Synthesizing Non Photo-Realistic Rendering Effects of Volumetric Strokes

Article *in* Journal of Information Science and Engineering · May 2012

Lieu-Hen Chen
National Chi Nan University

JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 28, 521-535 (2012)

Synthesizing Non Photo-Realistic Rendering Effects of Volumetric Strokes

LIEU-HEN CHEN, YI-HSIEN CHEN, SHUO-YAN LIN, TING-YU LIU AND WEN-CHIEN HSIEH
Department of Computer Science and Information Engineering
National Chi Nan University
Nantou, 545 Taiwan

NPR (Non Photo-Realistic Rendering) is an important issue in Computer Graphics. It presents a totally different style from those using realistic methods. In this paper we take advantage of a 3D NPR system in color presentation, and apply stroke merging and sketch-based modeling to synthesize NPR images with volumetric effects, such as oil painting, bas-relief sculpture, and clay relief. To represent these volumetric visual effects on canvas we adopt a mesh generation method which generates the thickness of strokes. By controlling the ratio of height and width to the base area, the surface curvature is adjusted to reflect the shape differences between impasto oil painting and clay relief. Then a mesh smoothing method is applied to each group of strokes. In addition, users can personalize their own color palette by using keywords such as the name of the artist, or by choosing color sets through our GUI which categorizes color space into an emotional map. Users choose by using keywords, then related images are gathered from online resources using web-mining technology, and the appropriate colors are extracted to construct a color chart by analyzing the images. Then, the color chart is used to control the distribution of colors on the canvas. As shown in the experiment results, our 3D NPR system creates interesting and vivid images of oil painting and clay relief works. Finally, we integrate our volumetric NPR image with a relief mapping technique to provide real-time interaction.

Keywords: computer graphics, non photorealistic rendering (NPR), impasto oil painting, clay relief, stroke, pigment

1. INTRODUCTION

The colors, textures, and material properties of artwork can greatly affect the impression of a viewer. Different colors have different meanings, and can affect the mood of viewers. The variance of textures and material properties contribute to the beauty of a work of art and emphasize subtle changes of detail.

As artwork is not limited to two dimensional planar canvases and boards, volume becomes another important factor. Besides, the lines and curves that are formed by continuous brushstrokes can guide the path of a viewer's eye. Different impressions and information are created when artwork is viewed from different positions and angles, whether it is intentional or not (Figs. 1 (a) and (c)). Furthermore, under different illumination conditions, shadows provide many visual clues about the composition. Therefore, representing volumetric effects on artwork such as oil paintings (Fig. 1 (d)), crewel embroidery, and clay/metal relief works (Fig. 1 (b)), is an important and challenging issue that should not be ignored.

Received June 21, 2010; revised August 8, 2010; accepted August 24, 2010.
Communicated by Tong-Yee Lee.

Fig. 1. Crewel embroidery [14] artwork can give dramatically dynamic impressions under different lighting conditions; (a) Illuminated from the right side; (b) Shows a bas-relief on a coin; (c) Exactly the same artwork but illuminated from the left side; (d) Demonstrates how oil painting pigments build up on a canvas.

Non-Photorealistic Rendering (NPR) has been an important research topic in the field of Computer Graphics. Instead of highlighting the realism of CG synthesized images, NPR focuses on the representation of painting styles, such as generating strokes that appear to be hand-drawn. According to the method of inputting object data, the approaches to synthesizing NPR images can be roughly divided into two categories, *i.e.*, 2D image-based methods and 3D model-based methods.

In 2D image-based studies, the source information of a scene and objects mainly relies on static real images [1, 3, 9, 10, 16, 21]. One of the biggest problems with this method is that it is difficult to recognize and separate the subject of a painting from the background in a static image. Therefore, details of the principal parts are either ignored or overly magnified when they are created. It is also difficult to synthesize continuous NPR animation even with a sequence of real images is provided.

To synthesize 2D images with dynamic lighting effects, Oliveira proposed a relief texture mapping technique [12]. However, there is still no appropriate solution to represent the thickness of pigment or clay on canvas.

In 3D model-based studies [11, 13, 15, 19, 20], there is more information available for composition. Therefore, strokes generated in this way have more flexibility and realism. However, the thickness of each stroke on canvas is still ignored. This results in output images that are always represented as plane images without the characteristic textures of impasto oil painting.

In this paper we implement an NPR system that creates volumetric strokes that represent the thickness of pigment and clay on canvas. In order to do this, we use 3D model-based input sources and take into account the thickness of the pigment on the painting area. Using these methods, we can simulate volumetric visual effects of oil paintings and sculptures.

