Technical Report
(Soft-Bodies)

Soft-Bodies

Spatially Coupled Shells

Benjamin Kenwright

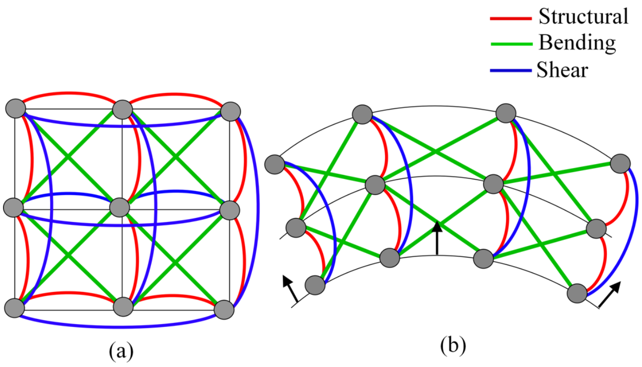

Fig. 1 Surface and Layers - Illustrating the coupled spatial connection of neighbouring constraints along the surface and parallel 'virtually' projected shells. Where (a) shows a single connected thin layer and (b) the interconnection of virtual layers to add local rigidity to the mesh. Creating 'virtual' shells parallel to surface and coupling them using distance constraints, allows us to create a self-supporting soft-body mesh.

Abstract We present a novel soft-body framework based upon the structural coupling of virtual shells. Our concept creates an effective solution that solves the problem for self-supporting thin-surface soft-body meshes. Structural constraints in combination with virtual layers allow us to simulate a responsive, aesthetically pleasing, smooth soft-body system. Our physically-based simulation framework is able to show significant characteristics, such as, jiggling and rippling behaviour, while remaining stable and usable. We demonstrate our technique using a variety of graphical meshes, which were simulated in real or near real-time.

Keywords soft-bodies, real-time, shells, cloth, coupled, springs, shear, structural, bending

Benjamin Kenwright

1 Introduction

This paper presents a novel technique for creating stable interactive soft-body effect. The field of soft-body dynamics is diverse, including the simulation of soft organic materials, such as muscle, fat, hair and vegetation, as well as other deformable materials, such as, clothing and fabric.

A thin planar surface of point-masses connected by constraints is unable to support itself under external forces (e.g., staying horizontal under the influence of gravity). While Grinspun et al. [11] solved the rigid thin surface problem by including angular constraints between adjacent triangles in additional to distance constraints, we solve the problem by adding additional 'virtual' shells to thin surface. The 'virtual' shells and thin surface together form a self-supporting structure. Although the concept itself is simple, the method provides a novel solution for the design and construction of thin layer soft-body systems.

Our method is able to handle structures that have loops and branches by providing regional rigidity across the surfaces while allowing for smooth deformations, even on low-dimensional coarse meshes. Our model does not require any off-line processing or overhead large matrix based solvers. The model is suited to non-closed shapes (e.g., vases or tubes which have a thin outer surface) and no internal outward pressure to keep the shape rigid (i.e., analogous to a balloon's internal pressure pushing outwards on the surface to counteract deformations).

Contribution The key contributions of this paper are: (1) as far as we are aware we are the first to present the concept of expanding the cloth analogy for structural, sheer, and bending, from a surface analogy to a virtually coupled shell system; (2) our model is able to handle complex structures that has loops and branches; (3) combined with a iterative position-based integrator our method allows for a very stable and practical solution; (4) the nature of our framework means it is highly suited to massively parallel execution environments, such as, the GPU; (5) we require no off-line preprocessing and the deformations are smooth even for coarse meshes.

2 Related Work

Soft-body systems is a popular multi-discipline topic (e.g., graphics, robotics, animation [20], and medical [5]). Many methods and models have been proposed to simulate deformable bodies, such as, (1) implicit surfaces [7], (2) finite difference approaches [23], (2) mass-spring systems [9,1], (3) the Boundary Element Method (BEM) [13], (4) the Finite Element Method (FEM) [18,17,6], (5) the Finite Volume Method (FVM) [22], (6) mesh-free particle systems [20,25,8]. While specific methods target accurate scientific/engineering simulations, we focus on provide visually plausible emulations. For example, the finite element method (FEM) targets accurate representations of the stress/decomposition of a model compared to more approximate systems, such as, a mass-spring method.

