A Physics-based Framework for Subdivision Surface Design with Automatic Rules Control

Hui Xie **Hong Qin**
Department of Computer Science, State University of New York at Stony Brook
Stony Brook, NY 11794-4400, U.S.A.
{xhui | qin}@cs.sunysb.edu

Abstract

The recent non-uniform subdivision approach extends traditional uniform subdivision schemes with variable rules, offering additional shape parameters (such as knot spacings) for feature control. Despite its flexibility, shape modification based on non-uniform subdivision usually requires designers to interactively adjust a large number of degrees of freedom (DOFs) to achieve the desired shapes, which can often be laborious. This paper extends the principle of variational subdivision and integrates the non-uniform subdivision schemes with powerful physics-based shape sculpting techniques, providing a universal method for arbitrary subdivision schemes with adjustable rules. The subdivision control points and knot spacings evolve in response to the shape deformation resulting from the numerical integration of Lagrangian dynamics equation or the optimization of shape energy functional. Thus, our system allows users to manipulate the desired shape in a direct and intuitive fashion. In addition, we propose a novel and efficient discrete functional evaluation method for polygonal meshes or point cloud of arbitrary topology based on implicit functions, in which no parameterization is needed. Finally, we develop a simple prototype sculpting system demonstrating many advantages of our novel physics-based, non-uniform subdivision modeling system.

Keyword: *Computer graphics, CAD, Physics-based modeling, subdivision.*

1 Introduction and Motivation

Recently, subdivision has been widely studied and exploited in graphical modeling systems because it provides a simple and uniform method for representing free-form surfaces with arbitrary topology to a certain degree of continuity without such cumbersome operations as trimming and patching. A recent extension to the conventional uniform subdivision scheme is the non-uniform subdivision scheme, which offers users the possibility of feature control. Typical features include sharp edges and sharp corners.

Despite the flexibility of subdivision schemes, existing subdivision sculpting systems (which are primarily based on manually manipulating the control vertices and directly specifying sharp features to each individual edge) are subject to some modeling difficulties: models for real-world objects often have an extremely large number of control points, causing manual editing and animation of a huge control mesh to be clumsy and laborious; models in CAD and animation are often subject to a set of aesthetic and engineering criteria, and meeting these criteria by indirectly moving the underlying control parameters may not be intuitive and effective; and design requirements from various application areas are often shape-oriented instead of control point oriented.

Physics-based modeling provides an effective solution to these difficulties by combining geometric objects with real-world physical attributes such as mass distribution, external forces, internal energy, etc. In our physics-based subdivision modeling environment, the shape control parameters are governed by physical laws and are determined through the numerical integration of dynamic equations. Thus, the new system incorporates not only the geometric features, but also physics-based properties to streamline the task of shape construction and modification.

Although physics-based modeling of subdivision surfaces has been available in recent years, existing methods primarily focus on the control points [7]. To fully realize the representation potential of non-uniform subdivision schemes, there has yet to emerge a systematic approach to take the non-uniform subdivision rules into the physics-based modeling framework. In this paper, we further incorporate the non-uniform subdivision rules, quantified as non-negative real values called knot spacings, into the generalized coordinates of a universal dynamic subdivision system. By employing subdivision knot spacings, more accurate control of surface curvature and shape features becomes possible.

2 Research Contributions

In this paper, we generalize *Dynamic Subdivision Surfaces* [7] to non-uniform subdivision schemes. The further incorporation of non-uniform subdivision rules to the already powerful physics-based subdivision sculpting tools makes possible much finer control of surface details such as normal, curvature, and more importantly, sharp features. This paper focuses mainly on non-uniform Catmull-Clark subdivision surfaces. However, the fundamental algorithms can be generalized to many other non-uniform subdivision schemes without any additional difficulty.

In our system, the user-specified physical properties (including mass density, damping distribution, external simulated forces, and energy functionals, as well as other physical and geometric constraints) govern the evolution of the dynamic Lagrangian equation. Among them, the potential energy functionals play a central role in the determination of the final surface shape. The energy functionals have the capability of quantifying user-centered aesthetic criteria, qualitative constraints, and functional requirements, serving as a basic sculpting toolkit for a large variety of applications, and providing a direct shape manipulation "language" to describe complicated models in an intuitive fashion, while avoiding the bewildering indirect manipulation of the underlying shape variables. This paper systematically discusses a number of commonly used functionals. Several primitive functionals are implemented in our system. We show that a weighted combination of this select set of primitive functionals can meet a large variety of requirements in many applications.

Moreover, we propose a discrete functional evaluation algorithm based on implicit surface approximation. Our new approach eliminates the need to find a local isometric parameterization of the surface.

3 Background

Our system is inspired by prior work. In this section, we discuss uniform and non-uniform recursive subdivision [11], variational subdivision [16, 3], D-NURBS [9], dynamic recursive subdivision [7] and our previous work.

Subdivision defines a curve or surface as the limit of a sequence of successively refined polygons or polyhedral meshes, respectively. Subdivision was first introduced as a fast curve rendering tool. Doo and Sabin and Catmull and Clark first initiated the use of recursive subdivision as a smooth surface definition method which overcomes the rigid topological limitation of tensor-product splines. Early subdivision schemes can be considered as a generalization to the well known spline-based knot-insertion algorithm. Later on, other forms of subdivision rules independent of splines were proposed. Fig. 1 presents the results of applying the Catmull-Clark subdivision scheme to generate a bicycle saddle.

Modeling through the use of subdivision can overcome many difficulties inherent to other modeling techniques. Here we summarize some of its important advantages:

1. *Topological Generality*: The generated surface is smooth everywhere, and no specific valence limitations are imposed on vertices.

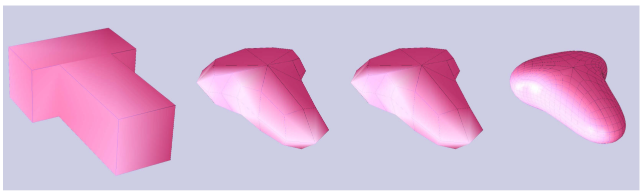
Figure 1. Uniform Catmull-Clark subdivision.

2. *Multiresolution*: Because of its recursive refinement structure, subdivision intrinsically supports level-of-detail rendering, multi-resolution editing, and other hierarchical algorithms.

3. *Uniformity of Representation*: Subdivision bridges the gap between discrete polygonal meshes and continuous spline patches. Many algorithms for either representation can find their applicability in subdivision settings.

4. *Properties Inherited from Splines*: Also, many other features of splines and polygonal meshes, such as local support, affine invariance, continuity, simplicity, etc, are inherited by subdivision.

Non-Uniform Recursive Subdivision Surfaces (NURSSes) are extensions of the non-uniform tensor product B-spline surfaces to arbitrary topological settings [11], analogous to the way that Doo-Sabin and Catmull-Clark surfaces are the generalization of uniform B-spline surfaces. The incorporation of non-uniform knot spacings to subdivision rules provides extra modeling capabilities to express features like creases and cusps, and the degree of sharpness of these features. The use of variable subdivision rules to model special features can be dated back to Hoppe and DeRose, et al [2]. Other similar tagging algorithms were subsequently proposed. Similar to Doo-Sabin and Catmull-Clark's schemes, NURSSes are based on non-uniform B-spline knot insertion algorithms. However, the knot spacings in the non-uniform subdivision scheme can be independently chosen, and they need not to satisfy a tensor product structure in which the knot spacings in the same row or column must take the same value. Fig. 2 shows the application of non-uniform Catmull-Clark subdivision to the same mesh as in Fig. 1 with various knot spacings attached. We can observe that small knot spacings tend to attract the limit surface toward that edge.

Variational Subdivision is based on the idea that one subdivision step can be considered as a topological splitting operation where new vertices are introduced to increase the degree of freedom, followed by a smoothing operation where the vertices are shifted in order to increase the overall smoothness [3]. Many existing subdivision rules are designed such that a particular quadratic energy functional can be minimized. Variational subdivision aims at finding a systematic approach toward the determination of subdivision rules when

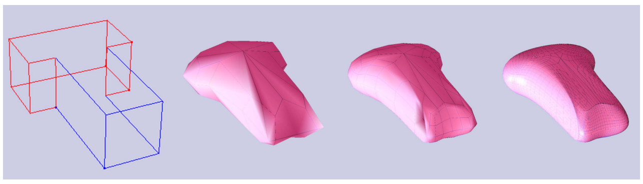

Figure 2. Non-uniform Catmull-Clark subdivision (knot spacings: red=1.00, blue=0.01).

given the quadratic form of an energy functional.

