Avatar Digitization From a Single Image

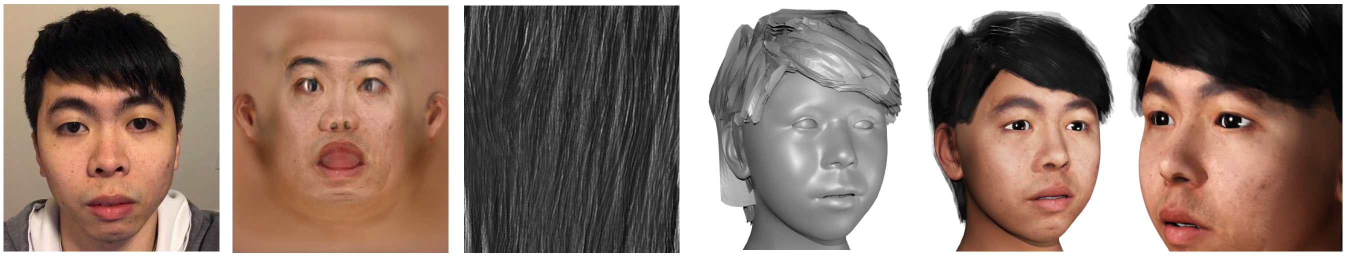

Figure 1: *We introduce an automatic approach for modeling 3D avatars with hair from a single input image. Our approach can infer complete and textured meshes for faces and volumetric strips for hair from a partially visible subject. Our avatar reconstruction includes eyes, teeth, and tongue models and is fully rigged using a combination of blendshapes and joint-based skeleton.*

Abstract

We present a fully automatic framework for creating a complete 3D avatar from a single unconstrained image. We digitize the entire model using a textured mesh representation for the head and volumetric strips with transparency for the hair. Our digitized models can be easily integrated into existing game engines and readily provide animation-friendly blendshapes and joint-based rigs. The proposed system integrates state-of-the-art advances in facial shape modeling, appearance inference, and a new pipeline for single-view hair generation based on hairstyle retrieval from a massive database, followed by a strand-to-hair-strip conversion method. We also introduce a novel algorithm for realistic hair texture synthesis for the strips based on feature correlation analysis using a deep neural network. Our generated models are visually comparable to state-of-the-art game characters, as well as avatar generation techniques based on multiple input images. We demonstrate the effectiveness of our approach on a variety of images taken in the wild, and show that compelling avatars can be generated by anyone without effort.

Keywords: dynamic avatar, face, hair, digitization, modeling, rigging, texture synthesis, data-driven, deep learning, neural network

Concepts: •**Computing methodologies** → *Mesh geometry models;*

1 Introduction

Alongside entertainment applications and the ability to immerse in a captivating alternate universe, the democratization of virtual reality (VR) has the potential to revolutionize 3D face-to-face communication and social interactions through compelling digital embodiments of ourselves, as demonstrated lately with the help of VR head mounted displays with facial sensing capabilities [Li et al. 2015; Thies et al. 2016b; Olszewski et al. 2016] or voice-driven technology demonstrated at Oculus Connect 3. Faithfully personalized 3D avatars would not only facilitate natural telepresence between participants in virtual worlds, but also enable new gaming experiences with individualized characters.

Recent progress in data-driven methods and deep learning research have catalyzed the development of high-quality 3D face modeling techniques from a single image [Thies et al. 2016a; Saito et al. 2016b] and even realistic strand-level hair models can now be generated from an image with minimal human input [Hu et al. 2015; Chai et al. 2016]. Despite efforts in real-time simulation [Chai et al. 2014], strand-based representations are still very difficult to integrate into game environments due to their rendering and simulation complexity. Furthermore, strands are not efficient for hairstyles with highly stochastic structures (messy, curly, afro-hair, etc.). Cao et al. [2016] have recently introduced a system that uses a highly versatile image-based mesh representation, but the volumetric structure of hair is not captured, and they require multiple photographs, as well as some manual intervention as input. Despite substantial advances in making avatar creation as easy as possible, the barriers to entry are still too high for commodity user adoption.

In this paper, we present the first framework that automatically generates a complete high-quality 3D avatar from a single unconstrained image using textured meshes for faces and volumetric strips for hair. By eliminating the need of multiple photographs and controlled capture, we can easily digitize our favorite celebrities or iconic figures from any Internet picture. Our digitized models are fully rigged with intuitive animation controls based on blendshapes and joint-skeletons, and can be easily integrated into existing game engines. In addition to unknown illumination conditions, reconstructing from images in the wild is further challenged by largely non-visible regions and occlusions. How does a face region look like when it is blocked by hair, and how can we predict the appearance of the back of a head if only the front is visible?

We address these challenges by carefully integrating multiple cutting edge techniques into a comprehensive facial digitization pipeline, and introduce a new single-view hair modeling algorithm for generating high-quality textured hair strips. For the face, we first fit a linear face model to a pre-segmented input image using a combination of landmark detection [Kazemi and Sullivan 2014] and a dense analysis-by-synthesis approach [Thies et al. 2016a] enhanced with visibility constraints. A deep learning-based texture inference method [Saito et al. 2016b] then produces a high-resolution and texture map of the entire head, even from low-resolution input photographs. Given the segmented hair region and an orientation field analysis, we compute a descriptor to retrieve the closest

SIGGRAPH 2017 Technical Papers, MONTH, DAY, and YEAR
ISBN: THIS-IS-A-SAMPLE-ISBN. . . $15.00
DOI: http://doi.acm.org/THIS/IS/A/SAMPLE/DOI

matching hairstyle from a large database of hair models similar to the method of Chai et al. [2016]. The closest matching hairstyle is then deformed to fit the segmented hair image and its orientation field. Each strand is converted into a set of strips that cover the hair volume. To generate photo-realistic and non-repeating textures of each hair strip, we present a new synthesis algorithm based on feature correlation analysis using deep neural networks. For visually pleasing animations, especially for long hairs, we also rig our hair model to the head skeleton using inverse distance skinning [Jacobson].

We demonstrate the effectiveness of our approach on a wide range of subjects with different hairstyles, and also show compelling animated results of dynamic avatars that are automatically generated from a single image. The output quality of our framework is comparable to state-of-the-art game characters, and superior to cutting-edge avatar modeling systems that are based on multiple input photographs [Ichim et al. 2015; Cao et al. 2016].

Contributions:

- We present the first fully automatic framework for complete 3D avatar modeling and rigging, that includes hair capture, from a single unconstrained image.
- Our facial digitization pipeline integrates the latest advances in facial segmentation, shape modeling, and appearance inference.
- We develop a new data-driven single-view hair digitization pipeline for generating volumetric hair structures, based on highly efficient and versatile hair strips.
- We introduce a deep learning-based texture synthesizing algorithm for producing hair strip textures that are photorealistic, non-repeating, and individualized to the input.

2 Background

Facial Modeling and Capture. Over the past two decades, a great amount of researches have been dedicated to the modeling and animation of digital faces. We refer to [Parke and Waters 2008] for a comprehensive introduction and overview. Though artist-friendly digital modeling tools have significantly evolved over the years, 3D scanning and performance capture technologies provide an attractive way to scale content creation and improve realism through accurate measurements from the physical world. While expensive and difficult to deploy, sophisticated 3D facial capture systems [Debevec et al. 2000; Ma et al. 2007; Beeler et al. 2010; Bradley et al. 2010; Beeler et al. 2011; Ghosh et al. 2011] are widely adopted in high-end production and have proven to be a critical component for creating photoreal digital actors. Different rigging techniques such as joint-based skeletons, blendshapes [Li et al. 2010; von der Pahlen et al. 2014], or muscle-based systems [Terzopoulos and Waters 1990; Sifakis et al. 2005] have been introduced to ensure intuitive control in facial animation and high-fidelity retargeting for performance capture. Dedicated systems for capture, rigging, and animation have also emerged for the treatment of secondary components such as eyes [Miller and Pinskiy 2009; Bérard et al. 2016], lips [Garrido et al. 2016b], and teeth [Wu et al. 2016]. Despite high-fidelity output, these capture and modeling systems are too complex for mainstream adoption.

