1000-9825/2001/12(07)0960-07 ©2001 Journal of Software 软件学报 Vol. 12, No. 7

A Cartoon Water Model for Stereo Animation*

YU Jin-hui, XU Xiao-gang, WAN Hua-gen, PENG Qun-sheng

(*State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou* 310027, *China*)

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

http://cad.zju.edu.cn

Received July 5, 2000; accepted February 26, 2001

Abstract: A procedural model is presented in this paper for generating water in the stereo animation. Here the emphasis is not on a realistic water animation, but a stylistic representation of the water as used in cartoon. In this modeling approach, hand-drawn water series are decomposed into stationary and moving components to abstract static and dynamic structures associated with water forms. Based on those structures, a hierarchical framework is constructed to generate 2D animation series that match those hand-drawn series. The model offers flexible control over the structure, shape, color, timing etc in a sequence of water effects to produce some other cartoon effects such as water jet by adding an additional level in the hierarchical framework. A significant contribution of this work is the extension of cartoon water effects to 3D, thus stereo cartoon water animations can be generated with ease in the CAVE by use of the depth information in the 3D cartoon water model.

Key words: water model; cartoon animation; stereo animation; computer animation

In cartoon animation, animated effects like water, fire and rain etc simulate our environment in a stylized manner to add realism, drama and atmosphere to the animation, and thus they are the important elements for an animator to master. For a large-scale and high quality production, it requires highly skilled professional effects animators to accomplish the task, which means a high cost in a high wage economy. With the usual limitations on time and budget, effects drawings are often made cycles and this means repeated cycles are required for the length of time. Because of this, the effects usually look mechanical.

In the past two decades, methods for representing 3D water surfaces, ocean waves, waterfalls have been studied by many workers. Whitted animated realistic reflections from ripples in a small pool by using ray tracing.[1] The ripple were created by bump mapping the flat pool surface, perturbing the surface normal according to a single sinusoidal function. Perlin has used bump mapping with a richer texture map to convincingly simulate the appearance of the ocean surface as one might see it from an aircraft well out to the sea.[2] Max used a "height field" algorithm to render explicitly modeled wave surfaces for his film "Carla's Island".[3] His wave model consisted of

* Supported by the National Natural Science Foundation of China under Grant No. 60073024 (国家自然科学基金); the SRF for ROCS, EM under Grant No.[2000] 367 (留学回国人员科研启动基金).

YU Jin-hui was born in 1960. He received his Ph.D. degree from the Department of Computing Science from the University of Glasgow, U.K., in 1999. He is an associate professor now. His research interests include computer animation and computer art. **XU Xiao-gang** was born in 1967. He received his Ph.D. degree from Dalian Institute of Science and Technology in 1999. He is a lecturer and post doctor. His research interests include computer image processing and computer graphics. **WAN Hua-gen** was born in 1968. He received his Ph.D. degree from Zhejiang University in 1999. He is a lecturer now. His research interests include CAD and computer animation. **PENG Qun-sheng** was born in 1947. He received his Ph.D. degree from the School of Computing Studies, the University of East Anglia, U.K., in 1983. He is a professor and his research interests include computer graphics, computer animation and virtual reality.

several superimposed linear sinusoidal waves simulating ocean waves of low amplitude. Peachey presented a model of ocean waves which is capable of simulating the appearance and behavior of waves as they approach a slopping beach, steepening, breaking, and producing a spray of water droplets from the crests of the waves.[4] Based on the Gerstner[5], or Rankine, model where particles of water describe circular or elliptical stationary orbits, Fournier used a parametric surface to model the ocean surface[6] including the effects of depth such as refraction and surf, and some of the effects of wind. In Ref. [7], Mastin *et al.* used a model based on the work of Pierson and Moskowitz who used wind-driven sea spectra, derived from observed data, to describe the motion of deep ocean waves in fully developed wind seas. Wave animation is invoked by manipulating the phase of the Fourier transforms. Sims developed some general tools for animating and rendering particle systems that permit both kinematic and dynamic control of particles. They are used to create effects such as wind, snow, waterfall and fire.[8] Mallinder presented a method of implicitly storing particles, and illustrates its use with the modeling of larger waterfalls.[9] Xu *et al.* simulated flowing water and waves based on a physical approach.[10] All of efforts mentioned above, however, focused on realistic representation of the phenomena and relatively little time has been spent in modeling the appearance of cartoon water effects, to say nothing of generating stereo cartoon water animations.

