

Research Article

A Practical Approach for Identity-Embodied 3D Artistic Face Modeling

Tanasai Sucontphunt

Graduate School of Applied Statistics, Navamindradhiraj Building, Floor No. 12, 118 Moo 3, Serithai Road, Klong-Chan, Bangkok 10240, Thailand

Correspondence should be addressed to Tanasai Sucontphunt; tanasai@gmail.com

Received 28 February 2014; Revised 28 May 2014; Accepted 6 June 2014; Published 24 June 2014

Academic Editor: Alexander Pasko

Copyright © 2014 Tanasai Sucontphunt. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper describes a practical technique for 3D artistic face modeling where a human identity can be inserted into a 3D artistic face. This approach can automatically extract the human identity from a 3D human face model and then transfer it to a 3D artistic face model in a controllable manner. Its core idea is to construct a face geometry space and a face texture space based on a precollected 3D face dataset. Then, these spaces are used to extract and blend the face models together based on their facial identities and styles. This approach can enable a novice user to interactively generate various artistic faces quickly using a slider control. Also, it can run in real-time on an off-the-shelf computer without GPU acceleration. This approach can be broadly used in various 3D artistic face modeling applications such as a rapid creation of a cartoon crowd with different cartoon characters.

1. Introduction

Modeling a 3D artistic face embodied with a specific characteristic is a nontrivial task even for skilled artists. Existing face crafting techniques that often manipulate, deform, or exaggerate an initial face model can be potentially used for this purpose. However, these methods typically require users to adjust internal parameters of the 3D face or to learn statistical models from a set of artist-drawn examples. Besides the painstaking trial-and-error efforts required by these methods, it is very difficult, if not impossible, for the users to easily control the level of the facial identity during the sculpting process.

Commercial applications provide tools for 3D face avatar modeling along other directions, such as asking users to manually compose facial components from a list of prebuilt templates (e.g., Second Life) or to linearly morph between a human face and a monster face (e.g., <http://www.evolver.com/>). Nevertheless, to the best of our knowledge, all these approaches do not provide users a simple control to adjust the level of the human identity while trying to maintain the original artistic style automatically.

In this paper, we present a practical approach for identity-embodied 3D artistic face modeling through automated

human identity blending algorithms. In this work, the human identity is referred to PCA coefficients from a projection of a human face model in a PCA subspace (which constructed from a collection of human faces). The cartoon style is referred to PCA coefficients from a projection of a cartoon face model plus the residues outside the PCA subspace (where residues = a cartoon face model – a reconstructed PCA projection). The residues are added to the cartoon style because, normally, a cartoon face contains many important facial features (styles) falling off the human's PCA subspace range.

As illustrated in Figure 2, this automatic system (shown in Figure 1) consists of the following three main steps. (i) Given an input 3D human face (called the *human identity* face or further abbreviated as the **HI-face** in this writing) and an input 3D artistic face (called the *artistic style* face or further abbreviated as the **AS-face**), both the HI-face and AS-face are projected to reduced PCA subspaces. (ii) Users can interactively adjust a slider to control the embodied levels of the HI-face on the AS-face by our introduced identity engineering (IE) algorithm. (iii) A resultant 3D artistic face model is automatically constructed by our 3D surface and texture synthesis algorithms. Through our experiment,



FIGURE 1: The proposed approach can automatically transfer the human identity of a human face to a provided 3D artistic face while providing a simple high-level control to users.

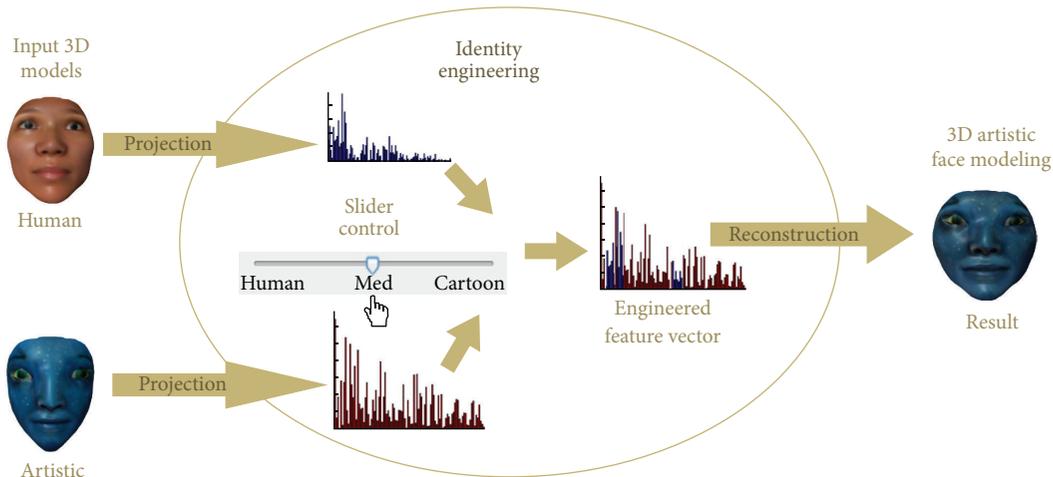


FIGURE 2: The schematic overview of our approach.

we found that our approach can synthesize various desired identity-embodied face characters while keeping artistic styles intact.

The remainder of this paper is organized as follows. Section 2 briefly reviews recent efforts most related to this work. Section 3 describes the offline data processing step in our approach. Section 4 describes the core algorithms used in this work. Section 5 details the geometric deformation and texture synthesis processes. Experimental results of our approach are reported in Section 6. Finally, limitations, future directions, and concluding remarks are presented in Section 7.

2. Related Work

Comprehensively reviewing various previous efforts on facial modeling and animation is clearly beyond the scope of this paper. Interested readers can refer to the recent survey [1]. In this section, we briefly review major efforts in two relevant research areas: cartoon/caricature modeling and 3D face modeling/deformation.

