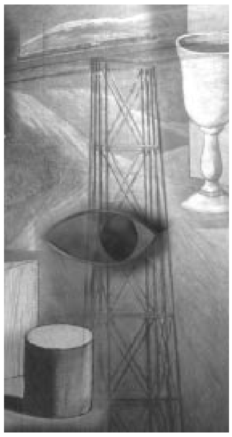
THE JOURNAL OF VISUALIZATION AND COMPUTER ANIMATION
J. Visual. Comput. Animat. **10**, 27–37 (1999)

Simple Cellular Automaton-based Simulation of Ink Behaviour and Its Application to Suibokuga-like 3D Rendering of Trees



*By Qing Zhang,[1] Youetsu Sato,[1] Jun-ya Takahashi,[1] Kazunobu Muraoka[1] and Norishige Chiba[1]**

**Suibokuga *is a style of monochrome painting characterized by the use of Chinese black ink (*sumi*), a complex interaction between brush, ink and paper, and such visual features as* Noutan *(shade),* Kasure *(scratchiness), and* Nijimi *(blur). In this paper we present a simple behavioural model of water and ink particles based on a 2D* cellular automaton *computational model, and its application to a* Suibokuga*-like rendering of 3D trees.* **

Received 17 March 1998; Revised 6 July 1998, 22 October 1998

KEY WORDS: non-photorealistic rendering; *Suibokuga*-like rendering; simulation of ink diffusion; cellular automaton

Introduction

Suibokuga is a style of monochrome painting characterized by the use of black ink (*sumi*), a solid form of charcoal or soot-based Chinese ink, which is ground on an inkstone (*suzuri*), diluted with water and applied with a brush to paper or silk. *Suibokuga* is also called *sumie* or ink pictures.

Noutan, *Kasure* and *Nijimi* are three of the remarkable visual features characteristic of *Suibokuga. Noutan* is an infinite range of monochromatic shade produced in a single brush stroke (Figure 1(a)). *Kasure* is the scratchy break-up of a brush's trajectory caused by an insufficient supply of ink from the brush (Figure 1(b)). *Nijimi* is a feathery pattern blurred by ink diffusion. Figure 1(c) and 1(d) show two typical patterns caused by thick and thin ink.

In this paper we first present a simple behavioural model of ink based on a 2D *cellular automaton* computation, along with paper and painting brush models for simulating the above phenomena. Next we apply the models to a *Suibokuga*-like rendering of 3D trees and discuss the advantages of our approach.

Related Work

A great deal of effort has been dedicated to the successful development of photorealistic computer graphics. In recent years, however, several researchers have presented techniques for generating aesthetically pleasing images that are both comprehensible and non-photorealistic, including paintings, sketches, technical illustrations and comics.[1–22] Small[23] presented a method that simulated the actions of pigment and water when applied to paper fibres using parallel cellular automata. Strassmann[2] presented a sophisticated method for generating a painting stroke that showed the visual effects of *Noutan* and *Kasure*, and succeeded in generating impressive *Suibokuga*-like paintings. Since then, studies on the computer simulation of brush strokes, especially on the strokes' natural outlines, as for calligraphy (*shodo*), have been vigorously, pursued by Japanese and Chinese researchers.[5,6,8,9,15]

Recently, Curtis *et al.*[22] proposed a model capable of simulating the behaviour of water colours and presented excellent simulation results for five characteristic phenomena, including *Kasure* and *Nijimi*. As is the case for our

*Correspondence to: N. Chiba, Department of Computer Science, Faculty of Engineering, Iwate University, Morioka 020, Japan. E-mail: nchiba@cis.iwate-u.ac.jp.
[1]Department of Computer Science, Faculty of Engineering, Iwate University, Morioka 020, Japan.

