

# Physically Based Animation of Sea Anemones in Real-Time

José J. Aliaga\*

Universidad Politécnica de Madrid, Spain

Caroline Larboulette†

Universidad Rey Juan Carlos, Madrid, Spain



Figure 1: Anemone and Fish animations. Top row, the *attractive* fish swims from right to left over a *Strawberry* anemone: when the *regal tang* fish is close enough, the associated Hole field is activated (white sphere) and the tentacles are deformed. Bottom row, the *repulsive* fish swims from left to right over an *Anthopleura xanthogrammica*: the associated Source field creates deformations of the anemone’s tentacles when the *clown fish* passes by.

## Abstract

This paper presents a technique for modeling and animating fiber-like objects such as sea anemones tentacles in real-time. Each fiber is described by a generalized cylinder defined around an articulated skeleton. The dynamics of each individual fiber is controlled by a physically based simulation that updates the position of the skeleton’s frames over time. We take into account the forces generated by the surrounding fluid as well as a stiffness function describing the bending behavior of the fiber. High level control of the animation is achieved through the use of four types of singularities to describe the three-dimensional continuous velocity field representing the fluid. We thus animate hundreds of fibers by key-framing only a small number of singularities. We apply this algorithm on a seascape composed of many sea anemones. We also show that our algorithm is more general and can be applied to other types of objects composed of fibers such as seagrasses.

**CR Categories:** I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

**Keywords:** sea anemones, tentacles, animation, simulation, fish, real-time

## 1 Introduction

Sea anemones are an important component of under water landscapes, especially tropical reefs. With the success of the movie *Finding Nemo*, modeling of seascapes has gained in popularity.

\*e-mail: jj.aliaga@upm.es

†e-mail: Caroline.Larboulette@urjc.es

However, there is still a lack of real-time techniques that could be used for video games or virtual environments.

Sea anemones come in many shapes, sizes and colors. However, most species have a foot that attaches themselves to the rocks or anchors in the sand. Their main central part is a multi-purpose gastrovascular cavity of constant volume to which tentacles are attached. As we can see on figure 2, those tentacles are a group of fibers generally arranged in cycle as a result of spiral phyllotaxis. Those fibers serve as a defense mechanism to poison aggressors or as a trap to capture prey. For example, they react to human fingers by trying to catch them. From the viewpoint of an external observer, it looks like the fingers *attract* the tentacles. Fish will also tend to *attract* them as they pass-by, except for the clownfish that lives together with sea anemones in a symbiotic relationship. In that case, the clownfish will tend to *repulse* the tentacles that will move around the fish to allow it to go through.



Figure 2: Photos of real anemones in their environment. The central gastrovascular cavity is attached to the rocks thanks to a foot and covered with poisonous deforming tentacles. On the left, the *Anthopleura xanthogrammica*. On the right, the *Actinia fragacea* also known as Strawberry anemone.

Because they are made of soft material, the fibers also react to the surrounding fluid by deforming. For example, they will bend in the direction of the current.

In this paper, we propose a method for modeling and animating sea anemones together with their surrounding environment. We do not attempt to model the behavior of the anemone in a biologically correct way. Because the sea anemone often remains in the same place for several days, weeks or even months, we believe it is important

to model the movement of the tentacles while in this more passive state. When the anemone is anchored to the floor, its tentacles react to the movement of the surrounding fluid and fish by swaying in a smooth way.

In our model, the fibers are automatically generated to follow a spiral phyllotaxis pattern and animated by a physically based simulation. The environment is described by a continuous vector field composed of four types of singularities that can be animated. Sources and Holes that respectively repulse and attract serve as a mean to model the other living entities such as fish. Whirlwinds and Directional Fields are used to create currents of water. Because it would be impossible to control the animation on a per-fiber basis, we offer high level control through the animation of those singularities that can either be attached to other entities or key-framed. We are thus able to animate hundreds of tentacles by only key-framing a few singularities. Indeed, animating each tentacle by key-framing would be prohibitively expensive.

## 1.1 Background

Previous work on sea anemones modeling focuses on the *Stomphia Coccinea* [Liang et al. 2001]. The anemone and its tentacles are modeled as an implicit surface using the *Blob Tree* [Wyvill et al. 1999; Liang and Wyvill 2001]. The position of the tentacles is computed as the result of a modified phyllotaxis model. The main drawback of this model is its prohibitive rendering time (over 13 hours on a cluster in 2001 for a single anemone). We use a polygonal model for both, the foot of the anemone and its tentacles. The deformation is driven by an internal skeleton. [Nur et al. 2001] presented a model to animate the escape response of the sea anemone from the starfish. They aimed at creating a biologically accurate behavior of the anemone and the deformation itself was done by manually keyframing the parameters associated to the implicit primitives describing the anemone. We propose a technique to animate the fibers when the anemone stays in place. Our technique could be combined to the model of [Nur et al. 2001] to obtain a more natural swaying of the tentacles for a considerably reduced manual work (physically based simulation versus manual keyframing).

