3D Facial Surface and Texture Synthesis Using 2D Landmarks From A Single Face Sketch

Tanasai Sucontphunt and Ulrich Neumann
University of Southern California
{sucontph | uneumann}@graphics.usc.edu

Abstract

This paper presents an approach to construct a 3D face model from a sparse set of 2D facial landmarks. This system takes the 2D landmarks extracted from a 2D portrait sketch image of an artist creation and estimates their corresponding 3D landmarks by using a statistical estimation. Then, the 3D surface and its texture are synthesized to fit to the 3D landmarks and its facial descriptions by using an example-based surface and texture synthesis technique. The 3D surface and its texture can also be further altered by a surface and texture blending technique which is particular useful for cartoon face modeling. The evaluation shows that the reconstructed 3D faces are highly similar to the ground-truth examples. This approach can be used in many face modeling applications such as in 3D avatar creation, cartoon face modeling, and police investigations.

1. Introduction

During the past several decades, a rich variety of techniques have been proposed to automatically create an individual-specific 3D face model. Besides expensive 3D scanning systems, numerous vision-based live capturing techniques (e.g., [1]) can create photorealistic 3D face models using sophisticated equipments. Notably, Blanz et al. [4] pioneered a 3D morphable face model by building truncated PCA spaces for 3D face geometry and texture based on a set of scanned 3D faces. This core technique has been extended for 3D face reconstruction from a single photograph (e.g., [10, 5]). However, photographs of the target person are not always available in many scenarios such as creation of fictional characters or creation of criminal suspect faces. When face photographs are unavailable, a 2D face drawing is a natural alternative. Although the 2D face drawing is often limited to the frontal view without color texture, it provides effective information for people to identify specific individual.

Motivated by the above applications and need, we look into the following interesting research problem: *Can we create a plausible textured 3D face model solely based on a single 2D face sketch input?* The problem is challenging since it is essentially ill-posed and underconstrained. Indeed, to date, few if any previous approaches have been reported to tackle it. In this paper, we propose a novel approach to create such textured 3D face models based on single 2D face sketch input. The core idea of our approach can be summarized as follows: (1) Based on a set of 2D facial landmarks that are extracted from the input sketch, their corresponding 3D facial landmarks are predicted by our probabilistic inference algorithm. (2) Based on the sparse set of the predicted 3D facial landmarks, we adapt the MeshIK framework [19] to construct the most plausible 3D face surface from a pool of example faces. (3) In order to generate a plausible facial texture, the predicted 3D facial landmarks together with the user-provided gender, ethnicity, and skin tone information are utilized in our texture search engine scheme. (4) The synthesized 3D face surface and texture can be further adjusted by a gradient blending technique which is particularly useful for cartoon face modeling. Finally, our evaluation results show that our proposed approach can effectively produce textured 3D faces that are measurably close to the ground-truth face examples.

2. Related Work

Most of existing face modeling efforts create realistic 3D face models by learning statistical human face knowledge from a pre-collected face dataset. At the early time, Decarlo *et al.* [8] proposed a data-driven approach to generate 3D face models by imposing facial anthropometry statistics as constraints. However, the 3D faces generated by their approach is not sufficiently realistic since the used anthropometric constraints are too sparse and general. Blanz and colleagues proposed various methods to create 3D faces from vague information [2] or a sparse set of the facial landmarks [3], based on their well-known morphable face model [4].

Besides the above 3D face modeling, researchers designed various methods to efficiently edit 3D face poses and expressions. For example, puppetry methods have been developed to directly control face poses and expressions based on the live performance of an actor/actress or selective control points [12, 11, 23]. Various sketching interfaces [6, 9, 16] have also been proposed to support different drawing strokes and directly edit 3D face models by exploiting a pre-recorded facial motion database. In these works, user sketching inputs are used to search for and optimize the resultant 3D face pose. In addition, Wang *et al.* [22] use the characteristics of face sketches (by a specific artist) and their corresponding photographs in order to synthesize a color texture for given novel 2D face sketch.

None of the above techniques constructs a photo-realistic 3D face model from a single 2D face drawing, partially due to the fact that reconstructing a 3D model from a 2D drawing is a well-known, under-constrained problem. In the past, researchers developed several successful 2D-to-3D estimation techniques including Analysis-By-Synthesis (ABS) [5] and Shape From Shading (SFS) [15]. However, compared with a 2D color/shade image, a 2D face sketch often only contains conceptual lines of the facial structure without a color or grayscale texture that is required by these 2D-to-3D estimation techniques [5, 15].

