

# Water Droplet Morphing Combining Rigid Transformation\*

Lanfen Lin, Shenghui Liao, RuoFeng Tong, and JinXiang Dong

State Key Laboratory of CAD and CG,  
Department of Computer Science and Engineering,  
Zhejiang University, China  
liaoashenhui@zju.edu.cn

**Abstract.** This paper presents a plausible method for large water droplet morphing, taken into account the rigid transformation. The droplet on a plane was represented by the contact area, which was described via “a variational implicit function” in 2D, together with a profile curve. Then we made use of distance fields generated from the boundary of contact area to drive the morphing. The deforming direction and speed were well controlled to give a very smooth and stable deformation. The morphing procedure was combined with rigid transformation synchronously, which yield more natural effect. Finally, the ray-tracing method was employed for rendering the realistic scene.

## 1 Introduction

Several different methods addressing water modeling and animation have been developed since the 1980's. Most of them concerned the motion of water in forms of waves and other connected fluids, for example, waves approaching and braking on a beach [1]. Realistic and practical animation of liquids [2] has also been made. Only a few methods proposed during the 1990's addressed the problems of the water droplets.

Kaneda et al [3] developed methods for realistic animation of water droplets on a glass plate, on curved surfaces, and meandering down a transparent surface. The main purpose is to generate a realistic animation, taken into account gravity of water droplets, inter-facial tensions, and so on. Fournier et al [4] presented a model that focuses on the simulation of large liquid droplets as they travel down a surface. The aim is to simulate the visual contour and shape of water droplets when affected by the underlying surface and other force fields. Malin [5] gave a method for animation of water droplets flowing on structured surfaces, the droplets in this method were affected by underlying bump mapped surface.

---

\* Project supported by National Grand Fundamental Research 973 (No.2002CB312106) of China.

\* Project supported by Natural Science Foundation (No.M603129) of Zhejiang Province, China.

All these methods are physical based. It is quite difficult to simulate the flow of water droplets for the purpose of high-precision engineering, due to the diversiform shape representation of the droplet and the complicated flow process. While, our method is not to make a simulation physically correct, but to make an effective and physically plausible droplets morphing on a plane.

## 2 The Main Idea of Our Approach

### 2.1 Droplet Representation

First, the droplet on a plane is represented by contact area between the droplet and the plane, together with a profile curve.

To describe the contact area between the droplet and the plane, such as Fig.1 (b), we employ “a variational implicit function” in 2D, proposed in Turk and O’Brien’s paper [6], which can model arbitrary 2D shape of general topology.

Yu et al [7] discussed the droplets’ shape on a simple plane in the gravitational field. We take advantage of their result and use a profile curve to describe the height of droplet within the contact area, as Fig.1 (a) illustrates (The x coordinate of profile curve means the distance from a point to the boundary of contact area, and y coordinate gives the height of droplet at this point).

When the contact area as well as the profile curve is ready, the ray-tracing method is employed for rendering the scene, to produce a realistic effect of the water droplet. Such as Fig.2 shows.

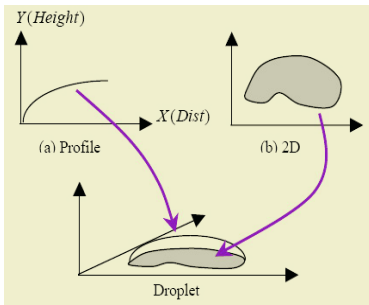


Fig. 1. Droplet representation

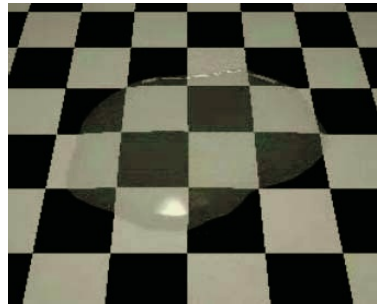


Fig. 2. Rendering scene

### 2.2 Initial Method

Our morphing between two (or more) droplets consists of blending between two profile curves and morphing between two contact areas. Blending between two profile curves is simple, thus we will concentrate on morphing between two contact areas, that is, a kind of 2D volume morphing.

