Interactive Thin Elastic Materials

Wen Tang

Faculty of Science and Technology, University of Bournemouth

Poole, Dorset, BH12 5BB, United Kingdom

Tel. (+44)1202 962498

email: wtang@bournemouth.ac.uk

Tao Ruan Wan

University of Bradford

Tel. (+44)1274 236086

email:t.wan@Bradford.ac.uk

Donjing Huang

Shanghai University

Abstract

Despite great strides in past years are being made to generate motions of elastic materials such as cloth and biological skin in virtual world, unfortunately, the computational cost of realistic high-resolution simulations currently precludes their use in interactive applications. Thin elastic materials such as cloth and biological skin often exhibit complex nonlinear elastic behaviors. However, modeling elastic nonlinearity can be computationally expensive and numerically unstable, imposing significant challenges for their use in interactive applications. This paper presents a novel simulation framework for simulating realistic material behaviours with interactive frame rate. Central to the framework is the use of a constraint-based multi-resolution solver for efficient and robust modelling of the material nonlinearity. We extend a strain limiting method to work on deformation gradients of triangulated surface models in three dimensional space with a novel data structure. The simulation framework utilises an iterative nonlinear Gauss-Seidel procedure and a multilevel hierarchy structure to achieve computational speed ups. As material nonlinearity are generated by enforcing strain limiting constraints at a multilevel hierarchy, our simulation system can rapidly accelerate the convergence of the large constraint system with simultaneous enforcement of boundary conditions. The simplicity and efficiency of the framework makes simulations of highly realistic thin elastic materials substantially fast and is applicable of simulations for interactive applications.

Keywords: computer animation, computer games, cloth simulation, interactive virtual reality applications

Introduction

In computer graphics and computer animations, when dealing with simulation realism of elastic materials, we must consider how to model material nonlinearity. For example for cloth and biological skin, nonlinear material behaviors are one of the most striking aspects affecting simulation realism and computational efficiency. These materials are soft and compliant to small strains but very tough and resistant to large deformations under large forces. Cloth materials are classic examples which easily tolerate small amounts of in-plane stretching due to interlocking structures of fibers and yarns, but once the structural give-way being taken up, cloth materials very resist to further stretches (i.e. noncompliant). The material nonlinearity can significantly alter the stability and deformation of the structure and surface appearances during a simulation. Many research efforts have been made to produce realistic results and some of them can be difficult to distinguish from reality [1, 2]. Realism has been further enhanced by using measured material properties [3]. Unfortunately, the computational cost of realistic high-resolution simulations currently precludes their use in interactive applications. On the other hand, simulating material nonlinearity requires identifying nonlinear stress-to-strain relationships and predicting related responses.

While with standard methods, using traditional finite element methods often involves setting up complex constitutive models to determine material deformation behaviors, simple spring-and-mass system demands using stiff material parameters to enforce the nonlinearity. Yet, even representing highly stiff springs with manually selected parameters, the process is in itself considered a difficult task. Previous work have considered various different approaches in order to realistically and/or accurately simulate biphasic nonlinear behaviors. For example, in cloth simulation, researchers have observed that one way to achieve fine scale details for winkles and folders in many fabrics is to model this nonlinear stress-to-strain relationship [4, 5].

One popular approach in dealing with nonlinear deformations is to use constraints. However, computational costs of large constraint system is prohibitively expensive, making interactive applications infeasible. Simulating highly noncompliant material characteristics is difficult since standard simulation methods perform poorly for these materials both in terms of computational efficiency and numerical stability. Multi-resolution approaches are well-suited for accelerating the global convergence of large constraint systems, in which multi-levels of mesh details are required to quickly resolve constraints [6]. Local re-meshing scheme has been proposed to avoid processing data between serval fine and coarse meshes at the cost of introducing many new unknowns in order to refine gradual variations to the scale relevant to simulation features [2]. In summary, with standard spring-and-mass systems and with finite element models, using stiff material coefficients to enforce material nonlinearity for realistic simulation will cause severe numerical integration difficulties. Numerical instability manifests the use of small time integration steps, which makes interactive applications infeasible.

This paper presents a novel simulation framework for simulating nonlinear material behaviours with interactive frame rate. Central to the framework is the use of a constraint-based multi-resolution solver for efficient and robust modelling of the material nonlinearity. We extend a strain limiting method to work on deformation gradients of triangulated surface models in three dimensional space with a novel data structure. The simulation framework utilises an iterative nonlinear Gauss-Seidel procedure and a multilevel hierarchy structure to achieve computational speed ups. As material nonlinearity are generated by enforcing strain limiting constraints at a multilevel hierarchy, our simulation system can rapidly accelerate the convergence of the large constraint system with simultaneous enforcement of boundary conditions. The simplicity and efficiency of the framework makes simulations of highly realistic thin elastic materials substantially fast and is applicable of simulations for interactive applications. The iterative Guass-Seidel algorithm is capable of precessing large numbers of constraints. We use a standard elastic model such as a finite element model for modelling the material deformations with small strains, whereas large strains such as in-plane stretching and out-of-plane bending are enforced by hard constraints using constraint-based projections. We develop simulation tools to aid interactive applications by keeping computation and implementation relatively simple. We extend the current body of work on strain-limiting and propose complementary techniques to improve existing methods:

- A unified simulation framework based on strain limiting triangular elements, which allows for a unstructured multi-mesh hierarchy to be defined, computed and used. Our method operates on per-edge strain components, which does not require inverting matrices. By contrast, previous strain limiting methods that operate on per-triangle element require to compute Singular Value Decomposition of 6×6 matrices on each triangle for anisotropic materials [4] or 2×2 matrices for isotropic materials [1], which can be computational expensive for real-time applications.

- Out-of-plane bending strain limiting is added by using a bending matric of [7, 8] and similar to that of [1]. Bending constraints are considered to be a complementary representation of material properties e.g. stiffness in bending to in-plane deformations.

- A multi-resolution scheme to accelerate large constraint system to converge rapidly, which overcomes limitations of using stiff material parameters on a single mesh. This multi-resolution scheme essentially carries out nonlinear Gauss-Seidel iterations on individual per-edge strain for each hierarchical mesh structure with a progressive constraint enforcement. The scheme greatly speeds up the rate of global system convergency. Since refinements obtained in coarse meshes should not be mapped into the finer mesh, causing the loss of details in the final simulation, we propose a heuristics based refinement procedure using deformations from the finer level combined with results of corrections on coarse levels to preserve simulation details.

