EUROGRAPHICS 2002 / G. Drettakis and H.-P. Seidel
(Guest Editors)

Volume 21 (2002), Number 3

STRANDS: Interactive Simulation of Thin Solids using Cosserat Models

Dinesh K. Pai

Department of Computer Science, The University of British Columbia, Vancouver, Canada
pai@cs.ubc.ca

Abstract

STRANDS *are thin elastic solids that are visually well approximated as smooth curves, and yet possess essential physical behaviors characteristic of solid objects such as twisting. Common examples in computer graphics include: sutures, catheters, and tendons in surgical simulation; hairs, ropes, and vegetation in animation. Physical models based on spring meshes or 3D finite elements for such thin solids are either inaccurate or inefficient for interactive simulation. In this paper we show that models based on the Cosserat theory of elastic rods are very well suited for interactive simulation of these objects. The physical model reduces to a system of spatial ordinary differential equations that can be solved efficiently for typical boundary conditions. The model handles the important geometric non-linearity due to large changes in shape. We introduce Cosserat-type physical models, describe efficient numerical methods for interactive simulation of these models, and implementation results.*

1. Introduction

Modeling deformable objects is one of the most challenging problems in physically-based computer graphics. In recent years there has been considerable progress on two fronts. Deformable models based on networks of masses and springs are now widely used. More recently, the principles of 3D elasticity and their numerical solution using the finite element method (FEM) or the boundary element method (BEM) are now well understood by the graphics community.

Our goal is to develop a general modeling primitive that is well suited for thin deformable objects. Examples of such objects are abundant: in computer animation, for example, they include wires, hairs, telephone cables, ropes, grape vines, and willow branches. Barzel [5] describes many uses of these types of thin objects in computer animation in general and the movie Toy Story, in particular. In feature films, animator control of the thin objects within a traditional keyframe animation pipeline is more important than physical simulation. However, in interactive applications such as surgical simulation and games, simulation can provide major benefits.

Our own motivation is in surgical simulation, particularly simulation of surgical sutures — the ubiquitous "threads" used in surgery for sewing, cutting, and tying. See Fig. 1. Skilled use of sutures is a fundamental manipulation that is taught to all surgeons [18]. Learning to handle sutures becomes particularly important in minimally invasive surgery using laparoscopic tools inserted inside the body, since the surgeon has to perform delicate operations while looking at a computer monitor. Sutures exhibit a variety of complex behaviors characteristic of solid deformable objects including global twisting and bending deformation as a result of local forces. Therefore simulating the realistic behavior of sutures is an important part of any surgical simulation system. Other surgically significant thin elastic objects include wires, catheters, nerves, tendons, and blood vessels.

Unfortunately, none of the techniques for physically-based modeling currently used in graphics is particularly well suited for modeling thin deformable objects. Modeling these as 3D elastic solids requires very fine FEM meshes to correctly capture the global twisting behavior that is observed when torques are applied along the axis of the strand (you can try this by rolling a mouse cable or dental floss between your fingers). BEM is well known to be inappropriate for modeling thin objects. Models using meshes of mass particles and springs have similar problems since they require a large number of particles and springs to correctly reproduce

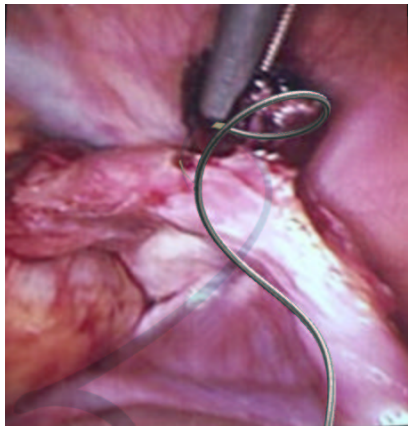

Figure 1: *A simulated strand of surgical suture that can twist and curl during manipulation of the needle during laparoscopic surgery.*

the twisting behavior and to prevent the mesh from collapsing during twisting. Models based on rigid bodies connected by torsional springs are more robust but still suffer from having a fixed, arbitrary discretization into rigid bodies: for example it is difficult to reproduce high curvature bending unless a very fine chain is used. It is also not clear how to relate the spring constants to elastic material parameters. In this paper we propose an alternative model, which is based on a solid theoretical footing, and yet quite efficient and suitable for interactive simulation.