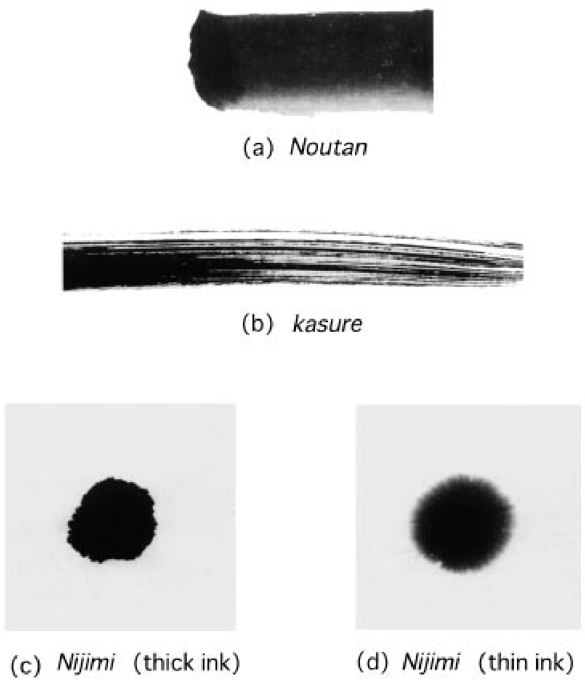


Figure 1. Pictures of real attractive visual effects

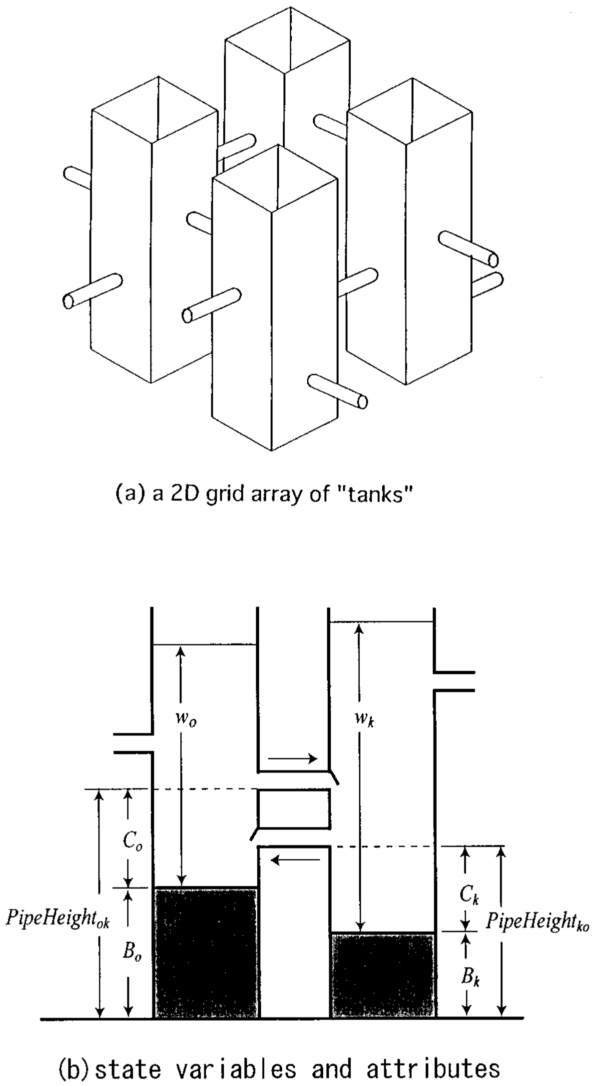


Figure 2. A 2D cellular automaton

proposed method, their method cannot generate well the feathery pattern peculiar to *Nijimi*. However, unlike our method, their method does not address the interactions between brush and paper or the brush strokes used in this type of painting.

Simple Models of Ink, Paper and Brushes

We represent the structure of paper and brushes necessary for our simulation by a common structure of a 2D grid array of 'tanks' linked with their neighbours by 'pipes' (see Figure 2(a)). Thus we will explain our computational models using the terminology of cellular automata.

Ink

We will present here the basic transfer and diffusion models of ink using a 2D cellular automaton. Ink distribution on paper and brushes is represented by the configuration of 2D cellular automata. Our 2D cellular automaton is as follows (see Figure 2(b)).

State Variables (W_{ij},I_{ij}) of a Cell P_{ij}

W_{ij} is the quantity of water staying in P_{ij}, i.e. the number of virtual 'water-particles', and I_{ij} is the number of 'ink particles' dissolved in water in P_{ij}.

Neighbourhood System

We use a *Neumann*-type neighbourhood system, i.e. P_{ij} is connected with its four neighbours $P_{i,j+1}$, $P_{i,j-1}$, $P_{i+1,j}$, and $P_{i-1,j}$. We denote by *Neighbour* the set of indices $(i,j+1)$,

$(i,j-1)$, $(i+1,j)$, and $(i-1,j)$, and denote (i,j) by o for convenience of description.

State Transition

The state transition is determined by a local transition function f as

$$S_o(t+\Delta t) := f(S_o(t), S_k(t) \mid k \in Neighbour)$$

where $S_o(t)$ is the state of cell P_o at time t, i.e.

$$S_o(t) = (W_o(t), I_o(t))$$

The local transition function f is implemented by iterating the following procedure n times in Δt.

Local Transition Function *f*

Step 1 (Transfer/Diffusion of Water Particles). (In the following descriptions we will omit the indices on time, as far as possible without confusion.) For every o let

$$W_o := W_o + \Sigma_{k \in Neighbour}(\Delta W_{ko} - \Delta W_{ok})$$
$$\text{if } W_o < 0, \quad \text{then } W_o := 0$$

where ΔW_{ko} *and* ΔW_{ok} are defined as

$$\Delta W_{ko} := \max\{0, 0{\cdot}25\ \alpha\ \min\{(B_k+W_k)-(B_o+W_o), (B_k+W_k)-PipeHeight_{ko}\}\}$$

$$\Delta W_{ok} := \max\{0, 0{\cdot}25\ \alpha\ \min\{(B_o+W_o)-(B_k+W_k), (B_o+W_o)-PipeHeight_{ok}\}\}$$

Here C_k and C_o are the capacities of water captured to cells P_o and P_k, respectively, B_o and B_k are the heights of the bottoms implying the capacities of cells P_o and P_k for water particles, respectively, $PipeHeight_{ko}$ is the pipe height from k to o, $(B_k+W_k)-PipeHeight_{ko}$ represents the water particles transferable from k to o, $(B_k+W_k)-(B_o+W_o)$ is the water level difference between k and o,

$$PipeHeight_{ko} = \max\{B_o, B_k + C_k\}$$
$$PipeHeight_{ok} = \max\{B_k, B_o + C_o\}$$

and α is the transfer/diffusion coefficient for water particles.

