Dynamic Rope Animation Using Cubic B-Splines

CS 523 - Advanced Computer Graphics
Project Final Report
by
Oncel Tuzel

December 20, 2004

Abstract

The purpose of this study is to develop a physics based rope animation system using splines. Cubic b-splines are used to represent the rope model and animation is performed by moving the control points of the spline.

1 Introduction

The earlier methods in rope animation are based on particle systems [1]. The rope is usually represented with masses connected with springs. The method can be generalized to three dimensional surfaces to perform cloth simulation [2]. Although the method is simple, it has several drawbacks. The main drawback is force can be applied to system only at the masses. Also the spring stiffness should be very high to approximate rope behavior which causes well known numerical instabilities.

These kind of problems lead researchers to focus on continuous mass methods. Representing rope as a parametric curve is one them [3][4][5]. There are several types of curves that are widely used in computer graphics. One of the most interesting class is the splines which have nice locality and continuity properties. The curve is represented with a set of control points and basis functions. Splines can be classified as approximating or interpolating according to its relation with the control points that generates them. If the curve passes through the control points then it is called as interpolating, otherwise it is called approximating. Examples of approximating splines are Bezier Splines, B-Splines and NURBS, whereas Hermitian splines and Catmull-Rom splines are inside the interpolating class.

In this study we use uniform cubic b-spline rope model. Cubic b-splines has second order continuity which makes them well suited for graphical purposes. Each spline segment is defined by four control points and control points are uniformly distributed according to parametric position on the curve. As mentioned before curve does not pass through the control points. The control points are the degrees of freedom of the rope and animation is performed by moving them.

This paper starts with describing the mechanical system that models the kinematics and mass distribution. Lagrangian formalism is used as the dynamic equation of the system. Later on the forces acting on the rope is explained.

2 Spline Kinematics

First we describe the kinematics of the rope. Kinematics is defined in terms of positions and velocities of any point on the rope. Let $n+1$ be the number of control points on the rope. The number of spline segments is equal to number of control points minus degree of the spline. For cubic b-spline we have $n-2$ segments. The position of a point on the rope is defined as

$$P_j(s) = \sum_{i=0}^{n} b_i^j(s) q_i(t) \tag{1}$$

where j is the corresponding rope segment, s is the parametric position on the segment b_i is the basis function and q_i is the control point. The velocities of the rope points are found by taking the time derivatives of the position

$$\dot{P}_j(s) = \sum_{i=0}^{n} b_i^j(s)\dot{q}_i(t) \tag{2}$$

For cubic b-splines we have a locality of four. This means each segment is defined by four control points and the respective basis functions. The basis functions are

$$\begin{aligned} b_0 &= -s^3 + 3s^2 - 3s + 1 \\ b_1 &= 3s^3 - 6s^2 + 4 \\ b_2 &= -3s^3 + 3s^2 + 3s + 1 \\ b_3 &= s^3 \end{aligned} \tag{3}$$

divided by 6. We can see from the equations that the kinematics of the rope is defined in terms of $n + 1$ position functions over time, $q_i(t)$ representing the 3D position of the control points so we have $3(n + 1)$ degrees of freedom.

In this project we assumed uniform mass distribution over the rope. But this can simply be modified by adding a continuous mass function through the rope [3].

3 Dynamic Equations

After defining the kinematics and mass distribution of the rope, the next step is to define the dynamic equation of the system. The dynamic equation is defined in terms of Lagrangian equation

$$\frac{d}{dt}\frac{\partial}{\partial \dot{q}_i^\alpha}K - \frac{\partial}{\partial q_i^\alpha}K = Q_i^\alpha - \frac{\partial}{\partial q_i^\alpha}E \tag{4}$$

for all $\alpha \in \{x, y, z\}$ and $i = 0..n$. The terms in the equation are, K is the kinetic energy function, q_i is the degrees of freedom E is the potential energy of the rope and Q_i is the power ratings of the external forces acting on the rope distributed over the degrees of freedom.