The purpose of this article is two fold. First, we introduce Cosserat elasticity as a useful, accurate, physical model for simulating thin deformable objects. Second, we show how the physical models can be discretized and solved efficiently for many applications in computer graphics. This combination of physical model and numerical solution, which we call a STRAND, forms a useful modeling primitive in computer graphics for a wide variety of thin deformable objects. We also briefly describe a specific application to the simulation of surgical sutures; however our purpose is not to describe a complete system which would require addressing other important issues such as rendering and interaction.

The rest of the paper is organized as follows. After describing some related work in Sec. 1.1, we present an elementary, self-contained introduction to Cosserat rods in Sec. 2. We describe our simulation algorithm in Sec. 3 and discuss how strands can be used in practice in Sec. 4.

1.1. Related Work

Thin strand-like solids have been modeled in computer graphics for many specific applications. Barzel [5] provides both the motivation for modeling ropes, springs, and other strand-like objects and useful techniques for use in keyframe animation. Perhaps the most common thin solid in computer animation is hair. We refer the reader to the recent survey by Magnenat-Thalmann et al. [15] The closest to the models proposed here are the "explicit" hair dynamics models [10, 12, 1, 23] which model hair as series of rigid bodies or masses connected by springs. These models are similar in spirit to the models we describe, except that our model is based on a sound theoretical footing and is more general, and therefore allows more choices in numerical solution methods.

The computer graphics community does not appear to be aware of Cosserat continua as models of thin objects, called shells and rods, developed in the solid mechanics community. There is a long history of these models, going back to the work of Euler on modeling the "elastica," Kirchhoff's theory of rods, leading to the formulation of rods and shells as curves and surfaces with directors, developed by the Cosserat brothers at the beginning of the twentieth century. Modern treatments of Cosserat models can be found in books by Antman [2] and Rubin [24] (see also [17]). Cosserat models have been used for a variety of tasks such as modeling utility cables (e.g., [13]), but perhaps the most popular use today is in modeling the mechanical behavior of DNA strands (e.g., [16]).

2. Strands as Cosserat Rods

We now describe the mathematical model of a Cosserat rod. This model is a special case of a more general theory of Cosserat continua which includes shells and points. More exhaustive formulations can be found in [2, 24]. In our description we borrow from the notation used for DNA modeling by Maddocks and his co-workers [14], and from the multibody robot dynamics literature [19]. Since the graphics community is familiar with multibody dynamics, we will try to show the connections between the two.

2.1. Kinematics

The configuration of a strand is described by a space curve $\mathbf{r}(s)$ and a coordinate frame of "directors" attached at each point on the curve. See Fig. 2. Following the usual practice in graphics and robotics, we will assemble these into a coordinate frame $\mathbf{E}(s) = [\mathbf{e}_1\ \mathbf{e}_2\ \mathbf{e}_3\ \mathbf{r}](s)$. This has the usual 4×4 matrix representation E if we express the vectors $\mathbf{e}_i$ and the point $\mathbf{r}$ in homogeneous coordinates with respect to any reference frame. Please notice the font convention used: a quantity written as "$\mathbf{a}$" is an abstract vector or tensor, while "a" is a matrix of its coordinates. We note that frame $\mathbf{E}$ is not the same as the Frenet frame of the curve, but provides extra information about the twisting of the curve. We will assume, however, that the frame is "adapted" to the curve, i.e., we take the "Z" axis, $\mathbf{e}_3$ to be aligned with the tangent to the curve, and we will assume that the parameter s is arc length at rest.