The inspiring work by Grinspun et al. [11], presented a single outer membrane method which connected the surface using distance constraints for the edges in conjunction with angular constraints for the faces to synthesize a soft-body mesh. Our method uses only distance constraints and attempts to account for the surfaces finite thickness by adding 'virtual' shells to add structural support (see Figure 3).

3 Method

Our approach builds upon a popular penalty-based cloth-spring simulation concept. We use a set of interconnected point-masses to represent the physical shape of the mesh. A planar mesh surface is connected using three types of spring, i.e., a bend, a structural, and a shear spring, to synthesize an aesthetically pleasing effect (i.e., smooth responsive deformations). The internal forces from the springs in combination with the topological configuration provide an interactive solution that is able to react to external forces, like gravity and collisions, in a realistic manner.

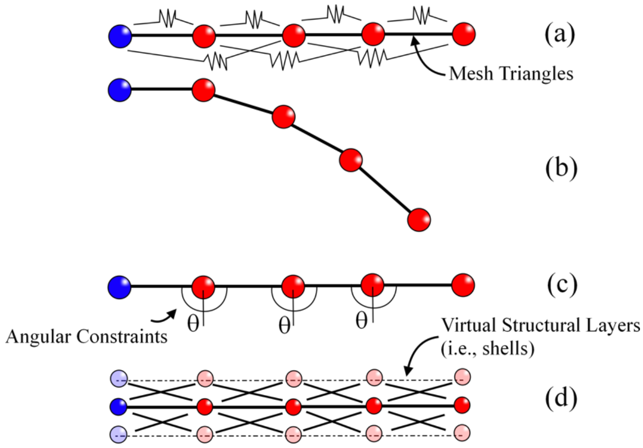

Fig. 3 Thin Surfaces Concept - Illustrating the problem this paper addresses and solves. (a) a thin surface can be connected by any number of distance constraints, however, due to the constraints being parallel, the surface will be unable to keep its rigidity; (b) simple example of a flat surface attached to a wall, the surface will bend under the influence of gravity; (c) Grinspun et al. [11] presented a solution to rectify the problem by adding angular springs to neighbouring faces; (d) our approach solves the problem by adding 'virtual' parallel layers to the surface that can be connected to provide rigidity to the mesh so it can support itself.

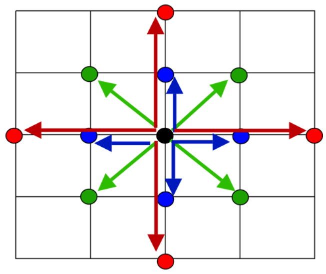

Fig. 4 Point-Mass Coupling - Spatial coupling of neighbouring constraints along the surface axis, which we can also apply to the vertically projected shells.

3.1 Dynamics

We define general constraints via a constraint function ([24,19,1]). Instead of computing forces as the derivative of a constraint function energy, we directly solve for the equilibrium configuration and project positions. With our method we derive a bending term for the material which uses a point based approach similar to the one proposed by Grinspun et al. [11] and Bridson et al. [3].

Position-based dynamics have been used for a variety of systems. For example, Jakobsen [12] built his physics engine (called Fysix) on a position-based approach. With the central idea of using a Verlet integrator to manipulate positions directly. The velocities are implicitly stored by the current and the previous positions of the point-masses. The constraints are en-

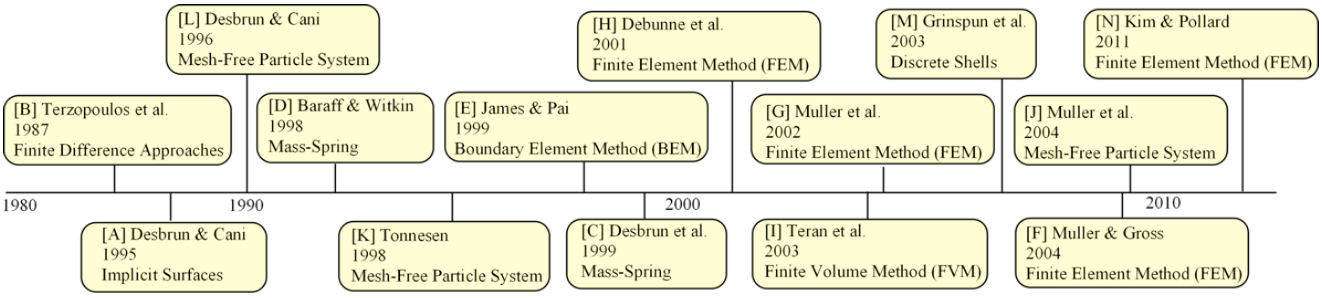

Fig. 2 **Timeline** - Graphical illustration of key soft-body systems over the duration of the past few decades. [A] [7], [B] [23], [C] [9], [D] [1], [E] [13], [F] [18], [G] [17], [H] [6], [I] [22], [J] [20] [K] [25], [L] [8], [M] [11], [N] [15].

