Stable Cloth Animation with Adaptive Level of Detail

Fernando Pedro Birra
Department of Informatics
Faculty of Sciences and Technology
New University of Lisbon
Portugal, P-2829-516 Monte Caparica
fpb@di.fct.unl.pt

Manuel Próspero dos Santos
Department of Informatics
Faculty of Sciences and Technology
New University of Lisbon
Portugal, P-2829-516 Monte Caparica
ps@di.fct.unl.pt

ABSTRACT

Cloth simulation is a highly intensive processor task. Higher details are often traded or sacrificed for speed in order to maintain the simulation at reasonable rates. This article proposes a technique to dynamically adjust the level of detail of cloth meshes. Mesh quality, during successive subdivision/simplification operations, is guaranteed by the subdivision scheme adopted. Also, topology of the meshes used in our work exhibits unique features making it attractive in the context of a gradual level of detail variation. Subdivision is triggered by geometric criteria based on estimated surface curvature, evaluated locally. Simplification is started in cloth regions where surface curvature is considered small enough. The choice of a subdivision scheme should avoid introducing, at all costs, discontinuities on the equations governing the simulation. However, reducing the level of detail inevitably leads to such discontinuous scenarios. Our original contribution, with this work, provides a set of techniques that allows us to overcome the numeric instability problems arising from surface simplification in cloth simulations.

Keywords

Particle systems, cloth modeling, level of detail, simulation.

1. INTRODUCTION

Particle systems are the modeling tool most frequently used to simulate cloth. The material is usually modeled as a set of particles with mass interacting with each other by means of forces applied on them. These forces are either the result of internal interactions, such as bending, shearing and stretching, or generated by external causes: gravity, user interaction or collision response. Internal forces are modeled connecting particles with each other inside a group or arrangement. In order for the system to evolve over time the set of ordinary differential equations (ODE) describing the movement of each particle needs to be numerically solved.

Solving the set of ODE equations requires a substantial amount of cpu time and the computational model may require to be evaluated several times for each time step.

WSCG SHORT Communication Paper Proceedings
WSCG'2004, February 2-6, 2004, Plzen, Czech Republic.
Copyright UNION Agency – Science Press

Adaptive subdivision and simplification of the cloth mesh can use the available cpu power more wisely to achieve realistic simulations with higher levels of detail. Since more detail is introduced only where it is needed the most, results can be similar to those obtained using static meshes generated with a higher level of detail. Regions where cloth is mostly flat can easily sacrifice detail.

In this paper we propose a simulation technique based on a continuous model for cloth dynamics, coupled with an adaptive subdivision and simplification scheme that preserves mesh quality throughout the whole simulation. This scheme also allows one of the most gradual variations on the number of polygons (triangles) and particles. Due to its simplicity a compact and efficient implementation can easily be achieved.

2. RELATED WORK

Pioneering work on cloth simulation based on particle systems used energy minimization techniques [Bre92a, Bre94a] to find stable final configurations for cloth draping. The energy functions used tried to model internal cloth mechanics and were defined by relating particles with each other in particular arrangements. In the first stage of the technique, cloth was let to fall freely and collision points between cloth particles and obstacles were computed. During the second stage, the total energy of the system was minimized and particles were displaced in accordance thus correcting the previous positions computed during the initial step.

Later, several mechanical models emerged to cope with cloth dynamics and not only final draping configurations. Particle's trajectories were simulated by numerically solving the ordinary differential equations describing their movements. These mechanical cloth models can be classified as either continuous or discrete depending on their mathematical formulation.

Examples of discrete models include the well known mass-spring cloth approximation [Pro95a] or the work by Eberhardt et al. [Ebe96a] where differential equations are deduced from energy functions similar to those described in [Bre92a]. Numerical integration techniques usually used to advance the system's state were based on explicit methods such as the explicit Euler or Runge-Kutta numerical solvers.

Physical models that treat cloth as a continuum may be driven by different concerns. The work of Baraff and Witkin [Bar98a] is primarily focused on the stability of the simulation, hence their use of an implicit integration technique that sacrifices precision in favor of numerical stability. The major contribution of their work is the combination of implicit integration and direct constraint satisfaction. To achieve this, authors developed a crafted version of the conjugate gradient method to handle constraints in an elegant and efficient way. Recall that implicit ODE solvers generate a linear system of equations which, for the cloth simulation problem, are very sparse and quite large. On the other hand, with a different objective in mind, the work of Dias et al. [Dia00a] incorporate experimental data taken from real cloth measurements into a mathematical model with the clear intention of faithful and realistic simulation of well known cloth features.