The PCA-based linear face models of [Blanz and Vetter 1999] have laid the foundations for the modern treatment of image-based 3D face modeling, with extensions to multi-view stereo [Blake et al. 2007], large-scale internet pictures [Kemelmacher-Shlizerman 2013], massive 3D scan datasets [Booth et al. 2016], and the use of shading cues [Kemelmacher-Shlizerman and Basri 2011]. Blanz and Vetter have demonstrated in their original work that compelling facial shapes and appearances with consistent parameterization can be extracted reliably from a single input image. To handle facial expressions, multi-linear face models have been used, that are based on PCA [Vlasic et al. 2005] and FACS-based blendshapes [Cao et al. 2014b]. The low dimensionality and effectiveness in representing faces have made linear face models particularly suitable for instant 3D face modeling and robust facial performance capture in monocular settings using depth sensors [Weise et al. 2009; Weise et al. 2011; Bouaziz et al. 2013; Li et al. 2013; Hsieh et al. 2015], as well as RGB video [Garrido et al. 2013; Shi et al. 2014; Cao et al. 2014a; Garrido et al. 2016a; Thies et al. 2016a; Saito et al. 2016a]. When modeling a 3D face automatically from an image, sparse 2D facial landmarks [Cootes et al. 2001; Cristinacce and Cootes 2008; Saragih et al. 2011; Xiong and De la Torre 2013] are typically used for robust initialization during fitting. State-of-the-art landmark detection methods achieve impressive efficiency by using explicit shape regressions [Cao et al. 2013; Ren et al. 2014; Kazemi and Sullivan 2014].

While linear models can estimate entire head models from a single view, the resulting textures are typically crude approximations of the subject, especially in the presence of details such as facial hear, complex skin tones, and wrinkles. In order to ensure likeness to the captured subject, existing 3D avatar creation systems often avoid the use of a purely linear appearance model, but use acquisitions from multiple views to build a more accurate texture map. Ichim et al. [Ichim et al. 2015] introduced a comprehensive pipeline for video-based avatar reconstruction in uncontrolled environments. They first produce a dense point cloud using multi-view stereo and then estimate a 3D face model using non-rigid registration. An integrated albedo texture map is then extracted using a combination of Poisson blending and light factorization via spherical harmonics. Their method is limited to a controlled acquisition procedure based on a semi-circular sweep of a hand-held sensor, and hair modeling is omitted. Chai et al. [Chai et al. 2015] presented a single-view system for high-quality 2.5D depth map reconstruction of a both faces and hair, using structural hair priors, silhouette, and shading cues. However, their technique is not suitable for avatars, as a full head cannot be produced nor animated. More recently, Cao et al. [2016] developed an end-to-end avatar creation system that can produce compelling face and hair models based on an image-based mesh representation. While their system can handle very large variations of hairstyles and also produce high-quality facial animations with fine-scale details, they require up to 32 input images and some manual guidance in segmentation and labeling.

Instead of a controlled capture procedure with multiple photographs, we propose a system that is fully automatic and only needs a single image as input. We also introduce a new hair digitization framework based on the highly efficient and flexible textured strips representation, often adopted in state-of-the-art games. Hair strips are more efficient for simulation and rendering than hair strands, but also achieve believable volumetric structures as opposed to the textured mesh representation used by [Cao et al. 2016].

Hair Modeling and Capture. An essential but intricate component for life-like avatars and CG characters is hair. In studio settings, human hair is traditionally modeled, simulated, and rendered using sophisticated design tools [Kim and Neumann 2002; Yuksel et al. 2009; Choe and Ko 2005; Weng et al. 2013]. We refer to the survey of Ward et al. [Ward et al. 2006] for an extensive overview. In analogy to faces, 3D hair capture techniques have been introduced to directly digitize hair from the physical world. High-fidelity acquisition systems typically involve controlled recording sessions, manual assistance, and complex hardware equipments,

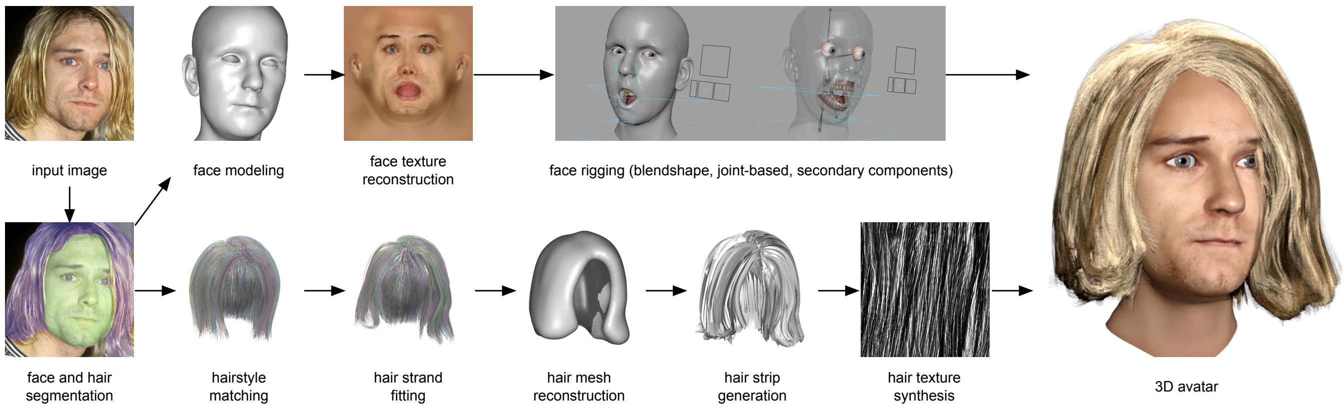

Figure 2: *Our single-view avatar creation framework is based on a pipeline for complete face digitization and one for hair strip digitization.*

such as multi-view stereo cameras [Paris et al. 2008; Jakob et al. 2009; Beeler et al. 2012; Luo et al. 2013; Echevarria et al. 2014] or even thermal imaging [Herrera et al. 2012].

Hu et al. [Hu et al. 2014a] demonstrated a highly robust multi-view hair modeling approach using a data-collection of pre-simulated hair strands, which can fully eliminate the need for manual hair segmentation. Since physically simulated hair strands are used as shape priors, their method can only handle unconstrained hairstyles. The same authors later introduced the use of procedurally generated hair patches [Hu et al. 2014b] to capture highly convoluted hairstyles such as braids. They also moved to a more accessible acquisition approach based on a single RGB-D camera, that is swept around the subject. Single-view hair digitization methods have been pioneered by Chai et al. [Chai et al. 2012; Chai et al. 2013] but rely on high-resolution input photographs and can only produce the frontal geometry of the hair. A database-driven approach by Hu et al. [Hu et al. 2015] later showed that the modeling of complete strand-level hairstyles is possible with the help of very few user strokes as guidance. A similar, but fully automatic approach has been furthered by [Chai et al. 2016] using a larger database for shape retrieval and a deep learning-technique for hair segmentation. While high-quality hair models can be generated, many hairstyles with multiple layers or stochastic structures (e.g., messy ones, afro-hair, etc.), are difficult to capture and not suited for strand-based representations. Furthermore, strand-based hair models are still difficult to integrate into real-time game environments, due to their complexity for real-time hair rendering and simulation. Inspired by cutting-edge game characters, our proposed hair digitization pipeline uses the highly efficient and versatile strip representation for hair. Our deep learning-based synthesis algorithm also produces individualized, realistic, and non-repeating hair textures for each strip.