Thus given the reference configuration of a strand, de-

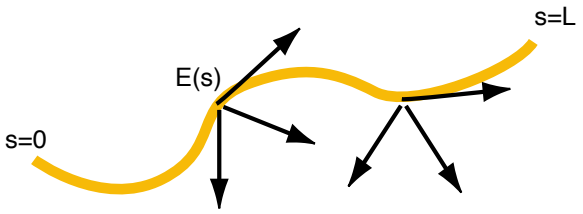

Figure 2: *A Cosserat Rod*

scribing it in any other configuration in which it may be twisted and bent is equivalent to describing the "motion" of the frame **E** as a function of s. Therefore, there is a close analogy between describing a Cosserat rod in space, and *describing the motion of a rigid body (frame), if s was interpreted as time*. Therefore, we can expect to find fast algorithms for computing the configuration of a rod which are similar to fast algorithms for dynamics [9, 4, 19]. We explore this in greater detail below and in Sec. 3.

As with rigid body motion, the s-derivative of the frame, $\mathsf{E}' = d\mathsf{E}/ds$, is best described in terms of "spatial derivatives," $\begin{pmatrix} \mathsf{u}^T, \mathsf{v}^T \end{pmatrix}^T$. We can formally define these as follows. Let

$$\mathsf{E}^{-1}\mathsf{E}' \equiv \begin{pmatrix} [\mathsf{u}] & \mathsf{v} \\ 0 & 0 \end{pmatrix}. \tag{1}$$

It is well known (and easy to show) that the 3×3 matrix $[\mathsf{u}]$ is the skew-symmetric matrix of the cross product, $\mathsf{u}\times$. We take these as the definitions of the "velocities" u and v. The vector u is called the Darboux vector and is analogous to the angular velocity, and v is analogous to the linear velocity of a point at the origin. For a rod, however, these are the rotational and translational *strains*, not velocities.

We note that Eq. 1 is a differential equation which defines the evolution of the director frame **E**. It is particularly convenient to integrate this in relative coordinates, since E is then the identity matrix, and

$$\boxed{\mathsf{E}' = \begin{pmatrix} [\mathsf{u}] & \mathsf{v} \\ 0 & 0 \end{pmatrix},} \tag{2}$$

which we call the *kinematic differential equation*.

2.2. Force Balance

We can similarly define a *stress differential equation*. Let $\xi = \begin{pmatrix} \mathsf{m}^T, \mathsf{n}^T \end{pmatrix}^T$ be the stresses at s, i.e., the transmitted torque and force per cross-section area of the rod. The change in stresses at s must be due to the applied torque and force per unit length, $\eta = \begin{pmatrix} \tau^T, \mathsf{f}^T \end{pmatrix}^T$, at s (e.g., due to gravity, inertia, or contact):

$$\tfrac{d}{ds}\xi = \eta. \tag{3}$$

As with the kinematic differential equation, we would like to compute this in relative coordinates, but there is the standard complication of differentiating a spatial force in "moving coordinates" (e.g., see [9, 19]). If we use "$'$" solely to denote differentiation in relative coordinates, we have the well known identity

$$\tfrac{d}{ds}\xi = \xi' + \begin{pmatrix} [\mathsf{u}] & [\mathsf{v}] \\ 0 & [\mathsf{u}] \end{pmatrix}\xi. \tag{4}$$

We can now combine these into the stress differential equation:

$$\boxed{\xi' = \eta - \begin{pmatrix} [\mathsf{u}] & [\mathsf{v}] \\ 0 & [\mathsf{u}] \end{pmatrix}\xi.} \tag{5}$$

2.3. Constitutive Laws

The relationship between the stress and strain is a material property, usually specified using an empirical constitutive law. For many materials, linear constitutive laws are adequate. It is important to note that a Cosserat rod can undergo a large global deformation while still having small strains at each point. Thus important nonlinear effects due to changes in shape are fully accounted for, even if the material behavior is linear.

For many applications in graphics, it is sufficient to assume the following simple constitute law:

$$\mathsf{m} = \mathsf{K}(\mathsf{u} - \hat{\mathsf{u}}), \tag{6a}$$

$$\mathsf{n} = \mathsf{L}(\mathsf{v} - \hat{\mathsf{v}}). \tag{6b}$$

where $\hat{\mathsf{u}}$ and $\hat{\mathsf{v}}$ are the strains at rest. Note that the rest strains could correspond to twisted coil, for instance a telephone cord or spring.