Also, we integrate personalized color charts based on web-mining technologies with our NPR system. A color chart is the basis of a color design. By adopting personalized color charts with an interactive interface, our system allows users to accurately arrange tints and colors that express their feelings. Finally, we integrate our volumetric NPR image with relief mapping techniques to perform real-time interactions between the light source and objects.

2. VOLUMETRIC STROKES SYNTHESIS

Most existing NPR systems focus on creating 2D images on a flat canvas that appear to be drawn by hand. This creates a situation where the volumetric strokes of oil paintings and reliefs are flattened. The interesting light and shade effects of volumetric strokes under changing illumination conditions are consequently ignored. Therefore, we integrated the mesh-generation method of Gallarotti [6] and Igarashi [17, 18] with our stroke synthesizing methods [7, 8] to generate volumetric strokes. The main flow chart of generating volumetric strokes is shown in Fig. 2. These can be divided into three parts: Pre-Processing, Stroke Calculating, and Volumetric Strokes Generating. We will discuss these in the following paragraphs.

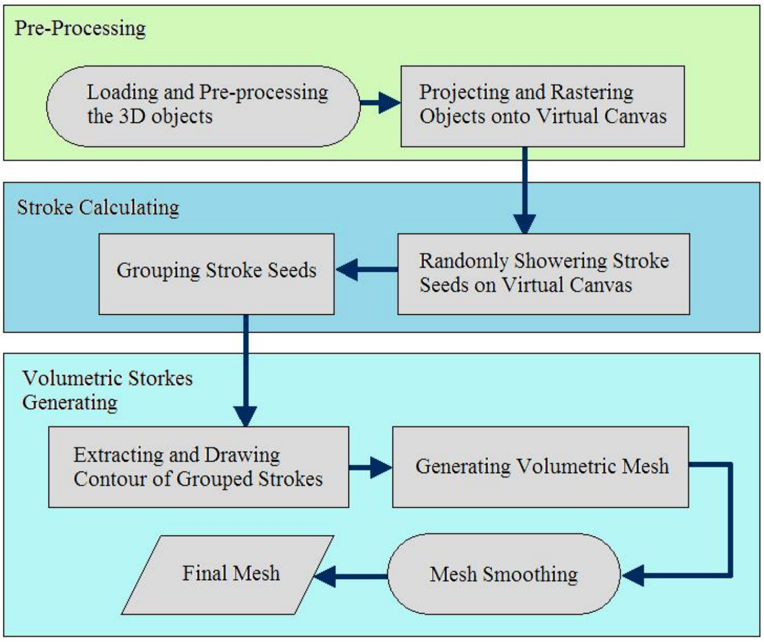


Fig. 2. Flowchart for generating volumetric strokes.

2.1 Pre-Processing

In our NPR-System, we collect all vertices in a given scene into a Vertex List, and create a Triangle List to manage the group relations and triangle characteristics. It includes an index, as well as group and spine information. After dealing with triangle information, we store the Color, Normal Vector, and Stroke Vector information of the triangle into a virtual canvas by performing Projection, Rasterization, and Z Depth-testing operations [7]. This virtual canvas is used in the next phase of painting.

2.2 Stroke Calculating

Scenery in painting is usually composed of many different objects. According to the distance and color of every object, different artists draw individual strokes in different ways. The one commonality in the technique is that most strokes tend to follow the direction of the nearest contour line of an object. To simulate this painting technique, we search for the nearest contour line for each stroke, and then assign the tangent direction of this contour line to the stroke vector [7].

The seeds of strokes are randomly distributed throughout the canvas. These seeds are shown as white points in Fig. 3. The lines extending from the white points show the color and direction of the stroke. Each stroke is generated from a seed and follows the stroke vector bi-directionally.

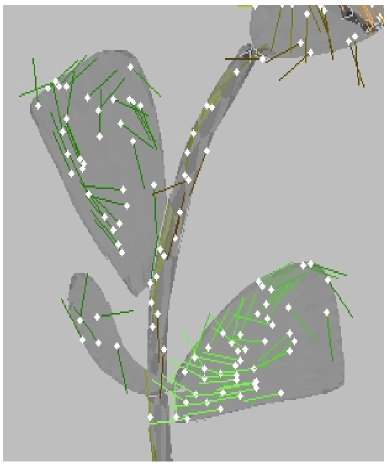


Fig. 3. Randomly showering stroke seeds.

