Crafting 3D Faces Using Free Form Portrait Sketching and Plausible Texture Inference

Tanasai Sucontphunt*
University of Southern California
University of Houston

Borom Tunwattanapong†
University of Southern California

Zhigang Deng‡
University of Houston

Ulrich Neumann§
University of Southern California

Abstract

In this paper we propose a sketch-based interface for drawing and generating a realistic 3D human face with texture. The free form sketch-based interface allows a user to intuitively draw a portrait as in traditional pencil sketching way. Then, the user's drawing is automatically reshaped to an accurate natural human facial shape with the guidance of a statistical description model, and an artistic-style portrait rendering technique is used to render the work-in-progress face sketch. Furthermore, with additional user-specified information, e.g., gender, ethnicity, and skin tone, a realistic face texture can be synthesized for the portrait through our probabilistic face texture inference model. Lastly, the textured portrait will be further used to construct a realistic 3D face model by the 3D morphable face model algorithm. Through our preliminary user evaluations, we found that with this system, even novice users were able to efficiently craft a sound 3D realistic face within three minutes.

Index Terms: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Texture I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques

1 Introduction

Sculpting 3D human faces from scratch is a tedious and painstaking task even for skilled animators. Blendshape or geometric deformation techniques alleviate the pains of this process to a certain extent, but intensive manual work is still involved, especially at their model setup stage. For example, building a well-behaviored blendshape face model [27, 19, 26] or model-specific controls [40] need special care. Another effective way is to take face images of real persons as input and then reconstruct 3D face models using statistical learning techniques [5, 1] or vision-based algorithms [49, 34]. These approaches require users to provide face images (or video clips) of real persons, but these images or video clips are not always available in many real-world applications.

Creating face models from vague memory is even more challenging. A number of tools including FACES [3], Identi-Kit [14], and SpotIt [17], were developed to aid users to create a 2D face sketch from vague memory for the purpose of identifying suspects as in a criminal investigation. However, these tools provide limit controls over user input and primarily focus on non-textured faces. In these systems, the user inputs are often limited to the selection of predefined face components. Also, the leading factors for object identification (i.e. salient facial feature lines of eyes, eyebrow, nose, mouth, and profile [13]) are not sufficiently exploited in the above approaches.

In this paper, we propose an intuitive drawing system for crafting human faces with plausible texture (Figure 1 shows its main components and information flow among them). It consists of a free-form portrait sketching component and an automated facial texture generation component. Inspired by the FiberMesh [33], this system allows users intuitively draw salient facial contour curves directly on the drawing interface, which automatically guides the generation of an artistic-style portrait sketch. During the course of the drawing process, a reduced face description is incorporated into the crafting algorithm on-the-fly to maintain the "faceness" of the face being drawn. In addition, we propose a novel algorithm to probabilistically infer plausible face texture for the sketched portrait based on certain user-specified facial attributes such as gender, ethnicity, and skin tone. Lastly, the 3D morphable face model algorithm [5] is used to reconstruct a photorealistic 3D face model based on the crafted portrait with inferred face texture. Our work can be straightforwardly used in many real-world applications such as criminal investigations, human face modeling for games and entertainment, and drawing applications in online social networks. Figure 2 shows a snapshot of our running system.

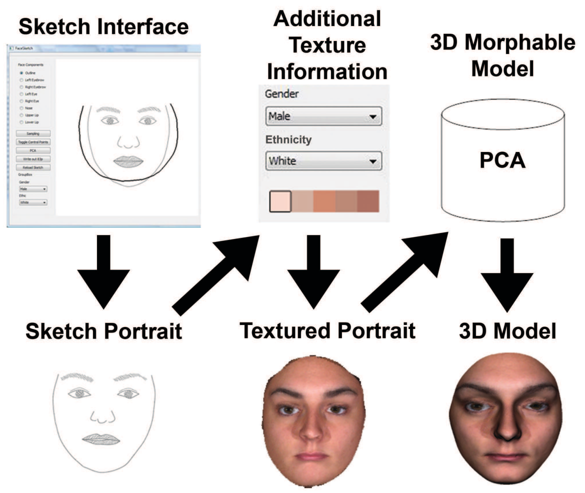

Figure 1: The schematic view of our system. It consists of three main components: a face sketching interface, automated facial texture generation, and 3D face construction.

The major distinctions of this work include:

- *Efficient portrait sketching*. Its portrait drawing system provides users an intuitive sketch-based interface similar to the traditional pencil sketching. This system also guides the automated correction of sketched portraits so that users without professional sketching skills are able to draw and create quality portraits easily. Particularly, in the sketching procedure, each face component, such as face boundary, eyebrows, eyes, nose, and mouth, automatically drives its underlying geometric deformation to maximally maintain its "faceness" through a learned statistical description model.
- *Automated face texture generation*. The colored texture of

*e-mail: sucontph@graphics.usc.edu
†e-mail: tunwatta@usc.edu
‡e-mail: zdeng@cs.uh.edu
§e-mail: uneumann@graphics.usc.edu

a face portrait is automatically generated from the sketched portrait and certain user-provided information including gender, ethnicity, and skin tone. A probabilistic mixture model is utilized to combine this information and search for the most appropriate face textures from a pre-collected face dataset. Then, the top-ranked face textures are merged and morphed to generate the resulting face texture.

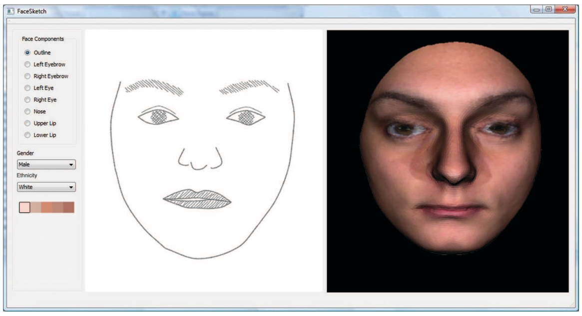

Figure 2: A snapshot of our running system that consists of three main panels. Left: parameter setting for a user to select face component and texture information. Middle: a sketching interface for a user to draw a free-form face. Right: the resulting photorealistic 3D face.

