Multi-Resolution Collision Handling for Cloth-like Simulations

Nitin Jain
UNC Chapel Hill
NC, USA 27599
nitin@cs.unc.edu

Ilknur Kabul
UNC Chapel Hill
NC, USA 27599
ilknurk@cs.unc.edu

Naga K. Govindaraju
UNC Chapel Hill
NC, USA 27599
naga@cs.unc.edu

Dinesh Manocha
UNC Chapel Hill
NC, USA 27599
dm@cs.unc.edu

Ming Lin
UNC Chapel Hill
NC, USA 27599
lin@cs.unc.edu

Abstract

We present a novel multi-resolution algorithm for simulation of complex cloth-like deforming meshes. Our algorithm precomputes a multi-resolution hierarchy by using a combination of "chromatic decomposition" [1] and polygonal simplification of the underlying mesh. At runtime we selectively refine or coarsen the mesh based on the collision proximity of the mesh primitives with non-adjacent primitives. Our algorithm handles all kind of contacts, including self collisions among mesh primitives. The multi-resolution hierarchy is used to compute simplification of contact manifolds and to accelerate collision detection and response computations. We have implemented our algorithm on a high-end PC and applied it to complex simulations with tens of thousands of polygons. In practice, our algorithm is able to achieve interactive performance, while maintaining good visual fidelity.

Keywords: chromatic decomposition, collision detection , multi-resolution simulation, collision response

Introduction

Modeling human characters is an integral part of computer animation. Simulating cloth is crucial for generating realistic virtual humans. Cloth is deformable. It can have many folds and wrinkles that result in self collisions, which is difficult to detect efficiently.

Cloth simulation involves collision detection between cloth and objects in the environment (e.g. the avatar itself) and self-collision detection, as well as computing robust and stable collision response. Numerous techniques for cloth simulation have been proposed (see a recent survey [2]). Recent advances in cloth simulation have made it possible to animate cloth with striking visual realism and have been used in special effects and featured animation films (e.g. Star Wars, Monsters Inc., Shrek II). However, most of existing techniques can easily take up to minutes to simulate a single frame depending on the complexity of contact geometry, making them unsuitable for interactive applications.

Main Results: We present a novel adaptive algorithm to accelerate cloth simulation and collision detection. Our technique is inspired by the concept of geometry simplification for rendering. We propose to precompute a multiresolution hierarchy based on "chromatic decomposition" [1] and the subdivision framework [3]. At runtime, this hierarchy is used to accelerate the simulation by adaptively selecting the appropriate levels of detail for collision detection and response, depending on the proximity tolerance between the cloth and other objects (or other non-adjacent polygons of the cloth itself). This hierarchy can also be used to guarantee a constant frame rate for interactive applications, if visual quality of the simulation is less critical.

We have implemented a prototype system that can automatically generate simplified representations for a cloth mesh, select appropriate resolutions for each region, and switch between them seamlessly. We have applied our algorithm to the dynamic simulation on several challenging scenes that have high numbers of folds and wrinkles. The preliminary results indicate a noticeable performance improvement on these benchmarks, with little loss on the visual fidelity of the simulation.

Organization: The rest of the paper is organized as follows. Section 2 gives a brief survey of related work. Section 3 presents an overview of our approach. Our hierarchy construction for cloth simulation is described in section 4. Next, the mechanism for accelerating dynamic simulation using the multiresolution hierarchy is discussed in section 5. Section 6 presents the results of our prototype implementation, and compares its performance against prior methods. Finally, we conclude with future research directions in section 7.

Related Work

In this section, we provide an overview of the previous work on multi-resolution methods for simulation, and collision detection and response for modeling cloth. Due to space constraints, we will not be able to provide a comprehensive survey and refer the readers to recent survey [1, 4]. We will limit our discussion to the work directly relevant to our approach.

Multi-Resolution Simulation

Multiresolution techniques have been proposed to accelerate simulation for deformable objects using finite element methods [5, 6], subdivision framework [7, 8, 9], skeleton-driven deformation [10], Green's functions [11], tetrahedralization of space [12], physically-based subdivision [13], multi-level optimization [14, 15], etc. Algorithms using multiresolution representations for collision detection between rigid objects have also been proposed in [16, 17, 18, 19]. Volkov and Li also suggested a multi-resolution technique for local refinement and simplification of cloth meshes [20]. However, their technique does not allow for fast collision detection. In this paper, we use multiresolution representations coupled with the chromatic decomposition [1] of the meshes to accelerate both the dynamics computation and collision detection for cloth-like simulation of deformable meshes.

Collision Detection and Response for Cloth

Many collision detection algorithms have been proposed for cloth and deformable objects. Repulsion forces have been used between the two potentially colliding primitives to prevent them from penetration [21, 22, 23] and actual collisions are tested between the primitives. Volino and Thalmann [24] presented a curvature-based test along with convexity properties to provide a sufficient condition for detecting self-collisions [25, 26, 27]. Although this test is versatile and can be applied in a hierarchical manner on large models, it can be expensive for interactive applications [27].

Other algorithms based on hierarchy of K-DOPs [25], subdivision surface representations [7], level-set methods [28], etc. have also been suggested. Some algorithms treat each polygonal primitive as a separate object and apply N-body collision detection algorithms based on sorting using enclosing axis-aligned bounding boxes (AABBs) or grids [29, 30]. But, culling based on AABBs or rectangular cells of a grid can be very inefficient. In addition, the storage requirements of coherence based sorting algorithms can grow as a quadratic function of the number of primitives.