2.1. Cartoon/Caricature Modeling. Our work focuses on a cartoon face modeling. However, it shares several common features with a caricature modeling since they both create a face that reflects a human identity. While a caricature modeling attempts to exaggerate a human face identity for

better communication [2], a cartoon face modeling attempts to keep an artistic style to emphasize its cartoon character.

As the earliest documented effort on computer generated caricature, Brennan [3] generates a 2D caricature sketch by exaggerating (scaling) the whole face drawing with respect to the average human face. Later, Koshimizu et al. [4] present a similar PICASSO system to produce a 2D image-based caricature. Hsu and Jain [5] use interactive snakes algorithm to detect facial components and then generate a 2D caricature sketch by scaling its difference from the average face. Mo et al. [6] generate a 2D caricature sketch by exaggerating each facial part based on its standard deviation. In recent years, example-based caricature generation techniques have been proposed to learn the drawing style of artists by using various statistics and machine learning algorithms, for example, partial least-squares based learning [7], eigen-space mapping [8], mean value coordinates [9], and golden ratio distance [10]. In addition, Akleman and Reisch [11] suggest a general five-step procedure to manually craft a stylish 3D caricature. From precollected art work portrait examples, many data-driven approaches [12–16] generate an art work portrait purely from a novel photograph. These works mainly capture a relation of an artistic style from an art work and its photograph from many examples to recreate a novel art work. Recently, Berger et al. [16] synthesize various artistic styles as well as abstraction levels of portrait sketches using a collection of artist art works. However, none of these approaches offers

a tool to modify an identity of an artistic face while keeping the artistic style as intact as possible. Besides, users can also introduce any novel artistic faces to our system without limiting to the predefined styles as in the other approaches.

2.2. 3D Face Modeling/Deformation. Early efforts on a 3D face modeling are typically based on multiple images [17]. By acquiring continuous surface deformation details, recent performance capture advances can produce high-fidelity 3D face mesh sequences based on the performance of a human subject, including multiresolution capture [18], stereo cameras-based capture [19, 20], and structured light capture [21]. Researchers have also developed a variety of algorithms, interfaces, and tools to allow users to directly manipulate 3D facial mesh models [22, 23]. Recent data-driven facial editing and deformation algorithms utilize the statistical correlations in precollected facial datasets [24–28]. Notably, Blanz and Vetter [29] build a 3D morphable face model by constructing principal component analysis (PCA) spaces from the 3D geometry and texture of a scanned 3D face dataset and demonstrate that the constructed face PCA spaces can be used for various face synthesis and editing applications. Besides, the puppetry methods have been developed to directly control face poses based on the live performance of an actor/actress [30–34]. However, all these methods are focused on modeling realistic 3D models for normal human faces, and none of them can automatically transfer or embed the identity of a human face to a 3D artistic face. Recently, Sucontphunt and Neumann [35] synthesize a 3D artistic face by combining facial feature points from a sketch portrait with an artistic face example’s surface and texture. However, this work purely deforms an artistic face example to fit the facial feature points. The surface of a resultant 3D artistic face is the random mixed between feature point constraints and the artistic face example. In contrast, our work interprets a human face’s identity as well as an artistic face’s style and provides a controller for blending them systematically.

2.3. Shape Interpolation. Shape interpolation techniques aim to morph in-between shape semantically by blending example shapes together. Linear interpolation on vertex positions is simple and fast but it suffers from nonlinear nature of high rotation parts between example shapes. Sumner et al. [36] interpolate 3D shapes from example 3D shapes using rotation invariant shape-space by projecting the shapes into linearly scaling-shearing shape-space and linearly logarithm rotation shape-space with deformation gradient [24]. Winkler et al. [37] interpolate 3D shapes by traversing a shape hierarchy from global to local scales focusing on its edge lengths and dihedral angles [38] to form translation and rotation invariant shape-spaces. Marras et al. [39] separate a 3D shape into linear and nonlinear interpolable segments in order to interpolate them properly to save a computational time. These techniques generate impressive plausible 3D shapes in the range of given 3D example shapes. However, none of these techniques focuses on maintaining unique features (3D facial identity and 3D artistic style in our case) of in-between 3D shape during an interpolation process.

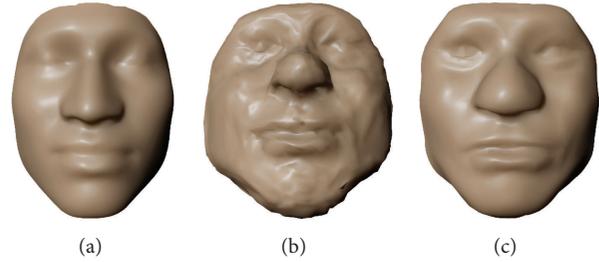


FIGURE 3: Comparison of different surface representations in blending process by applying random scaling factors to each representation: (a) the original face model, (b) vertex displacement representation, and (c) affine transformation representation.

3. Offline Data Processing

Our approach utilizes a precollected 3D human face dataset [40] consisting of 100 subjects (56 females and 44 males), ranging from 18 to 70 years old with diversified ethnicities. Each entry in the dataset consists of a 3D face mesh and its corresponding texture. We process the face dataset offline as follows. First, to build the correspondence for all the 3D face models in the dataset, we choose a face model outside the dataset as *the reference face*. Then, we register all the faces with the reference face using the deformation transferring algorithm proposed by Sumner and Popović [24]. To the end, all the registered faces have the same topology as the reference face. To statistically represent a human identity, we construct two reduced PCA subspaces from the dataset, as follows.

3.1. Face Geometry Subspace. The geometric identity of each 3D face is encoded as 3D affine transformations between the face and the average human face. We encode the face deformation (i.e., difference from the average human face) as affine transformations rather than naive vertex displacements because the former produces smoother surfaces during the blending process. Figure 3 illustrates example results of different surface representations when blending the surfaces with random scaling factors. To calculate the affine transformations for each face mesh, we employ the deformation gradient algorithm [24] because a 3D face does not contain a big translation, and this algorithm performs in linear time. Then, the PCA spaces are constructed from the obtained affine transformation data of all the faces.

