Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2012)
M. Lau (Posters Editor)

Activation Force Splines

Morten Engell-Nørregård , Kenny Erleben

Abstract

We present a method for simulating the active contraction of deformable models, usable for interactive animation of soft deformable objects. We present a novel physical principle as the governing equation for the coupling between the low dimensional 1D activation force model and the higher dimensional 2D/3D deformable model. Our activation splines are easy to set up and can be used for physics based animation of deformable models such as snake motion and locomotion of characters. Our approach generalises easily to both 2D and 3D simulations and is applicable in physics based games or animations due to its simplicity and very low computational cost.

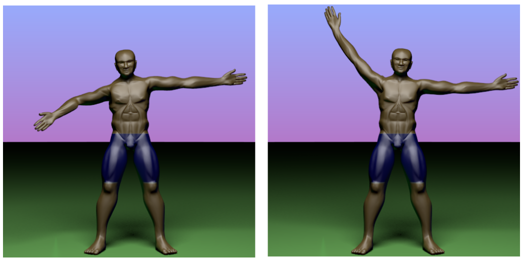

Figure 1: *Cartoon man flexing the arm. Using 5 activation splines the arm is lifted, lowered and bent.*

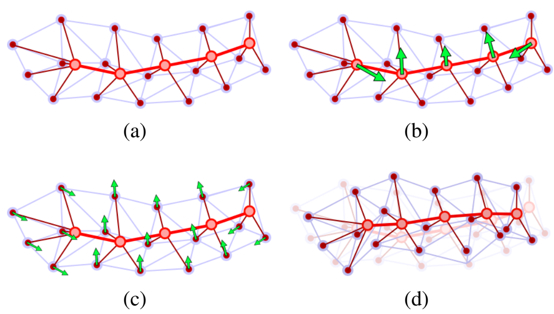

Figure 2: *Illustration of our pre-processing phase and simulation loop. As a pre-processing step (a) we bind a spline to a volume mesh. During the simulation loop we first compute spline forces (b), afterwards we transfer the forces to the volume mesh using neighbourhoods (c). Finally we update the deformable model and move the embedded spline (d).*

1. The Activation Spline Method

Our system consists of a passive deformable model represented by a volume mesh and one or more splines, used to represent the activation forces in the deformable model. Our approach may be compared to [LST09] But our focus is interactive performance so 1-4 min/frame is not viable in our context. The simulation loop for the method is sketched out in Figure 2. Without loss of generality, in the following, we will describe our method using a single spline. We assume that an animator or modeller have created a spline with K control points and any point on the spline can be found using

$$\mathbf{p}(s) = \sum_{k=1}^{K} N_k(s)\mathbf{g}_k \tag{1}$$

where $\mathbf{p} \in \mathbb{R}^D$ with $D = 2,3$ and $N_k : \mathbb{R} \mapsto \mathbb{R}_+$ is the k^{th} global basis function of the spline. The vector $\mathbf{g}_k \in \mathbb{R}^D$ is the corresponding control point and $s \in [0..L]$ is the spline parameter. The spline is assumed to be inside a volume mesh having V vertices, where $\mathbf{x}_j \in \mathbb{R}^D$ is the coordinates of the j^{th} vertex.

The governing equation of motion for any deformable model can be written abstractly as

$$\mathbf{M}\frac{d^2\mathbf{x}}{dt^2} + \mathbf{C}\frac{d\mathbf{x}}{dt} + \mathbf{k}(\mathbf{x}-\mathbf{x}_0) = \mathbf{F}_{\text{ext}} \tag{2}$$

where $\mathbf{M}$ is a mass matrix, $\mathbf{C}$ is a damping matrix and $\mathbf{k}$ is the elastic forces that depends on the current displacement field $\mathbf{x}-\mathbf{x}_0$. For linear elastic materials this is $\mathbf{k} = \mathbf{K}(\mathbf{x}-\mathbf{x}_0)$ where $\mathbf{K}$ is the stiffness matrix. The vectors $\mathbf{x}$ and $\mathbf{x}_0$ are the concatenation of the current and initial (rest) mesh vertex positions respectively. Thus, $\mathbf{x}, \mathbf{x}_0, \mathbf{k} \in \mathbb{R}^{DV}$ and $\mathbf{M}, \mathbf{C}, \mathbf{K} \in \mathbb{R}^{DV \times DV}$. [ESHD05] During a simulation, $\mathbf{k}$ is the elastic passive forces coming from the deformable model itself. The activation spline forces will be coupled to the model by adding a right hand side force term, $\mathbf{F}_{\text{act}} \in \mathbb{R}^{DV}$. The resulting equations of motion to be time integrated is written as

$$\mathbf{M}\frac{d^2\mathbf{x}}{dt^2} + \mathbf{C}\frac{d\mathbf{x}}{dt} + \mathbf{k}(\mathbf{x} - \mathbf{x}_0) = \mathbf{F}_{\text{ext}} + \mathbf{F}_{\text{act}} \tag{3}$$

We compute the spline forces at the discrete spline points.