Approximation algorithms have been proposed to reduce the cost of collision detection. For example, some algorithms either do not check for self-collisions [31, 32] or perform approximate collision detection using multiple layers [33] or voxelized grids [34]. It can be difficult to give rigorous bounds on the accuracy of a simulation using approximate collision detection.

In addition to detecting all the geometric collision, many algorithms have been proposed for cloth simulation [21, 22, 26] and robust handling of contacts [27, 28, 35, 36, 37], as well as improving stability of the simulation [21, 28, 36, 38].

Overview

In this section, we present an overview and notation for our multi-resolution collision detection algorithm. We briefly describe a novel collision detection algorithm to handle general deformable models without using multi-resolution representations. We then present the multi-resolution properties and the constraints used in our novel multi-resolution algorithm.

Multi-resolution Collision Handling

The goal of our multi-resolution collision handling algorithm is to improve the performance of collision detection and response algorithms using *dynamically* computed triangular representations that closely approximate the original mesh.

Adoption of a multi-resolution algorithm for cloth simulation aims to optimize time performance by representing only parts of the cloth model at high resolution, which are considered more important or critical. Our approach couples multiresolution simulation with collision detection and response, while maintaining the properties of simplification that ensure smoothness and visual realism.

In the rest of this paper, we use the following notation. The mesh at resolution front f having n primitives is represented using the symbol M_f, and its primitives are represented as p_i, $i = 1, \ldots, n$. Primitive p_i at resolution level m is represented by p_i^m. The subdivision hierarchy constructed from M is represented as $M^k, k = 1, 2, \ldots, n$, where the base mesh is at resolution level n. A refinement operation on primitive p_i^m is denoted by $refine(p_i^m)$. Δt is used to represent the maximum timestep between two consecutive steps of the simulator. A vertex is represented by v and an edge is represented by e. A set s consisting of a number of colors is written as C_s. n_i is the normal of the primitive p_i, and $angle(n_a, n_b)$ denotes the angle between the two vectors n_a and n_b.

Chromatic Decomposition

We use the chromatic decomposition algorithm [1] for detecting all collision contacts. The algorithm partitions the mesh into independent sets $S_1, S_2, \ldots, S_n$ such that $M_f = S_1 U S_2 U \ldots U S_n$, with the following constraints on each primitive in the set $S_i, i = 1, 2, \ldots, n$:

- **Constraint 1**: p_a is not adjacent to p_b if $p_a \in S_i, p_b \in S_i$.
- **Constraint 2**: There is at most one $p_a \in S_i$ that is adjacent to $p_b \in S_j$ for every pair of independent sets (S_i, S_j).

As an illustration, a simple example of the mesh decomposition achieved with a rectangular mesh is shown in Figure 1. We achieve the required mesh decomposition by transforming this problem into a graph coloring problem and then computing the independent sets on this graph. The transformation of M_f to a graph $G(V, E)$ is done as follows:

- $\forall p_i \in M_f, \exists$ a vertex $V(p_i) \in G$.
- Edge $E(V(p_i), V(p_j)) \in G$ if and only if p_i is adjacent to p_j or $\exists$ p_k such that both p_i and p_j are adjacent to it.

The second constraint ensures that Constraints 1 and 2 on primitives are satisfied.

Collision Detection

The chromatic decomposition divides the problem of determining self collisions into two distinct parts :

- Non-Adjacent Collision Detection (NACD) between non-adjacent primitives.
- Adjacent Collision Detection (ACD) between adjacent primitives.

Taking advantage of this division, the collision detection proceeds in 4 stages:

Stage 1: For every set S_i, cull away primitives in this set which do not overlap with any non-adjacent primitives using an AABB hierarchy.

Stage 2: In this step, we compute a Potentially Colliding Set (PCS) [1] associated with each independent set of M_f. The non-adjacent primitives within each set S_i, and the non-adjacent primitives between sets S_i and S_j are tested for collision. This is done by performing visibility computations using image-space occlusion queries on the swept volumes of the primitives in these sets. The image precision errors associated with occlusion queries are overcome by adding an offset to the swept volumes.

Stage 3: Perform exact vertex-face (VF) and edge-edge (EE) intersection tests on the union of the Potentially Colliding Sets computed in Stage 2.

Stage 4: Perform exact VF and EE tests between the adjacent primitives.

Multi-Resolution Collision Handling

The multi-resolution scheme has to be designed in such a way that the contact determination and collision response are integrated seamlessly. However, this turns out to be a challenging problem. The multi-resolution hierarchy requires a dynamic update since the particles are changing positions as the simulation proceeds. The positions of the simplified and the refined particles need to be computed on the fly during an upward and downward traversal respectively. This step can be computationally expensive in case of traversal of multiple levels. Also, adjacencies may change in the mesh as the hierarchy is traversed. Triangles that were adjacent in the previous timestep may become non-adjacent, and vice versa. This approach would require chromatic decomposition of the mesh on the fly which can be computationally very expensive.

To overcome these challenges, we maintain the following properties in our multi-resolution scheme:

- *Property 1*: p_i^m is adjacent to $p_j^n \Rightarrow |m-n| \leq 1$
- *Property 2*: If a primitive p_j^m has t children $p_i^{m+1}, i = 1, 2, \ldots, t$, then $refine(p_i^{m+1}) \Rightarrow refine(p_k^{m+1}), k = 1, 2, \ldots, t.\ \ k \neq i$
- *Property 3*: If p_i^m and p_j^n are adjacent to p_k^l then either $m = l$ or $n = l$

These constraints allow for fast simplification based on temporal and spatial coherence, while ensuring smoothness and visual realism. They also ensure that the chromatic decomposition of the mesh can be done as a pre-process, and is not required to be recomputed at every frame. An example of a chromatic-decomposed rectangular mesh hierarchy that follows the above properties is shown in Figure 2. The corresponding hierarchy is shown in Figure 3. Using these multi-resolution constraints is sufficient for handling self-collisions in general deformable models.

