Portrait Painting Using Active Templates

Mingtian Zhao* Song-Chun Zhu*
University of California, Los Angeles & Lotus Hill Institute

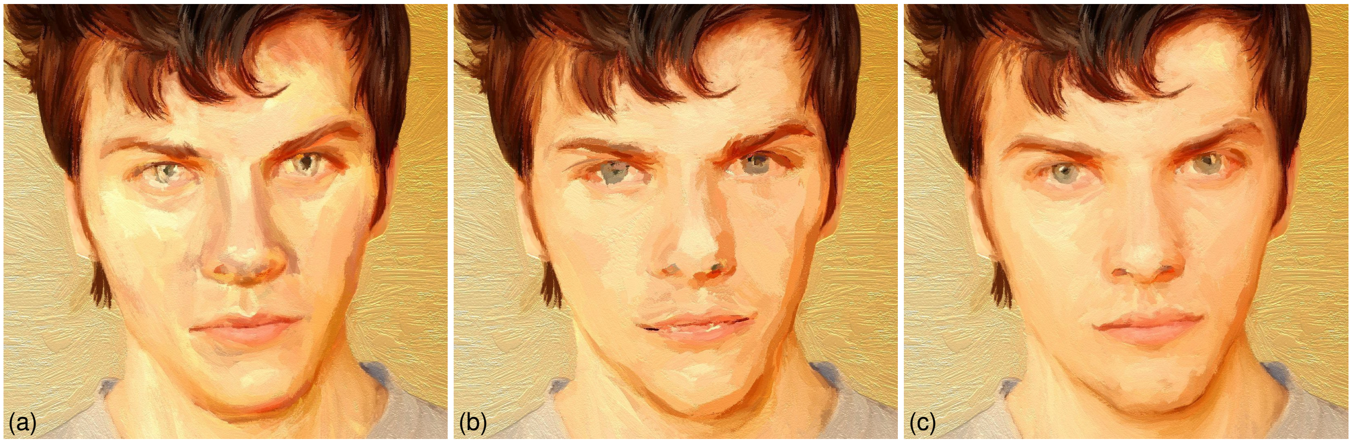

Figure 1: *Three portrait paintings rendered with different templates using our method. Their corresponding source photograph is in Fig.5. Notice: all painting images in this paper are best viewed on a color display at 400% zoom unless annotated otherwise.*

Abstract

Portraiture plays a substantial role in traditional painting, yet it has not been studied in depth in painterly rendering research. The difficulty in rendering human portraits is due to our acute visual perception to the structure of human face. To achieve satisfactory results, a portrait rendering algorithm should account for facial structure. In this paper, we present an example-based method to render portrait paintings from photographs, by transferring brush strokes from previously painted portrait templates by artists. These strokes carry rich information about not only the facial structure but also how artists depict the structure with large and decisive brush strokes and vibrant colors. With a dictionary of portrait painting templates for different types of faces, we show that this method can produce satisfactory results.

CR Categories: I.3.4 [Computer Graphics]: Graphics Utilities—Paint Systems; J.5. [Computer Applications]: Arts and Humanities—Fine Arts

Keywords: active template, human face, painterly rendering, portrait painting, stroke-based rendering

1 Introduction

Portrait painting, which depicts human faces and their expressions, requires the highest level of skills among common painting genres including landscape, still-life, etc. [Brooker 2010] Probably because of the high requirements, we have not seen much dedicated work on painterly rendering of portraits despite its popularity in traditional painting, and existing generic methods in the painterly rendering literature still cannot generate as satisfactory results as they can produce for non-portrait images.

Through continual observation and memorization since infancy, our visual systems have become so sensitive to the structure of human face that even a slight abnormality will be noticed [Sinha et al. 2006]. Therefore, to generate satisfactory portrait paintings, a rendering algorithm should account for facial structure. However, in most existing painterly rendering systems, a human face is treated simply as a generic image patch [Haeberli 1990; Meier 1996; Litwinowicz 1997; Hertzmann 1998; Hertzmann 2001; Collomosse and Hall 2002; Gooch et al. 2002; Hays and Essa 2004; Lu et al. 2010], or at the most, as one of the common semantic patterns such as trees and flowers which only affects general rendering parameters [Zeng et al. 2009; Lin et al. 2010; Zhao and Zhu 2010]. Without special structure-aware treatment for human faces, these generic painting methods are not likely to achieve good results in portrait rendering. They usually either destroy many necessary details or behave too cautiously to add enough painterly effects. For example, the two paintings in Fig.2 rendered using previous methods look almost like photographs.