Recently, Sucontphunt *et al.* [17] presented a 2D sketching interface for a 3D face modeling. Their approach allows a user to interactively draw a human face sketch and the sketch is used to infer a corresponding textured 3D face. However, their approach is limited to the use of modeling normal human faces and the inputted landmarks are required to be reshaped to match a normal human face shape in order to function properly. By contrast, the approach proposed in this paper follows a different technical path; it can take any single 2D face sketches as the input and synthesize not only 3D textured normal human faces but also various 3D textured identity-embedded cartoon faces efficiently.

3. Approach Overview

As schematically illustrated in Fig. 1, our approach consists of the following three main components, namely, *3D facial landmark estimation*, *3D face surface synthesis*, and *face texture synthesis*.

(1) **3D facial landmark estimation**: in this work, the input 2D face sketch is approximated as 2D facial landmarks that have been widely used as a common template in face detection applications. The 2D facial landmarks can be automatically detected by algorithms or manually labeled. Corresponding 3D facial landmarks are estimated from these obtained 2D facial landmarks by a statistical inference method, inspired by the work of [3]. The main distinction between our approach and [3] is, in our approach, the perspective distortions of the 2D landmarks are used as a clue to estimate their depths while [3] work bases on an orthogonal projection assumption. Also, the surface in our approach is represented by affine transformation matrices instead of vertex locations to maintain the smooth normal transition without the need of a heuristic regularization as in [3]. The 3D landmark estimation and 3D face surface synthesis are also decoupled in our approach to prevent an over-fitting problem. This separation also allows the surface to be altered later.

(2) **3D face surface synthesis**: from the obtained 3D facial landmarks, a 3D face surface is synthesized by an example-based non-linear interpolation technique.

(3) **Face texture synthesis**: we follow artists' practices to color a 2D face sketch. Basically, an artist typically paints a 2D portrait/sketch based on the implicit assumption of its facial descriptions, i.e. gender, ethnicity, and skin tone in his/her mind. As such, in this work, we formulate it as a search-and-synthesis problem by first searching for most proper textures in a library and then synthesizing a novel face texture.

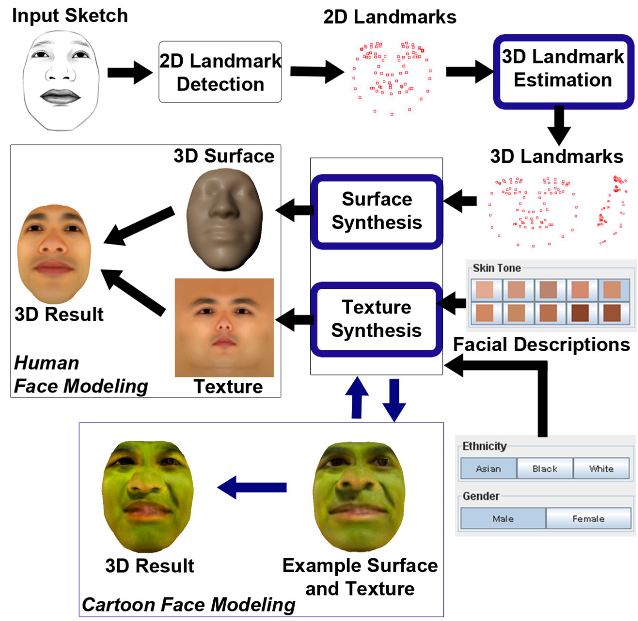

Figure 1. Schematic overview of our approach. It consists of the following three main components, namely, 3D facial landmark estimation, 3D face surface synthesis, and face texture synthesis.

Apart from normal human face, the system can synthesize a cartoon face based on an input cartoon example. In the cartoon face modeling, gradient blending techniques are employed to alter the 3D face surface and texture towards any cartoon face example.

4. Data Preparation

The 3D face dataset [24] containing 100 subjects of 56 females and 44 males with diversified ethnicities is used to construct a prior knowledge of the human face structure. Each 3D face contains a 3D face mesh, texture, and 83 facial landmarks. A deformation transfer technique [18] is used to create strict correspondences among all the face meshes to a face template. We selected the topology of 2043 vertices, 6042 edges, 4000 faces, and 12000 UV coordinates as the face template.