All the above methods use regularly divided meshes. While discrete models most often work with quadrilaterals, continuous models are better handled by triangular meshes. The first attempt to use adaptive refinement on a regular and rectilinear mesh was made by Hutchinson et al. [Hut96a] using Provot's [Pro95a] mass-spring model. Each time an angle between two consecutive springs was considered too big, the neighboring quadrilaterals were split generating a total of 16 new ones. Results were not very satisfactory and the time spent by the technique was quite considerable. Also, the mesh never simplified back, even when detail was no longer needed.

Zhang et al. [Zha01a] also used a mass-spring model and presented a regular subdivision method. Initially, the mesh is a low resolution regular triangle mesh. Cloth draping is simulated until the system reaches a steady state. At that time, each triangle is divided into 4 smaller triangles all over the cloth mesh. The system is then allowed to evolve again and the process repeats itself over several subdivision steps. Despite the inherent inefficiency of a regular subdivision scheme, the authors claim a 2.5:1 computational gain over a completely subdivided mesh used right from the start. Unfortunately this technique cannot be applied to true cloth animation problems.

In a recent work, Villard et al. [Vil02a] started from a quadrilateral mesh that is subdivided locally based on local geometric criteria. The test to decide if subdivision must take place occurs in a particle neighborhood and all four adjoining quadrilaterals are subdivided into 4 giving a total of 16 new quads. Particles created during subdivision are classified either as active or as virtual nodes in order to maintain mesh consistency. Forces applied to virtual nodes are transferred to the active nodes connected to them. The authors used a forward explicit Euler step thus requiring very small time steps to maintain stability. Also, dynamic simplification is left for future work.

Volkov and Ling [Vol02a] addressed the adaptive refinement and simplification of cloth meshes using the model developed in [Bar98a] in conjunction with the subdivision scheme proposed in [Kob00a] named $\sqrt{3}$-subdivision. This approach has the advantage of using general triangle meshes without any topology constraints. The system monitors the difference between presumable cloth curvature, estimated at shared edges, and the polygonal model approximation. Whenever this difference is greater than a predefined threshold, subdivision occurs on the two triangles sharing the offending edge. Simplification proceeds in a similar manner, this time when the simplified mesh is considered good enough. The authors did not explain if there are any advantages of using this particular subdivision scheme in the context of cloth simulation. Also, the results shown in their article make no mention to instability problems related to temporal discontinuity in the cloth mesh topology, introduced by the subdivision and simplification operations of the technique. Cloth instability most often arises due to the strong stretching forces involved. Whenever drastic changes in this term occur, be them caused by step sizes too big or sudden changes in these forces' direction, they will most certainly introduce cloth jumpiness or even more chaotic behaviors. This subdivision scheme generates both types of the above changes in the stretching forces.

Others [How98a, DeR98a, Etz00a] also used techniques employing the generation of new vertices but these are not directly involved in the evaluation of the cloth's mathematical model and are computed geometrically.

3. HIERARCHICAL SUBDIVISION OF REGULAR 4-8 MESHES

Our work uses regular 4-8 meshes which are a special case of 4-*k* meshes [Vel00a].

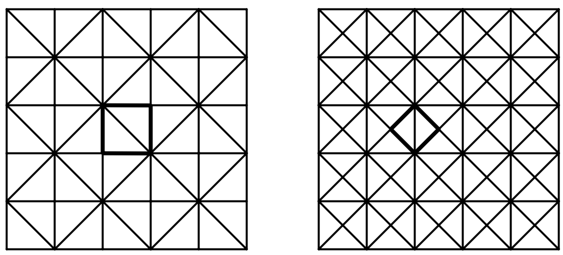


Figure 1 – Two regular 4-8 meshes

The topology, doubly illustrated in Figure 1, can be described as follows:

1. The mesh is made of quadrilaterals that are divided by one of their diagonals into two triangles.
2. The degree of interior nodes is either 4 or 8.
3. Each node with degree 4 has a 1-neighborhood where all nodes have degree 8.
4. Nodes with degree 8 have nodes on their 1-neighborhood with alternating degree between 4 and 8.