Technically, we are faced with a few immediate challenges in rendering a portrait painting:

- To effectively depict a face which conveys a strong impression of 3D structures, artists usually use large and decisive brush strokes and maintain sharp contrasts among them, as shown by the example in Fig.3a. The use of large strokes has to be accurate enough to avoid destroying necessary details for our visual perception of the facial structure.
- While realistic portrait photographs usually only have monotonous colors, skillful artists can depict faces with vibrant colors not present in the photographs, as shown by the example in Fig.3b. In rendering, we would like to achieve such a variety of colors with contrasts to each other, while still preserving normal face appearances.

*e-mails: {mtzhao|sczhu}@stat.ucla.edu

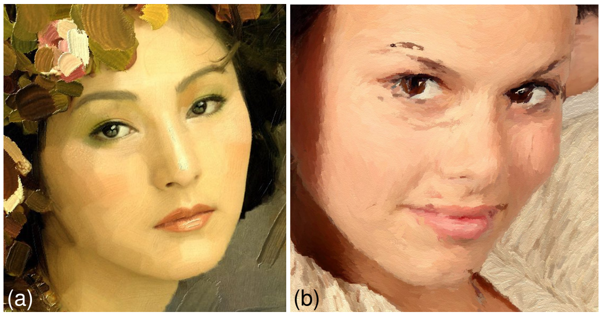

Figure 2: *Example portrait paintings rendered using previous methods. (a) is cropped from Fig.10b of Zeng et al. [2009], which is almost like a photograph. (b) is rendered with small and high-opacity strokes placed according to a generic orientation field (generated by diffusing the facial structure sketches, as shown by the black segments connecting the blue dots in Fig.5) [Hays and Essa 2004; Zeng et al. 2009]. This approach blurs out details and does not convey a strong impression of 3D structures as good portrait paintings usually do (e.g., the two paintings in Fig.3).*

Figure 3: *Faces cropped from real portrait paintings. (a) is from a practice portrait by Yifei Chen, which is depicted with large and decisive brush strokes, conveying a strong impression of 3D structures. (b) is from a self portrait by Paul Gauguin, which contains vibrant colors usually not existing in realistic portrait photographs.*

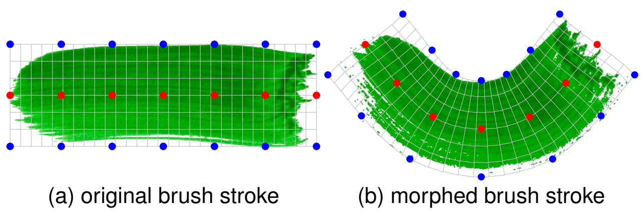

Figure 4: *Our brush stroke model. The brush texture maps are borrowed from the dictionary of Zeng et al. [2009]. The morphing is performed using thin-plate spline (TPS) transformation [Barrodale et al. 1993] based on backbone control points (red and blue dots), and texture mapping based on the quadrilateral mesh (gray grids).*

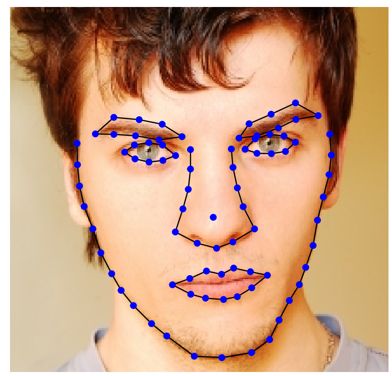

Figure 5: *The facial structure model we use contains 83 landmark points (blue dots) computed using the active appearance model (AAM) [Cootes et al. 2001]. Photograph courtesy of graur razvan ionut @freedigitalphotos.net.*

To address these problems, we propose an example-based method to render portrait paintings from photographs. We ask artists to make portrait paintings and record the sequences of strokes they paint using a fully manual digital painting system adapted from Zeng et al. [2009]. With their help, we build a dictionary containing over 100 portrait paintings with complete information of their stroke-by-stroke generating processes. The dictionary covers faces of different genders, ages, ethnic groups, poses, etc. With this dictionary, we render painterly portrait images by transferring brush strokes from source portraits painted by artists. In order to reuse these source portraits as templates for rendering various target images, we demand that their shapes should be able to deform, and their colors to shift, so we call them *active portrait templates*.

Compared with existing generic painting methods in the literature, the stroke sequences in our active portrait templates carry information about not only the facial structure but also how artists depict the structure with large and decisive strokes and vibrant colors, therefore our method can overcome the challenges mentioned above. Fig.1 shows an example of our results, in which the three paintings are rendered from the photograph in Fig.5 with different templates from our dictionary.

2 Related Work

To our knowledge, there is not much dedicated work on portraiture in the painterly rendering literature. DiPaola [2007] described a knowledge-based approach for painterly rendering of portraits, but this work focused mainly on the general methodology rather than a detailed rendering algorithm, and did not take advantage of the rich structural information of human faces.