Step 2 (Transfer of Ink Particles Accompanying Water Particles). The number of ink particles is dependent on the concentration of ink in the transferred water. For every o let

$$I_o := I_o + \Sigma_{k \in Neighbour}(\Delta I_{tko} - \Delta I_{tok})$$

where

$$\Delta I_{tko} = \Delta W_{ko}(I_k(t)/W_k(t))$$

$$\Delta I_{tok} = \Delta W_{ok}(I_o(t)/W_o(t))$$

Step 3 (Transfer of Ink Particles in Order to Balance the Concentrations). For every o let

$$I_o = I_o + \Sigma_{k \in Neighbour} \Delta I_{dko}$$

where

$$\Delta I_{dko} = \beta[I_k - W_k(I_o+I_k)/(W_o+W_k)]$$

$$= \beta(I_k W_o - I_o W_o)/(W_o + W_k)$$

and β is the diffusion coefficient for ink.

Step 4 (Evaporation of Water Particles). For every o let $W_o := W_o - \Delta W$, where ΔW is a unit quantity of water for evaporation. If $W_o < 0$, then let $W_o := 0$, and fix the ink particles of I_o to the cell P_o as dried ink.

Paper

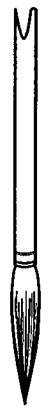
The paper used for *Suibokuga* is composed of a mesh of fibres that produces a capillary attraction and thereby affects the pattern of *Nijimi*. To account for these fibres, we model the paper as follows.

Modelling Paper

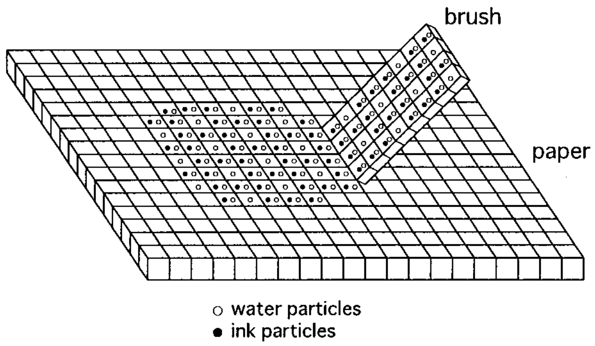
Step 0. Prepare a 2D cellular automaton representing paper. For every o, let $(B_o, C_o) :=$ a pair of appropriate constants proportional to the 'thickness' of the paper.

Step 1. To generate the paper texture formed by fibres, iterate the following steps appropriate times m according to the 'coarseness' of the paper, i.e. the coarser the paper, the fewer iterations.

Step 2-1. Generate a 2D line segment on the paper having constant length and random position and direction. (We consider this line segment a 'virtual' cluster of fibres, and assume that this cluster is visualized well by the



(a) a painting brush



(b) 2D cellular automata

Figure 3. A model of a brush

capillary attraction for ink, especially thin ink (see Figure 1(d)).)

Step 2-2. For each cell P_o lying under the line segment, let

$$(B_o, C_o) := (B_o - \Delta B, C_o + \Delta C)$$

where ΔB and ΔC correspond respectively to the capacity and the minimum capacity for storing water particles in a single cluster.

Brushes

Most *Suibokuga* painting brushes are composed of bristles shaped into a cylindrical cone (Figure 3(a)). Ink is captured in the narrow spaces between bristles by capillary attraction. To easily simulate ink behaviour in a brush, we also use a simple 2D cellular automaton as the model of a brush (Figure 3(b)), and simulate such phenomena as variable width of strokes, change in velocity of a brush in a stroke, flow of ink towards the tip of a brush, and unequal distribution of ink in a brush.

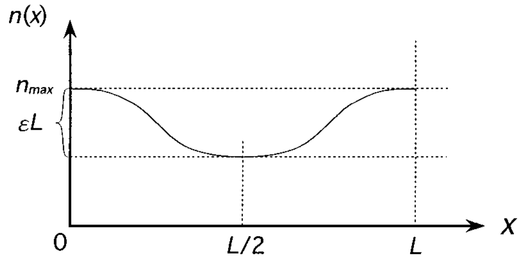


Figure 4. An assumption on drawing speed

Since brush pressure P widens the width B_w of a brush, we define the width B_w as

$$B_w = \gamma(1 - \exp P)$$

where γ is a coefficient. In the case of drawing a stroke of a tree in Section 4, in order to emphasize the difference in thickness of internodes, we use the brush pressure defined by

$$P = D^{1.5}$$

where D is the projected diameter of an internode. We assume that, during the drawing of a stroke, the number $n(x)$ of iterative computations for simulating the transfer and diffusion of ink required for passing over a unit cell at position x is given by (see Figure 4)

$$n(x) = n_{\max} - \varepsilon L\{1 - [(x - L/2)/(L/2)]^2\}^2 \quad 0 \leqslant x \leqslant L$$

where ε and $n_{\max}$ are appropriate constants, and should be chosen so that $n(x)$ becomes positive. Large $n(x)$ implies lower drawing speed. Although brush pressure P may be affected by the velocity V of a brush, we assumed constant pressure in a single stroke.

