2010 18th Pacific Conference on Computer Graphics and Applications

# Procedural Modeling of Water Caustics and Foamy Water for Cartoon Animation

Jing Liao
*State key lab of CAD&CG*
*Zhejiang University*
*Hangzhou, China*
*Email: liaojing@cad.zju.edu.cn*

Jinhui Yu
*State key lab of CAD&CG*
*Zhejiang University*
*Hangzhou, China*
*Email: jhyu@cad.zju.edu.cn*

Long Jia
*School of IT & Mathematical Sciences*
*University of Ballarat*
*Ballarat, Australia*
*Email: l.jia@ballarat.edu.au*

***Abstract*—We propose a method for procedural modeling and animation of cartoon water effects such as water caustics and foamy water. In our method we emulate the visual abstraction of these cartoon effects by the use of Voronoi diagrams and the motion abstraction by designing relevant controlling mechanisms corresponding to each effect. Our system enables the creation of cartoon effects with minimal intervention from the animator. Through high-level initial specification, the effects are animated procedurally in the style of hand-drawn cartoons.**

***Keywords*-procedural modeling; water effects; non-photorealistic rendering; cartoon animation;**

## I. Introduction

Abstraction in art indicates a departure from reality in the depiction of imagery. Cartoon animation as an art of abstraction has two dimensions: motion abstraction and visual abstraction [1]. In the context of cartoon water animations, water motion abstraction depends on various conditions such as flow rates, interactions between barriers, wind conditions, etc. Figure 1 shows some hand-drawn water effects taken from commercial animations, including water caustics seen from underwater (Figure 1(left)) in *Ponyo on a Cliff*[1], and longshore currents (Figure 1(right)) in *Ichigo Mashimaro* [2].

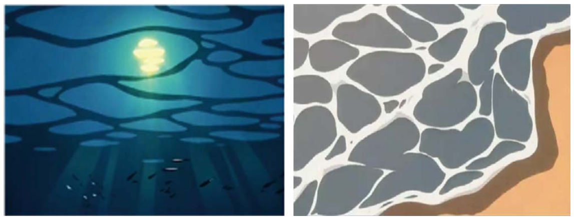

Figure 1. Hand-drawn water effects: water caustics(left)and longshore currents(right).

In this paper we propose an approach to model and render these cartoon water effects. Our method is based on observations that hand-drawn effects shown in Figure 1 closely resemble Voronoi diagrams (see Section II for detail) in structure, that is, the visual abstraction of these effects can be characterized by the Voronoi diagram. We therefore adopt the Voronoi diagram to emulate structures of these hand-drawn effects and design relevant rendering schemes and controlling mechanisms to animate them procedurally.

In the realistic water animation, many techniques have been proposed during the past three decades. For the comprehensive review please refer to [2] and [3]. In the cartoon water animation, only a few methods have been proposed during the past few years. A system capable of generating flames, drops, smoke puffs with hand-drawn style was proposed by Di Fiore et al. [4], in which hand-drawn visual elements are animated along the user defined path. Thornton [5] presented a particle based method to produce splash effects, by creating a modular rig composed of a series of nodes to describe the shape of the splash. A fluid based approach was proposed by Eden et al. [6] in which a 3D liquid surface simulated physically is rendered with bold lines and flat colors. Yu et al. [7] used a template based method to model the cartoon water effects including flowing water, ripples and water jet. Later on they [8] extended their work to handle the interaction between objects and cartoon water by introducing a grid based controlling mechanism.

