shows the visualization of input information and output fields.

4.2 HairSpatNet

Our *HairSpatNet* is designed to predict 3D geometry frame by frame, namely the input information comes from a single frame. The input $\mathcal{X}^t$ of frame t consists of dense 2D hair orientation, 2D confidence map, and bust depth. The output is the corresponding occupancy field and orientation field.

Architecture. Since our goal is to recover dynamic hairs, both global shape and local details are important. Thus, we cannot directly map the 3D hair geometry into a compact latent space [Saito et al. 2018] and train an image embedding network to predict the latent code for any input image, which will unavoidably lose information during the coding process [Zhang and Zheng 2019]. Our approach is based on [Zhang and Zheng 2019] where they use a residual network to locally learn the hair geometry, which can match the inputs well but needs tedious post processing to refine the global



(a) original strands (b) warp by simple query (c) ours

Fig. 4. Comparing our warping method with simple query. The warping interval is 4 frames.

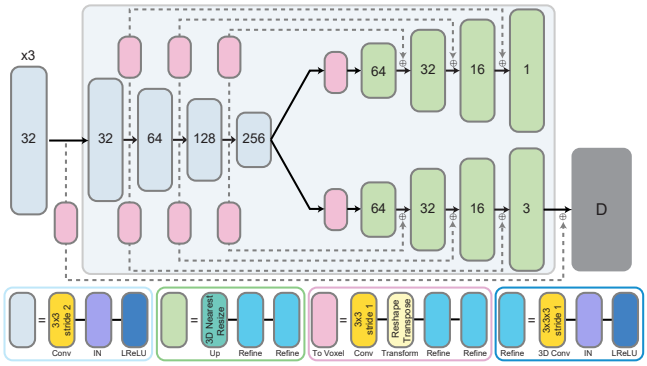


Fig. 5. Architecture of our *HairSpatNet* for an input $\mathcal{X}^t$ with shape of $1024 \times 1024 \times 4$. We first use three downsampling modules to convert $\mathcal{X}^t$ to $128 \times 128 \times 32$ which is then fed into our proposed U-Net. The contracting part has 4 downsampling modules with (32, 64, 128, 256) output channels. The expanding part has 4 upsampling modules with (64, 32, 16, 1) output channels for occupancy field and (64, 32, 16, 3) output channels for orientation field. Since the contracting part extracts 2D features while the expanding part processes 3D features, we use a learnable *toVoxel* module to build feature skip connections. The last upsampling layer does not employ Instance Normalization (IN).

shape. Thus, we utilize the U-Net architecture for this task. Our insights come from two ends. First, the contracting part of the U-Net can map the high-dimensional input into a latent space which is progressively decoded by the expanding part to learn the overall shape. Second, its skip connections can help inject more details into the decoding process. However, current architecture of U-Net is mainly designed for image-to-image tasks [Isola et al. 2017]. It can't be directly applied here. Thus, we modify the original architecture by inserting a learnable *toVoxel* module to build the skip connection between 2D and 3D features. *toVoxel* module first uses a convolution layer to change the channel number, then reshapes and transposes the features to make it compatible with the 3D features, which are followed by two refinement steps based on 3D convolutions. Fig. 5 shows the architecture of our *HairSpatNet*. We use two branches to learn the occupancy and orientation fields.

We also add a discriminator [Goodfellow et al. 2014] to the orientation branch to further enforce fine-grained details. Our discriminator D consists of 6 3D convolution layers followed by LReLU with stride 2 and kernel size $3 \times 3 \times 3$. The output channels are (32, 64, 128, 256, 256, 1). The input to it is $\Psi(\mathcal{X}^t)$ along with either prediction output (negative example) or the ground-truth (positive example), which are first channel-wise concatenated. Ψ denotes 3 downsampling modules along with our *toVoxel* module that converts 2D tensor to 3D feature (see Fig. 5). The final output score is calculated as the average of the feature of the final layer.

Loss Function. For occupancy field , it is a classification problem and we use binary cross entropy (BCE):

$$\mathcal{L}_{bce} = -\mathbb{E}[\gamma O^t_{occ} \odot \log(\hat{O}^t_{occ}) + (1-\gamma)(1-O^t_{occ}) \odot \log(1-\hat{O}^t_{occ})] \quad (4)$$

where $\odot$ denotes element-wise production, $\hat{O}^t_{occ}$ is produced by the *HairSpatNet* and O^t_{occ} is the ground truth, γ is used to balance

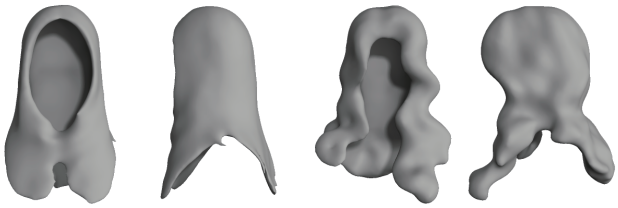


Fig. 6. The hair volume discrepancy between the hairs in the simulation system and the hairs in the database.

