THE SYNTHESIS OF TREES IN CHINESE LANDSCAPE PAINTING USING SILHOUETTE AND TEXTURE STROKES

Der-Lor Way[1,2] , Yu-Ru Lin[1] , Zen-Chung Shih[1]

[1]Department of Computer and Information Science, National Chiao Tung University
1001 Ta-Hsueh Rd, Hsinchu, Taiwan 30010,R.O.C.
Email: adler, zcshih@cis.nctu.edu.tw
[2]Department of Information Management, Van Nung Institute of Technology.
Email: adler@cc.vit.edu.tw

ABSTRACT

Practiced for more than three thousand years, Chinese painting emphasizes "implicit meaning", and involves painters' using a minimal number of brush strokes to express their deepest feelings. Landscapes are one of the most important themes in Chinese painting. Trees are the essential painting objects. This paper presents a set of novel methods to automatically draw trees in Chinese ink painting from 3D polygonal models. Outline rendering and texture generation uses the information of the silhouette, shade and orientation of three-dimensional model's surface to draw a particular tree. Four reference maps are established to analyze the information for the bark texture. These methods can draw various styles of bark texture by defining the texture patterns. Finally, this paper demonstrates some results obtained with our method.

Keywords: Non-Photorealistic Rendering (NPR), Chinese Landscape Painting, TS'UN, Texture Strokes, Silhouette, Curvature Map, Brush, and Texture Generation.

1. INTRODUCTION

Non-photorealistic rendering (NPR) injects aesthetic and stylized element into computer graphics. NPR primarily aims to convey not only compressed information but also a certain artistic spirit, by simulating various artistic styles such as pencil sketching, watercolor painting, oil painting, or Chinese ink painting. This approach differs from photorealistic rendering, which involves "creating synthetic images that are so lifelike they might be mistaken for photographs of real world scenes and objects" [Veryovka99].

Simulating the style of Chinese ink painting is not trivial. Such painting typically uses brushes and ink, and values the expression of the artistic conception far beyond precisely conveying the appearance of the painted subjects. A painter communicates her frame of mind to the viewers by blending the effects of brushes and ink [Chow79, Mei88, Liu84].

This paper presents a set of methods for drawing automatically three-dimensional trees in the style of Chinese ink painting. Trees are considered because they have been recorded as one of the most important subjects in Chinese landscapes painting [Lee97]. Figure 1 indicates the importance of trees in a Chinese landscape painting. Moreover, trees typically possess more complex geometry than other objects, such as mountains and rocks.

Figure 1: A Chinese landscape painting by Gong-Wang Huang

Figure 2 describes the purpose of the work. The painting algorithm will automatically render a tree (Fig-2b) in the style of Chinese ink painting when input a polygonal tree model (Fig-2a).

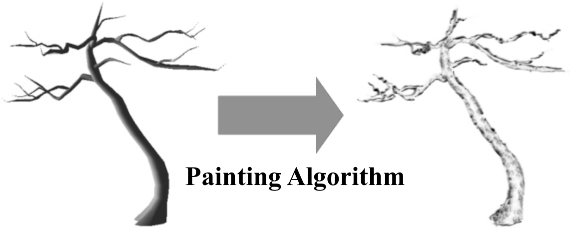


Figure 2: (a) polygonal models. (b) Chinese ink painting style.

The first requirement is to address the expressive brush strokes. The painting algorithm integrates the brush model presented by Der-Lor Way et al. [Way2001a, Way2001b] into a brush stroke handler. The algorithm is based on the traditional flow of Chinese ink painting.

The rest of this paper is organized as follows. Section 2 reviews works related to non-photorealistic rendering. Section 3 details outline drawing. Section 4 introduces the mechanisms of texture generation. Section 5 displays results including still images and animations. Finally, section 6 concludes and discusses future work.

2. PREVIOUS WORK

NPR is attracting increasing attention. Green has carried out much research in this field [Veryovka99]. We classify this body of research into three categories : object outline, texture, and brush and ink.

