Synthesis of cartoon fire based on hand-drawn samples

Yi Gu and Jinhui Yu*

State Key Lab of CAD&CG,
Zhejiang University, Hangzhou 310027, China
E-mail: guyi@cad.zju.edu.cn
E-mail: jhyu@cad.zju.edu.cn
*Corresponding author

Ren Peng

College of Computer Science and Technology,
Zhejiang University, Hangzhou 310027, China
E-mail: pengren@cs.zju.edu.cn

Abstract: In this paper we present an approach for synthesising and editing cartoon fire series based on hand-drawn samples. Our approach first extracts outlines of hand-drawn cartoon fire series and, based on the central reference skeleton, synthesises fire of arbitrary length by high level control on flame skeletons. With our approach users may edit cartoon fire series to achieve effects such as fire spreading and fire blown by wind from the side.

Keywords: cartoon fire; image synthesis; computer animation.

Reference to this paper should be made as follows: Gu, Y., Yu, J. and Peng, R. (2010) 'Synthesis of cartoon fire based on hand-drawn samples', *Int. J. Computer Applications in Technology*, Vol. 38, Nos. 1/2/3, pp.157–162.

Biographical notes: Yi Gu is currently a Master's student at Zhejiang University, China. He received his Bachelor Degree in Information and Computing Science from Shandong Economic University in 2008. His research interests include computer animation and stylised rendering.

Jinhui Yu is a Professor of Computer Science at Zhejiang University, China. He received his PhD in Computer Graphics from the University of Glasgow in 1999. His research interests include stylised computer animation and computer graphics art.

Ren Peng is an Associate Professor of Industrial Design at Zhejiang University, China. He received his Bachelor degree in Art Design from Jiangnan University in 1989. His research interests include industrial design, multimedia design and animation.

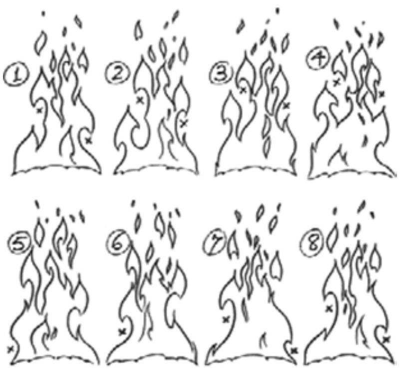
1 Introduction

In cartoon animation, fire is interpreted in a stylised manner and Figure 1 shows an example (Harold, 1981). Such fire series can be used directly to represent a bone-fire or a torch fire without wind effect. In some cases, however, fire flames not only move upward, but may also move in other directions, for instance, the bone-fire is blown by wind from side, or the torch is moved with hand. It is also necessary in some animations to represent fire spreading along a path, and fire flames in such a case become progressively bigger in size as they move. In hand-drawn animation representing these fire effects requires animators to draw different fire series; this may involve a lot of manual work. A significant reduction of animator's load can be expected if we can synthesis these fire effects by editing existing hand-drawn fire samples.

A straightforward method to do this is editing hand-drawn fire samples with some commercial software like Photoshop. Take the wind effect for example, we can deform a fire image (the third sample in Figure 1) through the shear filter in Photoshop by alternating the shape of the central reference skeleton of the image, as shown on the left of Figure 2. The deformed image can be used to simulate a light wind effect in animation.

To simulate a strong wind effect, we adjust the central reference skeleton further in the shear filter to get a more deformed fire image; as shown on the right of Figure 2, the resultant animation looks unnatural, however, because flames become progressively thinner and longer near the top of the fire.

Figure 1 Hand-drawn cartoon fire



In this paper we present an approach that allows users to synthesise cartoon fire effects based on hand-drawn samples in a simple manner. Our approach is able to preserve the artistic quality of the animator's hand-drawings and the resultant animation look more natural. The core idea of our approach is that by using axis transformation to deal with flames in hand-drawn fire samples, we gain the high level control over the size and orientation of flames by defining a local coordinate space within the image of the fire. We demonstrate the power of this approach by synthesising fire spreading and wind effects.