the training by putting more penalty for false negatives [Yang et al. 2017], and set to 0.85. For orientation field, we use $L1$ loss:

$$\mathcal{L}_f = \frac{1}{\sum_i O^t_{occ}(i)} ||O^t_{occ} \odot (\hat{O}^t_{ori} - O^t_{ori})||_1, \tag{5}$$

where we only learn the orientations for $O^t_{occ}(i) = 1$. The reason is that the generated occupancy $\hat{O}^t_{occ}$ cannot overlap with O^t_{occ} even when the network converges, namely $\hat{O}^t_{occ} \bigcup O^t_{occ} - \hat{O}^t_{occ} \bigcap O^t_{occ} \neq \emptyset$. In real scenarios, the orientations in these non-overlapping cells should be reasonable which are undefined during training and will be forced to zero vectors if simply minimizing the L1 loss between $\hat{O}^t_{ori}$ and O^t_{ori}. With this loss function, we find the network can automatically fill meaningful orientations in these cells due to the continuousness of convolution operations.

However, the above $L1$ loss will produce over-smoothed prediction results, for which we introduce a discriminator to further enforce fine-grained details. Similar to [Zhang and Zheng 2019], we use the WGAN-GP loss function [Gulrajani et al. 2017]. Different from [Zhang and Zheng 2019], the false example to the discriminator is the regulated prediction result, i.e., $\overline{O}^t_{ori} = O^t_{occ} \odot \hat{O}^t_{ori}$, forcing the discriminator to only focus on the defined orientations. The WGAN-GP loss is defined as:

$$\mathcal{L}_d = \mathbb{E}[D(\mathcal{X}^t, \overline{O}^t_{ori})] - \mathbb{E}[D(\mathcal{X}^t, O^t_{ori})] + \lambda_{gp}\mathcal{L}_{gp}, \tag{6}$$

where D denotes the discriminator, λ_{gp} is a weighting factor for the gradient penalty $\mathcal{L}_{gp}$, $\mathcal{L}_{gp} = \mathbb{E}[(||\nabla_{O^t_*} D(\mathcal{X}^t, O^t_*)||_2 - 1)^2]$, and O^t_* is a data point uniformly sampled along the straight line between $\overline{O}^t_{ori}$ and O^t_{ori}. In addition, we use the discriminator feature matching loss $\mathcal{L}_m$ as it can improve the convergence speed and training stability [Wang et al. 2018b].

Our overall objective function is

$$\mathcal{L}_{geo} = \mathcal{L}_{bce} + \mathcal{L}_f + \mathcal{L}_d + \lambda_m \mathcal{L}_m, \tag{7}$$

where λ_m balances the multiple objectives.

Training Details. For all the experiments, we set $\lambda_{gp} = 10.0$ and $\lambda_m = 0.00001$ based on cross validation and use the Adam solver [Kingma and Ba 2014] with a batch size of 4. All networks are jointly trained from scratch with an initial learning rate of 0.0001 for generator and learning rate of 0.0003 for discriminator.

4.3 HairTempNet

Our *HairTempNet* is designed to predict forward warping field $\hat{\mathcal{D}}^{t-1}_f$ in frame $t-1$ and backward warping field $\hat{\mathcal{D}}^t_b$ in frame t for two consecutive frames $t-1$ and t. Thus the input should consist of multiple frames' information: $\{\mathcal{X}^{t-o} | o = 0, ..., N\}$. We use $N = 2$ and we find there is no significant improvement with more frames (e.g., $N = 5$). Based on the hair input information of adjacent frames, we hope the network can learn to first predict the 2D motion for the local 2D orientation and then extrapolate the corresponding 3D motion.

Architecture. Our *HairTempNet* is also a modified U-Net architecture to learn both global motions and local details. We first downsample the input to extract features, and then use two independent branches to learn the bidirectional warping fields. Our *HairTempNet* has three main difference with respect to our *HairSpatNet*. The first is that all the Instance Normalization layers are removed to avoid normalizing the motion range. The second is that the output channels of two expanding branches are both (64, 32, 16, 3). Thirdly, we do not employ any discriminator, as we find it will introduce many noises.

Loss Function. Below shows the loss function of forward warping field and that of backward warping field is similarly defined. Similar to the regression of the orientation field, we use $L1$ loss and only learn the cells of interest:

$$\mathcal{L}_w = \frac{1}{\sum_i O^t_{occ}(i)} ||O^t_{occ} \odot (\hat{\mathcal{D}}^t_f - \mathcal{D}^t_f)||_1. \tag{8}$$

However, since the average hair volume in the simulation system is smaller than that in static database due to the influence of gravity, the prediction results, only reasonable in a "thinner" hair volume, sometimes are not compatible with *HairSpatNet* trained on both datasets (see Fig. 6). Instead of resorting to the cumbersome post processing, we let the deep network automatically dilate motions of the defined cells. To do this, we add a laplacian loss function:

$$\mathcal{L}_{lap} = \frac{1}{\sum_i 1 - O^t_{occ}(i)} ||(1 - O^t_{occ}) \odot (\frac{\partial^2 \hat{\mathcal{D}}^t_f}{\partial x^2} + \frac{\partial^2 \hat{\mathcal{D}}^t_f}{\partial y^2} + \frac{\partial^2 \hat{\mathcal{D}}^t_f}{\partial z^2})||_2^2. \tag{9}$$

