A Hybrid Physics-Based Subdivision Technique Using Coupled Dynamic and Subdivision Parameters

Sumantro Ray and Hong Qin
Department of Computer Science
State University of New York at Stony Brook
Stony Brook, NY 11794-4400
sray | qin@cs.sunysb.edu

Abstract

The last few decades have seen enormous progress in both geometric subdivision, and physics-based simulation techniques. Mesh-based dynamic systems often require both subdivision and physical simulation for realistic and accurate results. However, the simulation parameters have been independent of the subdivision parameters, and vice-versa. This paper attempts to bridge this gap. We propose a hybrid approach that combines the physics-based simulation techniques and geometric subdivision algorithms, and demonstrate a mass-spring based system with physics-based butterfly subdivision. The initial subdivision coefficients are extracted using the physical properties of the base (L_0) mesh. Latter subdivision steps generate both the geometric and physical properties of the subdivided (L_k) mesh. This approach conserves mass, center of gravity, linear momentum and external force, and minimizes the distance between the L_k and L_{k+1} meshes, at any time step. Our approach is general, efficient, and will serve as a foundation for many applications in many fields.

1. Introduction

Geometric modeling concerns with the computation and representation of various shapes of models. Hierarchical refinement of mesh-based geometric models leads to approximating techniques such as Doo-Sabin[10] and Catmull-Clark[3] subdivisions, and interpolating techniques such as the Butterfly[11] subdivision. The subdivision kernel is normally independent of the geometry of the model, and is a function of the topology (connectivity) of the associated mesh. In filter theory, such a subdivision kernel corresponds to a linear, time-invariant low-pass filter with local support.

Physics-based modeling allows geometric models to be governed by differential equations. This approach offers unsurpassed advantages - it is natural to control, intuitive to manipulate, and the end user does not need mathematical sophistication. However, it requires numerical simulation, which is time consuming, and appropriate Courant-Friedrichs-Levy (CFL) conditions must be imposed to ensure stability. A lot of progress in the last few decades has resulted in an explosion of research in both geometric and physics-based modeling techniques. The field of geometric subdivision has matured substantially since the pioneering works of Chaikin[5], Catmull-Clark[3], Dyn et al [11] and Doo-Sabin[9, 10]. Similarly, physics-based modeling techniques have seen myriad applications in animation[21, 1, 7], fluid dynamics[12, 20, 19], and Lagrangian dynamics/FEM[16, 15, 18, 4]. However, the simulation parameters used in physics-based modeling have generally been independent of the subdivision parameters, and vice-versa. Our paper attempts to combine the two approaches under a single hybrid framework.

Our approach uses a novel mass-spring formulation. We discretize the Lagrangian dynamic equations using z-transforms[18, 8]. This technique allows a complete decoupling of the geometry and the topology of the underlying mesh during simulation. We start with a triangular base mesh, with mass, damping and stiffness coefficients associated with the vertices and edges of the mesh. We then impose certain CFL conditions (Nyquist rate of sampling), and Newtonian invariance conditions (conservation of mass, center of gravity, linear momentum, and total external force) to compute a) mass, damping and stiffness of the subdivided mesh, and b) the butterfly subdivision kernel. Therefore, our approach uses the physical properties of a base mesh to construct both the physics and the geometry of the subdivided mesh.

This has several advantages. The resulting subdivision kernel is a function of the topology, as well as the physics of the base mesh. The high stiffness regions of the base mesh generate lower curvature regions (compared to traditional

Proceedings of the Computer Graphics International (CGI'04)
1530-1052/04 $20.00 © 2004 **IEEE**

geometric subdivisions) in the subdivided mesh, and vice-versa, which is both intuitive, and physically accurate. This allows us to model anisotropic materials, as well as multi-material junctions and interfaces; and ensures that different material patches at a junction do not blend and form one single patch in the subdivided mesh. Thus the subdivided mesh accurately reflects the inherent anisotropy of the base mesh. Also, similar user forces elicit similar dynamic responses from both the base and the subdivided meshes. This has applications in hierarchical animation and simulation. In addition, the global Newtonian invariants are conserved during subdivision.

The rest of the paper is organized as follows. Section 2 explores the background. Section 3 presents the contribution. Section 4 describes the dynamic solver. Section 5 describes the hybrid subdivision formulation. Section 6 presents the results. Section 7 concludes the paper.

2. Background

The study of subdivision curves and surfaces was pioneered by Chaikin[5]. Chaikin's corner cutting strategy, unfortunately, was limited to rectangular meshes. This drawback was soon overcome by Catmull-Clark[3], whose approximating subdivision technique generated B-Spline surfaces for arbitrary meshes. Doo-Sabin[10], extended tensor-product biquadratic splines. Dyn et al[11] developed a interpolating subdivision scheme for triangular meshes. Loop et al[14] developed an algorithm for creating smooth spline surfaces over irregular meshes.