The remainder of this paper is organized as follows. A brief review of related work is given in Section 2. Section 3 describes its offline data processing step. Section 4 describes its drawing interface, automatic reshaping process, and portrait rendering algorithm. Section 5 details how to generate plausible face texture from the drawn portrait and certain user-provided portrait information including gender, ethnicity, and skin tone. Section 6 explains the construction of a 3D face model from the textured portrait. Results are described in Section 7. Finally, in Section 8, concluding remarks and limitations are given and future research directions are discussed.

2 Related Work

In this section we only briefly review recent research efforts in 3D face sculpting that are most related to this work.

Free form sketching: Intensive efforts had been explored on free form sketching [33, 16, 22, 21, 10, 6, 23, 15, 40, 28]. These research efforts were generally focused on creating or altering 3D models using lines to control or guide their geometric deformation, which can be considered as an inverse process of Non-Photorealistic Rendering (NPR). For example, on top of the pioneering Teddy work [16], Nealen *et al.* [33] developed the FiberMesh system that inflates users' 2D stroke silhouette to 3D models. Furthermore, the users can modify the 3D models by pulling any stroke line on the surface of the models. Karpenko *et al.* [22] present an approach to construct hidden 3D curves by drawing 2D contour lines directly. Bourguignon *et al.* [6] introduce a novel method to create a 3D model from a 2D drawing by exploiting its depth cues. Kalnins *et al.* [20] present an effective way to draw strokes directly on 3D models where an interactive user interface is provided for stroke controls. These approaches demonstrated their successes in crafting generalized 3D models. However, without any domain-specific prior, the side effect of their generality is that it is difficult to predict the quality of crafted 3D models without considerable efforts. In our system, we heavily exploit prior knowledge about human faces from a pre-collected dataset of 3D face models for the purpose of face sketching, texturing, and modeling.

Face modeling: The blendshape approach or geometric deformation allows users to generate various facial expressions on a specific model by moving weight control sliders [35, 27, 26] or manipulating control curves [40]. Essentially, these approaches simultaneously move and edit a group of relevant vertices, which saves significant efforts for users. Data-driven 3D face modeling and editing techniques exploit the intrinsic correlations among different facial regions [19, 26, 8, 32, 25] or learn statistical models from a collected face dataset [5, 9, 42, 46, 11]. For example, Blanz and Vetter [5] construct a morphable face model by constructing Principle Component Analysis (PCA) spaces from the 3D geometry and texture of a scanned 3D face dataset. Decarlo *et al.* [9] present an efficient technique to generate 3D face models by imposing face anthropometry statistics as constraints. Lee and Soon [25] simulate exaggerated 3D facial shapes with skin details from a pre-recorded dataset. Sucontphunt *et al.* [42] presented a portrait-based facial expression posing approach where users can manipulate a number of fixed portrait control points which, in turn, are used to search for most similar facial expression poses in a given facial dataset. However, the above approaches mainly focus on sculpting facial expressions on specific 3D face models or crafting face models from input photos, and most of them do not provide an intuitive tool for crafting personalized 3D faces with plausible texture from scratch.

Face texture synthesis: Blanz *et al.* [4] generate 3D faces by combining the morphable face model [5] with additional example images. However, intensive efforts of manual parameter tuning are involved in their approach, and it does not provide direct and intuitive user controls over the shape of the generated faces. For image-based texture synthesis, Mohammed *et al.* [31] generate realistic face textures by an image quilting model. However, their approach focuses on synthesizing random face images without intuitive user controls. Mo *et al.* [29] reconstruct the missing part of a 2D face image based on a statistically learned face prior. Tsumura *et al.* [44] synthesize face skin and texture by extracting hemoglobin and melanin information in the skin. Hertzmann *et al.* [12] presented a technique called "image analogies" to automatically transfer the style of one image to another. Sucontphunt *et al.* [41] synthesize an artistic-style face texture painting to a 2D sketch with a texture-by-number approach from the image analogies [12]. However, the artifact nature of its procedure is not suitable for a realistic face texture generation. Wang and Tang [47] synthesize face textures based on a large set of photographs and corresponding artist-drawn sketch examples. None of these methods attempted to automatically infer a photorealistic face texture from a face portrait created from scratch, without the aid of a training set of example drawings.

3 Offline Data Processing

Our approach utilizes the knowledge of the human face structure from a 3D face dataset [48]. The used 3D face dataset consists of 100 subjects (56 females and 44 males), ranging from 18 to 70 years old with diversified ethnicities. Each entry in the dataset contains a 3D face mesh, texture, and 83 feature points. We processed the face dataset to construct a prior knowledge-base about the human face structure in the offline process. We elaborate major offline data processing steps as follows.

3.1 2D Facial Shape Data Processing

We pre-constructed a 2D morphable face model from X and Y coordinates of 83 feature points of all the 100 faces in the dataset. We computed the PCA bases (eigen-vectors) of these feature point positions. In this work, we kept the most dominant 29 eigen-vectors, and the reduced 2D face geometry space spanned by these eigenvectors retained more than 95% of the variations.

3.2 3D Mesh Data Processing

We also built a reduced PCA space from 3D face meshes in the dataset. Specifically, this process consists of two steps: *face mesh alignment* and *construction of 3D morphable face model*.

