# Cloth Animation with Adaptively Refined Meshes

**Ling Li and Vasily Volkov**
Department of Computing
Curtin University of Technology
GPO Box U1987 Perth, WA 6845

{ling|vasily}@cs.curtin.edu.au

## Abstract

Cloth animation is a very expensive process in terms of computational cost, due to the flexible nature of cloth objects. Since wrinkles and smooth areas co-exist commonly in cloth, it is tempting to reduce computational cost by avoiding redundant tessellation at the smooth areas. In this paper we present a method for dynamic adaptation of triangular meshes suitable for cloth simulation. A bottom-up approach is used for mesh refinement, which does not require precomputation and storage of multiresolution hierarchy. The hierarchy is constructed in runtime and allows reverting of the refinement locally. Local mesh refinement and simplification are triggered by curvature-induced criterion, where the curvature is estimated using methods of discrete differential geometry. The results presented are the realistic animation of garment worn by a walking mannequin generated with Baraff-Witkin type cloth solver enhanced with the mesh adaptation scheme.

***Keywords:*** Cloth Animation, Refinement and Simplification, Adaptive Mesh

## 1. Introduction

Cloth animation has received intensive attention in computer graphics in the last twenty years. Significant progress has been made in realistic cloth animation. Still it remains a computationally demanding task. It is naturally desirable to improve the performance and efficiency of cloth animation systems. Such improvement is often achieved at the cost of realism, using one or another simplification on the physical model or the integration method. In this paper an approach is proposed to improve the efficiency of cloth simulation system without degrading the simulation realism. It can be directly applied in the most elaborated cloth simulation techniques as an additional component.



The computational cost of cloth simulation directly depends on the mesh resolution, which determines the fineness of cloth details to be captured, i.e. wrinkles. The majority of existing cloth simulation methods relies on uniform resolution meshes, though geometric details are not at all distributed uniformly across a garment as shown in Fig. 1. Distributing mesh nodes over the cloth surface according to the local detail level could significantly reduce computational cost.

(a) (b)

Figure 1. Curvature (as represented by different colours in (b)) varies notably across a garment.

The approach of using adaptive mesh to improve the performance of cloth simulation attracted attention of researchers for more than a decade. Still, all existing adaptive approaches (Hutchinsin and Hewitt 1996, Thingvold and Cohen 1987, Villard and Borouchaki 2002) are severely limited compared with the state-of-art in the non-adaptive simulation (Baraff and Witkin 1998, Choi and Ko 2002, Volino et al. 1995). The limitations include: explicit integration, regular grid, spring-mass physics, and application in only the simplest simulations, such as the draping of a tablecloth. Besides, all the existing adaptive mesh algorithms include only the refinement, not the simplification, and their refinement criteria are simply tied to angle in an ad hoc manner (Hutchinsin and Hewitt 1996, Villard and Borouchaki 2002). Still, significant improvement in system performance was reported (Hutchinsin and Hewitt 1996).

Adaptive mesh scheme has been used commonly in simulation of 3D deformable objects (Debunne et al. 2001, Wu et al. 2001). Notably, mesh refinement algorithms are well-developed in view-dependent visualization area, e.g. terrain visualization (Duchaineau et al. 1997, Lindstrom and Pascucci 2001, Hoppe 1998), where high performance methods capable of maintaining continuous and intensively re-adapted triangular meshes are essential. Wu et al. were probably the first to apply these techniques to the field of deformable object simulation.

In this paper an algorithm is reported that introduces adaptive meshes into the most elaborated cloth simulation models based on irregular triangular meshes (Baraff and Witkin 1998, Eischen et al. 1996, Etzmuß et al. 2003, Volino et al. 1995 ). Irregular meshes are advantageous in cloth simulation, since they impose less restriction on the mesh boundaries. Our contribution is twofold. Firstly, a high performance method for mesh adaptation is presented. Given a coarse irregular triangular mesh as input, √3-refinement is used to locally adapt mesh resolution following a refinement criterion. The generated semi-regular mesh can be directly used in a standard triangle-based cloth simulation system. A history of refinement operations is maintained in a hierarchic structure to allow reversing the refinement locally. Secondly, more systematic approach is used to derive the refinement criterion. Measure of adequacy of the current local resolution against the local detail level is related to local curvature. Methods of discrete differential geometry are used to evaluate the mean curvature over the mesh. The mixed finite element / finite volume derivations of curvature estimation by Meyer et al. 2003 are extended to the case of triangular mesh with boundary.