Multi-Resolution Collision Detection

In this section, we describe our hierarchy construction, chromatic decomposition pre-process done on this multiresolution hierarchy, and how the constraints help in maintaining the invariants required by [1]

Hierarchy Construction

Our algorithm assumes that the input mesh is a regular subdivision mesh. If the input mesh is not a regular subdivision mesh, the method presented by [39] can be used to approximate the given cloth mesh M, by a regular subdivision mesh M_f, that is within a prescribed tolerance ϵ of M. This process is called remeshing, and the mesh obtained M_f may then be converted into multi-resolution form using any of the well-known techniques for multi-resolution analysis of subdivision meshes. The remeshing step itself proceeds in 3 stages:

Stage 1: Partition M into a number of triangular regions using harmonic maps. Associate with each of the n vertices of this triangulation, a canonical basis vector in R^m. This defines a mesh in R^m called the base complex K^0. A face of K^0 corresponds to a triangular region obtained via the harmonic maps.

Stage 2: Parametrize from the faces of K^0 to the corresponding triangular region. Call it ρ.

Stage 3: Perform J recursive 4-to-1 splits on the faces of K^0 to obtain K^J. Use ρ to map K^J into R^3, and construct M^J.

Figure 4 shows the subdivision hierarchies based on a triangular and a rectangular mesh.

Multi-Resolution Chromatic Decomposition

The hierarchy generated $M^k, k = 1, 2, \ldots, n$, is colored at each level k, using the chromatic decomposition algorithm. We ensure that if a certain color set C_r is used to color M^k, then C_r is not used to color either M^{k+1} or M^{k-1}. This can be achieved by coloring all odd-numbered levels using one color set, say C_s, and all even-numbered levels, using another color set, say C_t, with the following property:

- *Property 4*: $C_s \cap C_t = \emptyset$

The goal of this pre-processing is to ensure that Constraints 1 and 2 are satisfied, independent of the way the hierarchy is traversed during runtime. Any M_f that satisfies Properties

1, 2, 3 and 4 is a valid M_f.

Lemma 1: Any valid M_f satisfies Constraint 1.

Proof: Suppose the lemma does not hold. i.e. $\exists\, p_i^m$ and p_j^n which are adjacent and have the same color. By virtue of the chromatic decomposition algorithm, $m \neq n$. By *Property 4* , $|m-n| \neq 1$. Hence $|m-n| \geq 2$. This violates *Property 1* , and hence M_f is not valid.

Lemma 2: Any valid M_f satisfies Constraint 2.

Proof: Suppose the lemma does not hold. i.e. $\exists\, p_i^m$ and p_j^n, which are not adjacent to each other, but are both adjacent to p_k^l, and have the same color. Using Property 3, $m = l$ or $n = l$. If $m = n$, then due to chromatic decomposition algorithm, both p_i^m and p_j^n should have different colors leading to a contradiction. Therefore, suppose $m \neq n$. By *Property 4* , $|m-n| \neq 1$. Hence $|m-n| \geq 2$. Suppose $|m-n| = 2$. Then, by *Property 1* , $|m-l| = 1$ and $|n-l| = 1$. This contradicts *Property 3* that either $m = l$ or $n = l$. Hence M_f is not valid. (See Figure 5). A similar argument applies for $|m-n| > 2$.

Theorem 1: Any valid M_f satisfies Constraints 1 and 2, irrespective of the way the hierarchy is traversed at runtime.

Proof: Follows from *Lemma 1* and *Lemma 2*.

By virtue of *Theorem 1* , chromatic mesh decomposition at every step is unnecessary, and can be done as a pre-process that maintains Properties 1, 2, 3, and 4.

Collision detection using chromatic decomposition can therefore be applied to multi-resolution meshes with low processing overheads.

We use the 4-stage collision detection algorithm summarized in the overview section [1] in conjunction with the multi-resolution hierarchy presented in this section to perform self-collision detection. As long as the hierarchy is traversed in such a manner that the mesh at the traversal front (M_f) satisfies constraints 1 and 2 at all times, collision detection can be performed using chromatic decomposition.

We believe that our scheme would extend well to general mesh hierarchies i.e. hierarchies constructed from meshes that do not have a subdivision connectivity. As long as the hierarchy traversal satisfies constraints 1 and 2, the same multi-resolution formulation would apply in such a case as well.

Multi-Resolution Collision Response

While the previous section outlined the pre-processing involved to ensure correct collision detection and multi-resolution simulation, we describe the dynamics at runtime in this section. In particular, we first describe our hierarchy traversal scheme, followed by an explanation of the particle system dynamics simplification that occurs as the hierarchy is traversed.

Hierarchy Traversal

At runtime, as the simulation proceeds, certain parts of the cloth mesh may 'gain' detail and become more complex, while others may 'lose' detail and therefore become less complex. We adaptively refine or coarsen parts of the cloth mesh based on the resolution requirements of these parts. Hence, we maintain a "resolution front" in the hierarchy, and selectively update the positions of only those primitives that lie on the front, similar to "front tracking" used in collision detection for rigid objects [40]. The criteria that determine the complexity of parts of cloth mesh may vary with the user application. In particular, we use the following criterion to traverse the mesh :

If p_i is within a proximity ϵ of another primitive p_j, then $refine(p_i)$ and $refine(p_j)$. ϵ is a threshold distance that may be varied as per the needs of the application, and is a measure of the collision proximity of the two primitives.