Various work has previously been done to generate dynamic surfaces using physics-based modeling. Terzopoulos[21] demonstrated simple interactive sculpting using viscoelastic and plastic methods. Celniker[4] used finite-element optimization of energy functionals. Bloor et al[2] used similar optimizations through numerical methods. Dachille et al[6] used a finite difference method to solve the Lagrangian equation. Halstead et al[13] implemented smooth interpolation with Catmull-Clark surfaces using a thin-plate energy functional. Raviv and Elber[17] performed three dimensional free-form sculpting using scalar tri-variate functions. Qin and Terzopoulos[16] developed a framework for D-NURBS for physics-based design.

3. Contribution

Traditional geometric subdivision schemes derive the subdivision coefficients from the topology of the corresponding mesh. When combined with physics-based modeling, this approach is insufficient. Under a physics-based modeling framework, the subdivided mesh must satisfy certain physical invariants depending upon the underlying physics. For example, while modeling fluid-flow, the subdivided grid must conserve the mass, momentum and energy associated with any fluid dynamics system. Similarly, in a Lagrangian dynamics framework, certain physical quantities such as mass, center of gravity etc. must be conserved across hierarchical levels. This requires the subdivision kernel to be dependent upon both the topology as well as the physics of the corresponding mesh.

In this paper, the traditional geometric subdivision is integrated within a physics-based modeling framework. We use a simple mass-spring based model, and demonstrate the system by generalizing the butterfly interpolating subdivision scheme to incorporate physics-based weights. The following lists the different contributions of this paper.

Dynamic Solver: We model our dynamic solver around the Lagrangian differential equation. The z-transform ([18, 8]) based solver allows us to decouple the physics from geometry, and stability is ensured if the appropriate CFL condition (the Nyquist sampling rate, in this case) is maintained. The critical (Nyquist) sampling rate is easily computed from the mass and stiffness of the mesh using the Gerschgorin Circle Theorem. This approach is similar to the one described in our previous paper[18].

Hybrid Subdivision: The physical properties of the base mesh (mass, stiffness) are used to compute the modified butterfly subdivision kernel. The curvature of the subdivided mesh is inversely proportional to the stiffness distribution in the base mesh. In addition, the physical properties of the subdivided mesh are computed as a function of the physical properties of the base mesh and the coefficients in the subdivision kernel. The overall process guarantees the conservation of the global Newtonian variables.

External Forces: Application of an external force vector on the base mesh is transformed using the subdivision kernel to an appropriate force vector for the subdivided mesh, and vice-versa (Equation 4). This allows simultaneous simulation in both base and subdivided domains. This also allows us to model gravity, and external collision (with the ground, for example) simultaneously in different levels, while guaranteeing similar dynamic responses at both the base and the subdivided levels.

Dynamic Editing: Our subdivision kernel is independent of the geometry of the underlying mesh. However, since this kernel depends upon the distribution of mass and stiffness of the base mesh, modifying these physical properties also changes the subdivision kernel. This allows us to model multi-material systems, where different physical properties across a material junction in a base mesh result in a) different material properties across the junction in the subdivided mesh, and b) C^1 discontinuity (crease) at the junction.

Proceedings of the Computer Graphics International (CGI'04)
1530-1052/04 $20.00 © 2004 **IEEE**

4. Dynamic Solver

We begin with the Lagrangian equation, that defines the motion of a point, or a set of points in three dimensions, subject to internal and external forces. Our input contains a set of sampled vertices and edges with associated physical properties (mass, damping, stiffness).

4.1. Formulation

The Lagrangian equation of motion for any mesh vertex $\mathbf{r}_i$ is expressed as:

$$\mu_i \mathbf{r}_i'' + \rho_i \mathbf{r}_i' + \sum_j \kappa_{ij}[(\mathbf{r}_i - \mathbf{r}_j) - \lambda_{ij}\mathbf{U}_{ij}] = \mathbf{f}_i^{tot}$$

$$\mathbf{v}_i = \mathbf{r}_i' = d\mathbf{r}_i/dt$$

$$\mathbf{U}_{ij} = (\mathbf{r}_i - \mathbf{r}_j)/|(\mathbf{r}_i - \mathbf{r}_j)| \quad (1)$$

where $\mathbf{r}_j$ is a neighboring vertex. The mass, damping, and stiffnesses for the vertex $\mathbf{r}_i$ are given by μ_i, ρ_i, and κ_{ij} respectively. λ_{ij} is the rest-length of the spring joining $\mathbf{r}_i$ and $\mathbf{r}_j$. $\mathbf{f}_i^{tot}$ is the total external force at vertex $\mathbf{r}_i$. $\mathbf{U}_{ij}$ is the directional unit vector. The terms on the left-hand-side are the kinetic and damping components of the total force, and the internal (spring) force at $\mathbf{r}_i$. A typical mass-spring system for a triangular mesh is shown in Figure 1.