3 Avatar Modeling Framework

Our end-to-end pipeline for both face and hair digitization is illustrated in Figure 2. An initial pre-processing step computes pixel-level segmentation of the face and hair regions. We then produce a fully rigged avatar based on textured meshes and hair strips from this image.

Image Pre-Processing. Segmenting the face and hair regions of an input image improves the accuracy of the 3D model fitting process, as only relevant pixels are used as constraints. It also provides us additional occlusion areas, that need to be completed during texture reconstruction, for example when the face is covered by hair. For the hair modeling step, the silhouette of the segmented hair region will provide important matching cues.

We adopt the real-time and fully automatic semantic segmentation technique of [Saito et al. 2016a] which uses a two-stream deconvolution network to predict face and hair regions. This technique produces accurate and robust pixel-level segmentations for unconstrained photographs. While the original implementation is designed to process face regions, we repurpose the same convolutional neural network to segment hair. In contrast to the image pre-processing step of [Cao et al. 2016], ours is fully automatic.

To train our convolutional neural network, we collected 9269 images from the public LFW face dataset [Huang et al. 2007] and produce the corresponding binary segmentation masks for both faces and hair via Amazon Mechanical Turk. We detect the face in each image using the popular Viola-Jones face detector [2001] and normalize their positions and scales to a 128×128 image. To avoid overfitting, we augment the training dataset with random Gaussian-distributed transformation perturbations and produce 83421 images in total. The standard deviations are 10° for rotations, 5 pixels for translations, and 0.1 for scale, and the means are 0, 0, and 1.0 respectively. We further use a learning rate of 0.1, a momentum of 0.9, and weight decay of 0.0005 for the training. The optimization uses $50,000$ stochastic gradient descent (SGD) iterations which take roughly 10 hours on a machine with 16GB RAM and NVIDIA GTX Titan X GPU. We refer to the work of [Saito et al. 2016a] for implementation details. Once trained, the network outputs a multi-class probability map (for face and hair) from an arbitrary input image. A posthoc inference algorithm based on dense conditional random field (CRF) [Krähenbühl and Koltun 2011] is then used to extract the resulting binary segmentation mask.

Face Digitization. We decouple the digitization of faces and hair since they span entirely different spaces for shape, appearance, and deformation. While the full head topology of the face is consistent between subjects and expressions, the mesh of the hair model will be unique for each person.

We first fit a PCA-based linear face model for shape and appearance to the segmented face region. We first detect 2D landmarks based using the shape regression method of [Kazemi and Sullivan 2014] to initialize the fitting. We then adopt a variant of the efficient pixel-level analysis-through-synthesis optimization method of [Thies et al. 2016a] to solve for the PCA coefficients of the 3D face model and an initial low-frequency albedo map. We use our own artist head topology with identity shapes transferred from [Blanz and Vetter 1999] and expressions from [Cao et al.

2014b]. We also incorporate a visibility constraint into the model fitting process to improve occlusion handling and non-visible regions. We also construct a PCA-based appearance model for full head textures, using artist-painted skin textures in missing regions of the original data samples. We then infer high-frequency details to the frontal face regions even if they are not visible in the capture using a feature correlation analysis approach based on deep neural networks [Saito et al. 2016b]. Finally, we eliminate the expression coefficients of our linear face model to obtain the neutral expression. The resulting model is then translated and scaled to fit the eye-balls using the average pupillary distance of an adult human 66 mm. We then translate and scale the teeth/gum to fit pre-selected vertices of the mouth region. We ensure that these secondary components do not intersect the face using a penetration test for all the FACS expressions of our custom animation rig.

Hair Digitization. Our hair digitization pipeline can be loosely divided into three parts: (1) we use a strand representation to obtain an accurate model of the subject's hair, (2) the geometry is simplified from strands to strips to reduce rendering complexity and to capture more complex hair textures, and (3) realistic hair textures are generated for the strips by analyzing the structure from visible regions. We first prepare a hairstyle database based on artist designed models and then augment its content using combinatorial approach. We then search for the closest hairstyle to our input image based on the silhouette of its segmentation and the orientation field of the hair strands. As the retrieve hairstyle may not match the input exactly, we further perform a strand-level fitting step to deform the retrieved hairstyle to the input image. The enveloping surface of the hair strands is then constructed using a level-set approach to determine the normal directions of the hair strips. The hair strips are then generated from the hair strands using the local strand directions and the normal vector to the closest point of the reconstructed surface. Next, we produce a global hair texture map that is used by all hair strips, in which the uv-axes of the strips are consistent with those of the texture map. We first extract the feature correlation matrix in the observed hair region and match it with the closest one from a database that contains high-quality hairstyle textures. A large, photorealistic, and non-repeating texture map is then generated using a neural-synthesis approach.

Rigging and Animation. Since our linear face model is expressed by a combination of identity and expression coefficients [Saito et al. 2016b], we can easily obtain the neutral pose. From any input face, we can directly obtain their corresponding FACS-based expressions (including high-level controls) via transfer from a generic face model using an example-based approach [Li et al. 2010]. Our generic face is also equipped with skeleton joints based on linear blend skinning (LBS) [Parke and Waters 2008] in addition to blendshapes, as well as secondary components such as eyes, teeth, and tongue. Our model consists of 71 blendshapes, and 16 joints in total. Our face rig also abstracts the low-level deformation parameters with a smaller and more intuitive set of high-level controls and manipulation handles. We implemented our rig in both, the animation tool, Autodesk Maya, and the real-time game engine, Unity.

Though in high-end production, hair is typically represented by tens of thousands of individual hair strands and animated using physical simulation, we propose a hair strip representation commonly used in gaming. We rig our hair model directly with the skeleton joints of the head to add a minimal amount of dynamics when rotating the head. More sophisticated simulation techniques are possible and already demonstrated in modern games.

4 Face Digitization

We first build a fully textured head model using a multi-linear PCA face model. Given a single unconstrained image and the corresponding segmentation mask, we compute a shape V, a low-frequency facial albedo map I, a rigid head pose (R, t), a perspective transformation $\Pi_P(V)$ with the camera intrinsic matrix P, and illumination L, together with high-frequency textures from the visible skin region. Since the extracted high-frequency texture is incomplete from a single-view, we infer the complete texture map using a facial appearance inference method based on deep neural networks [Saito et al. 2016b].

3D Head Modeling. To obtain the unknown parameters $\chi = \{V, I, R, t, P, L\}$, we adopt the pipeline of [Thies et al. 2016a] which is based on morphable face models [Blanz and Vetter 1999] extended with a PCA-based facial expression model and an efficient optimization based on pixel color constraints. We further incorporate pixel-level visibility constraints using our segmentation mask obtained using the method of [Saito et al. 2016a].

We use a multi-linear PCA model to represent the low-frequency facial albedo I and the facial geometry V with $n = 10,822$ vertices and $21,510$ faces:

$$V(\alpha_{id}, \alpha_{exp}) = \bar{V} + A_{id}\alpha_{id} + A_{exp}\alpha_{exp},$$
$$I(\alpha_{al}) = \bar{I} + A_{al}\alpha_{al}.$$

Here $A_{id} \in \mathbf{R}^{3n\times 40}$, $A_{exp} \in \mathbf{R}^{3n\times 40}$, and $A_{al} \in \mathbf{R}^{3n\times 40}$ are the basis of a multivariate normal distribution for identity, expression, and albedo with the corresponding mean: $\bar{V} = \bar{V}_{id} + \bar{V}_{exp} \in \mathbf{R}^{3n}$, and $\bar{I} \in \mathbf{R}^{3n}$, and the corresponding standard deviation: $\sigma_{id} \in \mathbf{R}^{40}$, $\sigma_{exp} \in \mathbf{R}^{40}$, and $\sigma_{al} \in \mathbf{R}^{40}$. A_{id}, A_{al}, $\bar{V}$, and $\bar{I}$ are based on the Basel Face Model database [Paysan et al. 2009] and A_{exp} is obtained from FaceWarehouse [Cao et al. 2014b]. We assume Lambertian surface reflectanceand approximate illumination using second order Spherical Harmonics (SH).