Kobbelt [3] and Mallet [5] proposed an interpolatory variational scheme. Warren and Weimer extended the variational approach to approximating subdivision schemes which minimize arbitrary quadratic functionals, details of which can be found in [13]. The idea of using energy functionals to derive the subdivision rules can be extended to much more generalized cases where neither the energy form needs to be confined in quadratic form, nor do the subdivision rules need to be stationary. This generalized form leads to one inspiration of our physics-based modeling framework.

Physics-based Modeling incorporates physical properties into the shape geometry, allowing the physical laws to govern the deformation of the shape to meet desired global and local modeling criteria and other geometric constraints in order to facilitate the interactive modeling process in an intuitive fashion. Free-form deformable models were first introduced to computer graphics by Terzopoulos et al [12]. Terzopoulos, Fleischer, Celniker, Gossard, Bloor, Wilson, Welch, Witkin and many others established the foundations of physics-based modeling through energy functional optimization subject to hard or soft geometric constraints. Qin and Terzopoulos [9, 10] developed the dynamic NURBS (D-NURBS), in which, the NURBS geometry, married with time, mass, force, and potential energy functionals, dynamically evolves through the time integration of the Lagrangian equation. Mandal and Qin [7] further applied Lagragian dynamics to subdivision models, providing the dynamic subdivision. Physics-based modeling alleviates the laborious shape manipulation process by indirectly positioning and moving a large number of shape variables, and provides an intuitive framework for real- world geometry description and modification.

Automatic NURBS Knot Determination [15] was developed to facilitate the automatic determination of non-uniform knot vectors as well as other control variables for NURBS curves and surfaces through the unified methodology of energy minimization, variational principles, and numerical techniques. To incorporate non-uniform knots into our optimization framework, we systematically transform general NURBS geometry into a set of geometrically equivalent rational Bézier splines. Thus, we facilitate the mathematical derivation of NURBS Jacobian matrix through both symbolic and numerical computation. Within our energy optimization methodology, the system can allow users to interactively manipulate NURBS geometry in an intuitive fashion via a large variety of sculpting tools (e.g., geometric constraints, energy functionals) without worrying about how to set up control points, non-unity weights, and/or non-uniform knots. This paper aims at extending the knot-varying D-NURBS framework to more powerful and topology-flexible non-uniform subdivision schemes.

4 Definitions for Non-uniform Recursive Subdivision

Non-Uniform Recursive Subdivision Surfaces (NURSSes) are extensions of non-uniform B-spline surfaces to arbitrary topology and arbitrary knot spacing settings. Non-uniform B-splines have the capability to model discontinuity by multiple knots. In non-uniform recursive subdivision schemes, the non-negative knot spacings attached to each edge take the place of non-decreasing knot vectors, and no longer need to be in a strict tensor product configuration.

In the interest of space, we refer readers to [11] for the definition of non-uniform Catmull-Clark and Doo-Sabin subdivisions.

5 Physics-based Sculpting Algorithms and Formulations

Having extended the representing flexibility of traditional uniform subdivision schemes by marrying them with non-uniform knot spacings, we further alleviate the labor of shape manipulation by incorporating physics information such as time, mass, force, and strain energy into the non-uniform subdivision formulation. In this section, we present the physics-based, dynamic non-uniform subdivision formulation and relevant numerical algorithms.

5.1 Dynamic Non-uniform Subdivision

Allowing the knot spacings to change makes it extremely difficult to find an analytic formulation for the limit surface. We iteratively subdivide our initial control polygon to a satisfactory level (normally 3–4 levels since the geometric components proliferate exponentially), and view each vertex at the finest-level as a particle. The geometry of the finest-level surface, along with its attached physical attributes, comprises the particle system with which we are concerned (see Fig. 3).

A particle system can be characterized by the position $\mathbf{s}_i(t)$, velocity $\dot{\mathbf{s}}_i(t)$, mass μ_i and damping γ_i of each particle i, along with the inner potential energy $E(\mathbf{s}_0, \cdots, \mathbf{s}_n)$, simply denoted as $E(\mathbf{s})$. Among these symbols, the overstruck dot denotes a time derivative. We can now formulate

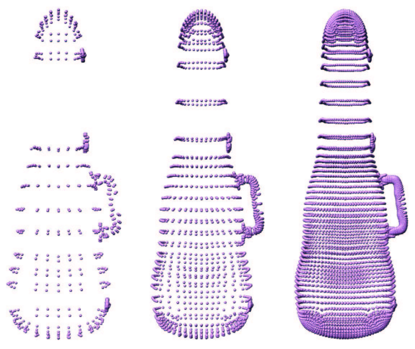

Figure 3. Subdivision surface as a particle system.

the standard dynamics of this particle system as:

$$\mu_i \ddot{\mathbf{s}}_i + \gamma_i \dot{\mathbf{s}}_i + \frac{\partial E(\mathbf{s})}{\partial \mathbf{s}_i} = \mathbf{f}_i^a, \quad \forall i, \tag{1}$$

where $\mathbf{f}_i^a$ refers to external applied force, and $\partial E(\mathbf{s})/\partial \mathbf{s}_i$ can be denoted as $-\mathbf{f}_i^p$, i.e., the potential induced force.

Prior to further discussion, we first clarify some terminology. We refer to the finest-level points as particles, and the coarsest level points (the initial mesh vertices) in our subdivision hierarchy as control points. The evolution of our subdivision surface over time can be uniquely characterized by the behavior of its control points $\mathbf{p}_i$ and knot spacings t_i, which in dynamics are collectively called generalized coordinates, denoted as g_i. Now, we treat these generalized coordinates as time variables, i.e., as functions of time $g_i(t)$. We can exploit Lagrangian dynamics to derive the equation governing the dynamic behavior of this set of generalized coordinates. The matrix form of Lagrangian work-energy equation is

$$\mathbf{M}\ddot{\mathbf{g}} + \mathbf{D}\dot{\mathbf{g}} + \frac{\partial E(\mathbf{g})}{\partial \mathbf{g}} = \mathbf{f}^a - \mathbf{B}\dot{\mathbf{g}}, \tag{2}$$

where $\mathbf{M}$ is called mass matrix which can be derived as

$$\mathbf{M} = \sum_i \mu_i \mathbf{J}_i^\top \mathbf{J}_i, \tag{3}$$

where μ_i is the mass of particle i, and $\mathbf{J}_i$ is the Jacobian of $\mathbf{s}_i$ with respect to its generalized coordinates g_i, that is,

$$\mathbf{J}_i = \begin{pmatrix} \frac{\partial \mathbf{s}_i}{\partial \mathbf{p}_0} & \cdots & \frac{\partial \mathbf{s}_i}{\partial \mathbf{p}_n} & \frac{\partial \mathbf{s}_i}{\partial t_0} & \cdots & \frac{\partial \mathbf{s}_i}{\partial t_k} \end{pmatrix}. \tag{4}$$

Similarly, the damping matrix $\mathbf{D}$ is defined as

$$\mathbf{D} = \sum_i \gamma_i \mathbf{J}_i^\top \mathbf{J}_i, \tag{5}$$

with γ_i as the damping coefficient. The generalized force vector, obtained through the principle of virtual work [1] done by the applied force distribution $\mathbf{f}_i(t)$ is

$$\mathbf{f}^a = \sum_i \mathbf{J}_i^\top \mathbf{f}_i(t). \tag{6}$$

Unfortunately, the Jacobian is not constant over time. This results in variable $\mathbf{M}$, $\mathbf{D}$ and an additional non-zero matrix,

$$\mathbf{B}(\mathbf{g}) = \sum_i \mu_i \mathbf{J}_i^\top \dot{\mathbf{J}}_i.$$

The potential energy $E(\mathbf{g})$ can adopt a large variety of functionals that result in different behaviors of our physics-based non-uniform subdivision. We will discuss this issue in the following subsections.