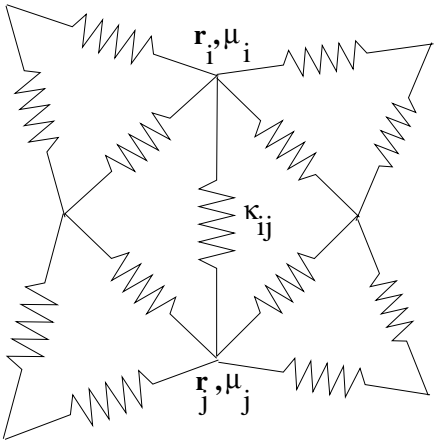

Figure 1. *Mass-Spring System: μ (vertex-mass) and κ (edge-stiffness)*

For a set of *N* vertices, the Lagrangian equation may be written in a matrix form (Equation 2), such that $\mathbf{R}$ and $\mathbf{V}$ are N×3 dimensional positional and velocity state vectors respectively.

$$M\mathbf{R}'' + D\mathbf{R}' + K\mathbf{R} = \mathbf{F}_{tot}$$
$$\mathbf{V} = \mathbf{R}' = D\mathbf{R}/Dt \quad (2)$$

where D/Dt is a matrix operator on the state vector $\mathbf{R}$. M and D are N×N diagonal mass and damping matrices. K is a N×N symmetric, sparse, stiffness matrix such that $-K_{ij}$ is the stiffness between the i^{th} and the j^{th} vertices. F_{tot} is the N×3 force-vector state. K_{ii} equals the sum of all neighboring stiffnesses at vertex $\mathbf{r}_i$. Therefore, elements of any row (or column) of K add up to zero. This is a direct consequence of the Newton's third law of motion.

Discretizing the above state equations using z-transforms, and expanding using Taylor's series, we get a discrete-time equation (Equation 3) that allows us to estimate the state of the system at time-sample N given the state at time-sample $N-1$ where G=$M^{-1}D/2$, Ω^2=$M^{-1}K$ is the state matrix for the dynamical system, and I is an identity matrix. H, X and their derivatives are digital transfer functions. T is the sampling time as computed in section **??**.

$$\begin{pmatrix} \mathbf{R}^N \\ \mathbf{V}^N \end{pmatrix} = \begin{pmatrix} I & H' & HM^{-1} \\ \mathbf{0} & X' & XM^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{R}^{N-1} \\ \mathbf{V}^{N-1} \\ \mathbf{F}_{tot}^{N-1} \end{pmatrix}$$

where $X = H' - GH, X' = H'' - GH'$ and

$$\begin{pmatrix} H \\ H' \\ H'' \end{pmatrix} = \begin{pmatrix} \Omega^{-2}(1 - cos\Omega T) \\ \Omega^{-1} sin\Omega T \\ cos\Omega T \end{pmatrix}$$
$$\simeq \begin{pmatrix} \frac{T^2}{2} & \frac{-T^4}{24} & \frac{T^6}{720} \\ T & \frac{-T^3}{6} & \frac{T^5}{120} \\ 1 & \frac{-T^2}{2} & \frac{T^4}{24} \end{pmatrix} \begin{pmatrix} 1 \\ \Omega^2 \\ \Omega^4 \end{pmatrix} \quad (3)$$

4.2. Solver Flow

The solver flow diagram is shown in Figure 2. Given the state at any time step $N-1$, the solver essentially acts as a discrete-time linear filter, and generates the state at time step N. The filter characteristics are wholly dictated by the state matrix Ω^2 (Equation 3), which is a function of the mass, stiffness, and the topology of the associated mesh. During hybrid subdivision, the subdivision kernel and the state matrix of the subdivided mesh are computed from the state matrix of the base mesh, as discussed in section 5.

Proceedings of the Computer Graphics International (CGI'04)
1530-1052/04 $20.00 © 2004 **IEEE**

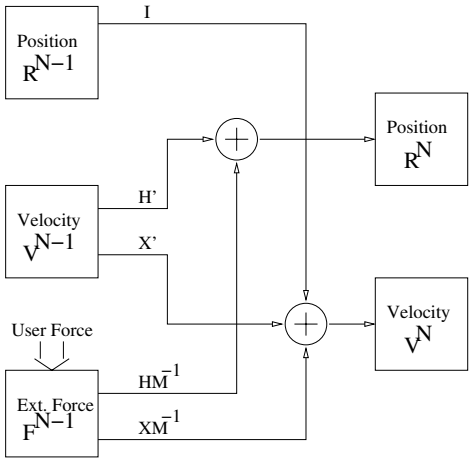

Figure 2. *Solver Flow*

5. Hybrid Subdivision

The subdivision coefficients in existing, mesh-based subdivision techniques depend upon the topology of the base mesh. However, this approach is erroneous when dealing with dynamic meshes for two important reasons. Firstly, the possible anisotropy in the physical properties of the mesh is not reflected in the subdivision process. Secondly, the physical invariants that are always part of any physics-based simulation are not satisfied if a traditional subdivision kernel is used. We combine the traditional, geometry-based subdivision coefficients and the physical properties of the mesh into a hybrid scheme. In particular, we generalize the interpolating butterfly subdivision scheme, under a dynamic mass-spring framework. In traditional butterfly subdivision, the weight w is uniform over the whole mesh. Generalizing the butterfly scheme under the physics-based approach allows us to introduce anisotropic weights, resulting in a) intuitive and accurate subdivision at multi-material junctions, and b) conservation of physical invariants.