First, we detect 2D facial landmarks $f_i \in \mathcal{F}$ using the method of Kazemi et al. [Kazemi and Sullivan 2014] in order to initialize the face fitting by minimizing the following energy:

$$E_{lan}(\chi) = \frac{1}{|\mathcal{F}|}\sum_{f_i\in\mathcal{F}} \|f_i - \Pi_P(RV_i + t)\|_2^2.$$

We further refine the shape and optimize low-frequency albedo, as well as illumination, by minimizing the photometric difference between the input image and a synthetic face rendering. The objective function is defined as:

$$E(\chi) = w_c E_c(\chi) + w_{lan}E_{lan}(\chi) + w_{reg}E_{reg}(\chi), \qquad (1)$$

with energy term weights $w_c = 1$, $w_{lan} = 10$, and $w_{reg} = 2.5 \times 10^{-5}$ for the photo-consistency term E_c, the landmark term E_{lan}, and the regularization term E_{reg}. Following [Saito et al. 2016b], we also ensure that the photo-consistency term E_c is only evaluated for visible face regions:

$$E_c(\chi) = \frac{1}{|\mathcal{M}|}\sum_{p\in\mathcal{M}} \|C_{input}(p) - C_{synth}(p)\|_2,$$

where C_{input} is the input image, C_{synth} the rendered image, and $p \in \mathcal{M}$ a visibility pixel given by the facial segmentation mask. The regularization term E_{reg} is defined as:

$$E_{reg}(\chi) = \sum_{i=1}^{40}\left[(\frac{\alpha_{id,i}}{\sigma_{id,i}})^2 + (\frac{\alpha_{al,i}}{\sigma_{al,i}})^2\right] + \sum_{i=1}^{40}(\frac{\alpha_{exp,i}}{\sigma_{exp,i}})^2.$$

This term encourages the coefficients of the multi-linear model to conform a normal distribution and reduces the chance to converge into a local minimum. We use an iteratively reweighted Gauss-Newton method to minimize the objective function (1) using three levels of image pyramids. In our experiments, 30, 10, and 3 Gauss-Newton steps were sufficient for convergence from the coarsest level to the finest one.

After optimization, a high-frequency albedo texture is obtained by factoring out the shading component consisting of the illumination L and the surface normal from the input image. The resulting texture map is stored in the uv texture map and used for high-fidelity texture inference.

Face Texture Reconstruction. After obtaining the low-frequency albedo map and a partially visible fine-scale texture, we can infer a complete high-frequency texture map, as shown in Figure 3, using a deep learning-based transfer technique and a high-resolution face database [Ma et al. 2015]. The technique has been recently introduced in [Saito et al. 2016b] and is based on the concept of feature correlation analysis using convolutional neural networks [Gatys et al. 2016].

Given an input image I and a filter response $F^l(I)$ on the layer l of a convolutional neural network, the feature correlation can be represented by a normalized Gramian matrix $G^l(I)$:

$$G^l(I) = \frac{1}{M_l} F^l(I) \left(F^l(I)\right)^T$$

Saito et al. [2016b] have found that high-quality facial details (e.g., pores, moles, etc.) can be captured and synthesized effectively using Gramian matrices. Let I_0 be the low-frequency texture map and I_h be the high-frequency albedo map with the corresponding visibility mask M_h. We aim to represent the desired feature correlation G_h as a convex combination of $G(I_i)$, where $I_1, ..., I_k$ are the high-resolution images in the texture database:

$$G^l_h = \sum_k w_k G^l(I_k), \forall l \quad \text{s.t.} \sum_{k=1}^{K} w_k = 1.$$

We compute an optimal blending weight $\{w_k\}$ by minimizing the difference between the feature correlation of the partial high-frequency texture I_h and the convex combination of the feature correlations in the database under the same visibility. This is formulated as the following problem:

$$\begin{array}{lrcll} \min\limits_{w} & \sum_l \left\| \sum_k w_k G^l_{\mathcal{M}}(I_k, M_h) - G^l_{\mathcal{M}}(I_h, M_h) \right\|_F & & & \\ \text{s.t.} & \sum_{k=1}^{K} w_k & = & 1 & \\ & w_k & \geq & 0 & \forall k \in \{1, \dots, K\} \end{array} \tag{2}$$

,

where $G_{\mathcal{M}}(I, M)$ is the Gramian Matrix computed from only the masked region M. This allows us to transfer multi-scale features of partially visible skin details to the complete texture. We refer to [Saito et al. 2016b] for more detail.

Once the desired G_h is computed, we update the albedo map I so that the resulting correlation $G(I)$ is similar to G_h, while preserving the low frequency spacial information $F^l(I_0)$ (i.e., position of eye brows, mouth, nose, and eyes):

$$\min_I \sum_{l \in L_F} \left\| F^l(I) - F^l(I_0) \right\|_F^2 + \alpha \sum_{l \in L_G} \left\| G^l(I) - G_h \right\|_F^2, \tag{3}$$

where L_G is a set of high-frequency preserving layers and L_F a set of low-frequency preserving layers in VGG-19 [Simonyan and Zisserman 2014]. A weight α balances the influence of high frequency and low frequency and $\alpha = 2000$ is used for all our experiments. Following Gatys et al. [2016], we solve Equation 3 using an L-BFGS solver. Since only frontal faces are available in the database, we can only enhance face regions in the front. To obtain a complete texture, we combine the results with the PCA-based low-frequency textures of the back of the head using Poisson blending [Pérez et al. 2003].

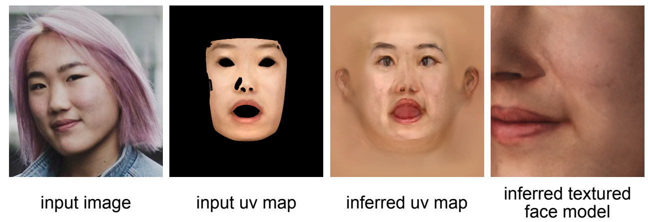

Figure 3: *We produce a complete and high-fidelity texture map from a partially visible and low resolution subject using a deep learning-based inference technique.*

5 Hair Digitization

Hairstyle Database. Starting from the *USC-HairSalon* 3D hairstyle database introduced in [Hu et al. 2015], we align all the hairstyle samples to the PCA mean head model $\bar{V}$ used in Section 4. Inspired by [Chai et al. 2015], we also increase the number of samples in *USC-HairSalon 3D* using a combinatorial process to eliminate the need of an online model generation which requires user interactions [Hu et al. 2015].

We first group each sample of the *USC-HairSalon 3D* into 5 clusters via k-means clustering using the root positions and the strand shapes as in [Wang et al. 2009]. Next, for every pair of hairstyles, we randomly pick a pair of strands among the cluster centroids. Next, we construct a new hairstyle using these two sampled strands as a guide using the volumetric combination method introduced in [Hu et al. 2015]. We further augment our database by flipping each hairstyle w.r.t. the x-axis plane, forming a total of 100,000 hairstyles.