The difficulties we encounter in modeling the cartoon water are as follows:

1. The actions of the cartoon water involve random components thus causing discontinuity of features across more than a few frames, while automatic methods for in-betweening[11~17] rely on some degree of continuity in both geometry and time, therefore, these automatic methods will fail utterly in such cases.

2. Cartoon water effects are 2D stylized drawings by the animator, therefore they are difficult to be presented by particles and there is no physical knowledge available to help us to build the model.

Nevertheless, there are several properties common to all cartoon effects. The most salient one is that all they are stylized representation of natural phenomena, thus their movements are consistent and predictable. It is possible for us to find out the underlying model to create the effects procedurally. Models for dealing with the cartoon fire and smoke, for instance, have been proposed by us.[18,19]

In this paper we present a model capable of reproducing the cartoon water in a 3D scene. The stereo cartoon water series can be created by running the model in real-time in the CAVE, thus we explore a new world-stereo cartoon animation.

In the following sections, we first describe how to decompose animator's hand drawings into different components from which we can abstract the static and dynamic structures of those drawings. Then, we construct the cartoon water model through synthetic computational means based on above decomposition. Finally, we describe how to reconstruct the 3D cartoon water model with which we are able to generate stereo cartoon water animations. Simultaneously, some examples are given in the paper.

1 Hand Drawn Cartoon Water

In this section we begin with tracing the animator's drawing process and construct our model using the information abstracted from the decomposition of the hand-drawn cartoon water series.

In the studio, the effect of flowing water for instance can be created by first establishing a plain background color for the water and then animating a series of shapes across it to simulate the feeling of movement we get when we watch a running stream. The shape can be abstract, stylized, or realistic (In contrast with photo-realistic representation, all cartoon representations can be regarded as the stylized which however can be classified further into the abstract, stylized, or realistic by the animator). Whatever shapes are selected, they should flow in a consistent way, following the chosen path of action. If the water action is more violent than a simple flowing movement, the animator can add another level of animation, depicting white caps at the top of the waves. Figure 1 shows two frames of different flowing water shapes drawn by the animator[20], the first can be regarded as the realistic one and the second stylized one.

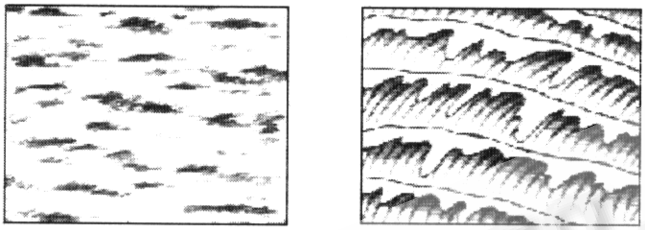

Fig. 1 Hand painted cartoon water

In this paper we focus on developing a flowing water model synthesizing the second drawing in Fig. 1. In light of proceeding description, we decompose the hand drawing into the following components in terms of stationary parts such as path *boundary curves* and moving parts such as *water shapes*, *caps* and *foam*, which all put together can be expressed by the following hierarchical structure:

Model→Boundary curve→Water shapes→Caps→Foam

The detailed implementation of the model will be given in the next section.

2 Implementation Framework

2.1 Boundary curves

Boundary curves are drawn to confine the path of the running water according to the scene, thus in our model we require the users to specify a few control points defining the main shape of the path which are interpolated by splines to get $LeftBD_i$ and $RightBD_i$ ($i=1,\ldots,BdN$) to draw the curves as shown in Fig. 2, where BdN is the number of points contained in two curves, respectively. With the index assigned in the two arrays we can define the reference direction of the flowing water. The number of interpolated points between the two successive control points is determined that it should be sufficient to ensure the motion continuity of horizontal waves, because the movements of the horizontal waves are represented by the displacement between two successive interpolated points on the boundary curves.

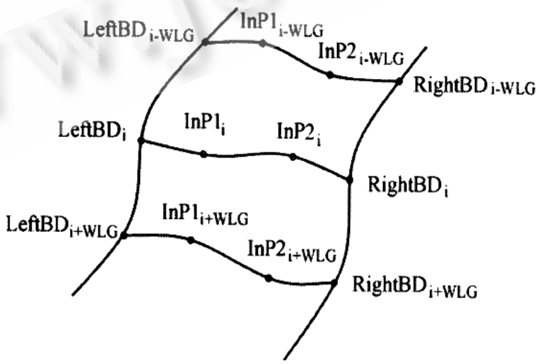

Fig. 2 Boundary curves and water shapes

2.2 Water shapes

Water shapes span in series the water path. The distance between two successive water shapes is defined by WLG, which is a number to count certain points contained, say, between $LeftBD_i$ and $LeftBD_i+WLG$. Provided with BdN and WLG, the number of water shapes contained along the water path can be simply obtained by $HrzN=BdN/WLG$.