This section is divided into the following parts. Subsection 5.1 explains the hybrid solver (mass-spring dynamic solver + subdivision). Subsection 5.2 briefly outlines the geometric butterfly subdivision. Subsection 5.3 integrates mass-spring solver and butterfly subdivision in an unified framework.

5.1. Hybrid Solver

A mass-spring system depends upon the state matrix Ω^2 for its dynamic behavior (Section 4). Let Ψ_0^{N-1}=$\{R, V, F\}_0^{N-1}$ define the state vector for the base mesh at any time sample $N-1$. If the subdivision coefficients are accumulated in a row-affine matrix P, we can compute the corresponding subdivided mesh state vector, as

$$\begin{aligned} R_1^{N-1} &= P\,R_0^{N-1} \\ V_1^{N-1} &= P\,V_0^{N-1} \\ F_1^{N-1} &= M_1\,P\,M_0^{-1}\,F_0^{N-1} \end{aligned} \tag{4}$$

where M_0 and M_1 are diagonal mass matrices for the base (L_0) and subdivided meshes (L_1), respectively, $\mu_{0i} = M_0^{ii}$ and $\mu_{1j} = M_1^{jj}$. Similar equations exist for time sample N (Figure 3). Conservation of mass, center of gravity, linear momentum and total external force gives us

$$\begin{aligned} \sum_{i\in L_0} \mu_{0i} &= \sum_{j\in L_1} \mu_{1j} \\ \sum_{i\in L_0} \mu_{0i}\mathbf{r}_0^i &= \sum_{j\in L_1} \mu_{1j}\mathbf{r}_1^j \\ \sum_{i\in L_0} \mu_{0i}\mathbf{v}_0^i &= \sum_{j\in L_1} \mu_{1j}\mathbf{v}_1^j \\ \sum_{i\in L_0} \mathbf{f}_0^i &= \sum_{j\in L_1} \mathbf{f}_1^j \end{aligned} \tag{5}$$

The above equations are satisfied if the mass vectors μ_0 and μ_1 are connected through the subdivision matrix P (Equation 6). Also, the error E between Ψ_1^N and $\overline{\Psi}_1^N$ has to be minimized (Figure 3).

$$\begin{aligned} \mu_{0i} &= \sum_{j\in L_1} \mu_{1j}P_{ji} \quad \forall i \in L_0 \\ \Omega_1^2\,P &\simeq P\,\Omega_0^2 \end{aligned} \tag{6}$$

Therefore, we can solve for the physics-based subdivision coefficients, as well as the physical properties of the subdivided mesh, by satisfying the constraints and the conditions in Equations 5 and 6. The above scheme is applicable to any subdivision scheme. In this paper, we focus on the butterfly subdivision for this purpose.

5.2. Butterfly Subdivision

The butterfly subdivision[11] is an interpolating subdivision scheme for triangular meshes. A typical subdivision is shown in Figure 4, where the subdivided mesh interpolates the base mesh. Each old triangle is split into four new triangles. New points are introduced for each edge of the base mesh, using a butterfly weighting coefficient $0 < w < \frac{1}{16}$.

5.3. Hybrid Butterfly Subdivision

In butterfly subdivision, the subdivision matrix P can be split into an upper identity matrix, I, and a lower coefficient

Proceedings of the Computer Graphics International (CGI'04)
1530-1052/04 $20.00 © 2004 **IEEE**