Face mesh alignment: To build the correspondences for all the 3D face models in the dataset, we first chose a standard face model outside the dataset as the template/standard mesh. Then, we deformed the standard model to match the models in the 3D face dataset using an iterated closet point algorithm [43] (83 feature points were used as the markers, refer to Fig. 3). The standard face model served as the source, and the face models in the 3D face dataset were used as the target in this deformation process. In this way, we created 100 aligned 3D face models with the same geometry/mesh structure (*i.e.*, the number of vertices and its connectivity structure).

Construction of 3D morphable face model: Based on the aligned 100 3D face models, we construct a 3D PCA morphable face model [5]. Also, we kept the most dominant eigen-vectors to form a reduced 3D face geometry space, while retaining more than 95% of the variations.

4 Sketching Interface

In the sketching area, users can free-form draw curves to craft a portrait by using a mouse or a tablet. It starts with a standard, artistic-style face portrait as the background in order to give the users rough guidelines about the portrait structure and locations. The users are allowed to draw certain face components including face boundary, eyebrows, eyes, nose, and mouth. For simplicity, the users are required to draw each component separately (i.e., one at a time). After each drawing, its artistic-style portrait is generated and rendered in real-time, which provides instant and vital visual feedback to the users' drawing operations. In addition, the users can redraw the curves to gradually change the face shape.

4.1 Sketching Tool

Before starting a drawing, users are required to select a specific face component. Once a face component is selected, the users can freely draw curves of the component shape in the sketching area as in the traditional pencil sketching. This process simplifies the application and ensures the robustness of the face component drawing rather than employing a fully automatic approach that typically requires a substantial number of training examples [39]. To smooth the drawn curves, we employ a Laplacian curve deformation [33] to remove noise from the users' mis-strokes as well as maintain the naturalness of the sketched face shape.

Then, our system extracts pre-defined control points (83 facial feature points, Fig. 3(a)) from these curves in a two-step protocol. First, the drawn curves are aligned to a component template which is used to identify and detect the anchor point and pivot points of each face component (refer to Fig. 3(b)). The anchor point is the starting or ending point of a curve, located at one of the corners, and the pivot points are other corner points on the sketching curve. These points are detected by our heuristic rules including relative positions and a corner detection algorithm. For example, the corners are detected as the critical points vertically along the curve. Second, from the anchor point and pivot points, the sketched curve is then re-sampled uniformly for the rest of the control points that are pre-defined for different facial components.

The anchor and pivot points for each face component are detected in the following way:

- **Eyebrows and eyes**: Anchor and pivot points are identified as the first and the other corner positions of the stroke, respectively.
- **Nose**: Two anchor points are identified as the first and the last positions of the stroke. Two pivot points are detected as the leftmost and the rightmost corners.
- **Upper lip**: Two anchor points are identified as the first and the other corner positions of the stroke. Three pivot points at the middle of the upper lip are detected as the stationary points along the top curve.
- **Lower lip**: Two anchor points are identified as two corner positions of the stroke.
- **Face boundary**: Two anchor points are identified as the first and the last positions of the stroke.

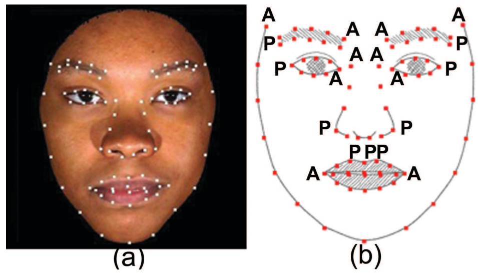

Figure 3: (a): 83 feature point locations. (b): the Anchor points (A) and the Pivot points (P) on the rendered portrait.

During the sketching process, to ensure the quality of the sketched portrait, the drawn curves are reshaped to a proper human facial shape using a 2D morphable face model that was constructed in the offline processing step (Section 3.1). This process also filters the 83 control points as the proper query feeding to the face texture generation (Section 5). At run-time, once users finish the drawing of each face component, its extracted control points are projected to the retained PCA space of 2D face shape, which generates the reshaped control points. Specifically, given a drawing vector S_i, we first project S_i to the retained eigen-vector matrix $EigMX$ to yield its reduced representation C_i (Eq. 1), and then we project C_i back to reconstruct its reshaped drawing vector $\hat{S}_i$ (Eq. 2).

$$C_i = EigMX^T.(S_i - \mu) \tag{1}$$

$$\hat{S}_i = \mu + EigMX * C_i \tag{2}$$

Here, μ is the mean shape of all the faces in the dataset. Fig. 4 illustrates this reshaping process. In addition, if users need to change the position of any control point, the users can click and drag the particular point to adjust the curves.

4.2 Portrait Rendering

At the final step of each drawing, the refined portrait is generated by our portrait rendering technique. Based on the control points extracted from the drawing, a non-photorealistic rendering technique is used to repaint the portrait in an artistic style while retaining its proportions. In this work, we extend a model-based approach [30] as the main renderer for prominent facial components. The key observation of this model-based approach is that artists are heavily guided by their prior knowledge of the human face structure rather than treating it as a generic object. This approach enables the generation of different line drawings (a range of stylization) for different face parts while keeping the control point positions identical. Fig. 3 shows the rendered result of the standard face by our technique. Based on the control points in Fig. 3, each face component is rendered differently in the following ways:

Eyebrows: Each eyebrow contains 10 control points. To render the eyebrow, we first create a bounding tube from these control points. Then, the pencil-style hatching with a pre-defined inclining degree is generated in the bounding tube to mimic the eyebrow appearance.

