Automatic Image-Based Pencil Sketch Rendering

WANG Jin (王 进)[1], BAO Hujun (鲍虎军)[1], ZHOU Weihua (周伟华)[1], PENG Qunsheng (彭群生)[1] and XU Yingqing (徐迎庆)[2]

[1]*State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310027, P.R. China*

[2]*Microsoft China Research, Beijing 100080, P.R. China*

E-mail: jwang@cad.zju.edu.cn

Received June 1, 2001; revised September 3, 2001.

Abstract This paper presents an automatic image-based approach for converting grey-scale images to pencil sketches, in which strokes follow the image features. The algorithm first extracts a dense direction field automatically using Logical/Linear operators which embody the drawing mechanism. Next, a reconstruction approach based on a sampling-and-interpolation scheme is introduced to generate stroke paths from the direction field. Finally, pencil strokes are rendered along the specified paths with consideration of image tone and artificial illumination. As an important application, the technique is applied to render portraits from images with little user interaction. The experimental results demonstrate that the approach can automatically achieve compelling pencil sketches from reference images.

Keywords non-photorealistic rendering, pencil sketch, image-based rendering, pen-and-ink illustration

1 Introduction

In recent years, a novel rendering scheme, called non-photorealistic rendering (NPR), has emerged as a way to produce illustrations with artistic, painterly style based on 3D models or reference images. In comparison with photorealistic images, such drawings have advantages such as the powerful ability to abstract details, clarify shapes and focus attention, which provides alternative display methods for 3D scenes and reference images.

1.1 Related Work

In the past years, researchers proposed a variety of techniques to generate artistic, painterly drawing style illustrations, such as halftone[1−3], pen-and-ink illustrations[4−9], watercolor paintings[10,11], oil paintings[11−14], impressionist paintings[15] and pencil sketches[16−19]. NPR systems can be classified into two categories: geometry-based NPR and image-based NPR. The former takes 3D scene descriptions as input, and the latter takes reference images as input. Since only image tone conveys the 3D information of scenes to the viewers essentially, the technique needs to extract the feature lines based on the image tone to drive the strokes[5,7,15,20,21].

Our work is related to image-based NPR for pencil sketches, which is motivated by the use of directional strokes. The basic idea appeared in several pioneering works on NPR[5,7,15,21]. In comparison with traditional systems, our system is much effective, and can achieve excellent painting results. To our knowledge, most of the NPR approaches for pencil sketches (for example, M.C. Sousa and J. Buchanan[16−18], etc.) need a lot of human interaction and they just provide the user with some interactive tools to draw pictures. Artdabbler, a commercial software, is typically such a system. In our work, strokes which are automatically extracted from the image by using gradient-based techniques are distributed over the image and drawn with supplied textures from the image.

The project is supported by the National Natural Science Funds for Distinguished Young Scholars of China (No. 69925204) and Innovative Research Groups (No. 60021201).

1.2 Overview

It is found that the existing image-based NPR approaches involve a lot of user interaction due to the following reasons:

(1) The source image is represented in pixels without explicit feature line information, especially in the regions where textures vary smoothly;

(2) Except for the physical illumination, an artist usually utilizes artificial illumination to strengthen the object features or present different artistic styles;

(3) Many kinds of pencils are demanded for expressing different levels of details in drawing a picture, e.g., a portrait.

To solve these problems, this paper presents a novel rendering approach with little interaction to create a pencil sketch based on a reference image. First we use Logical/Linear operations to extract short vectors from the image to produce a dense direction field. A set of representative strokes exhibiting the image features can then be derived. Additional strokes can be generated by interpolation. Image tone is conformed to when drawing each directional stroke. The potentials of our approach are demonstrated by some experimental pencil sketches of portraits.