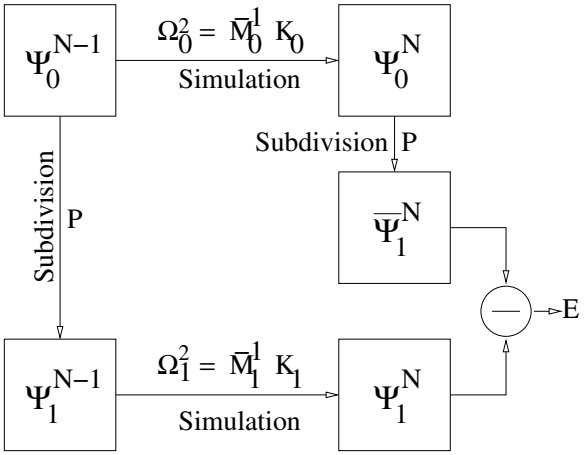

Figure 3. *Hybrid Subdivision Diagram: the error E must be minimized*

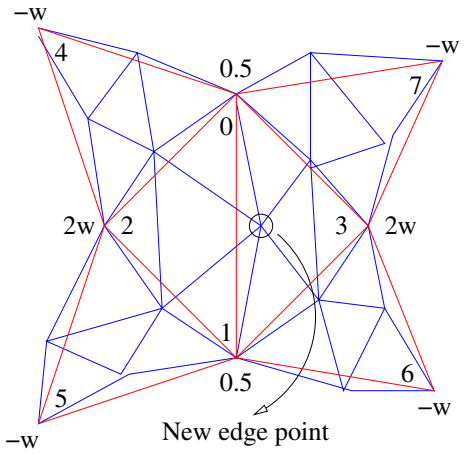

Figure 4. *Butterfly Subdivision: Base mesh (red), Subdivided mesh (blue)*

matrix Q (Equation 7). In traditional butterfly subdivision, Q is a function of a single variable w (subsection 5.2). In our case, each inserted edge point will have a unique w_k, depending upon the distribution of the physical properties in its neighborhood. Similar to P, the mass and stiffness matrices M_1 and K_1 are split into sub-matrices corresponding to a) interpolated points from the base mesh, and b) new inserted edge points. M_1^0 and M_1^1 are the diagonal mass matrices for the interpolated and inserted vertices respectively, whereas A and B are the stiffness sub-matrices (Figure 5). D is a diagonal sub-matrix, composed of the total incident stiffnesses at the interpolated vertices in the subdivided mesh.

$$P = \begin{pmatrix} I \\ Q \end{pmatrix}$$

$$M_1 = \begin{pmatrix} M_1^0 & 0 \\ 0 & M_1^1 \end{pmatrix}$$

$$K_1 = \begin{pmatrix} D & -A \\ -A^T & B \end{pmatrix} \tag{7}$$

For a diagonal matrix W, where $W_{kk} = w_k$ are the weights associated with the edges of the base mesh, the conditions from Equation 6 can be rewritten as:

$$\begin{aligned} \mu_{1i}^0 &= \mu_{0i} - \sum_{i \neq k} (Q^T W Q)_{ik} \mu_{0k} \\ \mu_{1i}^1 &= \sum_{k} (Q^T W Q)_{ik} \mu_{0k} \\ D - AQ &\simeq M_1^0 M_0^{-1} K_0 \\ -A^T + BQ &\simeq M_1^1 Q M_0^{-1} K_0 \end{aligned} \tag{8}$$

where μ_{1i}^0 and μ_{1i}^1 are the masses of the old and the new vertices respectively in the subdivided mesh. By solving for the above equations, we get the physics-based butterfly subdivision weights, and the physical properties for the subdivided mesh.

Let $(\mathbf{r}_i^0, \mathbf{r}_j^0)$ be an edge in the base mesh L_0, and $\mathbf{r}_k^1$ be the corresponding edge-vertex in the subdivided mesh (Figure 5). The stiffness of the edges connecting $\mathbf{r}_k^1$ and $\mathbf{r}_i^0$ is given by Equation 9, where κ_{ij}^0 is the stiffness associated with the edge (i, j) in the base mesh.

$$\kappa_{ik}^1 = 2\kappa_{ij}^0 \left(1 - \frac{\sum_{i \neq k} (Q^T W Q)_{ik} \mu_{0k}}{\mu_{0i}}\right) \tag{9}$$

Similarly, it can be shown that the butterfly weight w_k associated with the edge k in the subdivided mesh is inversely proportional to its base stiffness (κ_{ij}^0, and directly proportional to the average of the weighted stiffnesses of the neighboring edges (Equation 10, Figure 4).

Proceedings of the Computer Graphics International (CGI'04)
1530-1052/04 $20.00 © 2004 **IEEE**

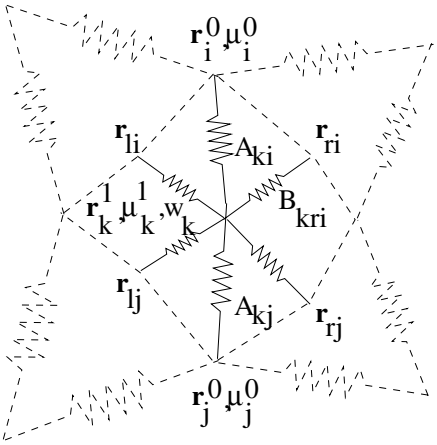

Figure 5. *Mass-Spring System in Subdivided Mesh: w_k is the weight associated with the edge-vertex, dotted outline shows the neighboring base mesh*

$$\begin{aligned} \kappa_{sum} &= 2(\kappa_{02} + \kappa_{03} + \kappa_{12} + \kappa_{13}) + \\ &\quad (\kappa_{04} + \kappa_{07} + \kappa_{15} + \kappa_{16}) + \\ &\quad (\kappa_{24} + \kappa_{25} + \kappa_{36} + \kappa_{37}) \\ w_k &= max(\frac{\kappa_{sum}}{256\kappa_{01}}, \frac{1}{16}) \end{aligned} \tag{10}$$