This term can also enforce piece-wise smoothness of warping fields. Thus, our overall objective function is

$$\mathcal{L}_{warp} = \mathcal{L}_w + \lambda_{lap}\mathcal{L}_{lap}, \tag{10}$$

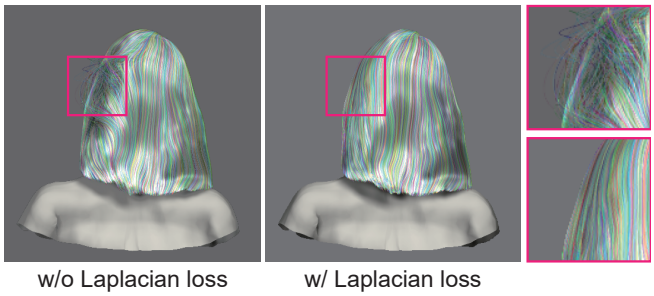


Fig. 7. The effect of the proposed Laplacian loss. The network trained with $\mathcal{L}_{lap}$ (right) generalizes better to the real data, especially towards the invisible part around the back of the head.

Fig. 8. The three hairstyles used in the simulation system to generate the dynamic hair set.

where λ_{lap} balances the multiple objectives. Fig. 7 demonstrates the effectiveness of $\mathcal{L}_{lap}$. We see that with $\mathcal{L}_{lap}$, our *HairTempNet* generates much smoother results towards the unregulated parts (e.g., the invisible back head part).

Training Details. For all the experiments, we set $\lambda_{lap} = 0.001$ based on cross validation and use the Adam solver [Kingma and Ba 2014] with a batch size of 4. All networks are jointly trained from scratch with an initial learning rate of 0.0001. Inspired by [Mayer et al. 2016], we also inject multilevel $L1$ losses for extra resolutions $(1/16, 1/8, ..., 1/2)$ to facilitate the training. The groundtruth for them are nearest-resized from that of the original voxel resolution $(1/1)$.

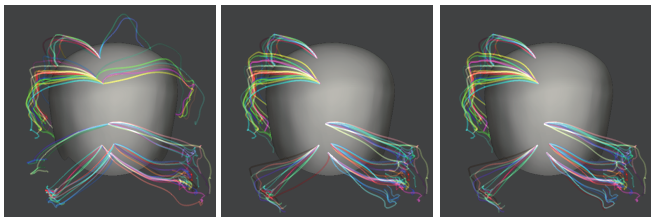
4.4 Training Data

As mentioned earlier, our training data consists of two sets: a dynamic set and a static set. We train our *HairSpatNet* on both sets while our *HairTempNet* on the dynamic set.

Dynamic set. We use a mass-spring model similar to [Selle et al. 2008] to generate sequences of dynamic hairs. The input to our simulation system includes two parts: a rest-state hair and a sequence of external forces to drive the hair movements. We select three high-quality hairstyles (straight, wavy, and long hairs, see Fig. 8) and use typical head movements (including roll, pitch and other random motions), and also random wind forces to generate dynamic sequences of hairs. Based on these movements, our full simulation generate an animation with 1000 frames for each hairstyle. We then augment the data by flipping each hair horizontally and obtain a training set of 6000 frames. We also prepare the test set by changing the forces and regenerating an animation with 100 frames for each hairstyle (300 frames in total).

Static set. In order to cover diverse hair styles, we collect 343 static hairstyles from USC-HairSalon dataset [Hu et al. 2015]. We also augment the data by rotating them around z axis and flipping them horizontally to obtain 3430 different hairs. We then randomly split the entire static set into a training set of 3230 hairs and a test set of 200 hairs.

Finally, we convert these strand-level hairs into volumetric fields using the method described in 4.1 to train our networks. To make our networks more robust against input variations, we also augment the training images by adding Gaussian noise and applying Gaussian blur.



(a) initial (b) iteration #1 (c) iteration #2

Fig. 9. Global hair correction. Initially, for the same root, the traced strands in different frames may appear at either side of the parting line. After 1-2 iterations of the refinement, these strands are corrected to be at one side of the parting line.

4.5 Hair Strand Growing and Optimization

After obtaining all the fields for a video sequence, we can grow the hair strands from geometry fields, and further enforce their temporal coherence under the guidance of warping fields. Finally, we use a modified version of the EM-like optimization procedure [Xu et al. 2014] to get the final hair strands.

Initial strands growing. Given the geometry fields of T frames, we use the hair growing method introduced in [Zhang et al. 2018] to grow hairs for each frame: $\{\mathcal{S}^t | t = 0, ..., T-1\}$. Different from [Zhang et al. 2018], we impose a constraint that each root can only be connected to one strand. Then the set of strands $\Theta_r = \{s_r^0, s_r^1, ..., s_r^{T-1}\}$ belonging to the same root r across T frames should be optimized to exhibit temporally-coherent motion.