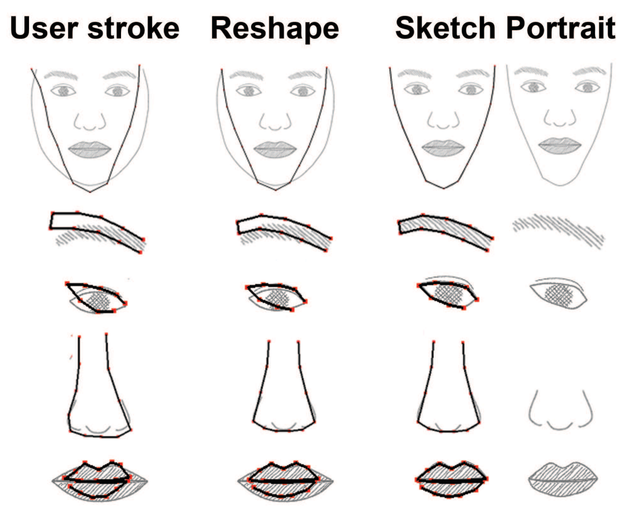

Figure 4: Interactive sketching process. Left: the user strokes after control points (shown as red dots) are extracted. Middle: the drawing curves are reshaped based on the pre-constructed 2D morphable face model to maintain the naturalness of the face. Right: The rendered portraits. Each row shows an example of each face component.

Eyes: Each eye contains 8 control points. Kochanek-Bartels spline is used to draw the top and bottom curves to create an eye shape. Then, an iris with the size of about a half of the eye space is generated by a hatching function. Finally, an eyelid is drawn just above the top of the eye shape using a spline.

Nose: The nose contains 12 control points. The nose shape is generated as the spline curves at the left, middle and right sides of the nose U-shape.

Mouth: The mouth contains 20 control points (10 for the upper lip and 10 for the lower lip). First, the upper and lower lips are drawn as spline curves. Then, a hatching function is used to fill in each lip to create its natural appearance.

Face boundary: Face boundary contains 15 control points. A spline curve is created and drawn to connect these points to form a smooth profile shape.

5 Face Texture Generation

From the above steps, the drawn portrait is a pencil-style drawing without a photorealistic texture. Inferring the plausible face texture for any face sketch is an intrinsically ill-posed problem. However, we can mimic how a human would choose proper textures from a collection of faces to paint a face sketch. In general, by providing certain face attributes such as gender, ethnicity, and skin tone, together with the face sketch, one can make the best guess on appropriate candidate face textures by balancing these information properly. Analogously, a probabilistic model that can associate and balance these information can be used to imitate and automate this process. Hence, the core part of our face texture generation is to employ a probabilistic model to rank the most appropriate textures in the pre-collected face dataset. Subsequently, the top ranked textures are merged and morphed properly to synthesize a novel face texture for the drawn portrait. Lastly, we warp the synthesized texture to the portrait shape. The portrait shape and the synthesized texture are then used to generate the final textured 3D model (Section 6).

5.1 Facial Texture Searching Algorithm

Modeling probabilistic associations between multi-type data is useful in many real-world applications such as automated language translation between English and other languages [7]. The relevance model has demonstrated its successes in modeling the associations between photos and their keyword descriptions [24, 18]. Specifically, it formulates the problem as estimating the joint probability of observing the photos' feature vectors and their keyword description. It shares certain similarities with our problem: in our case, we attempt to model the associations between face textures and face descriptions (e.g., face geometry from the drawing, gender, ethnicity, and skin tone).

The latent variable is the key to the above relevance model. In our work, the employed latent variable that links all the subspaces of face geometry, texture, gender, ethnicity, and skin tone is the person identity, by assuming that if two persons are similar in one subspace, then they should also be similar in the other subspaces. Specifically, we extend the Continuous Relevance Model (CRM) [24] to model the associations between a face identity and a face description. The CRM model associates the subspaces via their joint probabilities that are used to further rank results in a maximum likelihood manner. Their joint probabilities are computed by estimating their probabilities in each subspace of face descriptions. Particularly, for each identity in the dataset, we first compute its similarity to the given input in each subspace and then combine them together using a joint probability model. As a result, the face identity that yields the highest probability means the best similarity balancing across all the inputs.

The joint probability of observing a face geometry G, which is the 83 control points from the drawn portrait, a gender A_1, an ethnicity A_2, and a skin tone A_3 on the same face identity can be computed as the expectation over the identity I in the training set (Eq. 3):

$$P(G,A_1,A_2,A_3|I) = P_G(G|I) \prod_{i=1,2,3} P_{A_i}(A_i|I) \tag{3}$$

Here $P_G(G|I)$ is the probability of the face geometry G in the face geometry subspace given an identity I, $P_{A_i}(A_i|I)$ is the probability of gender A_1, ethnicity A_2, or skin tone A_3 in its corresponding subspace given an identity I. $P_{A_i}(A_i|I)$ can be pre-computed and stored in a lookup table in the offline processing step. In addition, if users only provide partial facial attribute information, then our approach will only use the provided partial information in the above Eq. 3.

Our specific goal is to calculate the ranking score of each identity in the dataset based on $P(G,A_1,A_2,A_3|I)$. Its main difficulty is to estimate its probability in each subspace properly. For the face geometry $P_G(G|I)$ subspace, the solution is trivial since we can calculate the likeness of an input face geometry G by comparing it with existing face geometries I in the dataset. Since not every element of the 83 feature points (i.e. the face geometry) is equally important, we first reduce its dimensionality by projecting it into the truncated PCA space constructed in Section 3.1 and use it as the geometry representation. Then, the Euclidean distance between this G and I in the reduced PCA space is used to infer the $P_G(G|I)$.

For $P_{A_3}(A_3|I)$, the skin tone similarity is also straightforward to measure. We cropped a part of the forehead region to represent a skin tone. In our work, we estimate $P_{A_3}(A_3|I)$ using a Gaussian Mixture Models (GMM) by first clustering similar skin tones in the dataset into K multivariate Gaussian distribution groups. K is chosen automatically using the Minimum Description Length (MDL) criteria [37] (K = 5 in our work). These K groups' centroids are also used as the skin tones for users to select in the input channel. Then, the likelihood of a skin tone (A_3) given a face identity (I) is defined as follows.

$$P_{A_3}(A_3|I) = \frac{1}{(2\pi)^{\frac{M}{2}}} |C_k|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(I-\mu_k)^t C_k^{-1}(I-\mu_k)\} \quad (4)$$

Here k is the group that the input A_3 belongs to, C_k is a M x M covariance matrix for the k^{th} group, and μ_k is the mean of the k^{th} group.