In addition, previous work on 3D animation of individual grass blades share some characteristics with the one of sea anemone fibers. Like [Perbet and Cani 2001; Bakay and Heidrich 2002; Endo et al. 2003; Ota et al. 2004], we use an internal skeleton to control the animation of each individual fiber. [Bakay and Heidrich 2002; Ota et al. 2004] compute the deformation of the skeleton thanks to a geometric technique similar to an inverse kinematics problem that gives the position of the intermediate nodes from the position of the top and bottom nodes. [Perbet and Cani 2001] and [Endo et al. 2003] express the skeleton position as a blending of pre-computed key positions. While any of those techniques could be used to animate the sea anemone tentacles, they all rely on keyframing or manual specification of each individual fiber deformation. We use a full physically based simulation of the dynamics of the skeleton that yields more precision and accuracy of the swaying movement as well as more natural deformation shapes over time.

Physically based simulation of branches swaying has been used for tree animation [Giacomo et al. 2001; Akagi and Kitajima 2006]. We use a similar idea on a single blade by propagating a moment down the chain. However, we have a dynamic approach combined to the use of an animated surrounding fluid composed of singularities. Indeed, while continuous fluid flows have been used in the past for creating motion paths to animate leaves [Wejchert and Haumann 1991] or to model static hair shapes [Hadap and Magnenat-Thalmann 2000], the key difference is that we use them in a non-

static framework where the singularities are attached to moving entities and/or key-framed which gives meaningful control of the animations.

To avoid the cost of a physically based simulation, [Stam 1997] has shown that a stochastic procedural technique based on the modal analysis of a tree deformation can generate visually appealing results. However, the interaction with entities like fish is not straightforward and the manipulation by an animator is necessary to create the desired deformation. We offer a higher level control with our technique by enabling the animator to influence the motion of the anemone tentacles by key-framing the position of fish swimming around rather than expressing the global deformation of the fibers directly.

## 1.2 Overview of the Proposed Technique

The environment is modeled by a continuous three dimensional vector field that represents the instantaneous velocity field of the fluid. Unlike previous methods, our field is not discretized in space [Endo et al. 2003] but continuous. It is modeled through the summation of four types of singularities [Wejchert and Haumann 1991] that we describe in section 2.

That field exerts pressure forces on the sea anemone fibers modeled as articulated skeletons covered with a skin (the model is described in section 3). Those forces are concentrated on the skeleton's nodes that are displaced towards equilibrium. The displacements of the nodes induce moments that tend to make the entire structure spin and that are taken into account to compute the displacement of the subsequent nodes down the chain. This computation is detailed in section 4.

The use of a continuous vector field to model the velocity of the fluid offers a key property to the animation of the fibers. Because they are influenced in a continuous way, self-collisions and collisions with other anemone fibers or fish are locally impossible by definition. Collision detection and response for the fibers is thus rather unnecessary except for collisions with objects that are not described through a singularity like the rocks in our implementation so far. This will be illustrated in section 5 and discussed in the future work section.

Notations: vectors whose names are composed of only one letter appear in bold. Non-bold or non-arrowed characters are scalars. For example,  $\mathbf{F}$  is a force while  $F$  is its modulus.

## 2 Modeling of the Conservative Fields

The environment is defined by a 3D continuous vector field of the form  $\mathbf{V}(\mathbf{p})$ . The intensity of the field  $f$  at a point  $\mathbf{p}(x, y, z)$  is given by the vector's modulus at this point:

$$f(\mathbf{p}) = \sqrt{V_x(\mathbf{p})^2 + V_y(\mathbf{p})^2 + V_z(\mathbf{p})^2}$$

The field is created by the summation of four types of singularities: Sources  $\mathbf{S}(\mathbf{p})$ , Holes  $\mathbf{H}(\mathbf{p})$ , Whirlwinds  $\mathbf{W}(\mathbf{p})$  and Directional fields  $\mathbf{D}(\mathbf{p})$  (see figure 3).

In an environment composed of  $S$  Sources,  $H$  Holes,  $W$  Whirlwinds and  $D$  Directional fields, the resulting vector field  $\mathbf{V}(\mathbf{p})$  is given by:

$$\mathbf{V}(\mathbf{p}) = \sum_{i=1}^S \mathbf{S}_i(\mathbf{p}) + \sum_{j=1}^H \mathbf{H}_j(\mathbf{p}) + \sum_{k=1}^W \mathbf{W}_k(\mathbf{p}) + \sum_{l=1}^D \mathbf{D}_l(\mathbf{p}) \quad (1)$$

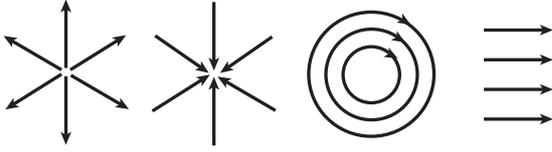


Figure 3: The four types of singularities used to model the fluid's velocity field. From left to right: Source, Hole, Whirlwind and Directional Field.

Sources, Holes and Whirlwinds are local fields. Their intensity is maximal in their center  $\mathbf{C}$  and inversely proportional to the distance of the sample point  $\mathbf{p}(x, y, z)$  to  $\mathbf{C}$ . The interesting consequence is that the computations can be greatly speeded up by using an octree decomposition of the singularities and bounding volumes for the influenced objects (see section 5). On the contrary, Directional fields are global and affect wider areas. A mathematical description of each type of field follows.