Simulation System Overview

We represent deformable thin material objects as triangular meshes. For the planar surface model the world space position of each vertex is denoted as $\vec{x}_i$ and its velocity being $\vec{v}_i$. We

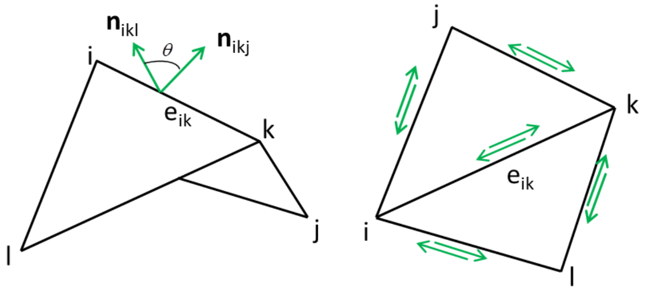

Figure 1: The out-of-plane and in-plane deformations of thin-shell objects represented in triangular mesh structures. Each deformation unit is formed by four vertices (i,j, k, and l) of two triangles sharing an edge e_{ik}

use the shorthand $\vec{x}_{ij}$ to represent the vector $\vec{x}_i - \vec{x}_j$ and the material space vector $\vec{u}_{ij}$ to express $\vec{u}_i - \vec{u}_j$. Lumped masses are used in our computation so that each node has a scalar mass m_i that is set to one third of the sum of material-space areas of its incident faces, multiplied by the material's area-density. Vertex normals $\vec{n}_i$ are computed using a weighted average of incident face normals.

For dynamic simulation, we take the approach of measuring in-plane membrane and out-plane bending separately. This is essentially computing the strain of triangular elements combined with a discrete bending metric described by Grinspun *et. al.* and Bridson *et. al.* [7, 8], as shown in Figure 1. This set up of configuration is invariant to rigid-body transformations. Thus, the physically-based deformation can be modeled by the sum of in-plane membrane and out-plane bending energies, expressed by:

$$\mathbf{W}(\mathbf{x}) = \mathbf{W_M}(\mathbf{x}) + \mathbf{W_B}(\mathbf{x}) \tag{1}$$

where $\mathbf{W_M}(\mathbf{x})$ is the membrane energy as a summation over lengths of edges shown as green arrows on the right image of the figure 1), and $\mathbf{W_B}(\mathbf{x})$ the discrete bending energy as a summation over mesh edges with the corresponding dihedral angle shown on the left image of the figure 1). These engineries are calculated by:

$$\mathbf{W_M}(\mathbf{x}) \quad = k_M \sum (1 - \|\tilde{\mathbf{x}}_{\mathbf{ij}}\| / \|\vec{\mathbf{u}}_{ij}\|)^2 \|\vec{\mathbf{u}}_{ij}\| \tag{2}$$

$$\mathbf{W_B}(\mathbf{x}) \qquad = k_B \sum (\theta_{ij} - \bar{\theta}_{ij})^2 \|\bar{\theta}_{ij}\| \tag{3}$$

where k_M and k_B are stretch and bending stiffness coefficients, respectively. Therefore, the dynamic system is governed by the ordinary differential equation of motion $\ddot{\mathbf{x}} = -\mathbf{M^{-1}}\nabla\mathbf{W}(\mathbf{x})$ where $\mathbf{x}$ is the vertices of the deformed geometry and $\mathbf{M}$ is the mass matrix. We use a semi-implicit integration scheme and nonlinear Gauss-Seidel iterative projections to constrain excessive deformations.

The strain limiting method is applied as a filtering step between the velocity and the position update. Our constraint-based techniques are independent of the choice of dynamic models, and would also work with other triangular methods such as triangular finite element methods. The Gauss-Seidel iterative approach has fast convergency when operates on small-scale. To avoid bias due to sequential iterations, we use random orderings. For large constraint system like ours, our multi-level hierarchy approach accelerate the system convergency rapidly as shown in testing results.

Strain Limiting Constraints

The basic concept of strain limiting is to impose hard constraints within a dynamic system. In doing so, edge length constraints can be imposed during updating node positions, while the bending angle between two triangle elements enforces restrictions on the flexural of the material. In our system, we enforce one-dimensional strain limits of length constraints. Given an edge connecting two nodes $\mathbf{x_a}$ and $\mathbf{x_b}$, the function of one-dimensional strain is defined as:

$$\varepsilon_{ab} = (|\mathbf{x_a} - \mathbf{x_b}| - l_0)/l_0 \tag{4}$$

where l_0 is the original edge length. Given the lower and upper bounds of the stretching/compression ratio $s = |\mathbf{x_a} - \mathbf{x_b}/l_0$ as $[s_{min}, s_{max}]$, the strain ε_{ab} is clamped to the range $[s_{min} - 1, \;\; s_{max} + 1]$ to enforce the constraint. For example. a material that is constrained to compress/stretch ratio within range $[-10\%, \;\; 10\%]$ would have $s_{min} = 0.90$ and $s_{max} = 1.10$. Length constraints are enforced by moving both nodes along edges or just move one node and let the other fixed if required by boundary conditions for a particular edge.

Dihedral angles across edges are used to enforce bending constraints [8, 7] as shown in the figure 1. Here, we adopt the same strategy used by Wang and his colleagues [5]. Bending constraints prevent the dihedral angle between two adjacent triangles from becoming too large or too small. Similarly, we enforce bending angle limiting coefficients $[\theta, \ \ \theta_{max}]$ i.e. degrees per unit length.

The strain limiting algorithm described above can yields the correct result in a single step for a single length constraint or bending constraint. However, each constraint enforcement affects positions of all nodes associated with it, resulting a *highly under-determined* system. Therefore, enforcing constraints over an entire mesh requires either the Jacobi's method or the Gauss-Seidel method to solve the linear system. We resolve to use the nonlinear Gauss-Seidel type solver that iteratively enforces each strain limit on triangular meshes. Boundary conditions are processed simultaneously with length and bending constraints so that enforcing one set of constraints would not violate others. The data structure designed in our system is flexible such that constraints required for boundary conditions can be enable or disabled during the run-time on a multi-resolution hierarchy.