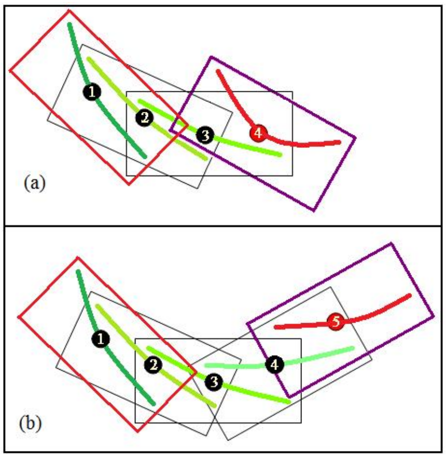


Fig. 4. Searching seeds for grouping.

While each artist may have an individual method, the common method is to repeatedly draw strokes from the outside in. However, the random seed has no order, so we use the inverse of the distance to determine its weight. This distance is calculated from every seed to its nearest silhouette. In other words, the closer to the silhouette the stroke is, the higher its weight will be.

In real painting, a painter may use larger brushes as needed to create different effects. To simulate this effect, we merge seeds with those that are near enough and have similar stroke directions into a larger stroke [8]. This merging step is summarized as follows:

First, we sort the seeds by their weight, and then cluster them into subgroups by checking them one by one, from large to small. At each step, the current seed serves as the center of a search window, and the distances from the center seed to other unchecked seeds are calculated. The search window is a rectangle consisting of the golden ratio of 1.618 of width to height. The nearest seed inside this rectangle is grouped with the center seed, if the included angle between these two strokes is within the threshold. For example, in Fig. 4 (a), V_4 is excluded because the included angle between V_3 and V_4 exceeds the threshold. Then, this nearest seed is excluded from the check list and used as the center of the next searching window. This process is repeated until the rectangle fails to overlap the initial one. For example, in Fig. 4 (b), V_1, V_2, V_3, and V_4 are grouped, but V_5 is excluded. By controlling the size of the grouping rectangle, different shapes of merging strokes can be created.

2.3 Volumetric Stroke Generating

2.3.1 Extracting boundary information of merged strokes

When the strokes of a group are all determined, a mathematical morphology closing method is adopted to obtain a smoother boundary of the target group [8]. This boundary information from each group is then extracted (Fig. 5). By controlling the size and shape of grouped strokes, we can mimic the shape characteristics of different painting styles.

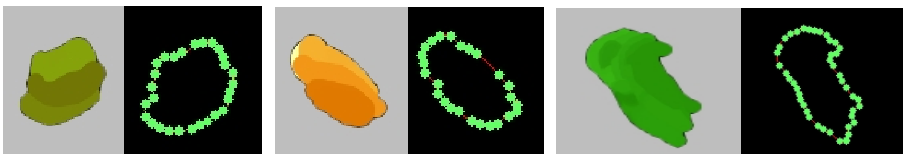


Fig. 5. Merging strokes into a group and extracting the boundary information.

2.3.2 Triangulation and refinement

Boundary information is then used to construct a rough volume based on Igarashi's method [17]. First, the boundary information is input into the Constrained Delaunay Triangulation – CDT modular. All boundary paths serve as the constraint edges of CDT for triangulation. All triangles are then classified to calculate the spine. The classification rules are listed in Table 1. After all triangles are classified, we link a spine line through every triangle using the rules in Table 2. The result of classifying triangles and constructing spine are illustrated in Fig. 6.

All remaining type *S* triangles that are intersected by a spine line are then subdivided into three sub-triangles. All of these triangles have two inner edges connected between the spine and boundary lines. After type *S* and *T* triangles are subdivided and replaced, respectively, all triangles are reclassified following the rules listed in Table 1.

Finally, the result of triangulation is further amended by applying an edge swapping operation as suggested by T. Y. Lee *et al.* in [20]. This is done by iteratively performing a

Table 1. Triangle classification rule.

Triangle Type	Characteristics
T – Terminal	A triangle with two boundary edges.
S – Sleeve	A triangle with one boundary edge.
J – Junction	A triangle without boundary edge.

Table 2. Spine connection rule.

Triangle Type	Action
T	No action.
S	Connect two centers of non-boundary edges.
J	Connect triangle center to non-boundary edge center.