One important issue in using affine transformation matrices is they cannot be linearly interpolated, while data points in a PCA space is expected to be linearly interpolable. In this work, we decompose an affine transformation matrix to a rotation component, R , and a scaling component, S , using polar decomposition. Then, we also use the exponential map to transform R to its logarithm space, $\log(R)$, to ensure its linear interpolable property as in MeshIK framework [36]. For each face mesh, we concatenate the $\log(R)$ and S of all the vertices to form two vectors, respectively. To the end, based on all the obtained vectors, we construct two truncated PCA subspaces, namely, $\phi_{\log(R)}$ and ϕ_S . In this work,

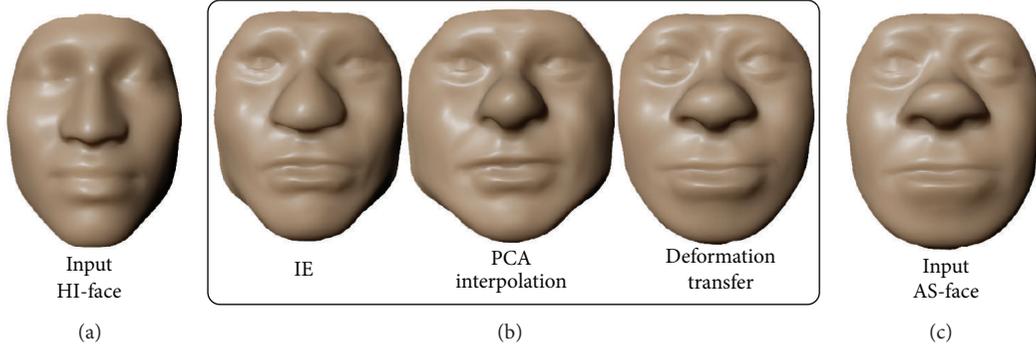


FIGURE 4: Comparison of different blending schemes. (a) Shape transferring using deformation transfer algorithm [24], (b) 50/50 linear interpolation between their corresponding PCA feature vectors, and (c) the result by our identity engineering (IE) algorithm.

we retain the 99 most significant eigenvectors to construct each truncated PCA subspace.

3.2. Face Texture Subspace. Similarly, we also construct a face texture PCA subspace, ϕ_{tex} , based on the textures of all the faces in the dataset. In this process, only the tone information represented by S and V components of the HSV color space of the face textures is used to avoid color artifacts. The color artifacts basically result from mixing the nonhuman color from the AS-face with the human texture from the HI-face in the RGB color space (will be discussed in more details in Section 5). Also, to prevent the obvious color artifacts from the blending process similar to the geometry part, the texture is represented by the image gradient from an average human face instead of direct pixel values.

4. Identity Engineering Algorithm

Since the PCA subspace is linear, we refer to the projected PCA coefficients of a face as its feature vector. In this subspace, we can interpolate two feature vectors as a way to combine two faces together. However, the simply combined/interpolated face could easily lose the major characteristics of both faces, as shown in Figure 4. Also, if we transfer the HI-face to the AS-face using a shape transferring technique such as the deformation transfer [24], the resultant face will be uncontrollably mixed between the AS-face and the HI-face, as shown in Figure 4 (geometric only) and Figure 12 (with texture).

For the above reason, we introduce an identity engineering (IE) algorithm that captures the uniqueness of the HI-face and transfers it to the AS-face automatically. This technique includes the ability to control the level of uniqueness of the HI-face. The IE algorithm consists of *data preparation*, *identity selection*, and *identity transfer steps*.

At the *data preparation step*, we first register both the input HI-face and AS-face with the reference face model through deformation transferring [24] as in the offline data processing step (Section 3). Then, the affine transformations of all the triangles of the registered HI-face and AS-face are extracted. Finally, the $\log(R)$ components of the two faces are transformed to two PCA feature vectors by projecting

them to the preconstructed PCA subspace, $\phi_{\log(R)}$. Similar PCA transformations are applied to the S component and the texture component (ϕ_{tex}). Since the IE algorithm needs to be applied to the $\log(R)$, S, and ϕ_{tex} components, separately; for convenience, we use V_h (or V_c) to refer to the obtained PCA feature vector of the HI-face (or the AS-face).

At the *identity selection step*, to identify the face uniqueness, the human likelihood of the V_h in its PCA space is computed by (1) [41]. Consider

$$L_{\text{human}} = \frac{1}{(2\pi)^{M/2}} \exp \left\{ -\frac{1}{2} \sum_{i=1}^M \frac{V_{h_i}^2}{v_i} \right\}, \quad (1)$$

where M is the number of the eigenvectors, v_i is the i th eigen-value, and V_{h_i} is i th coefficient of the feature vector V_h . L_{human} is the likelihood of V_h , which defines a similarity between the face V_h and the average human face. The low L_{human} (i.e., high V_h) means that the face is very unique (some facial features make the face away from an average face). Thus, the uniqueness score of the i th coefficient can be just calculated by (2). These uniqueness scores are the main criteria to select which coefficients should be transferred to V_c . To make the scores comparable, the scores are normalized to the range of [0, 1] and concatenated into one vector. We call this concatenated vector as a *score vector* S_h :

$$S_{h_i} = \frac{V_{h_i}^2}{v_i}. \quad (2)$$

At the *identity transfer step*, if any value in the score vector S_h (i.e., the uniqueness score of the i th coefficient) is larger than the user-specified threshold (via the slider bar control), then its corresponding coefficients of V_h will be transferred to V_c creating a resulting vector \tilde{V}_c as described in Algorithm 1. Therefore, when the users move the slider control from right to left, the resultant face will gradually reveal more identity of the HI-face. Figure 5 shows various resultant faces by varying the value of the slider. Figure 6 shows the 3D geometric views of two resultant artistic faces (the slider value is set to 0.5). Finally, we transform \tilde{V}_c back to their affine transformation matrix and texture spaces, which are used in the later sections in reconstructing the resultant 3D face and its texture.

