Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS (2015)
F. Jaillet, G. Zachmann, and F. Zara (Editors)

Area Preserving Strain Limiting

Dongsoo Han

Advanced Micro Devices

Abstract

In this paper we present a novel fast strain-limiting method that allows cloth to preserve its surface area efficiently. By preserving triangle area rather than edge length as other approaches do, this method does not remove the degrees of freedom of triangles and does not suffer from locking. Borrowing ideas from fluid simulation, we define pressures in each triangle and solve the global linear equation which shows a faster convergence over prior approaches which use Gauss-Seidel-like iterations. The linear equation is easy to build by using edge and normal vectors and can be solved using Conjugate Gradient solver with regularization which not only helps the solver converge fast but also allows users to have a control over the stretchiness of cloth materials. Our area preserving strain limiting (APSL) can be also used as stand-alone cloth solver with linear bending springs.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation I.6.8 [Simulation and Modeling]: Types of Simulation—Animation

1. Introduction

Textile materials have a wide range of properties in terms of extensibility from almost inextensible woven cotton to stretchable wool weaves. To deal with these various materials, researchers have modeled them with various approaches such as stiff springs or inextensible edge length constraints.

With stiff springs, it is possible to simulate a broad range of materials but it has shown unnecessary elongations or visually unpleasant over-stretching when the material is supposed to be almost inextensible. To overcome this problem, various strain-limiting methods were developed to enforce the edge length constraints.

In case the simulating primitive is a triangle, the triangular mesh could suffer from locking when edge-based strain limiting is applied too tightly. Basically, the locking is caused by losing the degrees of freedom (DOF) of triangle primitives in a mesh and eventually removes in-plane deformations.

Motivated by this fact, we present a novel approach to preserve surface area. The idea was borrowed from fluid simulation as we define hydrostatic pressures and create a global linear equation. To construct the linear system, we only need to know the edge lengths and normal vectors. The understanding of the system is rather geometric so it is easy to apply it to the existing simulators. By using regularization, we can conveniently use an off-the-shelf linear solver such as Conjugate Gradient and it shows a fast convergence over other Gauss-Seidel-like approaches. Also the regularization value gives us an ability to control how much the area should be preserved. Therefore we can simulate wider range of materials.

2. Related Work

Cloth simulation has a long history of research and many methods have been developed. Basically we can categorize it into three methods : stiff spring, continuum-based and constraint methods.

Baraff and Witkin [BW98] developed implicit integration method for stiff springs and could achieve large timesteps with high stiffness. By using springs for stretch, shear and bending forces, various textile materials could be simulated but it was difficult to handle inextensible materials. To overcome this problem, Bridson et al. [BMF03] used strain-limiting as a post-processing process after the main simulation solver. Their method is similar to Position-Based Dynamics (PBD) [MHHR07] by Müller et al. as solving the stretched edges one by one iteratively.

Continuum-based method can simulate various cloth physics properties more accurately on the discrete domain but it also suffers from the over-stretching problem as spring-based method. Thomaszewski et al. developed Continuum-based Strain Limiting [TPS09] which solves individual