## 2.1 Sources

The Source creates a *repulsive* radial vector field with a maximum intensity  $\Phi_{max}$  in its center  $\mathbf{C}$  and a decreasing intensity towards the exterior. Let  $d(\mathbf{p}, \mathbf{C})$  be the Euclidean distance from point  $\mathbf{p}$  to the center  $\mathbf{C}$ ;  $\Phi$  the nominal intensity of the field; and  $\Phi_{max}$  the maximum intensity of the field. The Source field can be expressed by the following equation:

$$\mathbf{S}(\mathbf{p}) = \frac{\Phi}{\frac{\Phi}{\Phi_{max}} + d(\mathbf{p}, \mathbf{C})^2} \frac{\vec{\mathbf{C}}\mathbf{p}}{\|\vec{\mathbf{C}}\mathbf{p}\|} \quad (2)$$

$\vec{\mathbf{C}}\mathbf{p}$  is the vector from the center  $\mathbf{C}$  to the point  $\mathbf{p}$  where the field is being evaluated.

A key property of Source fields is that two sources of equal intensity create a stationary field ( $\mathbf{V}(\mathbf{p}) = 0$ ) at their midpoint. It is thus possible to create a wall in the fluid by aligning two series of Sources with the same intensity.

## 2.2 Holes

The Hole or Sink is the opposite of the Source. The intensity is the same but the direction is opposite. Instead of being *repulsive*, the Hole is *attractive*. This can be expressed by the following equation (the variables are the same as for equation (2)):

$$\mathbf{H}(\mathbf{p}) = -\mathbf{S}(\mathbf{p}) = -\frac{\Phi}{\frac{\Phi}{\Phi_{max}} + d(\mathbf{p}, \mathbf{C})^2} \frac{\vec{\mathbf{C}}\mathbf{p}}{\|\vec{\mathbf{C}}\mathbf{p}\|} \quad (3)$$

## 2.3 Whirlwind

The intensity of the field generated by a Whirlwind (or Vortex) is the same as the one generated by the Source or the Hole. The direction of the field, however, is orthogonal to both,  $\vec{\mathbf{C}}\mathbf{p}$  and the axis of rotation of the vortex  $\vec{\mathbf{R}}$ . The sign of the intensity defines if the vortex rotates clockwise or counterclockwise. This can be expressed by the following equation (the variables are the same as for equation (2)):

$$\mathbf{W}(\mathbf{p}) = \pm \frac{\Phi}{\frac{\Phi}{\Phi_{max}} + d(\mathbf{p}, \mathbf{C})^2} \frac{\vec{\mathbf{C}}\mathbf{p} \times \vec{\mathbf{R}}}{\|\vec{\mathbf{C}}\mathbf{p} \times \vec{\mathbf{R}}\|} \quad (4)$$

## 2.4 Directional Field

The Directional field is the only one to be global. The field is usually constant in direction although the direction could be modulated by a sine or cosine function. Its intensity can also be constant, but more interesting patterns can be obtained for varying intensities over space and over time. For an intensity function  $\Phi(\mathbf{p}, t)$  and a constant unit vector  $\vec{\mathbf{v}}$ , it can be expressed by the following equation:

$$\mathbf{D}(\mathbf{p}) = \Phi(\mathbf{p}, t) \cdot \vec{\mathbf{v}}; \quad (5)$$

The Directional field is very useful to model current or wind. By modulating the intensity by a sine or cosine function, it is possible to obtain wave effects. Other interesting effects such as an increasing intensity with the altitude can be created thanks to the  $\mathbf{p}$  parameter.

## 3 Our Anemone Model

### 3.1 Modeling

An anemone is composed of a foot on top of which are attached individual fibers. Each fiber is represented as an articulated skeleton made of a series of segments connected at nodes (see figure 7, left). The number of segments as well as their length and width varies depending on the type of plant. The root of each fiber's skeleton (at the bottom) is anchored to the foot. The foot is itself anchored to the floor or other objects such as rocks.

We model the foot of the anemone as a wide single fiber. The tentacles are placed on the foot using a collision-based simulation of spiral phyllotaxis [Fowler et al. 1992]. Figure 4 shows some examples of patterns for various radii of the foot, the mouth and the base of the tentacles.

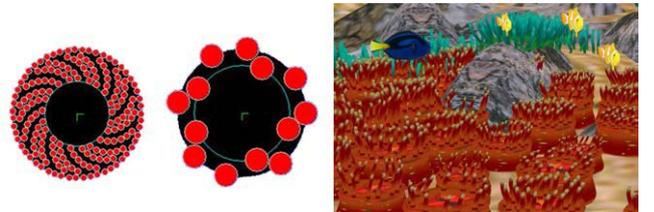


Figure 4: On the left, two examples of patterns with different radii for the anemone, the mouth and the tentacles. On the right, examples of various anemones automatically generated.

Alternatively, because we also use our model for seagrasses, we can model our plants by hand under Maya [Maya 2008] using the Skeleton tool and then export it to our animation software thanks to a MEL script we have developed.

### 3.2 Rendering

The geometry of both the foot and the fibers is created from the skeleton and the radii at the nodes. The radius  $r_i$  at a given node  $N_i$  defines a circle orthogonal to the segment  $L_i$ . Triangles are created to join consecutive circles (see figure 5). Textures are used to color the foot and tentacles of the anemone. Because the deformation of the fiber is skeleton driven, it would totally be possible to use another type of surface to represent the fiber such as a polygonal mesh deformed by skinning [Lewis et al. 2000] or implicit surfaces with skeletons [Bloomenthal and Wyvill 1990; Nur et al. 2001].