Finally, the edge-edge stiffnesses in the subdivided mesh are computed by solving a set of linear equations (Equation 12, Figure 5),

$$\kappa_{kl_i}^1(\frac{1}{2} - w_{li}) + \kappa_{kr_i}^1(\frac{1}{2} - w_{ri}) = \tag{11}$$

$$(M_1^1 Q M_0^{-1} K_0)_{kj} + \kappa_{ij}^0(\frac{\mu_{1j}^0}{\mu_{0j}} - \frac{\mu_{1i}^0}{\mu_{0i}}) \tag{12}$$

for every half-edge in the base mesh. Therefore, we have $E \times 2$ equations, and $E \times 2$ unknowns (since for every edge in the subdivided mesh, there are four edge-edge springs).

Therefore, given the base mesh and the physical properties (mass, stiffness) associated with it, we can compute a set of butterfly weights, and the physical properties for the subdivided mesh, such that the physical constraints and conditions (Equations 4, 5 and 6) are satisfied. For a regular mesh with isotropic mass and stiffness values, the subdivided mesh is identical to that obtained by a purely geometric subdivision.

6. Results

The simulation was run on a 800MHz Pentium-II computer. Table 1 gives the timing analysis, and the time requirements. All time in milli-seconds.

V,E,F	μ	κ	dT	SInit	Subd	Phys
514,1536,1024	0.0136	1.75	100	6	20	40
561,1584,1024	0.0321	2.00	150	5	30	70
1374,4057,2684	0.0136	1.75	250	5	60	110

Table 1. *Timing information for the models used in this paper: a) Tetrahedron b) Triangle-patch c) Cat. Mass μ is mass per vertex, Average stiffness is κ, Sampling time is dT. All other times are amortized per vertex (SInit: Initialization time for solver (O(V.k)), Subd: Subdivision matrix calculation time ($O(V_c.k_c)$), Phys: Physical parameters subdivision time ($O(V_c.k_c + E_c.6)$)). V and k are vertex count, and average edges-per-vertex, V_c and k_c are similar values for subdivided mesh, E_c is the edge count for the subdivided mesh.*

In Figure 6, we perform a dynamic simulation of three hierarchical levels simultaneously. The levels are L_0 (base mesh: 4 vertices), L_3 (intermediate level: 34 vertices), and L_5 (detailed level: 514 vertices). The user applies forces to the mesh in the intermediate level, and the system automatically updates all the lower and the higher levels. The plot shows the energy-levels for the three levels with changing time. Even though the user is animating at L_3, the system automatically transmits the forces to the other two levels. Since the hybrid subdivision has integrated the dynamics and subdivision, the conditions in Equation 6 are satisfied, and the time-behavior of L_0 and L_5 closely follow that of L_3.

Figure 7 shows the stiffness distribution for a typical subdivision. The three plots ((Figures $7(a)$, (b) and (c)) have a baseline stiffness $\kappa = 1.0$, and maximum stiffnesses of κ = 1.5, 2.5, and 2.75 respectively. The maximum stiffness increases with the level of subdivision, since smaller and smaller grid size necessitates a larger stiffness, in order to conserve the spring potential energy.

7. Conclusion

This paper presents a novel, hybrid technique that unifies subdivision and physics-based modeling under a single framework. We demonstrate the approach using the mass-spring dynamics, and butterfly subdivision. However, the underlying principle stays the same for any pair of dynamic system and subdivision scheme. The framework, given the

Proceedings of the Computer Graphics International (CGI'04)
1530-1052/04 $20.00 © 2004 **IEEE**

physics and the geometry of the base mesh, computes the physics-based subdivision coefficients, such that a) the material properties of the base mesh are accurately reflected in the subdivided mesh, and b) the invariants associated with the dynamical system are maintained across hierarchical levels. The user can dynamically modify the physical properties of the model at any level, and the subdivision coefficients change automatically. Future work will generalize this approach to finite-element based models, with any arbitrary subdivision scheme. In this paper, the topology of the subdivided mesh is governed only by the topology of the base mesh. We plan to extend our scheme to compute the topology of the subdivided mesh using the base mesh topology and physical properties. This will allow us to integrate the geometry, physics and topology of the subdivided mesh under a single unified framework.

8. Acknowledgements

This research was supported in part by the NSF grants IIS-0082035 and IIS-0097646, Honda Initiation Award, and an Alfred P. Sloan Fellowship.