In designing individual water shape, we first take $LeftBD_i$ and $RightBD_i$ $(i=1,\ldots,HrzN)$ as extremes, then introduce two intermediate points InP_1 and InP_2 (Fig. 2) by the following formulae:

$$InP_{1,i}=LeftBD_i+u_1(RightBD_i-LeftBD_i)+rnd(r)V(InP)$$

$$InP_{2,i}=LeftBD_i+u_2(RightBD_i-LeftBD_i)+rnd(r)V(InP)$$

where u_1 and u_2 are positional parameters which together with random variables of variance $V(InP)$ would make a wary line. Finally, we interpolate those four points, $LeftBD_i$, $InP_{1,i}$, $InP_{2,i}$ and $RightBD_i$ with splines to get $HrzWV_{i,j}$, $(i=1,\ldots,HrzN,\ j=1,\ldots,HrzM)$ that draw the water shape, where $HrzM$ is the number of points contained in the interpolated curve.

2.3 Caps

Modeling of each white cap at the top of water shapes involves two parts. One part is concerned with the front shape of the cap, which can be generated by the following steps:

1. Set $j=1$.

2. Generate a positional parameter p varying randomly between [0.25, 0.55] (The magnitude of p is determined through experiment).

3. Interpolate $HrzWV_{i,j}$ and $HrzWV_{i+1,j}$ linearly with above p to get a control point $CapCP_j$.

4. Increase index j and repeat above two steps to generate the successive control point $FCapCP_{j+1}$ until j reaches $HrzM$.

5. Interpolate $FCapCP_{i,j}(j=1,\ldots,HrzM)$ with a spline to get $FCap_{i,k}(k=1,\ldots,FCapN)$ (where $FCapN$ is the number of interpolated points) that draws the final front boundary line of the cap.

The other part is concerned with the back shape of the cap which is depicted readily by the water shape $HrzWV_{i,j}$, $(j=1,\ldots,HrzM)$. The complete contour of the ith cap $Cap_{i,l}(l=1,\ldots,CapN)$ can be derived by appending $FCap_{i,k}(k=1,\ldots,FCapN)$ to $HrzWV_{i,j}(j=1,\ldots,HrzM)$ where $CapN=FCapN+HrzM$ and all caps can be generated by varying index i from 1 to $NrzN$.

2.4 Foam

Foam can be represented by small circles with variations in size and distributed surrounding the contour of the white caps with a simple model.

2.5 Structure of the model

The structure of the model can be expressed as follows:

1. Initialize the model by specifying the following parameters:

   (a) BdN, number of points contained in two boundary curves;

   (b) $HrzM$, number of points contained in water shapes;

   (c) WLG, distance between water shapes;

   (d) Spd, moving speed of horizontal waves, where $|Spd|$ is the distance covered by Spd points on the boundary curve;

   (e) $HrzN=BdN/WLG$, number of water shapes.

2. Specify two sets of control points;

3. Interpolate control points to generate two boundary curves $LeftBD_i$, $RightBD_i$, $i=1,\ldots,BdN$.

For each frame t:

1. Generate water shapes $HrzWV_{i,j}(t)=HrzWV_{i+t*Spd,j}$.
2. Generate caps $Cap_{i,j}(t)$.
3. Distribute foam.

End (of each frame).

3 Coloring

In comparison with shape, color is less important for the water movement. This is true when we see early black and white animation. Even line drawn sequences by themselves can produce the illusion of movements. But coloring does improve the visual quality of the animation. One way to color water is using a plain color such as light blue to fill water area, dark blue to water shapes, white color to draw caps and the result as shown in Fig. 3 seems satisfactory. In this example some textures are added between water shapes according to a simple model, which makes the resultant images look more close to its hand drawn counter part (the second drawing in Fig. 1). An alternative method to color water is using gradient color varying from light to dark blue to fill the area surrounded by two adjacent water shapes and boundary curves in which case we discard textures between the water shapes and get a variant of flowing water derived from the same model (Fig. 4). Furthermore, effects such as sun rise and sunset can be simulated by simply changing the colors of flowing water over time t.

