REALTIME CG ANIMATION OF REALISTIC FISHES USING NURBS, INVERSE KINEMATICS AND A CO-OPERATIVE MOTION MODEL

Takahiro Kawamura, Hiroshi Dohi and Mitsuru Ishizuka

Faculty of Engineering, University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo 113, Japan
Tel: +81-3-3812-2111 ext.6755, Fax: +81-3-5802-2990
E-mail: kawamura@sak.iis.u-tokyo.ac.jp

Abstract

In this paper, we present a realistic CG (Computer Graphics) animation method to move fishes. These have highly realistic appearance and movement. NURBS(Non-Uniform Rational B-Spline) is employed for both modeling and deformation of CG fishes. This makes the surface data of the fishes very simple to obtain. We can create the very smooth shape of the fishes(one side) by using about 8×4 control points of NURBS. Realistic deformation of the fishes is also achieved by forming the control points into several groups and moving them. Inverse kinematics are also incorporated into this deformation. The fishes can swim naturally using a combination of 3 deformation patterns and 3 locomotion patterns. Autonomy of the fishes is realized by a mental model employing random numbers. Both schooling behavior and collision avoidance are obtained by an algorithm employing the detection vector and direction vector. Since the computing costs for this method are not high, we can run the program with general-purpose graphics workstations in realtime.

1 Introduction

In this paper, we present a realistic CG animation method to move fishes. These have highly realistic appearance and movement. Prior to our work, X.Tu and D.Terzopoulos made a similar artificial fish[5]. They tried to make the fish's movement model as close to actual physics as possible. Unlike their approach, we do not necessarily aim at realizing the movement model in accordance with actual physics. Rather, using an inexpensive simple method, we try to realize the movement model to be viewed as if it is natural and realistic in realtime.

T.W.Sederberg and S.R.Rarry presented FFD (Free Form Deformation) to deform 3-dimensional objects smoothly[4]. But this method can not be applied to realtime CG at present hardware performance. We employ NURBS (Non-Uniform Rational B-Spline) for modeling and deformation of various CG fishes because of its wide use.

NURBS allows to make the modeling data of the fishes very simple. We can create very smooth shape of the fishes (one side) by using about 8×4 control points of NURBS, except the long shaped fishes like sea snake. We don't consider uneven fishes like balloonfish.

We have made a fish movement model based on the theory that a common way of the movement of aquatic animals is like a progressive wave which begins at their heads, raises its amplitude and reduces its wavelength as it goes to their tails. In this model, the animals twist their bodies so that the wave goes through the bodies. We consider this model can be applied to spindle shaped fishes like cod, side-compressed shaped fishes like butterfly fish, and long shaped fishes like sea snake. But it cannot be applied directly to fishes with back-and-stomach-compressed shaps like flatfish, and hard fishes like boxfish.

It is thought that fish form a schools by their visions and lateral line organs. Thanks to this organ, the fish feels the low frequency oscillations generated by the other fish and the echo of its own in order to maintain a neighboring relationship. In this research, we simulate a school considering this fact.

The method of this paper allows to create various CG fishes having high reality without the help of animators in a short time, because they can be formed by modifying one basic data structure and mapping a photo texture to the structural data of the fish.

In this research, we use Graphics Library (GL) provided by Silicon Graphics, Inc. as a 3-dimensional CG tool including NURBS and texture mapping.

2 Modeling of a fish

The meshes of NURBS control points are arranged to the shape of the intended fish as illustrated in Fig.1. That is, one side of the fish is formed by a smooth NURBS surface defined by $M \times N (M \geq 4, N \geq 4)$ control points which fit exactly the photo texture as shown in Fig.2. At this time, the thickness of the fish (the width of Z direction) is set to a general value. The other side of the fish is also formed by the same method; as a result, we have a symmetrical fish model. The number of the multiple knots which are the end points of the knot vectors should be the same as $(s+1)$ where s is the order the B-spline basis function, so that both sides of the fish model are connected together without the gap. For example, when the order of NURBS is 3, the knot vectors in u and v directions of 8×4 control-points meshes are determined as below.

$$\vec{u} = [0.0, 0.0, 0.0, 0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.0, 1.0, 1.0] \quad (1)$$

$$\vec{v} = [0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0] \quad (2)$$

The arrangement of control points should be determined carefully. It is desirable that the spaces between control points should be arranged as equally as possible and they should be the characteristic points that define the shape of the fish, so that the smooth surface may be formed and the computation cost may be reduced.