DOI: 10.2312/vriphys.20151340

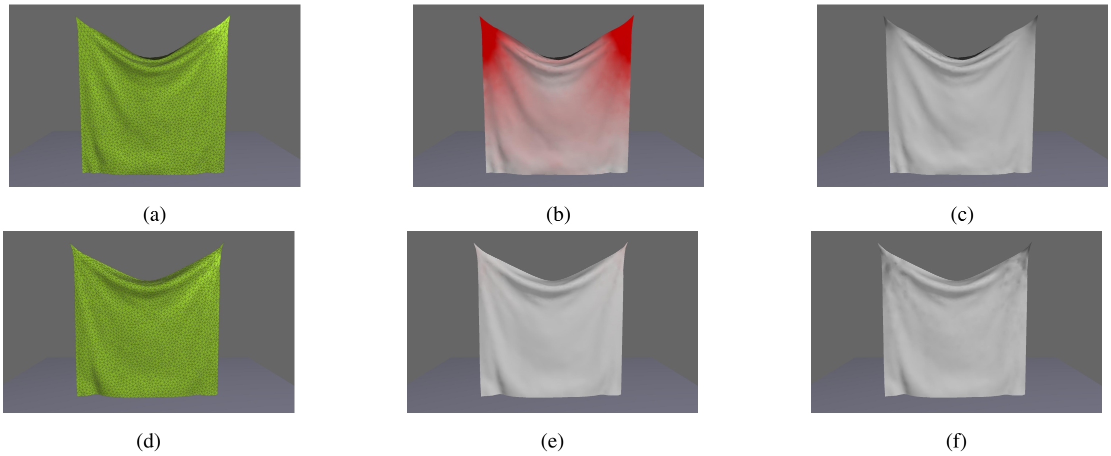


Figure 1: (a) is a simulated cloth model using implicit integration without strain limiting. (b) shows area change rate where the maximum rate is 96.3%. Pure red means the area change is above 10%. (c) shows normalized area change rate by maximum rate in the same cloth. (d) is simulated using implicit integration with area-preserving strain limiting. (e) shows area change rate where the maximum rate is 1.94%. (f) shows normalized area change rate. (a) and (d) were captured at the same frame.

strain problem per triangle and combines them in an iterative manner. By the nature of iterative solution, their method could have a slower convergence compared to a global enforcement such as the Fast Projection method by Goldenthal et al. [GHF*07].

Constraint-based approach does not require strain-limiting since it achieves inextensibility effectively. House et al. [HDB96] used constraint dynamics and introduced hierarchical approach for a fast convergence. Goldenthal et al. [GHF*07] introduced the Fast Projection method which used a direct solver to apply global enforcement on inextensible materials. Their method could achieve high enforcement but could have a limitation in the size of cloth mesh due to the high memory consumption of the dense direct solver. Also they did not provide a way to control the stiffness except checking the edge lengths after completing solving the linear system. They used a quad dominant mesh in order to avoid locking problem.

Position-Based Dynamics also uses constrained dynamics but instead formulating large linear equations, it solves individual constraint problem iteratively. Due to the slow convergence of Jacobi or Gauss-Seidel style iteration, PBD often takes long time to reach target inextensibility. To speed up the convergence, hierarchical mesh structure was developed as in [Mül08].

Müller et al. introduced Strain Based Dynamics in [MCKM14]. In their research, they used volume and area conversation as constraints. Unlike their approach, we use pressures to enforce area preservation.

Locking problem has been also researched for years. In triangular or tetrahedral mesh, strongly applied edge length based strain-limiting can cause losing DOF and lead to unnaturally stiff animation. Bender et al. [BDB11] proposed to use nonconforming model for locking-free simulation. Irving et al. [ISF07] used volume conservation for locking-free FEM simulation. Their approach has a similarity with our area preservation but ours uses geometric understanding to formulate the linear equation rather than relying on divergence theorem.

In terms of linear solver, various approaches have been used for cloth simulation and strain-limiting. Jacobi or Gauss-Seidel style iterative solvers have been widely used for PBD solver. Direct solver also has been used to solve over-constrained and possibly ill-conditioned linear equation as in [GHF*07]. To exploit the simple topology information, tridiagonal solver was used for inextensible hair simulation in [HH13].

Unlike a direct solver, Conjugate Gradient (CG) solver has been a popular choice but it could fail to solve if the linear system is over-constrained or ill-conditioned. In our approach, we use regularization to relax the linear system to use CG. This kind of regularization is also known in rigid body simulation community as Constraint Force Mixing (CFM) as in [Smi06]. With this, the linear system can be softened or damped so that CG can converge successfully. Also by modifying the regularization value, it is possible to control the amount of area preservation rate which is beneficial to the artists.

3. Formulation

As Figure 2 depicts, we define pressures at the barycenter of each triangle. With them, we define a vertex i's position displacement $\delta x_{i,a}$ by pressure p_a as below where $n_a = e_{ij} \times e_{jk}$, w_i is a inverse mass of vertex i and Δt is a timestep.