The proposed adaptive mesh scheme is tested in the most typical, though challenging, cloth simulation set: animation of garment worn by a walking mannequin. It is the first time that adaptive cloth simulation is employed in such complicated scenario with folding and unfolding of complex wrinkle patterns. The simulation results demonstrate high realism at reduced computational cost.

The rest of the paper is organized as follows. Section 2 reviews issues in cloth simulation, mesh adaptation and refinement criteria. The mesh adaptation algorithm is explained in Section 3. Section 4 discusses the refinement criterion and evaluation of mean curvature on triangular mesh with boundary. Results are discussed in Section 5 while Section 6 concludes this paper.

## 2. Related Work

### 2.1 Cloth Simulation

Appropriate equations of continuum mechanics has been used for cloth simulation, first in variational form, then reduced to PDE (Terzopolous et al. 1987) and then spatially discretized into ODEs. Classical discretization methods are finite differences (Terzopolous et al. 1987) and finite elements (Eischen et al. 1996, Etzmuß et al. 2003 ). In practice, ad hoc discretization methods have gained popularity (Baraff and Witkin 1998, DeRose et al. 1998, Provot 1995, Volino et al. 1995). They are not strictly and consistently derived from continuous equations. Instead they are stated directly in the discrete form. The popular spring-mass networks are reminiscent of finite difference methods: “stretch” springs connect adjacent points is a five-point stencil for discrete approximation of Laplacian and “flexion” springs is the wider stencil for the forth derivative approximation. Spring-mass method inherits restrictions of finite differences, i.e., plausible results are only provided for regular grids. The Baraff-Witkin approach (Baraff and Witkin 1998, for example, is based on irregular triangular meshes and is reminiscent to finite element approach. Notable issues in the physical models include preventing “super-elasticity” effect from linear elastic model (Provot 1995), buckling etc. (Choi and Ko 2002, Eischen et al. 1996, Feynman 1986).

The resulted ODE are stiff and it became standard to solve them using implicit method, which was first used by Terzopoulos et al. 1987, but became widespread only after the work by Baraff and Witkin 1998. Among recent contributions are precomputing implicit Euler matrix inverse (Desbrun et al. 1997) and Gear’s method (or BDF) (Choi and Ko 2002).

### 2.2 Mesh adaptation

Two important considerations for adaptive algorithms are: good mesh quality in terms of triangle aspect ratio, and the ability to locally reverse refinement. Mesh adaptation methods can be categorized into dealing with regular, irregular and semi-regular, i.e. regularly subdivided irregular meshes.

Regular meshes are the simplest but the most restrictive solution. Adaptive regular mesh is a collection of regular meshes of different resolution joined together. Topological restrictions produce cracks on the interface between different resolutions – the so-called T-vertices. Another problem is the poor approximation of the domain boundary, if it is not rectangular. This approach was used in the previous attempts of adaptive cloth simulation (Hutchinsin and Hewitt 1996, Villard and Borouchaki 2002).

Adaptive irregular meshes are less restrictive, and they produce continuous meshes. They are constructed in top-down approach, pre-computing the multiresolution hierarchy via simplification of the finest mesh down to the coarsest state. The pre-computed hierarchy requires considerable space for storage. Examples are progressive meshes (Hoppe 1998, Xia and Varsheney 1996) and Dobkin-Kirkpatrick meshes (De Berg and Dobrindt 1998, Lee et al. 1998) . Progressive meshes are not designed to produce the mesh of good quality and they usually do not, though applications to deformable object simulations exist (Wu et al. 2001). Dobkin-Kirkpatrick meshes are Delauney optimal (De Berg and Dobrindt 1998) . The hierarchy in their case is a number of uniform resolution irregular meshes, which are combined in runtime to get the adapted mesh. Debunne et al. (2001) relied on the same idea, but managed to simulate deformable models even without combining the meshes into a single conforming mesh.