There are several existing approaches for volume morphing. Pasko simply interpolated the values of corresponding nodes [8]. This may cause unnecessary

distortion and change in topology. Other techniques are based on a decomposition of the discrete values. Hughes considered Fournier decomposition [9], and T. He decomposed the functions with a wavelet transform [10]. Leros proposed a feature-based morphing [11]. Cohen presented a method that the interpolation of distance field was guided by a warp function controlled by anchor points [12]. But there is more or less artificial interference in these methods.

Our initial idea is using the intensity fields of implicit functions to produce potential fields, with which we can change the source intensity field to the target. This approach is somewhat like Whitaker’s Level-Set Model [13].

Assume  $F_1(X) = \delta$  is the source intensity field and  $F_2(X) = \delta$  is the target, where  $X = (x, y)$ . The morphing can be generated by the movement of  $X$ , as illustrated in Fig.3.

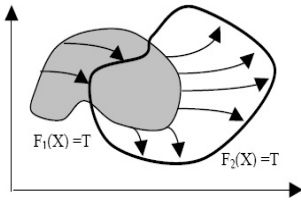


Fig. 3. Potential Field

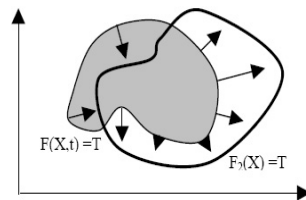


Fig. 4. Movement direction

The problems are:

1. How does  $X$  move (The direction and speed)?
2. How does the movement affect the intensity field function  $F$ ?

Obviously, the movement of the  $X$  can be described by regarding  $X$  as a function of time  $t$ . Thus the whole morphing procedure can be represented by  $F(X(t),t) = \delta, t \in (0, 1)$ , and  $F(X(0),0) = F_1(X), F(X(1),1) = F_2(X)$ . Because  $F$  equals to  $\delta$  over time, the time derivation should be zero:

$$\frac{dF}{dt} = \frac{\partial F(X(t), t)}{\partial t} + \nabla F(X(t), t) \bullet \frac{\partial X(t)}{\partial t} = 0 \text{ where } \nabla F(X, t) = \left( \frac{\partial F}{\partial x}, \frac{\partial F}{\partial y} \right) \tag{1}$$

Thus,

$$\frac{\partial F(X(t), t)}{\partial t} = -\nabla F(X(t), t) \bullet \frac{\partial X(t)}{\partial t} \tag{2}$$

Eq.2 describes how the movement of  $X$  (denoted by  $\frac{\partial X(t)}{\partial t}$ ) affects the intensity function  $F$ . Now the only problem is how does  $X$  move.

First of all, when  $t$  changes from  $t_0$  to  $t_n$ ,  $\frac{\partial X(t)}{\partial t}$  must changes  $F(X(t),t)$  from  $F_1(X)$  to  $F_2(X)$ , otherwise it will not be a morphing.

Rewrite Eq.2 in a discrete way:

$$\frac{F(X, t_{k+1}) - F(X, t_k)}{\Delta t} = -\nabla F(X, t_k) \bullet \frac{\partial X}{\partial t} \Big|_{t_k} \tag{3}$$

To make  $F(X(t),t)$  change from  $F_1(X)$ , just set  $F(X, t_0) = F_1(X)$ . The problem is how to ensure the deformation end at  $F_2(X)$ . Note that the intensity field at next time is controlled by Eq.3, whose parameter can be used to control the deformation is  $\frac{\partial X(t)}{\partial t}$ . One choice of  $\frac{\partial X(t)}{\partial t}$  that may satisfy this demand is

$$\frac{\partial X(t)}{\partial t} = - (F_2(X) - F(X, t)) \frac{\nabla F(X, t)}{\|\nabla F(X, t)\|} \tag{4}$$