The estimation of $P_{A_1}(A_1|I)$ and $P_{A_2}(A_2|I)$ is not straightforward since gender and ethnicity are not tightly explaining the face identity. In other words, the associations between gender/ethnicity and face identity are discrete and severely sparse. Therefore, we transform these discrete subspaces into continuous subspaces by clustering their genders and ethnicities in a texture similarity space.

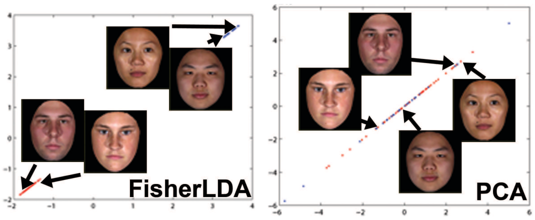

Figure 5: Illustration of the White and the Asian ethnic subspaces constructed by FisherLDA. The FisherLDA separates ethnicities properly (left), comparing with the traditional PCA approach (right).

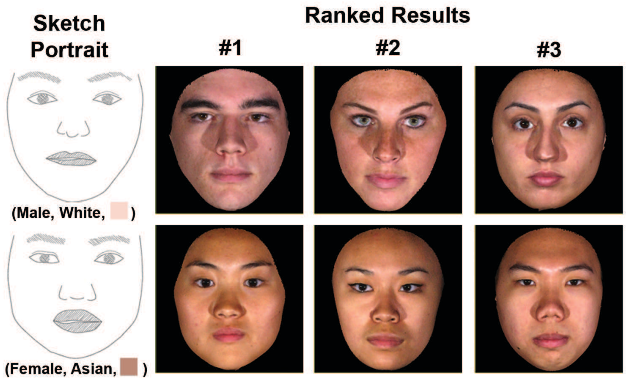

Figure 6: Three top-ranked face textures based on a given sketch portrait and additional information such as gender, ethnicity, and skin tone. The additional information is shown in the parenthesis in the format of (gender, ethnicity, skin tone). From the ranked results, there are mixtures of genders in both results since our search algorithm tries to balance portrait geometry, gender, ethnicity, and skin tone inputs. Particularly, in these two examples, face geometry, ethnicity, and skin tone have a stronger coherence than gender.

5.2 Facial Texture Space

In order to prepare the texture space for the estimation of $P_{A_1}(A_1|I)$ and $P_{A_2}(A_2|I)$, we use FisherLDA [2] to rearrange the textures in the dataset based on gender and ethnicity. The FisherLDA technique is similar to PCA method [45] in the way that it creates a subspace based on the dominant principle components of the dataset, but the FisherLDA subspace is also constructed to maximize between-group variances while minimizing within-group variances. As shown in Fig. 5, FisherLDA closely groups the face textures with the same ethnicity together by providing their ethnicity labels. Accordingly, the $P_{A_1}(A_1|I)$ and $P_{A_2}(A_2|I)$ are computed based on the Mahalanobis distance from I to the cluster of the inputted gender A_1 and ethnicity A_2 in the FisherLDA subspaces, respectively.

5.3 Texture Morphing and Warping

We morphed the three top-ranked textures using the Biharmonic Spline Interpolation [38] that is, essentially, a weighted linear combination (Eq. 5). Fig. 6 shows the three top-ranked textures, and Fig. 7 shows the result of morphing the textures. Note that users can specify the number of top ranked textures for morphing. From our empirical study, morphing three top-ranked textures generally gives appropriate results.

$$T_{morphed} = \sum_{i=1}^{3} w_i * T_i \quad (5)$$

Here w_i is the $P(G, A_1, A_2, A_3|I)$ of the texture T_i (the identity I's texture), and T_i is the i^{th} ranked texture.

After the texture morphing process, we further map the final texture to the portrait G. In this process, we use the control points of G and $T_{morphed}$ to guide the texture warping. Fig. 7 shows one texture warping example.

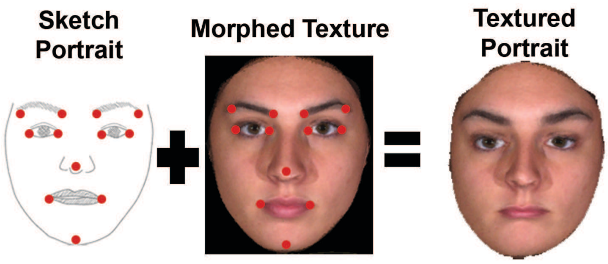

Figure 7: In the case of the top-row example in Fig. 6, based on the control points of the sketched portrait (partially shown as red dots), the morphed texture is mapped to the portrait.

6 3D Face Construction

Finally, we construct the 3D textured face model by using a 3D morphable model technique [5]. Given the obtained top-ranked face textures in the above step, we can also obtain their corresponding 3D face meshes. These meshes are the natural candidates to represent the 3D face model for the portrait. We morph these 3D meshes together with the same weights as in Eq. 5 to produce the target 3D face model. Then, we translate X and Y coordinates of the 83 control points on the target 3D face model according to X and Y values of the drawn portrait while retaining the Z coordinates of the target 3D face model. Subsequently, we use these 3D 83 control points to morph the remaining vertices of the target 3D face model using a Radial Basis Functions (RBF) based deformation technique [36]. Then, we map the textured portrait to the constructed 3D face model through a straightforward projective texture mapping. Finally, to preserve the "faceness" of the 3D model, we project the 3D target model to the 3D morphable face model space (created in Section 3.2) and reconstruct the final photorealistic 3D face model. It is noteworthy that comparing with the state of the art work by Blanz and Vetter [5] that requires an offline optimization process and thus cannot be real-time, our method is able to produce a photorealistic 3D face model in real-time based on interactive user sketching inputs.