The paper contact side of a 2D cellular automaton representing a brush moves in the stroke direction, while maintaining horizontal (or vertical) posture against the grid co-ordinates of a 2D cellular automaton representing the paper. The side of the brush automaton is maintained parallel to the vertical axis for strokes at angles of up to $\pi/4$ from the horizontal axis of the paper cellular automaton, and parallel to the horizontal axis for strokes of greater angle. An algorithm for moving a brush vertically

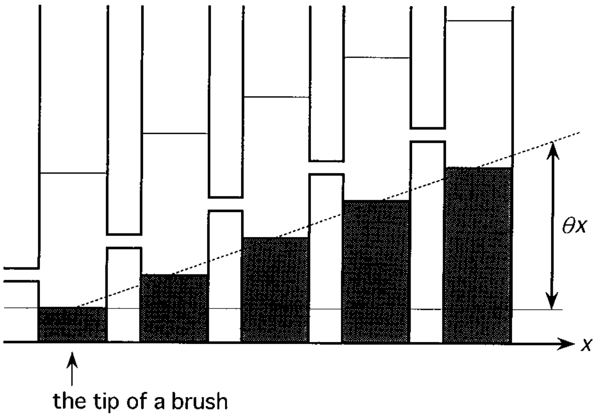


Figure 5. Modelling a brush

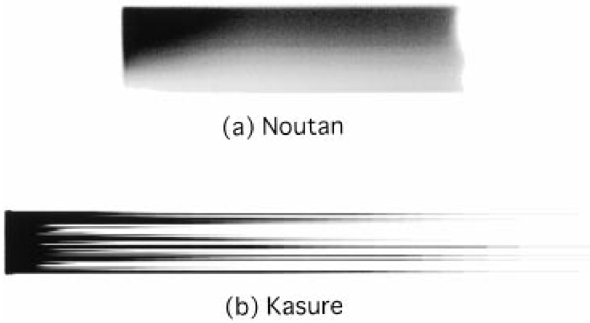


Figure 6. Simulated Kasure and Noutan on paper having uniform and isotropic properties

against the stroke direction should be developed. We model a brush as follows.

Modelling a Brush (See Figure 5)

Step 0. Prepare a 2D cellular automaton representing a brush.
Step 1. For every o, let (B_o, C_o):= a pair of appropriate constants proportional to the 'thickness' of the brush.
Step 2. To simulate the ink flow towards the tip of the brush, for every o, let

$$B_o = B_o + \theta x, \quad x \geqslant 0$$

where x is the distance from cell P_o to the tip and θ is a constant proportional to the gradient of the brush.
Step 3. To simulate the unequal distribution of ink in the brush, for every o let

$$B_o = B_o + F_o$$

where F_o is the corresponding element of a 2D $1/f$ noise F of the same resolution as the cellular automaton P.
Step 4. To simulate the effect of bristles, iterate the following steps the appropriate number of times. (This step is similar to Step 2 (effect of fibres) in the algorithm for paper modelling.)
Step 4-1. On the brush, generate a 2D line segment having constant length, random position and a direction parallel to the 'bristles'.
Step 4-2. For each cell P_o lying under the line segment, let

$$(B_o, C_o) := (B_o - \Delta B, C_o + \Delta C)$$

where ΔB and ΔC are appropriate constants.

Simulated Strokes

Figure 6 shows simulated *Noutan* and *Kasure* on paper having uniform and isotropic properties, i.e. modelled with Step 2 of the paper-modelling algorithm omitted.

Figure 6(a) presents a case with a gradual decrease in ink thickness between the upper and lower portions of the brush. This corresponds to the technique of *Suibokuga*, where the brush is angled slightly so that its side contacts the paper. The brush speed in Figure 6 is constant. The stroke in Figure 6(b) is generated by a brush containing ink whose initial quantity varies according to a semi-cylinder type function, i.e., the centre part of the brush has much more ink than its edges. The stroke itself simulates *Chokuhitu*, in which a brush is used vertically on paper.

Figure 7 shows examples of simulated *Nijimi* on paper modelled by taking into account the effect of fibres. Figures 7(a) and (b) are simulated on *fine* paper and *coarse* paper respectively. The left images are examples of thick ink, and the right ones of thin ink containing much more water.