The geometric explanation of Eq.4 is the movement speed of  $X$  is  $|F_2(X) - F(X, t)|$ . And the direction is along the current normal  $(-\frac{\nabla F(X,t)}{\|\nabla F(X,t)\|})$ , if  $X$  is inside the target object; else, the direction is the reverse of the normal, as illustrated in Fig.4. Substitute Eq.4 into Eq.3, get the iteration equation:

$$F(X, t_{k+1}) = F(X, t_k) + \|\nabla F(X, t_k)\| (F_2(X) - F(X, t_k)) \Delta t \tag{5}$$

where  $\nabla F \approx (\frac{F(x_{i+1}, y_j, t_k) - F(x_{i-1}, y_j, t_k)}{2h}, \frac{F(x_i, y_{j+1}, t_k) - F(x_i, y_{j-1}, t_k)}{2h})$ .

The ending condition of the iteration of Eq.5 is  $\max_{X \in \Omega} |F_2(X) - F(X, t)| < \varepsilon$ , where  $\Omega$  is the boundary of  $F(X,t)$ , and  $\varepsilon$  is a minimal positive constant.

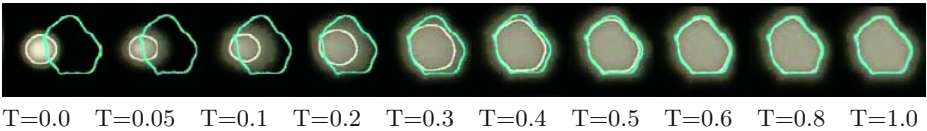


Fig. 5. Morphing using Eq.5

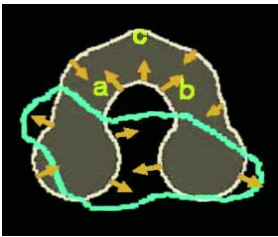


Fig. 6. Direction and distance problem

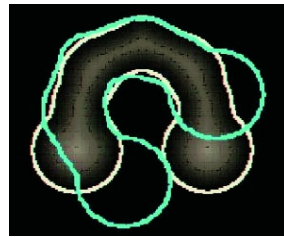


Fig. 7. Rigid deformation problem

By iterating Eq.5,  $F(X(t),t)$  changes gradually from  $F_1(X)$  into  $F_2(X)$ . Unfortunately, there are several unpleasant phenomena.

First problem: as Fig.5 shows, the morphing shape changes quickly at first, and then becomes very slow. Eq.4 gives cause for it, as the pixels move at  $|F_2(X) - F(X, t_k)|$ . When  $F(X, t_k)$  is approaching  $F_2(X)$ , the speed will get slower and slower. Problem 2: every pixel outside the target shrinks, as in Fig.6. Because according to Eq.4, every pixel's movement direction is reverse of the normal when out of the target. Actually, we expect the pixels between **a** and **b** expand instead of shrink. Problem 3: at the place far enough from the boundary

of target (such as  $\mathbf{c}$  in Fig.6), the intensity value may be all zero, this prevents us from controlling the movement. Problem 4: when the shape of source and target are the same, such as Fig.7, we expect the morphing changes by rigid deformation composed of pure rotation and translation. If they are different, we want least distortion of the in-between objects. Eq.5 can not gratify this demand.

To get a natural and pleasant morphing effect, these problems should be addressed. Section 3 gives a solution to the first 3 problems; and problem 4 is handled in section 4.

### 3 Distance Field Morphing

First, we generate the distance field from the boundary of contact area. Then, use the distance field instead of intensity field to control the deformation.