In some other areas of non-photorealistic rendering, the depiction of human faces has been widely studied, for example, graphite or color sketch [Li and Kobatake 1997; Chen et al. 2002a; Chen et al. 2002b; Chen et al. 2004; Luft and Deussen 2004; Gooch et al. 2004; Tresset and Leymarie 2005; Min et al. 2007; Wang and Tang 2009], artistic binarization/paper-cut [Meng et al. 2010], cartoon [Chopra and Meyer 2003], etc. Most of these methods take advantage of some facial structure information such as hierarchical facial representations and full 3D geometrical models, and can achieve very nice results.

3 Active Portrait Template

In our dictionary of portrait paintings $\mathcal{T} = \{\mathbf{T}_k, i = 1, 2, \cdots, K\}$, each active portrait template $\mathbf{T}_k$ consists of its original portrait photograph $\mathbf{I}_k$ which has a facial structure $\mathbf{F}_k$, and its sequence of brush strokes $\mathbf{S}_k$ painted by artists.

Brush Stroke Model. For modeling brush strokes, we borrow brush texture maps from the dictionary of Zeng et al. [2009], and adopt a curved stroke model, as shown in Fig.4. In addition to the texture map, each brush stroke has the following attributes:

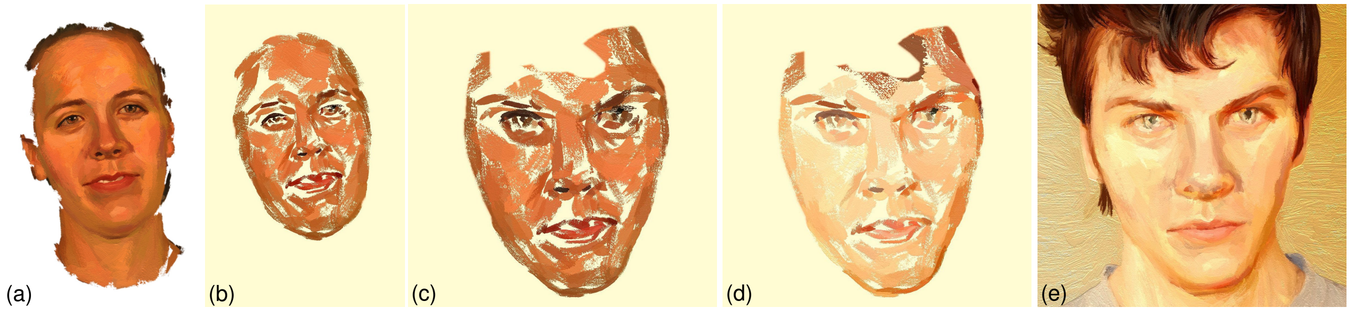

Figure 6: *Our rendering pipeline. (a) is a template selected from the dictionary (the one in Fig.7c). (b) displays only a few strokes of (a) on the canvas for better illustration. Using stroke position transformation we generate (c) from (b), and using stroke color shift we generate (d) from (c). Continuing with (d) to have all strokes painted, we get the final result (e). During position transformation, some strokes near the forehead go outside the facial region on the target photograph (obtained using interactive segmentation [Zeng et al. 2009]) and are deleted.*

1. A list of backbone control points (as marked by the red and blue dots in Fig.4),
2. The stroke width, which equals the distance between the two rows of blue dots on each side of the stroke, and
3. The stroke color. Our artists have the freedom to choose their desired colors for the brush strokes, and their choices are recorded in the templates.

In order to morph a brush stroke from the dictionary (e.g., Fig.4a) to match a stroke path on the canvas (e.g., a curve passing through the red dots in Fig.4b), we first compute the positions of the blue dots by offseting the red dots along the normal directions of the path. The normal directions are computed by approximating the path with a Catmull-Rom spline interpolating the red dots [Catmull and Rom 1974], and the offset distance is half of the stroke width. Then we compute a thin-plate spline (TPS) transformation [Barrodale et al. 1993] between the pairs of source and target dot positions (e.g., between the corresponding backbone control points in Figs.4a and 4b), and apply the transformation to the vertices of a quadrilateral mesh covering the source brush stroke to get the deformed mesh. Finally, we compute a texture mapping using the mesh, with a bilinear interpolation inside each quadrilateral.

Facial structure Model. For the facial structure $\mathbf{F}_k$, we use the representation introduced in the active appearance model (AAM) [Cootes et al. 2001], which contains 83 landmark points: 8 for each eye, 8 for each eyebrow, 14 for the nose, 12 for the mouth, and the rest 25 for the face contour (as shown in Fig.5). These landmark points are computed using AAM on the source photographs of the portrait templates, and manually fine-tuned for better accuracy. Using this representation, $\mathbf{F}_k$ is a 166-dimensional vector containing the (x, y)-coordinates of the 83 landmarks.

