ELSEVIER

Computer-Aided Design 32 (2000) 479–497

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

A novel FEM-based dynamic framework for subdivision surfaces

C. Mandal[a], H. Qin[b,*], B.C. Vemuri[c]

[a]*Sun Microsystems, Inc., 5 Omni Way, MS CHL05-105, Chelmsford, MA 01824, USA*
[b]*Department of Computer Science, State University of New York at Stony Brook, Stony Brook, NY 11794-4400, USA*
[c]*CISE Department, University of Florida, Gainesville, FL 32611, USA*

Abstract

Recursive subdivision on an initial control mesh generates a visually pleasing smooth surface in the limit. Nevertheless, users must carefully specify the initial mesh and/or painstakingly manipulate the control vertices at different levels of subdivision hierarchy to satisfy a diverse set of functional requirements and aesthetic criteria in the limit shape. This modeling drawback results from the lack of direct manipulation tools for the limit geometric shape. To improve the efficiency of interactive geometric modeling and engineering design, in this paper we integrate novel physics-based modeling techniques with powerful geometric subdivision principles, and develop a unified finite element method (FEM)-based methodology for arbitrary subdivision schemes. Strongly inspired by the recent research on Dynamic Non-Uniform Rational B-Splines (D-NURBS), we formulate and develop a dynamic framework that permits users to directly manipulate the limit surface obtained from any subdivision procedure via simulated "force" tools. The most significant contribution of our unified approach is the formulation of the limit surface of an arbitrary subdivision scheme as being composed of a single type of novel finite element. The specific geometric and dynamic features of our subdivision-based finite elements depend on the subdivision scheme used. We present our novel FEM for the modified butterfly and Catmull–Clark subdivision schemes, and generalize our dynamic framework to be applicable to other subdivision schemes. Our FEM-based approach significantly advances the state-of-the-art in physics-based geometric modeling since it provides a universal physics-based framework for any subdivision scheme. In addition, we systematically devise a mechanism that allows users to directly (not via control meshes) deform any subdivision surface; finally, we represent the limit surface of any subdivision scheme using a collection of subdivision-based novel finite elements. Our experiments demonstrate that the new unified FEM-based framework not only promises a greater potential for subdivision techniques in solid modeling, finite element analysis, and engineering design, but that it will further foster the applicability of subdivision geometry in a wide range of visual computing applications such as visualization, virtual reality, computer graphics, computer vision, robotics, and medical imaging as well.

Keywords: Physics-based modeling; Geometric modeling; Computer graphics; CAGD; Subdivision surfaces; Deformable models; Dynamics; Finite elements; Interactive techniques

1. Introduction

Efficiently modeling and intuitively manipulating complex shapes are of paramount significance to scientists and engineers in geometric and solid modeling, engineering design, manufacturing, animation and simulation, analysis and evaluation, rapid and virtual prototyping, visualization, and interaction with virtual environments. Since 1970s, tensor-product Non-Uniform Rational B-Splines (NURBS) have become an industry standard because of their many attractive properties. Nevertheless, the surface of arbitrary topology can not be represented using a single NURBS due to the global planar parameterization of NURBS. It requires that the surface be decomposed into a set of (trimmed) NURBS patches. Unfortunately, NURBS patching and trimming suffers from the following difficulties:

1. Trimming two NURBS patches to match along their common boundary involves the complex computation of surface–surface intersection (SSI), SSI algorithms generally are both computationally expensive and prone to numerical errors due to approximation; and
2. Complex and less intuitive continuity constraints across adjacent (trimmed) patches must be enforced throughout the deformation process.

In general, considerable amount of human intervention is required to guarantee that the patchwork of the underlying geometry is seamless.

Recently, subdivision geometry has emerged as a powerful geometric modeling technique which has been extensively employed to model smooth shapes of arbitrary topology in graphics, computer animation, and other

* Corresponding author. Tel.: +1-631-632-8450; fax: +1-631-632-8334.
E-mail addresses: chhandomay.mandal@sun.com (C. Mandal), qin@cs.sunysb.edu (H. Qin), vemuri@cise.ufl.edu (B.C. Vemuri).

0010-4485/00/$ - see front matter
PII: S0010-4485(00)00037-3

disciplines, mainly because of many advantages that are associated with subdivision geometry. In principle, the recursive subdivision scheme produces a visually pleasing smooth surface in the limit by repeated application of a fixed set of refinement rules on a user-specified initial control mesh. Subdivision principle, in particular, has exhibited attractive modeling potentials because of the following reasons:

- A single subdivision surface can represent shape of arbitrary topology. It requires neither trimming nor patching. Smoothness requirements can be automatically guaranteed. Subdivision naturally augments and generalizes B-splines and NURBS.
- Subdivision allows modelers to arrange control vertices in a more natural way, without the need to maintain a rectangular structure as required by NURBS. It greatly facilitates the creation of geometric features.
- Subdivision potentially allows the initial model to be refined locally. However, local refinement is not possible with NURBS, since an entire row and/or column of control points must be added to preserve the rectangular structure.

Despite the prevalence of diverse subdivision schemes in the graphics and geometric modeling literature, it is almost impossible to manipulate the limit surface (obtained through procedure-based subdivision) in a direct, natural, and intuitive way. The current state-of-the-art only permits modelers to interactively obtain the desired effects on the smooth surface by kinematically manipulating the control vertices at various levels of subdivision hierarchy. This design process is rather clumsy and laborious, in spite of the existence of many modern interactive hardware devices. Moreover, existing subdivision-based surfaces are *not yet* readily applicable for the efficient and accurate data exchange with standard formats such as B-splines and NURBS, hampering their widespread usage in solid modeling and engineering design applications. In this paper, we address the challenging problem of directly manipulating the limit subdivision surface at arbitrary locations/areas, and offer a novel solution to this problem by embedding purely geometric subdivision schemes in a physics-based modeling framework. Unlike the existing geometric solutions that only allow operations on control vertices, our methodology and algorithms permit users to physically modify the shape of subdivision surfaces at desired locations via *forces*. Consequently, this gives the user an intuitive and natural feeling that is uniquely produced while modeling with real clay/play-dough. Additionally, we will demonstrate that the proposed model can efficiently recover shapes from a cloud of 3D points.

1.1. Overview

The remainder of this paper is organized as follows. We shall briefly review the previous work on subdivision surfaces in Section 2. In Section 3, we discuss the prior work of physics-based modeling techniques and highlight the primary advantages of physics-based modeling in order to motivate our research contributions. Then, a dynamic framework for the interpolatory (modified) butterfly subdivision scheme is detailed in Section 4. We reformulate the dynamic framework for the approximating Catmull–Clark subdivision scheme using the proposed approach in Section 5. The dynamic framework for Loop's subdivision scheme is presented in Section 6. Section 7 presents a solution on how to develop a dynamic framework for any subdivision scheme. Experiments and applications are discussed in Section 8. Finally, we conclude the paper in Section 9.

2. Background

Chaikin [3] first introduced the concept of subdivision to the modeling community for generating a smooth curve from an arbitrary control polygon. Subsequently, a wide variety of subdivision schemes for modeling smooth surfaces of arbitrary topology have been derived following Chaikin's pioneering work on curve generation. The existing subdivision schemes can be broadly categorized into two distinct classes namely, (1) approximating subdivision techniques, and (2) interpolating subdivision techniques.