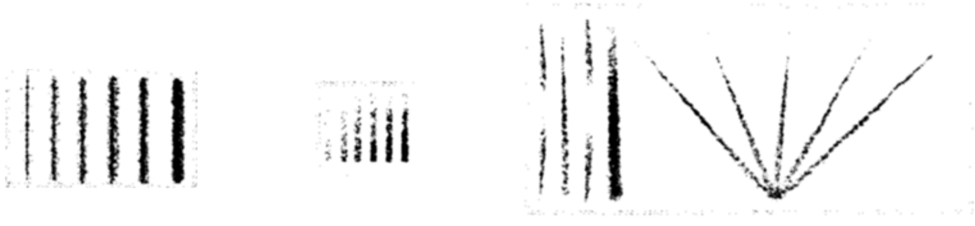
The rest of this paper is organized as follows. In Section 2, we present a brief description of the pencil and paper model. Section 3 introduces the rendering algorithm, including the extraction of feature line segments from the reference image, generation of directional strokes and strokes rendering with tone from the image. In Section 4, we show how to apply our approach to create portrait pencil sketches based on an image. Sections 5 and 6 present some experimental results and conclusions.

2 Pencil and Paper Model

In pencil sketch, the pencil drawing effects may be affected by different kinds of pencils and paper roughness. Sousa and Buchanan[16–18] proposed an observational model to simulate the pencil behaviors. In the model, they utilized the amount of graphite and clay to simulate the pencil hardness ranging from 9H (hardest) to 8B (softest), a polygon with pressure distribution around the pencil tip to simulate the contact surface, as well as the paper texture (a height field) to simulate paper roughness[10].

We make the following improvements on Sousa and Buchanan's pencil model to make it more suitable for our automatic image-based pencil sketch rendering:

(1) The tone of a stroke depends on the graphite powder distribution and density over the pencil tip polygon, which can be used to simulate pencil hardness and pressure distribution. In order to get a natural, smooth and continuous tone, the minimal unit of the graphite powder density may be less than 1 so that the strokes in the adjacent regions can blend continuously. In our model, we introduce conditional probability to score the graphite powder density. For instance, if the minimal unit of density is set as 0.01, the density 5.72 means that five points are unconditionally distributed in the tip polygon while an extra point is rendered in the polygon with 72% possibility. Similarly, the stroke size can be set in this way.



(a) (b) (c)

Fig.1. Strokes generated by the improved pencil model. (a) 6-layer strokes with a small pencil tip polygon of a 2B pencil. (b) 6-layer decay strokes with a large pencil tip polygon of a 2B pencil. (c) Five typical pencil strokes.

(2) To achieve the exquisite painting effects, we design an approach to generate mark-making effects by overlaying multiple pencil strokes. On one drawing paper, if two pencils A and B, whose densities are I_A and I_B respectively, are used to draw over the same location S in turn, the reflected density of graphite material deposited at S is: $I_S = I_A + I_B$.

(3) In practice, artists create the wiping effects by repeatedly wiping the drawn strokes with fingers, bread or soft spitball. In our implementation, we simulate these effects by implementing a bilinear interpolation scheme and a low band filter in a square neighborhood at the contact point (Fig.10(b) illustrates the wiping effect on the facial drawing).

3 Image-Based Pencil Sketch Generation

As mentioned in Subsection 1.2, the biggest difficulty associated with image-based NPR is how to specify stroke paths conforming to the feature curves of the image and how to paint different regions with respective kinds of pencils. In our implementation, the task can be accomplished by extracting feature lines from the original image.

3.1 Image Preprocessing

In order to express the depth field of scenes, artists usually make use of high contrast shading to emphasize the 3D structure of objects. Pencil sketches are manual work, however, it is difficult to produce all grayscales. In practice, ten levels of the tone are enough to express its alteration of object surfaces for painting purpose. Therefore, we need to preprocess the source image as follows:

(1) Using high band filter, the input image may be sharpened to emphasize image features. Sometimes, this preprocessing is important for extracting image feature lines using L/L operators described in the next section.

(2) We use a three-order B-Spline curve to strengthen the contrast of the input image. As shown in Fig.2, considering w_i ($i \in [0,3]$) as the weight, setting $w_0 = w_3 = 0$, we can use w_1 and w_2 to control the final effect. Next, the processed image is quantized so as to reduce the number of levels of gray scale. In the later rendering procedure, the graphite density is determined by the gray level. For instance, it is required that the number of levels of gray scale be reduced to $[0,9]$. The map is $f(s,t)$: target(s) ($s \in [0,9]$) $\leftrightarrow$ source(t) ($t \in [0,255]$), where source(t) and target(s) are the source image and the result one, respectively.