#### 3.1 Distance Field Transformation

The distance inside the area of object is positive, outside the area is negative, and zero on the boundary. Assume  $\text{dist}(X)$  is the Euclidean distance from point  $X$  to the boundary of contact area, define distance field as:

$$D(X) = \begin{cases} +\text{dist}(X) & \text{if } X \text{ is inside object;} \\ 0 & \text{if } X \text{ is on the boundary;} \\ -\text{dist}(X) & \text{if } X \text{ is outside object;} \end{cases} \tag{6}$$

Denote the source distance field as  $S(X)$ , the target field as  $T(X)$ , and the current deforming distance field as  $D(X(t),t)$ . Using the same general spirit as the previous study, we rewrite Eq.4 and the iteration Eq.5 as:

$$\frac{\partial X(t)}{\partial t} = -T(X) \frac{\nabla D(X(t),t)}{\|\nabla D(X(t),t)\|} \tag{7}$$

$$D(X, t_{k+1}) = D(X, t_k) + T(X) \bullet \|\nabla D(X, t_k)\| \tag{8}$$

Now, problem 3 is handled naturally, as  $T(X)$  still works at the place far enough from the target object.

#### 3.2 Control Morphing Speed and Direction

In the first problem, we want a stable morphing, that is, the maximum speed of the pixels at every time step should be identical. Assume  $N$  is the number of total morphing step. We give the stable iteration Equation:

$$D(X, t_{k+1}) = D(X, t_k) + \frac{T(X)}{N - k} \bullet \|\nabla D(X, t_k)\| \tag{9}$$

Morphing in Fig.8 is generated by applying Eq.9. As expected, the morphing speed is very stable. What's more, the number of total morphing step  $N$ , that is, the total morphing frames number, is now under control.

Now let's consider the morphing direction in problem 2. Assume  $\mathbf{N}_T(X)$  is the unit normal vector of target distance field, and  $\mathbf{N}_D(X)$  is that of current distance field. Note that in the area where  $\mathbf{a}$  and  $\mathbf{b}$  belong to (in Fig.6), the angle between  $\mathbf{N}_T(X)$  and  $\mathbf{N}_D(X)$  is an obtuse angle; while in the area where  $\mathbf{c}$  belongs to, it is an acute angle. So, the dot production  $\mathbf{N}_T(X) \bullet \mathbf{N}_D(X)$  can be employed to decide the morphing direction of the outside part, where  $T(X) < 0$ .

Thus we get the last iteration Equation:

$$D(X, t_{k+1}) = \begin{cases} D(X, t_k) + \frac{T(X)}{N-k} \bullet \|\nabla D(X, t_k)\| \bullet (\mathbf{N}_T(X) \bullet \mathbf{N}_D(X)), & T(X) < 0 \\ D(X, t_k) + \frac{T(X)}{N-k} \bullet \|\nabla D(X, t_k)\|, & T(X) > 0 \end{cases} \tag{10}$$

Applying Eq.10, Fig.9 gives a smooth and natural morphing. And the pixels between  $\mathbf{a}$  and  $\mathbf{b}$  all expend to the target rather than shrink.

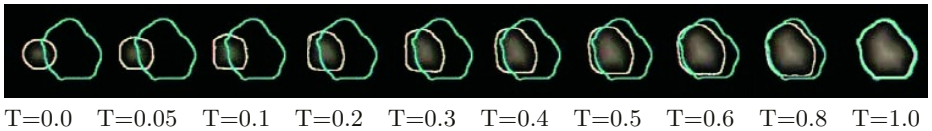


Fig. 8. Morphing using Eq.9

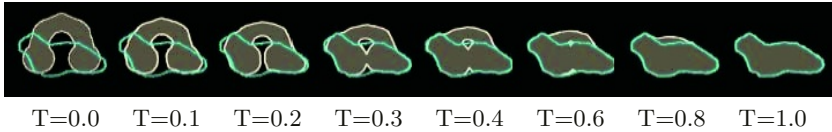


Fig. 9. Morphing using Eq.10

### 4 Combining Rigid Transformation

As in problem 4, in some cases, to get better morphing effect, we must transform (warp) source object  $O_s$ , to another object  $W(O_s)$ , which approximates the target object  $O_t$ , as well as possible. The transformation  $W$  is composed of a pure translation  $C$  and a rotation  $R$ .