Particle System Dynamics

Our cloth simulator is particle-system based, and uses verlet integration [41] for computing particle dynamics. As we coarsen the mesh, the number of particles decreases, but the total mass needs to be preserved. Hence the mass assigned to a parent particle is the sum of the masses of the child particles. The same applies when the mesh is refined. The mass is equally distributed among the child particles. We extend the simulation level of detail formulation of [13] to compute the particle system dynamics accordingly. The following procedure is used to calculate the position (P_{new}) and velocity (V_{new}) of the new particle formed by grouping n particles $p_1, p_2, \ldots, p_n$:

$$P_{new} = \frac{\sum_{i=1}^{n} m_i P_i}{\sum m_i}$$

$$V_{new} = \frac{\sum_{i=1}^{n} m_i V_i}{\sum m_i}$$

P_{new} and V_{new} are updated using the standard particle dynamics. When the parent particle is refined back to the child particles, we transfer the new velocities to them. Other material constants like stiffness do not depend on the particle sizes.

Results

All the results presented in this section have been obtained by running our algorithm on these benchmarks on a PC running Windows XP, with a 3.4 GHz processor, an NVIDIA GeForce 6800 graphics card, and 2 GB of RAM. We have implemented and tested our algorithm on a number of complex benchmarks :

- Cloth (33 K triangles) draping over a bunny
- Folding curtains (33 K triangles)
- Squatting human with a pair of pants (10 K triangles)

In our simulations, these benchmarks provide scenarios in which the appropriate level of detail for collision detection and response is adaptively chosen based on collision proximity.

Figure 9 shows the sequence of a piece of cloth draping over a bunny model. The cloth mesh consists of 33 K triangles at the highest resolution level. The parts of the cloth that come into contact with the bunny, or that get wrinkled together need to be simulated at higher resolution to avoid loss of simulation quality. The computation time per frame increases with increasing simulation complexity. Figure 6 presents the corresponding timings as the simulation proceeds.

Figure 10 shows the curtain opening sequence. As the curtains open, close proximity wrinkles appear in them, and hence those parts get refined to higher resolution. The curtains consist of 33K triangles at the highest level of resolution. Figure 7 presents the corresponding timings with increasing frame numbers.

Figure 11 shows the human jumping sequence. As the character squats, wrinkles appear around the knees, and the pants get selectively refined only in that portion. The pair of the pants together consist of about 10K triangles at the highest resolution. Figure 8 shows the associated timings with increasing frame numbers.

Comparison: We achieve a performance gain of 5 to 7 times using our multi-resolution scheme. The exact performance gain from cloth simulation and collision detection using our multi-resolution scheme varies depending on the contact scenarios, but both collision detection and contact response benefit from the multiresolution hierarchy noticeably.

For the cloth falling on a bunny scenario, the overall computation based on our scheme takes 40 - 55 msec per frame, while it takes 250-400 msec using prior methods [1]. The collision detection time increases as the simulation proceeds. In the curtain demo, our algorithm takes 40 - 65 msec of computation time per frame. Without the multi-resolution scheme, this scenario also takes 250 - 400 msec per frame. Finally, in the squatting human sequence, our scheme takes up to 45 msec per frame, while the computation time can take up to 270 msec otherwise.

Since the multi-resolution representation was computed offline, and the updates were made at runtime only to the particles that were at the resolution front, we found the overheads associated with keeping the multi-resolution representation for the deformable mesh to be very low. In most cases, the hierarchy updates took less than 10 msec per timestep.

It is also important to note that by using multi-resolution representations for cloth simulation, the simulation system can maintain a desired simulation rates by adaptively refining and coarsening the meshes. The accompanying video with the paper shows the three sequences as the simulation proceeds. Please note that these sequences are not real-time screen captures, due to high computational costs of rendering.

Analysis and Limitations

In practice, the number of independent sets (obtained after chromatic decomposition) affects the runtime performance of our algorithm. A larger number of sets is undesirable.

Our scheme ensures that the number of sets does not increase considerably due to multi-resolution analysis, and therefore the overhead of multi-resolution remains low. Say the base mesh requires m colors, then the total number of colors required by our scheme would be roughly $m + m/4$. This is because the number of mesh primitives at any level is one-fourth of those at the next lower level, and colors get reused for non-adjacent levels.

Since our approach yields an approximation to the underlying cloth mesh, we compute the error in the simulation induced by successive approximations. For any approximation of a one-dimensional function f, the remainder in the Taylor's formula ($|f''|h^2$) gives the approximation error. For a two dimensional function, f'' is given by the Laplacian of the function. The discrete mean curvature at a vertex is an estimation of the Laplacian in the Voronoi region of the vertex [42]. Hence, error at any vertex is proportional to the discrete mean curvature at that vertex. One may add a curvature based refinement criterion to our scheme (e.g. refine a primitive if the mean curvature at its vertices is greater than a user-specified threshold). That would automatically ensure that the approximation error is never higher than that allowed by the curvature threshold.

In our approach, popping artifacts may arise when the resolution level changes suddenly at any point. However, our refinement constraints ensure that the change in resolution levels is limited to just one level in any timestep, and that significantly reduces these artifacts. Post processing on the geometry may be added to further smooth out popping artifacts.

We have chosen AABB hierarchies, as in the original chromatic decomposition algorithm [1]. However, other bounding volumes, such as K-DOPs, can be used. Similarly, we have implemented a collision response method based on existing techniques [36, 41] in our current simulation system. Other collision response algorithms can be used in its place. We expect to see similar performance gain when using other underlying bounding volumes or collision response algorithms.