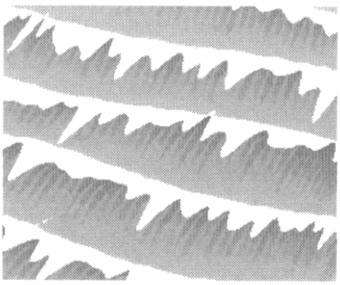

Fig. 3. Cartoon water with plain color

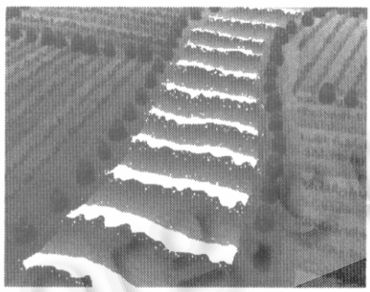

Fig. 4. Cartoon water with gradient color

4 Reconstruction of the Model in 3D

Since our model is derived from 2D drawings, it is naturally able to recreate 2D cartoon water effects (Fig. 3). In order to generate stereo animations, it is necessary for us to reconstruct the model in 3D. We first map control points of the two boundary curves in the existing 2D model to 3D world coordinates using

$$x'=x,\ y'=0,\ z'=y,$$

where x, y denote 2D coordinates and x', y', z' denote 3D world coordinates respectively. Next, we move those control points, on the $X'-Z'$ plane in the world coordinates to tune the shape of the boundary curves according to the scene. Provided with the 3D coordinates of the control points, we are able to calculate the third dimension of other parts such as water shapes, caps *etc* according to its internal hierarchical structure with ease. As a result, we achieve 3D cartoon water animation, as shown in Fig. 5, a strip of 6 frames sampled from our 3D water animation of 170 frames, and from which we can see that our 3D cartoon water animation allows arbitrary movement of the virtual camera. The animated effect looks correct when we play back water series generated by our model.

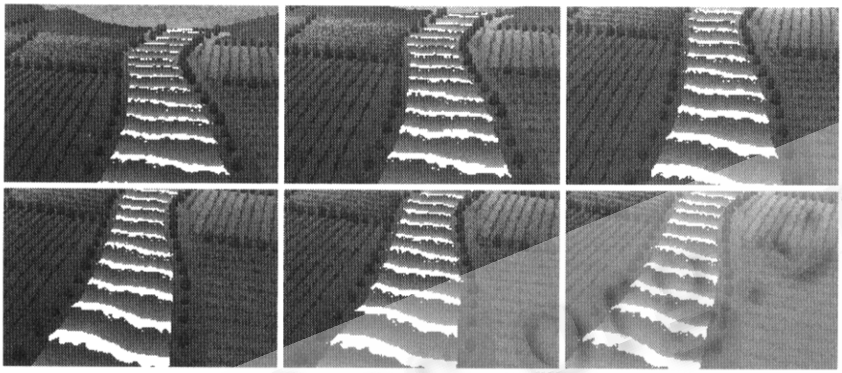

Fig. 5 A strip of 6 frames of 3D flowing water

Now that the model is reconstructed in 3D, we can create the stereo cartoon water animations in the CAVE with great ease using the depth information provided by the 3D model because our model is simple in the structure and can be run in the real-time.

5 Conclusions

In this paper we presented the result of research dealing with cartoon water based on hand drawing process. Our work shows that this modeling approach is a powerful paradigm for hand-painted image synthesis, because we capture the essential quality of the cartoon water motion constrained by hand drawing along, say, how different parts of the drawing are distributed in time and space.

Our hierarchical model allows flexible controls over shape, color, timing *etc* in a sequence of water effects, thus can be used directly for generating flowing water, including waterfalls, running taps, moving streams and rivers, procedurally. Moreover, the model is extensible by use of additional levels to generate some other cartoon effects such as water jet (Fig. 6). Actually the water jet resembles very much the flowing water in structure composed of boundary curves, water shapes and drops. In addition to water moving along the parabola, the water jet may animate itself, such as the jet coming from the nozzle held by a fireman. This can be dealt with simply by introducing a skeleton to governor the parabola and the two boundary curves can be added to the skeleton according to a simple model.