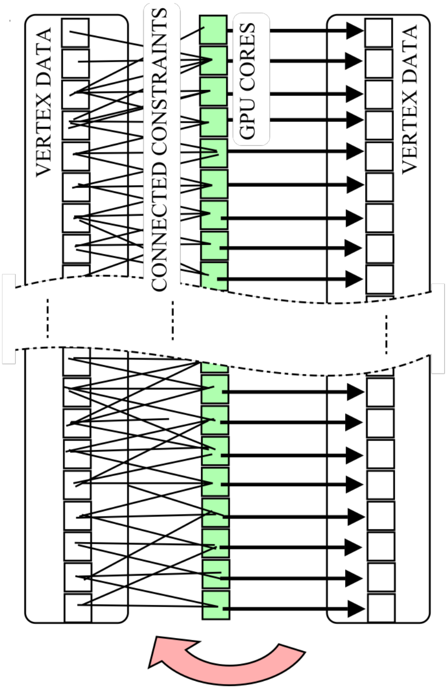

Fig. 5 **Ping-Pong** - Illustrating parallelism of our approach using a massively parallel execution environment (i.e., GPU and CUDA). We have two sets of point-mass data that we ping-pong back and forth to iteratively solve the distance constraints. Each point-mass knows which neighbouring point-masses it is connected. Hence, we can calculated each point-mass penalty error in parallel. We have two sets of data so that all the threads read from a fixed set of data (i.e., does not matter about the order of update).

forced by direct manipulation of the positions. Jakobsen demonstrated the efficient representation fo distance constraints that could be used to form the underpinnings of a stable and iterative control mesh. In this paper, we use position-based constraints for the coupled connection of point-masses and for the interaction with the environment. An important note is our model is decomposed of point masses and does not need to account for any angular calculations. Position-based methods have proven themselves an efficient and robust method in variety of soft body systems, such as, cloth [19], character animation [12], and fluid dynamics [16]. For a detailed introduction to position-based dynamics, we refer the reader to the interesting work of Müller [19] and Jakobsen [12].

Iterative An iterative methods automatically adjusts to errors during the development of the solution. If a solution cannot be found within a predefined time, we can use the best guess solution the iterative solver has found. Additionally, if we reach an acceptable solution earlier (i.e., within the pre-defined number of updates), we can skip further iterations. This concept is crucial in time-limited system, such as, video games and training simulations, since we can limit the computational time for a solution. Allowing us to trade quality for computational time.

3.2 Scalability

Low-Resolution Control Mesh In an endeavour to keep soft-body system interactive, we explored multi-resolution control meshes. As high resolution graphical meshes may posses vast number of polygons. Even with the enormous power of the GPU it can be difficult to create soft-body motions for large numbers of high-poly graphical meshes in real-time. However, we combine our approach with a control mesh. The high-poly graphical mesh has an associated low-dimensional physics mesh. We apply or soft-body framework to the coarse mesh correspondingly deforms the high-resolution graphical mesh. The control mesh technique reduces the computational overhead and enables us to create a solution that can achieve real-time frame-rates, even for high poly geometry. Additionally, it helps reduce the cost of computing collision detection information in virtual environments, with other bodies and the terrain.

Model Reduction & Mesh Embedding The two main techniques for reducing the complexity of a system can be classified into *modal reduction* and *mesh embedding*.