## II. Voronoi diagrams and Delaunay triangulations

Given $n$ distinct points (sites) $P = \{p_1, p_2, ..., p_n\}$ in the plane, the Voronoi polygon (cell) of a point $p_i$, $VP(p_i)$, is defined to be the set of all point $q$ in the plane for which $p_i$ is among the closest point to $q$ in $P$. That is,

$$VP(p_i) = \{q : |p_i - q| \leq |p_j - q|, i \neq j\}. \qquad (1)$$

The union of the boundaries of the Voronoi polygon is called the Voronoi diagram of $P$, denoted by $VD(P)$. A Voronoi edge is the boundary between two Voronoi polygons and a Voronoi vertex is the intersection of three or more Voronoi edges. We adopt the simple algorithm by Lischinski [9] for the incremental construction of the Delaunay triangulation and the Voronoi diagram. The $VD(P)$ constructed above only models underlying spatial structures of cartoon effects shown in Figure 1. To animate these cartoon effects we need to incorporate appropriate rendering and controlling mechanisms into $VD(P)$. The main idea in our method is to construct time-varying textures of structure as $VD(P)$ for

[1] ©Studio Ghibli
[2] ©Daume

978-0-7695-4205-8/10 $26.00 © 2010 IEEE
DOI 10.1109/PacificGraphics.2010.8

varying effects shown in Figure 1 and map the texture onto the water surface.

## III. Water caustics seen from under water

Water caustics result from light rays reflecting or refracting from a water surface and hence focusing only in certain areas of the surface. The original hand-drawn animation in Figure 1(left) represents the effect of water caustics seen from under water. In the figure curvy lines are drawn as caustic textures on the water surface, the light scattering effect is depicted by the scattered light near the sun and a few scattering beams in the deep water.

Figure 2 illustrates our model for water caustics. First, we generate a Voronoi diagram on a texture image to emulate the structure of water caustic textures in Figure 1(left), and render Voronoi edges with curvy lines to obtain water caustic textures. The texture image is mapped onto the flat water surface laid horizontally in the scene (Figure 2(a)). Next, we project the scene composed of the water surface and the background image surface onto the view plane at a view point below the water surface (Figure 2(b)). On a separate image we generate the light scattering effect (Figure 2(c)) and then blend it with the former projected image on the view plane to get the final result (Figure 2(d)). The following sub-sections describe our model in detail.

### A. *Water caustic textures*

We first use sites with uniform distribution to construct a Vornonoi diagram VD(P) on a texture image. Next we render Voronoi edges with curvy lines. On each Voronoi edge, we take its two end points (Voronoi vertices) $V_1$ and $V_2$ as extremes and introduce two intermediate points $P_1$ and $P_2$ by the following formula:

$$\begin{aligned} P_1 &= V_1 + u_1(V_2 - V_1) + \Delta, \\ P_2 &= V_1 + u_2(V_2 - V_1) + \Delta, \end{aligned} \tag{2}$$

where $u_1$ and $u_2$ are positional parameters satisfying $0 < u_1 < u_2 < 1$, $\Delta$ is a small random perturbation. The four points $V_1$, $P_1$, $P_2$ and $V_2$ are interpolated with the cardinal spline (in the rest of the paper, we use spline for short) to draw a curvy line between $V_1$ and $V_2$ as shown on the left of Figure 3. Each curvy line is further rendered with varying width along its length, and the line width is determined by simple sinusoidal functions, as shown by blue curves on the right of Figure 3.

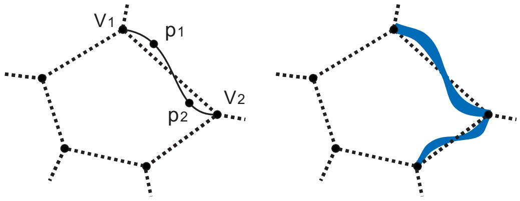

Figure 3. Rendering of Voronoi edges.

### B. *Dynamical control*

As for the dynamics of the water caustics, we tried different strategies to animate water caustic textures. At first, we added perturbations to Voronoi vertices every frame, the resultant animation of water caustic textures looks however jerky due to random perturbations added. In order to obtain smooth movements of water caustic textures, we alternatively generate small ellipses with initial Voronoi vertices of $VD(P)$ as reference centers and then animate Voronoi vertices along generated ellipse paths. On each frame of animation, Voronoi vertices are moved to new positions on their ellipse paths and curvy lines associated with Voronoi edges are generated afterward.