2.1 Object Outline

Drawing the outline of an object is critical in NPR, as it captures the first impression of shape of an object. An outline can be obtained in two ways [Hertz99]:

1. Object-based method [Hertz2000, Hertz99, Elber99, North2000, Lee97]:

This approach determines the silhouette in object space by finding the edges between the front and back faces. Markosian et al. [Lee97], and Hertzmann and Zorin [Hertz2000] presented algorithms to accelerate the location of silhouettes. The information concerning the edges enables strokes to be drawn.

2. Image-based method [Hertz99, Michael97, Veryovka99, Raskar99, Saito90, Corrêa98]:

This approach identifies the outlines in image space by rendering images in various ways and then applies a post-process to determine the outline pixels. However, the discrete pixels can't easily be linked as strokes.

The approach presented here for finding and drawing outlines uses the object-based method, and presents a novel algorithm to link outline pixels to create the required brush strokes.

2.2 Texture

Many factors are considered in generating the desired texture, such as orientation, distribution, view-dependency, and painting-view preservation of textures.

1. Orientation:

The texture is follows the orientation of the surface, to convey the 3D form of the surface. Salisbury et al. [Michael97] combined the user-defined direction field, stroke example set, and grayscale target image to create an orientable texture. The warped texture was created by Corrêa et al. [Corrêa98], by warping the 3D model to match the shape of the line art, and then rendering the model. Girshick et al. [Girsh2000] noted that the "principal directions of curvature" might communicate the surface shape better than lines in another direction. However, the aforementioned approaches, employing the curvature directions, are all based on parametric surfaces, since computing a polygonal surface is more difficult. Turk [Turk92] offers a curvature approximation.

2. Distribution

The distribution or density of a texture must be controlled to emphasize the shading of objects. The object's desired tone requires a dense distribution of textures. Similarly, Kowalski et al. [Michael99] measured the "desire" to place the "graftals", and used the "ID reference image" to obtained the triangle that corresponds to a given screen position. Hertzmann and Zorin [Hertz2000] used many user-tunable parameters to control the hatch density. These previous works inspire us to define the probability of texture appearing of each pixel to control the screen-space density.

3. View-dependency

An example of view-dependent textures is the use of fewer or smaller textures when further from the camera. Related discussions can be found in [Lee2000, Michael99].

4. Painting-view preservation

Katanics [Katanics] proposes a rendering scheme that preserves, for an arbitrary view, the look and style of features in the painted view. We consider that his scheme is reasonably suited to NPR, which simulates the painting of artists.

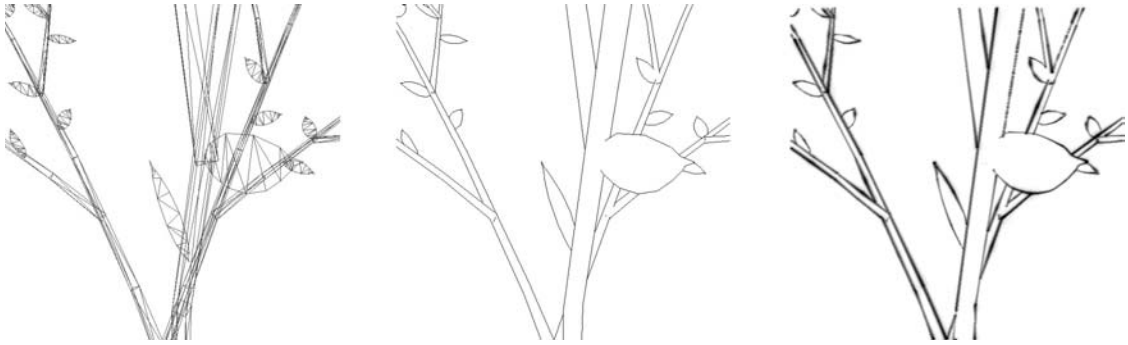


Figure 3: (a) Wire Frame (b) Silhouette Edge (c) Brush Stroke

2.3 Brushes and Ink

The brush strokes hold the secret of oriental writing and painting. Zhang et al. [Zhang99] presented a simple behavior model of water and ink particles, based on 2D cellular, automaton computational model to simulate the complex interaction between brush, ink and paper, and such visual features as shade, scratchiness, and blur. The "soft" brush of Lee [Lee99] responds elastically to the force exerted by an artist against the paper, and uses a polygon-shading rendering method.