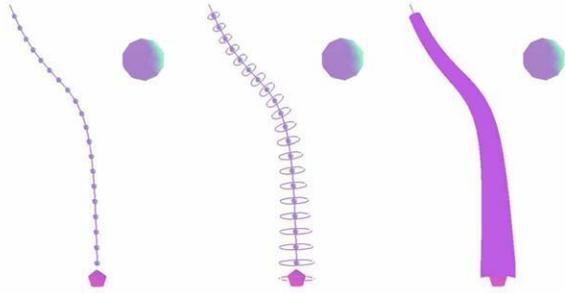


Figure 5: Our fiber model. From left to right: the nodes of the skeleton, the width of the nodes, the triangulation of the surface.

We generated 4 models of anemones inspired from existing species: the *Actinia fragacea* also known as Strawberry anemone; the *Anthopleura xanthogrammica*; the *Stomphia coccinea* and the *Anthothoe chilensis* (see figure 6). The position of the tentacles is automatically computed using a phyllotaxis model as described in the previous section. The textures have been created by hand from real photographs but are applied automatically.



Figure 6: Various types of anemones generated automatically from a small set of parameters. From left to right, top to bottom: the *Actinia fragacea*, the *Anthopleura xanthogrammica*, the *Anthothoe chilensis* and the *Stomphia coccinea*.

## 4 Deformation of a Fiber

The velocity field representing the fluid exerts pressure forces on the articulated skeleton. Those forces apply at the nodes to displace them. Each node has material properties such as stiffness that influence the way the chain bends and twists. It is important to note that

the length of each individual segment does not change during deformation. However, the fiber bends at the nodes towards equilibrium. This bending induces a moment on the next node down the chain. Indeed, the force applied at a node affects the whole structure that tends to spin.

The algorithm, that can be partially parallelized, can be summarized as follows:

```
//1. Computation of intensities at nodes
For each node Ni
  Force = 0;
  For each singularity
    Force += getForce(singularity, nodePosition);
  End for
  Force += ForceL(Ni+1) + Force_Moment;
//2. Computation of node displacement
BendingAngle = getBending(ForceT, Li);
//3. Propagation of moment down the chain
Moment = getMoment (ForceT, Li);
Force_Moment(Ni-1) = transmitMoment;
End for
```

with  $L_i = \|\vec{N}_i \vec{N}_{i-1}\|$  and  $Force = Force_L + Force_T$ .

The function *getForce* can be executed in parallel for nodes and singularities. Combined to an octree, the evaluation of the singularities intensities can be very efficient.

In our implementation, we evaluate the node forces, displacements and moments every 50ms. A smaller timestep increases stability but might prevent the simulation from running in real-time.

The three following subsections detail the three important steps mentioned in the algorithm:

1. Computation of the force from the velocity field ;
2. Computation of a node displacement ;
3. Computation of the transmitted moment.

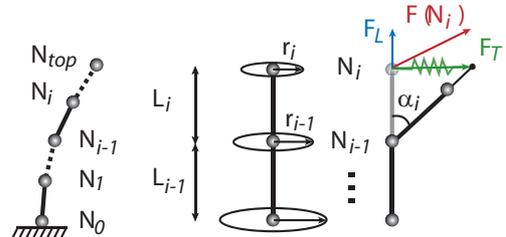


Figure 7: On the left, a complete articulated skeleton. In the middle, the radii and segment lengths related to the current node  $N_i$  and two nodes down the chain. On the right, the bending angle  $\alpha_i$  of node  $N_i$  as well as the forces exerted on this node.

### 4.1 Computation of the force from the velocity field

We approximate each fiber segment from  $N_{i-1}$  to  $N_i$  by a cylinder of radius  $r_i$  and height  $L_i$  (see figure 7). The pressure exerting on this cylinder at node  $N_i$  is given by equation (6).

$$\mathbf{F}_f = \frac{1}{2} \rho_{fluid} \mathbf{V}^2(N_i) L_i 2r_i C_D \frac{\mathbf{V}(N_i)}{\|\mathbf{V}(N_i)\|} \quad (6)$$

$\rho_{fluid}$  is the density of the fluid in  $kg.m^{-3}$ ;  $\mathbf{V}(N_i)$  is the instantaneous velocity at node  $N_i$  obtained from the distance field by equation (1);  $L_i$  is the length of segment  $N_i\vec{N}_{i-1}$ ;  $r_i$  is the radius of the fiber at  $N_i$  (see figure 7) and  $C_D$  is the drag coefficient that depends on the Reynolds number  $Re$ .

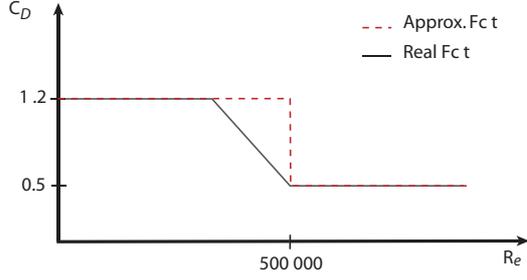


Figure 8: Relationship between the shape coefficient  $C_D$  and the Reynolds number  $Re$ . The (black) continuous curve depicts the real relationship while the (red) dashed curve shows the approximation we used in our implementation.

On the graph of figure 8, the (black) continuous curve shows the real relationship between  $C_D$  and  $Re$ . In our implementation we simplified this relationship by:

$$C_D = \begin{cases} 1.2 & \text{if } Re < 5.10^5, \\ 0.5 & \text{otherwise.} \end{cases} \quad (7)$$

depicted by the (red) dashed curve on the same graph.