Eigen-vectors of 3D facial landmarks, α_{LM}: Eigen-vectors of 3D facial landmarks are constructed from X, Y, and Z coordinates of the 83 facial landmarks of all the faces in the dataset. The eigen-vectors are calculated by the Principal Component Analysis (PCA) technique. In this work, the PCA subspace spanned by the retained eigen-vectors contains more than 95% of the variations. The retained eigen-vectors will be used to estimate the depths of 2D landmarks at the 3D facial landmark estimation step. Also, we divide the 3D landmarks into different regions ($\alpha_{LM_{Regional}}$): the eyes, eyebrows, nose, mouth and outline, which will be used during the face texture synthesis step.

Eigen-vectors of deformation gradients (DG), α_{DG}: The surface of each 3D face is encoded as DG [18] which is 3D affine transformation matrices between the surface and an average human face. Then, similar to the construction of α_{LM}, PCA is used to construct eigen-vectors from DG of all the faces in the dataset. One important issue is that affine transformation matrices cannot be linearly interpolated, while coefficients in a PCA space are expected to be linearly interpolatable. Thus, we first decompose an affine transformation matrix (A) to a rotation component, R, and a scaling component, S, using polar decomposition (i.e., $A = exp(log(R)) \times S$) as suggested in [19] and construct eigen-vectors for each component. The α_{DG} is also generated separately for each gender and ethnicity. These extracted eigen-vectors will be used during the face surface synthesis step.

Skin tone likelihood: We cropped a part of the forehead region of the facial texture to represent a skin tone. Then, Gaussian Mixture Models (GMMs) are used to calculate the skin tone likelihood by first clustering similar skin tones in the dataset into 10 multivariate Gaussian distribution groups. These groups' centroids are also used as the skin-tone bases for users to select, as shown in Fig. 1. Finally, a skin tone likelihood table ($\psi_{Texture}$) is constructed by calculating the likelihood of each face texture based on its distance to the centroids of all the skin-tone bases. The likelihood table will be used at the face texture synthesis step.

5. 3D Facial Landmark Estimation

Each detected 2D facial landmark consists of X and Y coordinates in an image space. To estimate the 3D landmarks, the Z coordinate (depth) of each 2D landmark needs to be reconstructed. If we directly use the X and Y coordinates of the 2D landmarks as the X and Y components of the 3D landmarks and just fill up the Z value, the 3D landmarks will be visually distorted and result in producing the distorted 3D surface (Fig. 2). This is because the 2D landmarks in the image space are already perspectively distorted in the rendering process. This perspective projection property is highly beneficial in inferring the depth information. Therefore, to properly estimate the 3D landmarks, the perspective projection needs to be taken into consideration when estimating the depths of the landmarks.

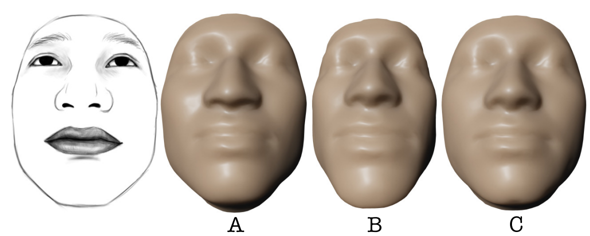

Figure 2. The input sketch is showing on the left. A: the 3D ground-truth model of the sketch. B: the reconstructed 3D face from the parallel projection. C: the reconstructed 3D face from the perspective projection.

We assume the artist's vision is a perfect camera and the center of projection is the middle of the 2D face sketch. In this way, the perspective projection of the 3D landmarks, $\hat{u}$ and $\hat{v}$, can be derived by the Eq. 1.

$$\begin{aligned} \hat{u} &= \tfrac{x.d}{z} \\ \hat{v} &= \tfrac{y.d}{z} \end{aligned} \tag{1}$$

Here, x, y, and z are 3D landmarks in the model space, and d is the focal length. In this work, we set the d to be 22 millimeters according to the average human eyes' focal length. Assuming the detected 2D landmarks from the sketch image are u and v, we can formulate the 3D facial landmark estimation problem as the following energy optimization problem.

$$\begin{gathered} \arg\ \min_{\hat{u},\hat{v}}\{\textstyle\sum_i(\|u_i - \hat{u}_i\|^2 + \|v_i - \hat{v}_i\|^2)\} \\ or \\ \arg\ \min_{x,y,z}\{\textstyle\sum_i(\|u_i - \tfrac{x_i.d}{z_i}\|^2 + \|v_i - \tfrac{y_i.d}{z_i}\|^2)\} \end{gathered} \tag{2}$$