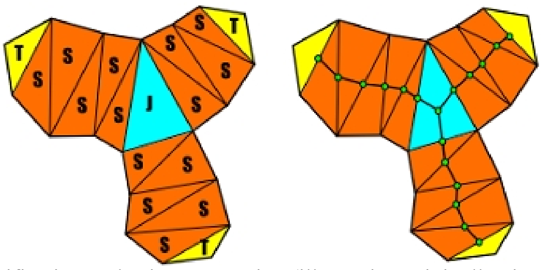


Fig. 6. Triangles classification and spine connection (illustration originally cited by Igarashi in [17]).

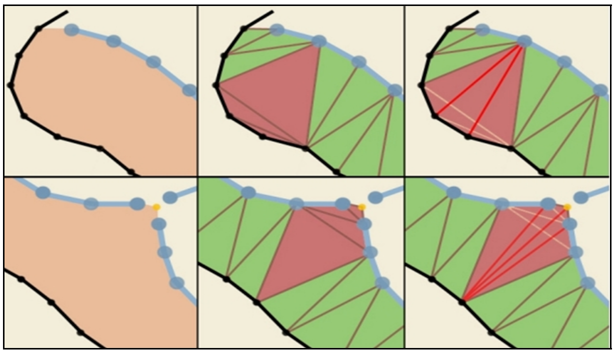
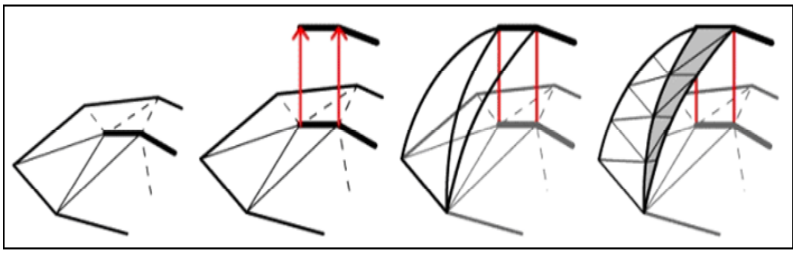


Fig. 7. The upper row traverses outward. The lower row traverses inward. Lines linked through the small vertices represent the boundary. Thicker lines linked through big vertices create a spine (illustration originally cited by T. Y. Lee *et al.* in [20]).

triangle traversal that starts from a type J triangle. Initially, the traversing direction is determined according to the relative position of the spine vertices and boundary vertices on the triangle (Fig. 7). Visited triangles are then stored in a queue. While traversing inward, the operation is stopped when any type J triangle vertex is encountered. While traversing outward, the operation is stopped when a terminal triangle is met. Finally, all triangles in the queue are removed and edge-swapped. This edge swapping operation is repeated until no type J triangles exist, or there is no edge that can be swapped.

2.3.3 Generating volumetric meshes

The next step is to elevate the spine vertices, following the method proposed by Igarashi [17, 18]. We take the average distance of every boundary vertex around the spine vertex, and use these distances as the height of the spine vertex (Fig. 8). However, as pointed out by Gallarotti [6], directly using the average distance as height may cause a jagged spine line. As shown in Fig. 9, the red vertices on the spine are connected to the vertices on the boundary in a different way as compared to the way in which the blue vertices on the spine connect. Consequently, lines connected to red vertices are longer. Therefore, the spine line may suffer from a jagged appearance.



(a) Before. (b) Elevate spines. (c) Elevate edges. (d) Sew elevated edges.

Fig. 8. Elevating spine lines (illustration originally cited by Igarashi in [18]).

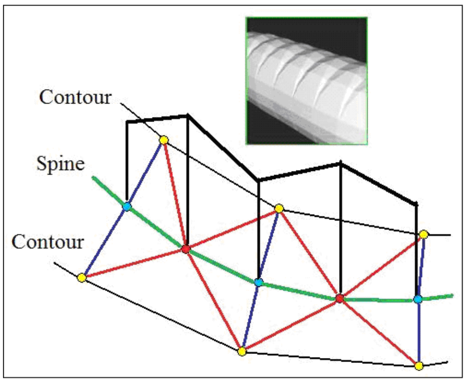


Fig. 9. Jagged spine.

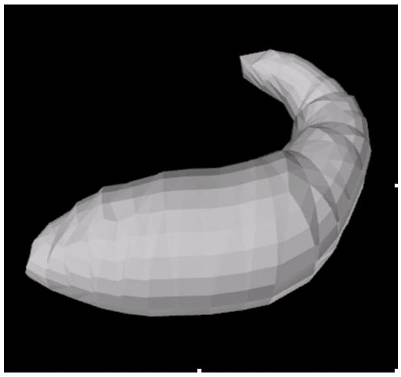


Fig. 10. The spine after amendment.