Figure 8 shows examples of 'multiple strokes' applied over the same area. In Figure 8(a) the preceding stroke is dried completely before the succeeding stroke is drawn, whilst in Figure 8(b), the preceding stroke is not dried. In this example for *Nijimi*, as in those above for *Noutan* and *Kasure*, the visual effect is reasonably simulated.

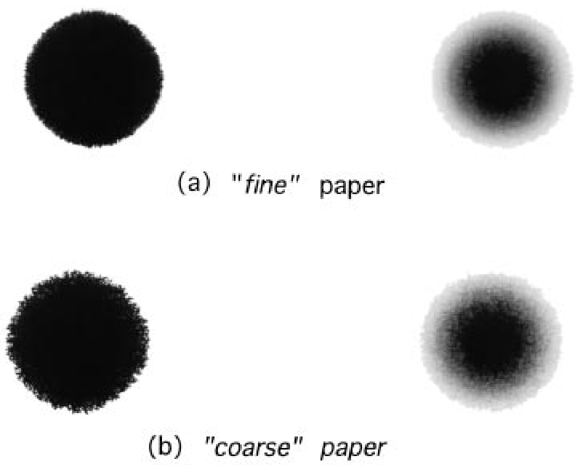


Figure 7. Simulated Nijimi *on paper modelled by taking into account effects of fibres*

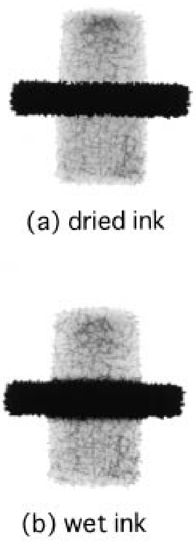


Figure 8. Simulated 'multiple strokes' on same place

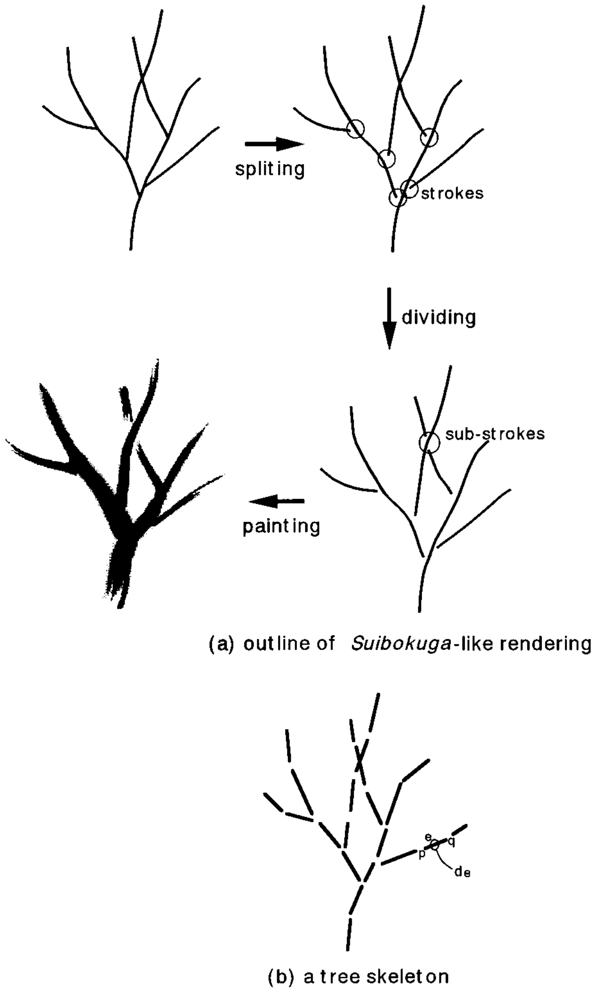


Figure 9. Suibokuga-*like rendering of trees*

Application to 3D *Suibokuga*-like Rendering of Trees

The outline of the rendering algorithm is as follows.

Suibokuga-like Rendering of a Tree (See Figure 9(a))

1. Split the tree into strokes, i.e. sequences of line segments each of which represents an internode of the tree.
2. Eliminate the hidden parts from each stroke under a certain criterion and define the substrokes for the remainder of the tree.
3. Draw the strokes individually by applying the models for ink transfer, diffusion, paper and brush.

The details of the algorithm are as follows.

Splitting Tree Skeletons into Strokes

Every painter has a unique way of painting. However, the proposed method uses the following strategy.

1. Move the brush from the root to a branch.
2. Draw a vital branch (internode) using one stroke. Vitality appears on a branch of strong apical dominance. Give priority to thick branches. If there are

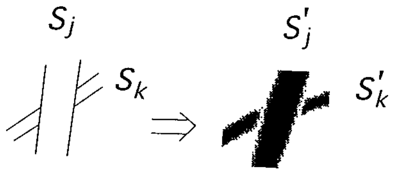


Figure 10. Hidden stroke elimination

two offspring branches, give priority to the one closer to the parent branch.
3. Give priority to offspring branches closer to the point of view.
4. If the angle between the offspring branch and the parent branch exceeds a limit, terminate the stroke.
5. Draw a background branch using a separate stroke if the branch is not hidden by a foreground branch.