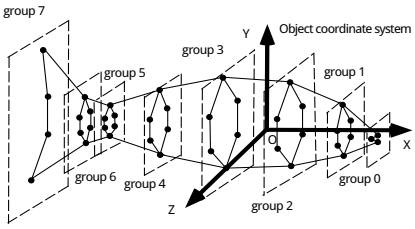

Fig.1: The way of setting control points(Lutjanus kasumira)

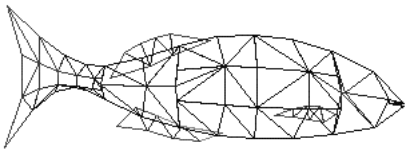

Fig.2: Surface model of Lutjanus kasumira(one side)

In order to deform the fish model naturally, we arrange the control points into several ring-shaped groups that cut the fish in round slices. Most fishes in the real world have the backbone that consists of many spines, and its muscles are segmented in the round slices that correspond to each spline. Accordingly, the movement of a fish's surface is also thought to be segmented in round slices.

The control points define the shape of the fish as well as the deformation. This rational way reduces the computation cost.

"Pectoral fin" and other attachments are formed as other objects, and combined to the NURBS body. "Pectoral fin" can be controlled, if its joint-angle is changed corresponding to the movement of the fish. We can control the opening and closing of the mouth by moving the

control points that define the shape of it.

In Fig.2, the sizes of the triangular polygons are not even. That is because the division to a triangular polygon is optimized according to the curvature of NURBS curve. We can specify the maximum length (by the unit of pixel) of the line segment edge that is formed by dividing the polygon or curve to the triangle. The maximum length in Fig.2 is 50.0. These functions are provided by Graphics Library.

3 Deformation of a fish

Both deformation and locomotion are the important factors to realize the realistic movement of a fish. In this section, we describe how the deformation is realized.

In order to realize the progressive wave that goes through the body of a fish, we have to move the groups of the control points so that the backbone of the fish get on the curve represented by a mathematical function. Accordingly, we have to search for the length of the curve so as not to change the spaces between the groups of the control points . But it is difficult in general to find a primitive function in order to solve an integral function, and a numerical value analysis does not satisfy the need of real time processing. It is also not easy to search for the angles between the joints by the way of putting a skeleton model of multi-joint structure on the curve. Considering the above facts , we deform a fish by rotating the groups of the control points around Y axis of the object coordinate system, and realize an approximate progressive wave that goes through the body of the fish.

Because NURBS defines a curve by the combination of sectional polynomials, we can change the limited part of its shape. We employ a method of multistage single-joint inverse kinematics so as not to stretch the length of a fish longer.

In this research, deformation of a fish is composed of three motion patterns. The first is "natural twist" that produces the propelling power for a fish to swim straight. The second is "left twist" that represents the twisting motion when a fish turns left. The third is "right twist" that also represents the twisting motion when a fish turns right.

3.1 Multistage single-joint inverse kinematics

Concerning the triangle OBC of Fig.3, we have the following two equations.

$$\theta'_3 = \arcsin\left\{\frac{r'_2}{r_3}\sin(\theta_3-\theta_2)\right\} \quad (3)$$

$$r'_3 = r'_2\cos(\theta_3-\theta_2)+r_3\cos\theta'_3 \quad (4)$$

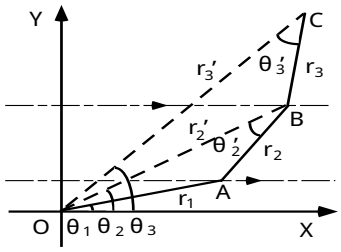

Fig.3: Multisatage one-joint inverse kinematics

Generally, as i expresses a positive integer, we have the following two recursive formulas.

$$\theta'_1 = 0,\ r'_1 = r_1$$

$$\theta'_{i+1} = \arcsin\left\{\frac{r'_i}{r_{i+1}}\sin(\theta_{i+1}-\theta_i)\right\} \quad (5)$$

$$r'_{i+1} = r'_i\cos(\theta_{i+1}-\theta_i)+r_{i+1}\cos\theta'_{i+1} \quad (6)$$

If we express the coordinate of the joint between the ith arm and the $(i+1)$th arm as (x_i, y_i), then (x_{i+1}, y_{i+1}) is calculated as below.

$$x_{i+1} = r_{i+1}\cos(\theta_{i+1}+\theta'_{i+1})+x_i \quad (7)$$

$$y_{i+1} = r_{i+1}\sin(\theta_{i+1}+\theta'_{i+1})+y_i \quad (8)$$

Thus, the coordinates of the points that accompany the $(i+1)$arm are calculated by rotating them around (x_i, y_i) at the angle of $(\theta_{i+1}+\theta'_{i+1})$. These relationships do not depend on whether each variable is positive or not. But θ_i is restricted as below, so that it may be within reach of the arms.

$$|\theta_{i+1}-\theta_i| \leq \arctan(\frac{r_{i+1}}{r_i}) \quad (9)$$

Though this principle can be expanded to 3-dimension, it is used in 2-dimension in this research.