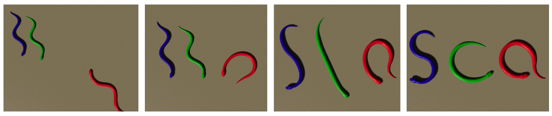

Figure 3: *A time lapse of snakes spelling "SCA". Even though the animation is based on a physical simulation, explicit control of the poses of the bodies can be obtained.*

This is done by creating a linear damped spring for each discrete segment of a spline. Let $\mathbf{e}_i \in \mathbb{R}^D$ be the unit direction vector from $\mathbf{x}_i$ to $\mathbf{x}_{i+1}$ then the spring force on $\mathbf{x}_{i+1}$ is

$$\mathbf{a}^i_{i+1} = -k_i\left(l_i - \alpha_i l_{0i}\right)\mathbf{e}_i - c_i\mathbf{e}_i\mathbf{e}_i^T\left(\mathbf{v}_{i+1} - \mathbf{v}_i\right) \tag{4}$$

and on $\mathbf{x}_i$ we have from Newton's third law of motion

$$\mathbf{a}^i_i = -\mathbf{a}^i_{i+1} \tag{5}$$

where the spline point velocity is given by $\mathbf{v}_i = \frac{d}{dt}\mathbf{p}_i$ for all i. Further, $k_i, c_i \in \mathbb{R}_+$ are the stiffness and damping coefficients respectively. Here l_i is the current length and l_{0i} is the initial rest length pre-computed during the binding process.
We have extended the standard spring damper law with the parameter $\alpha_i \in \mathbb{R}_+$. This is the parameter we use to control contraction of the spline. When $0 < \alpha_i < 1$ then the i^{th} segment of our spline contracts. We can now compute the spline force at the i^{th} spline point as

$$\mathbf{f}_i = \mathbf{a}^{i-1}_i + \mathbf{a}^i_i \tag{6}$$

If a spline force is wanted for an arbitrary s-value the spline basis functions are used to interpolate the wanted value.
We introduce a force equivalence principle between a force on a spline point $\mathbf{p}(s)$ and the volume integral of the activation force $\mathbf{F}(\mathbf{x})$ in a neighbourhood $\mathcal{N} \subset \mathbb{R}^D$ around the spline point,

$$\mathbf{f}(s) = \frac{1}{V_{\mathcal{N}(\mathbf{p}(s))}} \int_{\mathbf{x} \in \mathcal{N}(\mathbf{p}(s))} \mathbf{F}(\mathbf{x})\, dV. \tag{7}$$

where $V_{\mathcal{N}(\mathbf{p}(s))}$ is the total volume of the neighbourhood $\mathcal{N}(\mathbf{p}(s))$. Thus, $\mathbf{f}(s)$can be interpreted as the average force over the entire neighbourhood.
We use this simple principle to create a mapping between activation forces at discrete spline points and vertices in the ambient mesh, where the spline is embedded.

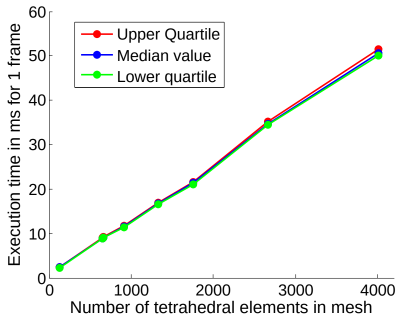

Figure 4: *The timings for contraction of a single tetrahedral mesh cylinder with varying mesh resolution. a single spline was fitted to the mesh and contracted for 1000 frames. The plot shows the median and upper and lower quartiles for different mesh resolutions*

2. Validation

To validate the method, we have implemented the method in Matlab and in C++.
We performed several tests with different mesh resolutions, using the c++ framework. We used a tetrahedral mesh cylinder with 1 embedded spline. The results can be seen in Figure 4. The timings shown are for a full simulation pass including visualisation. The spline activation is app. $0.5 - 1.0$ percent of the full simulation time and ranges from 0.02 to 0.3 ms for the shown mesh resolutions.
As an example of the simulation, we let three snakes crawl and deform to spell three letters, Figure 3 shows the result.
We also showcase a human bending his arm. For this simulation we used 5 splines in the arm and shoulder as well as a simple skeleton embedded in a mesh of app. 7000 tetrahedral elements. Figure 1 shows two frames from this simulation.

3. Conclusion

We have presented a method for modelling active contraction forces of deformable models, based on splines.
Our method presents an attractive compromise between precision and realism on the one hand, and speed and generality on the other. Actuation splines are an intuitive way of extending the simple line-actuator to a deformable contraction spline, using physics based principles.

References

[ESHD05] Erleben K., Sporring J., Henriksen K., Dohlman H.: *Physics-based Animation (Graphics Series)*. Charles River Media, Inc., Rockland, MA, USA, 2005. 1

[LST09] Lee S.-H., Sifakis E., Terzopoulos D.: Comprehensive biomechanical modeling and simulation of the upper body. *ACM Trans. Graph. 28* (September 2009), 99:1–99:17. 1