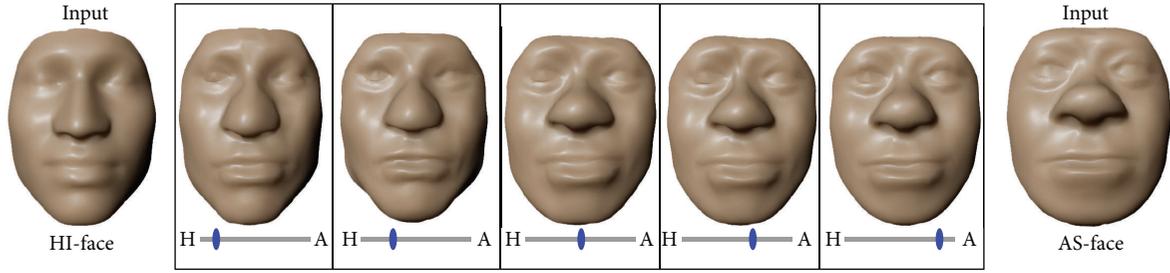


FIGURE 5: Results by varying the value of the slider control.

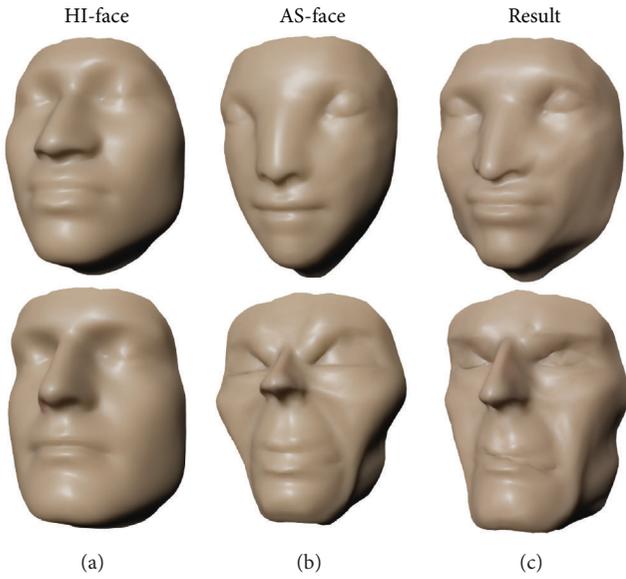


FIGURE 6: 3D geometric views of two resultant artistic faces (the slider value is set to 0.5).

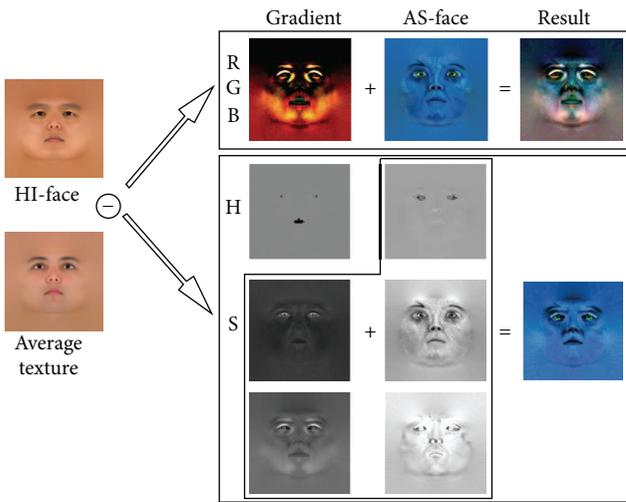


FIGURE 7: Comparison of the artistic texture synthesis process if different color spaces (RGB or HSV) are used. To emphasize the difference, the gradients are scaling up by 5. Blending in the RGB color space leads to a face texture with human-skin-base-color, while blending in the HSV color space (maintaining Hue) produces a face texture with artistic-style-base-color.

```

Input:  $V_h, V_c, S_h, threshold$ 
(1) for  $i = 1$  to size of  $(V_c)$  do
(2)   if  $S_h(i) \geq threshold$  then
(3)      $\tilde{V}_c(i) = V_h(i);$ 
(4)   else
(5)      $\tilde{V}_c(i) = V_c(i);$ 
(6)   end if
(7) end for
    
```

ALGORITHM 1: Identity engineering algorithm.

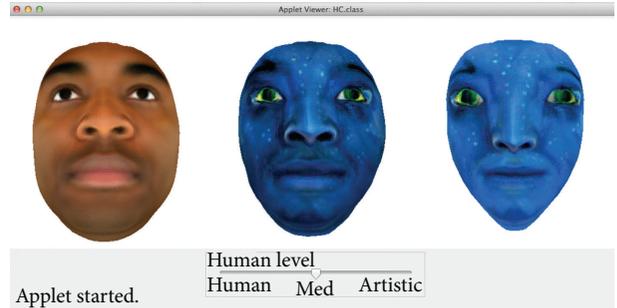


FIGURE 8: The runtime user interface of our approach. (Left): an input human face (HI-face), (Middle): the resultant identity-embodied artistic face with a slider to control a level of HI-face over the AS-face in the resultant face, (Right): an input artistic face (AS-face).

5. 3D Artistic Face Generation

5.1. Geometric Deformation. To reconstruct 3D meshes from the affine transformation matrices, the deformation transfer framework [24] is used to synthesize a 3D surface. In this framework, the affine transformation matrices are used to transform each triangle mesh on an average face surface individually. To keep the surface smooth, shared vertices among triangle meshes are used as constraints by solving them in a sparse matrix optimization (3):

$$\arg \min_x \left\| (Tx - \tilde{A}_c) \right\|, \quad (3)$$

where T is the linear operator of the average face surface, \tilde{A}_c is the affine transformations (from \tilde{V}_c) from the IE