The brush model proposed by Der-Lor Way et al. [Way2001a, Way2001b] can create the desired effects of brush strokes, and is integrated into the painting algorithm presented here. The brush model includes two mechanisms : stroke geometry and brush profile. The stroke geometry mechanism controls the path of a stroke, and the brush profile mechanism determines the various effects of ink deposition, such as darkness, wetness, and pressure, etc.

3. OUTLINE DRAWING

The outline, or silhouette, of a shape is a typically dominant feature. This research attempts to render attractive silhouette outlines for 3D geometry, creating brush-strokes along well-chosen paths around each object. The rending involves three distinct phases. First, the location of silhouette edges is determined. Second, links these silhouette edges into a long path. Third, each brush stroke is drawn in a style defined by the user.

The silhouette set for the smooth surface is the set of points P of the surface such that $(n(P) \cdot (P - C)) = 0$, where C is the viewpoint, n(P) is the normal vector of P. The silhouette is a union of flat areas, curves and points on the surface. However, the silhouettes of smooth surfaces are significantly different from those of their approximating polygonal meshes. For polygonal meshes, if the viewing vector (V) is defined as a vector from the viewpoint (C) to the viewing plane, then a front facing polygon is identified by the sign of the dot product of N (the polygon normal n(P)) and V. If the dot product $N \cdot V > 0$, then the polygon is front-facing; if $N \cdot V < 0$, then the polygon is back-facing, and if $N \cdot V = 0$, then the polygon is perpendicular to the viewing direction.

A silhouette edge is defined as one that connects a front-facing triangle to a back-facing triangle. The edge can also be a silhouette edge if it is not connected to another triangle (for example, the edge of leaf). Then, visibility and adjacency are computed from a 2D projection of the silhouette edges. The next step links these silhouette edges into long chains, or paths, that will form the basis of the brush strokes. These silhouette edges must be computed whenever the view changes. Figure 3 details the procedure for outline drawing. Figure 3(a) is a wireframe of the tree model. As for the above method, Fig 3(b) extracts the silhouette edge from Fig 3(a). Finally, brush strokes are applied to yield Fig 3(c).

4. TEXTURE GENERATION

Textures express the features of an object's surface. In Chinese painting, the process of texture generation is correctly known as "TS'UN", as meaning wrinkle. The variety of bark textures can be roughly classified into four major categories [Lu90]: (1) Straight texture: pine, willow, banyan, etc. (2) Scale-like (circular) texture: pine, etc. (3) Horizontal texture: firminia, etc. (4) Slant texture: cypress, poir, etc.

As described in Section 2.2, four important factors must be considered when generating the desired texture strokes.

1. Orientation

Bark textures in Chinese ink painting are formed by clusters of directed strokes. Therefore, a mechanism is required to determine the orientation of textures. Improving the surface depth and

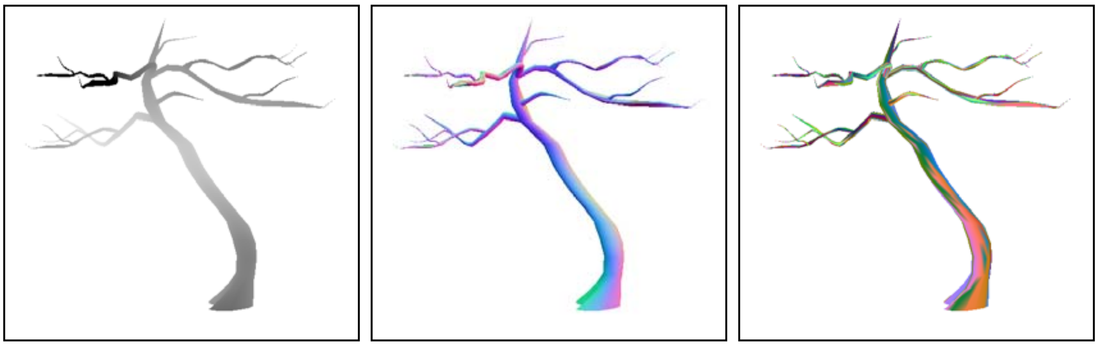


Figure 4: (a) Depth map. (b) Normal map. (c) Curvature map.