To deal with this problem, Gallarotti proposed a method that changes the height of the blue point by the average distance of incident points to the boundary vertices [6]. But this method needs another traversal to complete. In our system, we adopt a simple and direct approach for calculating the displacement: If we find that the height of a spine vertex is lower than its neighboring spine vertices, then the interpolated value of the height of the neighbors is used. As shown in Fig. 10, the jagged surfaces are smoothed down.

After the spine lines are elevated, we interpolate the height of all edges connected to distance calculated, and the height of the boundary is zero. We separate the path from the boundary to the spine into many parts and then interpolate the data points. While we interpolate, color information is added into the data point based on the origin stroke of the relative coordinates on the canvas. The normal vector of every vertex is also calculated to make sure the generated mesh matches the brush stroke.

The thickness of oil painting pigment on a canvas is relatively lower than the surface height on a relief. Therefore, the maximum height of a spine in a relief is approximately fifty times higher than it would be in a similar oil painting.

Finally, we perform the umbrella operator method used by Belyaev [2] iteratively and progressively to further smooth the shapes of the volumetric strokes:

(1) $P_{new} \leftarrow P_{old} + \lambda U(P_{old})$,
(2) $U(P) = (\Sigma_i Q_i)/n - P$.

In statement (1), the offset range after smoothing is proportioned to the value of vector λ. In this paper, λ is set to (1.2, 1.2, 1.0). The amendment range of any point p is related to the average distance of p to its neighboring points, Q_n.

3. ARRANGING THE TINTS AND COLORS

Arranging tints and colors is an important issue, not only for professional artists and designers, but also for amateurs and beginners. Even subtle changes in colors and shades can affect the emotions and impressions of viewers. It is useful and practical to provide an intuitive mapping function between colors and the feelings of users. By using color charts, colors can be accurately determined by their color value and hue. In addition, for color designing, using color charts can avoid any visual errors that may be caused by misunderstanding and intermingling colors.

Unfortunately, conventional color charts are designed and represented in 2D. It is very difficult for users to survey the whole color model and choose colors intuitively. Therefore, in this paper we integrate our volumetric NPR system with personalized 3D color charts based on web-mining technologies [11]. The possible meaning of colors is extracted by adopting a web-mining technique and 160 intuition tables of emotions.

Figs. 11 and 12 illustrate our visualization system of 3D color charts. The red circle in the middle of Fig. 11 indicates the 3D color charts. The yellow text in the upper left circle of Fig. 12 which introduces the information about the author is obtained through web-mining techniques.

In our system we construct the user-customized color charts by performing the following procedures as shown in Fig. 13. First, users can directly define their current psychological state by using a 3D emotion map, or indirectly by choosing keywords such as the name of an artist or styles of painting from the tables. Then, the Keronsoft Web Pictures Searcher is adopted to automatically search related images from the web. Colors are extracted to construct the color chart by analyzing the histograms of these images. Finally, the color chart is rendered in a 3D visualization system which allows users to interactively view and manage the distribution of colors.

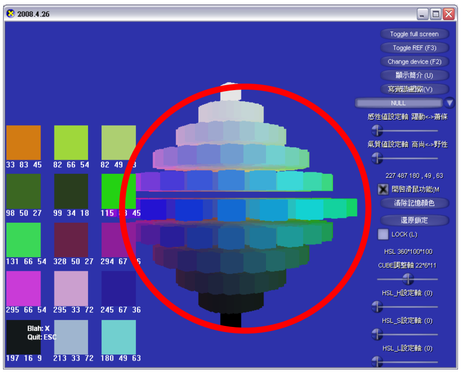


Fig. 11. 3D color charts.

Fig. 12. Artists' color combinations.