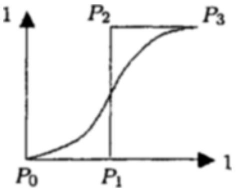


Fig.2. The curve used to strengthen contrast.

3.2 Image Feature Line Extraction

To automatically generate pencil sketches from images, we need to determine where to draw strokes to express the features exhibited in the input image. The process of edge detection is employed to accomplish our task. Many algorithms have been proposed based on some optimality criteria[22,23] and biological vision[24]. All of them based on the assumption of linearity to facilitate noise sensitivity analysis usually produce wrong response to uncharacteristic stimuli, and cannot produce the feature lines embodying the painting characters of images. Recently, Iverson and Zucker[25] have proposed Logical/Linear (L/L) operators to measure the low-order differential structure of image feature lines, which overcome some shortcomings of the traditional algorithms.

In L/L operators, Iverson and Zucker adopted the following terminology to categorize the features in images. An edge is the curve separating lighter and darker regions of an image, which is of zero order discontinuity along its normal cross-section. Lines are of the first-order discontinuity (creases) along their cross-section. Both edges and lines are called image feature lines, which are very important feature lines for drawings. Considering S and I as an source image and the set of an image curve, respectively, the definition of IMAGE CURVE is a *map* $m : S \to I$ such that

(Tangent) m is continuous on S, and

(Normal) a condition $N(condition_s)$ holds for all $s \in S$.

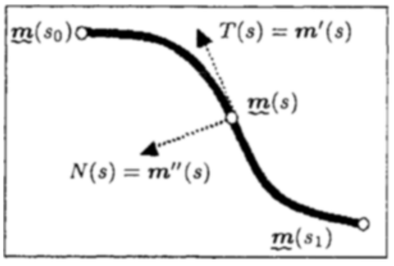


Fig.3. An image curve.

Here $N(condition)$ is the NORMAL CONDITION to determine the classification of the curve, and s represents one local cross-section which is the object processed (as in Fig.3). In a digital image, to a pixel $p(x, y)$, s is limited by $[x-\varepsilon, x+\varepsilon]$ and $[y-\varepsilon, y+\varepsilon]$.

Iverson and Zucker introduced some tangential conditions to guarantee continuity and normal conditions to select and categorize contrast profiles, which are efficiently defined by combining logic with the linear operator theory[25]. In comparison with traditional operators, L/L operators have the following advantages for our purpose:

(1) They can extract edges and lines independently. Therefore, the rendering style from the reference image for different types of feature lines can be controlled.

(2) In L/L operators, two parameters are introduced to control the distribution of the resulting feature lines[25]. The parameter ε is used to control the curve of separation, which ensures each resulting curve is localized within a region with the width less than 2ε pixels. The other parameter σ, concerning the variation of Gaussian distribution: $G_\sigma(x) = (1/\sqrt{2\pi\sigma})e^{-x^2/2\sigma^2}$, is to control the noise sensitivity of the resulting curves.

(3) L/L operators allow multiple image feature line segments with independent directions in the local neighborhood, so they can produce feature lines with T-junctions (bifurcation structures) popular in line drawings.

By using L/L operators, we can create a dense direction field controlled by two parameters ε and σ. Note that generating a long feature line is recognized as a difficult and time-consuming job. In our application, in order to preserve details and silhouettes, we just check the eight-connected neighborhood at each pixel to extract the feature line segments. Therefore, each segment in the direction field must be identical to one of the eight directions with length 1 or $\sqrt{2}$ as illustrated in Fig.4. Fig.5 shows the resultant image feature line segments with different parameters.

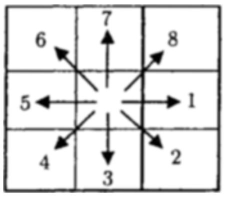


Fig.4. The eight directions detected in our approach.