Our algorithm using AABB hierarchies has low storage overhead. In practice, the storage requirement for our multiresolution algorithm is at most twice that of an algorithm without multi-resolution hierarchies. The update cost is also significantly lower due to the incremental front updates to take advantages of temporal coherence.

Conclusions

We have presented a multi-resolution approach to handle collisions in cloth like simulations. Our prototype implementation yields significant performance gain over previous methods, while maintaining good visual fidelity.

A possible future direction of research could be to extend the chromatic decomposition algorithm to handle cloth meshes that involve tear. We would also like to explore how this work could be applicable to various other common problems encountered by the cloth animation industry. Post-simulation processing is one such application where our multi-resolution scheme may be extended to achieve significant benefits.

Acknowledgements

This work was supported in part by Defense Advanced Research Projects Agency, Intel Corporation, National Science Foundation, Office of Naval Research, U.S. Army Research Office, and U.S. Army Research Development and Engineering Command. We would like to thank NVIDIA corporation for the hardware and driver support. We are also thankful to members of UNC GAMMA Research Group for useful discussions.

References

[1] N. Govindaraju, D. Knott, N. Jain, I. Kabal, R. Tamstorf, R. Gayle, M. Lin, and D. Manocha. Collision detection between deformable models using chromatic decomposition. *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH)*, 2005.

[2] D. House and D. Breen. *Cloth Modeling and Animation*. A. K. Peters, Ltd., 2000.

[3] P. Schröder and D. Zorin. Subdivision for modeling and animation. *ACM SIGGRAPH Course Notes*, 1998.

[4] M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, L. Raghupathi, A. Fuhrmann, M.-P. Cani, F. Faure, N. Magnenat-Thalmann, W. Strasser, and P. Volino. Collision detection for deformable objects. *Eurographics*, 2004.

[5] G. Debunne, M. Desbrun, M. P. Cani, and A. H. Barr. Dynamic real-time deformations using space and time adaptive sampling. *Proc. of ACM SIGGRAPH*, 2001.

[6] X. Wu, M.S. Downes, T. Goktekin, and F. Tendick. Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes. *Computer Graphics Forum (Proc. of Eurographics)*, 20(3):349–358, 2001.

[7] T. DeRose, M. Kass, and T. Troung. Subdivision surfaces in character animation. *Proc. of ACM SIGGRAPH*, 1998.

[8] S. Capell, S. Green, B. Curless, T. Duchamp, and Z. Popovic. A multiresolution framework for dynamic deformations. *Proc. of ACM SIGGRAPH Symposium on Computer Animation*, pages 41–48, 2002.

[9] E. Grinspun, P. Krysl, and P. Schröder. Charms: A simple framework for adaptive simulation. *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH)*, pages 281–290, 2002.

[10] S. Capell, S. Green, B. Curless, T. Duchamp, and Z. Popovic. Interactive skeleton-driven dynamic deformations. *Proc. of ACM SIGGRAPH*, 2002.

[11] D. L. James and D. K. Pai. Multiresolution green's function methods for interactive simulation of large-scale elastostatic objects. *ACM Trans. on Graphics*, 22(1):47 – 82, 2003.

[12] F. Ganovelli, P. Cignoni, C. Montani, and R. Scopigno. Model manipulation: A multiresolution model for soft objects supporting interactive cuts and lacerations. *Computer Graphics Forum*, 19(3), August 2000.

[13] D. O'Brien, S. Fisher, and M. Lin. Simulation level of detail for automatic simplification of particle system dynamics. *Proc. of Computer Animation*, pages 210–219, 2001.

[14] G. Hirota, R. Maheshwari, and M. Lin. Fast volume-preserving free-form deformation using multi-level optimization. *Proc. ACM Symposium on Solid Modeling and Applications*, 1999.

[15] G. Hirota, R. Maheshwari, and M. Lin. Fast volume-preserving free-form deformation using multi-level optimization. *Computer-Aided Design*, 32(8 & 9):499–512, 2000.

[16] L. Guibas, D. Hsu, and L. Zhang. *H-Walk*: Hierarchical distance computation for moving convex bodies. *Proc. of ACM Symposium on Computational Geometry*, 1999.

[17] S. Ehmann and M. C. Lin. Accelerated proximity queries between convex polyhedra using multi-level voronoi marching. *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2101–2106, 2000.

[18] M. A. Otaduy and M. C. Lin. CLODs: Dual hierarchies for multiresolution collision detection. *Eurographics Symposium on Geometry Processing*, pages 94–101, 2003.

[19] S. Yoon, B. Salomon, M. C. Lin, and D. Manocha. Fast collision detection between massive models using dynamic simplification. *Eurographics Symposium on Geometry Processing*, 2004.

[20] Vasily Volkov and Ling Li. Real-time refinement and simplification of adaptive triangular meshes. *Proc. IEEE Visualization*, 2003.

[21] David Baraff and Andrew Witkin. Large steps in cloth simulation. *Proc. of ACM SIGGRAPH*, pages 43–54, 1998.

[22] D. Breen, D. House, and M. Wozny. Predicting the drape of woven cloth using interacting particles. *Proc. of ACM SIGGRAPH*, pages 365–372, 1994.

[23] S. Huh, D. Metaxas, and N. Badler. Collision resolutions in cloth simulation. *Computer Animation, IEEE*, pages 604–611, 2001.

[24] P. Volino and N. Magnenat Thalmann. Efficient self-collision detection on smoothly discretized surface animations using geometrical shape regularity. *Computer Graphics Forum (EuroGraphics Proc.)*, 13(3):155–166, 1994.