A impulse based collision response [7] is implemented. The vertex is projected to the surface of the incident object and its velocity is adjusted according to the penetration depth. Frictions are treated in the similar way using penetration depthes to generate dynamic friction forces on each penetrating vertex along the relative tangential velocity.

Multi-resolution Approach

For small systems the nonlinear iterative solver works well and the convergency of such solver was studied in [9, 4]. Since the nonlinear Gauss-Seidel type solver deals with each constraint separately, convergency for systems involving large numbers of constraints is slow. Multi-resolution approach handles such problem by enforcing constraints on coarse mesh levels, while attempting to maintain details in finer meshes. At the heart of this approach is the hierarchical data structure for iterative projections and preserving details to produce high quality simulations. Otherwise, advantages of the rapid convergency brought forward by multi-resolution algorithm will be undermined by the lost of simulation quality.

The hierarchical construction

The hierarchical mesh structure required for a multi-resolution algorithm to work contains a single, highly-detailed base mesh. The subsequent coarser mesh at each level of the hierarchy is generated directly from the base mesh by a desired triangle count for each level. During the mesh simplification process, the quality of triangle elements are measured using a error metric described in [10]. There is a practical, but not theoretical, limit to how many levels of detail the hierarchy can contain. We use three to five levels of detail for optimal results in terms of memory storages, computational performance, and the quality of simulations. All testing examples in §6 are using four levels for a demonstration purpose.

We build a mesh hierarchy containing m levels $H^{(i)}$, for $i = 0, ..., m-1$, where $H^{(0)}$

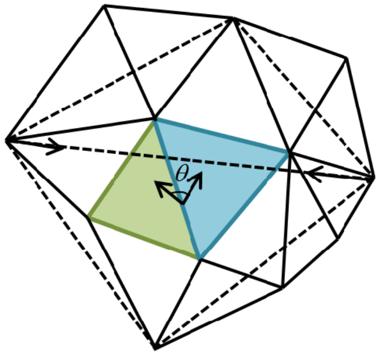

Figure 2: Multi-resolution mesh hierarchy construction: the solid lines show a subset triangles in the base mesh $H^{(0)}$. The dashed lines show two triangles in $H^{(i)}$ that cover neighbourhoods of the base mesh. Bending in the fine mesh appears as compression/expension at coarser levels.

is the original base mesh representing the finest mesh of the hierarchy, and $H^{(m-1)}$ is the coarsest mesh level constructed from the base mesh. Nodes at a coarser level i covers a subset of nodes at finer level $i-1$. Figure 2 shows the hierarchy construction of the mesh level, where solid lines show a subset triangles in the base mesh $H^{(0)}$, and dashed lines show two triangles in $H^{(i)}$ that cover proportions of the base mesh. Bending in the fine mesh appears as compression/expension at coarser levels. For each vertex in a fine mesh, we find corresponding barycentric coordinates at each coarse level, so that each vertex at the finest level is made aware of its parents at coarser levels to facilitate the down-sampling process. Similarly, corresponding barycentric coordinates of each vertex at a coarser level is computed to the closest triangle element in the one level up the hierarchical chain on finer mesh to enable the up-sampling process. Therefore, each vertex at level i has associated elements at levels $i-1$ and $i+1$. A scalar or a vector field defined over a coarser level can then be up-sampled to the finer level using linear interpolation. Propagating information from a base mesh to coarser meshes is handled in the same way. We term the up-sampling and down-sampling process as the *projection* process denoted as a projection operator $P^{(i)}_{(i-1)}$, projecting from level i to level $i-1$.

Processing the hierarchy

The multi-resolution hierarchy process starts with the physically-based computation on the bases mesh. Information of the base mesh is down-sampled to carry out the simulation on coarse meshes. At each coarser mesh level, nonlinear Gauss-Seidel iterations are run to enforce limiting constraints on the current coarser level. Results of the nonlinear strain limiting algorithm are then up-sampled on finer level up the hierarchical chain to enforce constraints on the base mesh.

We summarize this algorithm in the figure 3. The method described here can be used as an additional constraint enforcement step in a standard simulator, which is divided into a number of stages of computations:

- dynamic simulation and enforce constraints
- Down-sampling projections
- Strain limiting on coarse meshes

- Enforce constraints

- Up-sampling strain limited results

Algorithm 1 summarizes stages of a single hierarchical simulation in our system.

Algorithm 1:Multiresolution strain limiting

```
1: **Procedure** MULTI-RESOLUTION
2: $X_0 \longleftarrow \{x_0, x_1, x_2, ...\}$ // compute positions on $H^{(0)}$
3: $\dot{X}_0 \longleftarrow \{\dot{x}_0, \dot{x}_1, \dot{x}_2, ...\}$ // compute velocities on $H^{(0)}$
4: **for** all mesh levels $i = 0 \longrightarrow m - 1$ **do**
5: $H^{(i)} \longleftarrow H^{(0)}$  // sample positions on $H^{(i)}$
6: **end for**
7: **for** all constraints on $H^{(i)}$ **do**
8:     Nonlinear Gauss-Seidel on $H^{(i)}$
9: **end for**
10: **for** all particles on $H^{(i)}$ **do**
11:     Procedure PROJECTION($X_i$, $X_{i-1}$)
          // up-sampling from $H^{(i)}$ to $H^{(i-1)}$
12: **end for**
13: **goto** step 1
```

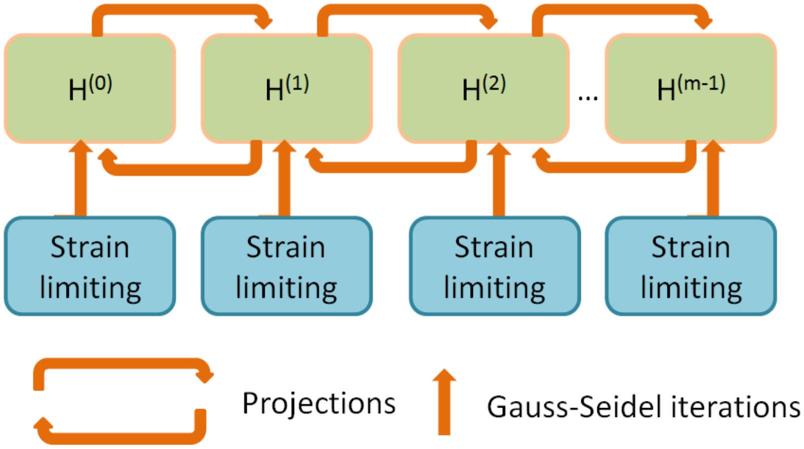

Figure 3: Multi-resolution mesh hierarchy process pipeline for our simulation system.