References

[1] D. Baraff and A. Witkin. Large steps in cloth simulation. In *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH*, pages 43–54, jul 1998.

[2] M. I. G. Bloor and M. J. Wilson. Using partial differential equations to generate free form surfaces. *Computer Aided Design, May 1990*, 22(4):202–212, 1990.

[3] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10:350–355, Sept. 1978.

[4] G. Celniker and D. Gossard. Deformable curve and surface finite-elements for free-form shape design. In *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 257–266, July 1991.

[5] G. Chaikin. An algorithm for high speed curve generation. *Computer Graphics and Image Processing*, 3:346–349, 1974.

[6] F. Dachille IX, H. Qin, A. Kaufman, and J. El-Sana. Haptic sculpting of dynamic surfaces. In *Proceedings of the Conference on the 1999 Symposium on interactive 3D Graphics*, pages 103–110, Apr. 26–28 1999.

[7] G. Debunne, M. Desbrun, A. Barr, and M.-P. Cani. Interactive multiresolution animation of deformable models. In N. Magnenat-Thalmann and D. Thalmann, editors, *Computer Animation and Simulation '99*, SpringerComputerScience, pages 133–144. Springer-Verlag Wien New York, 1999. Proceedings of the Eurographics Workshop in Milano, Italy, September 7–8, 1999.

[8] D. DeFatta, J. Lucas, and W. Hodgkiss. *Digitial Signal Processing - A system design approach*. John Wiley & Sons, first edition, 1988.

[9] D. Doo. A subdivision algorithm for smoothing down irregularly shaped polyhedrons. In *Proced. Int'l Conf. Ineractive Techniques in Computer Aided Design*, pages 157–165, 1978. Bologna, Italy, IEEE Computer Soc.

[10] D. Doo and M. Sabin. Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design*, 10:356–360, Sept. 1978.

[11] N. Dyn, D. Levin, and J. A. Gregory. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics*, 9(2):160–169, Apr. 1990.

[12] M. J. Gerald-Yamasaki. Cooperative visualization of computational fluid dynamics. In *Eurographics '93*, pages 497–508. Eurographics, 1993.

[13] M. Halstead, M. Kass, and T. DeRose. Efficient, fair interpolation using Catmull-Clark surfaces. *Computer Graphics*, 27(Annual Conference Series):35–44, 1993.

[14] C. Loop and T. DeRose. Generalized B-spline surfaces of arbitrary topology. volume 24, pages 347–356, Aug. 1990.

[15] H. Qin. Physics-based geometric design. *International Journal of Shape Modeling*, 2(2,3):139–188, Feb. 1996.

[16] H. Qin and D. Terzopoulos. D-NURBS: A Physics-Based Framework for Geometric Design. *IEEE Transactions on Visualization and Computer Graphics*, 2(1):85–96, Mar. 1996.

[17] A. Raviv and G. Elber. Three dimensional freeform sculpting via zero sets of scalar trivariate functions. In *Proceedings of the Fifth Symposium on Solid Modeling and Applications (SSMA-99)*, pages 246–257. ACM Press, June 9–11 1999.

[18] S. Ray and H. Qin. Novel solver for dynamic surfaces. In *Proceedings of Graphics Interface 2001*, pages 47–54, June 2001.

[19] J. Stam. Stochastic dynamics: Simulating the effects of turbulence on flexible structures. *Computer Graphics Forum*, 16(3):159–164, Aug. 1997.

[20] J. Stam and E. Fiume. Depicting fire and other gaseous phenomena using diffusion processes. In *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 129–136. ACM SIGGRAPH, Aug. 1995.

[21] D. Terzopoulos and K. Fleischer. Deformable models. *The Visual Computer*, 4(6):306–331, Dec. 1988.

Proceedings of the Computer Graphics International (CGI'04)
1530-1052/04 $20.00 © 2004 **IEEE**