To solve the above Eq. 2, x, y, and z are initialized with the average 3D landmarks calculated from the dataset. Then, at each iteration, x and y are updated by: $x_i \rightarrow x_i - \frac{\alpha_{x i}.z_i}{d}$ and $y_i \rightarrow y_i - \frac{\alpha_{y i}.z_i}{d}$, where α_x and α_y are the residues from the first and the second components of Eq. 2, respectively. To update the z, based on the observation that z is correlated with x and y as encoded by PCA of the 3D landmarks, PPCA algorithm [20] is employed to estimate z from the x and y by using the eigen-vectors of 3D landmarks (α_{LM}). Basically, PPCA estimation is achieved by assuming the Gaussian distribution over the PCA subspace and using the likelihood of the 3D landmarks in the subspace to optimize the objective function by EM algorithm. In this work, the objective function is a lower-bound of the log-likelihood as shown in Eq. 3.

$$\frac{D}{2}(1+\log\sigma^2)+\frac{1}{2}(Tr\{\Sigma\}-\log|\Sigma|)+\frac{1}{2}\|f\|^2-\frac{D_h}{2}\log\sigma^2_{prev} \tag{3}$$

Here D is the number of landmarks multiplied by 3 (for x, y and z), f is the coefficients of the 3D landmarks in PCA space constructed from the current x, y, and treating z as a missing value by initialized it with 0, σ^2 is the variance of reconstructed landmarks from f, σ^2_{prev} is the variance from the previous iteration, Σ is the covariance matrix of f (i.e., calculated from α_{LM} and σ^2_{prev}), and D_h is the number of landmarks. In each iteration, the E-step is performed on f which updating the z value, and the M-step is performed on σ^2. The optimizations of both Eq. 2 and Eq. 3 are repeated until they converge (i.e., the energy difference is less than a specified threshold).

6. 3D Face Surface Synthesis

To construct a 3D face surface, a set of DGs is used to craft the surface to fit to the 3D landmarks. Beside the human surface, our approach can synthesize an artistic style surface using in a cartoon face modeling.

6.1. Human Surface Generation

From the estimated 3D landmarks, a plausible human face surface can be molded from a set of DG examples. Specifically, MeshIK [19] is employed to interpolate the DGs of human face examples constraining by the 3D landmarks. Instead of using DGs directly from the human surface dataset, the new set of DGs is constructed from the major principle components of α_{DG} (constructed from Section 4) to cover a proper variation of human surfaces. Basically, these new DG examples are constructed from each eigen-vector in the α_{DG} which represents a major variance across the whole DG dataset. Each eigen-vector is also orthogonal to each other providing a proper span for the interpolation. These DG examples are called eigen-surfaces in this work.

An eigen-surface is constructed by adding the mean DG to a retained eigen-vector α_{DG}. In other words, $EigenSurface_i = \alpha_{DG}(i) + \mu_{DG}$. Note that the two components ($log(R)$ and S) are processed separately. Finally, these DGs are then used in the MeshIK framework. Figure 3 shows ten constructed eigen-surfaces (in a descending order by their eigen-values).

Figure 3. Ten constructed eigen-surfaces ordered by their eigen-values in a descending order from left to right (i.e. the top row is the first five highest eigen-value shapes). These surfaces are exaggerated by a scaling factor (=30) to visualize their deformation directions from the average surface showing in the left-most box.

The MeshIK is a least-square problem as shown in Eq. 4 and it can be solved by a sparse matrix solver. The set of DG examples, which are affine transformation matrices, are transferred according to their weighted sum by taking the 3D landmarks as constraints.

$$\arg\min_x \|Tx - E - S\| \tag{4}$$

Here, T is a linear operator constructed from an average face surface, x is the vector containing the resultant target vertex locations, S is the 3D landmarks (as constraints), and E is the set of DG examples (i.e., the collection of the obtained eigen-surfaces). Since each DG is the multiplication of R and S, the interpolation required to be solved non-linearly. This non-linear interpolation can be effectively solved by Gauss-Newton iterative algorithm.

6.2. Cartoon Surface Generation

In certain applications such as a cartoon face modeling, the face surface needs to be crafted in an artistic-style rather than a typical human face. The artistic surface can then be added by providing the artistic surface to the system. To avoid confusion, the human surface is abbreviated as the **HI-face** (human identity face) and the cartoon surface face is abbreviated as the **CS-face** (cartoon style face).