More general laws (e.g.,"hyperelastic" materials) are relatively easy to incorporate. Dynamics can be incorporated here as well by relating stress to deriviates of strain (see [2]). We focus on quasistatic models here which are good approximations for highly dissipative systems.

One common assumption is that the rod is inextensible and unshearable, which is a good model of typical stiff materials like surgical sutures whose shape changes mainly due to bending and twisting. This corresponds formally to an infinite stiffness L, but it is better to view this simply as fixing $\mathsf{v} = \hat{\mathsf{v}}$.

3. Simulating Strands

Equations 2 and 5, together with Eq. 6, form a system of coupled differential equations which must be solved to determine the configuration of a strand. The important fact about these equations is that they form a set of Ordinary Differential Equations (ODE) in one independent variable, rather than a system of partial differential equations that would result from full 3D elasticity. This is probably the most significant benefit of modeling strands as Cosserat rods.

However, it is still a boundary value ODE, and typically boundary conditions are specified not just at one end (which would lead to an initial value ODE, similar to those encountered in rigid body dynamics simulation). Depending on the type of boundary conditions, it can be considerably more difficult to solve. We first consider a particular type of two-point boundary condition, which we call the Standard BVP. These are as easy to solve as initial value problems using a fast algorithm closely related to linear time dynamics algorithms for rigid body dynamics [9, 4, 19]. Then we discuss how more general boundary value problems could be handled.

In the Standard BVP we specify the position and orientation at one end, and the stresses at the other. That is, for a rod of normalized length 1, we want

$$\mathsf{E}(s=0) = \mathsf{E}_0 \tag{7}$$
$$\xi(s=1) = \xi_1 \tag{8}$$

We further assume that the rod is inextensible and unshearable.

We can discretize the differential equations as follows. For simplicity, we show a first-order accurate (Euler) discretization here; higher order methods are similar. We use a notation from robotics [19] to keep track of local coordinates at mesh points. We will denote a value y at mesh point j as y_j. The step size between mesh points is denoted $h_j = s_{j+1} - s_j$. The homogeneous coordinates of frame i, relative to another frame j, is given by the 4×4 matrix ${}^{j}_{i}\mathsf{E}$. We will always use leading subscripts and superscripts to indicate frames. The coordinates of a vector $\mathbf{y}$, in frame i, are given by the column matrix ${}^{i}\mathsf{y}$. We convert it to frame j coordinates this way: ${}^{j}\mathsf{y} = {}^{j}_{i}\mathsf{E}\ {}^{i}\mathsf{y}$.

We first discretize the stress differential equation, Eq. 5. Since the stress boundary condition is specified at the end ($s = 1$) we start there and propagate stresses to $s = 0$. The translational stress is as follows:

$${}^{j}\mathsf{n}_{j-1} = {}^{j}\mathsf{n}_j + h_{j-1}\left([{}^{j}\mathsf{u}_j]\,{}^{j}\mathsf{n}_j - {}^{j}\mathsf{f}_j\right), \tag{9a}$$
$${}^{j\text{-}1}\mathsf{n}_{j-1} = {}^{j}_{j\text{-}1}\mathsf{E}^{T}\ {}^{j}\mathsf{n}_{j-1}. \tag{9b}$$

The last equation merely transforms the stress coordinates from frame j to j-1, and we use the inverse transpose because the stresses are covariant vectors (and therefore transform differently from contravariant vectors like strains).

Instead of integrating the rotational stress m, we eliminate it using the constitutive law Eq. 6 and integrate the rotational strain u instead.

$${}^{j}\mathsf{u}_{j-1} = {}^{j}\mathsf{u}_j - ({}^{j}\hat{\mathsf{u}}_j - {}^{j}\hat{\mathsf{u}}_{j-1}) + h_{j-1}K^{-1} \times \left([{}^{j}\mathsf{u}_j]K({}^{j}\mathsf{u}_j - {}^{j}\hat{\mathsf{u}}_j) + [{}^{j}\mathsf{v}_j]\,{}^{j}\mathsf{n}_j - {}^{j}\tau_j\right), \tag{10a}$$
$${}^{j\text{-}1}\mathsf{u}_{j-1} = {}^{j\text{-}1}_{j}\mathsf{E}\ {}^{j}\mathsf{u}_{j-1}. \tag{10b}$$