[25] J. Mezger, S. Kimmerle, and O. Etzmuβ. Hierarchical techniques in cloth detection for cloth animation. *Journal of WSCG*, 11(1), 2003.

[26] X. Provot. Collision and self-collision handling in cloth model dedicated to design garment. *Graphics Interface*, pages 177–189, 1997.

[27] P. Volino and N. Magnenat Thalmann. Accurate collision response on polygon meshes. *Computer Animation*, 2000.

[28] R. Bridson, S. Marino, and R. Fedkiw. Simualtion of clothing with folds and wrinkles. *Eurographics/SIGGRAPH Symposium on Computer Animation*, 2003.

[29] D. Baraff. *Dynamic simulation of non-penetrating rigid body simulation*. PhD thesis, Cornell University, 1992.

[30] J. Cohen, M. Lin, D. Manocha, and M. Ponamgi. I-COLLIDE: An interactive and exact collision detection system for large-scale environments. In *Proc. of ACM Interactive 3D Graphics Conference*, pages 189–196, 1995.

[31] F. Cordier and N. Magnenat-Thalmann. Real-time animation of dressed virtual humans. *Computer Graphics Forum*, 21(3), 2002.

[32] A. Fuhrmann, C. Gross, and V. Luckas. Interactive animation of cloth including self collision detection. *Journal of WSCG*, 11(1), 2003.

[33] Y. Kang and H. Cho. Bilayered approximate integration for rapid and plausible animation of virtual cloth with realistic wrinkles. *Computer Animation*, pages 604–611, 2002.

[34] M. Meyer, G. Debunne, M. Desbrun, and A. Barr. Interactive animation of cloth like objects in virtual reality. *Journal of Visualization and Computer Animation*, 12(1):1–12, 2000.

[35] D. Baraff, A. Witkin, and M. Kass. Untangling cloth. *Proc. of ACM SIGGRAPH*, pages 862–870, 2003.

[36] R. Bridson, R. Fedkiw, and J. Anderson. Robust treament for collisions, contact and friction for cloth animation. *Proc. of ACM SIGGRAPH*, 2002.

[37] K. Choi and H. Ko. Stable but responsive cloth. *Proc. of ACM SIGGRAPH*, pages 604–611, 2002.

[38] S. W. Choi and H.-P. Seidel. Linear onesided stability of MAT for weakly injective 3D domain. In *Proc. ACM Symposium on Solid Modeling and Applications*, pages 344–355, 2002.

[39] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Multiresolution analysis of arbitrary meshes. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 173–182. ACM SIGGRAPH, Addison Wesley, August 1995. held in Los Angeles, California, 06-11 August 1995.

[40] M. A. Otaduy and M. C. Lin. Sensation preserving simplification for haptic rendering. *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH)*, pages 543–553, 2003.

[41] H. Gould and J. Tobochnik. *An Introduction to Computer Simulation Methods*. Addison Wesley, 1988.

[42] N. Molino, R. Bridson, J. Teran, and R. Fedkiw. A crystalline, red green strategy for meshing highly deformable objects with tetrahedra. *Proc. 12th International Meshing Roundtable*, 2003.

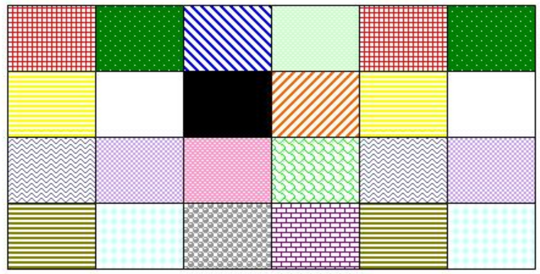

Figure 1: *Chromatic decomposition of a rectangular mesh. No two adjacent primitives have the same color.*

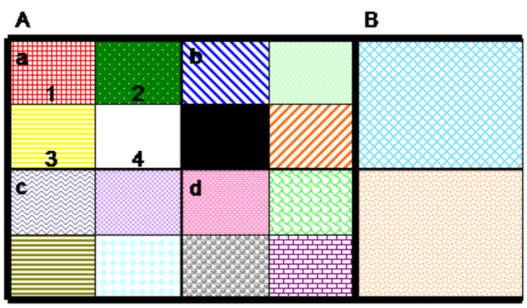

Figure 2: *Chromatic decomposition of a rectangular mesh having several resolution levels. The color sets used at level i and level i+1 are disjoint*

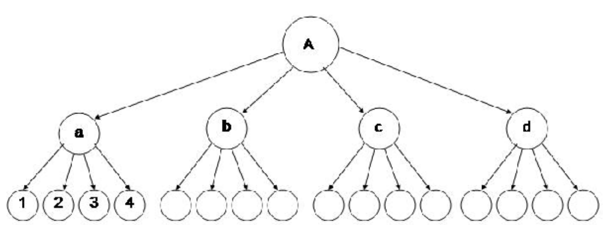

Figure 3: *The hierarchy corresponding to the multi-resolution rectangular mesh*

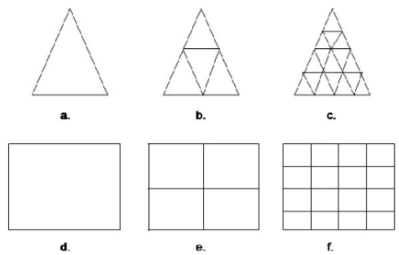

Figure 4: *The subdivision hierarchies for a triangular mesh (a-c), and a rectangular mesh (d-f) formed by recursive 4-to-1 splits*