Figure 2 Fire image deformed using shear filter

2 Related work

During the last two decades methods of depicting gaseous phenomena, such as haze, fog, clouds, dust, smoke and flames, have been studied by many workers. However, most of them have aimed at realistic representation of the phenomena, and there were few results relevant to the stylised representation of fire in the cartoon animation.

Reeves (1983) simulated firework effects by using particle systems. Perlin (1985) generated a solar corona using a turbulence function. Inakage (1990) presented a technique based on a physical model of combustion, and succeeded in the photo-realistic representation of the flames of a candle and a Bunsen burner. Ohshima and Itahashi (1988) proposed a simulation method employing 2D fractal texture and other processing techniques for generating animations of flames, such as in a bonfire and in candle flames. Augui et al. (1991) presented cellular automata with simple state transition rules for simulating flames like those of an alcohol lamp. Gardner (1992) modelled fire with fractal ellipsoids. Sakas and Gerth (1992) and Sakas (1993) proposed simulation methods based on the spectral theory of turbulence. Nishita et al. (1987) presented a display method for producing a still image of smoke. Chiba et al. (1994) simulated 2-D flames and smoke by visualising turbulence. Stam and Fiume (1995) used diffusion processes to animate fire and other gas phenomena. Takahashi et al. (1997) proposed a rock form modelling method based on the simulation of the growing process of columnar joints. Stam (1999) presented a stable algorithm that solves the full Navier-Stokes equations for gas and fire animations. Foster and Fedkiw (2001) and Nguyen et al. (2002) used Navier-Stokes equations to independently model both vapourised fuel and hot gaseous products. Beaudoin et al. (2001) integrated a set of techniques, including a method for simulating spreading on polygonal meshes, to produce realistic-looking animations of burning objects. Melek and Keyser (2002) used modified interactive fluid dynamics solver to describe the motion of a 3-gas system. Zhao et al. (2003) animated fire propagation using a volumetric fire propagating model. Ihrke and Magnor (2004) presented a tomographic method for reconstructing a volumetric model from multiple images of fire.

In contrast to realistic fire generation, only a few approaches have been proposed for stylistic fire generation. Yu and Patterson (1996) presented a framework for generating animation sequences which match the hand-drawn series. Di Fiore et al. (2004) constructed time-dependent visual components, such as flames, drops, puffs, etc and animated them along the path specified by the user.

Our work is partially inspired by the flow-based video synthesis and editing proposed in Kiran et al. (2004), the main difference between their approach and ours is in the choice of visual elements for synthesising fire: the first takes the dynamics and texture variation of the particles along user-defined flow lines in the input video as visual elements, while ours takes hand-drawn fire samples as visual elements. Since flames of conical shapes in hand-drawn fire samples (Figure 1) are bigger than particles taken from video, the method for synthesising realistic fire proposed in Kiran et al. (2004) cannot be used for dealing with deformation of flames in hand-drawn fire samples. We, therefore, propose a new approach to synthesising fire spreading and wind effects. Our goal is to provide the user with a tool that allows the high level control over the size and orientation of flames, and the synthesised results have the same quality as the hand-drawn fire series.

3 Our approach

Our approach involves edge detection, axial deformation, boundary handling and rendering, as detailed in the next three subsections.

3.1 Edge detection

We first use a painting system (e.g., Photoshop) to paint hand-drawn samples black to get binary images of fire series. Next, we adopt an edge detection algorithm based on 8-connected neighbours searching theme to extract the boundary points of binary images of fire sequences. The 8-connected neighbours are labelled as follows:

–1, –1	0, –1	1,–1
–1, 0	0, 0	1, 0
–1, 1	0, 1	1, 1

The algorithm involves the following steps:

Step 1: Scan the image and find an unmarked edge point in the binary images, mark it if it is an edge point.