If K is a scalar, then Eq. 10a simplifies to

$${}^{j}\mathsf{u}_{j-1} = {}^{j}\mathsf{u}_j - ({}^{j}\hat{\mathsf{u}}_j - {}^{j}\hat{\mathsf{u}}_{j-1}) + h_{j-1}\left([{}^{j}\hat{\mathsf{u}}_j]\,{}^{j}\mathsf{u}_j + K^{-1}\left([{}^{j}\mathsf{v}_j]\,{}^{j}\mathsf{n}_j - {}^{j}\tau_j\right)\right). \tag{11}$$

Thus we propagate u and n from end to start. For an inextensible, unshearable rod, $\mathsf{v} = \hat{\mathsf{v}} = \mathsf{e}_3$; the last equality is due to our use of an adapted frame. Since we now know all the strains in the strand, we can integrate the positions from start to end.

Integrating Eq. 2 is more complicated. One way is to convert the rotational part of the differential equation into one involving derivatives of unit quaternions (also known as Euler parameters), but integration requires enforcing a constraint on the magnitude of the quaternion. Instead we observe that

$$\mathsf{E} = \exp\left[\begin{pmatrix}[\mathsf{u}] & \mathsf{v}\\ 0 & 0\end{pmatrix}(s - s_j)\right] \tag{12}$$

satisfies the kinematic differential equation Eq. 2 at s_j. This can be directly discretized and written in a more convenient form as

$${}^{j}_{j+1}\mathsf{E} = \exp\left[\begin{pmatrix}[\omega] & \mathsf{v}\\ 0 & 0\end{pmatrix}t\right] \tag{13}$$

Where $\omega = {}^{j}\mathsf{u}_j/|{}^{j}\mathsf{u}_j|$, $\mathsf{v} = {}^{j}\mathsf{v}_j/|{}^{j}\mathsf{u}_j|$, and $t = |{}^{j}\mathsf{u}_j|h_j$. The matrix exponential in this form can be separated into rotational and translational parts as [6]

$${}^{j}_{j+1}\mathsf{E} = \begin{pmatrix}e^{[\omega]t} & (I - e^{[\omega]t})[\omega]\mathsf{v} + \omega\omega^{T}\mathsf{v}t\\ 0 & 1\end{pmatrix}. \tag{14}$$

We can compute the rotational part of Eq. 14 efficiently using Rodrigues' formula

$$e^{[\omega]t} = I + [\omega]\sin t + [\omega]^2(1 - \cos t). \tag{15}$$

Evaluation of the translational part of Eq. 14 is simplified for an inextensible rod with an adapted frame, since $\mathsf{v} = \mathsf{e}_3 = \begin{pmatrix}0 & 0 & 1\end{pmatrix}^{T}$. Together, ${}^{j}_{j+1}\mathsf{E}$, gives the frame at the next mesh point. It can be expressed relative to any coordinate frame k (including the world frame) as

$${}^{k}_{j+1}\mathsf{E} = {}^{k}_{j}\mathsf{E}\ {}^{j}_{j+1}\mathsf{E} \tag{16}$$

Therefore the Standard BVP can be solved very efficiently by two sweeps of the strand, from end to start to compute the stresses and strains, and from start to end to compute the position and orientation at each mesh node. This algorithm is entirely analogous to linear time dynamics algorithms for articulated rigid bodies connected by joints [9, 4, 22, 19] where instead of stresses, joint forces and torques are propagated in the first sweep. Similar algorithms are used in other areas where the Standard BVP arises, such as optimal state estimation using Kalman filters [21]. We note that unlike in articulated rigid body systems, the mesh points are not chosen by the physical model but for numerical convenience. Therefore

whe have more flexibility in chosing the step size h_j; for instance we could adaptively select h_j based on the curvature of the strand or error estimates in the numerical integration. In addition, by using a multistep discretization, we can also incorporate the influence of several neighbours to regularize the solution.