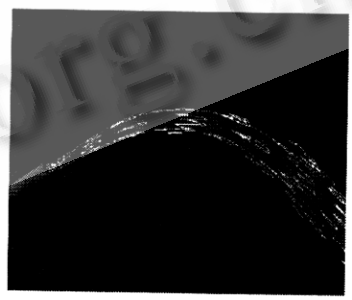

Fig. 6 A frame of water jet

A significant contribution of our work is that we extended 2D cartoon water effects to 3D. This novel result not only improves hand-drawn effects because it is almost impossible to achieve this by hand-drawn animations, but also enables us to animate effects with CAVE to generate stereo cartoon animations and the situation imagined hardly in the traditional hand-drawn animations.

References:

[1] Whitted, T. An improved illumination model for shaded display. Communications on the ACM, 1980,23:343～349.

[2] Perlin, K. An image synthesizer. Proceedings of the ACM Computer Graphics, 1985,19:287～286.

[3] Max, N.L. Vectorised procedural models for natural terrain: waves and island in the sunset. Proceedings of the ACM Computer Graphics, 1981,15:317～324.

[4] Peachey, D.R. Modeling waves and surf. Proceedings of the ACM Computer Graphics, 1986,19:65～74.

[5] Gerstner, F.J. Theorie der wellen. Ann. der Physik, 1809,32:412～440.

[6] Fournier, A., Reeves, W.T. A simple model of ocean waves. Proceedings of the ACM Computer Graphics, 1986,20:75～84.

[7] Watterberg, P.A., Mastin, G.A., Mareda, J.F. Fournier synthesis of ocean scenes. IEE Computer Graphics and Applications, 1987,7:16～23.

[8] Sims, K. Particle animation and rendering using data parallel computation. Proceedings of the ACM Computer Graphics, 1990,24:405～413.

[9] Mallinder, H. The modeling of large waterfalls using string texture. Visualization and Computer Animation, 1995,6:3～10.

[10] Xu, Y.Q., Su, C., Li, H., *et al*. Physically based simulation of water current and waves. Chinese Journal of Computers, 1996,19(supplement):153～160 (in Chinese).

[11] Burtnyk, N., Wein, M. Computer generated key-frame animation. Transactions on IECE of Japan, 1971,2:149～153.

[12] Reeves, W.T. Inbetweening for computer animation utilizing moving point constraints. Proceedings of the ACM Computer Graphics, 1981,15:263～269.

[13] Goldstein, E., Gostman, C. Polygon morphing using a multiresolution representation. In: Proceedings of the Computer Interface'95. 1995. 247～254.

[14] Sederberg, T.W., Gao, P., Wang, G., *et al*. 2D shape blending: an intrinsic solution to the vertex path problem. Proceedings of the ACM Computer Graphics, 1993,27:15～18.

[15] Yu, J. In-Betweening for computer animation using polar coordinate linear interpolation. CS Report Series, CSC 90/R23, University of Glasgow, 1990.

[16] Sederberg, T.W., Greenwood, E. A physical based approach to 2D shape blending. Proceedings of the ACM Computer Graphics, 1992,26:25～34.

[17] Ranjian, V., Fournier, A. Matching and interpolation of shapes using unions of circles. Computer Graphics Forum, 1996, 15:C-129～C-142.

[18] Yu, J., Patterson, W. A fire model for 2D computer animation. In: Proceedings of the Computer Animation and Simulation'96. Springer Computer Science EG, 1996,15:49～60.

[19] Yu, J., Xu, X., Peng, Q. Computer generation of cartoon smoke. Chinese Journal of Computers, 2000,23(9):987～990 (in Chinese).

[20] White, T. The Animator's Book. New York: Watson-Guptill, 1986.

附中文参考文献：

[10] 徐迎庆,苏成,李华,等.基于物理模型的流水及波浪模拟.计算机学报,1996,19(增刊):153～160.

[19] 于金辉,徐小刚,彭群生.计算机生成卡通烟雾动画.计算机学报,2000,23(9):987～990.

一个用于立体卡通动画的流水模型

于金辉， 徐晓刚， 万华根， 彭群生

(浙江大学 CAD & CG 国家重点实验室,浙江 杭州 310027)

摘要：提出了一个用于立体卡通动画的流水模型.介绍了如何把手工绘制的卡通流水画面分解成不同的部分,从中提取流水序列的静态信息与动态信息.基于这些信息,建立了一个具有等级结构的模型.通过调整少量的控制参数,模型可以自动生成与手工绘制卡通流水风格一致的动画序列.一个重要的工作是把卡通流水模型扩展到三维,并在CAVE立体投影系统中生成立体卡通流水动画.

关键词：流水模型;卡通动画;立体动画;计算机动画

中图法分类号：TP391 **文献标识码**：A