There are 2 options to synthesize a cartoon face surface: (1) Using only the CS-face in Eq. 4 as E to morph to the 3D landmarks (S). (2) Using both CS-face and HI-face synthesized in Section 6.1. The first choice is straightforward but the surface variations are limited to the CS-face as shown in Figure 4. The second choice produces more surface variation by combining the synthesized human surface as shown in Figure 6. For the second choice, to combine the HI-face and CS-face together, a surface blending technique is required. In a blending algorithm, the surface synthesis algorithm needs to take both the HI-face and the CS-face into consideration. This problem can be solved in the MeshIK framework by the following least-square equation (Eq. 5).

$$\arg_x \min \|(T_h + T_c)x - A_h - A_c - S\| \tag{5}$$

Here, T_h and T_c are linear operators constructed from the HI-face and CS-face respectively, A_h and A_c are the affine transformations of the HI-face and CS-face respectively; and x is the vector containing the vertex positions of the resultant identity-embodied cartoon face; again, S is the 3D landmarks setting as constraints. This blending technique basically compensates between fitting the surface to HI-face and adding CS-face while maintaining the surface smoothness.

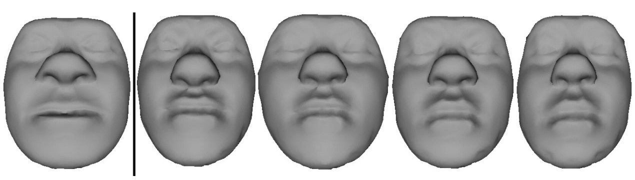

Figure 4. The surface synthesis with choice (1): using 3D landmarks to morph the CS-face directly. The left-most model is the CS-face and the rest from left to right are the four synthesized surfaces from the face in 1-4 rows respectively from Figure 6. The surfaces are very similar to each other. On the other hand, choice (2) which blends the human surfaces together yields higher surface variations as shown in "the cartoon result 1" column of Figure 6.

7. Face Texture Synthesis

To synthesize a facial texture, the gradient transfer technique similar to the one used in the surface synthesis (Section 6) is employed but in the color space instead of the 3D space.

7.1. Human Texture Generation

Firstly, a texture search engine scheme is employed to search for appropriate textures in the dataset. The estimated 3D landmarks and the selected skin tone are used as the search criterion with an assumption that similar textures share a similar landmark structure and skin tone. To properly search for the candidate textures, the searching is performed only within a specified group of genders (male or female) and ethnicities (European, Black, and Asian) upon the user's inputs. Finally, a joint probability combines these two searching criterion together as shown in Eq. 6.

$$P(LM, T|I) = P(LM|I) \times P(T|I) \tag{6}$$

Where I is the face candidate in the dataset (according to the selected gender and ethnicity) to be ranked, $P(LM|I)$ is the similarity score between the estimated 3D landmarks and I's 3D landmarks measured by their Euclidean distance in α_{LM} space, and $P(T|I)$ is the likelihood score in $\psi_{Texture}$ table from the selected skin tone to the skin tone of I. The highest score texture is then used as the facial texture candidate. Also, to provide more facial texture variations, the search is performed separately on each of the various facial regions which include the eyes, eyebrows, nose, mouth, and facial outline. Thus, each facial region will have its own texture candidate.

Secondly, to be able to smoothly combine all regional textures together without producing artifacts, the Poisson Image Editing [13] gradient transfer technique is employed as shown. To preserve a global continuity between regions, the texture of the outline region is picked as the base texture and the other regional textures will be transferred to it. Figure 5 shows an example of a texture synthesis resulting from the combining of five facial regions from five texture candidates.

7.2. Cartoon Texture Generation

Similar to the surface synthesis part (Section 6.2), the texture also can be created in an artistic style from the provided artistic texture example. In this work, the Poisson image editing technique is then modified to capture the texture gradient between the HI-face (from Section 7.1) and an average human texture and to transfer the gradient to the CS-face texture (the base texture). Note that, we transfer not only the texture gradient within the HI-face texture but also the gradient between the HI-face texture and an average human face texture; thus, the vector field representation is expanded accordingly.