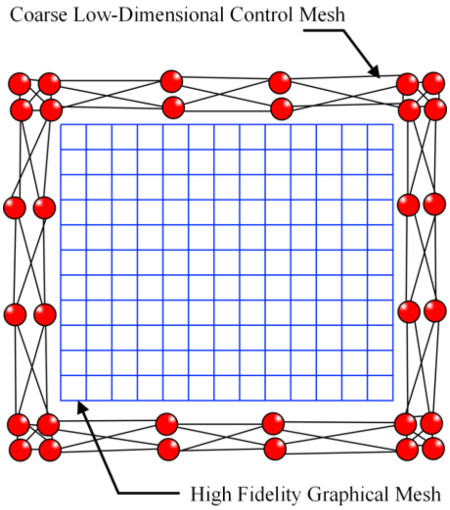

Fig. 6 Mesh Embedding - A 2D illustration of the low-dimensional deformable body driven system. The course volumetric mesh encloses the detailed graphical surface.

Modal reduction is a popular method for reducing the complexity of a finite element systems by using a linear subspace to span a small number of displacement basis vectors to represent the deformation in the body. The eigenmodes obtained from linear modal analysis would be the best basis vectors for small deformation. For large deformation, however, they are not sufficient to capture the non-linear deformation characteristics, so multiple techniques have been suggested to choose a good deformation basis set [2]. Model techniques have successfully been used for a number of real-time solutions, such as, surgery simulators and hand-soft body interaction.

Mesh embedding, which is also called *free-form* deformation [15], uses a low-dimensional coarse mesh to influence and control the high poly model. The location of the attached points for the deformable body is determined by interpolating the positions of the neighboring nodes in the mesh. Since the work by Faloutsos et al. [10], mesh embedding techniques have been widely used to simulate soft bodies in the graphics literature [21,14, 4] We used mesh embedding to reduce complexity of the deformable body in our simulation system because the technique enables us to easily reduce the computational overhead in scalable fashion without sacrificing the responsiveness of the underlying physical model. In our method, the control body (i.e., low-poly deformation mesh) is connected to the high-poly mesh using a rigid set interconnected elements that can be updated on the GPU. The mesh-embedded system then consists of a set of a deformable body that controls in a puppet like fashion the fine grained graphical mesh (see Figure 6).

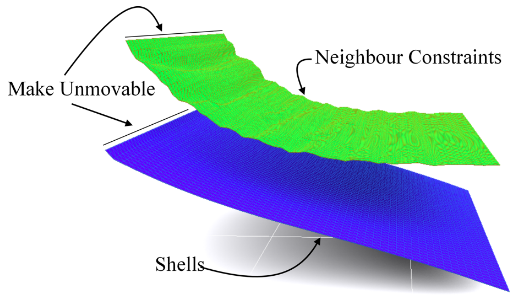

Fig. 7 Thin Mesh - An uncomplicated test case to demonstrate our approach. (top) Coupling only neighbours creates a cloth-like result that can twist and bend (i.e., no matter how many distance constraints are used). The only other way of making the mesh rigid is by adding angular constraints. (bottom) However, adding 'virtual' shells to the mesh constraints produces a more rigid solution that resists bending and twisting and holds its original shape.

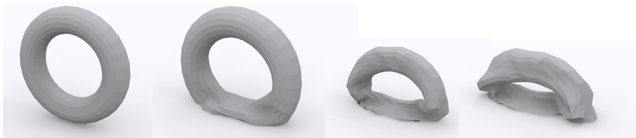

Fig. 8 Coupled-Vertices - Shows the simulation of a soft-body system connected using only the graphical mesh (i.e., vertices and triangle edges). The mesh is unable to support itself and the deformations are rough and coarse with abrupt and sharp edges.

4 Experimental Results

The results of our method are shown in Figure 7, 8, 9, 10, 12, and 11. In practice, different numbers of shell constraints can be added to create the desired visual effect. Additionally, combining technique with a model reduction control methodology in conjunction with the computational power of the GPU enables the solution to run at real-time or near-real time frame-rates.

5 Conclusion

We have presented a novel system for constructing self supporting soft-body meshes using an uncomplicated and effective methodology. Our model is able to handle complex structures that include loops and branches. The algorithm is robust and can be combined with techniques, such as, mesh embedding, to provide a solution for real-time environments, such as, games.

Acknowledgements

We would like to thank all the reviewers for taking the time out of their busy schedules to provide insightful and valuable comments to help to improve the quality of this paper.