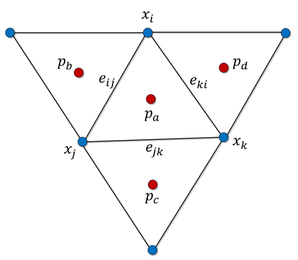


Figure 2: x_i is a vertex indexed i and p_a is pressure defined in a triangle indexed a. $e_{ij} = x_j - x_i$. n_a is a normal vector of the triangle a.

$$\delta x_{i,a} = w_i \left|e_{jk}\right| p_a \frac{n_a \times e_{jk}}{|n_a \times e_{jk}|} \Delta t^2 = w_i p_a \left(n_a \times e_{jk}\right) \Delta t^2 \quad (1)$$

This formulation is based on a simple observation that the vertex gets a pressure force proportional to the length of corresponding edge length and the direction is perpendicular to the edge. This idea is similar to the one found in fluid simulation research where Eulerian grid domain is irregular mesh as in [KFCO06], [FOK05], [CFL*07] and [KPNS10].

$\delta x_{i,c}$ is a vertex i's position displacement by pressure p_c which is adjacent to triangle a. To compute $\delta x_{i,c}$, we convert pressure p_c in the space of triangle a by being weighted by inner product of normal vectors. $\delta x_{i,c}$ will be zero if triangle a and c are not adjacent.

$$\begin{aligned} \delta x_{i,c} &= w_i \left|e_{jk}\right| (n_a \cdot n_c) p_c \frac{n_a \times e_{jk}}{|n_a \times e_{jk}|} \Delta t^2 \\ &= w_i (n_a \cdot n_c) p_c \left(n_a \times e_{jk}\right) \Delta t^2 \end{aligned} \quad (2)$$

And the area change of triangle a due to pressure p_a can be defined as below.

$$\Delta A_{a,a} = \tfrac{1}{2} \delta x_{i,a} \times e_{jk} = \tfrac{1}{2} w_i p_a |e_{jk}|^2 \Delta t^2 \quad (3)$$

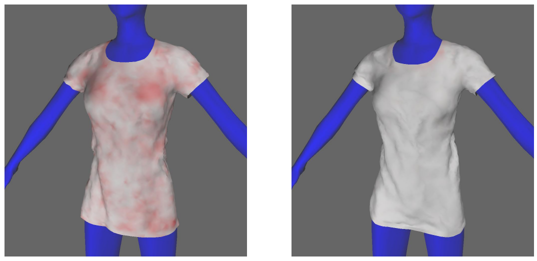
Triangle's area also gets changed by its neighboring pressures. $\Delta A_{a,c}$ is a area change of triangle a due to pressure p_c and can be defined as below.

$$\Delta A_{a,c} = \tfrac{1}{2} \delta x_{i,c} \times e_{jk} = \tfrac{1}{2} w_i (n_a \cdot n_c) p_c |e_{jk}|^2 \Delta t^2 \quad (4)$$

If triangle a and c are not adjacent, $\Delta A_{a,c}$ is zero.

Now we can define the area change of triangle a by all influencing pressures as below.

$$\begin{aligned} \Delta A_a &= \Delta A_{a,a} + \textstyle\sum_{b \in S} \Delta A_{a,b} \\ &= A_a^n - A_a^0 \end{aligned} \quad (5)$$



(a) (b)

Figure 3: Comparing area change rates. (a) is implicit solver only. (b) is implicit solver with APSL.

A_a^n is area of triangle a with the predicted vertex positions at the current timestep and A_a^0 is rest area. S is a set of triangle indices which are directly adjacent to triangle a. Due to the connectivity of triangle mesh, we can create a linear equation $\mathbf{Gp} = \mathbf{b}$ where the matrix $\mathbf{G}$ is sparse and symmetric and $\mathbf{p}$ is a unknown vector of pressures. $\mathbf{b}$ is a vector of current area changes from the rest state.

4. Solving a Linear System

In the linear system, the matrix $\mathbf{G}$ is sparse due to the triangle mesh topology. However the linear system can be overconstrained or ill-formed. It may be still possible to solve it using a direct solver such as PARDISO [Sch06], but it is unattractive because the linear system becomes quite large easily in the practical simulation scenario. Therefore we apply regularization by multiplying positive number $\alpha > 1$ to the diagonal elements of the matrix $\mathbf{G}$. This regularization has been widely used in various compute graphics areas such as cloth simulation [KGBS11], rigid body simulation [Smi06] and mesh processing [SCOIT05]. Please refer to [PL07] for details about regularization in compute graphics.