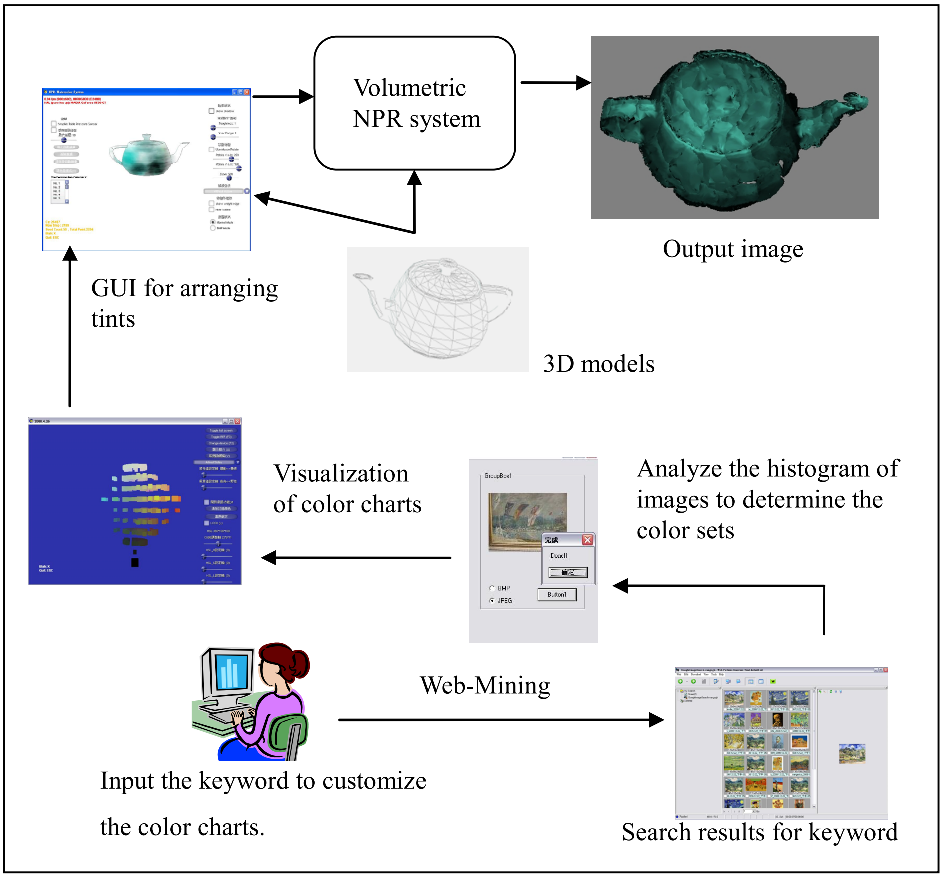


Fig. 13. Flowchart of the user-customized 3D color charts.

4. EXPERIMENTAL RESULTS

The following section demonstrates the results of our experiment. We render the volumetric NPR images of a flower, a teapot, and an ox to simulate impasto oil painting and clay/bronze relief effects. For shading volumetric strokes, the Gouraud shading method is adopted. The height value of volumetric strokes in these images is listed in Table 3 using the width ratio as the baseline.

Figs. 14 to 17 show the rendering result of a flower model. Figs. 14 and 15 are viewed at the same position and angle. In Fig. 16, the tint shifts toward the energetic axis on an emotional map. In contrast, the tint of Fig. 17 shifts toward the depressed axis. A point white-light source is located at the left back side of the camera at the same height for Figs. 14 and 15, and overhead of the camera for Figs. 16 and 17.

Table 3. Height of volumetric strokes in the result images.

Rendered Model	Flower (oil painting)	Teapot (relief)	Ox (relief)
Average Height	0.001425	0.08944	0.082653
Maximum Height	0.0032	0.1494	0.1992

Fig. 14. NPR image with planar strokes.

Fig. 15. Volumetric NPR image (oil painting).

Fig. 16. Flower in energetic mood.

Fig. 17. Flower in depressed mood.

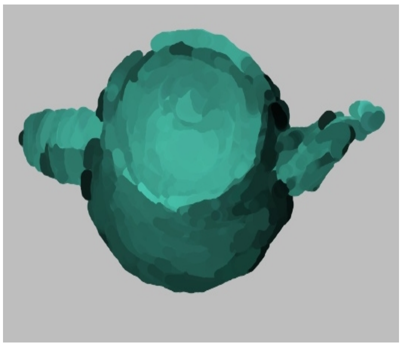


Fig. 18. NPR image with planar strokes.

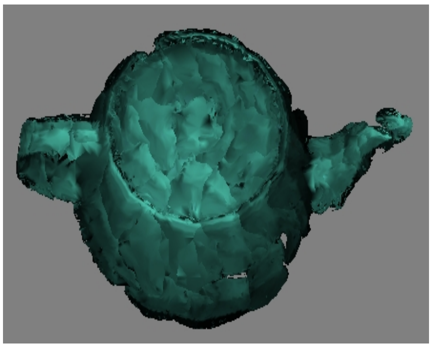


Fig. 19. Volumetric NPR image (clay-relief).