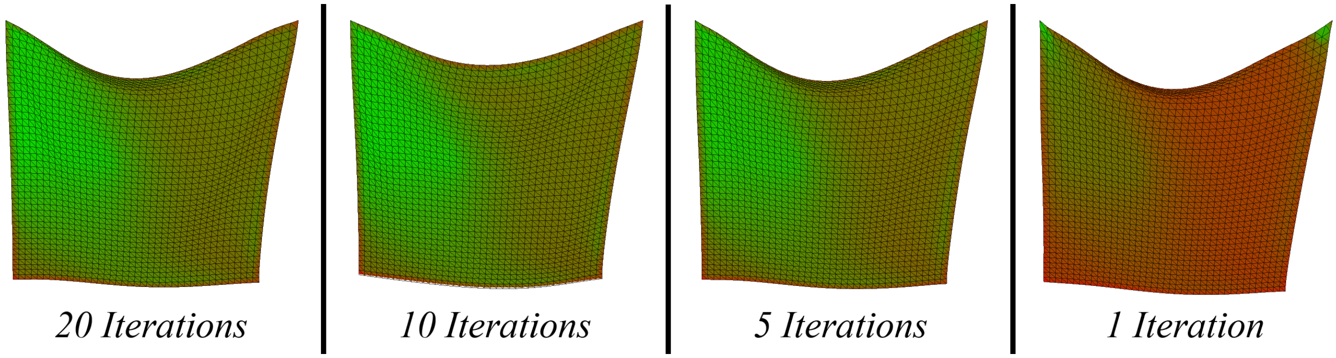

Figure 4: Comparison of the effect of the number of Gauss-Seidel iterations on simulating material stiffness. A four-levels multi-resolution hierarchy is used in these tests and the magnitude of red indicates in-plane stretches of triangle elements. As can be seen, more iterations produce more incompliant material behaviors.

Handling constraints

When dealing with multiple meshes, care must be taken to maintain all constraints are satisfied at all levels. Fixed vertices that are constrained at the base mesh must also enforced at all coarser levels. However, since certain constraints contained in finer mesh are not present on a coarser mesh, we compute the closest corresponding vertex on the coarser mesh and snap it to the same constraint position of the finer mesh where the distance is smaller than a certain threshold. Otherwise, we insert a new vertex and do a remesh process to add/delete triangles to the mesh with edge lengths that maximize the quality of triangles. This method mimics geodetic distances within the manifold of the mesh. In this way, we can ensure fixed constraints are satisfied at all levels of hierarchy. In doing so, the vertex snapping generates new distance constraints which need to be computed so that yielding constraints are not too loose or too tight because new edges may only be approximations of geodesics.

To determine on the coarser mesh which vertex is the closest to the fine level vertex we compute the average positions of all coarse neighbors of the vertex using the material space positions. We then choose a vertex on the coarser mesh that is the closest to the average position so that it generates evenly distributed edge lengths. The distance of the coarser constraints is set using the original positions of the mesh. In this way, finest level fixed constraints enforce boundary conditions simultaneously at all levels and these are enforced at each coarser level during multi-resolution strain limiting iterations.

Collision responses can be treated as impulse constraints, which are considered as the violation of a real-valued constraint function. We detect collisions between vertex and triangles and resolve these constraints as impulse response by applying a repulsive impulse along the constraint direction.

Detail persevering

The multi-resolution nonlinear solver works well for a large constraint system converge efficiently. However, detail preserving during up- and down- sampling processes is at the heart of the algorithm if it is to be useful in producing simulations without loosing intrinsic dynamic details, even though level of detail physics for objects far from the camera are quite useful in some game applications.

In our simulation, coarser levels of strain limiting stages act as correction steps, while the details in the base mesh are firstly saved in coarser meshes. Subsequent sampling projections correct the position of each vertex node on coarser meshes using barycentric weights w.r.t its parent triangles. Therefore, results of the nonlinear strain limiting at each level is propagated up the hierarchical chain not the entire solution. This guarantees that all small details in the high resolution levels are preserved as demonstrated in testing results. The projection procedure included in Algorithm 1 is summarized in Algorithm 2.

Algorithm 2 : **Projecting strain limiting**

```
Procedure PROJECTION($H^{(i)}$, $H^{(j)}$)
for $\mathbf{i = 0} \longrightarrow \{\mathbf{x_0}, \mathbf{x_1}, \mathbf{x_2}, ...\}$ in $H^{(i)}$
$X_i \longrightarrow \{x_0, x_1, x_2, ...\}$  // save positions on $H^{(i)}$
    for $j = 0 \longrightarrow all\ parents\ for\ this\ particle$ do
        $X_{proj} = X_i + \Sigma w_j X_j$
    end for
endfor
```

Results and Discussion

Simulation comparison

We have implemented these methods for modelling more accurate and detailed thin-shells for simulating nonlinear highly incompliant materials.