Adaptive semi-regular meshes enjoy simplicity of the regular meshes with robustness of their irregular counterpart. The hierarchy is constructed when necessary in bottom-up fashion using the refinement rules. Hence no precomputation and extensive storage is required. The generated meshes have good mesh quality, provided that the coarsest mesh is good enough. Classical variations include red-green refinement based on 1-to-4 split (Azuma et al. 2003, bank et al. 1983, Wood et al. 2000) and bintree meshes based on 1-to-2 split (Duchaineau et al. 1997, Velho and Zorin 2001). The coarsest resolution of bintree mesh is required to consist of pairs of right triangles sharing hypotenuse, though every triangular mesh can be converted into the bintree mesh doubling triangle count (and Zorin 2001). Volkov and Li (2003) describe a general method which can be used with a variety of regular refinement rules, including √3-subdivision (Kobbelt 2000), which is the *slowest*, in terms of resolution change per refinement pass. Moreover, √3-split is more local, than, for example, the next slowest 1-to-4 split – sharp resolution gradients are possible without introducing excessively slivery triangles (Kobbelt 2000). Allez et al. (2003) proposed a hierarchy-less approach to reversible √3-refinement. Naturally, their linear history of refinement operations allows working only in the FILO style (first-in last-out). For example, to simplify the first refined triangle, one must simplify all and then refine them all except the first again.

A totally different approach in adaptation is to refine the finite element functional space instead of the mesh (Grinspun et al. 2002).

## 3. Adaptive mesh

In order to enable the reverting of the refinement operations, history of refinement is maintained. In order to revert in an arbitrary spatial order, the history is stored in a hierarchical fashion. Hierarchy is the core structure for mesh adaptation. In our method, all local refinement and simplification operations affect the hierarchy first, and then the hierarchy is converted into a conforming triangular mesh. The hierarchy update is temporally coherent, i.e. little changes in geometrical shape of cloth result in a few hierarchy adjustment operations. Export of hierarchy to conforming mesh is not temporally coherent, but simple and computationally inexpensive.

### 3.1 Hierarchy Data Structures

The hierarchy nodes are understood as the triangles arising in the refinement process. The root nodes are given as input to the algorithm and form the coarsest triangulation. All other nodes are constructed in runtime using procedural refinement rule as described later in 3.2. Nodes of depth $i$ compose $i$-th resolution triangulation $M_i$ as shown in Fig. 2. $M_i$ will be referred in the paper as *i-th resolution layer*. $M_0$ is the coarsest layer. Parent-child links are defined between triangles at different layers, as specified by the refinement rule.

Since all higher resolution layers can be reconstructed from the coarsest layer at any moment using refinement rule, it is not necessary to store all of them permanently, which would require an abundant space. Instead, only the required parts of the hierarchy and the associated vertices are stored, as dictated by the refinement criterion.

In order to make the reconstruction and deconstruction of the hierarchy efficient, robust data structures are used to store vertices and triangles of hierarchy. The standard memory allocation approach is employed taking into account that the stored elements are small and of the same type (hierarchy triangles in one container and vertices in another). Elements are stored in an array, where some of its cells may not be occupied. Additional array is used to list the unoccupied cells. Insertion/removal operations pop or push cell indices from/to this list. In this way, insertion and removal do not invalidate references by index in the main array, still working in O(1) time. When there is no free cell and new element is to be inserted, arrays are resized by 100% to make such memory reallocations rare. In practice only 30~70% of the allocated memory is wasted.

A couple of such dynamic arrays are used: one for vertices, and one for each resolution layer.

### 3.2 Ö3-refinement rule

Refinement rule is a procedure for reconstructing resolution level $M_{i+1}$ from $M_i$. In √3-refinement rule, new vertices are inserted into the face centers. The face centers of two neighboring triangles and one of their common vertices produce a higher resolution triangle. The triangles incident to the child vertex of triangle $T \in M_i$ are considered as children of $T$, as shown in Fig. 2. Note that each generated child triangle has two parents. Moreover, neighboring non-root triangles always have one common parent. Refinement rule procedurally generates child vertices, child triangles and neighborhood links among them, which are needed for further finer refinement.

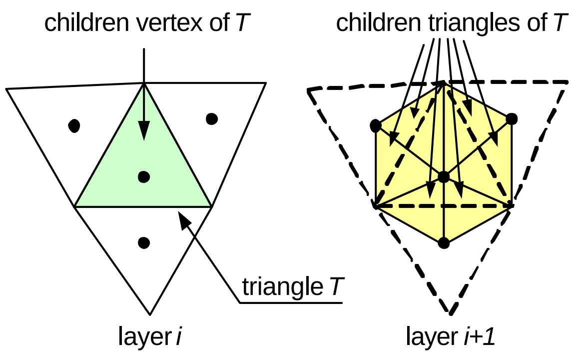


Fig. 2. Construction of the finer resolution layer with √3-refinement rule.