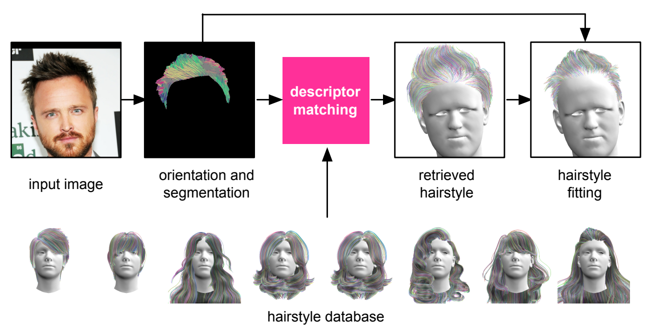

Figure 4: *We use the hair segmentation mask and orientation field to retrieve the best matching hairstyle from a largely augmented database. This hairstyle is then refined to fit the input segmentation and orientation field.*

Hairstyle Matching. Given the segmented hair region in the input image, we first search for a set of candidate hairstyles, which have similar silhouettes as the input. We adopt the highly efficient binary-edge descriptor from [Zitnick 2010] to describe silhouette regions. Once the number of potential candidates has been reduced to 100, we further compare the segmentation mask and hair orientations at the pixel level using rendered thumbnails to retrieve the most similar hairstyle [Chai et al. 2016]. Figure 4 demonstrates this matching scheme as well as the hairstyle fitting result. Following [Chai et al. 2016], we organize our database as thumbnails and descriptors for style matching. For each hairstyle in the database, we render the mask and the orientation map as a thumbnail from 35 different views, where 7 angles are uniformly sampled in $[-\pi/4, \pi/4]$ as yaw and 5 angles in $[-\pi/4, \pi/4]$ as pitch.

Hairstyle Fitting. The retrieved exemplar provides a rough estimation of the actual hairstyle, but the silhouette and orientation may not be perfectly aligned with the input because of the diversity of hairstyles. Hence we adopt a fitting algorithm to deform each strand in the exemplar to match the silhouette and orientation observed from the input image. More specifically, we first search for a smooth warping function $\mathcal{W}(\cdot)$ by maping vertices on the exemplar's silhouette onto new positions on the input's silhouette [Chai et al. 2016], and deform each hair strand with this 2D wrapping function by least moving distance. Then, we deform each strand according to the input 2D orientation map following the method introduced in [Hu et al. 2015]. We refer to [Hu et al. 2015; Chai et al. 2016] for more details.

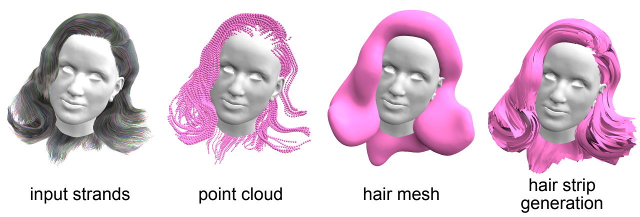

Figure 5: *To convert hair strands into strips, we first compute the surface of the hair to get the normals of the hair strips. The hair strands are resampled into a volumetric point cloud, then a surface extracted from its corresponding signed distance field. The hair strips are then formed along the strand directions and mesh surface normals.*

Hair Mesh Reconstruction. By considering each hair strand as a chain of particles, the set of all particles forms the outer surface of the entire hair. These intermediate representations are shown in Figure 5. This surface can be constructed using a signed distance field obtained by volumetric points samples [Zhu and Bridson 2005]. Such level-set extraction method is commonly used to extract liquid interfaces in fluid simulations. We use a variant of the marching cubes algorithm [Museth et al. 2013] to convert the implicit surface into a coarse mesh, which is used to estimate normal directions for our hair strips.

Hair Strips Generation. Given the fitted hair strands and the reconstructed hair mesh, we compose close and nearly parallel hair strands into a hair strip, which is a parametric piece-wise linear patch. This thin surface structure can carry realistic looking textures to provide additional variations of hair, such as curls, crossings, or thinner tips. Additionally, the transparency of the texture allows us to see through the overlay of different strips and provide an efficient way to achieve volumetric renderings of hair.

Luo et al. [2013] proposed a method to group short hair segments into a ribbon structure. Adopting the similar method, we start from the longest hair strand in the hairstyle as the center strand of the strip. By associating the normal of each vertex on the strand to the closest point on the hair mesh, we can expand the center strand on both sides of the binormal as well as its opposite direction. We compute the coverage of all hair strands by the current strip, and continue to expand the strip until no more strands are covered. Once a strip is generated, we remove all the covered strands in the hairstyle, and reinitiate process from the longest strand in the remaining subset hairstyle. Finally, we obtain a complete hair strips model, once all the hair strands are removed from the hairstyle. we refer to [Luo et al. 2013] for more details.

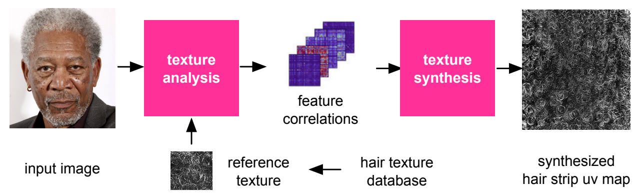

Figure 6: *To generate photorealistic and non-repetitive texture maps for the hair strips, we first extract the feature correlations from the segmented hair region and find the closest match in our hair texture style database. The final texture map for the hair strips is then synthesized by the matching feature correlation.*

Hair Texture Synthesis. We unwrap each wisp to be a long thin rectangle in uv-space and pack them along one axis. This allows us to build a rectangle hair texture where one direction corresponds to how strands are grown from top to down. In order to get a large enough texture for the whole hair mesh while keeping the appearance consistent and non-repeating, we adopt the recently introduced neural synthesis approach for style transfer [Gatys et al. 2016; Saito et al. 2016b], as we do in section 4. The appearance characteristic (often referred to as "style") can be described by the Gramian matrix, i.e. the correlation of middle-layer responses from a pre-trained neural network [Simonyan and Zisserman 2014]. In particular, the Gramian matrix can distinctively describe the appearance of an image at multiple scales and capture both from low-level and high-level features.

The texture synthesis consists of optimizing for all the pixel intensities by enforcing similarity between the output Gramian matrix from the synthesized texture and the one obtained from a high-quality reference image (our hairstyle texture). In our optimization

$$I^* = \min_I \sum_{l \in L_G} \left\| G^l(I) - G^* \right\|_F^2$$

where G^* is the reference. In order to find G^*, we first compute the Gramian matrix $\hat{G}$ from the segmented hair region of the input image. While $\hat{G}$ could describe the texture style of the hair, it may be deteriorated by segmented pixels near the boundary, low resolution and noisy input, or bad lighting conditions. We, therefore, search for the most similar matrix G^* from a pre-selected high-quality hairstyle texture database containing over 500 images for the synthesis. Figure 6 illustrates this synthesis process. Note that the Gramian matrix is anisotropic and rotation variant, thus the expanded texture will follow the same direction as reference texture. This is not a limitation but a useful property, given that the strip-based meshes are readily oriented and the desired texture can immediately follow its direction.

6 Results

We created fully-rigged 3D avatars with secondary components (eye, teeth, tongue, etc.) of subjects with different genders, ages,

Figure 7: *Our proposed framework successfully generates high-quality and fully rigged avatars from a single input image in the wild. We demonstrate the effectiveness on a wide range of subjects with different hairstyles. We visualize the face meshes and hair strips, as well as their textured renderings.*

and hairstyles. Our samples include Internet pictures of celebrities and our own image datasets. Each output is digitized automatically from a single picture of different resolutions and there is no a-priori knowledge about the scene illumination nor intrinsic camera parameters. Some heads are tilted, some are covered with hair, and some have expressions. The processed hairstyles also have different types of high-frequency hair textures ranging from straight ones, messy ones, and afro-hair styles. As illustrated in Figure 7, our proposed framework successfully digitizes believable models of faces and hair, with complete textures obtained via inference based on deep neural synthesis. Facial details are faithfully reproduced in unseen regions and non-repeating naturally looking hair textures can be generated on top of the hair strips. Our accompanying video also shows several animations produced by a professional animator using the provided controls of or avatar.