## IV. Longshore currents

Longshore currents are currents of water flowing parallel to the shore. The movements of the longshore currents consists of two phases: an *incoming wave phase* in which the wave crest begins to raise and reaches its maximum height, then the wave 'breaks' by decreasing its height as it moves on the beach, forming the foamy, bubbly surface; a *backwash phase* in which a seaward current that results from the receding swash on the beach face, after a wave breaks, joins the seaward movement of the wave trough toward the next incoming crest. As waves repeatedly hit the shore, water moves onto the beach and then retreats in a continuous cycle.

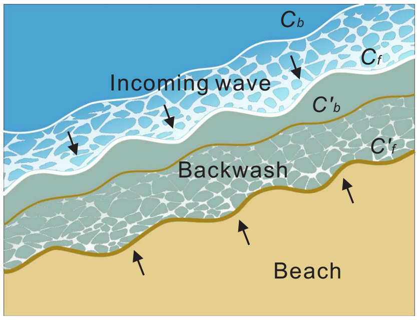

Figure 4. Longshore currents model.

Figure 4 illustrates our model for cartoon longshore currents. First, we model the region of the incoming wave with two parallel sinusoidal-like curves $C_f$ and $C_b$ with amplitude varying stochastically in time, where $C_f$ is the profile of the incoming wave near the beach and $C_b$ is the boundary of foamy water behind $C_f$. The distance between $C_f$ and $C_b$ controls the width of the incoming wave. Next, we map $C_f$ and $C_b$ onto the texture image and generate longshore current foamy water textures by rendering holes inside Voronoi cells on the incoming wave region, hole sizes are controlled progressively bigger from $C_f$ to $C_b$.

### A. *Foamy water textures*

Once $C_f$ and $C_b$ are obtained, we map the region between them onto the texture image and generate a Voronoi diagram

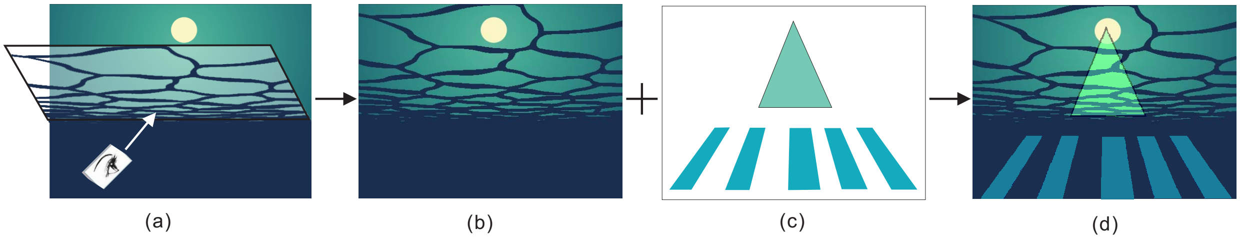

Figure 2. Water caustics model. (a) A scene composed of the water surface and the background image surface. (b) Projection of the scene on the view plane. (c) Image of light scattering effect. (d) Blending of the projected image and light scattering effect.

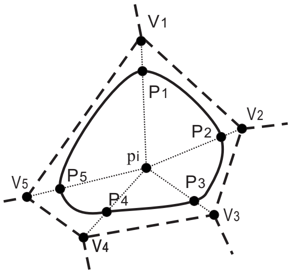

Figure 5. Hole rendering inside a voronoi cell $VP(p_i)$.

$VD(P)$ by putting some sites with uniform distribution inside the region. To render a hole inside the cell $VP(p_i)$, we pick up an intermediate point $P_k$ along the line formed by $p_i$ and vertices $V_k$ of $VP(p_i)$ with the following formula:

$$P_k = uV_k + (1-u)p_i \quad (k = 1, 2, ..., vn_i), \tag{3}$$

where $u \in [0, 1]$ is a parameter that controls the position of $P_k$ on the line, $vn_i$ is the vertex number associated with $VP(p_i)$ and can be uniquely determined after a $VP(p_i)$ is constructed [9]. A hole inside $VP(p_i)$ can be drawn by interpolating intermediate points $P_k$ in $VP(p_i)$ with the spline, as illustrated by the solid curve drawn inside a Voronoi cell with 5 vertices in Figure 5. The hole size is adjustable by varying the amplitude of $u$.