Although at first thought this kind of topology seems a little restrained, any triangle mesh can be transformed into a regular 4-8 mesh in a pre-processing step. Algorithmic details for such transformation are given in [Vel01a].

Figure 1 also shows the result of a completely uniform subdivision step, as the right hand mesh could be obtained from the left one in this manner. We can also identify, with a thicker line, the fundamental block (prototile) used by this subdivision scheme. It is formed by two triangles sharing a quadrilateral's diagonal. The elementary refinement operation is depicted in Figure 2 and it is described as the bisection of the block's internal edge with the following result:

1. The prototile's internal edge is split in two by insertion of a new vertex *v*.
2. Each original block's face is also divided in two by joining the newly inserted vertex *v* to the opposite vertices, *w* and *e*.

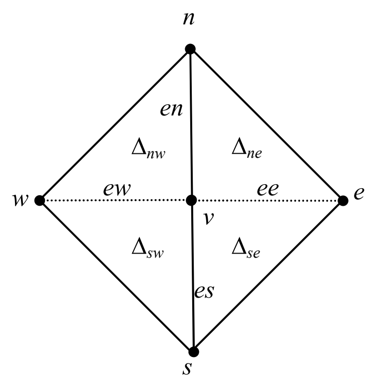


Figure 2 – Bissection of the block's internal edge

The advantage of applying this subdivision step to a conformant initial mesh is that the resulting mesh is also conformant. In other words, there are no cracks to be repaired as is the case when a triangle is split into 4 new ones.

On the left side of Figure 3, we can see the result of applying two consecutive subdivision steps to the mesh drawn with thicker lines. Dark dots represent the particles generated in the first step while grey dots represent those created in the second step.

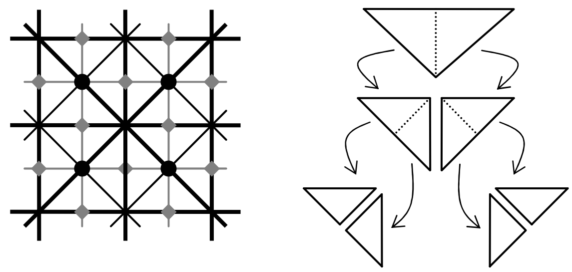
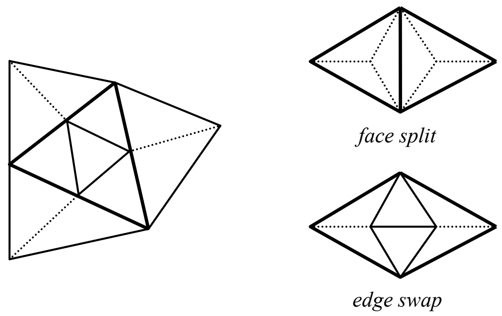


Figure 3 – Two subdivision levels and the corresponding hierarchy

4. BENEFITS OF 4-8 SUBDIVISION

A major concern while using adaptive level of detail meshes in cloth simulation is related to the discontinuities that may be introduced by refinement and simplification operations. Such discontinuities will certainly disturb the performance of the ODE solver and, worse of all, may produce an unstable simulation. It is pretty obvious that simplification will almost always lead to such discontinuities since the cloth mesh geometry changes abruptly, unless the triangles to be merged are completely coplanar. However, the same phenomena can also happen during refinement if a proper subdivision method isn't chosen.

When dealing with triangle meshes, the 1-to-4 Loop's subdivision method [Loo87a] is perhaps the one that immediately comes to mind. Each triangle is divided into four. A vertex is inserted in each edge of the original triangle and these are connected among themselves, forming a central triangle and three other ones around it. While the method maintains mesh consistency when applied uniformly, the same cannot be said in non uniform subdivisions. In this scenario, cracks may appear in the mesh if corrective measures aren't used. Figure 4(a) illustrates how the cracks can be eliminated by subdividing in two each of the neighboring triangles. These triangles are called irregular since they were not subdivided using the subdivision rule.



(a) Loop (b) √3-subdivision

Figure 4 – Other subdivision methods

A clear disadvantage is the need to keep track of regular and irregular triangles. Regular triangles belong to the subdivision hierarchy and can later be subdivided if needed. Irregular triangles, that exist only to avoid cracking, cannot be subdivided: when refinement is necessary they have to be removed and the original triangle subdivided regularly.