5.2 Numerical Integration of Lagrangian Equation

We can discretize the time-continuous Lagrangian equation in (2) to a finite difference equation. Note that, $\mathbf{B}$ is $\dot{\mathbf{M}}$,

$$(\mathbf{D}^t + \frac{\mathbf{M}^{t-2\Delta t} - 4\mathbf{M}^{t-\Delta t} + 3\mathbf{M}^t}{2\Delta t})\frac{\mathbf{g}^{t+\Delta t} - \mathbf{g}^{t-\Delta t}}{2\Delta t}$$

$$+\mathbf{M}^t \cdot \frac{\mathbf{g}^{t+\Delta t} - 2\mathbf{g}^t + \mathbf{g}^{t-\Delta t}}{\Delta t^2} + \frac{\partial E(\mathbf{g}^t)}{\partial \mathbf{g}} = \mathbf{f}^a, \tag{7}$$

where the superscript denotes evaluation of $\mathbf{g}$ at the time indicated.

In the equation, $\mathbf{g}^{t+\Delta t}$ is the unknown to be solved in each iterative step. We do not use $(\mathbf{M}^{t+\Delta t} - \mathbf{M}^{t-\Delta t})/2\Delta t$ to approximate $\dot{\mathbf{M}}$ because we employ a forward method.

This equation will lead to an algebraic equation,

$$\mathbf{A}\mathbf{g}^{t+\Delta t} + \mathbf{b} = 0, \tag{8}$$

where $\mathbf{A}$ and $\mathbf{b}$ are functions of previous state $\mathbf{g}^{t-2\Delta t}$, $\mathbf{g}^{t-\Delta t}$ and $\mathbf{g}^t$. The equation gives the current state $\mathbf{g}^{t+\Delta t}$. Thus the motion evolves along the time axis. The value of Δt is assigned in consideration of the tradeoff between accuracy and speed.

We can use Gaussian elimination method or conjugate gradient method to solve (8). For the latter case, (8) is equivalent to minimizing $\mathbf{g}^\top \mathbf{A}^\top \mathbf{A}\mathbf{g} - 2\mathbf{b}^\top \mathbf{A}\mathbf{g}$.

5.3 Energy Optimization

One disadvantage is that the incorporation of non-uniform knot spacings results in a non-stationary Jacobian throughout the dynamic evolution. We can not use any precomputation of the matrices as in [7]. This impairs the performance considerably.

Note that the equilibrium state of the our dynamic system in (2) can be simplified as:

$$\frac{\partial E(\mathbf{g})}{\partial \mathbf{g}} = \mathbf{f}^a|_{t=\infty} \Leftrightarrow \frac{\partial (E(\mathbf{g}) - \mathbf{g} \cdot \mathbf{f}^a|_{t=\infty})}{\partial \mathbf{g}} = 0. \tag{9}$$

It turns out to be a standard optimization problem of minimizing $E(\mathbf{g}) - \mathbf{g} \cdot \mathbf{f}^a|_{t=\infty}$.

Many mature optimization solvers can be employed for our system with satisfactory efficiency. Unlike variational subdivision, which is a level-by-level optimization, we only take the first level control points and knot spacings as generalized dynamic coordinates. A more advanced methodology, where the k^{th} level control points and knots can also be incorporated into our generalized coordinates, can be easily constructed (see [6]).

6 Discrete Energy Functionals for Subdivision Surface

The energy functionals, along with the external forces, will determine the final shape of our model, while other physics-based parameters such as mass distribution, and damping coefficients, only affect the in-between animation in the dynamic process of shape evolution.

In our system, many sculpting tools are implemented by specifying a proper energy functional. This energy functional is not necessary to have a strict physical meaning. A typical energy functional can be in the form of an integral of surface normal, curvature, differential area, the variation of curvature, etc. over the interested surface region:

$$\iint_{\Omega} F[\mathbf{s}]\mathrm{d}S. \tag{10}$$

For parametric surfaces, the integral is defined on a rectangular parametric domain, and can be evaluated by Gaussian quadrature or other forms of sampling schemes. The integrand can be easily constructed as an analytic combination of its partial derivatives with respect to the parameters [15],

Now we shall calculate energy functionals on subdivision surfaces with variable rules. Traditional uniform Catmull-Clark surfaces can be seen as a collection of B-spline patches. In [8], a systematic approach was proposed to transform Catmull-Clark surfaces to B-spline surface patches, each of which is defined over a parametric domain $[0,1]^2$, with special consideration made for patches near extraordinary vertices. This transformation allows the straightforward use of the parametric surface energy functionals developed for B-splines. Unfortunately, non-uniform subdivision surfaces and many non-B-spline subdivision schemes (such as Butterfly subdivision) can not follow this approach, due to the difficulties involved in finding an analytic formulation for the limit surface over rectangular domain patches. Mandal and Qin [6] developed a mutilevel structure using the recursively subdivided surface to approximate the limit surface, similar to the particle system developed in this paper. In this paper, however, we develop a more accurate and generalized formulation for normal, curvature and other forms of energies, as well as provide a universal methodology for energy functional construction.

6.1 Membrane Energy and Thin Plate Energy

The most frequently used energy functionals in geometric surface modeling are the first order strain energy (membrane energy) and curvature energy:

$$E_{mem}(\mathbf{s}) = \frac{1}{2}k \iint_{\Omega} \epsilon^2 \ \ \mathrm{d}S, \tag{11}$$

$$E_{curv}(\mathbf{s}) = \frac{1}{2}k \iint_{\Omega} \kappa_1^2 + \kappa_2^2 \ \ \mathrm{d}S, \tag{12}$$

where ϵ is the strain and κ_1, κ_2 are the two principal curvatures. For surface with isometric parameterization, assuming exerted small displacement or bending, they can be expressed as:

$$E_{mem}(\mathbf{s}) = \frac{1}{2}k \iint_{\Omega} \mathbf{s}_u^2 + \mathbf{s}_v^2 \ \ \mathrm{d}u\mathrm{d}v, \tag{13}$$

$$E_{curv}(\mathbf{s}) = \frac{1}{2}k \iint_{\Omega} \mathbf{s}_{uu}^2 + 2\mathbf{s}_{uv} + \mathbf{s}_{vv}^2 \ \ \mathrm{d}S. \tag{14}$$

The latter is also called thin plate energy. As we have pointed out, parameterization-independent forms of these formulations over discretized meshes are necessary for our particle system, as we do not want to construct an explicit local parameterization.

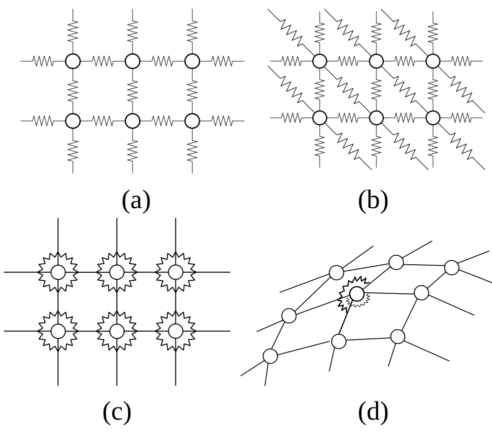

Figure 4. Mass-spring network resisting stretching, shearing and bending.

A practically effective and efficient method of approximating internal energy is by using mass-spring model, in which each mass point is connected by springs with its nearest neighbors. Ordinarily, the springs are directly assigned to the edges in the finest subdivided surface, (see 4(a)). If the finest-level configuration is not triangular based, diagonal springs may be needed to resist shearing, (see Fig. 4(b)). Another method for imposing shearing energy in rectangular settings is to introduce angular springs, as shown in Fig. 4(c). Angular springs resist the change in angle between each pair of adjacent edges.