Whenever a triangle $T$ is refined, it is first ensured that all its neighbors exist. If they are not existent, they are created by refining their parent at the coarser level. This forced refinement may recursively invoke refinement at even coarser levels. When the concerned triangle is on the boundary, the neighboring triangle can not exist. This special case will be discussed later in 3.3.

When the criterion indicates that the triangle should be refined, all its children have to be created. The most recent criterion value is stored for each triangle and is updated whenever refinement or simplification is performed. When the criterion flips to negative, the redundant children are removed, i.e., the children that do not have another parent with positive criterion. It may happen that some of the children to be removed have children in turn. In this case, the simplification is skipped.

The set of criterion decisions for each triangle at $M_i$ completely determine which part of $M_{i+1}$ is reconstructed, as shown in Fig. 3.

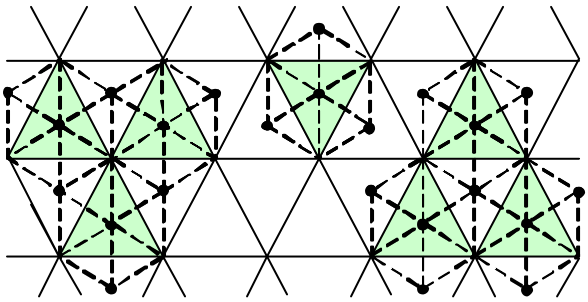

Fig. 3. Refinement state of the triangles (marked with color) uniquely defines what part of the higher resolution (dashed) is reconstructed.

## 3.3 Domain boundary

Consider a boundary triangle at layer $M_i$. When new vertex cannot be inserted into the neighbor's center due to the absence of the neighbor, it is inserted into the edge in the way to produce two right children triangles, as shown in Fig. 4. It is better than inserting it into the middle of the edge, which produces one obtuse-angle triangle, if the mesh is not perfectly regular. In either case, these triangles have poorer aspect ratios than the regular children. To avoid further decrease in the mesh quality, boundary at $M_{i+2}$ is constructed to simulate 1-to-9 split of $M_i$, just as in Kobbelt (2000).

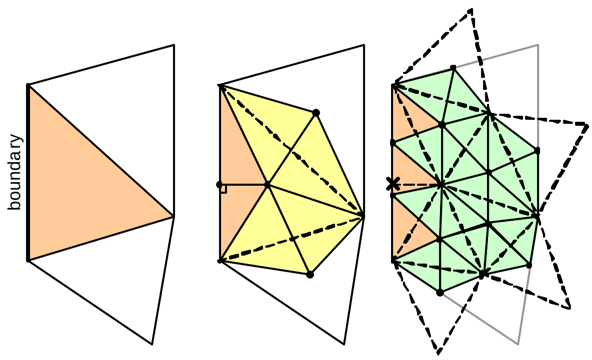


Fig. 4. Boundary triangles on even resolution layers are constructed in an alternative fashion.

## 3.4 Extracting conforming mesh

Having updated the hierarchy, the conforming triangulation is built, which is based on the same vertices as used in the resolution layers. The conforming mesh is constructed in a direct and strict manner similar to the red-green refinement. The hierarchy triangles with no children and hence having no finer representation contribute directly. Those with the full set of children have finer representation at layers above and do not contribute. Others are lying at the interface between resolutions and are triangulated conformingly. This triangulation is as simple and strict as the refinement rule as shown in Fig.6. The triangle count in the resulting conforming mesh is in practice 25% below the total number of triangles in the hierarchy.

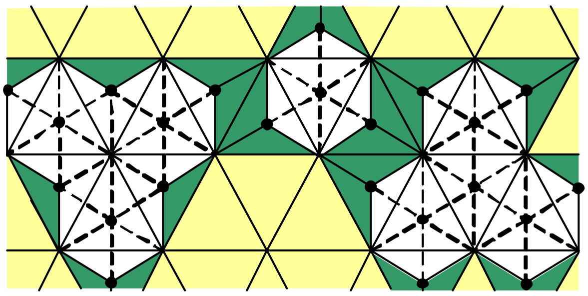

Fig. 5. Conforming mesh for the two hierarchy layers of Fig. 3. The colored triangles are the contribution of the coarser layer. White regions are recursively processed on the next layer.