With √3-subdivision there is also the need to distinguish between regular and irregular triangles. Triangles are first divided into 3 irregular triangles during the face split stage, as seen in Figure 4 (b). Irregular triangles cannot be further subdivided since the resulting mesh would degenerate. Instead, they are later converted into regular triangles when two adjacent triangles have been subdivided by a face split operation. This operation is also depicted in Figure 4 (b) and it is called edge swap.

During our research, we found that the major causes for simulation instability, when using adaptive level of detail, are the instantaneous geometric modifications to the mesh. Specifically, those that change the cloth's tangent plane at a given place. When these occur, due to the introduction of new triangles, the strong stretch forces that keep cloth particles tied to each other change their modulus and direction also instantaneously. Unfortunately the same does not apply to those forces' derivatives and the ODE solver will be faced with high discontinuities on the equations to be solved. We only mention stretching forces since these are generally at least one order of magnitude greater than shearing forces and several orders greater than bending forces.

We cannot say, in a definite way, that the above discontinuities forbid the use of subdivision schemes that locally change the cloth's tangent plane. However, in the absence of additional techniques to solve or disguise the problem, we can say that such subdivision schemes shouldn't be used. We believe that even if simulation divergence could be avoided, artifacts such as cloth oscillation would probably occur.

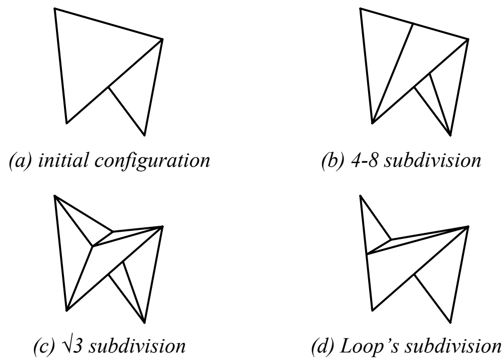


Figure 5 – Different subdivision methods applied to a pair of triangles

One of the greatest advantages of using 4-8 subdivision is its simplicity, making it highly efficient [Vel00a]. Figure 5 shows a subdivision example using different techniques. The setup is very simple, consisting only of a connected pair of triangles. The √3-subdivision example refers to the edge swap phase, after both triangles have been subdivided with face split operations. In the Loop subdivision example, the situation happens when two irregular triangles need to be converted into 4 regular ones. The example clearly shows that 4-8 subdivision is the only one able to generate triangles whose orientations are consistent with the initial setup. This feature is crucial to easily maintain continuity of physical properties, such as forces, at mesh nodes.

Moreover, 4-8 subdivision allows a more gradual transition between successive meshes since the number of particles and triangles generated by each subdivision operation is smaller when compared to other techniques. Uniform √3-subdivision triplicates the number of faces and a new vertex is generated for each triangle in the starting mesh. With Loop's 1-to-4 subdivision, the growth on the number of faces is even greater since they get multiplied by 4.

In our case, using 4-8 subdivision, the number of faces is multiplied by 2 and a new vertex is created for each pair of triangles for a complete uniform subdivision step.

5. PROPOSED TECHNIQUE

Our work builds from Baraff and Witkin's cloth model [Bar98a]. We also use the implicit Euler's solver with adaptive time stepping. This technique originates a sparse set of linear equations which are then solved using the modified conjugate gradient method as proposed in [Bar98a]. Our matrix and vector data structures are designed to exploit the sparse nature of the data they contain, turning matrix by vector multiplication into an operation with linear complexity on the number of occupied entries in the matrix.

The major goal of using adaptive level of refinement is to allow the introduction of detail where it is strictly needed and to avoid spending unnecessary cpu time simulating regions of cloth where a poorer polygonal model is sufficient. In the case of a flexible material such as cloth, the property that confers greater realism to the simulation if more detail is used is its ability to bend. Stretching and shearing behavior may be simulated with a coarser mesh since they occur inside the cloth's plane and are usually modeled with linear elastic behavior. This way, subdivision and simplification are naturally driven by criteria based on cloth's curvature. For the evaluation of these criteria to be efficient it needs to be done locally.