The translation is easily obtained by comparing the center of the source and that of the target. The problem is how to define the rotation to make  $W(O_s)$  approximate  $O_t$ .

First, in the space  $O_s$ , emit a line in each direction of  $\theta_i = i\Delta\theta$  ( $i=0, \dots, N-1$ ) from the center, calculate the distance to the farthest boundary. Store the distances in array  $D_s[i]$ . Define  $D_j[i] = \begin{cases} D_s[i + j], i + j < N \\ D_s[i + j - N], i + j \geq N \end{cases}$  ( $i, j=0, \dots, N-1$ ). Do the same thing to  $O_t$  to generate  $D_t[i]$ . Then calculate the correlation between  $D_j[i]$  and  $D_t[i]$ , and select the  $j$  which maximizes the correlation, that is, the  $j$  which minimizes  $\sum_{i=0}^{N-1} (D_t[i] - D_j[i])^2$ , to get the rotation angle  $\theta = j\Delta\theta$ .

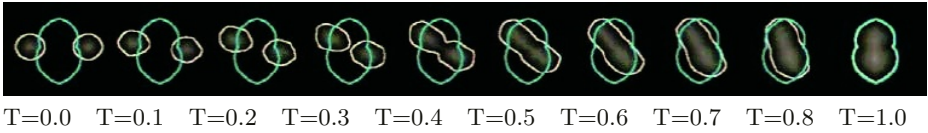


Fig. 10. Combine rigid transformation

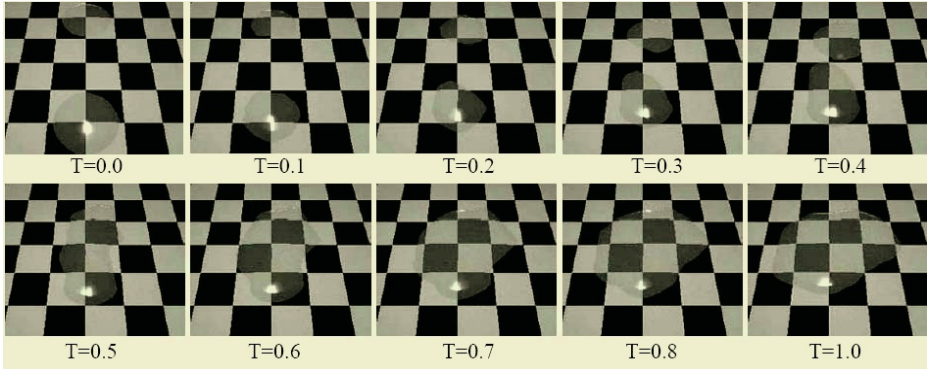


Fig. 11. Complete morphing Procedure

So, the rigid transformation used to rotate and translate the source object to match the coarse features of the target object is  $W(X) = R_\theta(X) + C$ , ( $\theta = j\Delta\theta$ ).

One thing must be pointed out is: when combine the morphing with the rigid transformation, the exact number of frames is pre-requisite. This is because the morphing speed must keep pace with the rigid transformation. If it is slower, it has not arrived the target when the rotation and translation have ended, and, contrariwise. Recall Eq.9 (also Eq.10), the number of total morphing step  $N$  is under control. So, they can be combined synchronously precisely.

Now, we can organize complete distance field morphing procedure as follows:

Step 1: Generate the transformation  $W$ , exerting on the source object  $O_s$ , which approximates the target object  $O_t$ , denoted as  $W(O_s)$ . The transformation  $W$  is composed of a pure translation  $C$  and a rotation  $R_\theta$ .