4 Rendering

We render a portrait painting from a given portrait photograph by transferring strokes from one of our templates in the dictionary. To do this, we need to select a template, then compute a transformation for the strokes in the template, to obtain their new positions according to the shape difference between the template and the target face, and their new colors according to the color difference between the template and the target photograph. Our rendering pipeline is illustrated in Fig.6.

Template Selection. We use a semi-automatic template selection strategy. The system computes a distance between the target portrait photograph $\mathbf{I}_\mathrm{T}$ and the photograph $\mathbf{I}_k$ of each template in the dictionary, and presents the top-10 templates with the smallest distances, from which the user can select one according to his/her desired styles. We use a distance metric

$$D(\mathbf{I}_\mathrm{T}, \mathbf{I}_k) = \alpha D_\mathrm{S}(\mathbf{I}_\mathrm{T}, \mathbf{I}_k) + (1-\alpha) D_\mathrm{C}(\mathbf{I}_\mathrm{T}, \mathbf{I}_k), \tag{1}$$

in which D_S and D_C are the shape and color differences, respectively, and α is a user-specified parameter balancing the two.

To compute the shape difference, we first do a principal component analysis (PCA) on all facial structure vectors $\mathbf{F}_k, k = 1, 2, \cdots, K$ in the dictionary, which yields a linear transformation

$$\mathbf{F}'_k = (\mathbf{F}_k - \mathbf{F}_0)\mathbf{W}_\mathbf{F}, \tag{2}$$

in which $\mathbf{F}_0 = \frac{1}{K}\sum_{k=1}^{K} \mathbf{F}_k$ contains the mean landmark coordinates, $\mathbf{W}_\mathbf{F}$ is the PCA projection matrix, and $\mathbf{F}'_k$ contains the projected coordinates in a reduced-dimension space (we use 5 dimensions) spanned by eigenvectors of the covariance matrix of $\mathbf{F}_k - \mathbf{F}_0, k = 1, 2, \cdots, K$ corresponding to the 5-largest eigenvalues (so $\mathbf{F}'_k$ is a 5-dimensional vector). We apply this PCA transformation to the landmarks $\mathbf{F}_\mathrm{T}$ of the target photograph $\mathbf{I}_\mathrm{T}$ (also computed using AAM and manually fine-tuned by the user if necessary) to get $\mathbf{F}'_\mathrm{T} = (\mathbf{F}_\mathrm{T} - \mathbf{F}_0)\mathbf{W}_\mathbf{F}$. Then the shape difference is computed with the Mahalanobis distance

$$D_\mathrm{S}(\mathbf{I}_\mathrm{T}, \mathbf{I}_k) = \sqrt{(\mathbf{F}'_\mathrm{T} - \mathbf{F}'_k)\mathbf{\Lambda}_\mathbf{F}^{-1}(\mathbf{F}'_\mathrm{T} - \mathbf{F}'_k)^\top}, \tag{3}$$

in which $\mathbf{\Lambda}_\mathbf{F}$ is a diagonal matrix containing the 5-largest eigenvalues of the covariance matrix of $\mathbf{F}_k - \mathbf{F}_0$. The purpose of using PCA here is to work in a reduced-dimension space (from 166 to 5 in our case) for faster computation, since we only need to compute the PCA once on the dictionary and use it for all future target photographs.

To compute the color difference, for each template, we compute a TPS transformation from its landmarks $\mathbf{F}_k$ to the mean coordinates $\mathbf{F}_0$, then morph its photograph $\mathbf{I}_k$ to $\mathbf{J}_k$ using the TPS. Similarly, we morph the target photograph $\mathbf{I}_\mathrm{T}$ to $\mathbf{J}_\mathrm{T}$ using a TPS transformation from $\mathbf{F}_\mathrm{T}$ to $\mathbf{F}_0$. In this way, all photographs are aligned according to the landmarks. After resizing the images to the same resolution, D_C is computed using PCA again (i.e., the distance used in the Eigenface method [Turk and Pentland 1991]):

$$\mathbf{J}'_k = (\mathbf{J}_k - \mathbf{J}_0)\mathbf{W}_\mathbf{J}, \tag{4}$$

$$\mathbf{J}'_\mathrm{T} = (\mathbf{J}_\mathrm{T} - \mathbf{J}_0)\mathbf{W}_\mathbf{J}, \tag{5}$$

$$D_\mathrm{C}(\mathbf{I}_\mathrm{T}, \mathbf{I}_k) = \sqrt{(\mathbf{J}'_\mathrm{T} - \mathbf{J}'_k)\mathbf{\Lambda}_\mathbf{J}^{-1}(\mathbf{J}'_\mathrm{T} - \mathbf{J}'_k)^\top}, \tag{6}$$