Among the approximating schemes, the techniques of Doo and Sabin [5] and Catmull and Clark [2] generalize the idea of obtaining uniform biquadratic and bicubic B-spline patches, respectively, from a rectangular control mesh. In Ref. [2], Catmull and Clark developed an algorithm for recursively generating a smooth surface from a polyhedral mesh of arbitrary topology. The Catmull–Clark subdivision surface, defined by an arbitrary initial mesh, can be reduced to a set of standard B-spline patches except at a finite number of degenerate points. Loop [14] presented a similar subdivision scheme based on the generalization of quartic triangular B-splines for triangular meshes. Hoppe et al. [10] further extended Loop's work to produce piecewise smooth surfaces with selected discontinuities. Halstead et al. [9] proposed an algorithm to construct a Catmull–Clark subdivision surface that interpolates the vertex mesh of arbitrary topology. Peters and Reif [18] proposed a simple subdivision scheme for smoothing polyhedra. Most recently, non-uniform Doo–Sabin and Catmull–Clark surfaces that generalize non-uniform tensor-product B-spline surfaces to arbitrary topologies were introduced by Sederberg et al. [24]. All the aforementioned schemes generalize recursive subdivision schemes for generating limit surfaces with a known parameterization. Various issues involved with the use of these approximating subdivision schemes for character animation were discussed at length by DeRose et al. [4].

The most well-known interpolation-based subdivision scheme is the "butterfly" algorithm proposed by Dyn et al. [7]. Butterfly method, like other subdivision schemes,

makes use of a small number of neighboring vertices for subdivision. It requires simple data structures and is rather straightforward to implement. Nevertheless, it needs a topologically regular setting of the initial (control) mesh in order to obtain a smooth C^1 limit surface. Zorin et al. [28] has developed an improved interpolatory subdivision scheme (which we call the *modified* butterfly scheme) that retains the simplicity of the butterfly scheme and results in much smoother surfaces even from irregular initial meshes. These interpolatory subdivision schemes have extensive applications in wavelets on manifolds, multiresolution editing, etc.

A variational approach for interpolatory refinement has been proposed by Kobbelt [11,12] and by Kobbelt and Schröder [13]. In this approach, the vertex positions in the refined mesh at each subdivision step are obtained by solving an optimization problem. Therefore, these schemes are global, i.e. every new vertex position depends on all the vertex positions of the coarser level mesh. The local refinement property which makes the subdivision schemes attractive for implementation in the graphics applications is not retained in the variational approach.

The derivation of various mathematical properties of the limit surface generated by the subdivision algorithms is rather complex. Doo and Sabin [6] first analyzed the smoothness behavior of the limit surface using the Fourier transform and an eigen-analysis of the subdivision matrix. Ball and Storry [1] and Reif [22] further extended Doo and Sabin's prior work on continuity properties of subdivision surfaces by deriving various necessary and sufficient conditions on smoothness for different subdivision schemes. Specific subdivision schemes were also analyzed by several other researchers [8,19,23,29]. Most recently, Stam [25,26] developed an exact point evaluation algorithm for both Catmull–Clark subdivision scheme and Loop's triangular subdivision scheme.

3. Motivation

Subdivision geometry has offered users extraordinary power and flexibility especially when used for modeling complex shapes of arbitrary topology. Nevertheless, it constitutes a purely geometric representation, and hence does not exploit the full potential of the underlying geometric formulation owing to the following reasons:

- Modelers are faced with the tedium of indirect shape refinement through time-consuming operations on a large number of topologically irregular control vertices and less intuitive modification on various subdivision rules. This process is clumsy and laborious especially for effectively representing and deforming highly complicated objects.
- Control point manipulation is not natural due to the fact that control points generally do not reside on the sculpted objects, hence, it often requires designers to make many non-intuitive decisions, and it is even more difficult to accurately quantify the refinement effect at arbitrary localized regions. Despite the advent of many modern 3D graphics interaction tools, these indirect geometric operations remain non-intuitive and laborious in general.
- Oftentimes it may not be enough to obtain the most "fair" surface that interpolates a set of (ordered or unorganized) data points. Typical design requirements may be posed in both quantitative and qualitative terms. For example, a certain number of local features such as bulges or inflections may be strongly desired while requiring geometric objects to satisfy global smoothness criteria in solid modeling and/or interactive graphics applications. Therefore, it can be very frustrating to enforce a diverse set of heterogeneous criteria simultaneously via the indirect approach.

In contrast, physics-based modeling can provide a superior approach to shape modeling that can overcome most of the limitations associated with traditional geometric modeling approaches. Free-form deformable models governed by the laws of continuum mechanics are of particular relevance in this context. Physics-based design augments (rather than supersedes) standard geometric design, offering attractive new advantages:

- Dynamic models respond to simulated force in a natural and predictable way. The dynamic formulation marries the model geometry with time, mass, damping, and constraints via a force balance equation. Dynamics facilitates interaction, especially direct manipulation and interactive sculpting of complex geometric models for real-time shape variation.
- Geometric design is a time-varying process because designers are often interested in not only the final static equilibrium shape but the intermediate shape variation as well. Dynamic models produce smooth, natural motions that are familiar and can be easily controlled.
- The equilibrium shape of a geometric object is characterized by a minimum of its potential energy, subject to imposed constraints. It is possible to formulate potential energy functionals that satisfy local and global design criteria. In particular, the elastic energy functionals will allow the imposition of global qualitative "fairness" criteria through quantitative means.
- Physics-based shape design can free designers from having to make non-intuitive decisions, such as moving control points to prescribed locations. In addition, non-expert users are able to concentrate on visual shape variation without necessarily comprehending the underlying mathematical formulation.
- Physics-based modeling techniques and real-time dynamics integrate geometry with physics in a natural and coherent way. The unified formulation is potentially relevant throughout the entire modeling, simulation, analysis, and manufacturing process. More importantly, it is potentially possible to introduce manufacturing constraints in the earlier design stage.

The dynamic approach subsumes all of the aforementioned modeling capabilities in a formulation which grounds everything in real-world physical behavior.

Free-form deformable models were first introduced to the modeling community by Terzopoulos et al. [27], and were refined by a number of researchers over the years. Qin and Terzopoulos [20] developed D-NURBS which are very sophisticated physics-based models suitable for representing a wide variety of free-form as well as standard analytic shapes. The D-NURBS have the advantage of interactive and direct manipulation of NURBS curves and surfaces, resulting in physically meaningful thus intuitively predictable motion and shape variation. However, a severe limitation of the existing deformable models, including D-NURBS, is that they are defined on a rectangular parametric domain. Therefore, it can be very difficult to model surfaces of arbitrary genus using these models. Subdivision schemes, in contrast, can model complex surfaces of arbitrary topology, and hence are a good candidate for incorporation of physics-based principles where by the modeler can directly manipulate the (complicated) limit surface in an intuitive way.

Previously we had introduced dynamic Catmull–Clark subdivision surfaces [15,16,21] where the smooth limit surface generated by the Catmull–Clark subdivision scheme was embedded in a physics-based modeling framework. The current research differs significantly from our prior work because the new approach taken in this paper is much more general. It aims to develop a systematic and universal mechanism with which any subdivision scheme can be formulated within the physics-based framework. The primary mathematical technique we resort to is finite element analysis. We shall first formulate a dynamic representation and equation for an interpolatory subdivision scheme—the modified butterfly subdivision method—where the limit surface, unlike other generalized spline-based subdivision schemes, does not have any closed-form analytic formulation [17]. Moreover, we shall reformulate the dynamic Catmull–Clark subdivision surface model using this novel methodology, and describe how to develop a unified dynamic framework for any subdivision scheme. The key contribution of this unified approach is to represent the smooth limit surface of any subdivision scheme using a collection of a single type of novel finite elements. The geometric and physical features of our subdivision-based finite elements depend only on the subdivision scheme involved. Our finite element method (FEM)-based approach significantly advances the state-of-the-art in physics-based geometric modeling because of the following.