The Reynolds number is a dimensionless number that compares the behavior of a fluid with an ideal fluid. It is computed as:

$$Re = \frac{\rho V 2r}{\mu} \quad (8)$$

$\mu$  is the viscosity coefficient of the fluid in  $N.s.m^{-2}$ .

For sea water,  $\mu_{H_2O} = 1.15.10^{-3} N.s.m^{-2}$  and  $\rho_{H_2O} = 1000 kg.m^{-3}$ . Hence  $Re = 9.10^5 V 2r > 5.10^5$  and  $C_D = 0.5$ .

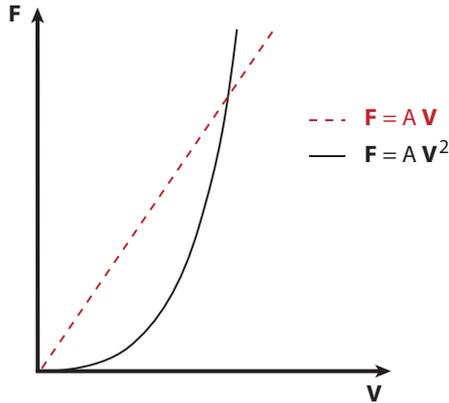


Figure 9: Simplification of the relationship between the force and the velocity.

For a fiber with constant diameter and distance between nodes constant, equation (6) simplifies into  $\mathbf{F}_f = A. \mathbf{V}^2$ ,  $A = constant$ . This however creates a quadratic relationship between the velocity field and the force as we can see on the graph of figure 9. To ensure a

more stable simulation, we approximate this relationship by a linear curve:

$$\mathbf{F}_f = A. \mathbf{V}(N_i) \quad (9)$$

with  $A = \frac{1}{2} \rho L r = constant$ .

This simplification is valid for velocities close to  $1.m.s^{-1}$  which corresponds to calm water. When an anemone decides to settle for a little while, it chooses a place with little current, usually protected by rocks. The forces are thus fast and easy to compute for a very small loss in accuracy and an improved stability.

## 4.2 Node Displacement

The node displacement is computed as the result of the application of an elastic force acting in the direction perpendicular to the fiber. Let's  $\mathbf{F}(N_i)$  be the net force applied to node  $N_i$ . For the top node of the fiber,  $\mathbf{F}(N_{top}) = \mathbf{F}_f(N_{top}) = A. \mathbf{V}(N_{top})$ . We will detail in section 4.3 how this net force is computed for the remaining nodes down the chain. As we can see on figure 7, this force can be decomposed into two components:  $\mathbf{F}(N_i) = \mathbf{F}_L(N_i) + \mathbf{F}_T(N_i)$ .  $\mathbf{F}_L$  is the part of the force in the direction of the current fiber segment  $N_{i-1}N_i$  while  $\mathbf{F}_T$  is the orthogonal component. As the fluid exerts forces as a pressure on the fiber, only the orthogonal component of the force is relevant to compute the displacement of the node.

Let's  $k_i$  denote the stiffness of the fiber at node  $N_i$ . The elastic force  $F_{elastic} = k_i u = F_T$ ,  $u$  being the elongation of the spring (see figure 7, right). As  $\mathbf{F}_T$  is orthogonal to the current segment, we can express  $\alpha_i$  as:

$$\tan(\alpha_i) = \frac{u}{L_i} = \frac{F_T}{k_i L_i} \quad (10)$$

It is thus straightforward to get the bending angle  $\alpha_i$  of the fiber and the new position of node  $N_i$ .

The stiffness parameter  $k_i$  influences how the fiber bends. It depends on the radius of the fiber  $r_i$  and on a material stiffness that is constant through the material:

$$k_i = r_i^n . K_{material} \quad (11)$$

The exponent  $n$  is a tunable parameter that is used to define how local the bending around a node is. Figure 10 shows different bending behaviors for different values of  $n$ . When  $n$  increases, the deformation becomes more local. This can be used to model a plant with very soft tips and stiff trunk.

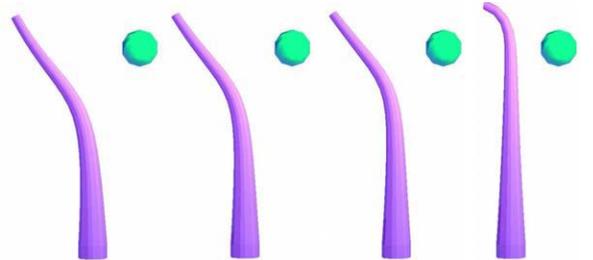


Figure 10: A fiber deformed by a Source. The parameters for the fiber and the singularity are the same in the three cases except for the exponent  $n$  of the radius influence in the stiffness function. From left to right:  $n = 1$ ,  $n = 2$ ,  $n = 3$  and  $n = 20$ . The higher the exponent, the more local the deformation is.