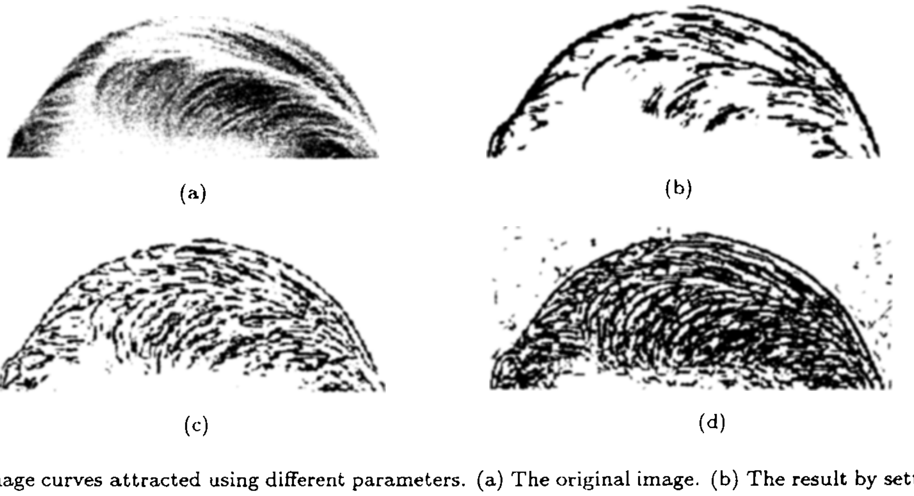


Fig.5. The image curves attracted using different parameters. (a) The original image. (b) The result by setting $\varepsilon = 17.5$ and $\sigma = 64$. (c) The result by setting $\varepsilon = 7$ and $\sigma = 32$. (d) The result by setting $\varepsilon = 5$ and $\sigma = 0.01$.

3.3 Stroke Generation

As illustrated in Fig.5, there are few segments in smooth regions with less intensity detail, while many segments yield in regions with high detail. Nevertheless, too many short segments gathered in a small region look noisy in vision. On the other hand, too few may not be capable of exhibiting three-dimensional structures. Apparently, the extracted feature line segments are still inadequate to be directly used to generate the desired strokes.

Many previous studies show that it is difficult to extract long feature lines from images. And this is also true in our case. Instead of tracing short segments one by one, we first generate some representative vectors of controllable length for each specified region to reveal the directions of feature lines in the region. An interpolation scheme is then used to generate a number of strokes conforming to the image tone of the region. In the following, we will discuss our approach to achieving the goal.

(1) Lowering the segment density and lengthening the segments in the regions with high detail so as to reveal the image feature lines and produce the continuous tone.

(2) Increasing the number of segments in smooth regions to exhibit the three-dimensional structure.

3.3.1 Representative Vectors

Without loss of generality, we assume that the input image has been partitioned into several regions with different features. Over each region, we set up a uniform two-layer grid structure. The resolution of the bottom layer is controlled by users. The top layer is generated by 3×3 subdivision of the bottom layer. In practice, the width of each cell in the bottom layer may be set as 15–36 pixels, which means the width of each cell in the top layer is 5–12 pixels.

To decrease the number of segments in the region, we generate a representative vector at each cell center of the top layer. As mentioned before, only eight directions are involved in the direction field, so we can encode them as codes 1000, 1001, 0001, 0101, 0100, 0110, 0010, 1010, which represent respectively the directions from 1 to 8 in Fig.2. Each segment can be represented by its start point (x, y) together with a code $q_1q_2q_3q_4$, where q_i $(i = 1, 2, 3, 4)$ is either 1 or 0. Assume that there are m segments in a top cell, denoted as $(x_i, y_i, q_{1i}, q_{2i}, q_{3i}, q_{4i})$ $(i = 1, 2, \ldots, m)$. The representative vector can be determined by (D_x, D_y): $D_x = \frac{l}{m}\sum_{i=1}^{m}(q_{1i} - q_{2i})$; $D_y = \frac{l}{m}\sum_{i=1}^{m}(q_{3i} - q_{4i})$, where l is the parameter regarding vector length controlled by a random process. If there are no line segments in a cell, the cell will be enlarged to include its adjacent cells, or will be kept blank.