The expression for the membrane tension energy can be discretely defined as:

$$E_{mem} = \frac{1}{2}\sum_{i,j} k_{ij} \ (\mathbf{v}_i - \mathbf{v}_j)^2, \tag{15}$$

where the summation is defined over all edges $\mathbf{e}_{ij} = \overrightarrow{\mathbf{v}_i\mathbf{v}_j}$. When the rest length is not small enough, the typical formula of spring energy may be helpful:

$$E_{mem} = \frac{1}{2}\sum_{i,j} k_{ij}\left(|\mathbf{v}_i - \mathbf{v}_j| - r_{ij}\right)^2, \tag{16}$$

where the summation is defined over all edges $\mathbf{e}_{ij} = \overrightarrow{\mathbf{v}_i\mathbf{v}_j}$, and r_{ij} is the rest length. The angular spring energy is defined as:

$$E_{ang} = \frac{1}{2}\sum_{i,j} \alpha_{ij}\Delta\theta_{ij}^2, \tag{17}$$

where summation is defined over all adjacent edge pairs $(\mathbf{e}_i\mathbf{e}_j) = (\overrightarrow{\mathbf{v}_i\mathbf{v}_k}, \overrightarrow{\mathbf{v}_j\mathbf{v}_k})$, and $\Delta\theta_{ij}$ is the displacement of the angle between these two adjacent edges. If the angular displacement is infinitesimal, we can derive the following expression for $\Delta\theta_{ij}$:

$$\Delta\theta_{ij} = \frac{\cos\phi_{ij} - \cos\theta_{ij}}{\sin\phi_{ij}}, \tag{18}$$

where ϕ_{ij} and θ_{ij} are the angles before and after deformation, respectively. Note that:

$$|\mathbf{e}_i||\mathbf{e}_j|\cos\theta_{ij} = \mathbf{e}_i\cdot\mathbf{e}_j = \frac{1}{2}(\mathbf{v}_i+\mathbf{v}_j-2\mathbf{v}_k)^2 - \frac{1}{2}\mathbf{e}_i^2 - \frac{1}{2}\mathbf{e}_j^2. \tag{19}$$

In many literatures, $(\mathbf{v}_i+\mathbf{v}_j-2\mathbf{v}_k)^2$ is used to impose shearing strain. This approach works well for those $\mathbf{e}_i$ and $\mathbf{e}_j$ pairs which have strong stretching stiffness and are isometric. However, experiments show that it still effectively works well even in circumstances of larger stretching deformation.

The bending energy is directly related to the change of total curvature, $\kappa_1^2 + \kappa_2^2$. One way to impose curvature energy is by introducing another set of angular springs, as shown in Fig. 4(d). This approach only provides a rough approximation of the total curvature, as it also includes some shearing energy, and only works when the edge lengths are near isometric.

As we have mentioned, our pseudo-physical properties can also accommodate geometric constraints. A satisfactory evaluation of the normal, curvature and other geometric quantities is critical in applications such as minimizing the variation of curvature. Before we introduce them, we present a novel efficient algorithm for discrete evaluation of normal and curvature through implicit surface approximation.

6.2 Discrete Normal and Curvature Evaluation through Implicit Surface Approximation

Normal and curvature evaluations serve as the basic building blocks for many geometric and physical energy functionals. Traditionally, the discrete curvature approximation involves the introduction of an isometric local parameterization[4, 14]. A typical procedure includes:

1. Define a neighborhood for each vertex.
2. Choose a parameterization for the neighborhood.
3. Find the approximating polynomial.
4. Evaluate normal/curvature.

An isometric local parameterization entails finding a local tangent plane, and mapping the surface points to this tangent plane. In this paper, we introduce a novel implicit function based approach to the discrete computation of the normal and curvature at a surface point. Within this new approach, the step of parameterization is no longer necessary. Instead, our approximating surface is defined by an isosurface of a quadratic implicit function. We will show that our approach is no more expensive than that developed in [4].

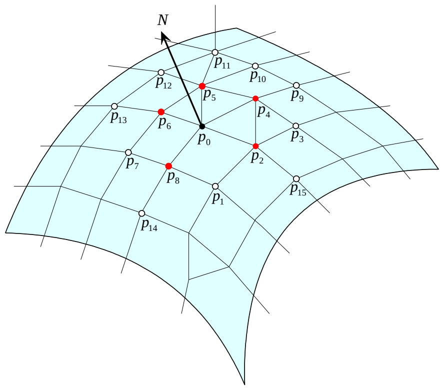

Figure 5. Approximating implicit surface.

To best convey our idea, we compute the normal and curvature at a point $\mathbf{p}_0$, in Fig. 5, for instance.

First, we choose the neighborhood of vertex $\mathbf{p}_0$. Depending on our accuracy requirement, we can choose its 1-neighbors, i.e., the solid points $\mathbf{p}_0$, $\mathbf{p}_2$, $\mathbf{p}_4$, $\mathbf{p}_5$, $\mathbf{p}_6$ and $\mathbf{p}_8$, or 2-neighbors $\mathbf{p}_0, \cdots, \mathbf{p}_{15}$.

Then, we construct an implicit surface $f(x, y, z) = 0$ approximating $\mathbf{p}_0$ and its neighbors. The form of f can be linear or higher order polynomials. The basic point is that, the number of unknown coefficients in f should be no more than that of $\mathbf{p}$'s. If we assume f is a sphere, i.e., $f(x, y, z) = x^2 + y^2 + z^2 - 2x_0x - 2y_0y - 2z_0z - c = 0$, at least 4 vertices are required. We construct the following equations:

$$f_i(x_i, y_i, z_i) = 0, \quad \forall\mathbf{p}_i. \tag{20}$$

We have a set of linear equations:

$$\mathbf{A}\mathbf{d} = \mathbf{c}; \quad \mathbf{d} \equiv (x_0, y_0, z_0, c)^\top, \tag{21}$$

where $\mathbf{d}$ collects all the unknown coefficients. We can solve this typically over-constrained linear system by least square fitting:

$$\mathbf{A}^\top\mathbf{A}\mathbf{d} = \mathbf{A}^\top\mathbf{c}. \tag{22}$$

Now we are able to calculate the normal $\mathbf{n}$, Gaussian curvature K, and mean curvature H by

$$\begin{cases} \mathbf{n}(\mathbf{p}_0) = \frac{1}{D}\nabla f(\mathbf{p}_0) \\ K(\mathbf{p}_0) = \frac{1}{D^2}\mathbf{n}^\top \mathbf{M}^* \mathbf{n}, \\ H(\mathbf{p}_0) = \frac{1}{2D}\mathbf{n}^\top \mathbf{H}\mathbf{n} \end{cases} \tag{23}$$

where $D = |\nabla f(\mathbf{p}_0)|$, $\mathbf{M}$ is the Hessian matrix of f, $\mathbf{M}^*$ is the adjoint of $\mathbf{M}$, and $\mathbf{H} = \mathbf{M} - trace(\mathbf{M})\mathbf{I}$, details of which can be found in Appendix A.

Generally, we can assume f is a quadratic surface:

$$ax^2 + by^2 + cz^2 + dxy + eyz + fzx + gx + hy + iz + j = 0.$$

Since $\mathbf{d} = 0$ is a trivial solution, a non-trivial solution needs to be figured out as the eigenvector of matrix $\mathbf{A}^\top \mathbf{A}$ corresponding to the least eigenvalue. Since the eigenvector problem for a 10×10 matrix is expensive, we prefer a linear equation. At least one more condition about the coefficients is required to obtain a non-trivial linear equation. Based on the knowledge of the shape, different choices can be made. We can assume $a = 1$ if we know $a \neq 0$, or $a + b + c + d + e + f = 1$, if we know it is not a plane.

The following procedure provides a robust algorithm to finding such a condition.

```
1. Translate $\mathbf{p}_0 \cdots \mathbf{p}_n$ such that $\mathbf{p}_0 = 0$.
2. Assume $f$ interpolate $\mathbf{p}_0 = 0$, thus $j = 0$.
3. Find a tentative normal $\tilde{\mathbf{n}}$ at $\mathbf{p}_0$. Since $\nabla f(\mathbf{p}_0 = 0) = (g, h, i)^\top$, $(g, h, i)^\top \approx \tilde{\mathbf{n}}$.
4. Assign 1.0 to one of $g$ or $h$ or $i$ with the maximum absolute tentative value.
5. Solve the remaining equation of only 8 unknowns.
```

A simple cross-product between two adjacent edges can provide a good tentative normal. The cost of the above procedure is negligible, with a good by-product of reducing the number of unknowns to 8.

When the implicit function f is a polynomial, all those quantities in Eqn. 23 are not expensive to compute. For the quadratic case, evaluating $\mathbf{M}$ and $\mathbf{M}^*$ needs 0 and 12 multiplications, respectively. This method can be easily generalized to point cloud and other structures.

If we view the discrete energy functionals defined in the preceding subsection as "qualitative" functionals, the implicit surface approximating algorithm provides a precise approach to the discrete evaluation of the surface normal, curvature and other quantities. In comparison with other discrete curvature evaluation approaches, this method is clean, compact, and efficient.