orientation perception is preferred, when directed textures depend on surface geometry. A curvature map is constructed to automatically suggest the orientation of textures.

2. Distribution

The distribution describes where to place the texture patterns. The density of texture patterns should be higher when closer to the silhouette, or dark areas, to emphasize the shading of objects. The relationship among texture patterns must also be considered, because different textures may exhibit different kinds of relationships. For example, texture patterns for simulating scale-like textures should be placed tightly around their neighboring patterns, whereas texture patterns for straight or horizontal textures may overlap their neighbors. The probability of texture appearance of each pixel is defined to control screen-space density, by computation from reference maps, including depth and normal maps.

3. View-dependency

Textures should be view-dependent to capture the distance from the camera. For example, fewer or smaller texture patterns are needed to draw a surface farther away. Restated, zooming can influence textures. View-dependent textures are realized by adjusting texturing parameters with camera distance.

4. Painting-view preservation

Even if the object is rendered in perspective, brush strokes that preserve their look and style in the painted view, should be generated. On the other word, brush strokes cannot be distorted or truncated. The curvature and profile of brush strokes should be preserved during texture mapping. Two-dimensional texture mapping is used to overcome the topological and parametric distortion hurdles. The texture mapping method should exhibit view-dependency and painting-view preservation. Different styles of texture must be defined procedurally to control locally the shape and shade of a texture pattern.

4.1 Reference Maps

Reference maps are first constructed to analyze the geometry and shading of a surface and thus determine orientation and distribution. At least four reference maps are used. (1) Depth map (MAPd). (2) Normal map (MAPn). (3) Curvature map (MAPc). (4) Object ID map (MAPoid).

Computing the gradient of the depth map (Fig. 4(a)) reveals the large variation in depth between adjacent pixels, such that C^0 surface discontinuities are detected. However, neither boundaries between objects of the same depth, nor creases are detected. Thus, the gradient computation of the normal map is combined with depth map to detect C^0 and C^1 discontinuities in the image, as illustrated in Fig. 4 (b). OpenGL is used to construct the depth map and the normal map. The intensity of a pixel in a depth map is proportional to the depth of that point in the scene. The normal map is an image that represents the surface normal at each point on an object.

The curvature measures the deviation of a curve from a straight line. It should be large where the curve wiggles, oscillates, or suddenly changes direction and should be small where the curve is nearly a straight line. Having an exact measure of curvature of an object would be ideal. However, such information is unavailable from a polygonal object because the object is not explicitly represented. The curvature approximation of Turk [Turk92] is used to approximate surface curvature from the polygonal data. After the first principle curvature is determined at each vertex of the object, the curvature map can be constructed as an image that represents the first principal direction at each point on an object. A curvature map is an image that records the direction of curvature of a point on an object at each pixel of the projection plane, as is shown in Fig. 4(c).

The object ID map is useful in rendering multiple objects, to distinguish the objects while accessing other reference maps. The object ID map is constructed by distinctly coloring every object.

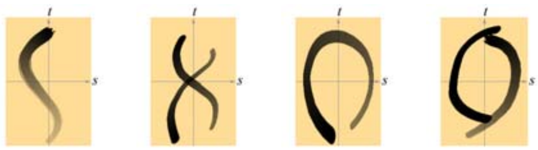
4.2 Procedural Texture

This paper uses a procedural textural approach, preserving the Chinese ink painting style over accurately rendering the surface. Accordingly, a 2D texture pattern that is mappable and renderable to is first created to preserve both the stroke path and the brush profile. Then, the texture pattern is mapped onto the image to create the desired surface appearance.