### B. Water surface deformation

To simulate height variations of incoming waves on the water mesh surface, we define the peak height on $C_f$ in the normalized height map by an empirical formula:

$$H_m(u) = \begin{cases} u/0.7 & u \leq 0.7, \\ (1.0-u)/0.3 & u > 0.7, \end{cases} \tag{4}$$

where $u \in [0, 1]$ is a spatial parameter that $u = 0$ corresponds to the position where the incoming wave is trigged, and $u = 1$ corresponds to the predefined limit position that $C_f$ can reach on the beach. The height on $C_b$ is defined as a constant. Next we can generate a hight map by use of linear interpolation between $C_f$ and $C_b$ over tht region sorrounded by $C_f$ and $C_b$. The height map is mapped onto the water surface to raise relevant vertices by displacement mapping. After the water surface is deformed, we map the longshore current texture onto raised water surface to produce a frame of animation (see Figure 7 in Section V).

### C. Dynamical control

During the animation, the incoming wave is trigged periodically at the user-specified position in the scene. As $C_f$ and $C_b$ are moved towards the beach, the width of the incoming wave and the hole size are increased progressively. When $C_f$ reaches the predefined limit position on the beach, the incoming wave stops moving and the backwash phase starts. During the backwash phase, the two curves turn to be $C'_f$ and $C'_b$ which are then moved backward (while the next incoming wave is still moved towards the beach). On the backwash region hole sizes are increased further, and remaining foamy water textures are blended with the beach texture image with time.

## V. Results

Models constructed above define cartoon water animations including water caustics and longshore currents. Through high level initial specification, the defined effects are animated procedurally. Our system allows further editing on some parameters to change the appearance of defined animations. Take the Voronoi cell for instance, its size can be altered by varying the distribution density of sites $p_i$, as shown by two variants of generated animation for the water caustic effects in Figure 6. To stage the scene properly we let some fish move under water in the animation.

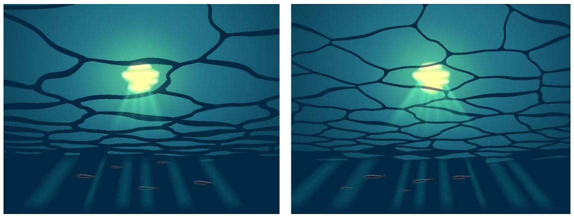

Figure 6. Two variants of generated animation for the water caustics seen from underwater.

To animate longshore current, users are required to specify the limit position that longshore current can reach on the beach in the scene, the speed of the longshore currents,