We can write the kinetic energy function as

$$K(t) = \frac{1}{2}\sum_{j=0}^{n-2}\int_0^1 \rho_j(s,t)\dot{P}_j(s,t)^2 ds \tag{5}$$

We assumed uniform mass distribution over the rope so we can reduce the mass function $\rho(s, t)$ to a constant term. Substituting the velocity function (2) in (5) we can rewrite the kinetic energy function as

$$K(t) = \frac{1}{2}\sum_{j=0}^{n-2}\int_0^1 \left(\sum_{i=0}^{n} b_i^j(s)\dot{q}_i(t)\right)^2 ds \tag{6}$$

Taking the derivative according to degrees of freedom parameter q_i^α the second term on the left of equation (4) becomes zero. The first term becomes

$$\frac{\partial}{\partial \dot{q}_i^\alpha} K = \sum_{j=0}^{n-2} \sum_{l=0}^{n} \left(\int_0^1 b_i^j(s) b_l^j(s) ds \right) \dot{q}_i^\alpha \tag{7}$$

The equation (4) can be simplified as

$$\frac{\partial}{\partial \dot{q}_i^\alpha} K = \sum_{l=0}^{n} M_{il} \dot{q}_l^\alpha \tag{8}$$

where mass matrix M is defined as

$$M_{il} = \sum_{j=0}^{n-2} \int_0^1 b_i^j(s) \dot{q}_i(t) ds \tag{9}$$

The full system can be rewritten in a very nice form

$$M \ddot{q}^\alpha = W^\alpha \tag{10}$$

where W is equal to the all external forces minus derivative of the potential energy according to the degrees of freedom. Mass matrix of the system has very nice properties in terms of computational complexity. It is symmetric and banded with band equal to the degree of the spline. So for our case it is three. Solution of the equation (10) requires O(n) time to compute. Moreover mass matrix is independent of time so it can be inverted once at the start of the simulation and used later on.

4 Forces Acting on the Rope

In order to animate the rope, forces are applied. The applied mechanism allows us to apply forces at any point on the rope. In equation (4) forces are due to two factors. First one is the external forces acting on the rope and second one is the forces due to potential energy of the rope. First we explain the force fields gravity, viscosity and later the generic forces such as rope elasticity, fixed point and collision. Due to assumption of the uniform mass distribution the mass functions are eliminated from the derivations.

4.1 Gravity

Gravity force derives from the potential energy of the rope and acts on the negative y axis. Potential energy due to gravity can be written as

$$\begin{aligned} E_{grav} &= \sum_{j=0}^{n-2} \int_0^1 P_j^y(s,t) ds \qquad (11) \\ &= \sum_{j=0}^{n-2} \int_0^1 \sum_{i=0}^{n} b_i^j(s) q_i^y(t) ds \end{aligned}$$

Taking the derivative according to degrees of freedom we find the power ratings of the force acting on the control points

$$W^y_{grav,i} = -\sum_{j=0}^{n-2} \int_0^1 b_i^j(s)ds \tag{12}$$

The derivatives according to x and z axes are equal to zero. This force is independent of time so it can be computed once and used later on.

4.2 Viscosity

The viscosity force is due to the air friction. It derives from the movement of the rope. The force acting on the infinitely small part of the rope due to viscosity can be written as

$$\begin{aligned} F_j(s,t) &= -C\dot{P}_j(s,t)ds \qquad (13) \\ &= -C\sum_{i=0}^{n} b_i^j(s)\dot{q}_i(t) \end{aligned}$$

where C is the viscosity constant. The force acting on a control point due to movement of this infinitely small part is equal to multiplication of the relative basis function of the control point evaluated at the parametric position of the part, with the force (13). If we integrate this force all over the rope we get the viscosity force distributed over the control points

$$\begin{aligned} W^\alpha_{visc,i} &= -C\sum_{j=0}^{n-2}\sum_{l=0}^{n} \dot{q}_l^\alpha(t) \int_0^1 b_l^j(s)b_i^j(s)ds \qquad (14) \\ &= -C\sum_{j=0}^{n-2} \dot{q}_l^\alpha(t) \sum_{l=0}^{n} \int_0^1 b_l^j(s)b_i^j(s)ds \end{aligned}$$

Again the integral term can be computed once and during animation it can be multiplied with the velocities of the degrees of freedom to find the forces.

4.3 Generic Force

Generic forces differ from the two kinds of forces defined before because it acts on a single point on the rope. Let j be the application rope segment and s be the parametric position of the force that is applied to the rope. The distribution of the forces to the control points are equal to the multiplication of the relative basis function of the control point evaluated at the parametric position of point

$$W^\alpha_{F,i} = F^\alpha b_i^j(s) \tag{15}$$

4.3.1 Rope Elasticity

Rope elasticity is a special case of the generic force. It is achieved by attaching elongated springs throughout the rope. If the length of the spring exceeds the rest length of the spring then force is applied to the connection points. If the spring shortens no force is applied. To prevent infinitely motion of the spring we use damped springs. Forces acting on the one connection point can be written as

$$F_{elas} = -\left[k_s(|l| - r) + k_d\frac{\dot{l}l}{|l|}\right]\frac{l}{|l|} \tag{16}$$

where k_s is the spring constant, k_d is the damping constant, l is the distance vector of the connection points, r is the rest position length and $\dot{l}$ can be computed by subtracting the relative velocities of the connection points. Force acting on the other connection point is equal to the negative sign of the force.