4.2.1 Controlling Texture Pattern

This section addresses the procedural control of texture shape by defining at least one brush stroke on the texture pattern. Let τ(G, B, s, t) be a unit pattern defined by the brush stroke set (G, B) at texture coordinate (s, t). Each brush stroke (g_i, b_i) of (G, B) is defined by two components - stroke geometry (path), g_i, and brush profile, b_i. In practice, g_i can be defined by a set of control points to specify the path of the stroke at a texture coordinate, and b_i can be defined by a set of parameters of the brush model. Figure 5 shows many examples of τ. Unlike traditional image-based texture, the shape of the procedural textures presented here can be locally controlled by slightly adjusting the profile of the brush or disturbing the control points of curve to generate a variable pattern τ', where

$$\tau'(G,B,s,t) = \tau(TurbG(G), TurbB(B), s, t)$$



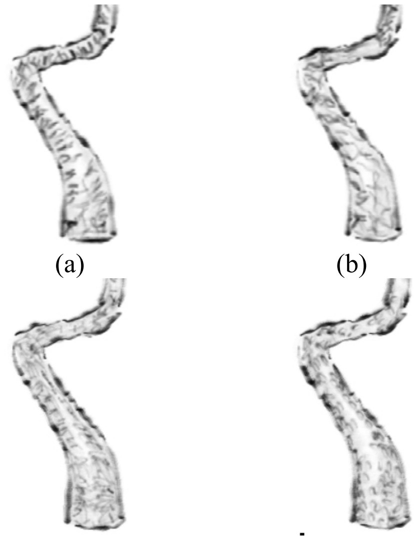
(a) (b) (c) (d)

Figure 5: (a) Single stroke (b) Knotted strokes (c) Scaled-like stroke (d) Coupled strokes

Controlling the turbulent functions make the entire texture patterns look slightly different from each other, creating a natural look. Figure 6 displays different textures generated by applying various texture patterns. The horizontal texture in Fig. 6 (a) is generated by single stroke patterns as shown in Fig. 5 (a), and the slant texture in Fig. 6 (b) is generated by knotted stroke patterns, as also shown in Fig. 5 (b). The textures in Fig. 6 (c) and (d) are both scale-like (or circular) and texture patterns, coupled stroke and scale-like stroke, which are shown in Fig. 5 (d) and (c).

4.2.2 Texture Mapping

The appropriate distribution of texture patterns is first analyzed to generate the desired texture on the rendering image. Let the probability of texture appearance of a pixel *p*(*x*, *y*) on the rendering image, be *λ*(*p*). Define



(a) (b) (c) (d)

Figure 6: Applying different styles of texture pattern.
(a) Horizontal texture with single stroke patterns.
(b) Slant texture with knotted strokes patterns.
(c) Scale-like texture with coupled strokes patterns.
(d) Scale-like texture with scaled stroke patterns.

$$\lambda(p) = \omega_\lambda \cdot \lambda_{view}(p) \cdot \lambda_{object}(p) \cdot \lambda_{relation}(p)$$

where

$$\lambda_{view}(p) = (1 - |N \cdot V|)^\delta + r_{view}$$

$$\lambda_{object}(p) = \begin{cases} 1 & if\ MAP_{Oid}(p) = Oid \\ 0 & if\ MAP_{Oid}(p) \neq Oid \end{cases}$$

$$\lambda_{relation}(p) = \begin{cases} 1 & if\ MAP_{mark}(p) \leq T_{mark} \\ 0 & if\ MAP_{mark}(p) > T_{mark} \end{cases}$$