Since the CS-face texture can be in any arbitrary color and not only limited to the reddish-base-color of

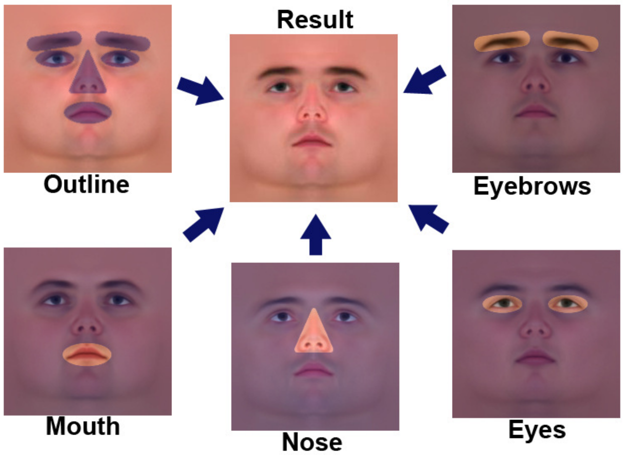

Figure 5. The resultant synthesized texture is shown in the middle. From the candidate textures, they are combined to the base texture (an outline region). Each candidate texture is represented by the unmarked texture area.

the human skin texture, there is an alternative way to blend the human texture to the cartoon texture without reveal the human reddish colors. In this alternative, HSV color space is used instead of the RGB color space to maintain the CS-face base-color. The tone channels (S and V) of HSV are the only channels to be transferred to the cartoon texture while the values in the Hue channel are kept intact. This is because the Hue channel represents the base-color of the CS-face while the S and V channels represents only the variation of the base-color. This will produce the texture that contains an appearance of the CS-face while only the tone variations are modified according to the HI-face.

8. Result

We conducted an experiment to evaluate our synthesized human faces and to illustrate the usefulness of the cartoon face modeling. The thirty ground-truth face models are generated from FaceGen [14] to create the novel 3D faces outside the dataset. From these ground-truth models, a sketch image of each model is drawn by an artist from its rendered images. These sketch images are then used as the main input to our system. Firstly, an ASM based detection [21] is used to roughly extract 2D facial landmarks. Then, the landmarks are manually adjusted to fit to the sketch image via a simple user interface. Also, the facial descriptions of the faces are manually labeled to reflect their descriptions. Figure 6 shows 6 examples of 30 examples generated from our system comparing their face sketch images, the ground-truth models, the synthesized human faces, and the synthesized cartoon faces.

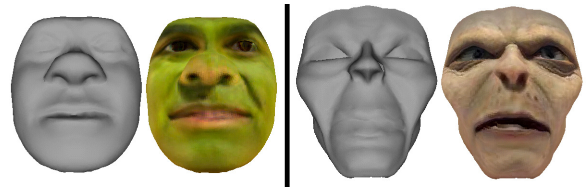

Figure 7. Two cartoon surface and texture examples are used in the experiment for the cartoon result 1 (the left cartoon face) and the cartoon result 2 (the right cartoon face) in Figure 6.

Figure 8-(A) visualizes the human surface reconstruction errors (averaged over the thirty face examples) by computing the Euclidean distances between the ground-truth and the synthesized human surfaces by our approach. Figure 8-(B) compares our approach with the Radial Basis Functions (RBF) based technique [17] where the 3D landmarks are used as the control points to deform a face surface. Our approach produces a higher accuracy than the RBF based technique since more variety of face surface examples are utilized to generate the target face surface.

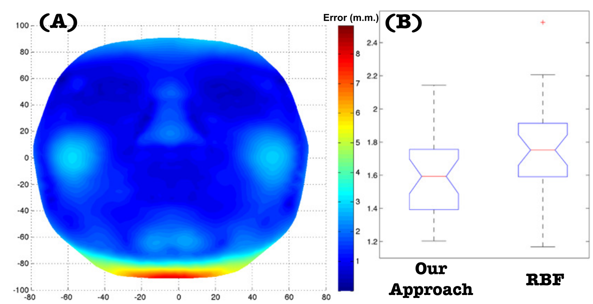

Figure 8. (A) The visualized surface reconstruction errors that are the Euclidean distances between each reconstructed vertex to its ground-truth vertex (averaged over thirty face examples). (B) Reconstruction error comparison between our approach and the RBF based technique [17]. On average, our approach produces less than 1.6 m.m. in errors over all the surfaces.