3.3.2 Interpolated Strokes

Obviously, strokes along the representative vectors are still insufficient to express the tone changes in drawings. Note that the tone of pencil sketches is achieved by overlaying multiple strokes in a region, thus the density of vectors needs to be increased with randomness in respective regions. We introduce an interpolation scheme to achieve this goal.

The interpolation scheme adopted by our approach originates from [26]. Consider the representative vectors s_i $(i = 1, 2, \ldots, n < 10)$ starting from (x_i, y_i) with vector (D_{xi}, D_{yi}) in the bottom layer, where (x_{ci}, y_{ci}) is the center of the i-th cell in the bottom layer. An interpolated vector starting at $P(x, y)$ can be determined by $(D_x, D_y) = (\sum_{i=1}^{n}\frac{(D_{xi}, D_{yi})}{d^2(P, S_i)})/(\sum_{i=1}^{n}\frac{1}{d^2(P, S_i)})$, where $d(P, S_i)$ is the distance from P to the representative vector S_i.

We draw additional strokes along these interpolated vectors. In order to emphasize the feature lines, we use the Gaussian distribution to locate the interpolated vectors so that a number of strokes are around the representative vectors.

The number of interpolated strokes is specified by image tone in the cell as follows: $N \propto k_1g + k_2h$, where g and h are the average intensity and the entropy of the differential histogram in the cell, respectively. Parameters k_1 and k_2 are the weights controlling drawing styles in the cell. Greater value of parameter k_2 contributes more interpolated strokes in regions with high detail, and relatively fewer in smooth regions. Fig.6 shows the interpolated strokes from the direction field.

Fig.6. The reconstructed stroke paths from the direction field. (a) One right eyebrow. (b) The dense direction field. (c) The representative orientations. (d) The interpolated stroke paths.

3.4 Strokes Rendering

During drawing, we use the pencil and paper model described in Section 2 to simulate the drawing effects with different types of pencils. The system provides the user with several parameters, such as stroke size, point distribution, point intensity, decay factor, etc., to control the final rendering effects. Since it is a laborious task to configure these parameters one by one, we determine the point distribution and intensity adaptively according to the image intensity or tone along feature lines. In our implementation, the intensity of strokes is determined by considering both the intensity of the input image and the artificial illumination: $I = (1-\beta)I_{light}(\frac{1+\omega\cos\alpha}{1+\omega}) + \beta I_o$, where I_o is the original intensity, I_{light} is the intensity of the artificial lights, $\omega \in [0,1]$ is the shape parameter of the intensity curve, α is the angle between the stroke and the positive direction of y-axis, β is the blending factor. Fig.7 shows pencil sketches rendered with parameters I_{light} and ω.

Fig.7. Results rendered with intensity parameters. $I_{light} = 102$, $\omega = 1.5$.

After configuring all stroke parameters in a region, the region can be rendered using the overlay technique described in the pencil model.

4 Pencil Sketches for Portraits

In pencil sketches, portraits are the most popular drawings. Portraitists often use different kinds of drawing skills to express the personality characteristics in different regions. For instance, large organs are sketched with fatter strokes, while small organs, such as eyes, eyebrows, mouth and nose, are drawn with relatively thinner strokes. The difficulty in automatically rendering portraits from images is to specify stroke parameters and density applied for drawing various important organs. In the following, we present a tool to semi-automatically draw portraits.

To meet our requirements, we first interactively position the following nine regions in the input image: hair, left eyebrow, right eyebrow, left eye, right eye, visor, nose, mouth and the rest. We do not independently specify ear regions because ears are relatively unimportant for portraits. To improve the interaction efficiency, we establish a local coordinate system at the middle point between the nose tip

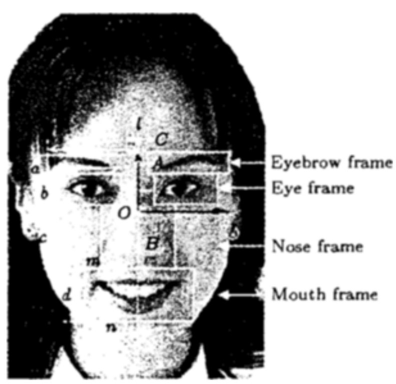


Fig.8. Face template for interactive portrait rendering.

and the eyebrow center, which is associated with a facial region template. The template is defined by six box frames illustrated in Fig.8. With the hair region and the facial template, we can roughly locate the nine major regions of portrait images conveniently.