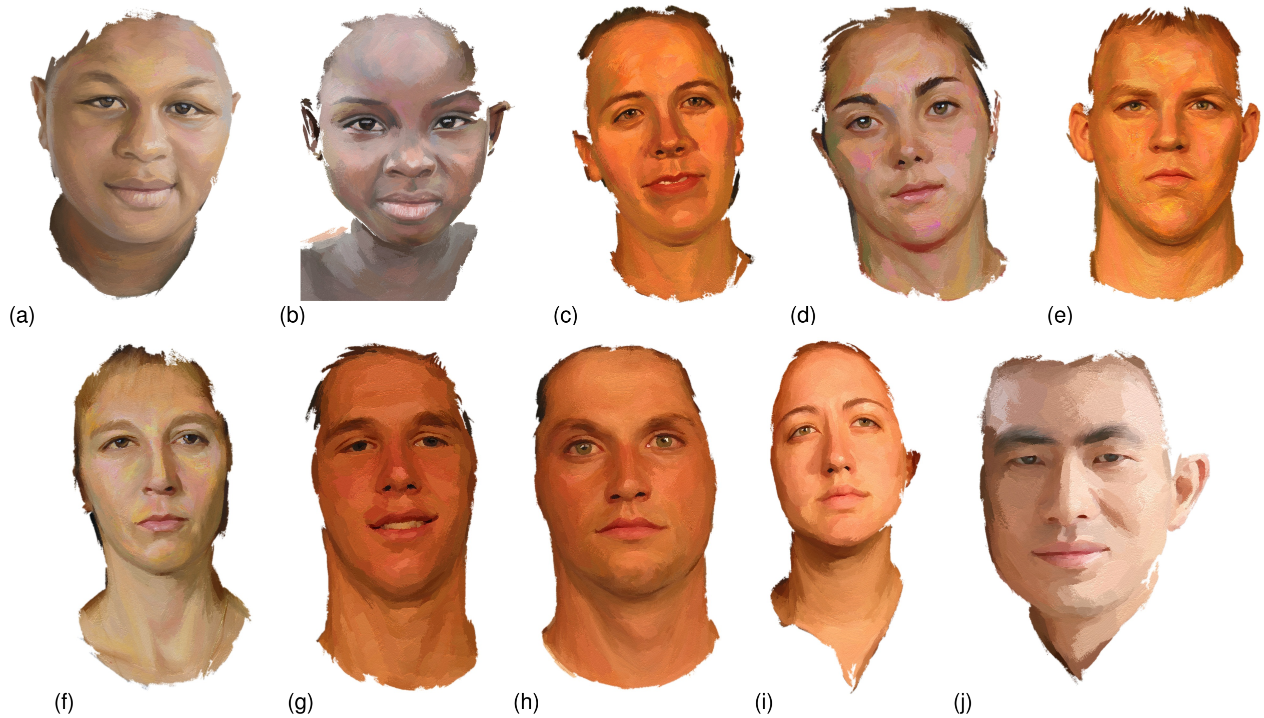

Figure 7: *Ten examples from our dictionary of portrait painting templates. For easier evaluation of our method and results by the readers, in this paper, we have only used templates displayed here to render all the results. The whole dictionary contains over 100 templates, covering faces of different genders, ages, ethnic groups, poses, etc. Zoom to 600% to view details.*

in which $\mathbf{J}_0$ is the mean of all $\mathbf{J}_k$, $\mathbf{W}_\mathbf{J}$ is the projection matrix, and $\mathbf{\Lambda}_\mathbf{J}$ is a diagonal matrix containing the 25-largest eigenvalues of the covariance matrix of $\mathbf{J}_k - \mathbf{J}_0$ (so $\mathbf{J}'_k$ and $\mathbf{J}'_\mathrm{T}$ are 25-dimensional vectors). Here we need a few more dimensions to model the color appearance than we have used for the facial structure in Eq.(2).

Stroke Position Transformation. With a selected template, we use TPS once more to compute the stroke position transformation. First, we compute the TPS transformation from $\mathbf{F}_k$ (of the selected template) to the target face structure $\mathbf{F}_\mathrm{T}$. Then for each stroke in the selected template, we compute the new positions of its backbone points on the target photograph using this TPS transformation. During the transformation, some strokes may go outside the facial region on the target photograph (obtained using interactive segmentation [Zeng et al. 2009]) and are deleted. Fig.6 illustrates the stroke position transformation, in which we obtain (c) from (b) through the transformation.