6.3 General Functionals

Our system provides a general modeling environment in the sense that it can accommodate a diverse set of energy functionals for applications of different purposes. We list some of the commonly used energy functionals implemented in our system. A linear combination of these functionals may address most practical requirements emerging in current modeling literature. In the following list are discretized versions of their corresponding continuous forms to polygonal meshes. $\mathbf{v}'_i$, $\mathbf{e}'_i$ and f'_i represent the original vertex $\mathbf{v}_i$, edge $\mathbf{e}_i$ and face f_i before deformation, respectively.

1. *Membrane Strain Energy*:

$$E_{hetero} = \sum_i k_i |\mathbf{e}_i|^2, \tag{24}$$

$$E_{iso} = \sum_i \left(\frac{1}{n(f_i)} \sum_{\mathbf{e}_j \in f_i} \frac{|\mathbf{e}_j|^2}{|\mathbf{e}'_j|^2} \right) S(f'_i), \tag{25}$$

where $n(f)$ denotes the valence of face f, $S(f')$ is the area of face f before deformation. $S(f')$ can also be replaced by $S(f)$. E_{hetero} is roughly equivalent to the integral in Eqn. 13 with a heterometric parameterization proportional to the original edge length; while E_{iso} provides a good approximation to the integral with an isometric parameterization.

2. *Spring Energy*:

$$E_{sp} = \sum_i (|\mathbf{e}_i| - r_i)^2, \tag{26}$$

where r_i is the rest length of edge $\mathbf{e}_i$.

3. *Integral of Principal Curvatures κ_1 and κ_2*:

$$E_{curv} = \sum_i (c_1\,\kappa_1^2 + c_2\,\kappa_1\kappa_2 + c_3\,\kappa_2^2)\; S(f'_i). \tag{27}$$

Some frequently used forms are the total curvature $\kappa_1^2 + \kappa_2^2$, Gaussian curvature $\kappa_1\kappa_2$, squared mean curvature $\frac{1}{4}(\kappa_1 + \kappa_2)^2$. An averaging process over a face can be similarly applied as in Eqn. 25.

4. *Curvature Preserving over a Region*:

$$E_{curvprsv} = \sum_i |K(\mathbf{v}_i) - K(\mathbf{v}'_i)|^2, \tag{28}$$

where $K(\mathbf{v}_i)$ denotes the total curvature at point $\mathbf{v}_i$. This functional conveys the idea of shape preserving. A hierarchical implementation of this tool would allow for as-rigid-as-possible transformation.

5. *Surface Area Preserving over a Region*:

$$E_{area} = |S_0 - \sum_i S(f_i)|^2. \tag{29}$$

S_0 is the designated area and $S_0 = 0$ results in area minimization.

6. *Variation of Local Surface Area*:

$$E_{rub} = \sum_i |S(f_i) - S(f'_i)|^2, \tag{30}$$

where $S(f'_i)$ denotes the original area of face f_i. This formula allows the simulation of the behavior of rubber.

7. *Normal and Position Control of a Face Point*:

$$E_{norm} = \sum_i k_i |\mathbf{n}(\mathbf{v}_i) - \mathbf{n}_i^0|^2, \tag{31}$$

$$E_{pos} = \sum_i k_i |\mathbf{v}_i - \mathbf{v}_i^0|^2. \tag{32}$$

These two functionals can be thought of as imposing external angular or line springs connecting surface points to a designated direction $\mathbf{n}_i^0$ or position $\mathbf{v}_i^0$, respectively.

8. *Least Motion and Uniform Distribution of Control Points*:

$$E_{lm} = \sum_i k_i |\mathbf{p}_i - \mathbf{p}_i'|^2, \tag{33}$$

$$E_{ud} = \sum_i k_i |\mathbf{e}_i|^2. \tag{34}$$

Instead of being defined on the limit surface, these two functionals are defined on the initial control mesh to impose additional constraints for under-determined systems (refer to [15]).

7 Numerical Method

7.1 Numerical Jacobian

Many subdivision schemes are generalization of splines. For spline-based schemes, analytic formulation of the limit surface can be found near regular points. Generally, non-spline subdivision and NURSSes do not have such property. The calculation of their Jacobian must resort to numerical methods, and can be expensive. Basically, the numerical Jacobian formulation,

$$\frac{\partial \mathbf{s}_j}{\partial g_i} = \frac{\mathbf{s}_j(\mathbf{g} + \Delta g_i) - \mathbf{s}_j(\mathbf{g} - \Delta g_i)}{2\Delta g_i}$$

can give an adequately satisfactory approximation with error bound $O(\Delta g_i^2)$. We employ a level-by-level updating approach to calculate the perturbed surface. Note that, when perturbing a shape variable, only a part of the surface in each level needs to be updated. This local control property of subdivision surfaces can drastically reduce the computational expense for gradient evaluation.

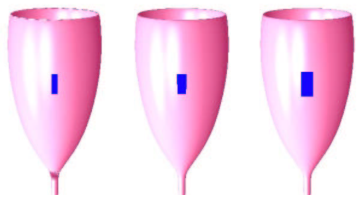

Figure 6. Updating propagation on Catmull-Clark subdivision surface.

An updating propagation method is employed in our system. All the updated vertices in each level of mesh are marked. As we have mentioned, most functionals take the form of an integral, the discrete functional evaluation procedure can use these marks to decide the region that needs recomputing. For Catmull-Clark surfaces, the updated area on the limit surface is about 4 times the updated area in Level 2 after perturbing a control point or knot spacing, as illustrated in Fig. 6.

7.2 Imposing Positive Knot Spacing Constraint

In NURSSes, because negative knot spacings result in undesirable shape (due to zero denominator in weight calculation), it is necessary to constrain the knot spacings to be non-negative. To simplify the system implementation, we further require all the knot spacings to be larger than a small positive number ϵ. We design the following algorithm to enforce this positive knot spacing constraint in our dynamic modeling system, which is much better than the penalty functional method in terms of performance.

```
1. At each integral step of solving Equation (7)
2.     For all knot spacings $k_i \in \mathbf{g}^t$
3.         If $k_i < \epsilon$ and $f_i^p + f_i^a < 0$ Set $f_i^p + f_i^a = 0$
4.     Calculate the new $\mathbf{g}^{t+\Delta t}$
5.     For all knot spacings $k_i \in \mathbf{g}^{t+\Delta t}$
6.         If $k_i < \epsilon$ Set $k_i = \epsilon$
7. Continue to the next step.
```

We have also proposed procedures to efficiently impose non-negative knot spacing constraint into our energy functional optimization framework. In the interest of space, we refer readers to [15] for details.

8 Energy-based Modeling Tools

This section outlines the typical physics-based procedures in our geometric modeling system. We also document a set of popular energy functionals we developed, which can be combined to address a large variety of modeling and design applications.

1. *Material Painting*: In our dynamic modeling system, the geometric surfaces are treated as physical thin plates, in which physical properties are not necessarily uniformly distributed. The non-uniform geometric and physical properties allow extra modeling flexibility such as confining physical tools to a small surface region, and characterizing non-uniform deformation.

2. *Spring-based Tools*: In our system, we use springs to implement external forces applied to a surface point. The quadratic nature of springs reduces the probability of divergence.

3. *Angular Spring Tools for Normal Control*: Users can specify a new normal direction at an arbitrary surface point, and the system subsequently figures out the corresponding changes to its underlying control points and knot spacings.

Since knot spacings are very sensitive to normal change, we employ a two-stage algorithm in which the control points are resolved before knot spacings to handle numerical instability. This process can be repeated several times.

4. *Area-based Tools*: Our area-based toolkit provides two forms of area functionals: a global area preservation functional and a differential area preservation functional. Global area preservation is not sensitive to area "flowing" from one surface patch to another. Essentially, local area preservation functional provides an approach to approximating rubber-like materials, which respond to the squeeze in one direction by protruding in another. In addition, our area preserving tools support deflation and inflation operations. Area minimization is merely a special case of deflating to zero.

5. *Curvature-based Tools*: Our system supports functionals of Gaussian curvature and mean curvature, built on our implicit surface approximating algorithm. The total curvature functional characterizes the resistance to bending forces. By minimizing curvature change of a surface region over time, we are able to preserve the local shape under certain boundary conditions. Variation of curvature with respect to space coordinates is also supported by means of linear approximation or quadratic approximation.