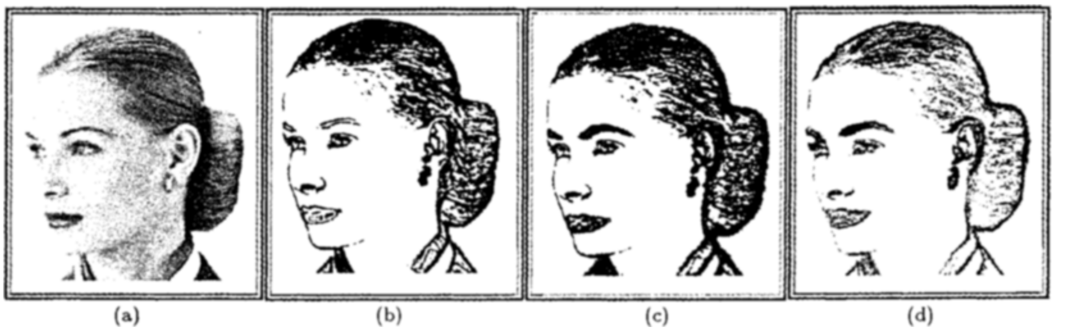
According to the physiology knowledge, the default frame ratio parameters in the template may be set as follows: $a = 2$; $b = 7$; $c = 50$; $d = 14$; $j = 22$; $k = 18$; $l = 4$; $m = 3$; $n = 16$. In our system, the user may transform (translate, rotate or scale) the template or change its frame parameters to fit the sites of the corresponding facial organs.

Once the template has been positioned correctly, we adopt the snake algorithm[27,28] to find out the border of each region. If necessary, the start point for tracing borders may be specified by the user. By assigning the stroke parameters to each region, the system will automatically implement our algorithm to generate a pencil sketch based on the input image.

5 Implementation and Results

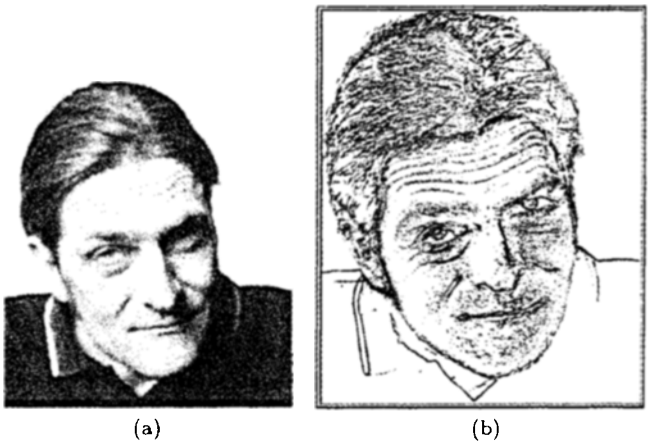
We have implemented our algorithm on a personal computer with Pentium III 600 CPU using Visual C++.

Fig.9 illustrates the full procedure of our approach. Fig.9(a) is the source image. We use Logical/Linear operations to produce a dense direction field showed in Fig.9(b). Based on the dense direction field we calculate a set of representative vectors. These vectors are interpolated. Fig.9(c) shows the interpolated strokes. Finally the interpolated strokes are used to drive the respective models



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

Fig.9. (a) Original image. (b) The dense direction field. (c) The interpolated stroke paths. (d) Effect picture (5B pencil).



(a) (b)

Fig.10. (a) Original image of a man. (b) The pencil sketch of the man.

for rendering and the resultant portrait is shown in Fig.9(d). In the processing procedure we set different parameters for different regions. For extracting the dense direction field from the source image, we set $\varepsilon = 15$, $\sigma = 8$ to the hair, the eyebrows, the mouth, the ear and the clothes which are to be rendered with a 5B pencil and set $\varepsilon = 4$, $\sigma = 0.015$ to the eyes, the nose and the visor which are to be rendered with a HB pencil. During the rendering procedure, we set $I_{light} = 122$, $w = 1.8$ to produce the shading effect.