7 Results and Preliminary Evaluations

To evaluate the effectiveness and usability of our approach, we conducted a preliminary user study. Six novice users were invited to participate in the study. As a familiarization step, we instructed all the participants how to use our system for about 5 minutes. In the study, the participants were asked to draw a portrait. Then, given additional information of gender, ethnicity, and skin tone, our system was used to generate its final 3D face model with photorealistic texture. We asked the participants to draw target faces based on given photo examples. Each participant was required to use our system to draw target faces as quickly as possible while maintaining the quality as much as they can. The used crafting time was recorded, and the produced final 3D face models were also retained. We found that using our system, on average they spent less than three minutes to sketch a face portrait and to generate its corresponding photorealistic 3D face model. Fig. 8 shows several examples of the final face models and corresponding crafting time used by the participants in the study.

To evaluate how our system can be used for general-purpose face drawing applications, we also asked the participants to perform a second experiment on generating faces based on their imagination with additional identity information. All of the participants finished this experiment within 40 seconds; and some of the results are shown in Fig. 9. Note that the participants spent significantly less time in the second user study comparing to the first one, because they were not required to carefully draw face shapes to match any specific photo examples. In addition, we performed an empirical validation study to analyze how the resulting face model depends on step-by-step user inputs (Fig. 10 shows one example). As shown in Fig. 10, when different user inputs were provided, the resulting face was progressively refined.

8 Discussion and Conclusions

In this paper, we propose an intuitive face sketching system to craft 3D realistic faces through a portrait drawing interface and automated face texture generation. Without any prior training in pencil sketching, users can craft a quality 3D face model intuitively and efficiently. To automatically maintain the "faceness" of the portrait being drawn, a reduced face description space is dynamically incorporated into our approach. Our face texture generation algorithm automatically paints photorealistic and plausible texture to the portrait based on certain user-provided information such as gender, ethnicity, and skin tone. Our preliminary user study results showed that using our system, a novice user is able to efficiently craft a quality 3D textured face model on average within three minutes.

The limitations of the current work are mainly connected with the used face dataset. The number of subjects within each ethnic group is not well balanced across the dataset, e.g., about 75% of the faces are Whites and Asians, which will potentially lead to certain model bias. Also, in the face texture generation process, due to the existing texture shadow in our face dataset, crafted faces are affected accordingly.

Our current system can be extended to include more information about human face attributes such as age range, eye colors, hair colors, *etc.* Since our system morphs 2D face texture and 3D face model separately, technically, we can employ it to generate only 2D face images without the need of a 3D face dataset. Moreover, instead of generating a static face texture from discrete information, we can integrate slide-like controls to adjust probabilistic combinations of certain face properties and thus provide an interactive face texture generation system. For example, users can provide ethnicity information as 70% White plus 30% Asian.

Not limited to the current simple portrait rendering, in the future we plan to add more varieties of artistic styles into our portrait rendering process. Furthermore, to enhance the visual realism of crafted 3D faces, we will extend our system to handle other parts of the face, *e.g.*, hair, beard, glasses, *etc.* In case the users require to input pencil-drawn sketches to the system, we can potentially employ the active shape model (ASM) to extract the 83 control points from the sketches. However, to ensure the proper alignment of the 83 control points, the users may need to manually adjust the portrait in our system. Also, to thoroughly evaluate the usability of our system, we plan to conduct a large-scale, in-depth user study involved with various levels of users in the future.

Acknowledgements

This work was supported in part by NSF IIS-0914965, Texas NHARP 003652-0058-2007, and NSF CNS-0932272. Tanasai Sucontphunt is supported by a fellowship from the Ministry of Science and Technology, Royal Thai Government. The authors also would like to thank Dr. Lijun Yin for the sharing of his BU-3DFE dataset. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] I. Albrecht, V. Blanz, J. Haber, and H.-P. Seidel. Creating face models from vague mental images. In *ACM SIGGRAPH 2005 Sketches*, 2005.

[2] P. N. Belhumeur, J. ao P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE PAMI*, 19:711–720, 1997.

[3] I. Biometrix. Faces, http://www.iqbiometrix.com/, 2007.

[4] V. Blanz, I. Albrecht, J. Haber, and H.-P. Seidel. Creating face models from vague mental images. *Comput. Graph. Forum*, 25(3):645–654, 2006.

[5] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proc. of SIGGRAPH'99*, pages 187–194, 1999.

[6] D. Bourguignon, R. Chaine, M.-P. Cani, and G. Drettakis. Relief: A modeling by drawing tool. In *Proc. of Eurographics Workshop on Sketch-based Interfaces and Modeling*, pages 151–160, 2004.

[7] P. E. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19:263–311, 1993.

[8] E. Chang and O. Jenkins. Sketching articulation and pose for facial animation. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2006.

[9] D. DeCarlo, D. Metaxas, and M. Stone. An anthropometric face model using variational techniques. In *Proc. of SIGGRAPH '98*, pages 67–74, 1998.

[10] G. M. Draper and P. K. Egbert. A gestural interface to free-form deformation. In *GI'03: Proc. of Graphics Interface'03*, pages 113–120, 2003.

[11] O. Gunnarsson and S.Maddock. Sketching faces. In *Proc. Eurographics Workshop on Sketch-Based Interfaces and Modeling*, 2008.

[12] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *Proc. of SIGGRAPH*, pages 327–340, 2001.

[13] D. D. Hoffman and M. Singh. Salience of visual parts. *Cognition*, 63(1):29–78, 1997.