9 Experiments

In the color-plate of this paper, we present several representative examples, each of which takes a weighted combination of the above techniques. In real applications, a single functional oftentimes may not achieve our goals for graphical modeling. By combining them together, we can get functionals much closer to the real-world physics. Note that, systematic approaches toward robust and efficient functionals is still an open issue. Note that, in all knot spacing maps, the knot spacings have been linearly interpolated onto faces, and red stands for knot spacings which are bigger than one, while green stands for knot spacings which are smaller than one. The initial (before deformed) shapes are assumed unity-knot spacing distribution everywhere.

Fig. 9 presents the manipulation of a hand through the use of springs, differential area preserving tools and membrane energy functionals. Note that, (a) defines the initial control mesh. We only allow a subset of the control points to move at any time, which are marked in green. In (b) we exert force to the thumb tip in the direction towards the central red point, with differential area preserving functional enforced in the surface region in green, and the whole hand under membrane tension. After the deformation performed in (b), the thumb reaches the central red point without losing its shape. Similarly applying this procedure to three of the other fingers as illustrated (d), we have the desired model in (e). For larger control meshes, the application of a much more expensive constraint to curvature preserving is also necessary, instead of only area-preserving. The manipulation process from (d) to (e) takes 371 seconds on a 1GHz PC. Finally, (f) shows the knot spacing map of (e).

The characters in Fig. 10 are made by force (spring) tools subject to differential area preservation and strain energy minimization constraints. Each character has 24–30 control points. The bottom figure shows the knot spacing map.

Fig. 11 shows that material properties influence the evolution of the shape over time considerably. In (a), the green points in the top figure are allowed to move. The middle rod with smaller stiffness coefficient in the dark area yields more deformation, while the bottom rod with stronger stiffness transmits the spring force to its neighbor area. In (b), the whole rod is subject to differential area preserving constraints. Essentially, this makes a good approximation of rubber.

Fig. 12 shows the application of the total curvature minimization tool to subdivision surface in (a). (b) is the initial control polygon. We only allow the side points (green points) to be movable as illustrated in (c). After becoming curvature minimization, the movable control points reach their new positions, as shown in (d). In (e), the new shape looks much more flat. (f) and (g) present the curvature maps of (a) and (e), respectively. In the curvature maps, red color denotes higher total curvature, while green represents lower total curvature. (h) shows the knot spacing map of the deformed shape in (e). Note that, the shape in (c), (d) and (h) have been rotated by about 45 degrees to give a better side view. This example takes 154 seconds on a 1GHz PC.

10 Data Structure and Implementation

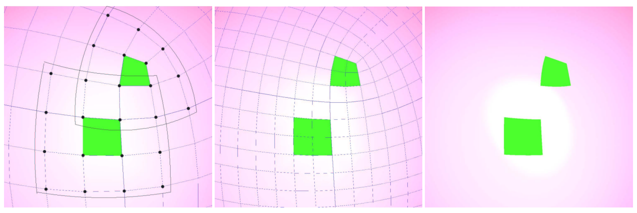

Figure 7. The face splitting property of Catmull-Clark surface.

Our system adopts a finite element data structure, based on the fact that the limit surface can be divided into a collection of smooth polygonal patches, each of which originates from recursively splitting a corresponding face in the original control mesh, and is determined by a finite number of control points within the vicinity of this face, as illustrated in Fig. 7 (with relevant control points encircled for each patch in the leftmost figure). We treat each of these smooth surface patches as an element. Within the finite element data structure, we have the flexibility to assign different mass, damping, energy functionals, applied forces and other properties on an element base. In addition, we have (taking $\mathbf{M}$ for instance)

$$\mathbf{M} = \sum_j \mathbf{M}_j = \sum_j \sum_{i \in E_j} \mu_i \mathbf{J}_i^\top \mathbf{J}_i, \qquad (35)$$

where j runs through all the elements. In $\mathbf{M}_j$, only the entries relevant to the finite number of control vertices and knot spacings of patch j are non-zero, which compose a submatrix. Thus, we can derive a parallel structure as illustrated in Fig. 8. The matrices $\mathbf{M}$, $\mathbf{D}$, applied force vector $\mathbf{f}^a$ and the gradient of the potential $\mathbf{f}^p$ can be evaluated patch-by-patch before they are assembled (summed) together. Each patch holds pointers to a set of generalized coordinates, as well as its local geometric and physical properties.

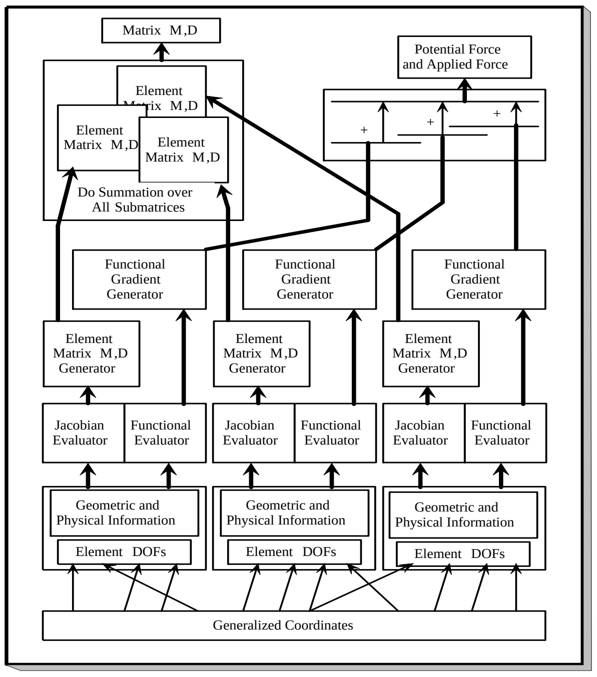
Matrix M,D
Element Matrix M,D
Element Matrix M,D
Element Matrix M,D
Do Summation over All Submatrices
Potential Force and Applied Force
+
+
+
Functional Gradient Generator
Functional Gradient Generator
Functional Gradient Generator
Element Matrix M,D Generator
Element Matrix M,D Generator
Element Matrix M,D Generator
Jacobian Evaluator
Functional Evaluator
Jacobian Evaluator
Functional Evaluator
Jacobian Evaluator
Functional Evaluator
Geometric and Physical Information
Element DOFs
Geometric and Physical Information
Element DOFs
Geometric and Physical Information
Element DOFs
Generalized Coordinates

Figure 8. FEM-based data structure and data flow.

11 Conclusions

In this paper, we have presented a new dynamic surface sculpting system based on the non-uniform subdivision surface. By incorporating the non-uniform subdivision rules into our dynamic framework, we have greatly enhanced the modeling capability of the prior dynamic subdivision system. Within our novel physics-based modeling framework, the subdivision control points as well as their associated non-uniform subdivision rules (knot spacings) dynamically respond to the user-specified mass distribution, damping coefficients, applied forces, energy functionals, and other physical and geometric constraints through the integration of the Lagrangian equation in an intuitive and predictable manner. We have developed a unified approach for efficient evaluation of subdivision surface Jacobian, energy functional gradient, with respect to both control points and knot spacings based on a discrete particle system. We have also proposed a systematic procedure constructing energy functionals, which play a central role in the determination of the final surface shape. A variety of basic energy functionals are discussed and compared. We demonstrated that a combination of these primitive functionals can be relevant to a large variety of applications. Based on the implicit surface approximation, a novel normal and curvature evaluation algorithm has been proposed, which avoids any local isometric parameterization and thus simplifies the approach proposed in [4].

We have also built a prototype interactive sculpting system based on our new dynamic non-uniform subdivision formulation and demonstrated that many applications benefit from the flexibility of variable rules control. This unified dynamic non-uniform subdivision sculpting system will exhibit its great potential in the realm of geometric modeling, virtual environment, engineering design, etc.

Acknowledgments

This research was supported in part by the NSF CAREER award CCR-9896123, the NSF grant DMI-9896170, the NSF ITR grant IIS-0082035, the NSF grant IIS-0097646, Alfred P. Sloan Fellowship, and Honda Initiation Award. We would also like to thank Kevin T. McDonnell and David Goldstein for their valuable suggestions on our paper writing.