For the conforming triangulation, neighborhood links and 1-ring triangles for each vertex are extracted. There is no need to store the entire list of all 1-ring triangles for each vertex. Instead only one of them is stored, while the rest are obtained walking around the vertex using the neighborhood links.

## 3.5 Physical properties

Cloth mesh has more attributes than only the geometry. These attributes depend on the physical model used. Here we discuss what properties must be assigned to the mesh used with Baraff-Witkin model (1998). Being FEM-like, it has attributes common for all solvers based on irregular meshes.

Compared to the ordinary mesh, cloth mesh is enhanced with vector velocities, material coordinates and masses. When inserting vertex into the center of triangle ($i$,$j$,$k$), this vertex is also assigned the velocity of the triangle center $(\mathbf{v}_i+\mathbf{v}_j+\mathbf{v}_k)/3$, and material coordinates of the center. Similarly, boundary cases are considered, where vertex is inserted onto an edge.

Masses of the vertices are assigned the masses of the associated surface patch – the Voronoi cell of the vertex. Formula for computing the area of the Voronoi cell can be found in Meyer et al. (2003), scaled by the cloth density to give the mass. Note that cloth density is given in the material coordinates, which are therefore more appropriate for the calculation of the Voronoi cell area. Density in world coordinates changes with stretch, though insignificantly.

All other properties, such as stiffness, are not mesh specific and independent of the resolution. Usually, physical models for elastic objects rely on real mechanical quantities, such as Young modulus of Poisson coefficient. Though Baraff-Witkin model does not use such considerations, it is similar in spirit. Hence no adjustments are needed for stiffness and damping parameters.

## 4. Results

The adaptive refinement was incorporated into a cloth simulation system which include the following components: Baraff-Witkin's physics and integration, voxel-based cloth-cloth proximity detection (Eischen and Bigliani 2000), hierarchical bounding boxes for cloth-rigid proximity detection (Eischen and Bigliani 2000), and the collision response by Volino et al. (1995). Adaptation was performed before each simulation step. The performance metrics were measured on a computer with an Intel Pentium 4 CPU 2.8GHz.

A few 10-second simulations of a dress worn on a walking mannequin are produced to demonstrate the advantage of incorporating adaptation into the cloth simulation system. The coarsest triangulation $M_0$ used as input consists of 472 triangles. The refinement is restricted to 5 resolution layers. The mesh at its finest resolution $M_4$ consists of 38,232 triangles. Construction of the entire hierarchy up to the finest layer $M_4$ from $M_0$ required 40ms, while deconstruction (simplification) required only 11ms. Extraction of the finest conforming mesh took 13ms. Adaptation time constituted typically only 7-8% of the simulation time. For example, one simulation step in the adaptive simulation with ~12000 triangles took 1.2 sec on average, while the adaptation was done in only 87ms per step.

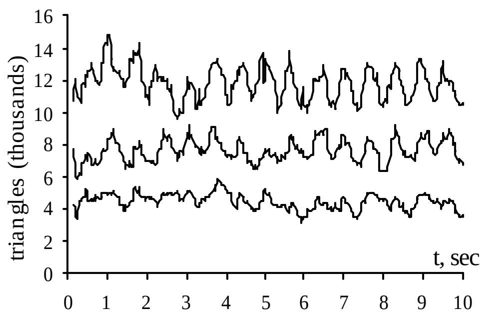


Fig. 6. Triangle count in the three different adaptive cloth simulations.

Fig. 6 illustrates the change of triangle numbers in adaptive simulations with different thresholds on local approximation error. Thresholds have been chosen to maintain the average triangle count at around 4000, 8000 and 12000.

Fig. 7. Mesh is dynamically adapted along the animation following the changes in deformation.

Fig. 7 shows the mesh adaptation in action. The resolution is increased when wrinkles formed and sharpened and is decreased later, when they are unfolded.

Fig. 8 compares the results generated using uniform and adaptive meshes, obtained with different thresholds. It can be seen that the adaptive method may produce much sharper creases at lower computational cost. It was also observed that adaptive refinement inhibits minor wrinkles. Regions of cloth holding minor wrinkles are simplified, which prevents their development into sharper ones. At the same time, sharp creases, always represented with finer mesh, tend to become even sharper.