Fig.10 provides another example. Effect images are zoomed out by sixteen times through extending the length of the vector lines to the fourfold size.

Stroke locations and sketch shading are two key problems in generating a pencil sketch. Our experimental results demonstrate that they are solved nicely. The proposed approach only contains few interactive operators.

6 Conclusions

An algorithm for automatically generating pencil sketches from images has been presented. To simulate the traditional sketching skills, strokes should be drawn to reveal the feature lines exhibited in the reference image. This goal is achieved by performing Logical/Linear operations, image feature line detection and automatic stroke rendering. Image tone is the major issue considered in the three processes. As an important application, the algorithm has been applied to portrait rendering and produces satisfying resultant images.

Future work may extend the algorithm to produce other artistic styles. It would be interesting to apply this approach to videos or 3D objects to produce pencil-style videos or animations[15].

References

[1] Veryorka O, Buchanan J. Comprehensive halftoning of 3D scenes. *Computer Graphics Forum, EUROGRAPHICS'99*, 1999, 18(3): 13–22.

[2] Deussen O, Hiller S. Floating points: A method for computing stipple drawings. *Computer Graphics Forum, EUROGRAPHICS'2000*, 2000, 19(3): 455–466.

[3] Buchanan J W. Special effects with half-toning. *Computer Graphics Forum, EUROGRAPHICS'96*, 1996, 15(3): 97–107.

[4] Winkenbach G. Rendering parametric surfaces in pen-and-ink. *Computer Graphics, ACM Siggraph'96*, 1996, pp.469–476.

[5] Salisbury M. Scale dependent reproduction of pen-and-ink illustrations. *Computer Graphics, ACM Siggraph'96*, 1996, pp.461–468.

[6] Zeleznik R C. SKETCH: An interface for sketching 3D scenes. *Computer Graphics, ACM Siggraph'96*, 1996, pp.163–170.

[7] Ostromoukhov V. Digital facial engraving. *Computer Graphics, ACM Siggraph'99*, 1999, pp.417–424.

[8] Salisbury M P, Anderson S E, Barzel R, Salesin D H. Interactive pen-and-ink illustration. *Computer Graphics, ACM Siggraph'94*, 1994, pp.101–108.

[9] Winkenbach G, Salesin D H. Computer-generated pen-and-ink illustration. *Computer Graphics, ACM Siggraph'94*, 1994, pp.91–100.

[10] Curtis C J, Anderson S E, Seims J E, Fleischer K W, Salesin D H. Computer-generated watercolor. *Computer Graphics, ACM Siggraph'97*, 1997, pp.421–430.

[11] Aaron Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. *Computer Graphics, ACM Siggraph'98*, 1998, pp.453–460.

[12] Meier B J. Walt disney feature animation: Painterly rendering for animation. *Computer Graphics, ACM Siggraph'96*, 1996, pp.477–484.

[13] Gregg Kreutz. Problem Solving for Oil Painters. Watson-Guptill Publications, 1986.

[14] Strothotte T, Preim B, Raab A, Schumann J, Forsey D R. How to render frames and influence people. *Computer Graphics Forum*, 1994, 13(3): 455–466.

[15] Litwinowicz P. Processing images and video for impressionist effect. *Computer Graphics, ACM Siggraph'97*, 1997, pp.407–414.

[16] Sousa M C. Observational model of blenders and erasers in computer-generated pencil rendering. In *Proceedings of Graphics Interface'99*, 1999, pp.157–166.

[17] Sousa M C, Buchanan J W. Observational Models of Graphite Pencil and Drawing Paper for Non-Photorealistic Rendering. In URL: http://www.cs.ualberta.ca/~mario/TR/pencil.ps