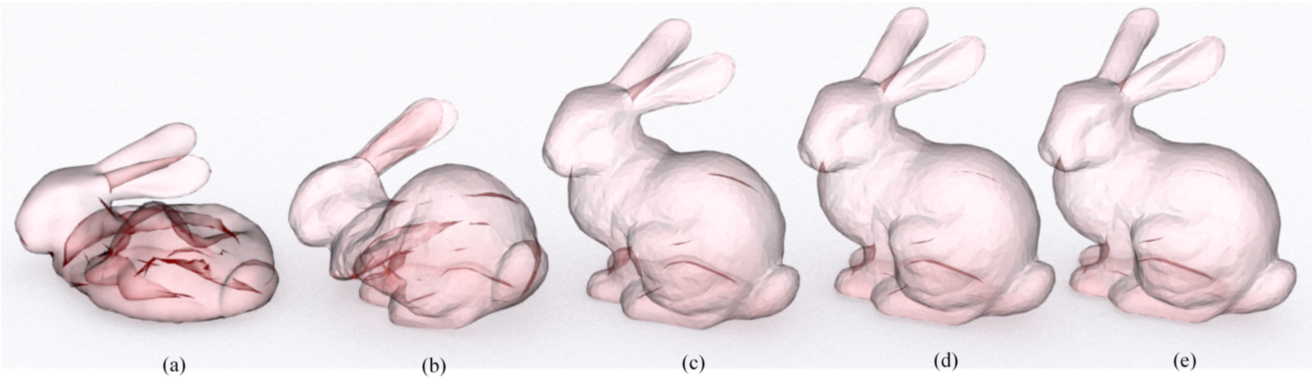

Fig. 10 Shell Numbers - Varying the number of coupled shells produces a more rigid solution (i.e., at the expense of more computations).

Fig. 11 Cow Soft-Body - The simulation of a cow shape with long thin legs. The model bends and deforms before coming to rest on the ground.

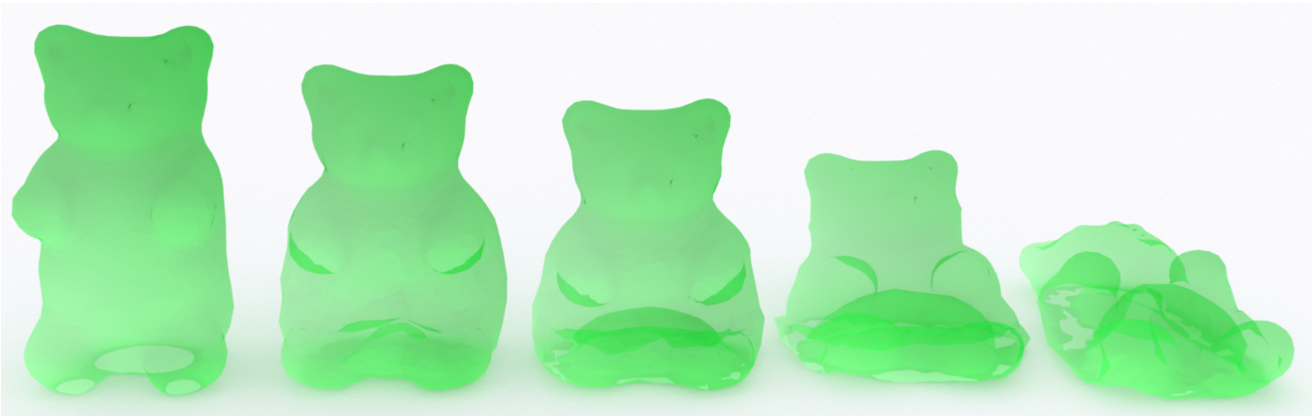

Fig. 12 Gummybear Soft-Body - The simulation of a gummybear shape falling and bending before rolling over onto its back coming to rest.

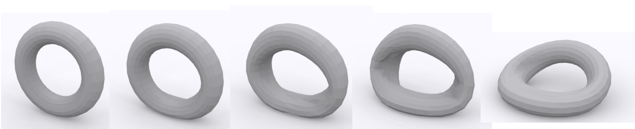

Fig. 9 Couple-Shells - Adding coupled shells enables us to construct a self-supporting soft-body structure compared to the basic system shown in Figure 8.