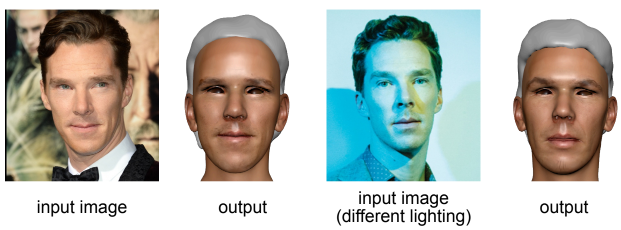

Figure 8: *Single-view modeling when the subject is captured under different lighting conditions.*

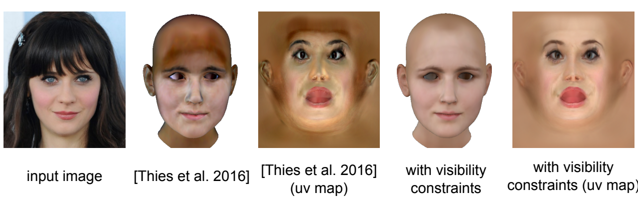

Figure 9: *We visualize the effect of our visibility constraints when estimating the PCA-based albedo map.*

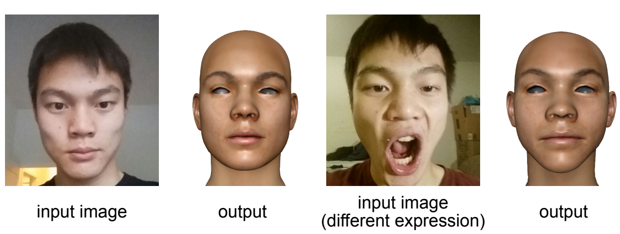

Figure 10: *Single-view modeling when the subject is performing different expressions.*

Evaluation. We evaluate the robustness of our system and consistency of the reconstruction on several challenging input examples. Our combined facial segmentation [Saito et al. 2016a], texture inference [Saito et al. 2016b] and PCA-based shape, appearance, and lighting estimation [Thies et al. 2016a] framework is robust to severe lighting conditions. In Figure 8, we can observe that the visual difference between the reconstructed albedo map of a same person, captured under extremely contrasting illuminations, is minimal. We also show that without our visibility constraints, subjects with hair fringes cannot be handled correctly as shown in Figure 9. We also demonstrate how our linear face model can discern between a person's identity and its expressions robustly. We reconstruct the same person in Figure 10 when performing different expressions. Our visualization shows the resulting avatar in the neutral pose. While some slightly noticeable dissimilarity remains, both outputs are plausible.

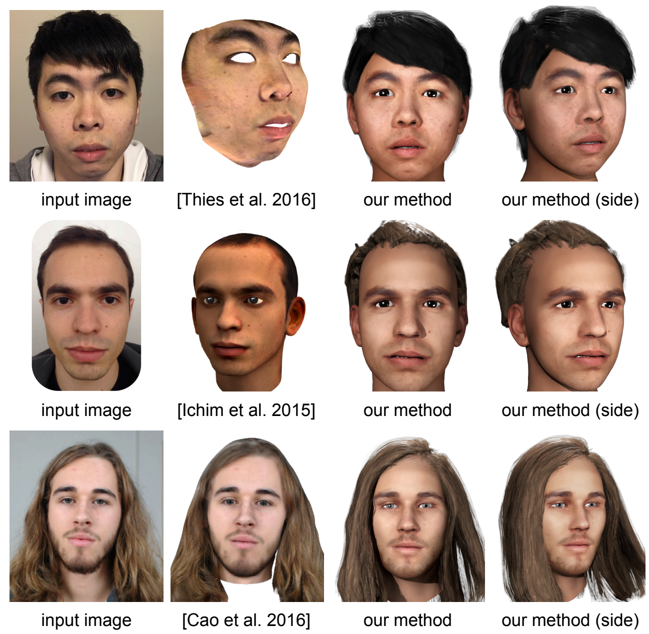

Figure 11: *We compare our method with several state-of-the-art avatar creation systems.*

Comparison. We compare our method against several state-of-the-art facial modeling techniques and avatar creation systems in Figure 11. Our framework can infer facial textures with more details comparing to linear morphable face models [Blanz and Vetter 1999; Thies et al. 2016a], as well as inpaint the non-visible regions, using a deep learning-based facial appearance inference method [Saito et al. 2016b]. In addition to producing high-quality hair models, our generated face meshes and textures are visually comparable to the video-based reconstruction system of Ichim et al. [2015]. We can also reproduce similarly compelling avatars as in [Cao et al. 2016], but using only one out of many of their input images. While their approach is still associated with some manual labor, our system is fully automatic.

Performance. All our experiments are performed using an Intel Core i7-5930K CPU with 3.5 GHz equipped with a GeForce GTX Titan X with 12 GB memory. 3D head model reconstruction takes 5 minutes in total, consisting of 0.5 second of face model fitting, 75 s of feature correlation extraction, 14 s of computing the convex blending weight, 172 s of the final synthesis optimization. The secondary component fitting and facial rigging are done within 1 second.

Hair strips reconstruction takes 2 seconds to retrieve the closest exemplar and 10 minutes to deform a hairstyle containing $10,000$ strands. 5 seconds are needed to reconstruct the hair mesh, and 10 minutes to generate the final hair strips. The hair texture reconstruction needs less than 10 seconds to compute the Gramian matrix of the visible region and retrieve the most similar one. And synthesizing a 500×500 texture takes 116 seconds (for $1,000$ iterations). We also tried to synthesize textures up to 1024×1024, which still takes less than 10 minutes to complete.

7 Discussion

We have demonstrated that the fully automatic digitization of 3D avatars, including hair, is possible from a single image, captured in an uncontrolled environment. We can produce animator-friendly rigged models of a person with intuitive blendshapes and joint-based controls, as illustrated in our animation examples. Even though the subject is only partially visible, the image of low resolution, and the illumination conditions unknown, we can obtain high-quality textured meshes of the face and compelling looking volumetric hair renderings similar to cutting-edge game characters. Our approach is qualitatively comparable to existing avatar creation systems, which require multiple photographs and manual input [Ichim et al. 2015; Cao et al. 2016].

The effectiveness of our methodology is grounded on a careful integration of state-of-the-art facial shape modeling and texture inference algorithms, as well as a data-driven hair modeling pipeline based on hair strips. Several key components, such as semantic segmentation and feature correlation analysis, are only possible due to recent advances in deep learning. Our efficient and versatile hair strip representation is compatible with existing game engines, such as Unity, and suitable for the integration in real-time environments. We have shown that our neural synthesis-based texture generation algorithm is highly effective in reproducing a wide variety of highly stochastic messy hairstyles. Our experiments also indicate the robustness of our system, where consistent results of the same subject can be obtained when captured from different angles, under contrasting lighting conditions, and with different input expressions.

Limitations. Though believable results can be produced, they are far from perfect. Due to the ill-posed problem of highly incomplete input, the low-dimensionality of our linear face models, and unknown intrinsic camera parameters, our shape models may not be fully accurate and our facial texture inference technique may add details in wrong places. With the dramatic progress in deep learning research, we believe that a massive collection of high-resolution 3D faces in controlled capture settings could be used to improve the fidelity of our face models, as well as the performance of shape fitting algorithms.

While the use of hair strips is highly efficient and a reasonable approximation of strand-based models [Hu et al. 2015; Chai et al. 2016], we only use a simple linear-blend skinning approach to add dynamics to the hair. However, convincing strand-level simulations [Chai et al. 2014] are not yet possible with our representation. Though the use of hair strips can capture the volumetric look of hair as opposed to image-based alternatives [Cao et al. 2016], we cannot handle props such as headwear or glasses. Our method would also fail for longer facial hair such as beards, since our database does not contain these objects. We believe that an object recognition approach and more samples in our database could make such digitization possible.

Future Work. Since our framework is designed around today's real-time rendering environments and facial animation systems, we are still using commonly used parametric models for faces and hair, and the results still look uncanny. In the future, we plan to explore end-to-end deep learning-based inference methods to generate more realistic avatars with dynamic textures and more compelling hair renderings. Researches in generative adversarial networks are promising directions.