The meaning of each term is as follows. ω_λ is the desired appearing weight. N is the normal vector accessed from $MAP_n(p)$, and V is the viewing direction. The scattered degree, δ , determines the density of the texture. A smaller δ represents a sparser the scattering of the texture patterns. Let $\delta = \omega_{view} \cdot D$, where ω_{view} is the desired scattered weight and D is the camera distance. Fewer texture patterns are observed when the camera is farther away from the object. r_{view} is a random factor that creates non-uniform textures. $\lambda_{view}(p)$ creates a denser texture nearer silhouettes. $\lambda_{object}(p)$ determines whether pixel p belongs to a certain object, so that different styles can be simultaneously assigned to different objects. The term, MAP_{mark}, refers to an additional map that marks up the texture appearing area. Setting the marking at threshold $T_{mark} = 1$ prevents the texture patterns from overlapping. An appearance threshold T_λ is chosen, and if $\lambda(p) > T_\lambda$, a texture pattern τ' centered at p, is mapped using the following mapping function.

A mapping matrix, M, is defined to map the geometric component, G, of texture pattern τ', to the screen space. The texture pattern centered at pixel p(x, y) in screen space, is mapped by $\hat{\tau}(G) = M \cdot \tau'(G)$, where

$$M = T(x, y) \cdot S_2(s_x, s_y) \cdot R(\theta + \phi) \cdot S_1(s_s, s_t)$$

The texture pattern is first scaled to the desired size. The first scale vector (S_s, S_t) is inversely proportional to the camera distance, D, and thus is decreased by zooming out. The vector is also inversely proportional to the depth value accessed by MAP_d(p), to show gradations between near and far fields. The texture pattern is then rotated to align the orientation angle θ with a deviation angle ϕ. θ is given by,

$\theta = \tan^{-1}(\frac{c_y}{c_x})$, where C($C_x$, C_y) is a 2D vector obtained by projecting the principal curvature direction T_1 accessed from MAP_c(p) to the view plane. ϕ is defined as the deviation of angle from the orientation angle. Adjusting ϕ allows identical styles of texture to be generated in different directions, creating for example, straight, horizontal and slant textures, as depicted in Fig. 7.

(a) (b) (c) (d)

Figure 7: The variety of deviation angle ϕ.

(a) Possible principal curvature direction.
(b) Horizontal texture with $\phi = 0$.
(c) Slant texture with $\phi = \pi / 4$.
(d) Straight texture with $\phi = \pi / 2$.

Figure 8: The texture appearance while zoom-in, comparing with Fig. 6 (d).

The second scale vector (S_x, S_y) is applied to resize the texture patterns to make them smaller nearer silhouettes. S_x is proportional to the length of n_x and S_y is proportional to the length of n_y. The 2D vector n(n_x, n_y) is computed by projecting the normal vector, N, which is accessed from MAP_n(p), onto the view plane. The texture pattern is then translated to the desired pixel location. Figure 8 compares the view-dependent texture mapping, affected by the second scale vector (S_x, S_y), to the texture mapping from Fig. 6 (d).

5. RESULTS

Many trees, synthesized by the new painting algorithm in the Chinese ink painting style, are illustrated. Figures 9 and 11 are the tree models. Figure 10 shows the results obtained by implementing Fig 9. Figure 12 renders the output from Fig 11. Figure 13 renders the tree from 3D model of Fig 13 (a), for various viewing positions. The presented algorithm can be applied to other subjects in the Chinese ink painting style. For example, the dotted texture patterns can be applied to mountain models to simulate the "raining dot" texture in a landscape scene, as shown in Fig. 14.

Figure 9: 3D tree model

Figure 10: Rendered output of Fig 9.

Figure 11: 3D tree model

Figure 12: Rendered output of Fig 11.