Step 2: Put the marked point as current search centre and take (0, –1) as the initial search direction.

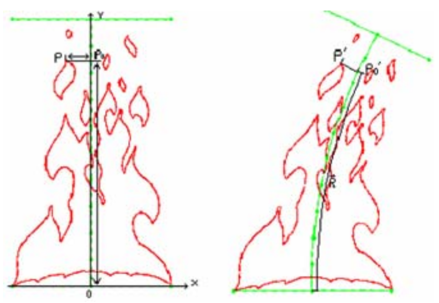
Step 3: Search for the edge point in its 8-connected neighbours clockwise, go to Step 4 if there is no unmarked edge point found, otherwise mark the searched point and set it as the new search centre, repeat Step 3 with the new search direction – the last search centre.

Step 4: Go to Step 1 until there is no unmarked edge points.

3.2 Axial deformation

In order to gain control over the fire image represented with boundary points of the fire body and top flames, we adopt the axial deformation theme by Peng et al. (1997) to deal with the deformation of fire series. First, we introduce an axis in the central part of the fire image, as shown in Figure 3. The axis is represented by a *B*-spline curve which, in the un-deformed case, coincides with the *Y*-axis in the local coordinate system *O-XY*, we then take the *Y*-axis as the deformable axial curve and embed the fire boundary in the deformation space defined by the axial curve.

Figure 3 Fire boundary determined by the axial deformation (see online version for colours)



Let $\vec{R}(u)=(x(u),y(u))$ be a *B*-spline curve defined over $[u_0, u_1]$ which interpolates its two end points and $\vec{N}(u)$ be the normal function of $\vec{R}(u)$, the arc length $s(u)$ of the curve $\vec{R}(u)$ can be expressed as:

$$s(u)=\int_{u1}^{u_0}\left\|\vec{R}'(u)\right\|^2 \mathrm{d}u = \int_{u1}^{u_0}\sqrt{(x'(u))^2+(y'(u))^2}\,\mathrm{d}u. \tag{1}$$

Let $P(x, y)$ be a sample point on the fire edge, its curve parameter u_y corresponding to the arc length y in the un-deformed case can be obtained by

$$u_y=(y-s_0)/(s_1-s_0) \tag{2}$$

where s_0 and s_1 correspond the height and the bottom of the fire image on the axial curve. With u_y we can find a point $R(u_y)$ on the deformed axial curve, and the new position of point in the deformed space can be determined by

$$\vec{P}'=\vec{R}(u_y)+x\cdot\vec{N}(u_y). \tag{3}$$

The deformed point $P'(x', y')$ along the axis of curve $(X(i), Y(i))$ can be therefore evaluated by the formula:

$$\begin{cases} x'=X(y)+d\cdot\cos(\theta(y)) \\ y'=Y(y)+d\cdot\sin(\theta(y)) \end{cases} \tag{4}$$

where

$$d=x\cdot\left(\mu_{\text{bottom}}-\frac{y}{fh}\cdot(\mu_{\text{bottom}}-\mu_{\text{top}})\right) \tag{5}$$

in which fh corresponds to the height of the fire body.

3.3 Boundary handling and rendering

To represent the transparent effect near the bottom of the fire body, we use alpha blending between the bottom part of the fire body and the background. To improve the realism of the fire near the bottom part of the fire body, we add perturbations to the bottom border to get an irregular boundary that varies with time.

In our approach the sampling points $p_i(x_i, y_i)$ of the bottom line are evaluated by:

$$\begin{cases} x_i=fw\cdot\dfrac{i-N/2}{N}+\Delta\left(\dfrac{fw}{2N}\right) \\ y_i=\sqrt{1-(x_i-x_0)^2}\cdot(0.75+\Delta(0.25)) \end{cases} \tag{6}$$

where N is the number of sampling points, fw is the width of the fire body, and $\Delta(u)$ is a random function with the value in the range $[0, u]$.