In our method, the solver not only converges fast, but also we can control how much we enforce the area preservation with regularization. By increasing the regularization value α, the area preservation gets applied loosely and the cloth mesh becomes stretchy as a result. In case of enforcing tight area preservation, we chose to use multiple sub-steps because the overall cost to solve multiple linear systems with a large regularization value is cheaper than one linear system with a small regularization value. As Figure 1 shows, we could achieve tight area preservation in less than 2%. Thanks to regularization, the each CG took average 3 iterations and we used 10 sub-steps for APSL after one implicit integration.

For simplicity, we used a fixed value for α but it is possible to choose it adaptively based on the result of CG solver. If the solver fails to converge or takes too much iterations, we can increase α for the next iteration. If the CG solver converges within a few iterations, we restore it.

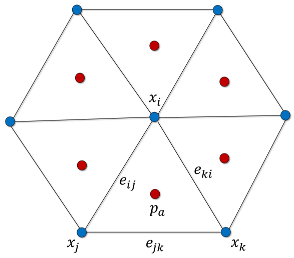


Figure 4: To compute position update Δx_i, we integrate all position displacements exerted by surrounding pressures.

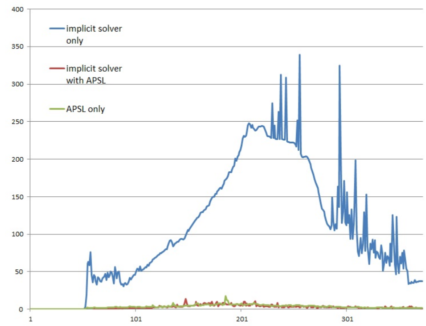


Figure 5: maximum area change rate (%) per frame for a simulation shown in Figure 7.

5. Position Update

After solving the linear equation, we compute the vertex position update Δx_i by integrating all position displacements exerted by surrounding pressures as below.

$$\Delta x_i = -\sum_{a \in S} \delta x_{i,a}$$

where vertex i is a part of triangle a
S is a set of triangle indices which surround the vertex i (6)

Algorithm 1 shows overall process. Our area-preserving strain limiting method is a velocity filter [BFA02]. So we update vertex velocity rather than position directly which helps plug our method into the existing simulation step easily.

Algorithm 1: Area preserving strain limiting is a velocity filter. We run it multiple times until the target threshold meets or a fixed times

Input: $\tilde{\mathbf{v}}$ // velocity
Input: $\tilde{\mathbf{x}}$ // position
$\mathbf{x}_j \leftarrow \tilde{\mathbf{x}} + \Delta t \tilde{\mathbf{v}}$ // predict unconstrained position
while *area is above target area threshold* ***or*** *terminates after a fixed iterations* **do**
  Set regularization value $\alpha > 1$;
  Solve linear system for $\mathbf{p}$;
  Evaluate (6) $\Delta\mathbf{x}$;
  Update $\mathbf{x}_j \leftarrow \mathbf{x}_j + \Delta\mathbf{x}$;
Output: $\tilde{\mathbf{v}} \leftarrow \tilde{\mathbf{v}} + \frac{1}{\Delta t}(\mathbf{x}_j - \tilde{\mathbf{x}})$

6. Stand-alone Solver

It is possible to use area preserving strain limiting as a stand-alone cloth simulation solver. In this case, we need a way to prevent triangles from becoming very skinny. By using edge length based PBD style approach, we can enforce minimum edge lengths. For our tests, we simply used linear bending springs which also act as edge length enforcement.

As Figure 6 shows, area preserving strain limiting can replace traditional cloth solvers.

7. Results and Discussion

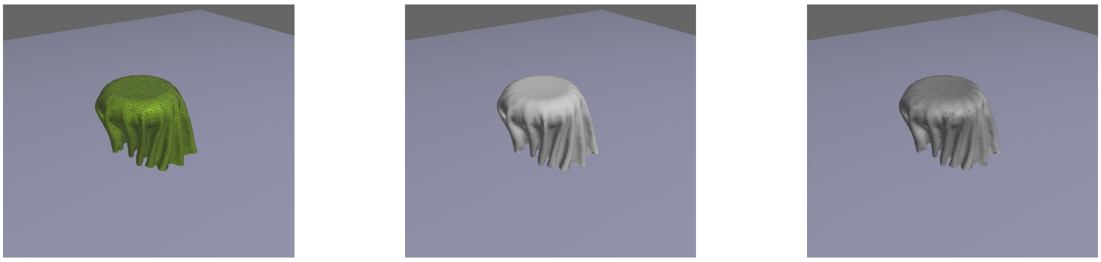
Figure 5 shows maximum area change rate (%) for simulation showing in Figure 7 from frame 0 to 370. Implicit solver without APSL shows large area changes which are often more than 300%. APSL as post-processing or stand-alone shows consistent area preservation showing almost a flat horizontal line close to the bottom axis.