3.2 Natural twist

"Natural twist" is realized by rotating each group of the control points around Y axis. For a mathematical function that realize the twist, we use here the following function.

$$\theta_i = \frac{\pi}{8}a\cdot\sin(\omega t - d_i), \quad (10)$$

where θ_i is the rotation angle of the ith group of the control points, a is the parameter that controls the amplitude of the twist, ω is the angular velocity that controls the speed of the twist, t is time, and d_i is the phase difference from the 0th group of the control points.

In case of Lutjanus kasumira(Fig.4), determined values are shown in Table.1, where x_i is the initial X coordinate of the ith group of the control points. Here, ω is $\pi/4$, and t is expressed as $\pi/4\times n$ where n is the frame number(in case that a fish is rendered at the rate of 4 frames/sec).

Fig.4: Animation of natural twist

Table.1: Data example of natural twist

i	x_i	r_i	d_i
0	0.2466	0.0269	$\pi/8$
1	0.2197	0.1345	$\pi/8$
2	0.0852	0.0852	$\pi/8\times 2$
3	-0.0763	-0.0763	$\pi/8\times 3$
4	-0.2376	-0.1613	$\pi/8\times 4$
5	-0.3990	-0.1614	$\pi/8\times 5$
6	-0.5066	-0.1076	$\pi/8\times 6$
7	-0.6142	-0.1076	$\pi/8\times 7$

3.3 Left twist and Right twist

The principle of left twist is the same as that of the natural twist. In this research, we use the following function that realizes left twist. Right twist is realized symmetrically.

$$\text{In case of } x_i > 0, \quad \theta_i = \frac{\pi}{6}a\cdot|sin(\omega t - d_i)| \quad (11)$$

$$\text{In case of } x_i < 0, \quad \theta_i = -\frac{\pi}{6}a\cdot|sin(\omega t - d_i)| \quad (12)$$

Table.2: Data example of left twist

i	x_i	r_i	d_i
0	0.2466	0.0269	0
1	0.2197	0.1345	0
2	0.0852	0.0852	$\pi/8$
3	-0.0763	-0.0763	$\pi/8$
4	-0.2376	-0.1613	$\pi/8\times 2$
5	-0.3990	-0.1614	$\pi/8\times 3$
6	-0.5066	-0.1076	$\pi/8\times 4$
7	-0.6142	-0.1076	$\pi/8\times 5$

The twist accompanies a direction change in the way of twisting the head with the tail fixed and then releasing the tail with the head fixed. This is realized by rotating the object coordinate system around Y axis using θ_0 and θ_7. In case of Lutjanus kasumira, determined values are shown in Table.2. ω is $\pi/2$ in this case.

4 Locomotion of a fish

Locomotion is composed of 3 motion patterns, i.e, straight move, left turn and right turn. It is thought that a fish suffers the friction and the pressure that is in proportion to its speed, when

the speed is not so high. Accordingly, we use the following equation as a locomotion model.

$$m\frac{d^2\vec{r}}{dt^2} + k\frac{d\vec{r}}{dt} = \vec{F}, \tag{13}$$

where $\vec{r}$ is the position vector of the fish in the global coordinate system, m is the mass of the fish, k is the resistance coefficient that the fish suffers from water, and $\vec{F}$ is the sum of the propelling force of the fish, the lifting and sinking force of it and the external force such as a flow. The velocity expresses the movement of the fish to the next frame. The propelling force is in proportion to the amplitude a and the angular velocity ω in Eqn.10, and its direction agrees with that of the fish. If the velocity of the previous frame is expressed as $\vec{v}$, the equation is transformed to,

$$\frac{d\vec{r}}{dt} = \frac{\vec{F}}{k} + (\vec{v} - \frac{\vec{F}}{k})exp(-\frac{k}{m}t). \tag{14}$$

The movement of each frame is determined by this calculation.