1. It provides a universal physics-based solution to any subdivision scheme beyond prevalent spline-like subdivision techniques.
2. A natural mechanism that allows users to intuitively deform any subdivision surface has been systematically devised.
3. The limit surface of any subdivision schemes has been represented using a single type of novel subdivision-based finite elements.
4. Our subdivision-based finite elements are potentially of great interest to FEM communities.

4. Dynamic butterfly subdivision surfaces

This section discusses a dynamic framework for an interpolatory subdivision scheme namely, the (modified) butterfly subdivision technique. First, a brief overview of the (modified) butterfly subdivision scheme is presented. Next, a local geometric parameterization technique for the limit surface of the (modified) butterfly subdivision is detailed. Our parameterization method is then used to derive the new triangular FEM for the *butterfly-based* subdivision scheme. Finally, the implementation details are described. Note that, we will further generalize our physics-based formulation for other interpolatory subdivision schemes in Section 7.

4.1. The (modified) butterfly subdivision

The butterfly subdivision scheme [7] starts with an initial triangular mesh (a.k.a. the control mesh) defined by a set of control vertices. In each step of subdivision, the initial (control) mesh is refined through the transformation of each triangular face into a patch with four *smaller* triangular faces. After one step of refinement, the new mesh in the *finer* level retains the vertices of each triangular face in the previous level and hence, interpolates the *coarser* mesh in the previous level. In addition, every edge in each triangular face is split by adding a new vertex whose position is obtained by an affine combination of the neighboring vertex positions in the coarser level. For instance, the mesh in Fig. 1(b) is obtained by subdividing the initial mesh shown in Fig. 1(a) once. Note that, all the newly introduced vertices corresponding to the edges in the original mesh have degree six, whereas the position and degree of all original vertices do not change in the refined mesh.

In the original butterfly scheme, the new vertices corresponding to the edges in the previous level are obtained using an eight-point stencil. It produces a smooth C^1 surface in the limit except at the *extraordinary* points corresponding to the *extraordinary* vertices (vertices with degree not equal to six) in the initial mesh [28]. Since all the vertices introduced through subdivision have degree six, the number of extraordinary points in the smooth limit surface equals to the number of extraordinary vertices in the initial mesh. Recently, the *original* butterfly scheme has been modified by Zorin et al. [28] to obtain better smoothness properties at the extraordinary points. In this *modified* butterfly subdivision

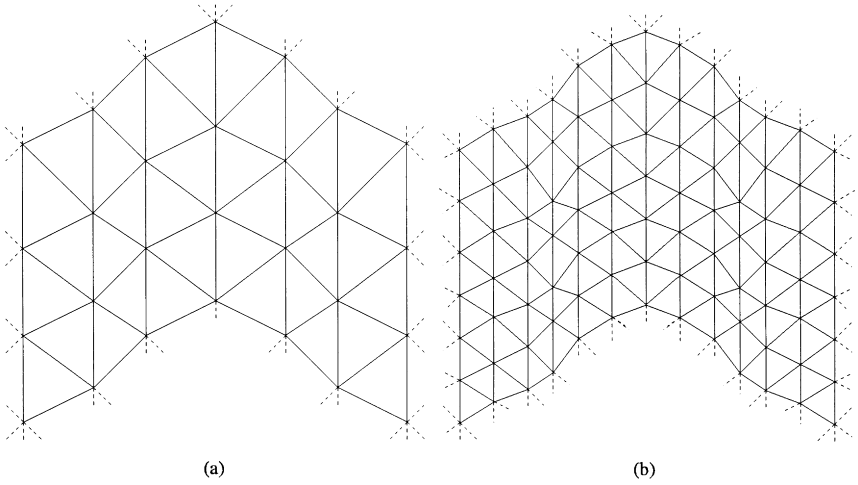


Fig. 1. (a) The control polygon with triangular faces. (b) The refined mesh obtained after one subdivision step using butterfly subdivision rules.

technique, all the edges had been categorized into three classes:

1. Edges connecting two vertices of degree six (a 10 point stencil, as shown in Fig. 2(a), is used to obtain the new vertex positions corresponding to these edges).
2. Edges connecting a vertex of degree six and a vertex of degree $n \neq 6$ (the corresponding stencil to obtain new vertex position is shown in Fig. 2(b), where $q = 0.75$ is the weight associated with the vertex of degree $n \neq 6$, and $s_i = (0.25 + \cos(2\pi i/n) + 0.5\cos(4\pi i/n))/n$, $i = 0, 1, \ldots, n-1$, are the weights associated with the vertices of degree six).
3. Edges connecting two vertices of degree $n \neq 6$.

The last case cannot occur except in the initial mesh as the newly introduced vertices are of degree six, and the new vertex position in this last case is obtained by averaging the positions obtained through the use of stencil shown in Fig. 2(b) at each of those two extraordinary vertices.

4.2. Formulation

This section systematically formulates the dynamic framework for the modified butterfly subdivision scheme. Unlike the approximating schemes, the geometry of the limit surface obtained via modified butterfly subdivision does not have any closed-form analytic expression even for a regular mesh. Therefore, the key issue is to define an appropriate parametric domain and derive a local parameterization for *butterfly-based* subdivision. These relevant geometric components are critical to the development of our physics-based FEM for the limit surface of butterfly scheme.

The smooth limit surface defined by the modified butterfly subdivision technique is of arbitrary topology where a global parameterization is impossible. Nevertheless, the

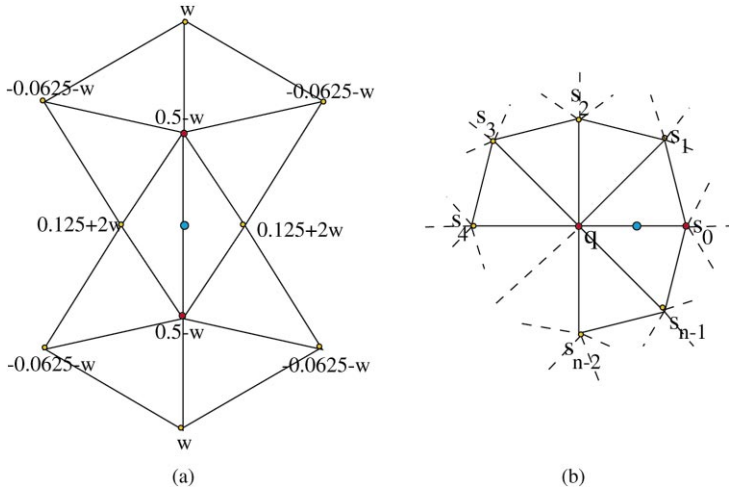


Fig. 2. (a) The weighing factors of contributing vertex positions for an edge connecting two vertices of degree 6. (b) The corresponding case when one vertex is of degree n and the other is of degree 6.

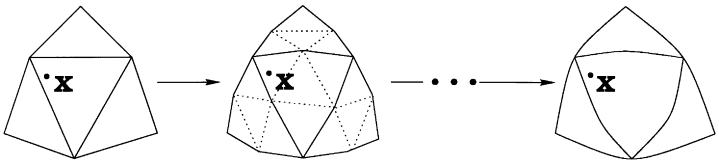


Fig. 3. The smoothing effect of the subdivision process on the triangles of the initial mesh.