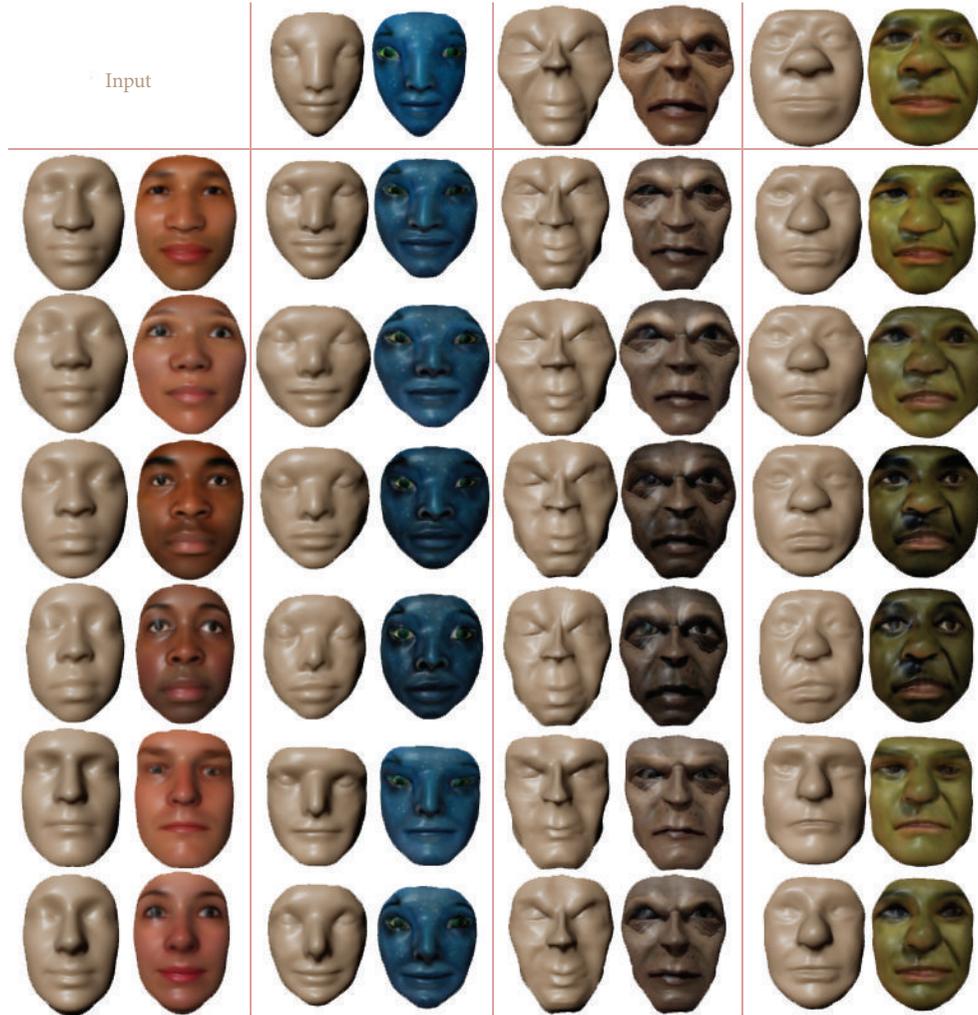


FIGURE 9: Example results by our approach. The leftmost column shows the input HI-faces. The top row shows the input AS-faces.

algorithm, and x is the vector containing the vertex positions of the resultant identity-embodied artistic face. This sparse matrix optimization can be solved by Cholesky factorization algorithm [24] to speed up the process.

5.2. Texture Enhancement. The IE algorithm is used in the texture space the same way as in the geometry space with slight modification. Since the AS-face texture may not be limited to the reddish-base-color space of the normal human skin texture, we need to come up with an effective means to blend the HI-face texture with the AS-face texture. Specifically, we use HSV color space for texture blending, instead of the conventional RGB color space, to maintain the AS-face base-color. The tone channels (S and V) of HSV are the only channels to be transferred to the artistic face texture, while the values in the Hue channel are kept intact. The main reason is, approximately, the Hue channel which represents the base color of the AS-face, while the S and V channels represent the variations of the base color (its facial identity). This will produce the face texture that contains the appearance of the AS-face with the tone of the HI-face. However, in

case the human base color needed to be kept as well, we can interpolate Hue value that prefers artistic's Hue over the human's Hue, for example, using quadratic interpolation. Figure 7 compares the artistic face texture synthesis process if the RGB or HSV color space is used. Thus, the IE algorithm is performed only under S and V channels as encoded in ϕ_{tex} .

To reconstruct the facial texture, we employ the Poisson image editing technique [42] to transfer the image gradient to the average human texture, as shown below:

$$\arg \min_F \sum_{\text{all pixels}} \|F - C - A\|, \quad (4)$$

where F is the resultant face texture, C is the image gradient from the IE algorithm, and A is the average face texture.

6. Results and Evaluations

The goal of this work is to keep the artistic-style intact as much as possible while adding new identities to it but not aiming to keep the human identity to be recognizable out of the artistic face consistently. Thus, the goal of our evaluation