A Implicit Normal and Curvature

In this section, we derive the formulation of normal and curvature evaluation for an implicit surface $f(x, y, z) = 0$. Assuming $\mathbf{p} = (x, y, z)^\top$ is a point on the implicit surface, we can construct a parameterization for this implicit surface as:

$$\mathbf{p}(u, v) = \begin{pmatrix} x & y & z \end{pmatrix}^\top = \begin{pmatrix} u & v & z(u, v) \end{pmatrix}^\top.$$

The differentials of each component satisfy,

$$f_x \mathrm{d}x + f_y \mathrm{d}y + f_z \mathrm{d}z = 0.$$

Thus,

$$\left\{ \begin{array}{lll} \frac{\partial \mathbf{p}}{\partial u}\Big|_{u,v} & = \begin{pmatrix} \partial x/\partial u \\ \partial y/\partial u \\ \partial z/\partial u \end{pmatrix} & = \begin{pmatrix} 1 \\ 0 \\ -f_x/f_z \end{pmatrix} \\ \frac{\partial \mathbf{p}_v}{\partial v}\Big|_{u,v} & = \begin{pmatrix} \partial x/\partial v \\ \partial y/\partial v \\ \partial z/\partial v \end{pmatrix} & = \begin{pmatrix} 0 \\ 1 \\ -f_y/f_z \end{pmatrix}, \end{array} \right. \qquad (36)$$

and

$$\left\{ \begin{array}{ll} \frac{\partial^2 \mathbf{p}}{\partial u^2}\Big|_{u,v} & = \begin{pmatrix} 0 & 0 & \partial^2 z/\partial u^2 \end{pmatrix}^\top \\ \frac{\partial^2 \mathbf{p}}{\partial u \partial v}\Big|_{u,v} & = \begin{pmatrix} 0 & 0 & \partial^2 z/\partial u \partial v \end{pmatrix}^\top \\ \frac{\partial^2 \mathbf{p}}{\partial v^2}\Big|_{u,v} & = \begin{pmatrix} 0 & 0 & \partial^2 z/\partial v^2 \end{pmatrix}^\top, \end{array} \right. \qquad (37)$$

where

$$\frac{\partial^2 z}{\partial u^2} = -\frac{\partial (f_x/f_z)}{\partial u} = -\frac{\frac{\partial f_x}{\partial u} f_z - \frac{\partial f_z}{\partial u} f_x}{f_z^2} =$$

$$\frac{-(f_{xx}x_u + f_{xy}y_u + f_{xz}z_u)f_z + (f_{zx}x_u + f_{zy}y_u + f_{zz}z_u)f_x}{f_z^2}.$$

Further, we have

$$\left\{\begin{array}{lcl} \frac{\partial^2 z}{\partial u^2} & = & -\frac{f_{xx}f_z^2 - 2f_{xz}f_xf_z + f_{zz}f_x^2}{f_z^3} \\ \frac{\partial^2 z}{\partial v^2} & = & -\frac{f_{yy}f_z^2 - 2f_{yz}f_yf_z + f_{zz}f_y^2}{f_z^3} \\ \frac{\partial^2 z}{\partial u \partial v} & = & -\frac{f_{xy}f_z^2 - f_{xz}f_yf_z - f_{yz}f_xf_z + f_{zz}f_xf_y}{f_z^3}. \end{array}\right. \tag{38}$$

We can derive the coefficients of the first fundamental form

$$\begin{array}{lll} E & = |\mathbf{p}_u|^2 & = 1 + f_x^2/f_z^2, \\ F & = \mathbf{p}_u \cdot \mathbf{p}_v & = f_xf_y/f_z^2, \\ G & = |\mathbf{p}_v|^2 & = 1 + f_y^2/f_z^2, \end{array} \tag{39}$$

and

$$\sqrt{EG - F^2} = \sqrt{1 + f_x^2/f_z^2 + f_y^2/f_z^2}. \tag{40}$$

Denote $D = \sqrt{f_x^2 + f_y^2 + f_z^2}$. The coefficients of the second fundamental form are:

$$\begin{array}{ll} e & = \det(\mathbf{p}_{uu}, \mathbf{p}_u, \mathbf{p}_v)/\sqrt{EG - F^2} = \\ & -(f_{xx}f_z^2 - 2f_{xz}f_xf_z + f_{zz}f_x^2)/(f_z^2 D) \\ f & = \det(\mathbf{p}_{uv}, \mathbf{p}_u, \mathbf{p}_v)/\sqrt{EG - F^2} = \\ & -(f_{xy}f_z^2 - f_{xz}f_yf_z - f_{yz}f_xf_z + f_{zz}f_xf_y)/(f_z^2 D) \\ g & = \det(\mathbf{p}_{vv}, \mathbf{p}_u, \mathbf{p}_v)/\sqrt{EG - F^2} = \\ & -(f_{yy}f_z^2 - 2f_{yz}f_yf_z + f_{zz}f_y^2)/(f_z^2 D). \end{array} \tag{41}$$

The Hessian matrix of f is

$$\mathbf{M} = \begin{pmatrix} f_{xx} & f_{xy} & f_{xz} \\ f_{xy} & f_{yy} & f_{yz} \\ f_{xz} & f_{yz} & f_{zz} \end{pmatrix}. \tag{42}$$

And its adjoint matrix $\mathbf{M}^* =$

$$\begin{pmatrix} \begin{vmatrix} f_{yy} & f_{yz} \\ f_{yz} & f_{zz} \end{vmatrix} & \begin{vmatrix} f_{yz} & f_{yx} \\ f_{zz} & f_{zx} \end{vmatrix} & \begin{vmatrix} f_{yx} & f_{yy} \\ f_{zx} & f_{zy} \end{vmatrix} \\ \begin{vmatrix} f_{zy} & f_{zz} \\ f_{xy} & f_{xz} \end{vmatrix} & \begin{vmatrix} f_{zz} & f_{zx} \\ f_{xz} & f_{xx} \end{vmatrix} & \begin{vmatrix} f_{zx} & f_{zy} \\ f_{xx} & f_{xy} \end{vmatrix} \\ \begin{vmatrix} f_{xy} & f_{xz} \\ f_{yy} & f_{yz} \end{vmatrix} & \begin{vmatrix} f_{xz} & f_{xx} \\ f_{yz} & f_{yx} \end{vmatrix} & \begin{vmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{vmatrix} \end{pmatrix}, \tag{43}$$

where $|\mathbf{X}|$ represents the determinant of $\mathbf{X}$. We can observe it is a symmetric matrix. Now we can derive

$$eg - f^2 = \frac{1}{f_z^2 D^2} \begin{pmatrix} f_x & f_y & f_z \end{pmatrix} \cdot \mathbf{M}^* \cdot \begin{pmatrix} f_x & f_y & f_z \end{pmatrix}^\top.$$

The Gaussian curvature at point $\mathbf{p}$ is

$$K(\mathbf{p}) = (eg - f^2)/(EG - F^2) = \\ \frac{1}{D^4} \begin{pmatrix} f_x & f_y & f_z \end{pmatrix} \cdot \mathbf{M}^* \cdot \begin{pmatrix} f_x & f_y & f_z \end{pmatrix}^\top. \tag{44}$$

Define $\mathbf{H} = \mathbf{M} - (f_{xx} + f_{yy} + f_{zz})\mathbf{I}$. Similarly we can get the mean curvature at point $\mathbf{p}$ as

$$H(\mathbf{p}) = \frac{1}{2}(eG - 2fF + gE)/(EG - F^2) = \\ \frac{1}{2D^3} \begin{pmatrix} f_x & f_y & f_z \end{pmatrix} \cdot \mathbf{H} \cdot \begin{pmatrix} f_x & f_y & f_z \end{pmatrix}^\top. \tag{45}$$

The normal of a surface point $\mathbf{p}$ is obviously:

$$\mathbf{n} = \frac{1}{D} \begin{pmatrix} f_x & f_y & f_z \end{pmatrix}^\top. \tag{46}$$

In summary,

$$K(\mathbf{p}) = \frac{1}{D^2} \mathbf{n}^\top \cdot \mathbf{M}^* \cdot \mathbf{n} \tag{47}$$

$$H(\mathbf{p}) = \frac{1}{2D} \mathbf{n}^\top \cdot \mathbf{H} \cdot \mathbf{n}. \tag{48}$$

References

[1] B. Gossick. *Hamilton's Principle and Physical Systems*. Academic Press, New York and London, 1967.

[2] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise smooth surface reconstruction. In *Computer Graphics Proceedings, ACM SIGGRAPH Annual Conference Series*, pages 295–302, July 1994.