4. Modeling with Strands

We now describe how the strand model, i.e., a Cosserat rod model discretized as shown in Sec. 3, can be used to model thin objects that arise in computer graphics.

The most direct application arises in simulation with haptic force feedback. Haptic devices apply forces on a rigid manipulandum (e.g., a stylus or laparoscope held in the user's hand) based on the motion of the manipulandum. Examples include the PHANToM from Sensable Technologies [20] and the Laparoscopic Impulse Engine from Immersion Corporation [8].

Since haptic devices exchange energy with the human hand, and not just information, stability is a major concern. An obvious way to use a haptic device held in a user's hand would be to impose displacement boundary conditions on a strand based on the position of the hand, and apply the computed forces to the hand. However, this obvious approach has several problems. Real haptic devices can generate relatively small forces and emulate only finite stiffnesses, which are actually quite small compared to the stiffnesses of real objects. Therefore one can easily overcome the maximum forces imposed by the haptic device to extend the strand beyond its maximum length. Also a directly coupled system like this could become unstable due to lags in the simulation. A popular way to overcome these difficulties is to use what has been dubbed a Virtual Coupling by Colgate and coworkers [7]. This was generalized to elastostatic contact simulation by James and Pai [11].

For strands, the real position of the hand can be coupled to the end of the strand through a generalized (3D) spring and damper. Thus the boundary conditions imposed on the strand due to the haptic device are forces and torques produced by this virtual coupling and not positions. The strand could be attached at the other end to the environment and therefore have the position and orientation satisfied. For example, in a laparoscopic simulation (e.g., see Fig. 1) the haptic device imposes forces at the needle end of the surgical suture, while the opposite end is inserted in human tissue. Interacting with human hair or a telephone cord modeled as a strand is similar. All these models reduce to the Standard BVP and therefore can be efficiently solved using the algorithm described in Sec. 3.

It is significantly more difficult to solve the ODEs if positions are specified at both ends, or at multiple points along the length of the strand. In this case we can use "shooting" techniques [3]. In single shooting, we search for the stresses ξ at one end such that the computed position E matches the boundary condition E_1. The strand integration effectively computes the nonlinear residual twist $\rho(\xi)$ extracted from $\mathsf{E}(\xi)^{-1}\mathsf{E}_1$, whose roots are found using Newton's method. Multiple boundary conditions are treated similarly, but result in a larger non-linear system. It may also be numerically advantageous to introduce intermediate nodes; this is the method of multiple shooting [3]. These are well studied techniques but nevertheless are significantly less easy to use and less robust than the simple strand model.

We have implemented the strand model described in Secs. 2 and 3 in Java, in the context of simulation of surgical sutures. The suture is attached rigidly at one end to the environment. We simulated both a straightforward strand model with the user imposing stress boundary conditions at the other end and the more complex boundary constraint where the user imposes the position and orientation of one end. In the latter case, we solve for the resulting position using Newton iterations as described above, with backtracking line search. The Jacobian matrix of the residual $\rho(\xi)$ is numerically estimated using several invocations of the strand algorithm; therefore efficient integration of the Standard BVP is important even for non-standard BVPs.

Fig. 1 shows a screen shot of the simulation and the accompanying video shows the resulting motions for the more difficult case where we specify the position and orientation at the needle. The simulation is extremely fast, easily simulating sutures discretized with hundreds of mesh points at 30Hz on 700MHz PIII computer, in Java.

5. Conclusions and Future Work

We have proposed a modeling primitive, called a *strand*, which is useful for modeling thin elastic objects in computer graphics. The physics of a strand is based on the theory of Cosserat rods. This results in a spatial ordinary differential equation that can be efficiently integrated for common boundary conditions and is therefore suitable for interactive applications in surgical simulation and games. In future work, we plan to develop fast collision detection algorithms for strands, and to incorporate contact constraints more explicitly. Other extensions briefly touched upon in this paper but which have not been described in detail due to space limitations include modeling with Cosserat shells and points, and incorporating time-stepping methods for dynamic strands. We also plan to explore other applications of strands in animation, including animation of vegetation, hair, and cloth.