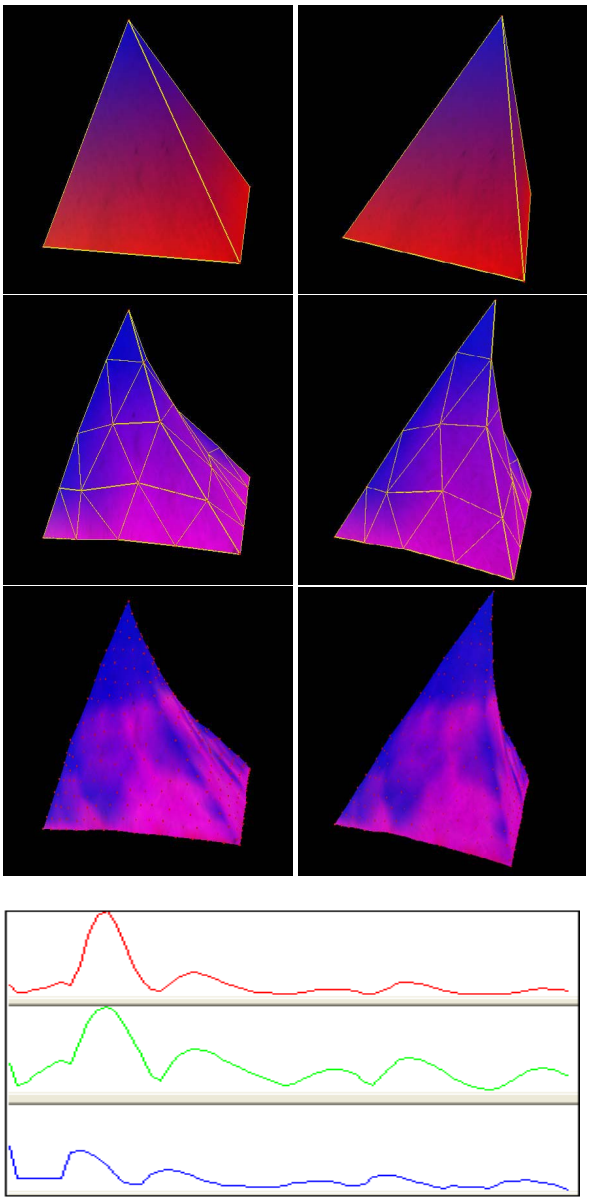

Figure 6. *First row (a),(b): Base mesh (L_0 4 vertices) at timesteps $\tau = 0$ and 60 Second row (c),(d): Intermediate mesh (L_3 34 vertices) at $\tau = 0$ and 60 Third row (e),(f): Detailed mesh (L_5 514 vertices) at $\tau = 0$ and 60 Fourth row (g): Energy plots for the three mesh simulations w.r.t. τ, $0 \leq \tau \leq 100$. The energy is plotted on a scale $0 \leq E \leq 1$*

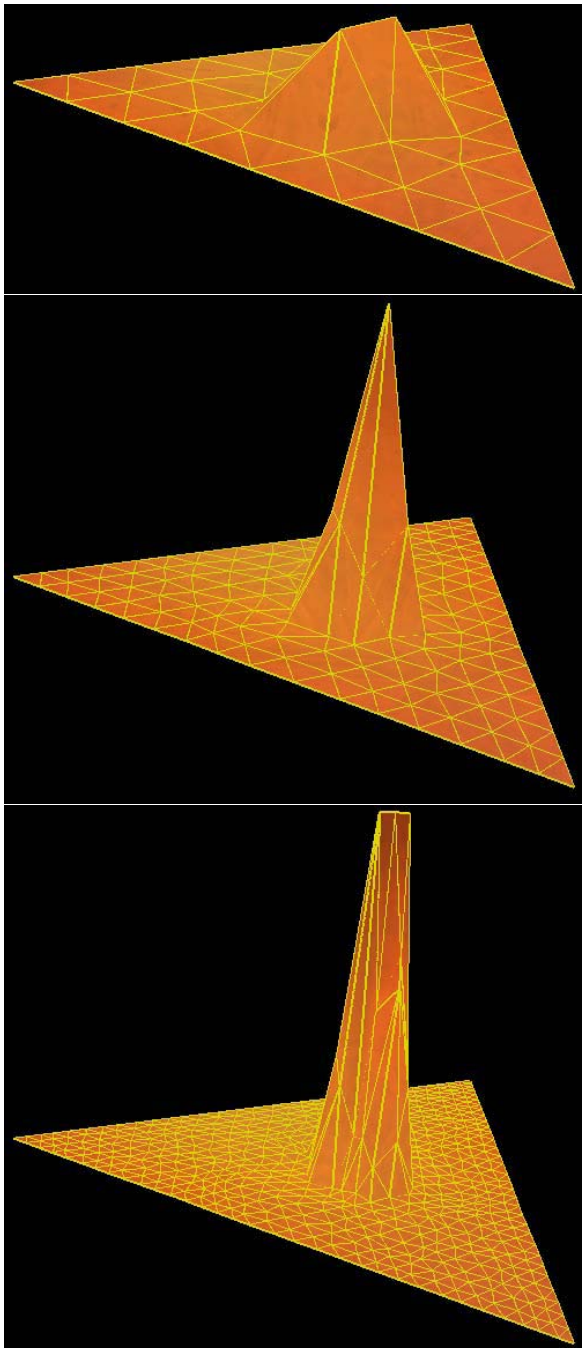

Figure 7. *Stiffness distribution for a subdivision on a triangular mesh. Top (a): (45 vertices, $1.0 \leq \kappa \leq 1.5$) Center (b): (153 vertices, $1.0 \leq \kappa \leq 2.5$) Bottom (c): (561 vertices, $1.0 \leq \kappa \leq 2.75$)*

Proceedings of the Computer Graphics International (CGI'04)
1530-1052/04 $20.00 © 2004 **IEEE**