We measured the area change rate before collision handling step. So it may be possible that collision impulses can cause area changes. However our main goal of using APSL is to resolve implicit solver's over-stretching problem without introducing locking issue. So small area change due to collision impulses is not much a problem here.

8. Conclusion

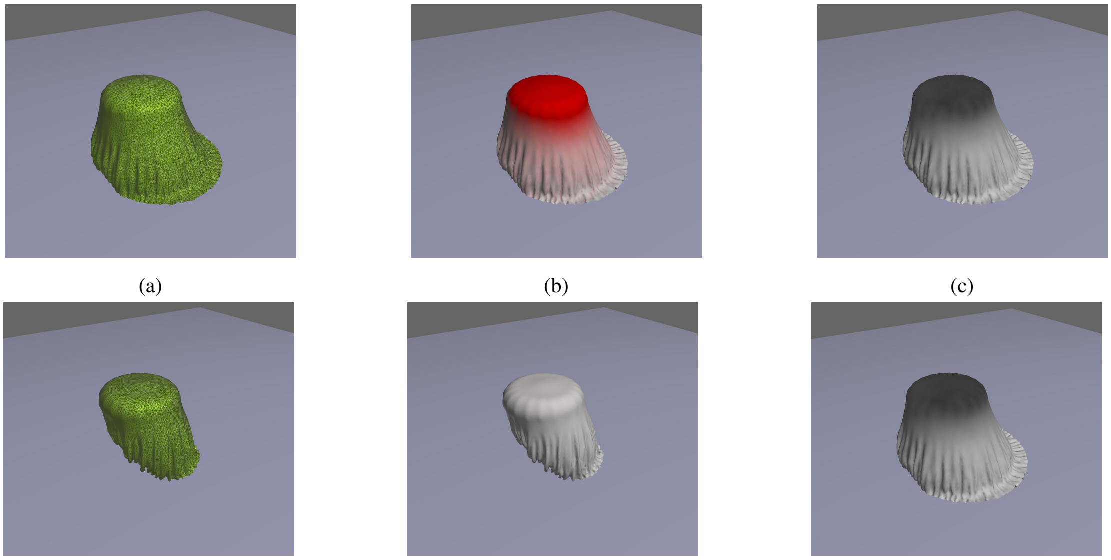
In this paper, we proposed a novel method to reduce excessive stretching in spring based cloth simulation with a faster convergence without introducing locking problem.

In the future, we would like to incorporate the bending springs into a part of linear system as in implicit stiff spring integration so that bending springs can be better controlled when APSL acts as a stand-alone solver. Also we want to explore volume preservation for volumetric mesh.



(a) (b) (c)

Figure 6: APSL as a stand-alone solver. The maximum area change rate is 4.77%. (a) is simulated cloth model. (b) shows area change rate in red. Since the area preservation was enforced tightly, the mesh shows white color which means the area was preserved well. (c) shows normalized area change rate by the maximum rate in the cloth. Thanks to APSL, the area deformation is even across the mesh.



(a) (b) (c)

(d) (e) (f)

Figure 7: (a) is a simulated cloth model using implicit integration without strain limiting. (b) shows area change rate where the maximum rate is 105.6%. Pure red means the area change rate is above 50%. (c) shows normalized area change rate by maximum rate in the same cloth. (d) is simulated using implicit integration with area-preserving strain limiting. (e) shows area change rate where the maximum rate is 3.64%. (f) shows normalized area change rate. (a) and (d) were captured at the same frame.