In figure 4 we show a classical example of a square piece of sheet hanging at its two corners. Depending on the stretchiness of the material, this may give rise to various characteristic deformation patterns. We show simulated results in this scenario for several iteration strategies. The red color indicates the magnitude of over stretched regions across the surface of the sheet, demonstrating the strain limiting effect on the stiffness of the material. With the increasing number of iterations, the over stretching of the material are decreased exponentially, which shows that constraints are being rigorously enforced. The 20 iterations setup

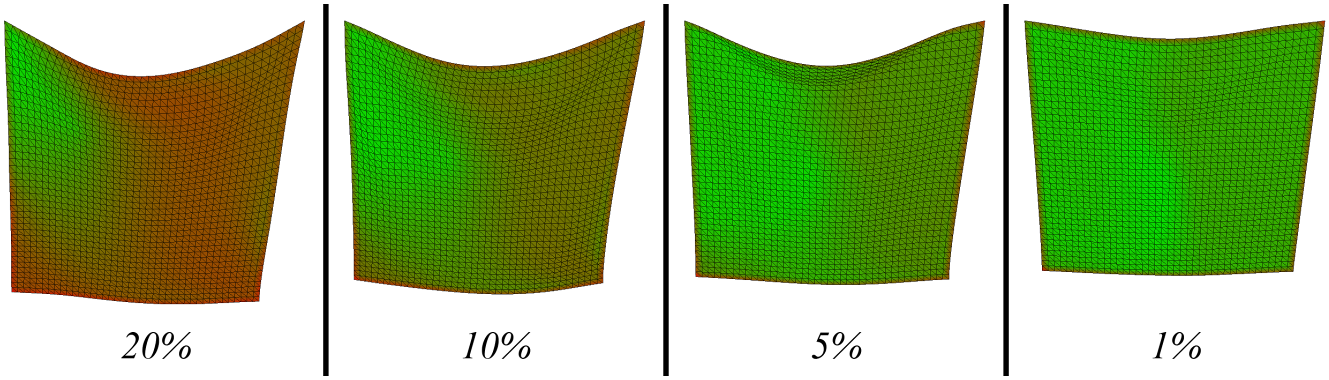

Figure 5: Comparison of effects of strain limiting thresholds on simulating material stiffness. A four-levels multi-resolution hierarchy is used in these tests and the magnitude of red indicates in-plane stretches of triangle elements. As can be seen, our system generates an overall non-stretched thin-shell with 1% strain limiting threshold as shown on the far right image.

produces an overall much less stretched thin-shell configuration, showing the effectiveness of our proposed algorithm.

Figure 5 shows the same simulation setup to test different strain limiting thresholds that control the resistance of the material to stretching beyond the material's natural rest state. The substantial in-plane stiffness enforced by small values cause the sheet to resist stretching, instead of having curvatures that undergo large significant in-plane deformations. As shown in the figure 5, using multi-resolution approach only requires 20 iterations at each level to enable a highly constrained $[-1\%, 1\%]$ threshold system to reach an overall non-stretching configuration. To avoid bias due to sequential iterations, we use random orderings to process constraints, causing the system to converge more rapidly.

To further demonstrate the effectiveness of the multi-resolution algorithm in resolving large number of constraints, we show in the figure 10 an example of a flat sheet being fixed at its four corners, deforming into a star shape by a rigid object in the center. As can be observed that constraints are enforced more effectively on coarser levels with the same number of iterations, enabling these meshes to reach an overall un-stretched state more rapidly. Thus, by experimenting with different number of iterations on coarser levels that do not have to use the same number of iterations as with the fine level accordingly, the optimal results that strike a balance between the cost of computation and the simulation accuracy can be achieved. As it is shown that the loss of simulation details on the coarser meshes are clearly visible. Therefore, it is vitally important that simulations of coarse level should not be causing artefact in final results. Our projection algorithm preserves fine details on $level0$ by strain limiting corrections only as shown here in the figure 11.

Figure 11 further compares the quality of strain limiting between simulations using a single mesh and a multi-level hierarchy. Single mesh simulation requires 200 the Gauss-Seidel type iterations to obtain an almost identical result to that of the multi-resolution approach on the finest level, in which 50 iterations are used on each coarser level. In figure 11, it also demonstrates that details of simulation are preserved well in the finest mesh of the multi-resolution simulation, which has almost identical folders that are achieved with the single mesh undergoing the same user interactions.

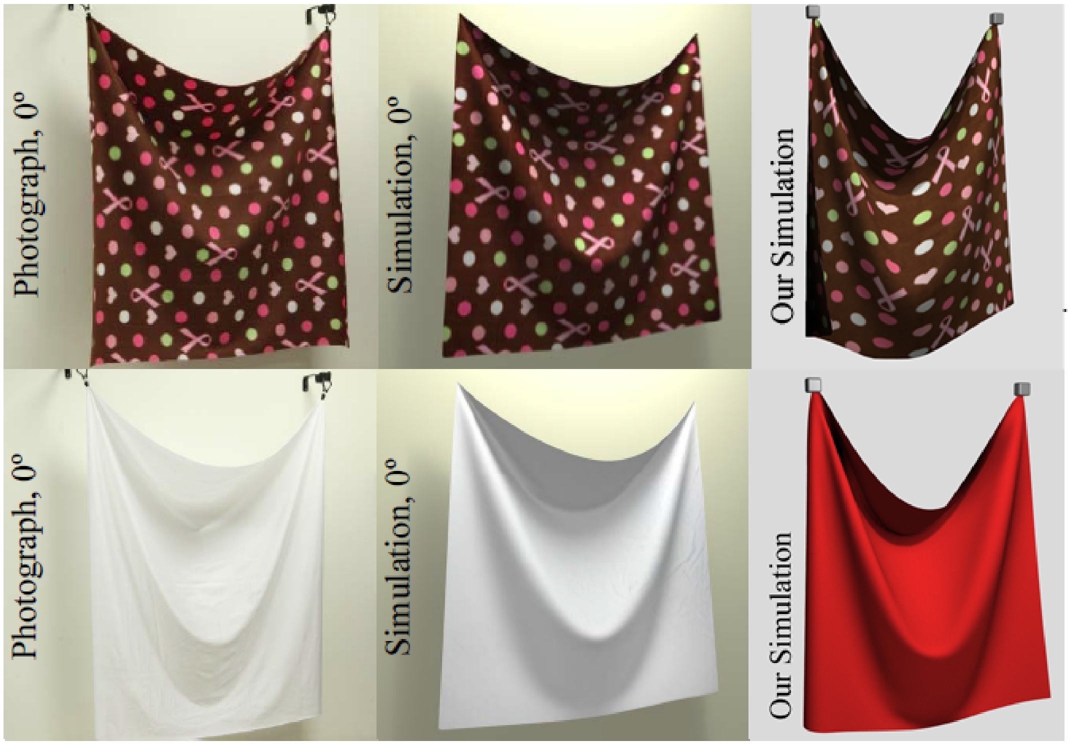

Figure 6: A comparison of simulation results to real hanging sheets by using different material parameters: Left column of images are photographs of real sheets; middle column of images are simulated sheets generated by Wang using a data-driven elastic model; right column of images are our simulation results (permissions of using their photographs and images are kindly given by Wang).