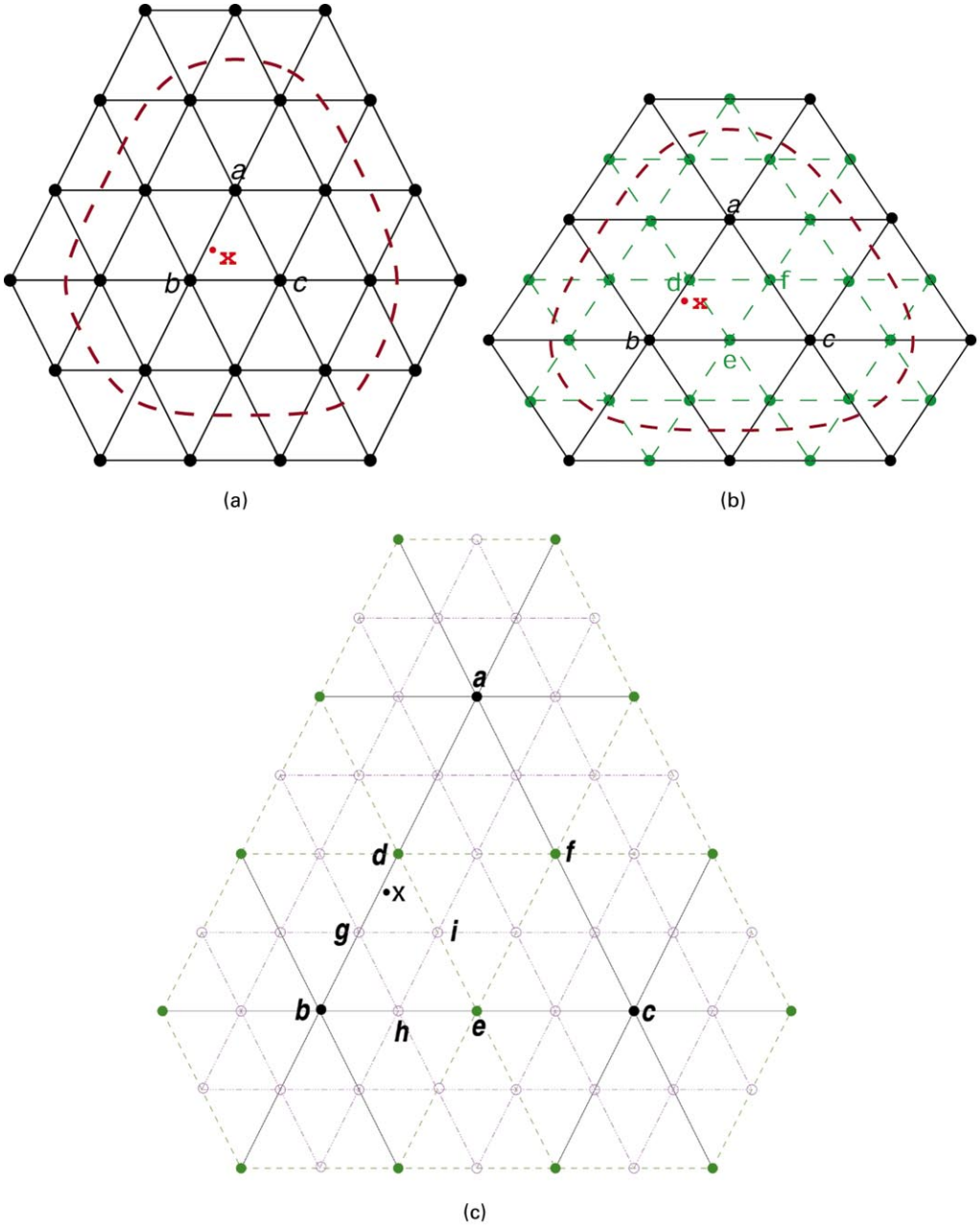


Fig. 4. Tracking a point **x** through various levels of subdivision: (a) initial mesh; (b) the selected section (enclosed by dotted lines) of the mesh in (a), after one subdivision step; (c) the selected section of the mesh in (b), after another subdivision step.

limit surface can be locally parameterized over the geometric domain defined by the initial mesh. The idea is to track an arbitrary point on the initial mesh across the mesh hierarchy obtained via the subdivision process (see Figs. 3 and 4), so that a correspondence can be established between the point being tracked in the initial mesh and its image on the limit surface.

The modified butterfly subdivision scheme starts with an initial set of triangular faces. The recursive application of the subdivision rules smoothes out each triangular face, and in the limit, we obtain a smooth surface consisting of a collection of smooth triangular patches. The subdivision process and the triangular decomposition of the limit surface is depicted in Fig. 3. Note that, the limit surface can be represented by the same number of smooth triangular patches as that of the triangular faces in the initial mesh. Therefore, the limit surface **s** can be expressed as

$$\mathbf{s} = \sum_{k=1}^{n} \mathbf{s}_k, \tag{1}$$

where n is the number of triangular faces in the initial mesh and $\mathbf{s}_k$ the smooth triangular patch in the limit surface corresponding to the kth triangular face in the initial mesh.

We now describe the parameterization of the limit surface over the initial mesh. The procedure can be best explained through the following example. A simple planar mesh shown in Fig. 4(a) is chosen as the initial mesh. An arbitrary point x inside the triangular face *abc* is tracked over the meshes obtained through subdivision. The vertices in the initial mesh are darkly shaded in Fig. 4. After one step of subdivision, the initial mesh is refined by addition of new vertices, which are lightly shaded. Another subdivision step on this refined mesh leads to a finer mesh with introduction of new vertices which are unshaded. It may be noted that *any point inside the smooth triangular patch in the limit surface corresponding to the face abc in the initial mesh depends only on the vertices in the initial mesh which are within the 2-neighborhood of the vertices* **a**, **b** *and* **c** *due to the local nature of the subdivision process* (the k-neighborhood of a vertex includes all the vertices that can be reached following at most k edges from the given vertex). For example, the vertex **d**, introduced after first subdivision step, can be obtained using the 10 point stencil shown in Fig. 2(a) on the edge *ab*. All the contributing vertices in the initial mesh are within the 1-neighborhood of the vertices **a** and **b**. A 10 point stencil can be used again in the next subdivision step on the edge *db* to obtain the vertex **g**. Some of the contributing vertices at this level of subdivision, for example, the (lightly shaded) 1-neighbors of the vertex **b** (except **d** and **e**) in Fig. 4(b), depend on some vertices in the initial mesh which are within the 2-neighborhood of the vertices **a**, **b** and **c** in the initial mesh.