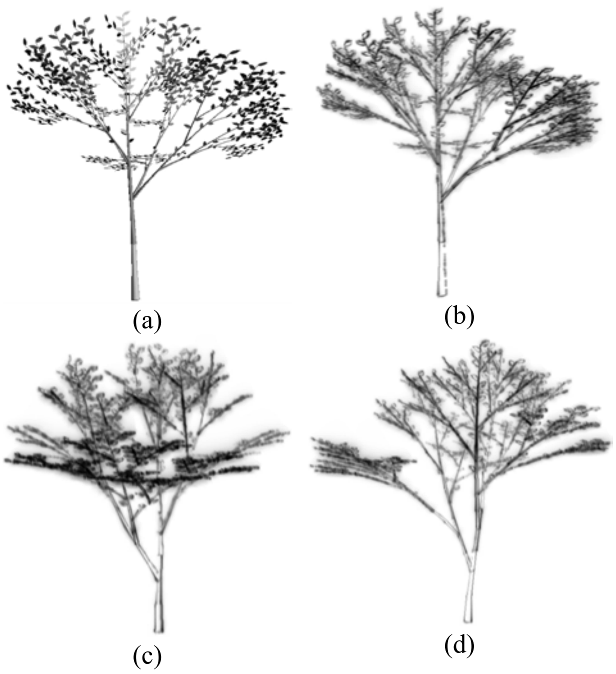


Figure 13: Rendering the tree in various viewing positions.

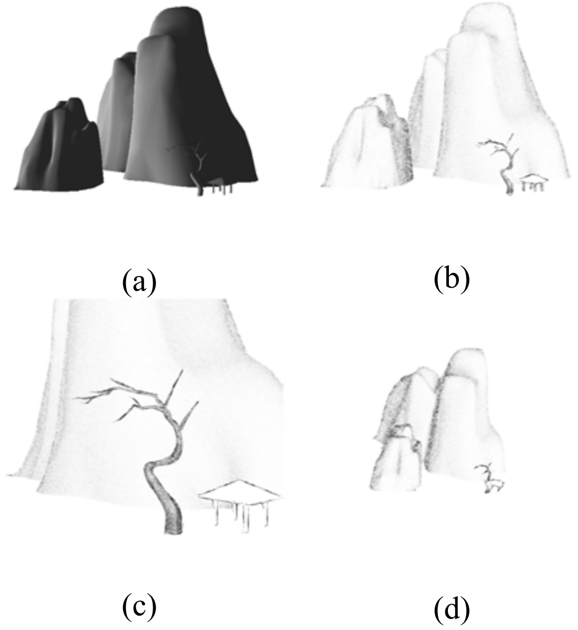


Figure 14: Rendering the landscape scene in different viewing positions.

6. CONCLUSIONS AND FUTURE WORK

This paper presented methods for automatically synthesizing three-dimensional trees in Chinese landscape painting. Outline rendering and texture generation uses the information of the silhouette, shade and orientation of three-dimensional model's surface to draw a particular tree. Four reference maps are constructed to analyze the information and then generate brush strokes of the bark texture.

These experiments have opened up many topics for future research.

1. Chinese ink painting includes many tree styles. They include boneless and detailed-one (Kung-Pi), etc. Developing all these styles is fairly important in the future.
2. How to extracting skeleton from 3D polygonal model? The skeleton of tree is very typical information for Chinese boneless ink painting.
3. The foliage of trees is diversiform. The tree skeleton and the foliage are processed separately automatically to reduce the complexity of the tree model.
4. Two consecutive frames in an animation sequence are likely to be similar. Frame-to-frame coherence could be exploited by reusing information.

ACKNOWLEDGEMENT

The authors would like to thank the National Science Council of the Republic of China for financially supporting this research under Contract No. NSC_90-2213-E-009-128.

7. REFERENCE

[Chow79] Chow Chian Chiu and Chow Leung Chen Ying. Chinese Painting: A Comprehensive Guide. Published in 1979 by Art Book Co., Ltd.

[Corrêa98] Wagner Toledo Corrêa, Robert J. Jensen, et al. "Texture Mapping for Cel Animation". In Michael Cohen, editor, *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, pages 435–446. ACM SIGGRAPH, Addison Wesley, July 1998. ISBN 0-89791-999-8.

[Elber99] Gershon Elber. "Interactive Line Art Rendering of Freeform Surfaces". *EUROGRAPHICS '99*, Vol. 18, No. 3.