### 4.3 Propagation of the Moments and Forces down the chain

The displacement of a node induces a moment that tends to make the remainder of the chain spin. This moment  $\mathbf{M}(N_i)$  is a rotation vector that is orthogonal to both  $\mathbf{F}_T$  and  $N_{i-1}\vec{N}_i$ . Its norm is  $M(N_i) = F_T \cdot L_i$ . It generates a force on node  $N_{i-1}$  that can be expressed as:

$$\mathbf{F}_M(N_{i-1}) = \frac{M(N_i)}{L_{i-1}} \cdot \frac{N_{i-1}\vec{N}_{i-2} \times \mathbf{M}(N_i)}{\|N_{i-1}\vec{N}_{i-2} \times \mathbf{M}(N_i)\|} \quad (12)$$

The  $\mathbf{F}_L$  component of the force is directly transmitted to the following node.

At a given node  $N_i$ , the net force can be expressed as:

$$\mathbf{F}(N_i) = A \cdot \mathbf{V}(N_i) + \mathbf{F}_L(N_{i+1}) + \mathbf{F}_M(N_i) \quad (13)$$

For the first node (at the top),  $\mathbf{F}_L(N_{top+1}) = \vec{0}$  and  $\mathbf{F}_M(N_{top}) = \vec{0}$ . Figure 11 left shows the forces created by a singularity on the nodes of a fiber. On the right, we can see the modulus of the accumulated transmitted moments down the nodes (for visualization purposes, we did not apply the deformation of the fiber). We can see that this moment increases, causing an increase in stress. However, as the radius of the fiber also increases, the resulting deformation is smaller.

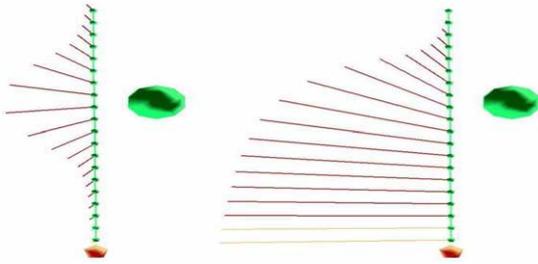


Figure 11: A fiber deformed by a Source. On the left, the force due to the singularity acting on the nodes ( $A \cdot \mathbf{V}(N_i)$ ). On the right, the modulus of the accumulated transmitted moments ( $M(N_i)$ ).

The last node is fixed to the floor. It is thus not displaced. The carried forces and moments acting on that node might be quite strong as we can see on figure 11, right. If the net force is above a threshold, the fiber could break.

## 5 Results

The animations presented in this section have been computed in real-time on a PC equipped with an AMD Athlon 64 X2 Core 2 2.21 GHz with 2Go of RAM and an NVIDIA 9800 GX2 graphics board. Our program has been developed with Java and runs under Windows XP.

### 5.1 Singularities and Fibers

Figure 12 shows the influence of the three types of singularities on a group of 100 identical fibers: a Source (green), a Hole (red) and a Whirlwind (yellow). The parameters we have used for the

singularities are the following:  $\phi_{Source} = 10000$ ;  $\phi_{Hole} = 10000$  and  $\phi_{Whirlwind} = 10000$ . Each fiber is composed of 22 nodes equally spaced by  $L_i = 5$  units. The fiber radius goes from 4 to 1 from bottom to top, decreasing the fiber's radius by 6% at each step. Stiffness of the fibers is 1.

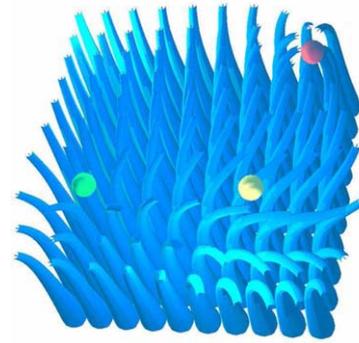


Figure 12: Influence of a Source (green), a Hole (red) and a Whirlwind (yellow) on a group of identical fibers.

**Performance** To test the performance of our algorithm, we have tested it on 36 Strawberry anemones, composed of 40 fibers each. Each foot as well as each fiber is animated thanks to 10 internal nodes which makes a total of 14760 skeleton nodes. The geometry is rendered from the 126720 vertices defined from the skeleton nodes. The animation runs at 17 fps (see figure 13). While the obtained framerate is satisfying, better performance could be obtained by using a more efficient programming environment as well as a parallel implementation of the vector field forces evaluation.

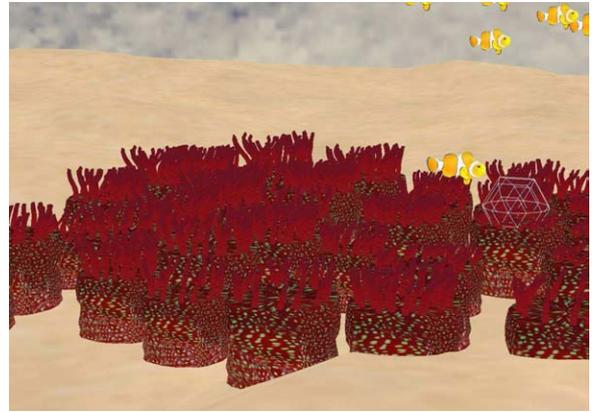


Figure 13: Influence of a Source (fish) on 36 anemones. 14760 nodes are animated at 17 fps.

**Bounding Volumes** To speed-up the computations, we have used spherical bounding volumes around each anemone. While the directional field simulating the current of water influences all nodes at all times, the moving singularities influence only the anemones that are within a certain distance. As a consequence, equations (2) to (4) that apply to each node of each fiber of each anemone do not need to be evaluated. This represents 14760 evaluations of the vector field for each fish in the environment in our example containing 36 anemones. Figure 14 shows such a bounding sphere.