In hand-drawn fire series, the fire body is painted with the gradient colour from orange in the centre to red near the boundary parts. To simulate this effect, we estimate the centre $\vec{P}(x,y)$ of the gradient colour by the following equations:

$$\begin{cases} x=\dfrac{fw}{8}-\Delta\left(\dfrac{fw}{4}\right) \\ y=\dfrac{fh}{5}+\Delta\left(\dfrac{fh}{10}\right) \end{cases}. \tag{7}$$

We then blend the orange and red colours using a factor $\alpha 1=\exp(-\|\vec{P}-\vec{P}'\|)$ inside the fire body. A few small brighter colours are added randomly in the bottom region of the fire body, which are blended with their surrounding part colours by a factor $\alpha 2 = \exp(-\|\vec{P} - \vec{P'}\|)^2$, to enhance the liveliness of the fire body.

4 Simulation of the wind effect

To simulate the effects of wind on fire flames, we need to define the skeleton that governs the dynamics of the fire body first. Our system allows users to control the skeleton dynamics either automatically with models or manually.

In this section, we propose a very simple model to simulate the wind effect of fire. First, we define a delta angle $\Delta\theta$ and use the increment angle $\theta = \exp[-(1 - i/N)]\cdot\Delta\theta$ progressively to rotate the vector formed by every two adjacent control points on the vertical reference skeleton, where i is the index of the ith vector on the reference skeleton and N is the number of control points in skeleton. The rotated vectors are connected in succession to get the bend skeleton. Clearly, by varying the value of $\Delta\theta$ we can change the degree of the skeleton bended. To simulate the wind's effect on fire we need just to modulate the value of $\Delta\theta$ with some time varying models, and the following is an example in which the control points $\vec{P}_i(x_i, y_i)$ can be evaluated by

$$\begin{cases} x_i = x_{i-1} + d\cdot\cos\left(\theta + \Delta\left(\frac{\Delta\theta}{8}\right)\right) \\ y_i = y_{i-1} + d\cdot\sin\left(\theta + \Delta\left(\frac{\Delta\theta}{8}\right)\right) \end{cases} \tag{8}$$

where d is the distance between $\vec{P}_i$ and $\vec{P}_{i-1}$.

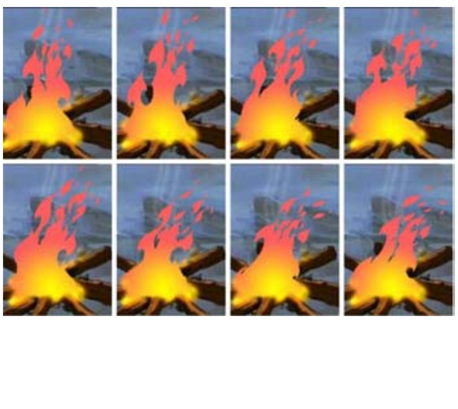
It should be pointed out that different time varying models can be used in a combinational manner. If we intend to make an animation of 56 frames in length for instance, the first 16 frames animate the fire series shown in Figure 1 in two cycles, and the next 40 frames animate the fire series with the skeleton deformed by modulating the delta angle with the following model:

$$\Delta\theta = \exp[-(t-8)^2]\cdot\theta_{\max} \tag{9}$$

where $\theta_{\max}$ is the maximum angle that the skeleton is bent.

A strip of 8 frames for animating wind effect of fire is shown in Figure 4 from which we can see that fire flames are deformed in a more natural manner than those in Figure 2.

Figure 4 A strip of 8 frames for animating wind effect (see online version for colours)



5 Simulation of fire spreading

During the spreading of the fire, the fire flames become progressively bigger in size as they move along the path, and then reach the normal size.

To synthesise the spreading of the fire, we first define the fire body along the X axis for reference, next, we divide the fire body into two parts: the stable part in which fire samples are placed with equal intervals and their heights remain roughly unchanged, and the transit part in which fire samples are places with intervals proportional to their heights which change progressively from the head of the fire to the stable part. The intervals between two adjacent samples should be small enough to ensure the overlap of two neighbouring fire samples on the reference axis, otherwise the synthesised fire may look 'broken' on the fire body along the reference axis.