[3] L. Kobbelt. A variational approach to subdivision. *Computer-Aided Geometric Design*, 13:743 – 761, 1996.

[4] L. Kobbelt. Discrete fairing. In *Proceedings of the seventh IMA Conference on the Mathematics of Surfaces'97*, pages 101–131, 1997.

[5] J.-L. Mallet. Discrete smooth interpolation. *ACM Transactions on Graphics*, 8(2):121–144, April 1989.

[6] C. Mandal, H. Qin, and B. Vemuri. Dynamic modeling of butterfly subdivision surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 6(3):265–287, July-September 2000.

[7] C. Mandal, H. Qin, and B. Vemuri. A novel FEM-based dynamic framework for subdivision surfaces. *Computer Aided Design*, 32(8&9):479–497, 2000. Special issue on solid modeling.

[8] H. Qin, C. Mandal, and B. Vemuri. Dynamic Catmull-Clark subdivision surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 4(3):215–229, July 1998.

[9] H. Qin and D. Terzopoulos. NURBS with Lagrangian dynamics (abstract). In *Third SIAM Conference on Geometric Design*, page A27, Tempe, Arizona, November 1993. SIAM.

[10] H. Qin and D. Terzopoulos. Physics-based NURBS swung surfaces. In *Proceedings of IMA Conference on Mathematics of Surfaces VI*, pages 267–290, London, UK, September 1994. Oxford University Press.

[11] T. Sederberg, J. Zheng, D. Sewell, and M. Sabin. Non-uniform recursive subdivision surfaces. In *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH*, pages 387–394, July 1998.

[12] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *Computer Graphics*, 21(4):205–214, 1987.

[13] H. Weimer and J. Warren. Subdivision schemes for thin plate splines. *Computer Graphics Forum*, 17(3):303–313, 1998. (Proceedings of EuroGraphics'98).

[14] W. Welch and A. Witkin. Free-form shape design using triangulated surfaces. In *Proceedings of SIGGRAPH 94*, Computer Graphics Proceedings, Annual Conference Series, pages 247–256, Orlando, Florida, July 1994. ACM SIGGRAPH / ACM Press. ISBN 0-89791-667-0.

[15] H. Xie and H. Qin. A novel optimization approach to the effective computation of NURBS knots. *International Journal of Shape Modeling*, 7(2):199–227, 2001.

[16] D. Zorin, P. Schröder, W. Sweldens, and et al. Subdivision for modeling and animation, course notes. *SIGGRAPH 2000*, August 2000.

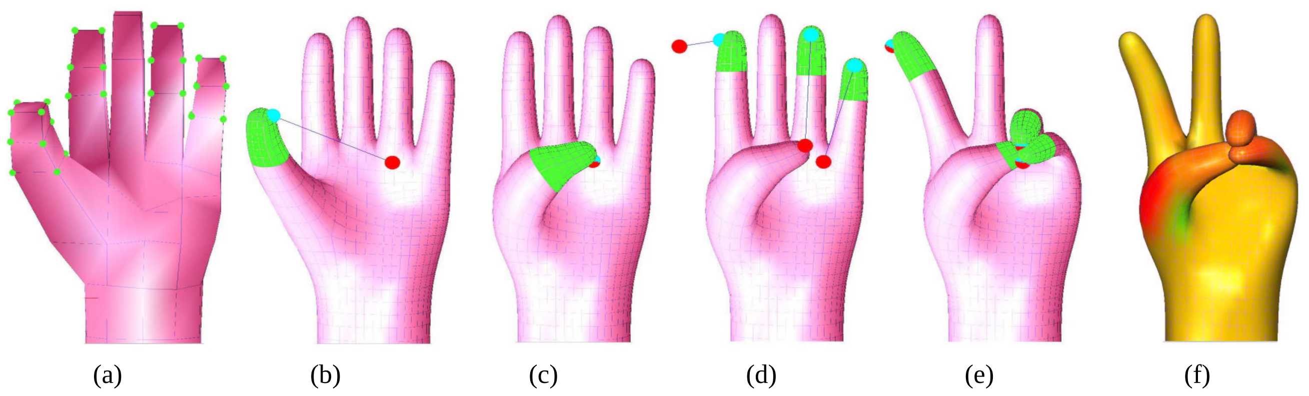

Figure 9. This figure demonstrates the manipulation of a hand through the use of springs, differential area preserving tools, and membrane energy functionals.

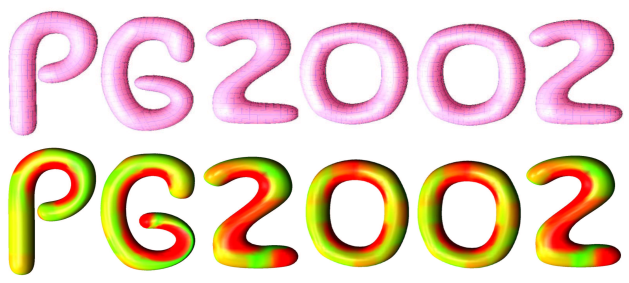

Figure 10. These characters are made by using force tools subject to differential area preservation and strain energy minimization constraints.

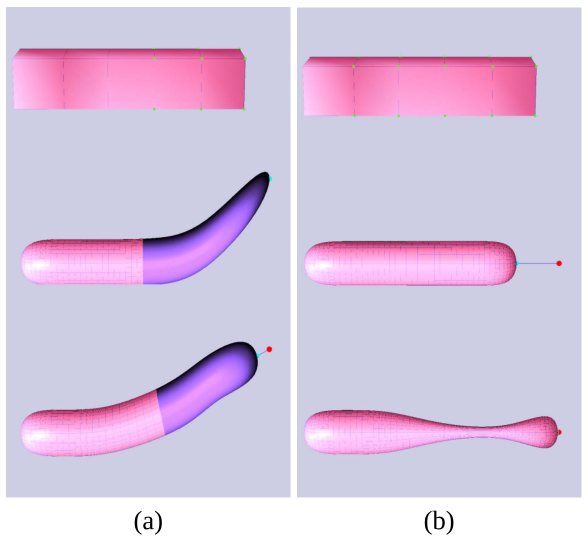

Figure 11. The material properties influence the shape evolution over time considerably.

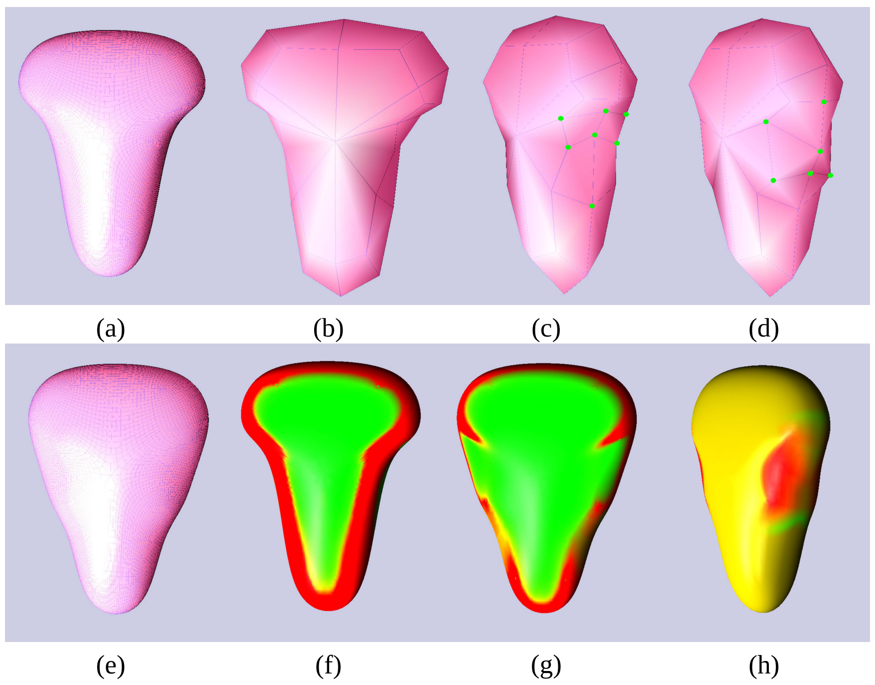

Figure 12. This figure shows the application of the total curvature minimization tool to a subdivision surface. Note that (c), (d), and (h) are rotated by about 45 degrees to give a clear side view.