Fig. 20. Volumetric NPR image (bronze relief).

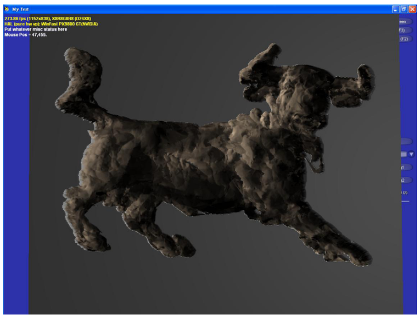


Fig. 21. Volumetric NPR image (clay relief).

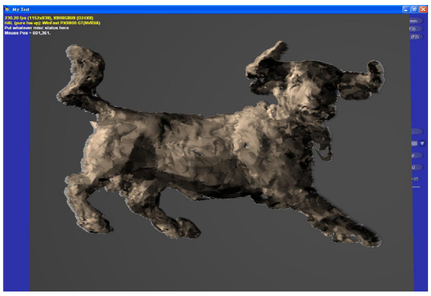


Fig. 22. Volumetric NPR image (clay relief).

Fig. 23. Volumetric NPR image (stone relief).

Figs. 18 and 19 show the rendering results of a teapot model. Fig. 18 is a conventional 2D NPR image. Fig. 19 is our volumetric NPR image which simulates a clay relief of tea pot.

Finally, to further accelerate the rendering speed, we integrate a relief mapping method into our system [4, 5, 12]. By directly converting the volumetric information of strokes, relief maps can be automatically generated. Adopting relief mapping allows the intermediate images to be smoothly rendered in real-time, and enables users to manipulate and interact with the light source and scene. Also, the map displacement problem of relief mapping is improved by adopting the length normalization approach proposed by C. F. Chang [22]. Figs. 20-23 are rendered by relief mapping. The position of a light source in these figures is determined by the cursor on a mouse. Fig. 20 illustrates a volumetric NPR image of a bronze relief. Figs. 21 and 22 are volumetric NPR images of a clay relief. These images are captured from a continuous image sequence while the user moves the cursor from left to right. Fig. 23 illustrates an example of a stone relief.

5. CONCLUSION AND FUTURE WORK

In this research, we integrated a 3D model-based NPR system with an approach using volumetric stroke generation. Furthermore, users can manipulate tints and colors by using customized 3D color charts based on web-mining technology. After users input the 3D model data, the system can output animation using volumetric strokes effects that simulates impasto oil painting or clay relief techniques. In comparison with conventional 2D planar images, this 3D animation provides users and viewers with a realistic and stereoscopic method of interaction.

To simulate oil paint, we adopt a method similar to squeezing paint directly onto a canvas, but this simulated method does not conform to a real physical model. More precise dynamics are needed to improve the simulation, such as the amount of the pigment, the moisture content of the pigment, or the pressure on the brush in order to get a more accurate painting result. Another challenge for future work is to simulate the anisotropic optical effects of crewel embroidery. Light interacts with the highly concentrated colorful fibers of embroidery in a very complicated way. This complex interaction introduces a further area for research and refinement.

The current version of GUI only allows users to rearrange the tints and colors on a canvas. We are working to develop a new GUI that allows the user to manipulate the thickness of a stroke through a direct "push or pull" application within the interface. By overcoming these problems, we aim to facilitate user friendly tools that assist in the creation of NPR artworks.

REFERENCES

1. A. Hertzmann, "Fast paint texture," in *Proceedings of the 2nd International Symposium on Non-Photorealistic Animation and Rendering*, 2002, pp. 91-96.
2. A. Belyaev, "Mesh smoothing and enhancing curvature estimation," http://www.mpi-inf.mpg.de/˜ag4-gm/handouts/06gm_surf3.pdf.
3. C. R. Yan, M. T. Chi, T. Y. Lee, and W. C. Lin, "Stylized rendering using samples of a painted image," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 14, 2008, pp. 468-480.
4. D. Bourguignon, R. Chaine, M. Cani, and G. Drettakis, "Relief: A modeling by drawing tool," in *Proceedings of the Eurographics Workshop on Sketch-Based Interfaces and Modeling*, 2004, pp. 151-160.
5. F. Policarpo, M. M. Oliveira, and J. Comba, "Real-time relief mapping on arbitrary polygonal surfaces," in *Proceedings of ACM SIGGRAPH Symposium on Interactive*