Step 2: Generate a new distance field  $W(S)$  according to the transformation  $W$ , by assigning the value of point  $X$  in field  $S$  to the point  $W(X)$  in field  $W(S)$ , where  $W(X) = R_\theta(X) + C$ .

Step 3: Generate a series of intermediate fields  $M(W(S), T, t)$  by morphing  $W(S)$  to  $T$ , using Eq.10.

Step 4: Transform every field  $M(W(S), T, t)$  to field  $W_t^{-1}M(W(S), T, t)$ , by assigning the value at point  $X$  in field  $M(W(S), T, t)$  to the point  $W_t^{-1}(X)$  in field  $W_t^{-1}M(W(S), T, t)$ , where  $W_t^{-1}(X) = R_{-(1-t)\theta}(X) - (1-t)C$ ,  $t \in [0, 1]$ .

The series of  $W_t^{-1}M(W(S), T, t)$  is the in-between fields from  $S$  to  $T$ .

Fig.10 shows a morphing combining rigid transformation. As expected, the morphing procedure combines rigid transformation synchronously smoothly.

Then, we can calculate the contact area between the droplet and the plane, as well as take advantage of the profile curve directly from these intermediate

distance fields, which are used to represent the water droplet. Subsequently, the scene is rendered by ray-tracing method. Fig.11 is our final example, which shows a complete water droplets morphing procedure. We can see that the plausible morphing procedure gives a very smooth and natural effect.

## 5 Conclusion

We have presented a metamorphosis method between two (or more) large water droplets with arbitrary shape and topology, without any artificial interaction. The morphing procedure is not physical based, but driven by distance field. Various instances are considered, thus the morphing speed and direction are well controlled. Synchronous rigid transformation is taken into account to give a more natural effect.

## References

1. Peachey D.: Modeling Waves and Surf. *Computer Graphics*, 20(4), pp. 65-74, 1986.
2. Foster N, Fedkiw R.: Practical Animation of Liquids. Proceedings of the 28th annual conference on Computer graphics and interactive techniques, 2001.
3. Kaneda K, Shinya I, Yamashita H.: Animation of Water Droplets Moving Down a Surface. *The Journal of Visualization and Computer Animation* 10, 1999.
4. Fournier P, Habibi A, Poulin P.: Simulating the flow of liquid droplets. Proceedings of Graphics Interface 98 pp.133-42, 1998.
5. Malin Jonsson, Anders Hast.: Animation of Water Droplet Flow on Structured Surfaces. Conference: Sigrad 2002, Norrköping, pp. 17-22, 2002
6. Turk G., O'Brien J.: Shape transformation using variational implicit functions. Proceedings of ACM SIGGRAPH 99, pages 335-342, 1999.
7. Yu Y-J, Jung H-Y, Cho H-G.: A new water droplet model using metaball in the gravitational field. *Computer & Graphics* 23, pp. 213-222, 1999.
8. Alexander Pasko, Vladimir Savchenko.: Constructing functionally defined surfaces. *Implicit Surfaces'95*, Grenoble, France, pp. 97-106, April, 1995.
9. J. F. Hughes.: Scheduled Fourier volume morphing. *SIGGRAPH Computer Graphics*, 26(2), pp. 43-45, 1992.
10. T. He, S. Wang and A. Kaufman.: Wavelet-based volume morphing. In Proceedings of IEEE Visualization '94, Washington, D.C., pp. 85-92, 1994.
11. A. Leros, C. D. Garfinkle and M. Levoy.: Feature-based volume metamorphosis. Proc. SIGGRAPH 95, Los Angeles, California, pp. 449-456, August 6-11, 1995.
12. D. Cohen-Or, D. Levin, and A. Solomivici.: Three-dimensional distance field metamorphosis. *ACM Transactions on Graphics*, 17(2), pp. 116-141, 1998.
13. D. Breen, R. Whitaker.: A level set approach for the metamorphosis of solid models. *IEEE Transactions on Visualization and Computer Graphics*, 7(2):173-192, 2001.