Finally, we move the synthesised fire body along the predefined path in the scene by mapping the positions of the fire samples on the reference axis to the path of fire spreading on the scene. Our interface allows the user to specify parameters to control the size of the synthesised fire over the different parts along the path and the speed of fire spreading.

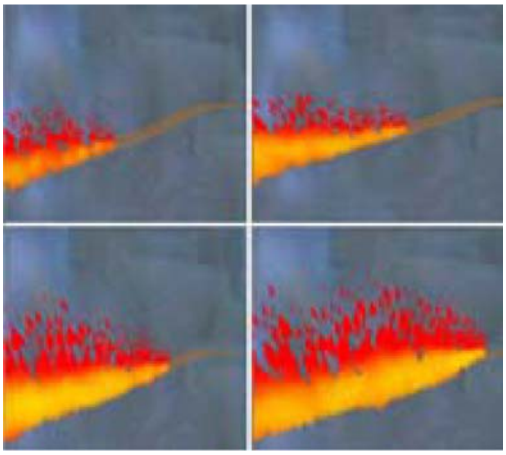
The procedure for animating fire spreading can be expressed with the following pseudo codes:

```
Predefined parameters:
T , timer,
Speed , Speed of spreading.
x_head , start position of fire.
h_head , height of first flames.
N_head , number of samples in the transit part.
h_stable , height of fire for stable flames.
d_stable , distance between two neighboring samples.
FireSpreading()
{
    h_i = h_head , x_i ;
    //height and position of the i th sample.
    while( T )
    {
        x_head += speed ;
        x_i = x_head ;
        while( x_i > 0 ){
            if( i < N_head ){
                k = 1 - (1 - h_head/h_stable) · sin(π/2 · i/N_head);
                h_i = k · h_stable ;
                x_i -= k · d_stable ;
            }else{
                h_i = h_stable ;
                x_i -= d_stable ;
            }
            Draw_Sample( x_i , h_i );
            Inc( i );
        }
        Transparentize_BottomBorder();
        Add_BrightColors();
    } //end for while
} //end of the procedure
```

To simulate the transparent effect near the bottom of the fire body, we use alpha blending between the bottom part of the fire body and the background. Since the path of fire spreading is determined by interpolating some control points specified by the user with a spline, we add perturbations to the path line to improve the realism of the fire near the bottom part of the fire body.

Bright colours are finally added onto the samples. Figure 5 shows a 4 frames strip of fire spreading along a path synthesised by our approach.

Figure 5 A strip of 4 frames of fire spreading (see online version for colours)



6 Conclusion and future work

In this paper we present a framework for synthesising cartoon fire effects based on hand-drawn cartoon fire samples. Our system provides a tool for users with little training for the cartoon effects drawings, and enables them to produce impressive cartoon effects such as fire blown by wind and fire spreading.

Since cartoon animation is stylised representation of the reality, the visual interpretation of cartoon fire is diverse from animator to animator. In the case where users intend to synthesise a different style of cartoon fire based on hand-drawn samples, our system remains appropriate.

In our future work, we intend to study the methods to deal interactions between obstacle objects and the cartoon fire.

Acknowledgements

This work is supported by the State Key Program of National Natural Science Foundation of China (No. 60933007), the Key Technologies R&D Program of China (No. 2007BAH11B02) and Zhejiang Provincial Science and Technology Plan Project (No. 2007C21043).

References

Augui, T., Kohno, Y. and Nakajima, M. (1991) 'Generating 2-dimensional flame images in computer graphics', *Trans. IECE of Japan*, Vol. 2, pp.184–189.

Beaudoin, P., Paquet, S. and Poulin, P. (2001) 'Realistic and controllable fire simulation', *Proceedings of Graphics Interface*, Ottawa, Canada, pp.159–166.