Subdivision and Simplification Criteria

The curvature error estimator we use measures the distance from the polygonal model surface to an idealized curved surface that locally approximates our cloth's curvature. For each pair of triangles sharing an edge, we determine the distance from the shared edge to the opposite vertices and compute its average, L. Next, the pair of triangles is conceptually replaced by two symmetric triangles where the opposite vertex of each of them is also a distance L apart from the shared edge. The cloth's local curvature is measured by a circumference that passes through those two vertices and the middle of the connecting edge. Figure 6 illustrates what has just been described. The estimated error of the polygonal approximation to the idealized cloth surface is then given by:

$$d = \frac{L\left(1 - \cos\left(\theta/2\right)\right)}{2\sin\left(\theta/2\right)},$$

where θ is the angle formed by the normals of the considered triangles.

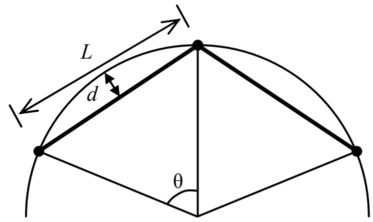


Figure 6 –Local cloth's curvature determination

During each simulation step, the curvature error estimator is computed for each shared edge. Infringing edges are then selected to trigger subdivision or simplification. Subdivision takes place when the error estimator is greater than a specified threshold. Simplification occurs exactly in the opposite scenario – when the estimator value is small enough. Despite the simplicity, real experiments proved that, for simplification, other constraints need to be satisfied to stably reduce the mesh resolution.

In [Vol02a] the same approximation error estimator is used, while in [Vil02a] the error is estimated at mesh nodes by measuring the greatest difference between the weighted normal at the vertex and the neighboring triangles' normals.

Adaptive Subdivision Algorithm

In a general way, mesh subdivision schemes are decomposed in two types of rules:

- Face rules – compute where new mesh nodes should be placed
- Vertex rules – determine new positions for the existing nodes that were involved in the subdivision.

In the context of cloth simulation we think that only the first set of rules should be used. In fact, by using vertex rules we would instantaneously change the positions of the particles involved as well as the orientations of the neighboring triangles. Besides, efficiency is a major concern in cloth simulation and leaving out vertex rules will improve the speed of the technique.

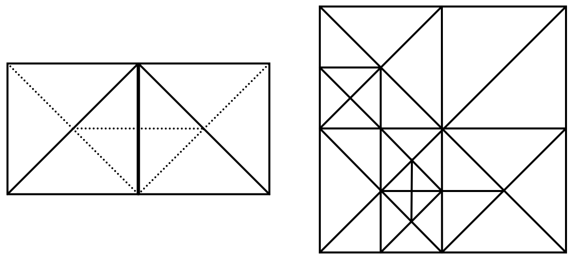


Figure 7 – Examples of boundary edge subdivision and non-uniform subdivision

Adaptive 4-8 mesh subdivision needs to distinguish between internal (Figure 2) and boundary block edges. If the edge that triggered subdivision is not internal to a basic block we will need to subdivide the neighboring blocks first. This situation is depicted in Figure 7, along with an example of adaptive subdivision on a mesh. There is also an additional complication that is easy to deal with. During refinement of boundary block edges the two neighboring blocks may have been refined to different levels of detail. Fortunately they can only differ in one level of detail. So, at most one additional subdivision step may be required first. Our recursive implementation is presented below in pseudo-code:

```
Subdivide48(Edge h)
begin
  if h.splited then return
  if h.left.level = max_level return
  le = BreakingEdge(h.left)
  re = BreakingEdge(h.right)
  if h.left.level < h.right.level then
      Subdivide48(le)
  else if h.right.level < h.level.right then
      Subdivide48(re)
  // Create  4 new triangles nw, sw, ne e se
  // as well as new edges en,es,ew, ee
  //...
  nw.level=sw.level=ne.level=se.level=h.left.level+1
  if h.type = internal then
      edge_type = external
  else edge_type = internal
  en.type=ee.type=es.type=ew.type=edge_type
  return
end
```

The function *BreakingEdge* looks for the edge of a triangle marked as *internal*. Our implementation is slightly more complex than the one presented here since it is necessary to deal with mesh boundaries. In that case only the *left* field of the edge is filled with information.