Uniform 4228 triangles

adaptive 4863 triangles

uniform 12924 triangles — adaptive 12681 triangles

Fig. 8. Snapshots of the non-adaptive and adaptive simulations.

## 5. Conclusion

An elaborated mesh adaptation system designed for cloth simulation is presented. It operates on semi-regular meshes, which enjoy the benefits of both irregular and regular meshes: imposing very little restrictions on the mesh boundary and at the same time very simple and computationally efficient.

Two vital components were explained: the estimation of the adequate local resolution and the subsequent mesh refinement and simplification. Mesh refinement algorithm deals with hierarchy of resolution layers – uniform meshes of different resolutions. The coarsest layer of the hierarchy is given as input, the rest is reconstructed in runtime using √3-refinement rule. Redundant parts of the hierarchy are deconstructed to save the memory space. The resolution layers are converted into a conforming mesh using a straightforward and fast algorithm

The adaptation is driven by the refinement criterion, which estimates local approximation error using discrete mean curvature at the mesh vertices. The formulae for curvature estimation were derived and the dependence of the approximation error on the triangle size was discussed.

Results include animation of a dress worn by a walking mannequin, which was never before demonstrated in the adaptive cloth simulation. Mesh adaptation overhead was only 7-8% of the typical cloth simulation step. Behavior of cloth at the adaptive simulations was natural and realistic, generally demonstrated sharper and more detailed wrinkles comparing to the non-adaptive simulation with similar triangle count. However it was observed that the system inhibits some wrinkles, preventing minor bending deformations. At the same time, sharp creases tend to became even sharper. Therefore our approach should be distinctively advantageous where minor bending deformations are not common, i.e. when bending stiffness is high and cloth is forced to bend into the sharp creases by strong external forces, such as at the elbows of a jacket. It is unclear at this stage whether this artifact should be attributed to the adaptation technique or the deficiency of the physical model used. We believe that introducing buckling into the physical model may ameliorate the problem.

The whole idea of adaptive meshes is to reduce computational cost while preserving the quality of simulation. We have obtained realistic simulations with reduced triangle count, although as common to cloth simulation problem, there is no quantitative method to numerically measure the realism Compared to previous adaptive mesh approaches in cloth simulation, ours is significantly more elaborated and consistent. Many advanced techniques in various different fields have been used, such as methods of discrete differential geometry or view-dependent visualization, to construct an approach as much mature as possible.

The approach proposed is transparent to the cloth simulation method. Therefore any FEM, FVM or ad hoc method based on linear triangular elements can be augmented with the presented adaptation system.

**REFERENCES**


Alliez, P., Laurent, N., Sanson, H. and Schmitt, F. (2003), Efficient view-dependent refinement on 3D meshes using √3-subdivision. *The Visual Computer*, vol. 19, no. 4, pp. 205–221.

Azuma, D. I., Wood, D. N., Curless, B., Duchamp, T., Salesin, D. H. and Stuetzle, W. (2003), View-dependent refinement of multiresolution meshes with subdivision connectivity. *Proceedings of AFRIGRAPH 2003*, ACM Press, pp. 69–78.

Bank, R.E, Sherman, A.H. and Weiser, A. (1983), Refinement Algorithms and Data Structures for Regular Local Mesh Refinement. *Scientific Computing*, R. Stepleman et al., ed., vol. 44, IMACS North-Holland, Amsterdam, 1983, pp. 3–17.

Baraff, D. and Witkin, A. (1998), Large Steps in Cloth Simulation. *SIGGRAPH '98 Conference Proceedings*, ACM Press, pp. 43–54.

Choi, K.-J. and Ko, H.-S. (2002), Stable but responsive cloth. *ACM Transactions on Graphics*, vol. 21, no. 3, Proc. ACM SIGGRAPH 2002, pp. 604–611.

De Berg, M. and Dobrindt, K. T. G. (19980, On levels of details in terrains. *Graphical Models and Image Processing*, vol. 60, no. 1, pp. 1–12.

Debunne, G., Desbrun, M., Cani, M.-P. and Barr, A.H. (2001), Dynamic real-time deformations using space & time adaptive sampling. *SIGGRAPH 2001 Conference Proceedings*, pp. 31-36.

DeRose, T., Kass, M., and Truong, T. (1998), Subdivision surfaces in character animation. *SIGGRAPH '98 Conference Proceedings*, pp. 85–94.