Spacetime Optimization. We optimize Θ_r for each root r independently. Let $\mathcal{W}_m^n(s)$ denote the function that warps strand s from frame m to frame n sequentially using the forward warping fields if $m < n$ or backward fields otherwise (see details in fourth section of 4.1). In the E-step, for each root r in frame t, we update its strand s_r^t with the estimated strand $\tilde{s}_r^t$ by averaging the warped strands from neighboring frames:

$$\tilde{s}_r^t = \sum_{j=-\Delta}^{\Delta} \omega^j \mathcal{W}_{t+j}^t(s_r^{t+j}), \tag{11}$$

where the warped strands are uniformly resampled before averaging, ω^j is the gaussian weight and $\mathcal{W}_t^t(s_r^t) = s_r^t$. In the M-step, we solve a linear system to align the shape of $\tilde{s}_r^t$ with the local spatial constraints imposed by current orientation, as in [Xu et al. 2014]. We further deform the frontal strands according to the 2D orientation map, similar to [Hu et al. 2015]. We find 10 EM iterations and $\Delta = \pm 3$ sufficient to converge to a good result in most cases.

Hair Correction. The spacetime optimization assumes that the initial strands in Θ_r have consistent motion. Due to the noise estimation in the orientation fields and the randomness of the hair growing algorithm, this assumption could be violated occasionally. For example, for hair roots around a parting line, they can be connected by hair strands from either side (see Fig. 9). As a consequence, the E step (equation 11) in these regions can be highly unreliable, which will collapse the spacetime optimization, as shown in Fig. 10.

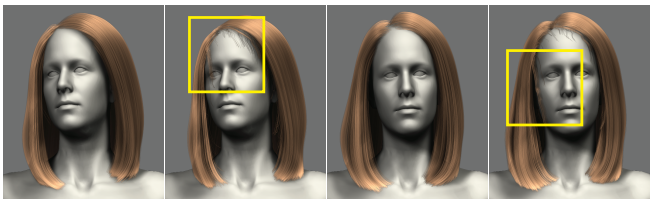


Fig. 10. The effects of the hair correction procedure. The result without the correction has obvious artifacts, as highlighted in the yellow boxes.

We thus propose a correction procedure independently for each root r to enforce its initial strands' temporal coherence before spacetime optimization. We first define the coarse growing direction for a strand $s_r^t \in \Theta_r$ as $\boldsymbol{d}_r^t = \frac{(s_r^t(\kappa) - s_r^t(0))}{||(s_r^t(\kappa) - s_r^t(0))||}$, $s_r^t(\kappa)$ being the κ-th vertex from scalp and $s_r^t(0)$ its root vertex. We use $\kappa = 10$ in all our cases. Then, we define the aligned coarse direction of s_r^t as $Q_t^{-1}\boldsymbol{d}_r^t$, where Q_t^{-1} is the inverse of the corresponding head transform. The correction algorithm alternates between the following two steps until converging or exceeding the maximum iterations (e.g., 10):

- In the first step, we use K-Means to divide Θ_r into two groups C_g and C_b based on their aligned coarse directions. We let C_g include more strands than C_b for the ease of the second step. Then we judge whether the algorithm converges or not. Specifically, we regress two mean directions $\tilde{\boldsymbol{d}}_r^g$ and $\tilde{\boldsymbol{d}}_r^b$ respectively for the aligned coarse directions of C_g and that of C_b and we compute the dot product between them $\beta' = < \tilde{\boldsymbol{d}}_r^g, \tilde{\boldsymbol{d}}_r^b >$. If $|\beta - \beta'| < 0.15$, β being initialized as -1, the algorithm stops. Otherwise, we update β with β'.
- In the second step, we use the strands in C_g to correct the strands in C_b. For a strand $s_r^m \in C_b$, we correct it only if one of its neighbors s_r^{m+1} or s_r^{m-1} belongs to C_g. If $s_r^{m+1} \in C_g$, we use our backward warping field $\mathcal{D}_b^{m+1}$ to obtain the warped strand $\tilde{s}_r^{m+1} = \mathcal{W}_{m+1}^m(s_r^{m+1})$. Then by tracing the cells of current geometry fields around the $\tilde{s}_r^{m+1}$, we regenerate a bunch of strands where we find the one whose coarse direction best match the coarse direction of $\tilde{s}_r^{m+1}$ and replace s_r^m with it. Finally, we remove s_r^m from C_b and add it to C_g. This procedure will repeat many times until $C_b = \emptyset$.

This algorithm has the following advantages. First, it offers better initial hair strands and thus can stabilize and fasten the subsequent spacetime optimization. Second, it can be parallelized for each root.

5 EXPERIMENTS

We have tested our method on real videos with four different types of hairstyles including long hair, short hair, wavy hair, and straight hair. Each hairstyle undergoes different types of motions induced either by head movements, wind blow, or hand interaction. Fig. 16 shows a few representative examples, which demonstrate that our results can faithfully recover both the hair shape and the hair motion closely matching the video input. We refer to the supplementary video for the dynamic results.

The input videos typically contain 100~300 frames which last for 3~10s. All of them are captured with a single monocular camera without any specific controlled setups, and the reconstructed hair models consist of 10K~15K strands (we interpolate to 30-50K for rendering purposes). Note that the quality of the reconstructed hair depends on the input video quality and the resolution of the volumetric fields. In our experiments, we find an input resolution of 1024×1024 and a grid resolution of $128 \times 128 \times 96$ could lead to satisfactory results while imposing a relatively light burden for the training procedure.