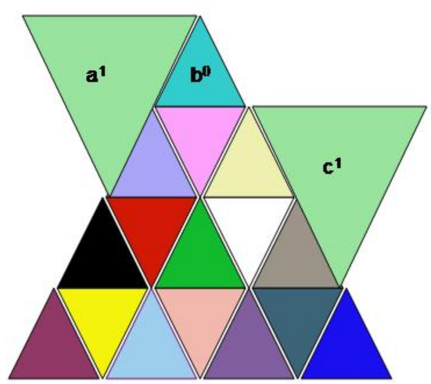

Figure 5: *Primitives* a^1 *and* c^1 *are adjacent to* b^0*, and satisfy Properties 1-4. However, such a configuration cannot arise in a mesh having subdivision connectivity*

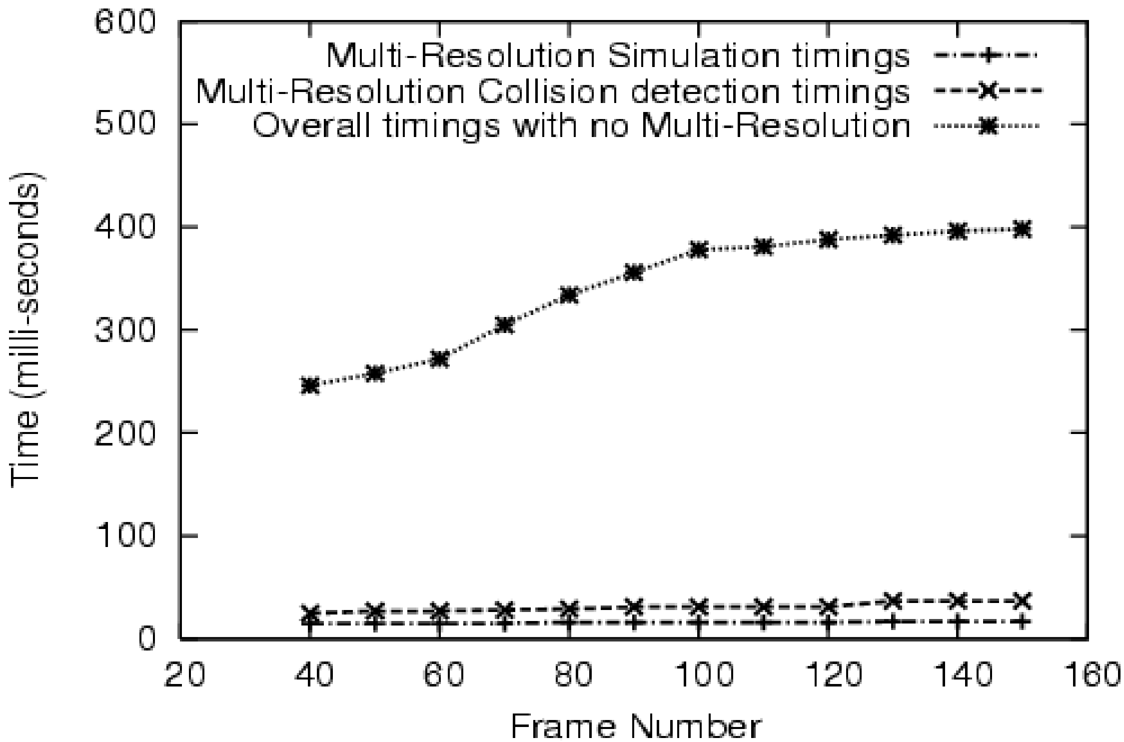

Figure 6: *Simulation and collision detection timings on the cloth draping over a bunny scenario.*

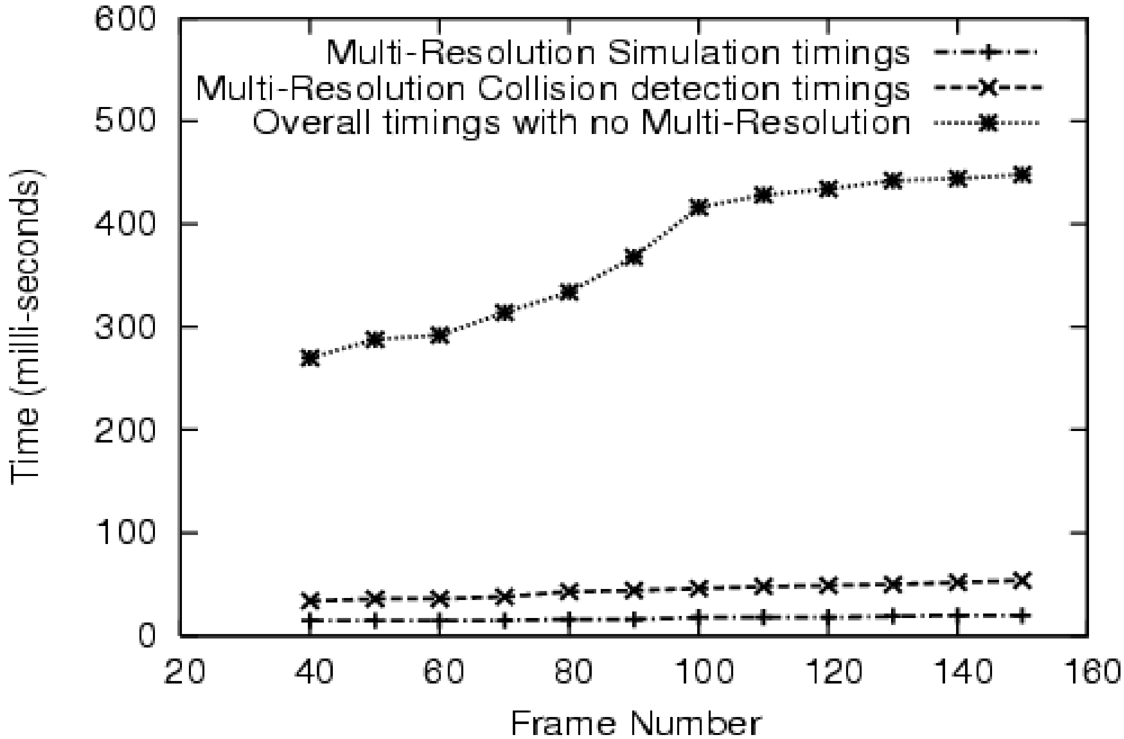

Figure 7: *Simulation and collision detection timings on the folding curtains model*