In the rest of the formulation, superscripts are used to indicate the subdivision level. For example, $\mathbf{v}^j_{uvw}$ denotes the collection of vertices at level j which control the smooth patch in the limit surface corresponding to the triangular face *uvw* at the jth level of subdivision. Let $\mathbf{v}^0_{abc}$ be the collection of vertices in the initial mesh that are within the 2-neighbourhood of the vertices **a**, **b** and **c** (marked black in Fig. 4(a)). Let the number of such vertices be r. Then, the vector $\mathbf{v}^0_{abc}$, which is concatenation of the (x, y, z) positions for all the r vertices, is of dimension $3r$. Based on the above observation of the 2-*neighborhood* property, the geometry of the smooth triangular patch in the limit surface corresponding to the triangular face *abc* in the initial mesh is uniquely determined by these r vertices. Because of the recursive characteristic, there now exists four subdivision matrices $(\mathbf{A}_{abc})_{\mathrm{t}}$, $(\mathbf{A}_{abc})_{\mathrm{l}}$ $(\mathbf{A}_{abc})_{\mathrm{r}}$ and $(\mathbf{A}_{abc})_{\mathrm{m}}$ of dimension $(3r,3r)$ such that

$$\begin{aligned} \mathbf{v}^1_{adf} &= (\mathbf{A}_{abc})_{\mathrm{t}}\mathbf{v}^0_{abc}, \qquad \mathbf{v}^1_{bed} = (\mathbf{A}_{abc})_{\mathrm{l}}\mathbf{v}^0_{abc}, \\ \mathbf{v}^1_{cfe} &= (\mathbf{A}_{abc})_{\mathrm{r}}\mathbf{v}^0_{abc}, \qquad \mathbf{v}^1_{def} = (\mathbf{A}_{abc})_{\mathrm{m}}\mathbf{v}^0_{abc}, \end{aligned} \tag{2}$$

where the subscripts t, l, r and m denote tope, left, right and middle triangle positions, respectively (indicating the relative position of the new triangle with respect to the *original* triangle), and $\mathbf{v}^1_{adf}$, $\mathbf{v}^1_{bed}$, $\mathbf{v}^1_{cfe}$ and $\mathbf{v}^1_{def}$ are the concatenation of the (x, y, z) positions for the vertices in the 2-neighborhood of the corresponding triangle within the newly obtained refined mesh after one subdivision. Note that, the new vertices in this level of subdivision are lightly shaded in Fig. 4(b). The 2-neighborhood configuration of the vertices in the newly obtained triangles is exactly the same as that of the original triangle, hence local subdivision matrices are square and the vector dimensions on both sides of Eq. (2) are the same.

Carrying out one more level of subdivision, a new set of vertices which are unshaded in Fig. 4(c) are obtained along with the old vertices. Adopting a similar approach as in the derivation of Eq. (2), it can be shown that

$$\begin{aligned} \mathbf{v}^2_{dgi} &= (\mathbf{A}_{bed})_{\mathrm{t}}\mathbf{v}^1_{bed}, \qquad \mathbf{v}^2_{bhg} = (\mathbf{A}_{bed})_{\mathrm{l}}\mathbf{v}^1_{bed}, \\ \mathbf{v}^2_{eih} &= (\mathbf{A}_{bed})_{\mathrm{r}}\mathbf{v}^1_{bed}, \qquad \mathbf{v}^2_{ghi} = (\mathbf{A}_{bed})_{\mathrm{m}}\mathbf{v}^1_{bed}. \end{aligned} \tag{3}$$

The relative position and geometric structure face *dgi* in Fig. 4(c) with respect to the triangular face *bed* is topologically the same as of the triangular face *adf* in Fig. 4(b) with respect to the triangular face *abc*. Therefore, we can obtain $(\mathbf{A}_{bed})_{\mathrm{t}} = (\mathbf{A}_{abc})_{\mathrm{t}}$. Based on the similar reasoning, Eq. (3) can be rewritten as

$$\begin{aligned} \mathbf{v}^2_{dgi} &= (\mathbf{A}_{bed})_{\mathrm{t}}\mathbf{v}^1_{bed} = (\mathbf{A}_{abc})_{\mathrm{t}}\mathbf{v}^1_{bed}, \\ \mathbf{v}^2_{bhg} &= (\mathbf{A}_{bed})_{\mathrm{l}}\mathbf{v}^1_{bed} = (\mathbf{A}_{abc})_{\mathrm{l}}\mathbf{v}^1_{bed}, \\ \mathbf{v}^2_{eih} &= (\mathbf{A}_{bed})_{\mathrm{r}}\mathbf{v}^1_{bed} = (\mathbf{A}_{abc})_{\mathrm{r}}\mathbf{v}^1_{bed} \\ \mathbf{v}^2_{ghi} &= (\mathbf{A}_{bed})_{\mathrm{m}}\mathbf{v}^1_{bed} = (\mathbf{A}_{abc})_{\mathrm{m}}\mathbf{v}^1_{bed}. \end{aligned} \tag{4}$$

Combining Eqs. (2) and (4), it can be shown that

$$\begin{aligned} \mathbf{v}^2_{dgi} &= (\mathbf{A}_{abc})_{\mathrm{t}}(\mathbf{A}_{abc})_{\mathrm{l}}\mathbf{v}^0_{abc}, \qquad \mathbf{v}^2_{bhg} = (\mathbf{A}_{abc})_{\mathrm{l}}(\mathbf{A}_{abc})_{\mathrm{l}}\mathbf{v}^0_{abc}, \\ \mathbf{v}^2_{eih} &= (\mathbf{A}_{abc})_{\mathrm{r}}(\mathbf{A}_{abc})_{\mathrm{l}}\mathbf{v}^0_{abc}, \qquad \mathbf{v}^2_{ghi} = (\mathbf{A}_{abc})_{\mathrm{m}}(\mathbf{A}_{abc})_{\mathrm{l}}\mathbf{v}^0_{abc}. \end{aligned} \tag{5}$$

Let $\mathbf{x}$ be a point with barycentric coordinates $(\alpha^0_{abc}, \beta^0_{abc}, \gamma^0_{abc})$ inside the triangular face *abc*. When the initial mesh is refined, $\mathbf{x}$ becomes a point inside the triangular face *bed* with barycentric coordinates $(\alpha^1_{bed}, \beta^1_{bed}, \gamma^1_{bed})$. Another level of subdivision causes $\mathbf{x}$ to be included in the triangular face *dgi* barycentric coordinates $(\alpha^2_{dgi}, \beta^2_{dgi}, \gamma^2_{dgi})$. Let $\mathbf{s}^j_{abc}$ denote the *j*th level approximation of the smooth triangular patch $\mathbf{s}_{abc}$ in the limit surface corresponding to the triangular face *abc* in the initial mesh. Now $\mathbf{v}^0_{abc}$ can be written as

$$\mathbf{v}^0_{abc} = [\overbrace{a_x, b_x, c_x, \ldots,}^{r}\ \overbrace{a_y, b_y, c_y, \ldots,}^{r}\ \overbrace{a_z, b_z, c_z, \ldots}^{r}\ ]^{\mathrm{T}},$$

where the subscripts *x*, *y* and *z* indicate the *x*, *y* and *z* coordinates of the corresponding vertex position, respectively. The expressions for $\mathbf{v}^1_{bed}$ and $\mathbf{v}^0_{dgi}$ can also be written in a similar manner. Next, the matrix $\mathbf{B}^0_{abc}$ can be constructed as follows:

$$\mathbf{B}^0_{abc}(\mathrm{x}) = \begin{bmatrix} \overbrace{\alpha^0_{abc}, \beta^0_{abc}, \gamma^0_{abc}, 0, \ldots, 0,}^{r}\ \overbrace{0, \ldots, 0,}^{r}\ \overbrace{0, \ldots, 0}^{r} \\ \overbrace{0, \ldots, 0,}^{r}\ \overbrace{\alpha^0_{abc}, \beta^0_{abc}, \gamma^0_{abc}, 0, \ldots, 0,}^{r}\ \overbrace{0, \ldots, 0}^{r} \\ \overbrace{0, \ldots, 0,}^{r}\ \overbrace{0, \ldots, 0,}^{r}\ \overbrace{\alpha^0_{abc}, \beta^0_{abc}, \gamma^0_{abc}, 0, \ldots, 0}^{r} \end{bmatrix}.$$