4.3.2 Fixed Point Force

Fixed point in the system is performed by penalty method [6]. One end of the rope is attached to a fixed point in the world by a damped spring. Motion is performed by moving the constant point. The point is moved by a constant velocity according to the user input. The force is calculated by equation (16).

4.3.3 Collision Force

Only collision with the ground plane is defined. Collision is detected by checking the y coordinate of sampled points from the rope. If the position is close enough to the ground plane, collision occurs. Response force is again calculated with penalty method. A damped spring pushing contact points in positive y direction is temporarily inserted. Also friction force is applied to the contact points acting on the negative velocity direction on $x - z$ plane.

5 Constraints

Although penalty based methods works fine for our system better performance can be achieved by adding the constraints directly to the dynamics equation. Lagrange multipliers can be used to solve the final system. Using the method the equations become

$$\frac{d}{dt}\frac{\partial}{\partial \dot{q}_i^\alpha}K - \frac{\partial}{\partial q_i^\alpha}K = W_i^\alpha + \sum_{c=0}^{n} cL_{ci}^\alpha \lambda_c \tag{17}$$

and

$$\sum_\alpha \sum_{l=0}^{n} L_{cl}^\alpha = D_c \tag{18}$$

whereλ_c is the Lagrange multiplier of the constraint c. The first equation adds the required force into the system and the second one relates the nc constraints to the unknowns.

With the Lagrange multipliers several nice properties of the dynamic equation are lost. If the constraints realized change with the time, it is impossible to invert the mass matrix at the start of animation and use later on. Moreover the matrix is not sparse anymore and inverting the system requires $O(n^3)$ operations. Lagrange multiplier method is not implemented in the system.

6 Integration

Solving the dynamic equation we calculate the accelerations of the control points. The next step is to update the velocity and position by numerical integration. In the implementation, Newton-Euler integration is used. We approximate the rope elasticity by a finite number of springs. To make system act naturally, the spring constants should be very high. This results in high accelerations and if time steps are not small enough causes numerical instabilities.

The results show that Newton-Euler integration is satisfactory for our animation system. Although the stability is not guarantied, using higher degree integration methods the stability of the system can be increased [7]. Also implicit methods might be used to achieve guarantied stability [8].

7 Results

The system is implemented using Visual C++ and OpenGL. CLAPACK library is used for numerical routines. The system runs realtime on a P3 1GHz machine with 512 MB memory. At one second 10000 iterations are calculated. In the current implementation 20 internal springs are attached to perform rope elasticity. Collision detection and response is performed throughout 20 points. The rope has 11 control points. The results are satisfactorily realistic. Due to few number of control points sometimes motions look more smooth than it should be. Using more control points animation becomes more realistic with increased complexity.

References

[1] A. Witkin and D. Baraff. Physically based modeling: Particle dynamics. Technical report, Pixar Research, 2001.

[2] X. Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *Graphics Interface*, pages 147–154, 1995.

[3] Y. Remion, J.M. Nourrit, and D. Gillard. A dynamic animation engine for generic spline objects. *Journal of Visualization and Computer Animation*, pages 17 – 26, 1995.

[4] Y. Remion, J.M. Nourrit, and D. Gillard. Dynamic animation of spline like objects. In *WSCG*, pages 426–432, 1999.

[5] J. Lenoir, P. Meseure, and C. Grisoni, L.and Chaillou. Surgical thread simulation. *Modelling and Simulation for Computer-aided Medicine and Surgery (MS4CMS)*, 2002.

[6] M. Moore and J. Wilhelms. Collision detection and response for computer animationr3. *SIGGRAPH Comput. Graph.*, 22(4):289–298, 1988.

[7] A. Witkin and D. Baraff. Physically based modeling: Differential equation basics. Technical report, Pixar Research, 2001.

[8] D. Baraff and A. Witkin. Large steps in cloth simulation. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 43–54. ACM Press, 1998.