Chiba, N., Muraoka, K., Takahashi, H. and Miura, M. (1994) 'Two dimensional visual simulation of flames, smoke and the spread of fire', *Visualization and Computer Animation,* Vol. 5, pp.37–53.

Di Fiore, F., Claes, J. and Van Reeth, F. (2004) 'A framework for user control on stylised animation of gaseous phenomena', *Computer Animation and Virtual World*, Vol. 18, pp.405–414.

Foster, N. and Fedkiw, R. (2001) 'Practical animation of liquids', *Proceedings of SIGGRAPH*, Los Angeles, USA, pp.23–30.

Gardner, G. (1992) 'Fractal ellipsoid fire', *SIGGRAPH Video Review*, Vol. 14, pp.184–189.

Harold, W. (1981) *Timing for Animation*, Focal Press Limited, London.

Ihrke, I. and Magnor, M. (2004) 'Image-based tomographic reconstruction of flames', *Proceedings of Eurographics/ ACM SIGGRAPH Symposium on Computer Animation*, Grenoble, France, pp.365–373.

Inakage, M. (1990) 'A simple model of flames', *Proceedings of Computer Graphics International*, Singapore, pp.71–81.

Kiran, S., Steven, B., Seitz, M., Jessica, K., Pradeep, H. and Khosla, K. (2004) 'Flow-based video synthesis and editing', *Proceedings of SIGGRAPH*, Los Angeles, USA, pp.360–363.

Melek, Z. and Keyser, J. (2002) 'Interactive simulation of fire', *Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, Beijing, China, pp.462–466.

Nguyen, D.Q., Fedkiw, R. and Jensen, H.W. (2002) 'Physically based modeling and animation of fire', *ACM Transactions on Graphics*, Vol. 21, pp.721–728.

Nishita, T., Miyawaki, Y. and Nakamae, E. (1987) 'A shading model for atmospheric scattering considering luminous intensity distribution of light sources', *Computer Graphic*s, Vol. 21, pp.303–310.

Ohshima, T. and Itahashi, S. (1988) 'Texture animation', *Proceedings of NICOGRAPH*, Tokyo, Japan, pp.110–119.

Peng, Q-S., Jin, X-G. and Feng, J-Q. (1997) 'Arc-length-based axial deformation and length preserved animation', *Proceedings of Computer Animation*, Geneva, Switzerland, pp.86–92.

Perlin, K. (1985) 'An image synthesizer', *Computer Graphics*, Vol. 19, pp.278–296.

Reeves, W.T. (1983) 'Particle system – a technique for modeling a class of fussy objects', *Computer Graphics*, Vol. 17, pp.358–376.

Sakas, G. (1993) 'Modeling and animating turbulent gaseous phenomena', *The Visual Computer*, Vol. 9, pp.200–212.

Sakas, G. and Gerth, M. (1992) 'Sampling and anti-aliasing of discrete 3-D volume density textures', *Proceedings of EUROGRAPHICS*, Cambridge, UK, pp.107–117.

Stam, J. and Fiume, E. (1995) 'Depicting fire and other gaseous phenomena using diffusion', *Proceedings of SIGGRAPH*, Los Angeles, USA, pp.129–136.

Stam, J. (1999) 'Stable fluids', *Proceedings of SIGGRAPH*, Los Angeles, USA, pp.121–128.

Takahashi, J., Takahashi, H. and Chiba, N. (1997) 'Image synthesis of flickering scenes including simulated flames', *IEICE Transactions on Information Systems*, Vol. E80-D, pp.1102–1108.

Yu, J-H. and Patterson, J.W. (1996) 'A fire model for 2-D computer animation', *Proceedings of Computer Animation and Simulation*, Geneva, Switzerland, pp.49–60.

Zhao, Y., Wei, X., Fan, Z., Kaufman, A. and Qin, H. (2003) 'Voxels on fire', *Proceedings of 14th IEEE Visualization Conference*, Seattle, USA, pp.271–278.