[Girsh2000] Ahna Girshick, Victoria Interrante, et al. "Line Direction Matters: An Argument For The Use Of Principal Directions In 3D Line Drawings". *Proceedings of the first international symposium on NPAR*, 2000, pages 43–52.

[Hertz2000] Aaron Hertzmann and Denis Zorin. "Illustrating smooth surfaces". *Proceedings of the conference on Computer graphics*, 2000, pages 517–526.

[Hertz99] Aaron Hertzmann. "Introduction to 3D Non-Photorealistic Rendering: Silhouettes and Outlines". *SIGGRAPH '99*, Course 17, Chapter 7.

[Katanics] George Katanics and Tasso Lappas. "Deep Canvas: Integrating 3-D Painting and Painterly Rendering".

[Lee99] Jintae Lee. "Simulating Oriental Black-Ink Painting". *IEEE Computer Graphics and Applications*, May 1999.

[Lee2000] Lee Markosian, Barbara J. Meier, et al. "Art-Based Rendering with Continuous Levels of Detail". *Proceedings of the first international symposium on NPAR*, 2000, pages 59–66.

[Lee97] Lee Markosian, Michael A. Kowalski, et al. "Real-Time Nonphotorealistic Rendering". *Proceedings of the 24th annual conference on Computer graphics & interactive techniques*, 1997, pages 415–420.

[Liu84] Yung Liu. "Ten Thousand Mountains". Published in the United States in 1984 by Shui-Yun-Chai Studio, pages 56–73 East Hampton Boulevard, Bayside, New York 11364.

[Lu90] Lu Xi Jiong. Drawing Trees. Published in 1990 by Art Book Co., Ltd. ISBN 957-9045-08-9.

[Mei88] Gao Ling Mei. Chinese Painting by Chang Da-Chien. Published in 1988 by Art Book Co., Ltd.

[Michael97] Michael P. Salisbury, Michael T. Wong, et al. "Orientable textures for image-based pen-and-ink illustration". *Proceedings of the 24th annual conference on Computer graphics & interactive techniques*, 1997, pages 401–406.

[Michael99] Michael A. Kowalski, Lee Markosian, J. D. et al. "Art-Based Rendering of Fur, Grass, and Trees". *Proceedings of the SIGGRAPH 1999 annual conference on Computer graphics*, 1999, pages 433–438.

[North2000] J. D. Northrup and Lee Markosian. "Artistic Silhouettes: A Hybrid Approach". *Proceedings of the first international symposium on NPAR*, 2000, pages 31–37.

[Raskar99] Ramesh Raskar and Michael Cohen. "Image precision silhouette edges". *Proceedings of the 1999 symposium on Interactive 3D graphics*, 1999, pages 135–140.

[Saito90] Takafumi Saito and Tokiichiro Takahashi. "Comprehensible rendering of 3-D shapes." *SIGGRAPH '90 Proceedings*, Vol. 24, pages 197–206, August 1990.

[Turk92] Greg Turk. "Re-Tiling Polygonal Surfaces". *Computer Graphics*, 26, 2, July 1992, pages 55–64.

[Veryovka99] O.Veryovka and J. Buchanan. "Comprehensive Halftoning of 3D Scenes". *EUROGRAPHICS '99*, Vol. 18, No. 3.

[Way2001a] Der-Lor Way, Chih-Wei Hsu, Hsin-Yi Chiu, Zen-Chung Shih, "Computer-Generated Chinese Painting for Landscapes and Portraits", *Proceedings of WSCG 2001.* pp. 387-394.

[Way2001b] Der-Lor Way, Zen-Chung Shih, "The Synthesis of Rock Textures in Chinese Landscape Painting". *COMPUTER GRAPHICS Forum Volume 20, Number 3,* pp. C123-C131, 2001.

[Zhang99] Qing Zhang, Youetsu Sato, Jun-ya Takahashi, et al. "Simple Cellular Automatic-based Simulation of Ink Behavior and It's Application to Suibokuga-like 3D Rendering of Trees". *Visualization & Computer Animation*, Vol. 10, pages 23–27, 1999.