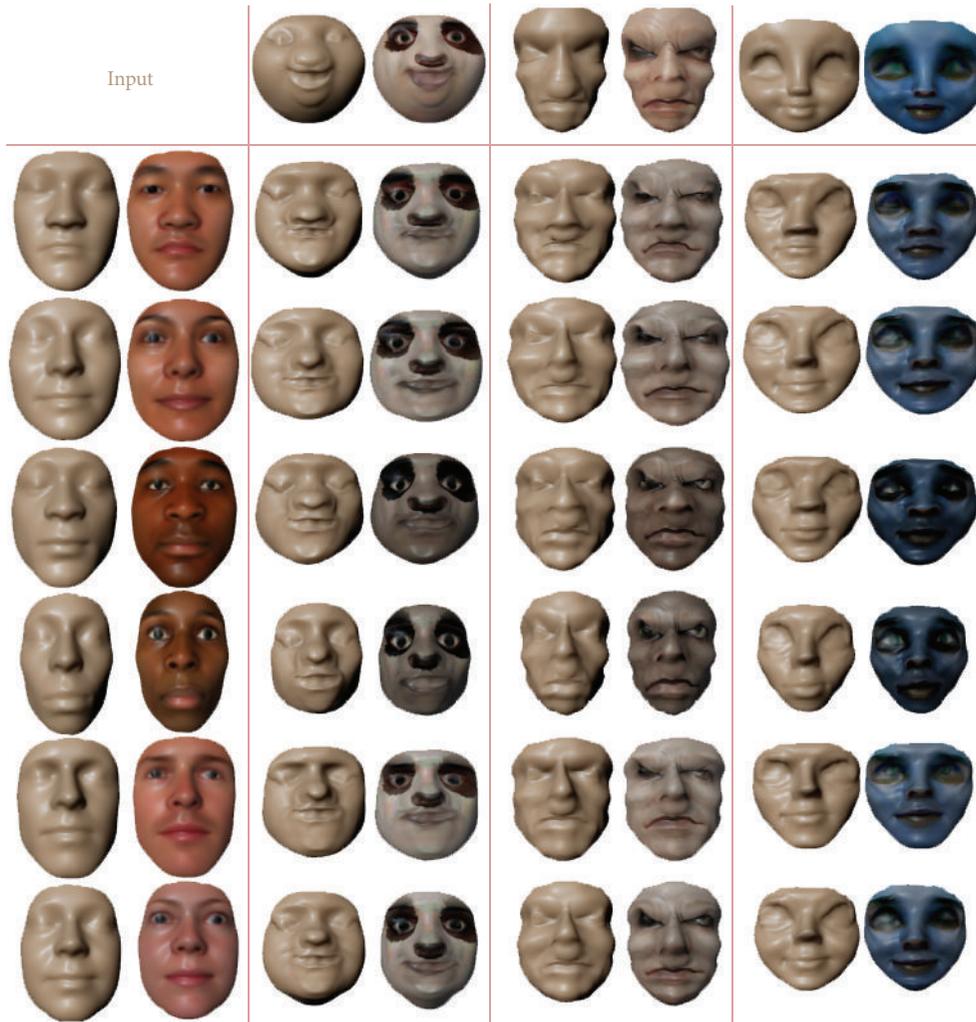


FIGURE 10: The second group of example results by our approach. The leftmost column shows the input HI-faces. The top row shows the input AS-faces.

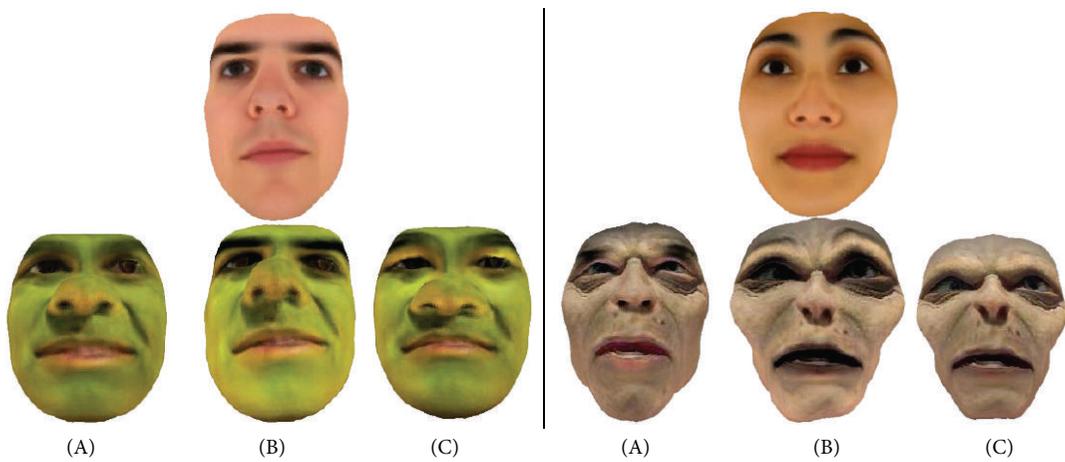


FIGURE 11: Questionnaire examples. (Left): the questionnaire that has the highest recognition rate (94.1 percent) and (Right) the questionnaire that has the lowest recognition rate (58.8 percent). The correct answers of both are (B).

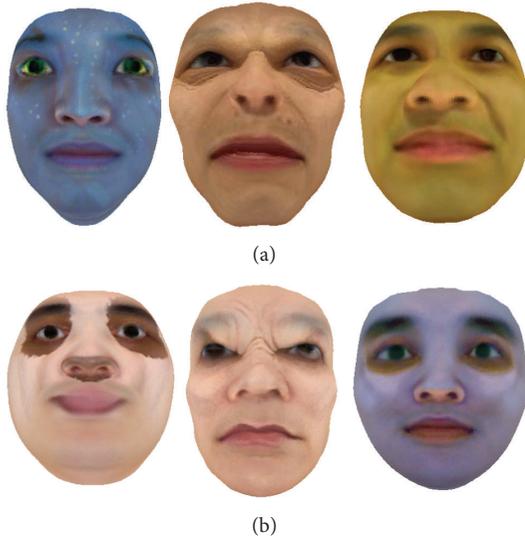


FIGURE 12: Linear interpolation results. (a) The results from the first row of Figure 9 but using linear interpolation technique. (b) The results from the first row of Figure 10 but using linear interpolation technique.

is to check the variation of the synthesized artistic faces. The artistic faces synthesized from the same AS-face with a specific HI-face should resemble this specific HI-face than other HI-faces.

To do so, first, we allow three novice users to use our developed user interface (UI) as shown in Figure 8 to create target artistic faces. This UI provides users a slider to adjust the level of HI-face and AS-face. Specifically, the slider is used to adjust the value *threshold* in Algorithm 1. The users were instructed to create cartoon characters that retain the original artistic style but also contain a hint of the target human face identity. Each face contains 3000 vertices with 5000 triangles. Figures 9 and 10 show some results generated by the users.