[18] Sousa M C, Buchanan J W. Computer-generated graphite pencil rendering of 3D polygonal models. *Computer Graphics Forum, EUROGRAPHICS'99*, 1999, 18(3): 195–208.
[19] Melissa Hao. Computer-Generated Figure Drawing. In URL: http://www.mit.edu/~mhao/drawing
[20] Shiraishi M, Yamaguchi Y. An algorithm for automatic painterly rendering based on local source image approximation. In *Proceedings of NPR'2000*, 2000, pp.53–58.
[21] Salibury M P, Wong M T, Hughes J F, Salesin D H. Orientable textures for image-based pen-and-ink illustration. *Computer Graphics, ACM Siggraph'97*, 1997, pp.401–406.
[22] Canny J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6): 679–698.
[23] Heuckel M H. An operator which locates edges in digital pictures. *Journal of Association for Computing Machinery*, 1971, 18(3): 113–125.
[24] Jones J P, Palmer L A. An evaluation of the two-dimensional gabor filter model simple receptive fields in cat striate cortex. *Journal of Neurophysics*, 1987, 58(b): 1233–1258.
[25] Iverson L A, Zucker S W. Logical/Linear Operators for Image Curves. In URL: ftp://ftp.cim.mcgill.ca/pub/people/leei/loglin.tar.gz
[26] Nielson G, Foley T, Hamann B, Lane D. Visualizing and modeling scattered multivariate data. *IEEE Computer Graphics and Applications*, IEEE Compute. Soc. Press, Los Alamitos CA, May, 1991, 11(3): 47–55.
[27] Kass M, Within A, Terzopoulos D. Snake: Active contour model. *International Journal of Computer Vision*, 1988, pp.321–331.
[28] Wong Y Y, Yuen P C, Tong C S. Segmented snake for contour detection. *Pattern Recognition*, 1998, 31(11): 111–119.
[29] Ostromoukhov V, Hersch R D. Artistic screening. *Computer Graphics, ACM Siggraph'95*, 1995, pp.219–227.
[30] Dooley D, Cohen M. Automatic illustration of 3D geometric models: Lines. In *Proceedings of 1990 Symposium on Interactive 3D Graphics*, 1990, pp.77–82.
[31] Markosian L, Meier B J, Kowąlski M A, Holden L S, Northrup J D, Hughers J F. Art-based rendering with continuous levels of detail. In *Proceedings of NPR'2000*, 2000, pp.59–66.
[32] Elber G. Line art rendering via a coverage of isoparametric curves. *IEEE Transactions on Visualization and Computer Graphics*, 1998, 1(3): 231–239.
[33] Whitted T. Anti-aliased line drawing using brush extrusion. *Computer Graphics, ACM Siggraph'83*, 1983, pp.151–156.
[34] Lansdown J, Schofield S. Expressive rendering: A review of nonphotorealistic techniques. *IEEE Computer Graphics and Applications*, 1995, 15(3): 29–37.
[35] Berkel P V. SIAS, strokes interpreted animated sequences. *Computer Graphics Forum*, 1989, 8(8): 35–45.
[36] Strassmann S. Hairy brushes. *Computer Graphics, ACM Siggraph'86*, 1986, pp.225–232.
[37] Hsu S C, Lee I H H. Drawing and animation using skeletal strokes. *Computer Graphics, ACM Siggraph'94*, 1994, pp.109–118.
[38] Nothrup J D, Markosian L. Artistic silhouettes: A hybrid approach. In *Proceedings of NPR*, 2000, pp.31–37.
[39] Girshick A, Interrants V, Haker S, Lemoine T. Line direction matters: An argument for the use of principle direction in 3D line drawing. In *Proceedings of NPR*, 2000, pp.43–52.

WANG Jin received his M.S. degree in computer application from Zhejiang University in 1999. He is now a Ph.D. candidate at the State Key Lab of CAD&CG at Zhejiang University. His research interests include computer vision, NPR and image processing.

BAO Hujun received his Ph.D. degree from Zhejiang University in 1993. He is now a professor at the university. His main research interests are realtime rendering technique, digital geometry processing and field-based surface modeling.

ZHOU Weihua received his B.S. degree in engineering and communications from Zhejiang University in 1999. His research interests include computer graphics.

PENG Qunsheng received his Ph.D. degree in computer science from East Anglia University in 1983. He is now a professor and the Director of the State Key Lab of CAD&CG at Zhejiang University. His main research interests include realistic image synthesis, 3D modeling and virual reality.

XU Yingqing is a researcher of Microsoft China Research. His main research interests include computer graphics, digital geometry processing and virtual reality.