The simplification process works in reverse way and only edges resulting from previous subdivision steps need to be considered for mesh reduction.

Stable Mesh Simplification

By choosing the 4-8 subdivision scheme we eliminate a major cause of simulation instability during mesh refinement and reduction. A cause for the feared and chaotic cloth behavior is instantaneous geometry changes in total discordance with dynamics equations. However, simplification operations are still problematic, even with 4-8 subdivision. It is not possible to avoid discontinuity since the only scenario where it does not occur is for a perfectly rigid and coplanar arrangement of a pair of triangles.

Our first approach was to proceed with simplification whenever our estimator was evaluated to be sufficiently smaller than a given threshold. However, this simple approach proved to be catastrophic. A more careful analysis shows that its co-planarity is not enough. We need to be certain that the arrangement is not temporary and that the triangles are stable in relation to each other. The condition vector $C(\mathbf{x})$, proposed in [Bar98a] and used to model forces that counter cloth bending, is defined as the angle θ between the normals of two adjacent triangles (Figure 8).

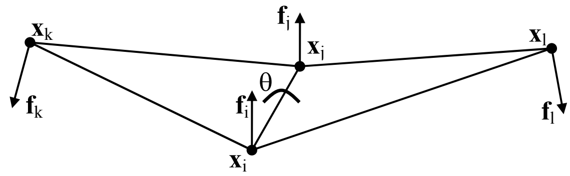


Figure 8 - Bending Condition

In a situation where the angle remains stable, the time derivative $\partial C(\mathbf{x})/\partial t$ should also be very small. In this case, the bending condition depends only on the positions of the 4 particles involved ($\mathbf{x}_i$, $\mathbf{x}_j$, $\mathbf{x}_k$ e $\mathbf{x}_l$) and we can write:

$$\frac{\partial C(\mathbf{x})}{\partial t} = \frac{\partial C(\mathbf{x})}{\partial \mathbf{x}_i}\mathbf{v}_i + \frac{\partial C(\mathbf{x})}{\partial \mathbf{x}_j}\mathbf{v}_j + \frac{\partial C(\mathbf{x})}{\partial \mathbf{x}_k}\mathbf{v}_k + \frac{\partial C(\mathbf{x})}{\partial \mathbf{x}_l}\mathbf{v}_l$$

Fortunately, all the gradients and velocities in the above equation are readily available and the extra cost incurred is negligent. Instead of checking for a zero value on the time derivative of the bending condition θ, we simply wait for its absolute value to become lower than a certain limit. With this technique we eliminate almost all instability problems observed during mesh simplification.

Another problem that we faced was derived from tight local subdivision and simplification loops. Although they are usually transitory they created mesh instability, a disturbing visual effect and a lot of CPU power was spent uselessly. To avoid this problem we devised a simple strategy that prevents newly created triangles from being destroyed almost immediately after being created. Triangle destruction is only allowed after a certain amount of simulation steps. Note that we do not prevent new triangles from being further subdivided, which is an essential feature for adaptive subdivision to quickly cope with rapid changes in curvature.