References

BEELER, T., BICKEL, B., BEARDSLEY, P., SUMNER, B., AND GROSS, M. 2010. High-quality single-shot capture of facial geometry. *ACM Trans. on Graphics (Proc. SIGGRAPH) 29*, 3, 40:1–40:9.

BEELER, T., HAHN, F., BRADLEY, D., BICKEL, B., BEARDSLEY, P., GOTSMAN, C., SUMNER, R. W., AND GROSS, M. 2011. High-quality passive facial performance capture using anchor frames. *ACM Trans. Graph. 30* (August), 75:1–75:10.

BEELER, T., BICKEL, B., NORIS, G., MARSCHNER, S., BEARDSLEY, P., SUMNER, R. W., AND GROSS, M. 2012. Coupled 3d reconstruction of sparse facial hair and skin. *ACM Trans. Graph. 31* (August), 117:1–117:10.

BÉRARD, P., BRADLEY, D., GROSS, M., AND BEELER, T. 2016. Lightweight eye capture using a parametric model. *ACM Trans. Graph. 35*, 4 (July), 117:1–117:12.

BLAKE, A., ROMDHANI, S., VETTER, T., AMBERG, B., AND FITZGIBBON, A. 2007. Reconstructing high quality face-surfaces using model based stereo. *2007 11th IEEE International Conference on Computer Vision 00*, undefined, 1–8.

BLANZ, V., AND VETTER, T. 1999. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 187–194.

BOOTH, J., ROUSSOS, A., ZAFEIRIOU, S., PONNIAH, A., AND DUNAWAY, D. 2016. A 3d morphable model learnt from 10,000 faces. In *IEEE CVPR*.

BOUAZIZ, S., WANG, Y., AND PAULY, M. 2013. Online modeling for realtime facial animation. *ACM Trans. Graph. 32*, 4 (July), 40:1–40:10.

BRADLEY, D., HEIDRICH, W., POPA, T., AND SHEFFER, A. 2010. High resolution passive facial performance capture. *ACM Trans. Graph. 29*, 41:1–41:10.

CAO, X., WEI, Y., WEN, F., AND SUN, J. 2013. Face alignment by explicit shape regression. *International Journal of Computer Vision*.

CAO, C., HOU, Q., AND ZHOU, K. 2014. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Trans. Graph. 33*, 4 (July), 43:1–43:10.

CAO, C., WENG, Y., ZHOU, S., TONG, Y., AND ZHOU, K. 2014. Facewarehouse: A 3d facial expression database for visual computing. *IEEE TVCG 20*, 3, 413–425.

CAO, C., WU, H., WENG, Y., SHAO, T., AND ZHOU, K. 2016. Real-time facial animation with image-based dynamic avatars. *ACM Transactions on Graphics (TOG) 35*, 4, 126.

CHAI, M., WANG, L., WENG, Y., YU, Y., GUO, B., AND ZHOU, K. 2012. Single-view hair modeling for portrait manipulation. *ACM Trans. Graph. 31*, 4 (July), 116:1–116:8.

CHAI, M., WANG, L., WENG, Y., JIN, X., AND ZHOU, K. 2013. Dynamic hair manipulation in images and videos. *ACM Trans. Graph. 32*, 4 (July), 75:1–75:8.

CHAI, M., ZHENG, C., AND ZHOU, K. 2014. A reduced model for interactive hairs. *ACM Trans. Graph. 33*, 4 (July), 124:1–124:11.

CHAI, M., LUO, L., SUNKAVALLI, K., CARR, N., HADAP, S., AND ZHOU, K. 2015. High-quality hair modeling from a single portrait photo. *ACM Trans. Graph. 34*, 6 (Oct.), 204:1–204:10.

CHAI, M., SHAO, T., WU, H., WENG, Y., AND ZHOU, K. 2016. Autohair: Fully automatic hair modeling from a single image. *ACM Trans. Graph. 35*, 4 (July), 116:1–116:12.

CHOE, B., AND KO, H. 2005. A statistical wisp model and pseudophysical approaches for interactivehairstyle generation. *IEEE Trans. Vis. Comput. Graph. 11*, 2, 160–170.

COOTES, T. F., EDWARDS, G. J., AND TAYLOR, C. J. 2001. Active appearance models. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 6, 681–685.

CRISTINACCE, D., AND COOTES, T. 2008. Automatic feature localisation with constrained local models. *Pattern Recogn. 41*, 10, 3054–3067.

DEBEVEC, P., HAWKINS, T., TCHOU, C., DUIKER, H.-P., AND SAROKIN, W. 2000. Acquiring the Reflectance Field of a Human Face. In *SIGGRAPH*.

ECHEVARRIA, J. I., BRADLEY, D., GUTIERREZ, D., AND BEELER, T. 2014. Capturing and stylizing hair for 3d fabrication. *ACM Trans. Graph. 33*, 4 (July), 125:1–125:11.

GARRIDO, P., VALGAERTS, L., WU, C., AND THEOBALT, C. 2013. Reconstructing detailed dynamic face geometry from monocular video. In *ACM Trans. Graph. (Proceedings of SIGGRAPH Asia 2013)*, vol. 32, 158:1–158:10.

GARRIDO, P., ZOLLHÖFER, M., CASAS, D., VALGAERTS, L., VARANASI, K., PEREZ, P., AND THEOBALT, C. 2016. Reconstruction of personalized 3d face rigs from monocular video. *ACM Trans. Graph. (Presented at SIGGRAPH 2016) 35*, 3, 28:1–28:15.

GARRIDO, P., ZOLLHÖFER, M., WU, C., BRADLEY, D., PÉREZ, P., BEELER, T., AND THEOBALT, C. 2016. Corrective 3d reconstruction of lips from monocular video. *ACM Trans. Graph. 35*, 6 (Nov.), 219:1–219:11.

GATYS, L. A., ECKER, A. S., AND BETHGE, M. 2016. Image style transfer using convolutional neural networks. In *IEEE CVPR*, 2414–2423.

GHOSH, A., FYFFE, G., TUNWATTANAPONG, B., BUSCH, J., YU, X., AND DEBEVEC, P. 2011. Multiview face capture using polarized spherical gradient illumination. *ACM Trans. Graph. 30*, 6, 129:1–129:10.

HERRERA, T. L., ZINKE, A., AND WEBER, A. 2012. Lighting hair from the inside: A thermal approach to hair reconstruction. *ACM Trans. Graph. 31*, 6 (Nov.), 146:1–146:9.

HSIEH, P.-L., MA, C., YU, J., AND LI, H. 2015. Unconstrained realtime facial performance capture. In *Computer Vision and Pattern Recognition (CVPR)*.

HU, L., MA, C., LUO, L., AND LI, H. 2014. Robust hair capture using simulated examples. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2014) 33*, 4 (July).

HU, L., MA, C., LUO, L., WEI, L.-Y., AND LI, H. 2014. Capturing braided hairstyles. *ACM Transactions on Graphics (Proceedings SIGGRAPH Asia 2014) 33*, 6 (December).

HU, L., MA, C., LUO, L., AND LI, H. 2015. Single-view hair modeling using a hairstyle database. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2015) 34*, 4 (July).

HUANG, G. B., RAMESH, M., BERG, T., AND LEARNED-MILLER, E. 2007. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Tech. Rep. 07-49, University of Massachusetts, Amherst, October.

ICHIM, A. E., BOUAZIZ, S., AND PAULY, M. 2015. Dynamic 3d avatar creation from hand-held video input. *ACM Trans. Graph. 34*, 4, 45:1–45:14.

JACOBSON, A. Part ii: Automatic skinning via constrained energy optimization.

JAKOB, W., MOON, J. T., AND MARSCHNER, S. 2009. Capturing hair assemblies fiber by fiber. *ACM Trans. Graph. 28*, 5 (Dec.), 164:1–164:9.