9. Conclusion

This work presents a technique to construct a 3D face model from a sparse set of facial landmarks of a portrait sketch image. From a sketch image, the facial landmark depths are estimated in the perspective view. Then, 3D meshes as well as a plausible texture are synthesized to fit the landmarks automatically. To extend the surface and texture generation, the surface and texture blending techniques are developed for a cartoon face modeling as well. The evaluation shows that our system performs well even when using only 83 landmarks to reconstruct the 3D face. This work

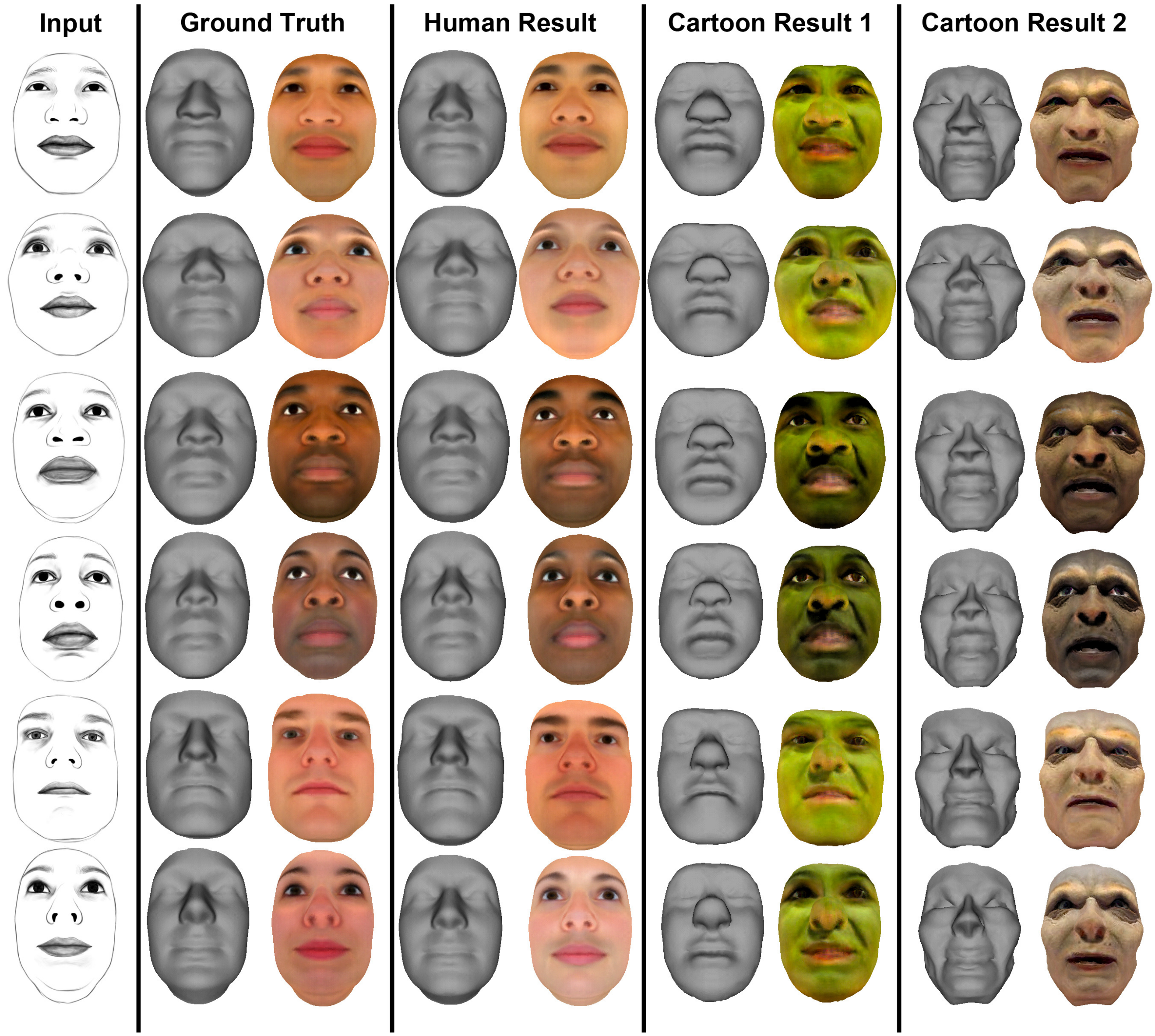

Figure 6. Example results from the system. The left-most column shows input sketches drawn by an artist. The next column shows the ground-truth models (surface and textured face respectively). The human result column shows the 3D human results generated by our system. The cartoon result 1 and the cartoon result 2 columns show the 3D cartoon results generated by our system. The CS-faces used in these two columns are the cartoon faces from Figure 7.

can be used in a number of applications in various industries such in the entertainment industry and in law enforcement.