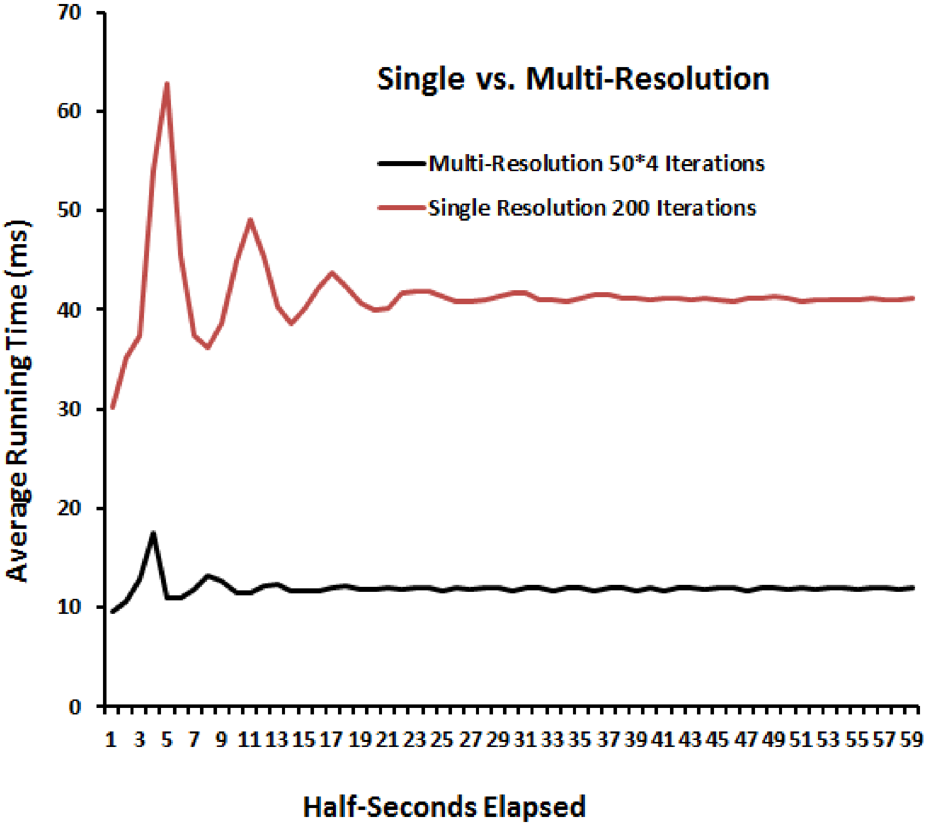

Figure 7: Performance timing for examples on a square sheet of 1600 vertices to compare average timing between a single level and a four-levels hierarchy strain limiting with $[-10\%, 10\%]$ limiting values, material density $0.113kg/m^2$ for a 100% polyester material.

0.1 Computing time comparison

As shown in above examples, one advantage of the proposed method is attaining similar or even improved simulation results from significantly less strain limiting iterations.

In figure 7, we show a half-seconds elapsed timing for 200 iterations using a single level of detail compared with a total of 200 iterations on a four-levels of hierarchy with 50 iterations on each level. In figure 7, it is clearly observable that, in this testing case, multi-resolution approach performed superbly in terms of computational efficiency. This result can be explained that on coarser levels, the proposed nonlinear Guass-Seidel algorithm has much less constraints to process and the number of elements that potentially violate strain limiting thresholds are dramatically reduced by coarsening the base mesh.

At the beginning of the simulation, there is a large number of elements violating constraints. As a result of which, computational costs to enforce these constraints are much higher and shown rapid transient high computational costs. However, multi-resolution procedure prevents rapid transient motion at the beginning of the simulation and requires less computational power and the constraint system is more quickly to converge.

Figure 8 shows the average running time for a four-levels of hierarchy using different number of iterations. Depending on applications, we have experimented various simulation setups and obtained real-time simulations by using 10 to 50 iterations on each level for optimal results. These timing graph were collected on a 3.30GHz Intel Core i5 2500 CPU with 8GB RAM. In these tests we are using measured real material parameters (i.e. material densities) provided in [11]. While these measured data sets including the one published in [12] are great resources, further work is needed to develop simulation models that produce more realistic dynamics of simulated materials that match their real-world counterpart.

Figure 6 shows a comparison of real hanging sheets with simulated sheets of different materials. Images in the left column are photographs of real sheets provided in [5] and middle column images show their simulated counterparts generated by Wang using his proposed data-driven elastic model for cloth [11], which are compared with the right column of

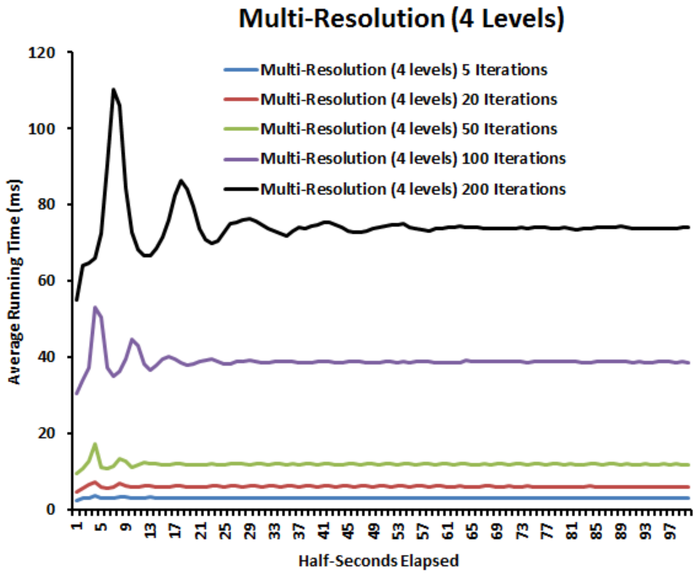

Figure 8: Performance timing for examples on a square sheet of 1600 vertices on a four-levels hierarchy to compare average timing between different strain limiting values, material density $0.113kg/m^2$ of 100% polyester material.

images of our simulation results by choosing the same materials.