Stroke Color Shift. Since colors are usually different between the template (more precisely, its corresponding source photograph) and the target photograph, we need to shift the color of each brush stroke in the template to match the target. To achieve this, we first prepare three relevant colors for each stroke:

1. In the template, the stroke color $\mathbf{C}_\mathrm{A}$ chosen by our artist.
2. In the template, the color of the source photograph $\mathbf{C}_\mathrm{S}$ in the area of the stroke, which is approximated by the weighted average of the pixel colors at the backbone control points (the red dots only). In practice, heavier weights are given to points near the middle of the stroke, and lower weights to the head and tail. The weighted average is computed in the perceptually uniform CIELAB color space.
3. In the target photograph, the color $\mathbf{C}_\mathrm{T}$ in the area of the stroke, which is also approximated using pixel colors at the backbone control points.

With $\mathbf{C}_\mathrm{A}$, $\mathbf{C}_\mathrm{S}$ and $\mathbf{C}_\mathrm{T}$, the new stroke color $\mathbf{C}_\mathrm{N}$ is computed in the CIELCH color space (a cylindrical form of CIELAB) including three channels for lightness ℓ_N, chroma c_N and hue h_N, respectively. We use a similar color transfer idea to Reinhard et al. [2001] in order to maintain contrasts:

$$\ell_\mathrm{N} = \ell_\mathrm{T} + (\ell_\mathrm{A} - \ell_\mathrm{S})\min\{1, \ell_\mathrm{T}/\ell_\mathrm{S}\}, \tag{7}$$
$$c_\mathrm{N} = c_\mathrm{T} + (c_\mathrm{A} - c_\mathrm{S})\min\{1, c_\mathrm{T}/c_\mathrm{S}\}, \tag{8}$$
$$h_\mathrm{N} = h_\mathrm{T} + (h_\mathrm{A} - h_\mathrm{S}), \tag{9}$$

in which the $\min\{\}$ term is useful for avoiding extreme colors by constraining the color deviation from the photograph, and h_N is periodic over intervals of 2π.

Finally, $\mathbf{C}_\mathrm{N}$ is translated back to the RGB color space and the stroke is rendered onto the canvas, as illustrated in Fig.6d.

5 Results

For better illustration, and easier evaluation by the readers, instead of the semi-automatic template selection method described in Section 4, we have only used the templates displayed in Fig.7 for generating all results displayed in this paper. Each of these templates contains approximately 1000–1500 brush strokes. In our results, the non-face regions are rendered using the method of Zeng et al. [2009].

Fig.1 displays an example result generated using our method. The three portrait paintings are rendered with the templates displayed in Figs.7c, 7g and 7h, respectively. In these portraits, structures of facial parts are well preserved, although large brushes are used. In

Figure 8: *Our rendering results. The three images are rendered with templates shown in Figs.7c, 7g, and 7h, respectively. Zoom to 400% to view details.*

Figure 9: *Our rendering results. The three images are rendered with templates shown in Figs.7a, 7c, and 7f, respectively. Zoom to 400% to view details.*

particular, while this is a male face, Fig.1a is rendered with a female portrait template (see the rendering process illustrated in Fig.6) but the result is still satisfactory.

Figs.8 and 9 show experiments of our method on faces with dark skin colors. Most of these results are rendered with templates of much lighter skin colors, which shows that the color shift algorithm can preserve the color contrasts among strokes during global color changes.

Fig.10 displays a profile face example. The results reveal a small problem with the eyes. Some strokes depicting the eyeballs are not placed in the most appropriate positions. The main reasons for this include (1) our facial structure model does not cover such tiny parts, and (2) the TPS transformation we use is global and non-rigid which does not handle local deformations very well [Schaefer et al. 2006]. This problem is also observed in Fig.8b, in which we can notice some "ghost teeth" on the lips brought from the template, since we do not model the teeth or lips in the AAM. The same template is also used in Figs.1b, 11b, and 12c, but their symptoms are less obvious probably due to thinner lips.

Figs.11 and 12 compare our results with those generated by previous methods (see Fig.2). Clearly our results have stronger painterly effects, and depict more impressive 3D face structures.

6 Conclusion

In this paper, we have presented a novel method for painterly rendering of human portraits. This method takes advantage of both facial structure information and techniques of artists, by using previously painted portraits by artists as active templates, and transferring brush strokes from them to synthesize new portrait paintings for target photographs. Our example-based solution overcomes the challenges in balancing between large and decisive strokes for sharp 3D structures and our acute visual perception to human face, and in depicting faces with vibrant colors not present in original photographs. According to our experiments, this method can generate satisfactory results and provides a useful tool for painterly rendering of human portraits.

For future work, there are a few aspects in which we can improve the method.