Desbrun, M., Schröder, P. and Barr, A. (1999), Interactive animation of structured deformable objects. *Proc. Graphics Interface '99*, pp. 1–8.

Duchaineau, M.A., Wolinsky, M., Sigeti, D.E., Miller, M.C., Aldrich, C., and Mineev-Weinstein, M.B. (1997), ROAMing terrain: real-time optimally adapting meshes. *Proc. IEEE Visualization '97*, pp. 81–88.

Eischen, J. and Bigliani, R. (2000), Collision detection in cloth modelling. In D. House and D. Breen, editors, *Cloth Modeling and Animation*, pp. 196–218, A.K. Peters.

Eischen, J.W, Deng, S. and Clapp T.G. (1996), Finite-element modeling and control of flexible fabric parts. *IEEE Computer Graphics and Applications*, vol. 16, no. 5, pp.71–80.

Etzmuß, O., Keckeisen, M. and Straßer, W. (2003), A fast finite element solution for cloth modelling. *Proc. Pacific Graphics 2003*, pp. 244–251.

Feynman, C.R. (1986), Modeling the appearance of cloth. *Master's thesis*, Massachusetts Institute of Technology, Available at: http://theses.mit.edu

Grinspun, E., Krysl, P. and Schröder, P. (2002), CHARMS: a simple framework for adaptive simulation. *ACM Transactions on Graphics*, vol. 21, no. 3, Proc. ACM SIGGRAPH 2002, pp. 281–290.

Hoppe, H. (1998), Smooth view-dependent level-of-detail control and its application to terrain rendering. *Proc. IEEE Visualization* 1998, pp. 35–42.

Hutchinson, D., Preston, M. and Hewitt, T. (1996), Adaptive refinement for mass/spring simulations. *Proceedings of the European workshop on Computer animation and simulation '96*, pp. 31–45.

Kobbelt, L. (2000). v3-subdivision. *SIGGRAPH 2000 Conference Proceedings*, pp. 103–112.

Lee, A. W. F., Sweldens, W., Schröder, P., Cowsar, L. and Dobkin, D. (1998), MAPS: Multiresolution Adaptive Parameterization of Surfaces. *SIGGRAPH '98 Conference Proceedings*, pp. 95–104.

Lindstrom, P. and Pascucci, V. (2001), Visualization of large terrains made easy. *Proc. IEEE Visualization 2001*, pp. 363–370, 574.

Meyer, M., Desbrun, M., Schöder, P. and Barr A.H. (2003), Discrete differential-geometry operators for triangulated 2-manifolds. In H.-C. Hege and K. Polthier, editors, *Visualization and Mathematics III*, pp. 35–57, Springer-Verlag, Heidelberg.

Provot, X. (1995), Deformation constraints in a mass-spring model to describe rigid cloth behaviour. *Proc. Graphics Interface '95*, pp 147–154.

Terzopolous, D., Platt, J., Barr, A. and Fleischer K. (1987), Elastically deformable models. *SIGGRAPH '87 Conference Proceedings*, pp. 205–214.

Thingvold, J., and Cohen, E. (1987), Physical modeling with B-spline surfaces for interactive design and animation. *Computer Graphics*, vol. 24, no. 2, pp. 129–137.

Velho, L. and Zorin, D. (2001), 4-8 Subdivision, *Computer Aided Geometric Design*, vol. 18, no. 5, pp. 397–427.

Villard, J. and Borouchaki, H. (2002), Adaptive meshing for cloth animation. *Proc. 11th International Meshing Roundtable*, pp. 243–252.

Volino, P., Courchesne, M. and Thalmann, N. M. (1995), Versatile and efficient techniques for simulating cloth and other deformable objects. *SIGGRAPH '95 Conference Proceedings*, pp. 137–144.

Volkov, V. and Li, L. (2003), Real-time refinement and simplification of adaptive triangular meshes. *Proc. IEEE Visualization 2003*, pp. 155–162.

Wood, Z. J., Desbrun, M., Schröder, P. and Breen, D. (2000), Semi-regular mesh extraction from volumes. *Proc. IEEE Visualization 2000*, pp. 275–282.

Wu, X., Downes, M.S., Goktekin, T. and Tendick, F. (2001), Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes. *Proc. Eurographics 2001*, pp. 349–358.

Xia, J. C. and Varshney, A. (1996), Dynamic view-dependent simplification for polygonal models. *Proc. IEEE Visualization '96*, pp. 32