The matrices $\mathbf{B}^1_{bed}$ and $\mathbf{B}^2_{dgi}$ can also be constructed in a similar fashion. Now $\mathbf{s}^0_{abc}(\mathbf{x})$, $\mathbf{s}^1_{abc}(\mathbf{x})$, and $\mathbf{s}^2_{abc}(\mathbf{x})$ can be written as

$$\begin{aligned} \mathbf{s}^0_{abc}(\mathbf{x}) &= \mathbf{B}^0_{abc}(\mathbf{x})\mathbf{v}^0_{abc}, \\ \mathbf{s}^1_{abc}(\mathbf{x}) &= \mathbf{B}^1_{bed}(\mathbf{x})\mathbf{v}^1_{bed} = \mathbf{B}^1_{bed}(\mathbf{x})(\mathbf{A}_{abc})_{\mathrm{l}}\mathbf{v}^0_{abc}, \\ \mathbf{s}^2_{abc}(\mathbf{x}) &= \mathbf{B}^2_{dgi}(\mathbf{x})\mathbf{v}^2_{dgi} = \mathbf{B}^2_{dgi}(\mathbf{x})(\mathbf{A}_{abc})_{\mathrm{t}}\mathbf{v}^1_{bed} \\ &= \mathbf{B}^2_{dgi}(\mathbf{x})(\mathbf{A}_{abc})_{\mathrm{t}}(\mathbf{A}_{abc})_{\mathrm{l}}\mathbf{v}^0_{abc}. \end{aligned} \tag{6}$$

Proceeding in a similar way, the expression for $\mathbf{s}^j_{abc}(\mathbf{x})$, *j*th level approximation of $\mathbf{s}_{abc}(\mathbf{x})$, is given by

$$\begin{aligned} \mathbf{s}^j_{abc}(\mathbf{x}) &= \mathbf{B}^j_{uvw}(\mathbf{x})\overbrace{(\mathbf{A}_{abc})_{\mathrm{m}}\ldots(\mathbf{A}_{abc})_{\mathrm{t}}(\mathbf{A}_{abc})_{\mathrm{l}}}^{j}\ \mathbf{v}^0_{abc} \\ &= \mathbf{B}^j_{uvw}(\mathbf{x})(\mathbf{A}^j_{abc})\mathbf{v}^0_{abc} = \mathbf{B}^j_{abc}(\mathbf{x})\mathbf{v}^0_{abc}, \end{aligned} \tag{7}$$

where $\mathbf{x}$ is inside the triangular face *uvw* at level *j* (with an assumption that *uvw* is the triangular face in the *middle* with respect to its *coarser-level* original triangular face in the *previous* level), $(\mathbf{A}^j_{abc}) = (\mathbf{A}_{abc})_{\mathrm{m}}\ldots(\mathbf{A}_{abc})_{\mathrm{t}}(\mathbf{A}_{abc})_{\mathrm{l}}$ and $\mathbf{B}^j_{abc}(\mathbf{x}) = \mathbf{B}^j_{uvw}(\mathbf{x})(\mathbf{A}^j_{abc})$. It may be noted that the sequence of applying $(\mathbf{A}_{abc})_{\mathrm{t}}$, $(\mathbf{A}_{abc})_{\mathrm{l}}$, $(\mathbf{A}_{abc})_{\mathrm{r}}$ and $(\mathbf{A}_{abc})_{\mathrm{m}}$ depends on the triangle inside which the tracked point *x* falls after each subdivision step. Finally, the local geometric parameterization procedure can be completed by writing

$$\mathbf{s}_{abc}(\mathbf{x}) = (\lim_{j\to\infty} \mathbf{B}^j_{abc}(\mathbf{x}))\mathbf{v}^0_{abc} = \mathbf{B}_{abc}(\mathbf{x})\mathbf{v}^0_{abc}. \tag{8}$$

Note that, $\mathbf{B}_{abc}$ is the collection of basis functions at the vertices $\mathbf{v}^0_{abc}$. It may also be noted that the modified butterfly subdivision scheme is a *stationary* subdivision process, and hence new vertex positions are obtained by affine combinations of nearby vertices. This guarantees that each row of the matrices $(\mathbf{A}_{abc})_{\mathrm{t}}$, $(\mathbf{A}_{abc})_{\mathrm{l}}$, $(\mathbf{A}_{abc})_{\mathrm{r}}$ and $(\mathbf{A}_{abc})_{\mathrm{m}}$ sums to one. The largest eigenvalue of such matrices is 1 and therefore the mathematical limit in Eq. (8) exists. Now, assuming the triangular face *abc* is the *k*th face in the initial mesh, Eq. (8) can be written as

$$\mathbf{s}_k(\mathbf{x}) = \mathbf{B}_k(\mathbf{x})\mathbf{v}^0_k = \mathbf{B}_k(x)\mathbf{A}_k\mathbf{p}, \tag{9}$$

where $\mathbf{p}$ is the concatenation of the (x, y, z) positions of all vertices in the initial mesh and the matrix $\mathbf{A}_k$, when post-multiplied by $\mathbf{p}$, only selects the vertices $\mathbf{v}^0_k$ defining the *k*th smooth triangular patch in the limit surface. If there are *t* vertices in the initial mesh and *r* of them control the *k*th patch, then $\mathbf{p}$ is a vector of dimension 3*t*, $\mathbf{A}_k$ is a matrix of dimension (3*r*, 3*t*) and $\mathbf{B}_k(\mathrm{x})$ is a matrix of dimension (3,3*r*).

Combining Eqs. (1) and (9), it can be shown that

$$\mathbf{s}(\mathbf{x}) = \left(\sum_{k=1}^{\mathrm{n}} \mathbf{B}_k(\mathbf{x})\mathbf{A}_k\right)\mathbf{p} = \mathbf{J}(\mathbf{x})\mathbf{p}, \tag{10}$$

where $\mathbf{J}$, a matrix of dimension (3, 3*t*), is the collection of basis functions for the corresponding vertices in the initial mesh. The vector $\mathbf{p}$ is also known as the degrees of freedom vector of the smooth limit surface $\mathbf{s}$.

We now treat the vertex positions in the initial mesh defining the smooth limit surface $\mathbf{s}$ as time variables in order to develop the new dynamic butterfly subdivision model. The velocity of the surface model can be expressed as $\dot{\mathbf{s}}(\mathbf{x},\mathbf{p}) = \mathbf{J}(\mathbf{x})\dot{\mathbf{p}}$, where an overstruck dot denotes a time derivative and $\mathbf{x} \in S^0$, S^0 being the domain defined by the initial mesh. Note that, S^0 is the parametric domain of the limit surface, each triangle of the initial control mesh serves as a local parametric domain for its corresponding triangular patch.