References

[BDB11] Bender J., Diziol R., Bayer D.: Simulating inextensible cloth using locking-free triangle meshes. In *VRIPHYS* (2011), Bender J., Erleben K., Galin E., (Eds.), Eurographics Association, pp. 11–17. 2

[BFA02] Bridson R., Fedkiw R., Anderson J.: Robust treatment of collisions, contact and friction for cloth animation. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2002), SIGGRAPH '02, ACM, pp. 594–603. 4

[BMF03] Bridson R., Marino S., Fedkiw R.: Simulation of clothing with folds and wrinkles. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, Switzerland, Switzerland, 2003), SCA '03, Eurographics Association, pp. 28–36. 1

[BW98] Baraff D., Witkin A.: Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1998), SIGGRAPH '98, ACM, pp. 43–54. 1

[CFL*07] Chentanez N., Feldman B. E., Labelle F., O'Brien J. F., Shewchuk J. R.: Liquid simulation on lattice-based tetrahedral meshes. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2007* (2007), pp. 219–228. 3

[FOK05] Feldman B. E., O'Brien J. F., Klingner B. M.:

Animating gases with hybrid meshes. In *Proceedings of ACM SIGGRAPH 2005* (Aug. 2005). 3

[GHF*07] GOLDENTHAL R., HARMON D., FATTAL R., BERCOVIER M., GRINSPUN E.: Efficient simulation of inextensible cloth. *ACM Trans. Graph. 26*, 3 (July 2007). 2

[HDB96] HOUSE D., DEVAUL R. W., BREEN D. E.: Towards simulating cloth dynamics using interacting particles. *International Journal of Clothing Science and Technology 8* (1996), 75–94. 2

[HH13] HAN D., HARADA T.: Tridiagonal matrix formulation for inextensible hair strand simulation. In *VRIPHYS 13: 10th Workshop on Virtual Reality Interactions and Physical Simulations, Lille, France, 2013. Proceedings* (2013), pp. 11–16. 2

[ISF07] IRVING G., SCHROEDER C., FEDKIW R.: Volume conserving finite element simulations of deformable models. *ACM Trans. Graph. 26*, 3 (July 2007). 2

[KFCO06] KLINGNER B. M., FELDMAN B. E., CHENTANEZ N., O'BRIEN J. F.: Fluid animation with dynamic meshes. In *Proceedings of ACM SIGGRAPH 2006* (Aug. 2006), pp. 820–825. 3

[KGBS11] KAVAN L., GERSZEWSKI D., BARGTEIL A., SLOAN P.-P.: Physics-inspired upsampling for cloth simulation in games. *ACM Trans. Graph. 30*, 4 (2011), 93:1–93:9. 3

[KPNS10] KANG N., PARK J., NOH J., SHIN S. Y.: A hybrid approach to multiple fluid simulation using volume fractions. *Computer Graphics Forum 29*, 2 (2010), 685–694. 3

[MCKM14] MÃIJLLER M., CHENTANEZ N., KIM T.-Y., MACKLIN M.: Strain Based Dynamics. In *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation* (2014), Koltun V., Sifakis E., (Eds.), The Eurographics Association. 2

[MHHR07] MÜLLER M., HEIDELBERGER B., HENNIX M., RATCLIFF J.: Position based dynamics. *J. Vis. Comun. Image Represent. 18*, 2 (Apr. 2007), 109–118. 1

[Mül08] MÜLLER M.: Hierarchical position based dynamics. In *VRIPHYS* (2008), pp. 1–10. 2

[PL07] PIGHIN F., LEWIS J. P.: Practical least-squares for computer graphics: Video files associated with this course are available from the citation page. In *ACM SIGGRAPH 2007 Courses* (New York, NY, USA, 2007), SIGGRAPH '07, ACM, pp. 1–57. 3

[Sch06] SCHENK OLAF G. K.: On fast factorization pivoting methods for sparse symmetric indefinite systems. *ETNA. Electronic Transactions on Numerical Analysis [electronic only] 23* (2006), 158–179. 3

[SCOIT05] SORKINE O., COHEN-OR D., IRONY D., TOLEDO S.: Geometry-aware bases for shape approximation. *Visualization and Computer Graphics, IEEE Transactions on 11*, 2 (March 2005), 171–180. 3

[Smi06] SMITH R.: Open dynamics engine v0.5 user guide. 2, 3

[TPS09] THOMASZEWSKI B., PABST S., STRAÃ§ER W.: Continuum-based strain limiting. *Computer Graphics Forum 28*, 2 (2009), 569–576. 1