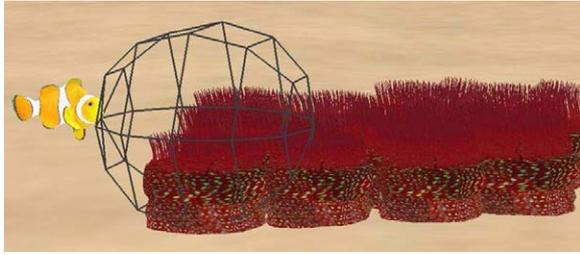


Figure 14: On the left, you can see the bounding volume around the fibers of one anemone. As long as the fish remains outside the volume, it does not influence the tentacles.

## 5.2 Singularity Keyframing

Previous models of grass or trees animation require the keyframing of individual fibers or the procedural specification of a bending position. Both those techniques are unsuitable to generate seascapes where a lot of anemones are present. Keyframing the fibers individually is not feasible and the use of precomputed bending deformation creates patterns that are too easily identifiable. In addition, it may be hard to include the influence of fish on the swaying of fibers.

Because we use singularities to model our fluid, we gain high level control of the animation. It is thus possible to obtain a swaying field of anemones and influence of fish by keyframing the singularities in position and intensity. In the following examples, the singularity is attached to the fish. Figure 15 shows the influence of changing the intensity  $\phi$  of the singularity that is keyframed from a Source to a Hole. Figure 16 shows the keyframing of the position of the singularity (the fish moves) and the corresponding deformation of the anemone. When the fish is far away from the anemone, its behavior is only influenced by the current. When the fish comes close enough, it influences the deformation of the anemone.



Figure 15: Influence of key-framing the intensity of the singularity attached to the fish. The anemone is deformed while the fish doesn't move.

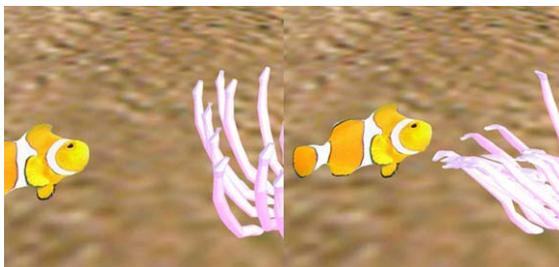


Figure 16: Influence of a Hole attached to a moving fish on an anemone. When the fish comes closer, it *attracts* the anemone. In real life, the anemone reacts to the fish by trying to capture it.

## 5.3 Anemones and Seagrasses

To generate a complex seascape, we have implemented an automatic terrain generator that uses Perlin Noise [Ebert et al. 1998]. We have added some rocks and different kinds of anemones and seagrasses in the background, animated by a gentle current. The seagrasses are simulated with the same algorithm as the anemones. The results can be seen on figure 17. Videos can be seen on the publication's website<sup>1</sup>.

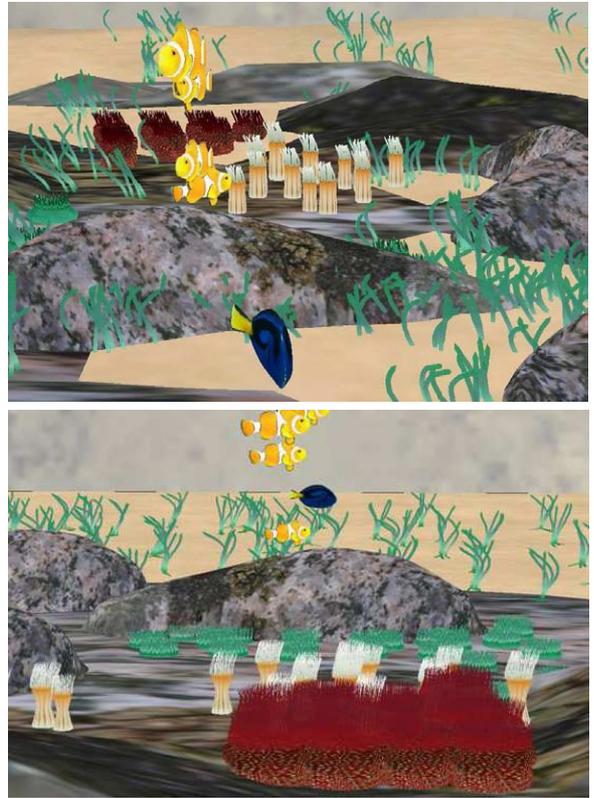


Figure 17: We have simulated various kinds of anemones and of fishes as well as seagrasses with our technique.

## 5.4 Other plants

We have applied our algorithm to the generation of other types of plants (see figure 18). From a set of 50 fibers with constant radius  $r_i = 3$  composed of 10 nodes equally spaced by  $L_i = 5$ , we automatically generate additional fibers by interpolation.