3D Graphics and Games, 2005, pp. 155-162.
6. F. Gallarotti, "Constructing 3D geometry through user sketch," Course Final Project, State University of New York at Stony Brook, 2004.
7. J. Lin, "Shape-oriented brush stroke synthesis in non-photorealistic rendering," Master Thesis, Department of Computer Science and Information Engineering, National Chi Nan University, 2006.
8. J. L. Yang, "Simulating the oil painting in an NPR 3D animation system," Master Thesis, Department of Computer Science and Information Engineering, National Chi Nan University, 2007.
9. J. Henry, M. Hajime, H. Toshiyuki, D. Yoshinori, and N. Tomoyuki, "A method for creating region-based and stroke-based artistic images," *The Institute of Electronics, Information and Communication Engineers Transactions*, Vol. J88-D-II, 2005, pp. 358-367.
10. K. Zeng, M. Zhao, C. Xiong, and S. C. Zhu, "From image parsing to painterly rendering," *ACM Transactions on Graphics*, Vol. 29, 2009, pp. 1-11.
11. L. H. Chen, Y. H. Ho, T. Y. Liu, and W. C. Hsieh, "A watercolor NPR system with web-mining 3D color charts," *Arts and Technology*, Vol. 30, 2010, pp. 72-79.
12. M. M. Oliveira, G. Bishop, and D. McAllister, "Relief texture mapping," in *Proceedings of the 27th International Conference on Computer Graphics and Interactive Techniques*, 2000, pp. 359-368.
13. M. T. Chi and T. Y. Lee, "Stylized and abstract painterly rendering system using a multi-scale segmented sphere hierarchy," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 12, 2006, pp. 61-72.
14. Q. F. Wu, "Majestic mountain in clouds," 55cm × 76cm crewel embroidery, 1977.
15. R. D. Kalnins, L. Markosian, B. J. Meier, M. A. Kowalski, J. C. Lee, P. L. Davidson, M. Webb, J. F. Hughes, and A. Finkelstein, "WYSIWYG NPR: Drawing strokes directly on 3D models," in *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, Vol. 21, 2002, pp. 755-762.
16. S. Brooks, "Mixed media painting and portraiture," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 13, 2007, pp. 1041-1054.
17. T. Igrarashi, S. Matuoka, and H. Tanaka, "Teddy: A sketching interface for 3D freeform design," in *Proceedings of the 26th International Conference on Computer Graphics and Interactive Techniques*, 1999, pp. 409-416.
18. T. Igarashi and J. F. Hughes, "Smooth meshes for sketch-based freeform modeling," in *Proceedings of ACM Symposium on Interactive 3D Graphics*, 2003, pp. 139-142.
19. T. Y. Lee, C. R. Yan, and M. T. Chi, "Stylized rendering for anatomic visualization," *IEEE Computing in Science and Engineering*, Vol. 9, 2007, pp. 13-19.
20. T. Y. Lee, S. U. Yan, Y. N. Chen, and M. T. Chi, "Real-time 3D artistic rendering system," in *Proceedings of International Workshop on Intelligent Information Hiding and Multimedia Signal Processing*, Vol. 3683, 2005, pp. 456-462.
21. W. V. Baxter, J. Wendt, and M. C. Lin, "IMPaSTo: A realistic, interactive model for paint," in *Proceedings of the 3rd International Symposium on Non-Photorealistic Animation and Rendering*, 2004, pp. 45-148.
22. Y. C. Chen and C. F. Chang, "A prism-free method for silhouette rendering in inverse displacement mapping," *Computer Graphics Forum*, Vol. 27, 2008, pp. 1929-2936.

Lieu-Hen Chen (陳履恆) is currently an Associate Professor of the Department of Computer Science and Information Engineering at National Chi Nan University, Taiwan. His research interests include Computer Graphics and Digital Arts. He received a B.S. in Computer Science and Information Engineering from National Taiwan University, and his M.S. and Ph.D. degrees in Electrical and Electronic Engineering from the University of Tokyo.

Yi-Hsien Chen (陳沂顯) received his M.S. degree in Computer Science and Information Engineering from National Chi Nan University in 2009.

Shuo-Yan Lin (林碩彥) received his M.S. degree in Computer Science and Information Engineering from National Chi Nan University in 2010.

Ting-Yu Liu (劉亭瑜) received her M.S. degree in Computer Science and Information Engineering from National Chi Nan University in 2009.

Wen-Chien Hsieh (謝文婕) received her B.S. degree in Computer Science and Information Engineering from National Chi Nan University in 2010.