One of the limitations of this work is that it can only synthesize a neutral facial expression. To overcome this issue, the dataset can be extended to cover the other 3D facial expressions. Also, the number of face models (100 faces) in the dataset is relatively small, which may not be sufficient to fit a probabilistic model (the Gaussian distribution assumption). Besides, this work only focuses on a frontal portrait input; thus, a user interface can be developed for adjusting the surface from many other views. Moreover, many portrait sketches normally contain more hatching and shading information such as wrinkles, ridges, and valleys, than those comprising of salient facial feature lines. This information, though not as important as the salient facial feature lines, can produce a more realistic result. So, the inverse-NPR can be explored to make use of this information in the future. For the cartoon face modeling, we plan to explore other intuitive user interfaces such as a suggestive user interface [7] that can suggests plausible cartoon models to a user from an example cartoon set. Finally, our work can be further extended to other 3D modeling domains where their 3D models contain specific patterns.

Acknowledgment

We thank Zhigang Deng and Polrat Thipayanusorn for help in reviewing the paper to the current version.

References

[1] T. Beeler, F. Hahn, D. Bradley, B. Bickel, P. Beardsley, C. Gotsman, R. W. Sumner, and M. Gross. High-quality passive facial performance capture using anchor frames. *ACM Trans. Graph.*, 30:75:1–75:10, August 2011. 1

[2] 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. 2

[3] V. Blanz, A. Mehl, T. Vetter, and H.-P. Seidel. A statistical method for robust 3d surface reconstruction from sparse data. In *Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium*, 3DPVT '04, pages 293–300, Washington, DC, USA, 2004. IEEE Computer Society. 2

[4] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proc. of SIGGRAPH'99*, pages 187–194, 1999. 1, 2

[5] P. Breuer, K. in Kim, W. Kienzle, B. Schölkopf, and V. Blanz. Automatic 3d face reconstruction from single images or video. In *Automatic Face & Gesture Recognition, 2008.*, 2008. 1, 2

[6] E. Chang and O. Jenkins. Sketching articulation and pose for facial animation. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2006. 2

[7] S. Chaudhuri, E. Kalogerakis, L. Guibas, and V. Koltun. Probabilistic reasoning for assembly-based 3D modeling. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 30(4), 2011. 7

[8] D. DeCarlo, D. Metaxas, and M. Stone. An anthropometric face model using variational techniques. In *Proc. of SIGGRAPH '98*, pages 67–74, 1998. 1

[9] O. Gunnarsson and S. Maddock. Sketching faces. In *Proc. Eurographics Workshop on Sketch-Based Interfaces and Modeling*, 2008. 2

[10] J. Lee, J. Lee, B. Moghaddam, B. Moghaddam, H. Pfister, H. Pfister, R. Machiraju, and R. Machiraju. A bilinear illumination model for robust face recognition. In *ICCV05*, pages 1177–1184, 2005. 1

[11] H. Li, T. Weise, and M. Pauly. Example-based facial rigging. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2010)*, 29(3), July 2010. 2

[12] M. Meyer and J. Anderson. Key point subspace acceleration and soft caching. In *ACM SIGGRAPH 2007 papers*, SIGGRAPH '07, New York, NY, USA, 2007. ACM. 2

[13] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, 2003. 5

[14] Singular-Inversions. Facegen, http://www.facegen.com/, 2007. 6

[15] W. A. Smith and E. R. Hancock. Facial shape-from-shading and recognition using principal geodesic analysis and robust statistics. *Int. J. Comput. Vision*, 76:71–91, January 2008. 2

[16] 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. 2

[17] T. Sucontphunt, B. Tunwattanapong, Z. Deng, and U. Neumann. Crafting 3d faces using free form portrait sketching and plausible texture inference. In *GI'10: Proc. of Graphics Interface*, Ottawa, Ontario, Canada, 2010. 2, 6

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

[19] R. W. Sumner, M. Zwicker, C. Gotsman, and J. Popović. Mesh-based inverse kinematics. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 488–495. ACM, 2005. 1, 3, 4

[20] M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Comput.*, 11(2):443–482, 1999. 4

[21] B. van Ginneken, A. F. Frangi, J. J. Staal, B. M. T. H. Romeny, and M. A. Viergever. Active shape model segmentation with optimal features. *IEEE Transactions on Medical Imaging*, 21:924–933, 2002. 6

[22] X. Wang and X. Tang. Face photo-sketch synthesis and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):1955–1967, 2009. 2

[23] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. In *ACM SIGGRAPH 2011 papers*, SIGGRAPH '11, pages 77:1–77:10, New York, NY, USA, 2011. ACM. 2

[24] 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. 3