Acknowledgments

The author would like to thank Paul Kry and Doug James for help with the animation and for helpful comments. The background image texture for Fig. 1 is reproduced with the

permission of Dr. E. M. Lichten. Our laboratory is supported in part by grants from IRIS and NSERC.

References

1. Ken-ichi Anjyo, Yoshiaki Usami, and Tsuneya Kurihara. A simple method for extracting the natural beauty of hair. *Computer Graphics (SIGGRAPH '92 Proceedings)*, 26(2):111–120, July 1992.
2. S. S. Antman. *Nonlinear Problems of Elasticity*. Springer-Verlag, 1995.
3. U. Ascher and L. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, Philadelphia, PA, 1998.
4. David Baraff. Linear-time simulation using Lagrange multipliers. In *SIGGRAPH 96 Conference Proceedings*, pages 137–146, 1996.
5. Ronen Barzel. Faking dynamics of ropes and springs. *IEEE Computer Graphics and Applications*, 17(3):31–39, May/June 1997.
6. R. W. Brockett. Robotic manipulators and the product of exponentials formula. *Mathematical Theory of Networks and Systems*, pages 120–129, 1984.
7. J. E. Colgate, M. C. Stanley, and J. M. Brown. Issues in haptic display of tool use. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 140–145, 1995.
8. Immersion Corp. Impulse engine, http://-www.immersion.com/products/custom-/laproimpulse.shtml.
9. Roy Featherstone. *Robot dynamics algorithms*. Kluwer, 1987.
10. Sunil Hadap and Nadia Magnenat-Thalmann. Modeling dynamic hair as a continuum. *Computer Graphics Forum*, 20(3), 2001. ISSN 1067-7055.
11. Doug L. James and Dinesh K. Pai. A unified treatment of elastostatic contact simulation for real time haptics. *Haptics-e, the Electronic Journal of Haptics Research*, 2(1), 2001. `http://www.haptics-e.org`.
12. T. Kim and U. Neumann. A thin shell volume for modeling human hair. In *Proceedings of the Conference on Computer Animation (CA-00)*, pages 104–111, Los Alamitos, CA, May 3–5 2000. IEEE Press.
13. C-L. Liu and N. C. Perkins. Complex spatial equilibria of u-joint supported cables under torque, thrust and self-weight. *Int. J. Solids Structures*, 30(3):271–285, 1995.
14. John Maddocks. http://lcvmwww.epfl.ch/.
15. Nadia Magnenat-Thalmann, Sunil Hadap, and Prem Kalra. State of the art in hair simulation. International Workshop on Human Modeling and Animation, Seoul, Korea, June 2000, Korea Computer Graphics Society, pp. 3-9.
16. R.S. Manning, J.H. Maddocks, and J.D. Kahn. A continuum rod model of sequence-dependent DNA structure. *J. Chemical Physics*, 105(1):5626–5646, 1996.
17. O'Reilly. On constitutive relations for elastic rods. *Int. J. Solids Structures*, 35(11):1009–1024, 1998.
18. D. Ota, B. Loftin, T. Saito, R. Lea, and J. Keller. Virtual reality in surgical education. *Computers in Biology and Medicine*, 25(2):127–137, 1995.
19. D. K. Pai, U. M. Ascher, and P. G. Kry. Forward dynamics algorithms for multibody chains and contact. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pages 857–863, San Francisco, April 2000.
20. PHANToM. http://www.sensable.com/.
21. G Rodriguez. Kalman filtering, smoothing and recursive robot arm forward and inverse dyna mics. *IEEE Journal of Robotics and Automation*, 3(6):624–639, 1987.
22. G. Rodriguez, A. Jain, and K. Kreutz-Delgado. A spatial operator algebra for manipulator modeling and control. *The International Journal of Robotics Research*, 10(4):371–381, 1991.
23. R. E. Rosenblum, W. E. Carlson, and E. Tripp, III. Simulating the structure and dynamics of human hair: modelling, rendering and animation. *J. Visualization and Comp. Animation*, 2(4):141–148, October–December 1991.
24. M. B. Rubin. *Cosserat Theories: Shells, Rods and Points*. Kluwer, 2000.