In this paper we use the geometrical models of trees obtained by implementing the growth model proposed by Chiba *et al*.[24] Our growth model outputs a binary tree, i.e. a tree modelled on the simple dichotomy of its skeleton and thickness (diameter) (see Figure 9(b)). A skeleton is a set of internodes of a tree. An internode of a tree is represented by a line segment $e=(p,q)$ and a diameter d_e, where p and q are position vectors of the terminal points.

Splitting Algorithm

Step 0. Redefine the thickness of internodes by their projected thickness to the predefined screen.
Step 1. While there exists an internode, iterate the following Steps 1-1, 1-2 and 1-3.
Step 1-1. Find the internode E_S nearest to the root.
Step 1-2. Determine the stroke S, i.e. a sequence of internodes starting from the internode E_S, by traversing the remaining internodes, as will be described later.
Step 1-3. Delete the internodes contained in the stroke S from the tree.
Step 2. Let the set $\{S\}$ be the strokes found in Step 1, and $\{S'\}$ be their projected two-dimensional strokes.

Details of Step 1-2

Step 1. $E:=E_S$; $S:=E_S$.
Step 2. Iterate the following steps.
Step 2-1. If the current internode E has no descendant, then return.
Step 2-2. If E has only one descendant E_d, then $E:=E_d$ and $S:=S \circledcirc E_d$, where the operation $\circledcirc$ denotes concatenation.
Step 2-3. If E has two descendants, choose one of the descendant internodes E_d which satisfies one of the following conditions: $S:=S \circledcirc E_d$. (These conditions should be checked according to the order.)

Condition 1. E_d is thicker than the other.
Condition 2. The direction of E_d is nearer to that of E than the other.
Condition 3. E_d is closer to the viewpoint than the other. (We check this condition by the center of the internodes.)