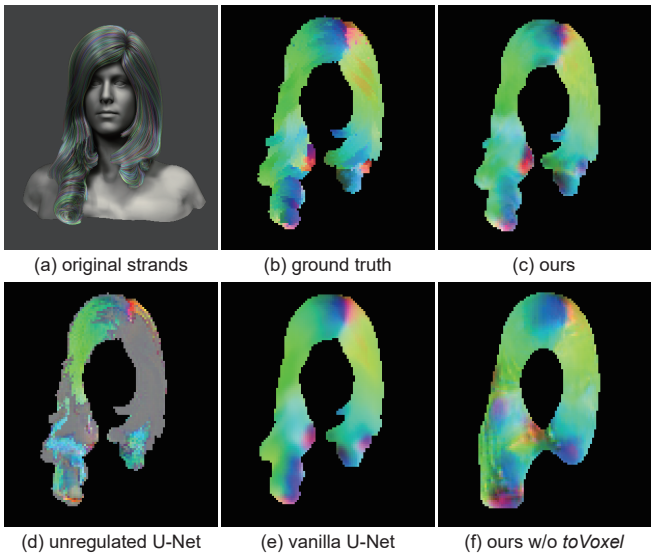


Fig. 11. Comparisons of different approaches for our *HairSpatNet*. We show (a) the original strands, (b) the ground-truth orientation field after projection, (c) ours, (d) unregulated U-Net (ours with regular GAN loss), (e) vanilla U-Net (ours without GAN loss), and (f) ours without *toVoxel* module.

Timings. The training of our *HairSpatNet* and *HairTempNet* takes 10 days and 15 days respectively. For a frame number of 100, it typically takes 15-30 minutes to generate the final dynamic hair models at runtime. The network prediction is instant. Hair growth cross frames takes around 5 minutes and the spacetime optimization typically takes 10-20 minutes to converge (depending on the hair length). All tests are conducted on a PC with an i7-8700 CPU(3.2GHz), 64G main memory, and a GeForce 2080Ti GPU with 11G memory.

5.1 Evaluation

We conduct qualitative and quantitative experiments to evaluate the effectiveness and various algorithmic choices of our *HairSpatNet* and *HairTempNet*.

Evaluation of HairSpatNet. We evaluate our *HairSpatNet* in terms of energy function and the architecture. We first run an ablation study on the proposed energy function for our *HairSpatNet* by comparing it with two alternatives: vanilla U-Net and unregulated U-Net. We refer to vanilla U-Net as our proposed U-Net trained without any GAN loss (i.e., only using $\mathcal{L}_{bce}$ and $\mathcal{L}_f$) while the unregulated U-Net as the U-Net trained with regular GAN loss (the false example is directly fed into the discriminator without the proposed regulation mechanism). For a fair comparison, we keep all the parameters and the network architecture as the same with the exception of the

Table 1. Quantitative comparisons of different approaches for our *HairSpatNet* on the two data sets. We use precision for occupancy field while *L2* error for orientation field.

Method	Static Set		Dynamic Set	
	Precision	L2	Precision	L2
Vanilla U-Net	0.7854	0.0760	0.8852	0.0958
Unregulated U-Net	0.7339	0.4814	0.8434	0.3809
Ours(w/o *toVoxel*)	0.6912	0.1645	0.8206	0.1680
Ours	0.7723	0.1360	0.8675	0.1568

energy function used for training. We conduct the experiments on dynamic and static test sets.

Fig. 11 and Table 1 show the comparisons on the test set. It is evident that the vanilla U-Net produces over-smoothed results although it achieves slightly lower error for reconstruction of orientation field, compared to our proposed loss function.

The unregulated U-Net produces bad results both qualitatively and quantitatively. Due to the cells in the hair volume (defined cells) are extremely sparse with respect to the cells out of the hair volume (undefined cells and their orientations are set to zero vectors in the training set), the discriminator tends to focus more on the undefined cells and forcing them to zero vectors. This in turn collapses the training process and makes the generated orientation fields incompatible with the corresponding occupancy fields (e.g., some orientations in the predicted hair volume are close to zero vectors, as shown in Fig. 11). With the regulation mechanism, there is no gradients back-propagated through these undefined cells, making the training procedure much easier. In addition, the generator can automatically fill meaningful orientations in these undefined cells due to the continuousness of convolution operations.

We also run an ablation study on the proposed *toVoxel* module by comparing it with the network without it while the rest are the same. Fig. 11 and Table 1 show the comparisons. Obviously, our proposed U-Net achieves overwhelming results both in quality and quantity, compared to the network without *toVoxel* module. This is consistent with our insight that the contracting part and the expanding part of the network can only learn the coarse geometry while the *toVoxel* module can inject fine-grained details into the decoding process.