Second, the recognition rate (how/whether people can recognize the artistic face as an individual person) is conducted with a user study comparing our method with the traditional linear interpolation technique. Seventeen computer science student volunteers participated in this study. It was composed of 11 questionnaires, each of which asks the participants to select the artistic face that most resembles the input human face from 3 other same artistic-style face candidates (the other 2 candidates are chosen randomly from the results) as shown in Figure 11. The experiments are conducted for our technique and for the linear interpolation technique separately.

For our technique, based on the collected survey results, the average recognition rate is 88.5 percent, and its standard deviation is 11.2. However, if we consider the majority vote scheme (i.e., the majority-voted face for each questionnaire is the one picked by nine or more out of the seventeen participants), all of the 11 questionnaires were correctly answered. In this sense, the recognition rate is perfect. Also, five out of the seventeen participants answered all the questionnaires correctly. For the linear interpolation technique, the average



FIGURE 13: The results of using our approach in an entertainment application. From left to right: the input human photograph, the selected AS-face, and the final result. Each 3D HI-face is created from a photograph of the celebrity by using FaceGen [43]. Note that the synthesized faces are only the facial part but we adjust all the skin parts of the celebrities to match the AS-face styles. Also, some faces are rotated to fit to the photograph face poses.

recognition rate is 62.1 percent, and its standard deviation is 27.3. For the majority vote scheme, 7 out of 11 were correctly answered. The examples of the linear interpolation results are shown in Figure 12.

In conclusion, these results imply that, on average, the artistic faces created by our technique can be consistently distinguished as their input human face identities by most users than the traditional linear interpolation technique.

7. Discussion and Conclusion

In this paper, we present a novel modeling tool to automatically generate 3D artistic faces embodied with individual identities. By providing a simple slider, even novice users are able to interactively produce desired artistic faces in a few seconds. Also, it can run in real-time on an off-the-shelf computer without GPU acceleration.

Through our experiments, we found that our approach can be used to synthesize various artistic face characters that are differentiated by the human identities while maintaining the artistic style. It should be noted that the identity-embodied artistic faces created by our approach are not intended to be used in a face recognition or other similar applications but to model various artistic faces for character differentiation which can be used in entertainment applications, for example, for creating a specific monster character face from a person. Figure 13 shows an example of using this technique to create monster faces out of the celebrity faces.

The current approach has several limitations. First, our approach is built on top of a precollected face dataset. However, the number of faces (100 faces) in the dataset is relatively small, which may not be sufficient to fit a well-behaved probabilistic model (the Gaussian distribution assumption). This issue limits the identity capturing ability of our approach. Second, our current approach cannot handle the generation of various facial expressions and animations on the resultant artistic faces, although animation is not the focus of the current work. By extending existing facial expression retargeting and animation techniques, we expect realistic facial expressions and animations can be soundly reproduced on the resultant artistic face models.

In the future, apart from the slider control, we plan to explore other user-friendly interfaces such as suggestive user interfaces [44] that allow users to select plausible shapes from example models or a region-based controller for modifying each facial region separately. For the texture enhancement, we plan to further develop a texture synthesis technique that can be adapted to different 3D facial geometry such as re-rendering necessary shadows and lighting effects.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The author would like to thank Professor Zhigang Deng for his help in revising the paper to the current version.

References

- [1] Z. Deng and J. Noh, *Computer Facial Animation: A Survey, Data-Driven 3D Facial Animation*, Springer, 2007.
- [2] B. Gooch, E. Reinhard, and A. Gooch, "Human facial illustrations: Creation and psychophysical evaluation," *ACM Transactions on Graphics*, vol. 23, no. 1, pp. 27–44, 2004.
- [3] S. E. Brennan, *The Caricature Generator: Leonardo*, vol. 18, ACM, 1985.
- [4] H. Koshimizu, M. Tominaga, T. Fujiwara, and K. Murakami, "On KANSEI facial image processing for computerized facial caricaturing system PICASSO," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC '99)*, vol. 6, pp. 294–299, Tokyo, Japan, October 1999.
- [5] R. Hsu and A. K. Jain, "Generating discriminating cartoon faces using interacting snakes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 11, pp. 1388–1398, 2003.
- [6] Z. Mo, J. P. Lewis, and U. Neumann, "Improved automatic caricature by feature normalization and exaggeration," in *Proceedings of the ACM SIGGRAPH Sketches (SIGGRAPH '04)*, p. 57, August 2004.
- [7] L. Liang, H. Chen, X. Ying-Qing, and S. Heung-Yeung, "Example-based caricature generation with exaggeration," in *Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, p. 386, IEEE Computer Society, 2002.
- [8] J. Liu, Y. Chen, and G. Wen, "Mapping learning in eigenspace for harmonious caricature generation," in *Proceedings of the 14th Annual ACM International Conference on Multimedia (MULTIMEDIA '06)*, pp. 683–686, 2006.
- [9] H. Yu and J. J. Zhang, "Mean value coordinates-based caricature and expression synthesis," *Signal, Image and Video Processing*, vol. 7, no. 5, pp. 899–910, 2013.
- [10] N. K. Le, Y. P. Why, and G. Ashraf, "Shape stylized face caricatures," in *Advances in Multimedia Modeling*, vol. 6523 of *Lecture Notes in Computer Science*, pp. 536–547, Springer, Berlin, Germany, 2011.
- [11] E. Akleman and J. Reisch, "Modeling expressive 3D caricatures," in *Proceedings of the International ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '04)*, pp. 1–61, Los Angeles, Calif, USA, August 2004.
- [12] H. Chen, N. Zheng, L. Liang, Y. Li, Y. Xu, and H. Shum, "PicToon: a personalized image-based cartoon system," in *Proceedings of the 10th ACM International Conference on Multimedia*, pp. 171–178, Juan les Pins, France, December 2002.
- [13] H. Chen, Z. Liu, C. Rose, Y. Xu, H. Y. Shum, and D. Salesin, "Example-based composite sketching of human portraits," in *Proceedings of the 3rd International Symposium on Non-Photorealistic Animation and Rendering (NPAR '04)*, pp. 95–153, 2004.
- [14] X. Wang and X. Tang, "Face photo-sketch synthesis and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, pp. 1955–1967, 2009.
- [15] M. Meng, M. Zhao, and S.-C. Zhu, "Artistic paper-cut of human portraits," in *Proceedings of the International Conference on Multimedia (MM '10)*, pp. 931–934, 2010.
- [16] I. Berger, A. Shamir, M. Mahler, E. Carter, and J. Hodgins, "Style and abstraction in portrait sketching," *ACM Transactions on Graphics*, vol. 32, no. 4, article 55, pp. 1–12, 2013.
- [17] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin, "Synthesizing realistic facial expressions from photographs," in *Proceedings of the Annual Conference on Computer Graphics (SIGGRAPH '98)*, pp. 75–84, July 1998.
- [18] B. Bickel, M. Botsch, R. Angst et al., "Multi-scale capture of facial geometry and motion," *ACM Transactions on Graphics*, vol. 26, no. 3, article 33, 2007.
- [19] D. Bradley, W. Heidrich, T. Popa, and A. Sheffer, "High resolution passive facial performance capture," *ACM Transactions on Graphics*, vol. 29, no. 4, article 41, 2010.
- [20] T. Beeler, F. Hahn, D. Bradley et al., "High-quality passive facial performance capture using anchor frames," *ACM Transactions on Graphics*, vol. 30, no. 4, article 75, 2011.
- [21] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz, "Spacetime faces: high resolution capture for modeling and animation," in *Proceedings of the ACM SIGGRAPH (SIGGRAPH '04)*, pp. 548–558, August 2004.
- [22] K. Singh and E. Fiume, "Wires: a geometric deformation technique," in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*, pp. 405–414, Orlando, Fla, USA, July 1998.
- [23] J. P. Lewis, J. Mooser, Z. Deng, and U. Neumann, "Reducing blendshape interference by selected motion attenuation," in *Proceeding of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pp. 25–29, April 2005.
- [24] R. W. Sumner and J. Popović, "Deformation transfer for triangle meshes," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 399–405, 2004.
- [25] P. Joshi, W. C. Tien, M. Desbrun, and F. Pighin, "Learning controls for blend shape based realistic facial animation," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 35–42, 2003.