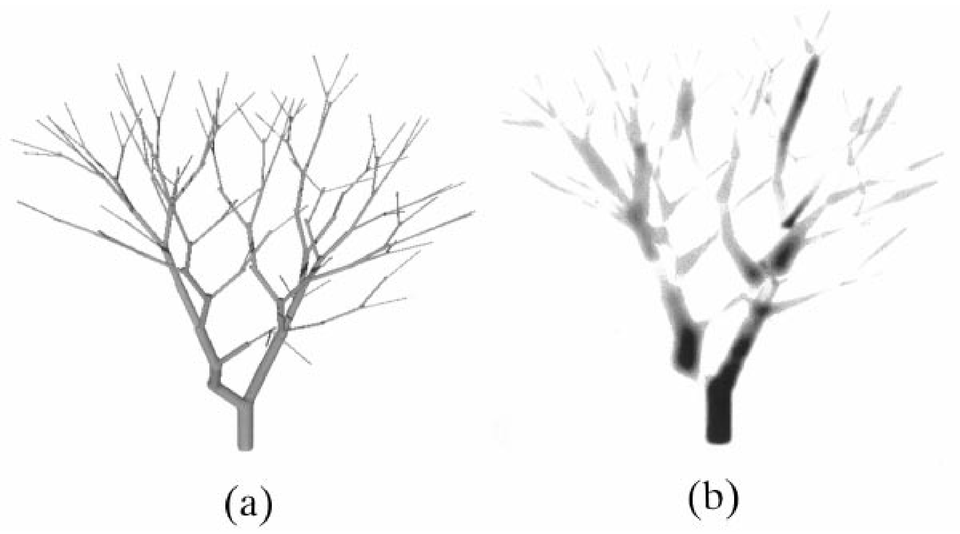


Figure 11. 3D Suibokuga*-like rendering of a tree*

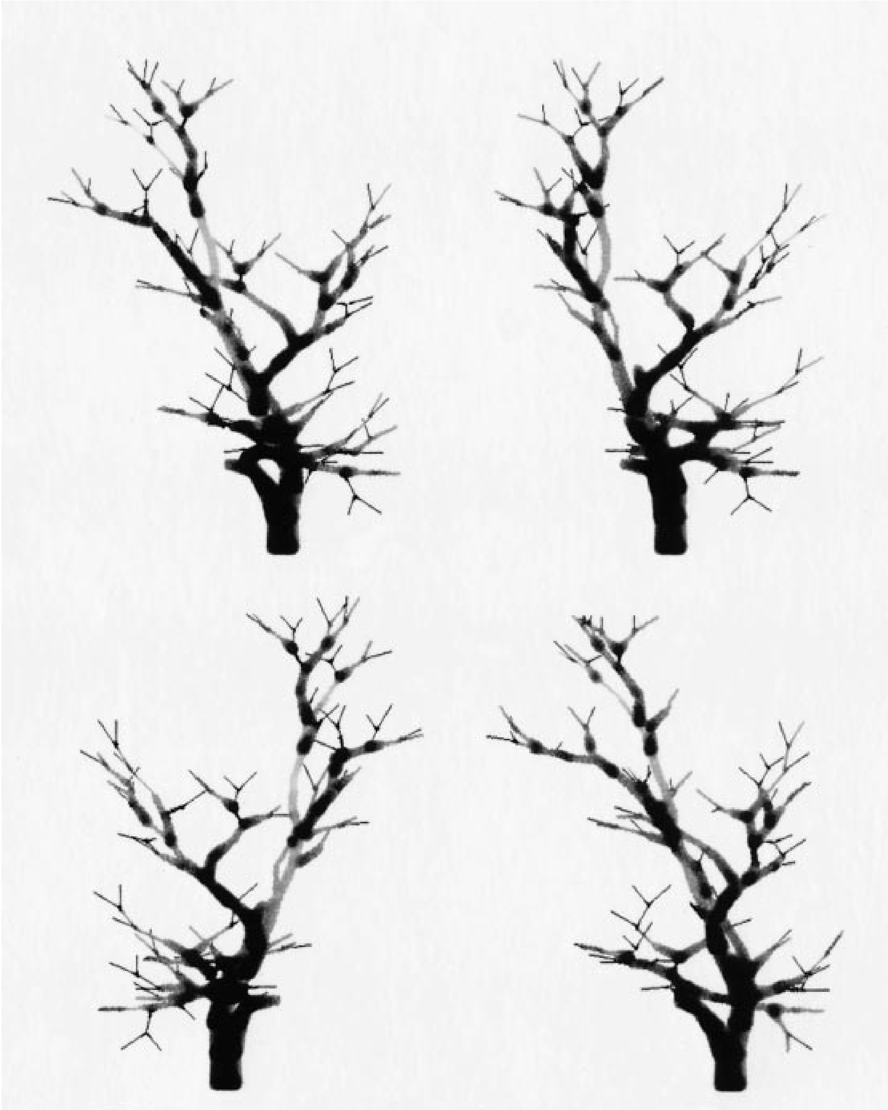


Figure 12. Suibokuga-*like images obtained by rotating viewpoint*

Step 2-4. If the angle between E and E_d is greater than the predefined threshold φ, then delete E_d from S and return.

Hidden Stroke Elimination

In our current rendering procedure we draw the split strokes $\{S'\}$ in descending order of their thickness, S'_1, S'_2, S'_3, ... S'_u, where a stroke's thickness is defined by the thickness of its first internode.

When a succeeding thin stroke S_k runs behind some preceding thick stroke S_j, it is natural that S'_k is drawn by skipping a brush across the ink painting of S'_j (see Figure 10). Thus we divide such strokes into several substrokes by the following algorithm.

Dividing Algorithm

Step 0. Prepare a Z-buffer Z and a screen D of the same resolution as Z. Construct a geometrical model of the tree by representing the internodes with polygonized cylinders having a diameter with a projected length arbitrarily greater than the expected width of the ink-painted strokes. Assign the index k of stroke S_k to the colours of polygons composing the geometrical model for the stroke S_k.

Figure 13. Effect of thickness of ink (using paper model which has no fibre)

Step 1. Render the tree by using a Z-buffer algorithm for the stroke S_k.
Step 2. For $k{:}=1$ to u, iterate the following steps.
Step 2-1. Scanning D along the projected stroke S'_k, find the colour numbers $c_1, c_2, c_3, \ldots, c_v$ smaller than k. (A brush should be skipped over the substrokes corresponding to the parts having these colours.)
Step 2-2. Divide the stroke S'_k into substrokes by 'erasing' the parts of the stroke S'_k corresponding to the colours $c_1, c_2, c_3, \ldots, c_v$.
Step 2-3. Replace S'_k in the set of strokes with the substrokes.

Simulated *Suibokuga* of Trees

Figure 11 shows simulated *Suibokuga*-like paintings of a tree. The left image is obtained by applying the Z-buffer algorithm to the polygon-based geometrical model produced by replacing an internode with a part of a polygonized cylindrical cone. Figure 12 shows various images of a single tree which are rendered by rotating the viewpoint. Figure 13 shows a different *Suibokuga*-like painting of the tree in Figure 11(a), simulated by thin ink on paper having fibres that contribute no effect. By contrast, Figure 14 shows the same tree with the effect of fibres taken into account.

Currently, the parameters of the proposed models are determined appropriately by trial and error, and fixed for one rendering. We took about 40 min to render a tree in Figure 12 or 13 on an HP8000/C160 workstation.

Conclusion

In this paper we have presented a behavioural model of water and ink particles based on a 2D *cellular automaton*

Figure 14. Effect of thickness of ink (using paper model which has fibres)

computational model, and its application to *Suibokuga*-like rendering of 3D trees. Several demonstrations have also been provided to show the efficacy of these methods. In the future our method might be extended to the treatment of some of the following painting techniques and phenomena.