5 Algorithm for schools

Reynolds' pioneering work is famous for animating the synchronized motions of flocks, schools, or herds[1]. His algorithm is as follows:

1. Collision Avoidance: avoid collisions with nearby flockmates
2. Velocity Matching: attempt to match velocity with nearby flockmates
3. Flock Centering: attempt to stay close to nearby flockmates

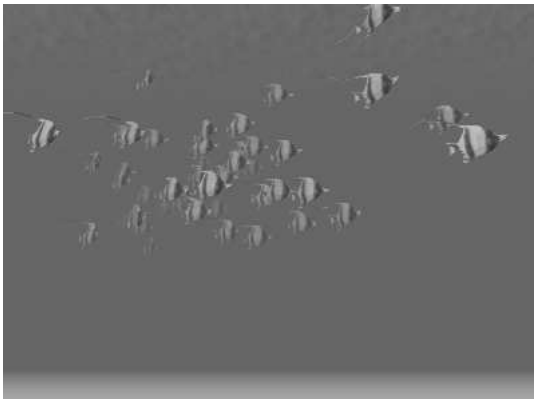

Fig.5: Schooling behavior of Zanclus cornutus(on SGI Onyx RE2)

In this research, we impose the above restrictions so severely that a fish swim at roughly an equal velocity and interval when it is drawn into a school. A fish sets its velocity and interval to those of the front fishes. The front fishes are found by the following equation.

$$\vec{c} \cdot \vec{d} = f, \tag{15}$$

where $\vec{c}$ is the detection vector that is a signal detected by the fish as to indicate the direction and distance to an obstacle or an other fish, and $\vec{d}$ is the direction vector that is the swimming direction and speed of the fish. If f is positive, they are in front of the fish.

Collision avoidance is realized by the following way. If a fish gets closer to an obstacle or an other fish than a threshold value, the new direction is determined by the following equation, where p is a parameter based on the distance to the obstacle. All the vectors are regularized.

$$\vec{c} \times \vec{d} = \vec{n} \tag{16}$$

$$\vec{n} \times \vec{d} = \vec{n'} \tag{17}$$

$$\vec{d} = \vec{d} + p \cdot \vec{n'} \tag{18}$$

In this way, if an obstacle exists lower left, the fish avoids it to upper right direction. Using the parameter p, we can realize the degree of avoidance, such that the fish avoids rapidly when the obstacle is close.

This algorithm requires the computational time of

$$O(N^2) + O(MN), \tag{19}$$

where M is the number of obstacles, and N is the number of fish. This is a serious problem of this algorithm. It is hoped to be improved by parallel processing, etc. Fig.5 shows a snapshot of the schooling behavior by this algorithm.

6 Conclusion

We have realized a realtime CG animation of highly realistic fishes incorporating deformation, locomotion and, further, an algorithm for schools. This system is attempted to improve the reality of fishes in both model and movement, compared with the realtime CG fish animation system that we have developed previously[2][3]. Tropical fishes are the important objects that give us natural relaxation. We think these virtual fishes will be useful, because raising tropical fishes in a water tank requires a lot of care, for example, feeding, changing the water, and so on. Our research way provides a basic technology for an electronic aquarium under consideration.

Fig.6: Virtual underwater environment (on SGI Onyx RE2)

X.Tu and D.Terzopoulos developed the various movement patterns, such as avoid, school, eat, mate, leave, wander. We want to consider such various movement patterns, because our present system does not support them.

A texture-mapped fish in the size of 400×200 pixels can swim naturally at the rate of 4 frames/sec on SGI Indy SC R4000. With this fish CG software we can run an underwater virtual environment and walk through it to watch about 10 kinds of fishes with a general-purpose high-end graphics workstation in realtime. Fig.6 illustrates one shot of its picture.

References

[1] C.W.Reynolds: "Flocks, Herds, and Schools: A Distributed behavioral Model", SIGGRAPH'87, pp.25-34(1987).

[2] H.Dohi and M.Ishizuka: "Realtime Synthesis of Super-Realistic Moving Goldfish using Small-scale Parallel Transputers", Communicating with Virtual Worlds(Proc. CGI'93, N.M.Thalmann, D.Thalmann(eds.)), Springer-Verlag, pp.462-472(1993).

[3] M.Fujiki, O.Hasegawa and M.Ishizuka: "A Prototype of Goldfish Software Robot with Real-time Response Function by a Parallel Computer", Proc. IEEE Int'l Workshop on Robot and Human Communications(ROMAN'92), pp.323-328(1992).

[4] T.W.Sederberg and S.R.Parry: "Free-Form Deformation of Solid Geometric Models", SIGGRAPH'86, pp.18-22(1986).

[5] X.Tu and D.Terzopoulos: "Artificial Fishes: Physics, Locomotion, Perception, Behavior", SIGGRAPH'94, pp.43-50(1994).