- A richer and more accurate representation of facial structure will be helpful. The representation should cover facial parts such as eyeballs, teeth, muscles, and wrinkles, as well as drastic changes in the structure such as open vs. closed mouth or eyes, and frontal vs. profile face. A locally rigid shape deformation algorithm should perform better than TPS on this facial representation.

Figure 10: *Our rendering results. The three images are rendered with templates shown in Figs.7e, 7f, and 7i, respectively. Zoom to 400% to view details.*

Figure 11: *Our rendering results. The three images are rendered with templates shown in Figs.7e, 7g, and 7i, respectively. Zoom to 200% to view details.*

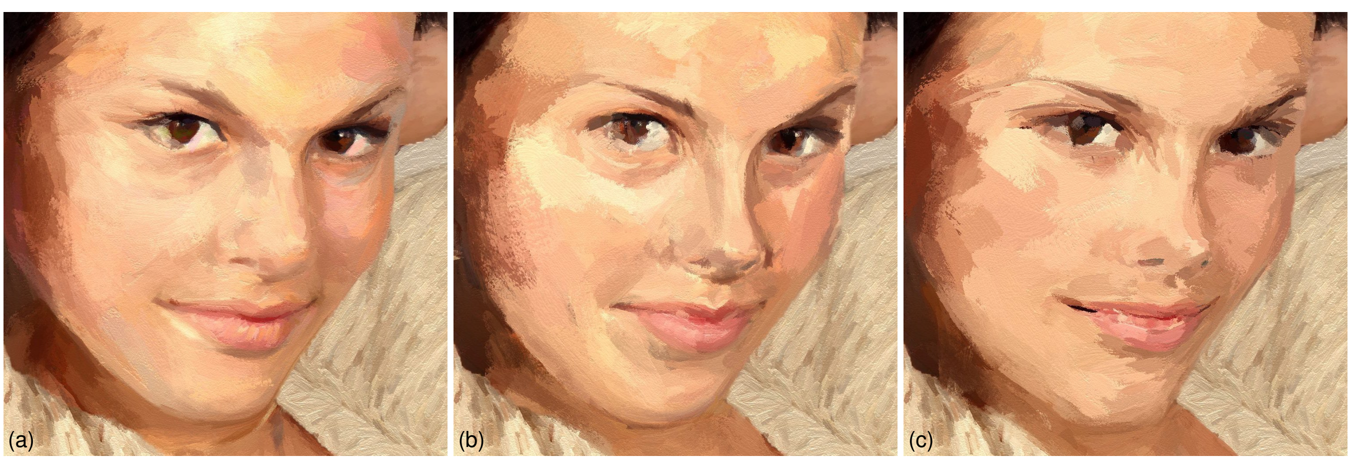

Figure 12: *Our rendering results. The three images are rendered with templates shown in Figs.7a, 7c, and 7g, respectively. Zoom to 400% to view details.*

- We expect a better color shift algorithm to make the colors more vibrant in the rendered portrait images, even if the colors in the selected template are not.
- A decomposition of the face into parts, along with a good recomposition algorithm, can greatly improve the power of the dictionary. Even using a very small dictionary, by combining parts from different original templates, we can generate many more new templates than the original ones painted by artists. To achieve this, the coherence among facial parts from different templates will be a topic for future study.

Project Website

More portrait paintings rendered using our method are available at *http://www.stat.ucla.edu/∼mtzhao/research/portrait-painting/*.

Acknowledgments

We are grateful to Yaling Yang and Xiaolan Ye for their assistance in the experiments. We thank Wenze Hu, Amy Morrow, Brandon Rothrock, Zhangzhang Si, Benjamin Yao, Yibiao Zhao, and the anonymous reviewers for nice suggestions on improving the presentation of this paper. The work at UCLA was supported by an ONR MURI grant N000141010933 and an NSF IIS grant 1018751, and the work at LHI was supported by two NSFC grants 60832004 and 90920009.

References

BARRODALE, I., SKEA, D., BERKLEY, M., KUWAHARA, R., AND POECKERT, R. 1993. Warping digital images using thin plate splines. *Pattern Recog. 26*, 2, 375–376.

BROOKER, S. 2010. *Portrait Painting Atelier: Old Master Techniques and Contemporary Applications*. Watson-Guptill.

CATMULL, E., AND ROM, R. 1974. A class of local interpolating splines. *Computer Aided Geometric Design*, 317–326.

CHEN, H., LIANG, L., XU, Y.-Q., SHUM, H.-Y., AND ZHENG, N.-N. 2002. Example-based automatic portraiture. In *Proc. ACCV '02*, 44–49.