## 6 Conclusion and Future Work

We have presented a physically based model to animate sea anemone tentacles. To obtain a fast and stable algorithm, we have simplified some functions so that they remain valid for the conditions in which anemones settle (e.g. calm waters). Similarly, we have approximated some parameters by constants. On the other hand, we allow the user to tune the most relevant parameters in order to create a wide variety of plants. Those include the varying

<sup>1</sup><http://www.gmr.v.es/Publications/2009/AL09/>



Figure 18: We have simulated other types of plants with our technique. To the left: the 50 guide fibers are submitted to gravity; to the right: the resulting plant is obtained by interpolation of the guide fibers.

radius along the fiber, the exponent of this radius in the stiffness function and the material stiffness constant. We have shown that our model can be used for animating sea anemones and seagrasses. We plan on using it to model other kinds of plants such as grass and trees by tuning the fluid properties and fibers parameters. However, a non-experienced user might find it a little difficult to tune those parameters to simulate a given type of plant. We thus plan on creating a table of parameters that could be used as a reference.

Because the deformation of the fibers is entirely dependent on the singularities representing the environment, high level control of the animation is achieved through the control of their position and intensity over time. For example, a Hole can be attached to a fish so that it attracts the anemone fibers when coming too close as it would happen in real life (although for different reasons). It is thus easy to create an animation of a complicated landscape.

Finally, we have mentioned that because the velocity field representing the fluid is continuous, there is no need for self-collisions detection and response. This is also true for anemone-to-anemone or anemone-to-fish collisions. However, all of the objects that are not associated a vector field might be responsible for interpenetrations. For example, in some cases, the anemone fibers could go inside the rocks. It would be tedious (but possible) to add one or several vector fields to wrap each non-animated object of the scene. However, we believe that in many cases (such as the rock example), it would be more efficient to implement a penalty method for collision detection and response, especially as the surface of the fiber is described by a generalized cylinder around a skeleton. The skeleton nodes could thus be used to apply the penalty method.

The other strong aspect of our algorithm is that the velocity field of the fluid can be computed in parallel for each node of the skeleton of each fiber. The speed of our algorithm could thus be improved by a GPU implementation.

## Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments. This work has been partially supported by the Spanish Ministry of Education and Science (grant TIN2007-67188).

## References

AKAGI, Y., AND KITAJIMA, K. 2006. Computer animation of swaying trees based on physical simulation. *Computers & Graphics* (Aug.), 529–539.

- BAKAY, B., AND HEIDRICH, W. 2002. Real-time animated grass. In *Proceedings of Eurographics (short paper)*.
- BLOOMENTHAL, J., AND WYVILL, B. 1990. Interactive techniques for implicit modeling. *Computer Graphics, Proceedings of Symposium on Interactive 3D Graphics*, 2, 109–116.
- EBERT, D. S., MUSGRAVE, F. K., PEACHEY, D., PERLIN, K., AND WORLEY, S. 1998. *Texturing & Modeling: A Procedural Approach ; Second Edition*. AP Professional, Cambridge.
- ENDO, L. C. Y., MORIMOTO, C. H., AND FABRIS, A. E. 2003. Real-time animation of the underbrush. In *WSCG Short Papers Proceedings*.
- FOWLER, D. R., PRUSINKIEWICZ, P., AND BATTJES, J. 1992. A collision-based model of spiral phyllotaxis. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 361–368.
- GIACOMO, T. D., CAPO, S., AND FAURE, F. 2001. An interactive forest. In *Eurographics Workshop on Computer Animation and Simulation (EGCAS)*, Springer, M.-P. Cani, N. Magnenat-Thalmann, and D. Thalmann, Eds., 65–74. Manchester.
- HADAP, S., AND MAGNENAT-THALMANN, N. 2000. Interactive hair styler based on fluid flow. *Proceedings of Computer Animation and Simulation 2000* (Jan), 87–100.
- LEWIS, J. P., CORDNER, M., AND FONG, N. 2000. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of SIGGRAPH '00, ACM Computer Graphics*, ACM Press/Addison-Wesley Publishing Co., 165–172.
- LIANG, X., AND WYVILL, B. 2001. Hierarchical implicit surface refinement. In *CGI '01: Proceedings of the International Conference on Computer Graphics*, IEEE Computer Society, Washington, DC, USA, 291.
- LIANG, X., NUR, M. A., AND WYVILL, B. 2001. Modeling the structure of the sea anemone and the sea star using hierarchical implicit surfaces. *Computer Graphics and Applications, Pacific Conference on 0*, 0265.
- MAYA, 2008. Autodesk.
- NUR, M. A., LIANG, X., WYVILL, B., AND BOURNE, G. 2001. Animating the escape response of the sea anemone, *Stomphia coccinea* from the starfish, *dermasteris imbricata* modeled using implicit surfaces. *Computer Graphics International 2001. Proceedings*, 81–88.
- OTA, S., TAMURA, M., FUJIMOTO, T., MURAOKA, K., AND CHIBA, N. 2004. A hybrid method for real-time animation of trees swaying in wind fields. *The Visual Computer* 20, 10.
- PERBET, F., AND CANI, M.-P. 2001. Animating prairies in real-time. In *ACM-SIGGRAPH Symposium on Interactive 3D Graphics (I3D)*.
- STAM, J. 1997. Stochastic dynamics: Simulating the effects of turbulence on flexible structures. *Computer Graphics Forum (Proceedings of Eurographics'97)* 16, 3, 159–164.
- WEJCHERT, J., AND HAUMANN, D. 1991. Animation aerodynamics. *Compter Graphics (SIGGRAPH'91 proceedings)*.
- WYVILL, B., GUY, A., AND GALIN, E. 1999. Extending the csg tree - warping, blending and boolean operations in an implicit surface modeling system. *Comput. Graph. Forum* 18, 2, 149–158.