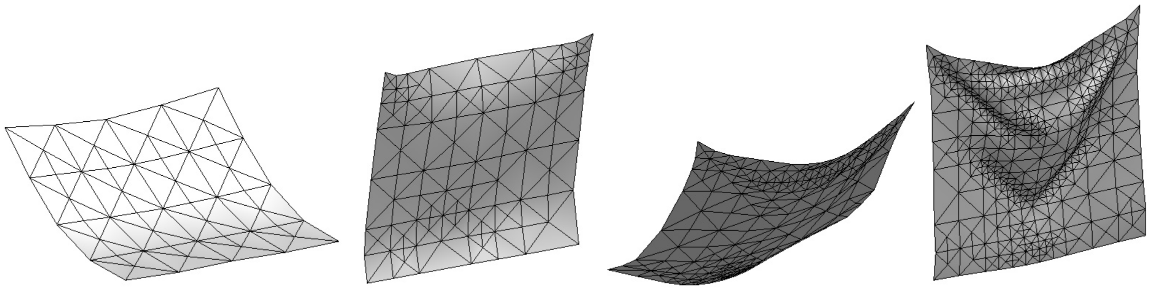


Figure 9 – Piece of cloth under gravity hanged by two corners

6. RESULTS

Figure 9 shows several steps of an experiment starting from a piece of cloth, uniformly subdivided, and consisting of 25 particles and 32 triangles. Subdivision was limited to 6 levels deep. The captured images show how our adaptive subdivision algorithm can be used to simulate and reveal high levels of detail in cloth simulation. The pictures also show how the technique adapts quite quickly to changing conditions during the simulation. To reveal the same amount of detail a uniformly refined mesh would need far more particles. Total CPU time spent per frame, including subdivision and simplification steps, varied from a mere 3.27 ms to 835.4 ms for an average value of 284.5 ms. Using a regularly subdivided mesh built from triangles with the same surface area as the smaller ones created adaptively would require a mesh of 33x33 particles. In that case, also in average, each simulation step would take around 3012 ms of CPU time. In other words, ten times more CPU time would be required.

A different experiment is presented in Figure 10. In this case, a previously regularly subdivided mesh is let to fall over a table top. The simplification process is clearly visible in the center of the mesh while further subdivision occurs at the table top boundary.

7. DISCUSSION AND FUTURE WORK

In this paper we showed how dynamic level of detail can be exploited in the context of cloth simulation. Our technique relies on local geometric criteria to decide whether the mesh is to be subdivided or simplified. We introduced several techniques to deal with instability which is expectable during simplification or mesh reduction. However this problem is not confined to mesh reduction. We have showed that also subdivision operations can be potential sources of mesh instability. Here, the choice of a suitable subdivision scheme plays a fundamental role on the solution of the stability problem.

Our choice of a subdivision scheme was based on the geometric continuity during the subdivision step in order to avoid discontinuities which would, at least, produce cloth jumpiness and slow down the simulation step to avoid numerical divergence. By using 4-8 subdivision we can easily maintain the continuity of the simulated physical properties at mesh nodes. Another advantage of this scheme is the reduced number of new elements (particles and triangles) introduced during each elementary subdivision step. This feature allows a more gradual evolution of mesh complexity. Mesh quality is also kept throughout the whole simulation process and the simplicity of the adopted scheme makes it quite attractive as its implementation is pretty straight forward.

Our experiments revealed that the extra cost of the subdivision and simplification steps are far out weighted by the CPU time savings obtained by using an adaptive mesh with much less triangles. This makes this technique highly attractive when simulation time is an issue.

We based our cloth model on the work of Baraff and Witkin [Bar98a]. An open issue of their work, until a short time ago, was the convergence of the algorithm used to solve the linear set of equations generated by the implicit Euler step. Recently, Ascher et al. [Asc02a] not only proved its convergence but also suggested how it could be accelerated, by means of a small modification during initialization. We haven't tried this modification yet in our work but it should be straight forward.

In terms of efficiency, it would also be interesting to exploit a geometry cache to accelerate subdivision operations. Our implementation immediately discards the geometry information associated with simplified triangles. This information could be useful later when cloth is subdivided again in those regions. In a different, but perhaps more fruitful direction, modern graphics hardware is now available for general programming. Since the bottleneck is currently on the CPU side, by using the GPU to solve the linear system of equations [Bol03a] would allow the CPU to run faster on other parts of the computation accelerating the process as a whole.

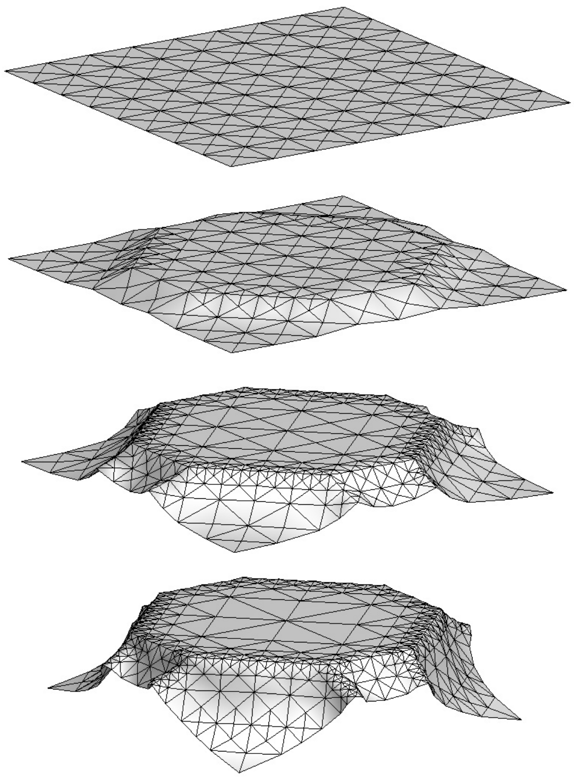


Figure 10 - Cloth falling over a table top

8. REFERENCES

[Asc02a] Ascher, U.M., and Boxerman, E. On the modified conjugate gradient method in cloth simulation, http://www.cs.ubc.ca/~ascher.