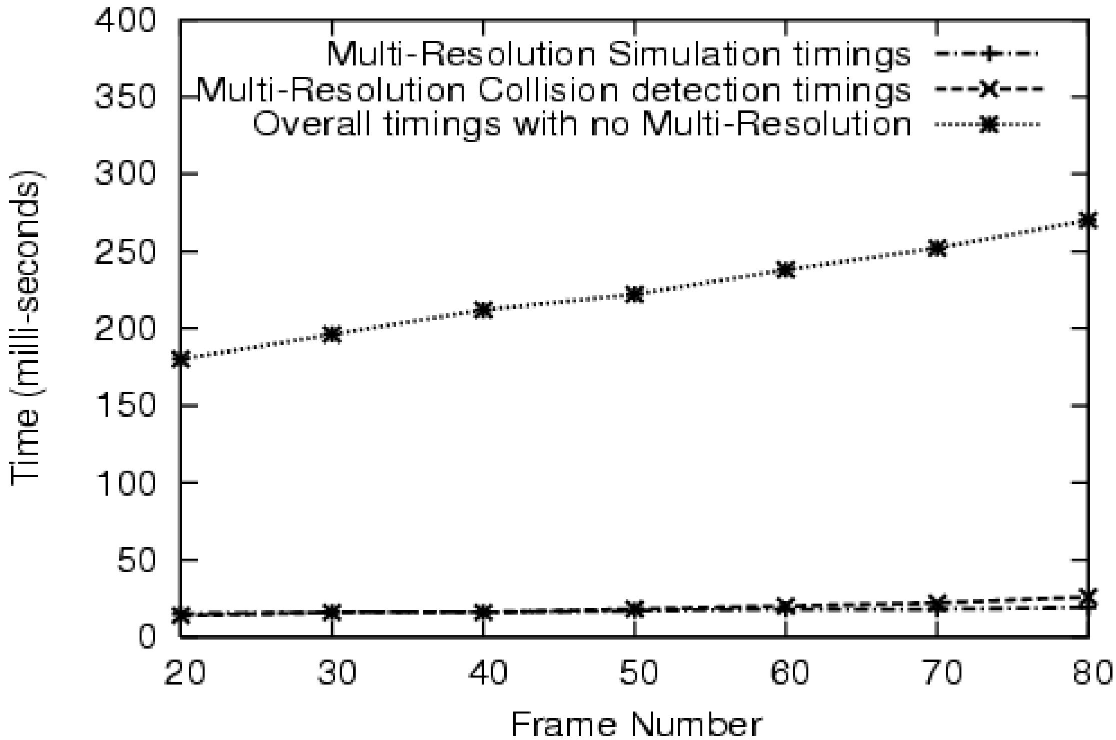

Figure 8: *Simulation and collision detection timings on the squatting human model*

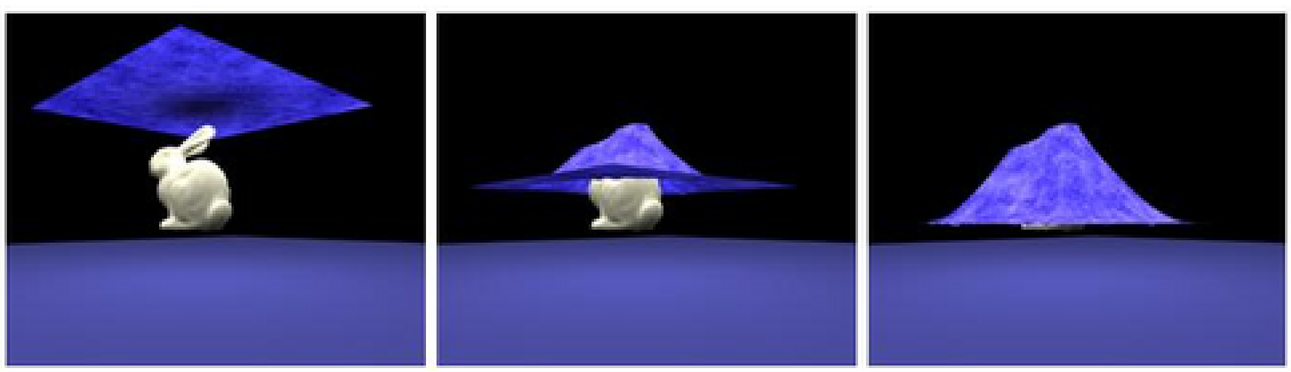

Figure 9: *A simulation sequence taken from the cloth (33K tris) draping simulation. Close proximity wrinkles appear as the cloth drapes over the bunny.*

Figure 10: *A simulation sequence taken from the opening curtain (33K tris) simulation. Close proximity wrinkles appear as the curtains fold.*

Figure 11: *A simulation sequence taken from a jumping virtual human simulation. Close proximity wrinkles appear on the pants (10K tris), as the character squats.*