- *On the behavioural model of ink*. Because ink is a kind of colloidal liquid, there appear more complex interesting phenomena than those simulated in this paper; for example, thick ink has high viscosity and thin ink causes separation of ink particles from water, thus creating more complex *Nijimi*.
- *On brush technique*. The current brush model represents *Chokuhitu* and *Sokuhitu* to good effect. Another usage suitable for painting jagged rocks or surfaces, *Kappitu* (literally 'a dried brush') might be considered. In *Kappitu* the brush is dipped in ink and quickly wiped off before being drawn across the paper. This separates the tip of the brush into several clusters and produces *chaotic* strokes.
- *On stroke technique.* In this paper an internode is painted with a single stroke, a painting technique called *Mokkotuhou*. *Korokuhou*, in which single strokes form the outlines of objects, might also be of interest.
- *On the kind of objects represented*. We need to extend the kind of paintable objects. For example, to represent *Sansuiga* (landscape paintings), which is one of the broad categories of *Suibokuga*, we must develop algorithms that can derive the appropriate strokes for mountains and water flows on the basis of their geometrical models.

References

1. R. Greene, 'The drawing prism: a versatile graphic input device', *Proceedings of SIGGRAPH 85*, 1985, pp. 103–110.
2. S. Strassmann, 'Hairy brushes', *Proceedings of SIGGRAPH 86*, 1986, pp. 225–232.
3. T. Saito and T. Takahashi, 'Comprehensible rendering of 3-D shapes', *Proceedings of SIGGRAPH 90*, 1990, pp. 197–206.
4. P. Haeberli, 'Paint by numbers: abstract image representations', *Proceedings of SIGGRAPH 90*, 1990, pp. 207–214.
5. Q. Guo and T. L. Kunii, 'Modeling the diffuse paintings of "sumie" ', in *Modeling in Computer-Graphics*, Springer, New York, 1991, pp. 329–338.
6. B. Pham, 'Expressive brush strokes', *CVGIP, Graphical Models and Image Processing*, **53**(1), 1–6 (1991).
7. [Coc92].
8. T. Nishita, S. Takita and E. Nakamae, 'A display algorithm of brush strokes using Bezier functions', in *Communicating with Virtual Worlds*, Springer, New York, 1993, pp. 244–257.
9. J. Zhang and Y. Ohno, 'An expression method for brush strokes using skeleton', *Information Processing Society of Japan SIG Notes*, **93-CG-61,** 1–8 (1993) (in Japanese).
10. G. Winkenbach and D. H. Salesin, 'Computer-generated pen-and-ink illustration', *Proceedings of SIGGRAPH 94*, 1994, pp. 91–100.
11. M. P. Salisbury, S. E. Anderson, R. Barzel and D. H. Salesin, 'Interactive pen-and-ink illustration', *Proceedings of SIGGRAPH 94*, 1994, pp. 101–108.
12. S. C. Hsu and I. H. H. Lee, 'Drawing and animation using skeletal strokes', *Proceedings of SIGGRAPH 94*, 1994, pp. 109–118.
13. 'Special issue on non-photorealistic rendering', *Information Processing Society of Japan SIG Notes*, **95-CG-76** (1995) (in Japanese).
14. J. Lansdown and S. Schofield, 'Expressive rendering: a review of non-photorealistic techniques', *IEEE Computer Graphics and Applications*, **15**(3), 29–37 (1995).
15. Q. Guo, 'Generating realistic calligraphy words', *Proceedings of the Sixth International Conference on Human–Computer Interaction (HCI International '95)*, 1995, pp. 129–134.
16. M. Salisbury, C. Anderson, D. Lischinski and D. H. Salesin, 'Scale-dependent reproduction of pen-and-ink illustrations', *Proceedings of SIGGRAPH 96*, 1996, pp. 461–468.
17. G. Winkenbach and D. H. Salesin, 'Rendering parametric surfaces in pen and ink', *Proceedings of SIGGRAPH 96*, 1996, pp. 469–476.
18. B. J. Meier, 'Painterly rendering for animation', *Proceedings of SIGGRAPH 96*, 1996, pp. 477–484.
19. M. P. Salisbury *et al.*, 'Orientable textures for image-based pen-and-ink illustration', *Proceedings of SIGGRAPH 97*, 1997, pp. 401–406.
20. P. Litwinowicz, 'Processing images and video for an impressionist effect', *Proceedings of SIGGRAPH 97*, 1997, pp. 407–414.
21. L. Markosian *et al.*, 'Real-time nonphotorealistic rendering', *Proceedings of SIGGRAPH 97*, 1997, pp. 414–420.
22. C. J. Curtis *et al.*, 'Computer-generated watercolor', *Proceedings of SIGGRAPH 97*, 1997, pp. 421–430.
23. D. Small, 'Simulating watercolor by modeling diffusion, pigment, and paper fibers', *SPIE Conference Proceedings*, **1460,** 140–146 (1991).
24. N. Chiba, K. Ohshida, K. Muraoka, M. Miura and N. Saito, 'A growth model having the abilities of growth-regulations for simulating visual nature of botanical trees', *Computers & Graphics*, **18**(4), 469–479 (1994).