[14] Identi-It. Identi-it software, http://swat.smithandwesson.com/, 2007.

[15] T. Igarashi and J. F. Hughes. Smooth meshes for sketch-based freeform modeling. In *ACM SIGGRAPH 2006 Courses*, 2006.

[16] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: a sketching interface for 3D freeform design. In *Proc. of SIGGRAPH '99*, pages 409–416, 1999.

[17] ITC-irst. Spotit, http://spotit.itc.it/, 2007.

[18] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *SIGIR '03*, pages 119–126, 2003.

[19] P. Joshi, W. Tien, M. Desbrun, and F. Pighin. Learning controls for blend shape based realistic facial animation. In *SCA'03: Proc. of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 35–42, 2003.

[20] R. Kalnins, L. Markosian, B. Meier, M. Kowalski, J. Lee, P. Davidson, M. Webb, J. Hughes, and A. Finkelstein. WYSIWYG NPR: drawing strokes directly on 3D models. In *Proc. SIGGRAPH 2002*, pages 755–762, 2002.

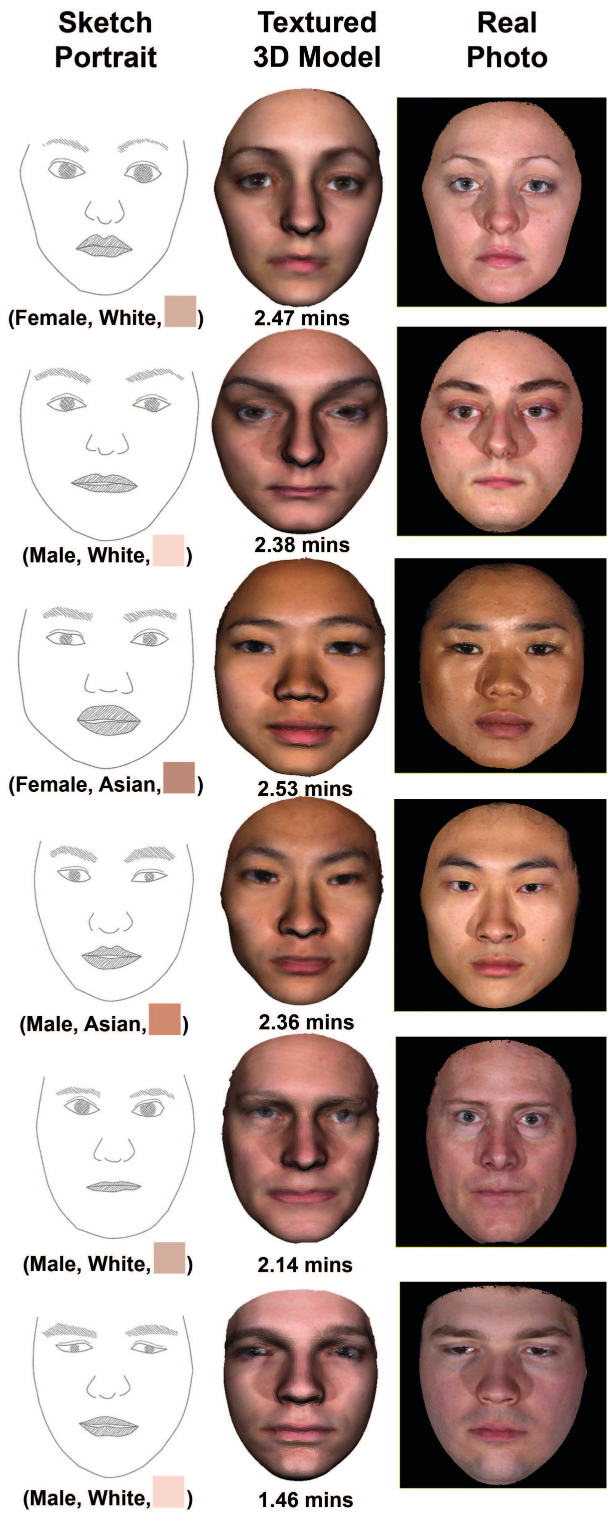

Figure 8: Left: sketch portraits by the participants, middle: 3D textured face models with time usage (only the sketching time), and right: the used reference photos. The additional information is shown in the parenthesis.

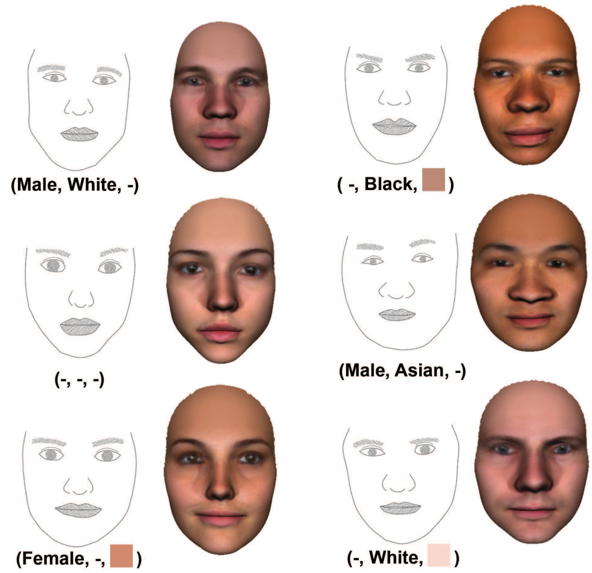

Figure 9: Examples of generated faces based on the users' sketches and partial identity information. The additional information are shown in the parenthesis in the format of (gender, ethnicity, skin tone) where "-" means N/A (not available).