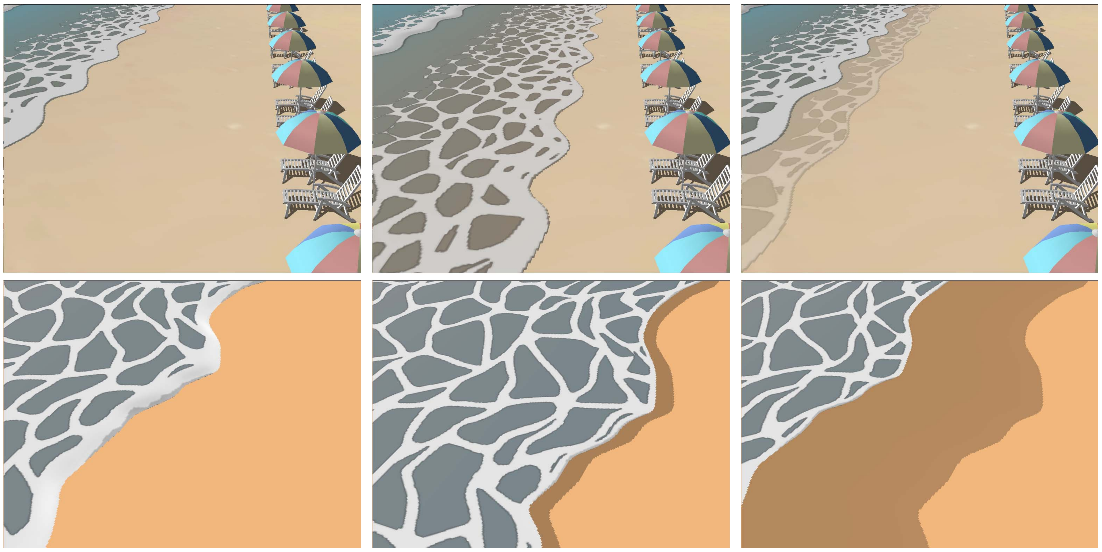

Figure 7. Two versions of generated longshore currents animation.

the distance between the two succeeding incoming waves. In Figure 7 we show two versions of longshore current animations. The upper 3 frames are taken from one cycle of the longshore current movements in the animation with a changing view point. The bottom 3 frames are taken from the animation with a fixed view position, simulating the hand-drawn series in Figure 1(right).

## VI. Conclusion

We have demonstrated a procedural system capable of generating cartoon effects ranging from water caustics to foamy water such as longshore currents. The main contributions of our approach are the emulation of visual abstraction of these cartoon effects by using Voronoi diagrams and emulation of the motion abstraction by designing relevant controlling mechanisms for each effect. Our models enable the creation of cartoon effects with minimal intervention from the animator. In our approach the effects are animated procedurally in the style of cartoons. Thus, the strength of our approach to animation lies in the fact that this scheme can reduce animator's workload significantly - it turns the role of the animator from hand-drawing slave to the high level model controller.

## Acknowledgment

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

## References

[1] C. Bregler, L. Loeb, E. Chuang, H. Deshpande, ”Turning to the masters: Motion capturing cartoons,” ACM Transactions on Graphics, vol. 21(3), July 2002, pp. 399-407, doi:10.1145/566654.566595.

[2] N. Adabala, S. Manohar, ”Techniques for realistic visualization of fluids: a survey,” Computer Graphics Forum, vol. 21, May 2003, pp. 65-82, doi:10.1111/1467-8659.00566.

[3] A. Iglesias, ”Computer graphics for water modeling and rendering: a survey,” Future Generation Computer Systems, vol. 20(8), Nov. 2004, pp. 1355-1374, doi:10.1016/j.future.2004.05.026.

[4] F. Di Fiore, J. Claes, F. V. Reeth, ”A framework for user control on stylised animation of gaseous phenomena,” Proc. Computer Animation and Social Agents (CASA 04), John Wiley & Sons, 2004, pp.171-178.

[5] J. D. Thornton, ”Directable simulation of stylized water splash effects in 3d space,” Proc. International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2006 Sketches), Boston, USA. ACM Press, 2006, pp. 94, doi:10.1145/1179849.1179967.

[6] A. M. Eden, A. W. Bargteil, T. G. Goktekin, S. B. Eisinger, J. F. O'brien, ”A method for cartoon-style rendering of liquid animations,” Proc. Graphics Interface (GI07), ACM Press, 2007, pp. 51-55, doi:10.1145/1268517.1268528.

[7] J.-H. Yu, X. Jiang, C. Yao, H.-Y. Chen, ”Real-time cartoon water animation,” Computer Animation and Virtual Worlds, vol. 18(4-5), Sep. 2007, pp. 405-414, doi:10.1002/cav.v18:4/5.

[8] J.-H. Yu, J. Liao, J. Patterson, ”Modeling the interaction between objects and cartoon water,” Computer Animation and Virtual Worlds, vol. Aug. 2008, vo. 19(3-4), pp.375-385, doi:10.1002/cav.v19:3/4.

[9] D. Lischinski, ”Incremental delaunay triangulation,” in Academic Press Graphics Gems Series (Graphics gems IV), 1994, pp.47-59.