- [26] W. Ma, A. Jones, J. Chiang et al., “Facial performance synthesis using deformation-driven polynomial displacement maps,” *ACM Transactions on Graphics*, vol. 27, no. 5, article 121, pp. 1–10, 2008.
- [27] W. Feng, B. Kim, and Y. Yu, “Real-time data driven deformation using kernel canonical correlation analysis,” in *Proceeding of the ACM SIGGRAPH Papers (SIGGRAPH '08)*, pp. 1–91, August 2008.
- [28] T. Suontphunt, Z. Mo, U. Neumann, and Z. Deng, “Interactive 3d facial expression posing through 2d portrait manipulation,” in *Proceedings of The Graphics Interface*, pp. 177–184, Windsor, Canada, May 2008.
- [29] V. Blanz and T. Vetter, “A morphable model for the synthesis of 3D faces,” in *Proceedings of the ACM 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*, pp. 187–194, Los Angeles, Calif, USA, August 1999.
- [30] D. Vlasic, B. Matthew, H. Pfister, and J. Popovic, “Face transfer with multilinear models,” *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 426–433, 2005.
- [31] H. Li, T. Weise, and M. Pauly, “Example-based facial rigging,” *ACM Transactions on Graphics*, vol. 29, no. 4, article 32, 2010.
- [32] T. Weise, H. Li, L. van Gool, and M. Pauly, “Face/off: live facial puppetry,” in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '09)*, pp. 7–16, New Orleans, La, USA, August 2009.
- [33] T. Weise, S. Bouaziz, H. Li, and M. Pauly, “Realtime performance-based facial animation,” *ACM Transactions on Graphics*, vol. 30, no. 4, article 77, 2011.
- [34] H. Li, J. Yu, Y. Ye, and C. Bregler, “Realtime facial animation with on-the-fly correctives,” *ACM Transactions on Graphics*, vol. 32, no. 4, article 42, 2013.
- [35] T. Suontphunt and U. Neumann, “3D facial surface and texture synthesis using 2D landmarks from a single face sketch,” in *Proceedings of 3D Imaging, Modeling, Processing, Visualization and Transmission (3DimPVT '12)*, pp. 152–159, Zurich, Switzerland, October 2012.
- [36] R. W. Sumner, M. Zwicker, C. Gotsman, and J. Popović, “Mesh-based inverse kinematics,” in *Proceedings of ACM SIGGRAPH*, pp. 488–495, 2005.
- [37] T. Winkler, J. Drieseberg, M. Alexa, and K. Hormann, “Multi-scale geometry interpolation,” *Computer Graphics Forum*, vol. 29, no. 2, pp. 309–318, 2010, Proceedings of Eurographics.
- [38] T. W. Sederberg, P. Gao, G. Wang, and H. Mu, “2-D shape blending: an intrinsic solution to the vertex path problem,” in *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, pp. 15–18, August 1993.
- [39] S. Marras, T. J. Cashman, and K. Hormann, “Efficient interpolation of articulated shapes using mixed shape spaces,” *Computer Graphics Forum*, vol. 32, no. 8, pp. 258–270, 2013.
- [40] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato, “A 3D facial expression database for facial behavior research,” in *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR '06)*, pp. 211–216, April 2006.
- [41] M. E. Tipping and C. M. Bishop, “Probabilistic principal component analysis,” *Journal of the Royal Statistical Society B*, vol. 61, no. 3, pp. 611–622, 1999.
- [42] P. Pérez, M. Gangnet, and A. Blake, “Poisson image editing,” *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 313–318, 2003.
- [43] Singular-Inversions, “Facegen,” 2007, <http://www.facegen.com/>.
- [44] S. Chaudhuri, E. Kalogerakis, L. Guibas, and V. Koltun, “Probabilistic reasoning for assembly-based 3D modeling,” *ACM Transactions on Graphics*, vol. 30, no. 4, article 35, 2011.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