Analysis and Conclusion

We have presented a simulation framework for real-time modeling triangular thin-shell mesh, producing highly constrained incompliant elements with computational efficiency that is suitable for virtual reality applications. Our multi-resolution strain limiting scheme preserves dynamical details and handles user interactions well. We have also proposed data structures that make efficient implementations of the multi-resolution algorithm and effectively enforce different types of constraints at multi-levels of hierarchy. Our immediate objective is to make simulation of highly in-complaint materials substantially faster. Although working with static structures maintain a cache-friendly memory layout of the mesh, we believe that dynamic data structures will enable the adaptive re-meshing for added simulation complexity such as modeling highly irregular mesh adaptively for detailed creases and tearing.

In our current implementation, we use a semi-implicity integration scheme that keeps the simplicity and stability of the algorithm, but potential drawbacks of semi-implicit solver need to be further investigated, for example extra residuals and numerical damping on simulation effects. We have also experimented with the approach in which correction steps act as a velocity filter [7, 5] and achieved similar simulation results. Finally, we show our test scene in a game environment with Atrium Sponza Palace with 10 curtains of simulated cloth in real-time plus 5 visible lights for interactive rendering

For most of the strain limiting thresholds controlling the material stiffness, same values work well for different scenarios and different materials, which can be considered as an advantage of the algorithm being material independent. However, defining the exact relationship between the stretching and the bending stiffness remains an open challenge, especially for realistic nonlinear material models as in our simulation context. It is difficult to establish a model that would predict material buckling relating to its in-plane stretching, so that dynamic motion of the material appears more consistent with bending and buckling. We plan to improve collision detections including self-collision detections by using open source libraries [13].

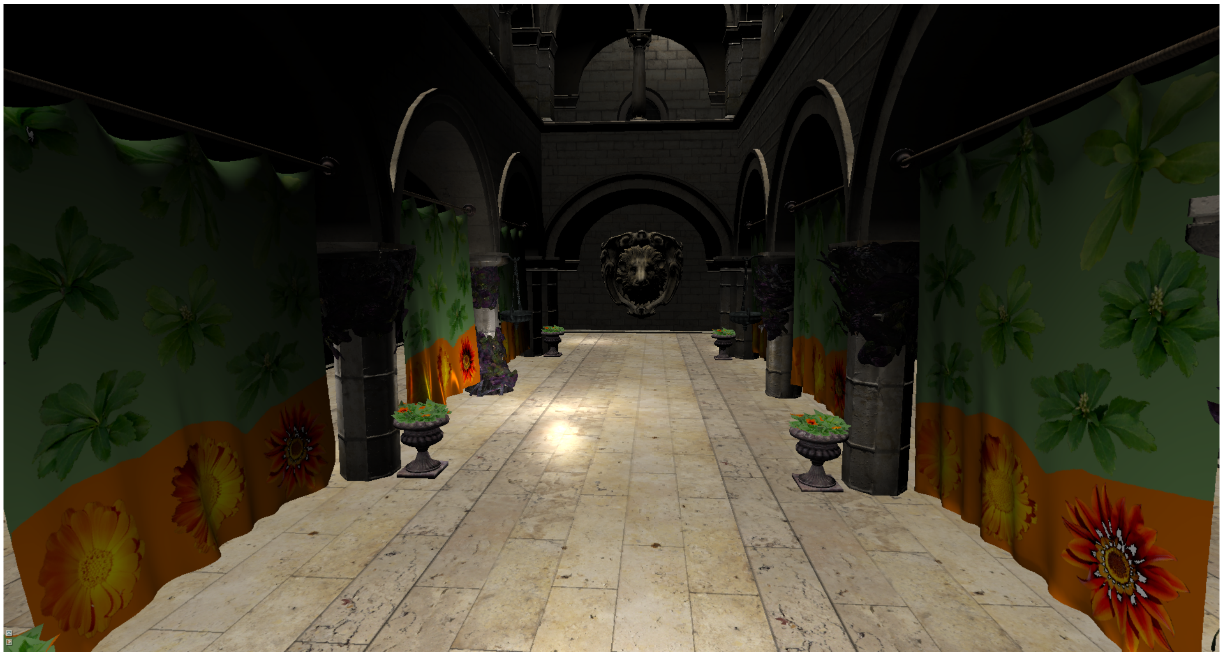

Figure 9: Test scene: Atrium Sponza Palace with 10 curtains of simulated cloth in real-time plus 5 visible lights for interactive rendering, 25 fps archived.

References

[1] M. Müller and N. Chentanez. Wrinkle meshes. *In Proc.' 10 ACM SIGGRAPH/Eurographics Symposium on Computer Animation.*, pages 85–92, 2010.

[2] R. Narain, A. Samii, and J. F. O'Brien. Adaptive anisotropic remeshing for cloth simulation. *In Proc.' 10 ACM SIGGRAPH/Eurographics Symposium on Computer Animation.*, 31, 6(Nov.):147:1–10, 2012.

[3] H. Wang, J. F. O'Brien, and R. Ramamoorthi. Data-driven elastic models for cloth: modeling and measurement. *ACM Transactions on Graphics*, 30(4)::71:1–11, 2011.

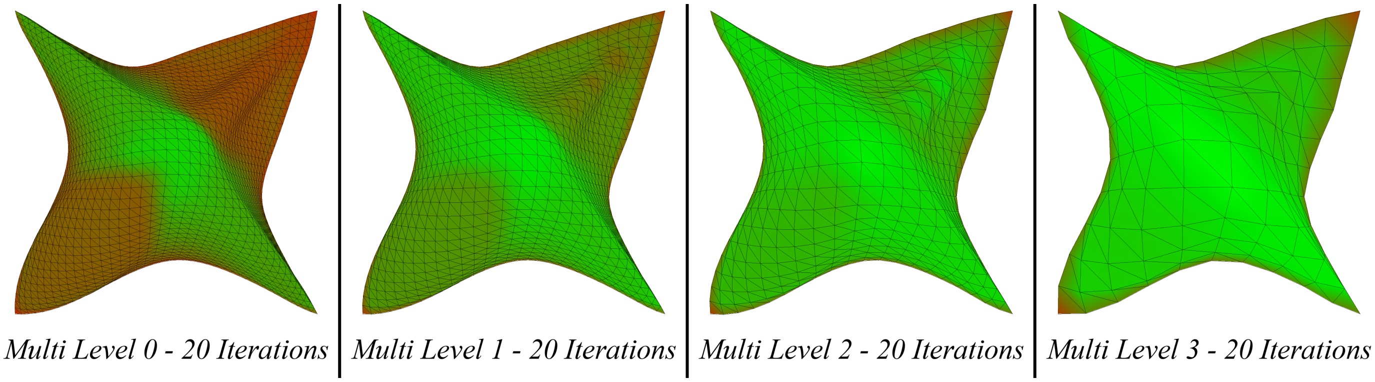
Multi Level 0 - 20 Iterations | *Multi Level 1 - 20 Iterations* | *Multi Level 2 - 20 Iterations* | *Multi Level 3 - 20 Iterations*