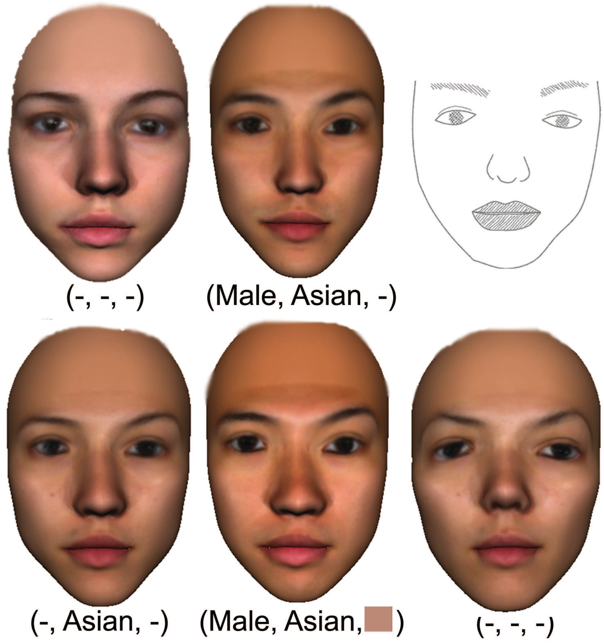

Figure 10: An example of the empirical validation study. The user's inputs are shown in the parenthesis in the form of (gender, ethnicity, skin tone) where "-" represents N/A (not available). In the last column, by changing only the eyes on the sketch (showing on the top), the resulting face texture is also changed accordingly.

[21] O. Karpenko, J. Hughes, and R. Raskar. Free-form sketching with variational implicit surfaces. *Computer Graphics Forum*, 21(3):585–594, 2002.

[22] O. A. Karpenko and J. F. Hughes. Smoothsketch: 3D free-form shapes from complex sketches. *ACM Trans. Graph.*, 25(3):589–598, 2006.

[23] Y. Kho and M. Garland. Sketching mesh deformations. *ACM Trans. Graph.*, 24(3):934–934, 2005.

[24] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *NIPS*. MIT Press, 2003.

[25] W.-S. Lee and A. Soon. Facial shape and 3d skin: Research articles. *Comput. Animat. Virtual Worlds*, 17(3-4):501–512, 2006.

[26] J. Lewis, J. Mooser, Z. Deng, and U. Neumann. Reducing blendshape interference by selected motion attenuation. In *I3DG'05: Proc. of the 2005 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 25–29, 2005.

[27] J. P. Lewis, M. Cordner, and N. Fong. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *Proc. of SIGGRAPH'00*, pages 165–172, 2000.

[28] H. Lipson. Creating geometry from sketch-based input. In *ACM SIGGRAPH 2007 courses*, 2007.

[29] Z. Mo, J. Lewis, and U. Neumann. Face inpainting with local linear representations. In *Proc. of British Machine Vision Conference (BMVC) 04*, 2004.

[30] Z. Mo, J. Lewis, and U. Neumann. Improved automatic caricature by feature normalization and exaggeration. In *SIGGRAPH 2004 Sketches*, 2004.

[31] U. Mohammed, S. J. D. Prince, and J. Kautz. Visio-lization: generating novel facial images. In *Proc. of SIGGRAPH*, pages 1–8, 2009.

[32] G. Nataneli and P. Faloutsos. Sketching facial expressions. In *Proc. of SIGGRAPH 2007 Technical Sketches*, 2007.

[33] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa. Fibermesh: designing freeform surfaces with 3D curves. *ACM Trans. Graph.*, 26(3), 2007.

[34] I. K. Park, H. Zhang, and V. Vezhnevets. Image-based 3d face modeling system. *EURASIP J. Appl. Signal Process.*, 2005(1):2072–2090, 2005.

[35] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. Salesin. Synthesizing realistic facial expressions from photographs. In *Proc. of SIGGRAPH'98*, pages 75–84, 1998.

[36] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin. Synthesizing realistic facial expressions from photographs. In *Proc. ACM SIGGRAPH Courses*, page 19, 2006.

[37] J. Rissanen. A universal prior for integers and estimation by minimum description length. *Ann. of Statist.*, 11(2):416–431, 1983.

[38] D. T. Sandwell. Biharmonic spline interpolation of geos-3 and seasat altimeter data. *Geophys. Res. Lett.*, 14:139–142, February 1987.

[39] D. Sharon and M. van de Panne. Constellation models for sketch recognition. In *Proc. Eurographics workshop on sketch based interfaces and modeling*, 2006.

[40] K. Singh and E. Fiume. Wires: A geometric deformation technique. In *Proc. of SIGGRAPH 98*, pages 405–414, 1998.

[41] T. Sucontphunt, Z. Deng, and U. Neumann. Crafting personalized facial avatars using editable portrait and photograph example. In *VR*, pages 259–260, 2009.

[42] T. Sucontphunt, Z. Mo, U. Neumann, and Z. Deng. Interactive 3D facial expression posing through 2d portrait manipulation. In *GI '08: Proceedings of graphics interface 2008*, pages 177–184, 2008.

[43] R. W. Sumner and J. Popović. Deformation transfer for triangle meshes. *ACM Trans. Graph.*, 23(3):399–405, 2004.

[44] N. Tsumura, N. Ojima, K. Sato, M. Shiraishi, H. Shimizu, H. Nabeshima, S. Akazaki, K. Hori, and Y. Miyake. Image-based skin color and texture analysis/synthesis by extracting hemoglobin and melanin information in the skin. In *ACM SIGGRAPH 2003*, pages 770–779, 2003.

[45] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *CVPR'91*, pages 586–591, 1991.

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

[47] X. Wang and X. Tang. Face photo-sketch synthesis and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):1955–1967, 2009.

[48] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato. A 3d facial expression database for facial behavior research. In *FGR '06*, pages 211–216, 2006.

[49] L. Yin and K. Weiss. Generating 3d views of facial expressions from frontal face video based on topographic analysis. In *Proc. of ACM MULTIMEDIA '04*, pages 360–363, 2004.