CHEN, H., ZHENG, N.-N., LIANG, L., LI, Y., XU, Y.-Q., AND SHUM, H.-Y. 2002. Pictoon: a personalized image-based cartoon system. In *Proc. MM '02*, 171–178.

CHEN, H., LIU, Z., ROSE, C., XU, Y., SHUM, H.-Y., AND SALESIN, D. 2004. Example-based composite sketching of human portraits. In *Proc. NPAR '04*, 95–153.

CHOPRA, P., AND MEYER, J. 2003. Modeling an infinite emotion space for expressionistic cartoon face animation. In *Prof. CGIM '03*, 13–18.

COLLOMOSSE, J. P., AND HALL, P. M. 2002. Painterly rendering using image salience. In *Proc. EGUK '02*, 122–128.

COOTES, T. F., EDWARDS, G. J., AND TAYLOR, C. J. 2001. Active appearance models. *IEEE Trans. Pattern Anal. Mach. Intell. 23*, 6, 681–685.

DIPAOLA, S. 2007. Painterly rendered portraits from photographs using a knowledge-based approach. In *Proc. SPIE 6492, 649203*.

GOOCH, B., COOMBE, G., AND SHIRLEY, P. 2002. Artistic vision: painterly rendering using computer vision techniques. In *Proc. NPAR '02*, 83–91.

GOOCH, B., REINHARD, E., AND GOOCH, A. 2004. Human facial illustrations: Creation and psychophysical evaluation. *ACM Trans. Graph. 23*, 1, 27–44.

HAEBERLI, P. 1990. Paint by numbers: abstract image representations. In *Proc. SIGGRAPH '90*, 207–214.

HAYS, J., AND ESSA, I. 2004. Image and video based painterly animation. In *Proc. NPAR '04*, 113–120.

HERTZMANN, A. 1998. Painterly rendering with curved brush strokes of multiple sizes. In *Proc. SIGGRAPH '98*, 453–460.

HERTZMANN, A. 2001. Paint by relaxation. In *Proc. CGI '01*, 47–54.

LI, Y., AND KOBATAKE, H. 1997. Extraction of facial sketch image based on morphological processing. In *Proc. ICIP '97, Vol.3*, 316–319.

LIN, L., ZENG, K., LV, H., WANG, Y., XU, Y., AND ZHU, S.-C. 2010. Painterly animation using video semantics and feature correspondence. In *Proc. NPAR '10*, 73–80.

LITWINOWICZ, P. 1997. Processing images and video for an impressionist effect. In *Proc. SIGGRAPH '97*, 407–414.

LU, J., SANDER, P. V., AND FINKELSTEIN, A. 2010. Interactive painterly stylization of images, videos and 3D animations. In *Proc. I3D '10*, 127–134.

LUFT, T., AND DEUSSEN, O. 2004. Non-photorealistic real-time rendering of characteristic faces. In *Proc. PG '04*, 339–347.

MEIER, B. J. 1996. Painterly rendering for animation. In *Proc. SIGGRAPH '96*, 477–484.

MENG, M., ZHAO, M., AND ZHU, S.-C. 2010. Artistic paper-cut of human portraits. In *Proc. MM '10*, 931–934.

MIN, F., SUO, J.-L., ZHU, S.-C., AND SANG, N. 2007. An automatic portrait system based on and-or graph representation. In *Proc. EMMCVPR '07*, 184–197.

REINHARD, E., ASHIKHMIN, M., GOOCH, B., AND SHIRLEY, P. 2001. Color transfer between images. *IEEE Comput. Graph. Appl. 21*, 5, 34–41.

SCHAEFER, S., MCPHAIL, T., AND WARREN, J. 2006. Image deformation using moving least squares. *ACM Trans. Graph. 25*, 3, 533–540.

SINHA, P., BALAS, B., OSTROVSKY, Y., AND RUSSELL, R. 2006. Face recognition by humans: Nineteen results all computer vision researchers should know about. *Proc. IEEE 94*, 11, 1948–1962.

TRESSET, P., AND LEYMARIE, F. F. 2005. Generative portrait sketching. In *Proc. VSMM '05*, 739–748.

TURK, M. A., AND PENTLAND, A. P. 1991. Eigenfaces for recognition. *Journal of Cognitive Neuroscience 3*, 1, 71–86.

WANG, X., AND TANG, X. 2009. Face photo-sketch synthesis and recognition. *IEEE Trans. Pattern Anal. Mach. Intell. 31*, 11, 1955–1967.

ZENG, K., ZHAO, M., XIONG, C., AND ZHU, S.-C. 2009. From image parsing to painterly rendering. *ACM Trans. Graph. 29*, 1, 2:1–2:11.

ZHAO, M., AND ZHU, S.-C. 2010. Sisley the abstract painter. In *Proc. NPAR '10*, 99–107.