KAZEMI, V., AND SULLIVAN, J. 2014. One millisecond face alignment with an ensemble of regression trees. In *IEEE CVPR*, 1867–1874.

KEMELMACHER-SHLIZERMAN, I., AND BASRI, R. 2011. 3d face reconstruction from a single image using a single reference face shape. *IEEE TPAMI 33*, 2, 394–405.

KEMELMACHER-SHLIZERMAN, I. 2013. Internet-based morphable model. *IEEE ICCV*.

KIM, T.-Y., AND NEUMANN, U. 2002. Interactive multiresolution hair modeling and editing. *ACM Trans. Graph. 21*, 3 (July), 620–629.

KRÄHENBÜHL, P., AND KOLTUN, V. 2011. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems*, 109–117.

LI, H., WEISE, T., AND PAULY, M. 2010. Example-based facial rigging. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2010) 29*, 3 (July).

LI, H., YU, J., YE, Y., AND BREGLER, C. 2013. Realtime facial animation with on-the-fly correctives. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2013) 32*, 4 (July).

LI, H., TRUTOIU, L., OLSZEWSKI, K., WEI, L., TRUTNA, T., HSIEH, P.-L., NICHOLLS, A., AND MA, C. 2015. Facial performance sensing head-mounted display. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2015) 34*, 4 (July).

LUO, L., LI, H., AND RUSINKIEWICZ, S. 2013. Structure-aware hair capture. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2013) 32*, 4 (July).

MA, W.-C., HAWKINS, T., PEERS, P., CHABERT, C.-F., WEISS, M., AND DEBEVEC, P. 2007. Rapid Acquisition of Specular and Diffuse Normal Maps from Polarized Spherical Gradient Illumination. In *Eurographics Symposium on Rendering*.

MA, D. S., CORRELL, J., AND WITTENBRINK, B. 2015. The chicago face database: A free stimulus set of faces and norming data. *Behavior Research Methods 47*, 4, 1122–1135.

MILLER, E., AND PINSKIY, D. 2009. Realistic eye motion using procedural geometric methods. In *SIGGRAPH 2009: Talks*, ACM, New York, NY, USA, SIGGRAPH '09, 75:1–75:1.

MUSETH, K., LAIT, J., JOHANSON, J., BUDSBERG, J., HENDERSON, R., ALDEN, M., CUCKA, P., HILL, D., AND PEARCE, A. 2013. Openvdb: An open-source data structure and toolkit for high-resolution volumes. In *ACM SIGGRAPH 2013 Courses*, ACM, New York, NY, USA, SIGGRAPH '13, 19:1–19:1.

OLSZEWSKI, K., LIM, J. J., SAITO, S., AND LI, H. 2016. High-fidelity facial and speech animation for vr hmds. *ACM Transactions on Graphics (Proceedings SIGGRAPH Asia 2016) 35*, 6 (December).

PARIS, S., CHANG, W., KOZHUSHNYAN, O. I., JAROSZ, W., MATUSIK, W., ZWICKER, M., AND DURAND, F. 2008.

Hair photobooth: Geometric and photometric acquisition of real hairstyles. *ACM Trans. Graph. 27*, 3 (Aug.), 30:1–30:9.

Parke, F. I., and Waters, K. 2008. *Computer Facial Animation*, second ed. AK Peters Ltd.

Paysan, P., Knothe, R., Amberg, B., Romdhani, S., and Vetter, T. 2009. A 3d face model for pose and illumination invariant face recognition. In *Advanced video and signal based surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, IEEE, 296–301.

Pérez, P., Gangnet, M., and Blake, A. 2003. Poisson image editing. In *ACM Transactions on Graphics (TOG)*, vol. 22, ACM, 313–318.

Ramanan, D. 2012. Face detection, pose estimation, and landmark localization in the wild. In *IEEE CVPR*, 2879–2886.

Ren, S., Cao, X., Wei, Y., and Sun, J. 2014. Face alignment at 3000 fps via regressing local binary features. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, IEEE, 1685–1692.

Saito, S., Li, T., and Li, H. 2016. Real-time facial segmentation and performance capture from rgb input. In *ECCV*.

Saito, S., Wei, L., Hu, L., Nagano, K., and Li, H., 2016. Photorealistic facial texture inference using deep neural networks.

Saragih, J. M., Lucey, S., and Cohn, J. F. 2011. Deformable model fitting by regularized landmark mean-shift. *Int. J. Comput. Vision 91*, 2, 200–215.

Shi, F., Wu, H.-T., Tong, X., and Chai, J. 2014. Automatic acquisition of high-fidelity facial performances using monocular videos. *ACM Trans. Graph. 33*, 6, 222:1–222:13.

Sifakis, E., Neverov, I., and Fedkiw, R. 2005. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. Graph. 24*, 3 (July), 417–425.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR abs/1409.1556*.

Terzopoulos, D., and Waters, K. 1990. Physically-based facial modelling, analysis, and animation. *The journal of visualization and computer animation 1*, 2, 73–80.

Thies, J., Zollhöfer, M., Stamminger, M., Theobalt, C., and Niessner, M. 2016. Face2face: Real-time face capture and reenactment of rgb videos. In *IEEE CVPR*.

Thies, J., Zollöfer, M., Stamminger, M., Theobalt, C., and Niessner, M. 2016. FaceVR: Real-Time Facial Reenactment and Eye Gaze Control in Virtual Reality. *arXiv preprint arXiv:1610.03151*.

Viola, P., and Jones, M. 2001. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, IEEE, I–511.

Vlasic, D., Brand, M., Pfister, H., and Popović, J. 2005. Face transfer with multilinear models. *ACM Trans. Graph. 24*, 3 (July), 426–433.

von der Pahlen, J., Jimenez, J., Danvoye, E., Debevec, P., Fyffe, G., and Alexander, O. 2014. Digital ira and beyond: Creating real-time photoreal digital actors. In *ACM SIGGRAPH 2014 Courses*, ACM, New York, NY, USA, SIGGRAPH '14, 1:1–1:384.

Wang, L., Yu, Y., Zhou, K., and Guo, B. 2009. Example-based hair geometry synthesis. *ACM Trans. Graph. 28*, 3, 56:1–56:9.

Ward, K., Bertails, F., yong Kim, T., Marschner, S. R., paule Cani, M., and Lin, M. C. 2006. A survey on hair modeling: styling, simulation, and rendering. In *IEEE TRANSACTION ON VISUALIZATION AND COMPUTER GRAPHICS*, 213–234.

Weise, T., Li, H., Gool, L. V., and Pauly, M. 2009. Face/off: Live facial puppetry. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer animation (Proc. SCA'09)*, Eurographics Association, ETH Zurich.

Weise, T., Bouaziz, S., Li, H., and Pauly, M. 2011. Real-time performance-based facial animation. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2011) 30*, 4 (July).

Weng, Y., Wang, L., Li, X., Chai, M., and Zhou, K. 2013. Hair Interpolation for Portrait Morphing. *Computer Graphics Forum*.

Wu, C., Bradley, D., Garrido, P., Zollhöfer, M., Theobalt, C., Gross, M., and Beeler, T. 2016. Model-based teeth reconstruction. *ACM Trans. Graph. 35*, 6 (Nov.), 220:1–220:13.

Xiong, X., and De la Torre, F. 2013. Supervised descent method and its applications to face alignment. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, IEEE, 532–539.

Yuksel, C., Schaefer, S., and Keyser, J. 2009. Hair meshes. *ACM Trans. Graph. 28*, 5 (Dec.), 166:1–166:7.

Zhu, Y., and Bridson, R. 2005. Animating sand as a fluid. *ACM Trans. Graph. 24*, 3 (July), 965–972.

Zitnick, C. L. 2010. Binary coherent edge descriptors. In *Proceedings of the 11th European Conference on Computer Vision: Part II*, ECCV'10, 170–182.