4.3. Finite element procedure

In Section 4.2, we have demonstrated that the smooth limit surface of butterfly subdivision can be represented by a collection of smooth triangular patches. In our dynamic framework, we now consider each patch of the limit surface as a finite element. The number of such patches is equal to the number of triangular faces in the initial mesh as mentioned earlier. The concept of decomposing the smooth limit surface into a collection of elements is illustrated in Fig. 5. We also show the parametric domain and control vertices for shaded elements in Fig. 5. The governing

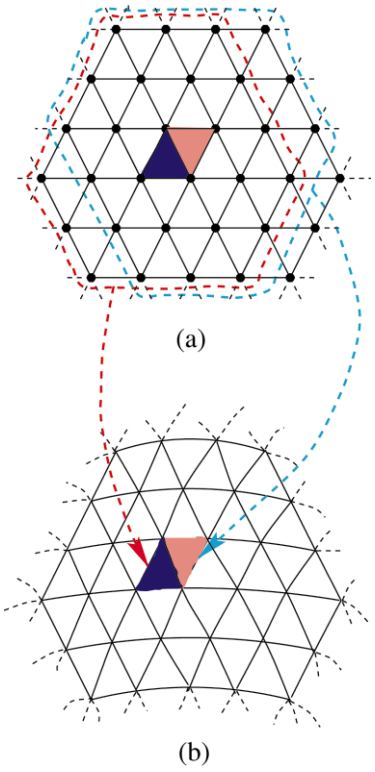


Fig. 5. (a) An initial mesh, and (b) the corresponding limit surface. The domains of the shaded elements in the limit surface are the corresponding triangular faces in the initial mesh. The encircled vertices in (a) are the degrees of freedom for the corresponding element.

motion equation of this subdivision-based FEM model is given by

$$\mathbf{M}\ddot{\mathbf{p}} + \mathbf{D}\dot{\mathbf{p}} + \mathbf{K}\mathbf{p} = \mathbf{f}_{\mathbf{p}}, \tag{11}$$

where $\mathbf{f}_{\mathbf{p}}$ is the generalized force vector, and **M**, **D**, and **K** are the mass, damping and stiffness matrices of the physical model. In the interest of space, we provide an outline on how to derive the mass, damping and stiffness matrices for these finite elements so that a numerical solution to the governing second-order differential equation can be obtained using popular finite element analysis techniques. We use the same example as in Section 4.2 (refer to Fig. 4) to introduce the relevant concepts and derive our FEM model.

The mass matrix for the element $\mathbf{s}_{abc}$, corresponding to the triangular face *abc*, can be expressed as

$$\mathbf{M}_{abc} = \int_{\mathbf{x}\in\mathbf{s}_{abc}} \mu(\mathbf{x})\mathbf{B}_{abc}^{\mathrm{T}}(\mathbf{x})\mathbf{B}_{abc}(\mathbf{x})\,\mathrm{d}\mathbf{x}. \tag{12}$$

However, the basis functions (stored as entries in $\mathbf{B}_{abc}$) do not have any analytic form, hence computing this integral is a difficult proposition. We solve this problem by approximating the smooth triangular patch in the limit surface corresponding to the face *abc* in the initial mesh by a triangular mesh with 4^j faces obtained after *j* levels of subdivision of the original triangular face *abc* (each subdivision step splits one triangular face into 4 triangular faces). In addition, we choose a discretized form of mass distribution function which has non-zero values at the vertex positions of the *j*th subdivision level mesh to simplify the implementation matter. Then the mass matrix can be approximated as

$$\mathbf{M}_{abc} = \sum_{i=1}^{k} \mu(\mathbf{v}_i^j)\{\mathbf{B}_{abc}^i(\mathbf{v}_i^j)\}^{\mathrm{T}}\{\mathbf{B}_{abc}^j(\mathbf{v}_i^j)\}, \tag{13}$$

where *k* is the number of vertices in the triangular mesh with 4^j faces. This approximation has been found to be very effective and efficient for the implementation of FEM procedure. The computation of elemental damping matrix follows suit.

Physics-based models have both kinetic and potential energies. We now define the internal (e.g. elastic) energy of the subdivision-based dynamic model by assigning deformation energy to each element. We take a similar approach as shown above and consider the *j*th level approximation of the element. Throughout this paper, in particular, we assign spring-like energy to the approximated of the element. Throughout this paper, in particular, we assign spring-like energy to the approximated model because of its simplicity and efficient computation. The energy at the *j*th level of approximation can be defined as

$$E_{abc} \approx E_{abc}^j = \frac{1}{2}\sum_{\Omega} \frac{k_{lm}(|\mathbf{v}_l^j - \mathbf{v}_m^j| - \mathscr{L}_{lm})^2}{|\mathbf{v}_l^j - \mathbf{v}_m^j|^2}(\mathbf{v}_l^j - \mathbf{v}_m^j)^2, \tag{14}$$

where k_{lm} is the spring-controlling variable, $\mathbf{v}_l^j$ and $\mathbf{v}_m^j$, the *l*th and *m*th vertex in the *j*th level mesh, are in the 1-neighborhood of each other, Ω is the domain defined by all such vertex pairs, $\mathscr{L}_{lm}$ is the natural length of the spring connected between $\mathbf{v}_l^j$ and $\mathbf{v}_m^j$. Let $\mathbf{v}_{abc}^j$ be the concatenation of the (x, y, z) positions of all the vertices in the *j*th subdivision level of the triangular face *abc* in the initial mesh, so the internal force due to the above energy is

$$\mathbf{f}_{\mathrm{int}} = \frac{\partial E_{abc}^j}{\partial \mathbf{v}_{abc}^j} = (\mathbf{K}_{abc}^j)\{\mathbf{v}_{abc}^j\}.$$

Note that, the vertex positions in $\mathbf{v}_{abc}^j$ are obtained by a linear combination of the vertex positions in $\mathbf{v}_{abc}^0$, and hence we can write $\mathbf{v}_{abc}^j = (\mathbf{A}_{abc}^j)\mathbf{v}_{abc}^0$ where $(\mathbf{A}_{abc}^j)$ is the transformation (subdivision) matrix. Therefore, the expression for the elemental stiffness matrix is given by $\mathbf{K}_{abc} = (\mathbf{A}_{abc}^j)^{\mathrm{T}}(\mathbf{K}_{abc}^j)(\mathbf{A}_{abc}^j)$. It may be noted that this approach is applicable for modeling isotropic as well as an isotropic phenomena because k_{lm}, the spring-controlling variable, can be a time-dependent function in general, in addition, the entries in $\mathbf{K}_{abc}^j$ depend on the distance between the connected vertices. Therefore, unlike other elemental matrices, the stiffness matrix is a function of time which requires the recomputation at each time step in principle. Note that, the above spring-like energy is only one simple candidate of many possible choices. A large variety of functional formulations (such as simple *thin-plate-under-tension* energy or complex curvature-based energy) can be

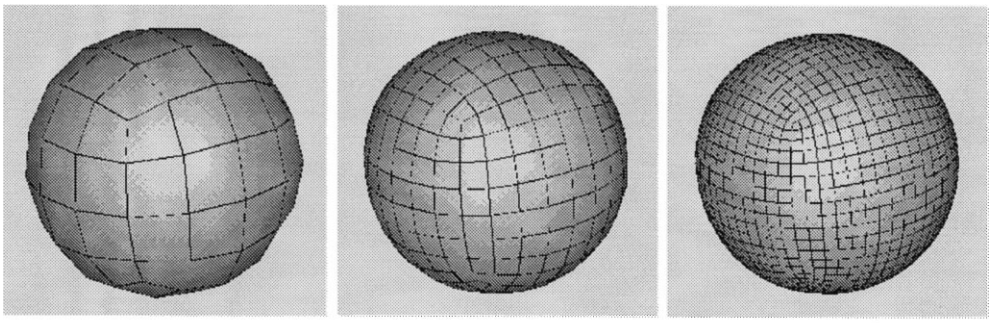


Fig. 6. Catmull–Clark subdivision: (a) initial mesh; (b) mesh obtained after one step of Catmull–Clark subdivision; and (c) obtained after another subdivision step.

employed to describe a wide range of material and physical behaviors such as linear elastic deformation and/or non-linear plastic deformation.

5. Dynamic Catmull–Clark subdivision surfaces

This section considers a new FEM model based on an approximating subdivision scheme, namely, Catmull–Clark subdivision technique. Please note, the dynamic formulation of Catmull–Clark subdivision previously proposed in Refs. [15,16,21] could not be generalized to other approximating subdivision schemes. The framework developed in this section can be easily generalized to other approximating subdivision schemes as shown in Section 7. In fact, a dynamic framework for Loop's technique (another popular approximating subdivision scheme) has been discussed in Section 6 using the algorithm proposed in this section. We first outline the Catmull–Clark subdivision scheme. Next, we present the dynamic formulation. In particular, we address the difference between the current work and prior results [15,16,21]. Finally, we discuss the finite element implementation.