References

1. D. Baraff and A. Witkin. Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 43–54. ACM, 1998.

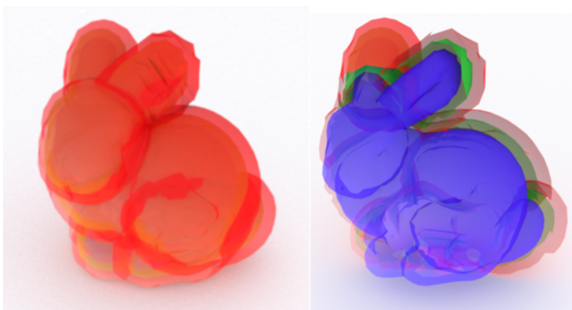

Fig. 13 Rabbit Shells - Visualizing three structural shells for a rabbit mesh (right) shows the shells cut away.

2. J. Barbič and D. L. James. Real-time subspace integration for st. venant-kirchhoff deformable models. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 982–990. ACM, 2005.
3. R. Bridson, S. Marino, and R. Fedkiw. Simulation of clothing with folds and wrinkles. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 28–36. Eurographics Association, 2003.
4. S. Capell, M. Burkhart, B. Curless, T. Duchamp, and Z. Popović. Physically based rigging for deformable characters. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 301–310. ACM, 2005.
5. S. Cotin, H. Delingette, and N. Ayache. Real-time elastic deformations of soft tissues for surgery simulation. *Visualization and Computer Graphics, IEEE Transactions on*, 5(1):62–73, 1999.
6. G. Debunne, M. Desbrun, M.-P. Cani, and A. H. Barr. Dynamic real-time deformations using space & time adaptive sampling. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 31–36. ACM, 2001.
7. M. Desbrun and M.-P. Gascuel. Animating soft substances with implicit surfaces. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 287–290. ACM, 1995.
8. M. Desbrun and M.-P. Gascuel. *Smoothed particles: A new paradigm for animating highly deformable bodies.* Springer, 1996.
9. M. Desbrun, P. Schröder, and A. Barr. Interactive animation of structured deformable objects. In *Graphics Interface*, volume 99, page 10. Citeseer, 1999.
10. P. Faloutsos, M. Van De Panne, and D. Terzopoulos. Dynamic free-form deformations for animation synthesis. *Visualization and Computer Graphics, IEEE Transactions on*, 3(3):201–214, 1997.
11. E. Grinspun, A. N. Hirani, M. Desbrun, and P. Schröder. Discrete shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 62–67. Eurographics Association, 2003.
12. T. Jakobsen. Advanced character physics. In *Game Developers Conference*, pages 383–401, 2001.
13. D. L. James and D. K. Pai. Artdefo: accurate real time deformable objects. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 65–72. ACM Press/Addison-Wesley Publishing Co., 1999.
14. L. Kharevych, P. Mullen, H. Owhadi, and M. Desbrun. Numerical coarsening of inhomogeneous elastic materials. In *ACM Transactions on Graphics (TOG)*, volume 28, page 51. ACM, 2009.
15. J. Kim and N. S. Pollard. Fast simulation of skeleton-driven deformable body characters. *ACM Transactions on Graphics (TOG)*, 30(5):121, 2011.
16. M. Macklin and M. Müller. Position based fluids. *ACM Transactions on Graphics (TOG)*, 32(4):104, 2013.
17. M. Müller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler. Stable real-time deformations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 49–54. ACM, 2002.
18. M. Müller and M. Gross. Interactive virtual materials. In *Proceedings of Graphics Interface 2004*, pages 239–246. Canadian Human-Computer Communications Society, 2004.
19. M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, 2007.
20. M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross, and M. Alexa. Point based animation of elastic, plastic and melting objects. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 141–151. Eurographics Association, 2004.
21. M. Nesme, P. G. Kry, L. Jeřábková, and F. Faure. Preserving topology and elasticity for embedded deformable models. In *ACM Transactions on Graphics (TOG)*, volume 28, page 52. ACM, 2009.
22. J. Teran, S. Blemker, V. Hing, and R. Fedkiw. Finite volume methods for the simulation of skeletal muscle. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 68–74. Eurographics Association, 2003.
23. D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. In *ACM Siggraph Computer Graphics*, volume 21, pages 205–214. ACM, 1987.
24. M. Teschner, B. Heidelberger, M. Muller, and M. Gross. A versatile and robust model for geometrically complex deformable solids. In *Computer Graphics International, 2004. Proceedings*, pages 312–319. IEEE, 2004.
25. D. L. Tonnesen. *Dynamically coupled particle systems for geometric modeling, reconstruction, and animation.* PhD thesis, University of Toronto, 1998.