Evaluation of HairTempNet. To evaluate the effectiveness of our *HairTempNet*, we conduct the experiments on the dynamic test set. We test on two forces: wind force and head movement. We compare our generated warping fields with ground truth fields. Table 2 show the quantitative error of the reconstructed motion fields (representative visual results are shown in Fig. 12). In the table, we additionally examine the number of frames N used for input information. We keep all the rest the same except N. It shows that the influence of N is negligible and $N = 2$ is enough to achieve good results. Note that the model trained with larger N fails to quantitatively surpass that with $N = 2$. One possible reason is that N is not large enough to build long-term dependencies (e.g., $N >> 7$). On the other hand, training the model with larger N entails more training corpus and iterations. We leave this exploration to future work.

To further evaluate the influence of the quality of generated fields on the final results, we conduct 4 set of experiments alternating between the generated 3D fields (i.e., occupancy field, orientation field, and warping field) and the ground truth fields. Fig. 12 shows representative qualitative results. In all testing examples, our method is able to faithfully recover high-fidelity hair motions.

Table 2. Quantitative comparisons for different N (number of input frames used for our *HairTempNet*).

Method	L2(forward)	L2(backward)	L2(average)
$N = 7$	0.0331	0.0341	0.0336
$N = 5$	0.0344	0.0348	0.0346
$N = 2(ours)$	0.0308	0.0303	0.0306

5.2 Comparisons

To evaluate the effectiveness of our *HairSpatNet* architecture, we first conduct comparisons with HairGAN [Zhang and Zheng 2019] and the method of [Saito et al. 2018]. For a fair comparison, we use the same processed 2D hair information and final deformation algorithm as in [Zhang and Zheng 2019]. As shown in Fig. 13, three methods generate comparable results, with our method performing slightly better in some hair regions. Due to the lack of occupancy field, HairGAN has out-of-volume problem where the generated hair shape does not match the input mask and requires tedious post-refinement. Their network learns local features while ours learns both global and local features. Subsequently, the overall hair growth trend in our result match the input image slightly better (see the parting region around the top-left corner, Fig. 13 top row). In [Saito et al. 2018], since they use the image embedding network to predict the latent code, the hair details are unavoidably lost due to a series of down sampling operations, e.g., the generated hair layering does not match the input well as shown in the bottom row of Fig. 13.

We also compare our method with two alternative dynamic hair modelling methods. First, we compare our method with a straightforward solution which uses optical flow to track the 2D hair motion and propagate the motion to 3D [Chai et al. 2013]. Fig. 14 shows the corresponding results. It is evident that a simple optical flow-based method could easily fail when the hair strands undergo either fast motion (resulting in severe blur) or occlusion and thus require manual efforts to specify pixel-level correspondences as in [Chai et al. 2013]. On the other hand, such 2D-based approach could easily generate artifacts for invisible strands such as those on the back. In both cases, our method succeeds to generate reliable results.

In a second experiment, we compare our method with the state-of-the-art multi-view based dynamic hair capture method of [Xu et al. 2014]. We take a single-view input video acquired from the authors to generate our results. Fig. 15 shows the results and the corresponding input view. Since the face of the bust model in the video is invisible, we manually align the head model. The results show that our generated hair shape and motion are similar to that of [Xu et al. 2014], but the result of [Xu et al. 2014] contains more shape and motion details. The light curly hair details are lost partially because our volume grid resolution is much lower than theirs ($128 \times 128 \times 96$ in our case, $\sim$ 7mm per voxel, compared to 3mm per voxel in [Xu et al. 2014]). However, their method requires complex environmental setup (e.g., 21 calibrated GoPro cameras) and a much longer processing time while our method only requires a consumer-level

Fig. 12. Validation of our *HairSpatNet* and *HairTempNet* on the synthetic data. Our method generates consistently high-quality results compared to that of ground truth. o./o. strands for occupancy field/orientation field and w. strands for warping fields.

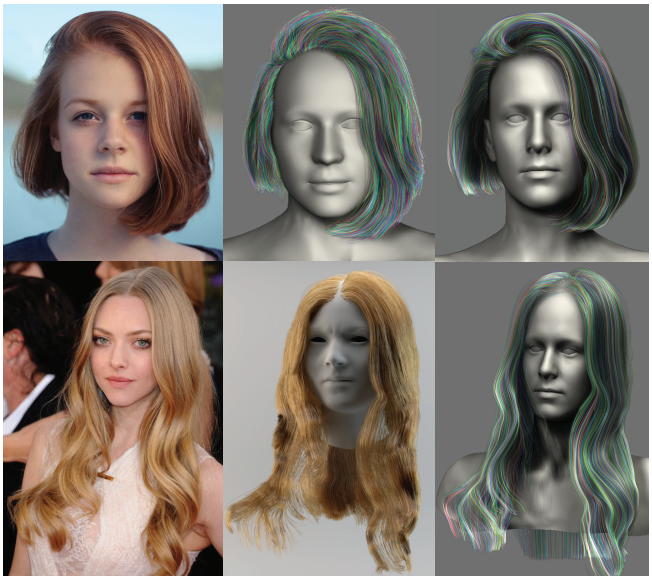


Fig. 13. Comparisons with [Zhang and Zheng 2019] (the first row, original image courtesy of Jason Merritt) and [Saito et al. 2018] (the second row, original image courtesy of Christopher Campbell). For each comparison, from left to right: input image, the result of the previous method and our result.

handhold monocular camera. Please see the supplemental video for the dynamic animation results.