Figure 10: Results of 20 iterations on each level of a multi-resolution hierarchy, in which a thin-sheet is pined at its four corners with a sphere interacting under the sheet, deforming the material into a star configuration. Coarser levels reach non-stretching states more rapidly. Significant loss of simulation details are also clearly evident in coarser meshes. Yet, fine details are persevered well on the finest level with our proposed projection procedure.

[4] B. Thomaszewski, S. Pabst, and W. Strasser. Continuum-based strain limiting. *In Proceedings of Eurographics.*, 28:569–576, 2009.

[5] H. Wang, J. F. O'Brien, and R. Ramamoorthi. Multiresolution isotropic strain limiting. *In Proceedings of ACM SIGGRAPH Aisa*, 160::1–10, 2010.

[6] M. Müller. Hierarchical position based dynamics. 2009.

[7] R. Bridson, S. Marino, and R. Fedkiw. Simulation of clothing with folds and wrinkles. *In Proceedings of SCA*, pages 28–36, 2003.

[8] E. Grinspun, A. N. Hirani, and P. Desbrun, M.and Schröder. Discrete shells. *In Proceedings of SCA*, pages 62–67, 2003.

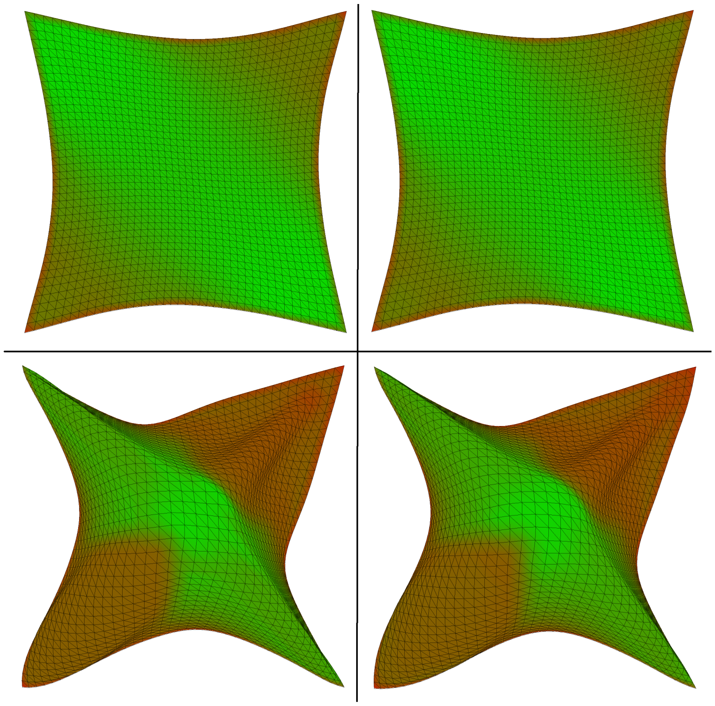

Figure 11: Comparison of results between a single mesh shown in the left column of images with a four-levels of hierarchy shown in the right column of images. The single mesh algorithm requires 200 iterations to obtain almost identical results from 50 iterations on a

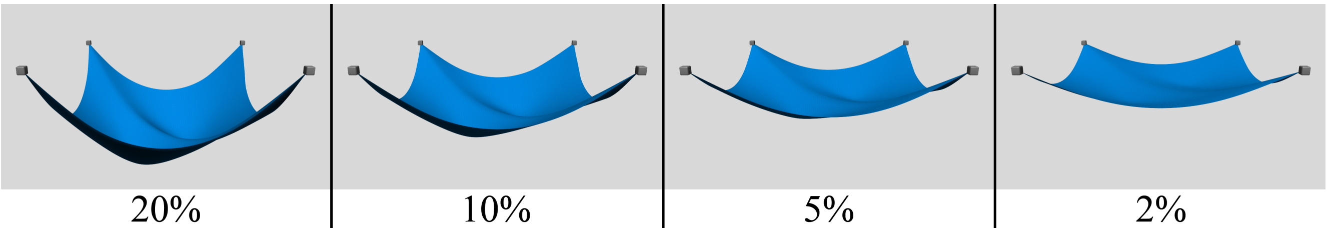

Figure 12: Various strain limiting thresholds produce different results for a thin-sheet using a multi-resolution hierarchy. Images show final resting configurations of the finest mesh of the hierarchy. Four-levels hierarchy is used with 20 iterations at each level. From left to right, strains limiting values are $[-20\%, 20\%]$, $[-10\%, 10\%]$, $[-5\%, 5\%]$ and $[-2\%, 2\%]$, decreasing limiting thresholds generated highly constrained material configuration with intricate surface details.

[9] A. Schmitt, J. Bender, and H. Prautzsch. On the convergence and correctness of impulse-based dynamic simulation. *Internal Tech. rep. 17, Institut fr Betriebs- und Dialogsysteme.*, 2005.

[10] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. *In Proceedings of SIGGRAPH*, pages 209–216, 1997.

[11] H. Wang, J. F. O'Brien, and R. Ramamoorthi. Data-driven elastic models for cloth: modeling and measurement. *ACM Transactions on Graphics (SIGGRAPH 2011)*, 30(4):71:1–71:12, 2011.

[12] P. Volino, N. Magnenat-Thalmann, and F. Faure. A simple approach to nonlinear tensile stiffness for accurate cloth simulation. *ACM Transactions on Graphics*, 2009.

[13] M. Tang, D. Manocha, and R. Tong. Fast continuous collision detection using deforming non-penetration filters. *I3D '10: Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, pages 7–13, 2010.