[Bar98a] Baraff, D., and Witkin, A. Large steps in cloth simulation, in SIGGRAPH 98 Conference Proceedings, Annual Conference Series, pp.43-54, ACM SIGGRAPH, 1998.

[Bol03a] Bolz, J., Farmer, I., Grinspun, E., and Schröder, P. Sparse Matrix Solvers on the GPU: Conjugate Gradients and Multigrid, ACM Transactions on Graphics (TOG), Volume 22, Issue 3 (July 2003), pages 917-924.

[Bre92a] Breen, D.E., House, D.H., and Getto, P.H. A physically based particle model of woven cloth, The Visual Computer, 8(5-6):264-277, June 1992.

[Bre94a] Breen, D.E., House, D.H., and Wozny, M.J. Predicting the drape of woven cloth using interacting particle systems, Computer Graphics (Proc. SIGGRAPH), pp.365-372, 1994.

[DeR98a] DeRose, T., Kass, M., and Truong, T. Subdivision surfaces in character animation. In Proceedings of the 25th annual conference in computer graphics and interactive techniques, pp.85-94, ACM Press, 1998.

[Dia00a] Dias, J.M.S., Gamito, M.N., and Rebordão, J.M. A discretized linear elastic model of cloth buckling and drape. Textile Research Journal, 70(4):285-297, April 2000.

[Ebe96a] Eberhardt, B., Weber, A., and Strasser, W. A fast, flexible, particle-system for cloth draping, IEEE Computer Graphics and Applications, 16(5):52-59, September 1996.

[Etz00a] Etzmuss, O., Eberhardt, B., Hauth, M., and Strasser, W. Collision adaptive particle systems, Proceedings Pacific Graphics 2000, pp.338-347, 2000.

[How98a] Howlett, P., Hewitt, W.T. Mass-spring simulation using adaptive non-active points, Computer Graphics Forum, 17(3), 1998.

[Hut96a] Hutchinson, D., Preston, M., and Hewitt, T. Adaptive refinement for mass/spring simulations, in Computer Animation and Simulation '96, Eurographics, pp.31-45, Springer-Verlag Wien New York, 1996.

[Kob00a] Kobbelt, L. √3-Subdivision, in proceedings of the 27th annual conference on computer graphics and interactive techniques, pp.103-112, ACM Press/Addison-Wesley Publishing Co., 2000.

[Loo87a] Loop, C.T. Smooth subdivision surfaces based on triangles, Master's Thesis, Department of Mathematics, University of Utah, August 1987.

[Pro95a] Provot, X. Deformation constraints in a mass-spring model to describe rigid cloth behaviour, Computer Interface Proceedings, pp.147-154, Quebec City, Canada, May 1995.

[Vel00a] Velho, L., and Gomes, J. Variable resolution 4-k meshes: Concepts and Applications, Computer Graphics Fórum, 19(4):195-214, 2000.

[Vel01a] Velho, L., and Zorin, D. 4-8 Subdivision, Computer Aided Geometric Design, 18(5): 397-427, 2001.

[Vil02a] Villard, J., and Borouchaki, H. Adaptive meshing for cloth animation, 11th International Meshing Roundtable, pp.243-252, Ithaca, New York, USA, September 2002.

[Vol02a] Volkov, V., and Ling, L. Adaptive local refinement and simplification of cloth meshes, in First International Conference on Information Technology & Applications-ICITA 2002, Bathurst, Australia, 25-28 November 2002. http://charybdis.mit.csu.edu.au/~mantolov/CD/ICITA2002/papers/266-5.pdf

[Zha01a] Zhang, D., and Yuen, M. Cloth simulation using multilevel meshes, Computers & Graphics, 25(3): 383-389, 2001.