5.3 Limitations and Discussions

Although our method can generate plausible dynamic hair models from real-world videos with different types of hairstyles, the results are still not as good as those generated by state-of-the-art multi-view methods and exhibit artifacts like 'over-smoothing' and 'jittering'. First, compared to [Xu et al. 2014], our method generates comparable results in global hair shape and motion, but lacks small hair details. This is because not only our volume grid resolution is much lower than theirs but also the E step in space-time optimization, which serves as a Gaussian filter, will smooth the results. As seen in Fig. 12(b), even with the ground truth fields, the reconstruction tends to be smooth. Increasing the grid resolution may help, but will significantly slow down the training speed. Second, hair length changing across frames is another noticeable artifact. The occupancy fields of neighboring frames may have a significant discrepancy at the boundaries; consequently the strands near the boundaries will appear to vary in length. We can leverage warp fields to refine the occupancy fields to make them more temporally-coherent before hair growing and optimization. Another way to alleviate this problem is to use temporal convolution or recurrent neural networks (RNN). Third, some of our results also demonstrate spurious hair motions. We sample seeds randomly in the hair volume rather than from the scalp; consequently, the hair strands in individual frames could exhibit slight differences. The orientation fields generated by the *HairSpatNet* in the stationary part could be disturbed by the moving hairs from the other part, which also exacerbates this problem. Although we execute space-time optimization

Fig. 14. A direct optical flow-based method [Chai et al. 2013] (with no user intervention) could easily lead to undesirable results (middle), especially when large motion or occlusion occurs. Our result is shown on the right.

Fig. 15. Our method achieves compatible results with the multi-view method of [Xu et al. 2014] (middle). Their method retains more accurate details of the hair geometry thanks to the multi-view acquisition.

afterwards, the spurious motions are alleviated but still exist. Fourth, our method would fail to faithfully recover stray hair strands whose motions exhibit sudden changes from the nearby frames. This could happen when different layers of hairs cross over each other during large-scale motions. An example is shown in Fig. 16, the middle column of the third row. Moreover, if the hair motion is highly chaotic, our method will fail and produce over-smoothed results. Fifth, the dynamic hair models constructed by our method only match the input video sequences, and may not resemble the ground-truth hair at novel views. Finally, our method relies on robust head fitting algorithms. If this step fails, manual efforts will be required.

6 CONCLUSION

Reconstructing dynamic hair models from video sequences remains challenging. We introduced the first deep learning based method for modeling dynamic hair from a monocular video sequence, which could be captured by a handhold camera or downloaded from Internet. We have demonstrated the efficacy of our method on several real-world hairstyles driven by head movements or external forces. The constructed dynamic hair models closely resemble the input video frames. We expect that our work could inspire further research on this difficult problem.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their constructive comments, Menglei Chai, Lvdi Wang and Xin Tong for help on comparisons. This work is partially supported by the National Key Research & Development Program of China (2018YFE0100900), NSF China (No. U1609215), and the Fundamental Research Funds for the Central Universities.

REFERENCES

Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. 2014. Face alignment by explicit shape regression. *International Journal of Computer Vision* 107, 2 (2014), 177–190.

Menglei Chai, Linjie Luo, Kalyan Sunkavalli, Nathan Carr, Sunil Hadap, and Kun Zhou. 2015. High-quality hair modeling from a single portrait photo. *ACM Trans. Graph.* 34, 6 (2015), 204:1–204:10.

Menglei Chai, Tianjia Shao, Hongzhi Wu, Yanlin Weng, and Kun Zhou. 2016. AutoHair: Fully Automatic Hair Modeling from a Single Image. *ACM Trans. Graph.* 35, 4 (2016), 116:1–116:12.

Menglei Chai, Lvdi Wang, Yanlin Weng, Xiaogang Jin, and Kun Zhou. 2013. Dynamic Hair Manipulation in Images and Videos. *ACM Trans. Graph.* 32, 4 (2013), 75:1–75:8.

Menglei Chai, Lvdi Wang, Yanlin Weng, Yizhou Yu, Baining Guo, and Kun Zhou. 2012. Single-view Hair Modeling for Portrait Manipulation. *ACM Trans. Graph.* 31, 4 (2012), 116:1–116:8.

Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. 2017. Coherent online video style transfer. In *Proceedings of the IEEE International Conference on Computer Vision*. 1105–1114.

Jose I Echevarria, Derek Bradley, Diego Gutierrez, and Thabo Beeler. 2014. Capturing and stylizing hair for 3D fabrication. *ACM Trans. Graph.* 33, 4 (2014), 125:1–125:11.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*. 5767–5777.

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

Liwen Hu, Derek Bradley, Hao Li, and Thabo Beeler. 2017a. Simulation-Ready Hair Capture. *Computer Graphics Forum* 36, 2 (2017), 281–294.

Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. 2014a. Robust hair capture using simulated examples. *ACM Trans. Graph.* 33, 4 (2014), 126:1–126:10.

Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. 2015. Single-view hair modeling using a hairstyle database. *ACM Trans. Graph.* 34, 4 (2015), 125:1–125:9.

Liwen Hu, Chongyang Ma, Linjie Luo, Li-Yi Wei, and Hao Li. 2014b. Capturing Braided Hairstyles. *ACM Trans. Graph.* 33, 6 (2014), 225:1–225:9.

Liwen Hu, Shunsuke Saito, Lingyu Wei, Koki Nagano, Jaewoo Seo, Jens Fursund, Iman Sadeghi, Carrie Sun, Yen-Chun Chen, and Hao Li. 2017b. Avatar digitization from a single image for real-time rendering. *ACM Trans. Graph.* 36, 6 (2017), 195:1–195:14.

Takahito Ishikawa, Yosuke Kazama, Eiji Sugisaki, and Shigeo Morishima. 2007. Hair Motion Reconstruction Using Motion Capture System. In *ACM SIGGRAPH 2007 Posters (SIGGRAPH '07)*. Article 78.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*. 5967–5976.

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

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Shu Liang, Xiufeng Huang, Xianyu Meng, Kunyao Chen, Linda G. Shapiro, and Ira Kemelmacher-Shlizerman. 2018. Video to Fully Automatic 3D Hair Model. *ACM Trans. Graph.* 37, 6 (2018), 206:1–206:14.

Linjie Luo, Hao Li, and Szymon Rusinkiewicz. 2013. Structure-aware hair capture. *ACM Trans. Graph.* 32, 4 (2013), 76:1–76:12.

Linjie Luo, Hao Li, Thibaut Weise, Sylvain Paris, Mark Pauly, and Szymon Rusinkiewicz. 2011. *Dynamic Hair Capture*. Technical Report TR-907-11. Princeton University.

L. Luo, C. Zhang, Z. Zhang, and S. Rusinkiewicz. 2013. Wide-Baseline Hair Capture Using Strand-Based Refinement. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*. 265–272.

Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. 2016. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4040–4048.

Giljoo Nam, Chenglei Wu, Min H. Kim, and Yaser Sheikh. 2019. Strand-Accurate Multi-View Hair Capture. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 155–164.

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

Shunsuke Saito, Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. 2018. 3D Hair Synthesis Using Volumetric Variational Autoencoders. *ACM Trans. Graph.* 37, 6 (2018), 208:1–208:12.

Andrew Selle, Michael Lentine, and Ronald Fedkiw. 2008. A mass spring model for hair simulation. In *ACM Transactions on Graphics (TOG)*, Vol. 27. 64:1–64:11.

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

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018a. Video-to-Video Synthesis. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). 1144–1156.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018b. High-Resolution Image Synthesis and Semantic Manipulation With Conditional GANs. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kelly Ward, Florence Bertails, Tae-Yong Kim, Stephen R Marschner, Marie-Paule Cani, and Ming C Lin. 2007. A survey on hair modeling: Styling, simulation, and rendering. *IEEE Trans. Vis. Comp. Graph.* 13, 2 (2007), 213–234.

Zexiang Xu, Hsiang-Tao Wu, Lvdi Wang, Changxi Zheng, Xin Tong, and Yue Qi. 2014. Dynamic Hair Capture Using Spacetime Optimization. *ACM Trans. Graph.* 33, 6 (2014), 224:1–224:11.

Fig. 16. Dynamic hair modeling results on various hairstyles. The full motion results are shown in the supplementary video. Note that we use a single bust model throughout our experiments, hence the face of the bust model may not fully match that of the video.

Tatsuhisa Yamaguchi, Bennett Wilburn, and Eyal Ofek. 2009. Video-based modeling of dynamic hair. In *Pacific-Rim Symposium on Image and Video Technology*. Springer, 585–596.

Bo Yang, Hongkai Wen, Sen Wang, Ronald Clark, Andrew Markham, and Niki Trigoni. 2017. 3d object reconstruction from a single depth view with adversarial learning. In *Proceedings of the IEEE International Conference on Computer Vision*. 679–688.

Meng Zhang, Menglei Chai, Hongzhi Wu, Hao Yang, and Kun Zhou. 2017. A Data-driven Approach to Four-view Image-based Hair Modeling. *ACM Trans. Graph.* 36, 4 (2017), 156:1–156:11.

Meng Zhang, Pan Wu, Hongzhi Wu, Yanlin Weng, Youyi Zheng, and Kun Zhou. 2018. Modeling Hair from an RGB-D Camera. *ACM Trans. Graph.* 37, 6 (2018), 205:1–205:10.

Meng Zhang and Youyi Zheng. 2019. Hair-GANs: Recovering 3D Hair Structure from a Single Image. *The Visual Informatics* (2019), to appear.

Qing Zhang, Jing Tong, Huamin Wang, Zhigeng Pan, and Ruigang Yang. 2012. Simulation Guided Hair Dynamics Modeling from Video. *Comput. Graph. Forum* 31 (2012), 2003–2010.

Yi Zhou, Liwen Hu, Jun Xing, Weikai Chen, Han-Wei Kung, Xin Tong, and Hao Li. 2018. HairNet: Single-View Hair Reconstruction using Convolutional Neural Networks. In *Proceedings of the European Conference on Computer Vision*. 235–251.