5.1. Catmull–Clark subdivision scheme

Catmull–Clark subdivision scheme, like any other subdivision scheme, starts with an user-defined mesh of arbitrary topology. If refines the initial mesh by adding new vertices, edges and faces with each step of subdivision following a fixed set of subdivision rules. In the limit, a sequence of recursively refined polyhedral meshes will converge to a smooth surface. The subdivision rules are as follows:

1. For each face, a new face point is introduced which is the average of all the old vertices defining the face.
2. For each (non-boundary) edge, a new edge point is introduced which is the average of the following four points: two old vertices defining the edge and two new face points of the faces adjacent to the edge.
3. For each (non-boundary) vertex V, new vertex is introduced whose position is $F/n + 2E/n + ((n-3)V)/n$, where F is the average of the new face vertices of all faces adjacent to the old vertex V, E the average of the midpoints of all edges incident on the old vertex V and n the number of the edges incident on the vertex.
4. New edges are formed by connecting each new face point to the new edge points of the edges defining the old face and by connecting each new vertex point to the new edge points of all old edges incident on the old vertex point.
5. New faces are defined as faces enclosed by new edges.

An example of Catmull–Clark subdivision on an initial mesh is shown in Fig. 6. The most important property of the Catmull–Clark subdivision surfaces is that a smooth surface can be generated from any control mesh of arbitrary topology. Catmull–Clark subdivision surfaces include standard bicubic B-spline surface as their special case (i.e. the limit surface is a bicubic B-spline surface for a rectangular mesh with all non-boundary vertices of degree 4). In addition, the aforementioned subdivision rules generalize the recursive bicubic B-spline patch subdivision algorithm. for non-rectangular meshes, the limit surface converges to a bicubic B-spline surface except at a finite number of extraordinary points. These extraordinary points correspond to extraordinary vertices (vertices whose degree is not equal to 4) in the mesh. Note that, after the first subdivision, all faces are quadrilaterals, hence all new vertices created subsequently will have four incident edges. The number of extraordinary points on the limit surface is a constant, and is equal to the number of extraordinary vertices in the refined mesh obtained after applying one step of the Catmull–Clark subdivision on the initial mesh. The limit surface is curvature-continuos everywhere except at extraordinary vertices, where only tangent plane continuity is achieved.

5.2. Formulation

A systematic formulation of the newly proposed dynamic framework for Catmull–Clark subdivision surfaces is presented in this section. The key difference between the dynamic model developed in Refs. [15,16,21] and the one presented here is the representation of the limit surface. The previously proposed approach leads to diverse types of finite

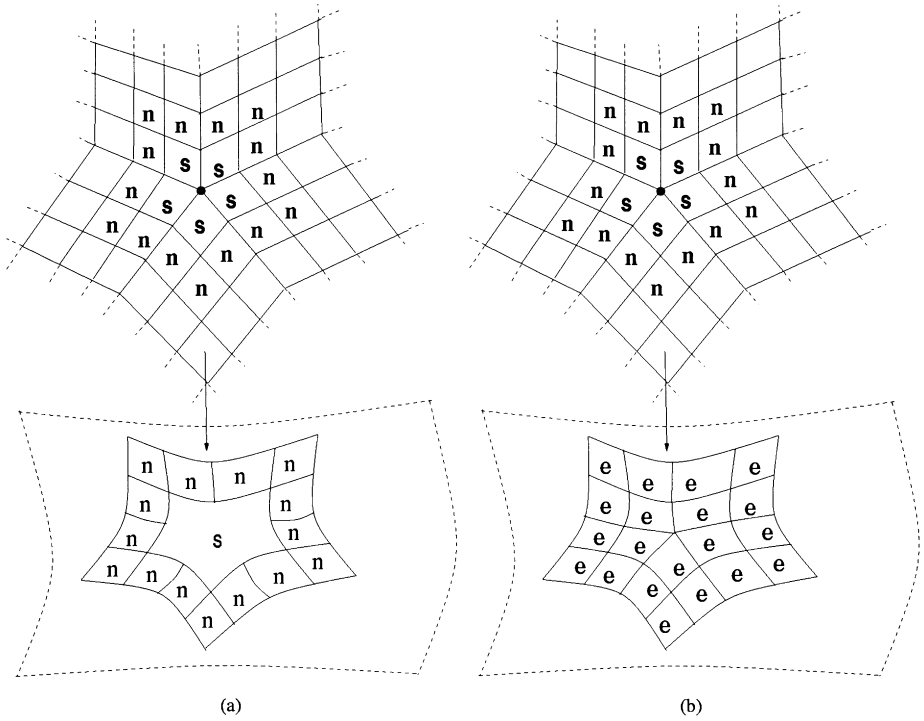


Fig. 7. A control mesh with an extraordinary vertex of degree 5 and the corresponding limit surface: (a) using the concepts developed in Refs. [15,16,21], where the limit surface consists of quadrilateral normal elements and a pentagonal special element; (b) using the unified approach developed in this paper, where the limit surface consists of one single type of quadrilateral finite element.

elements, whereas the present approach leads to a single type of finite elements. This is illustrated with a schematic diagram in Fig. 7.

Following the concepts developed in Refs. [15,16,21], the limit surface of the control mesh shown in Fig. 7, consists of quadrilateral bicubic B-spline patches corresponding to the faces marked 'n' (faces with no extraordinary points), and a pentagonal patch corresponding to the faces marked 's' (faces having one extraordinary vertex of degree five) (Fig. 7(a)). However, in this section, it has been shown that the entire limit surface can be expressed as a collection of quadrilateral patches as shown in Fig. 7(b) using the algorithm proposed by Stam [25]. We next discuss a local parameterization of the limit surface, which is critical to embed the limit surface in a dynamic framework.

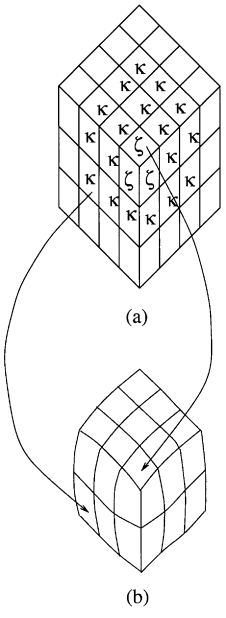


Fig. 8. In Catmull–Clark subdivision, each non-boundary quadrilateral face in the control mesh has a corresponding quadrilateral patch in the limit surface: (a) control mesh; and (b) limit surface.

As mentioned earlier, the control mesh (after at most one subdivision step) for the Catmull–Clark subdivision scheme consists of quadrilateral faces which lead to quadrilateral patches in the limit surface. For the sake of formulation simplicity, it has been assumed that each face has at most one extraordinary vertex. If this assumption is not valid, then one more subdivision step needs to be performed on the current control mesh in order to obtain a new control mesh on which the following analysis can be carried out. The number of quadrilateral patches in the limit surface is equal to the number of non-boundary quadrilateral faces in the control mesh (Fig. 8). Therefore, the smooth limit surface $\mathbf{s}$ can be expressed as

$$\mathbf{s} = \sum_{l=1}^{n} \mathbf{s}_l, \tag{15}$$

where n is the number of non-boundary faces in the control mesh and $\mathbf{s}_l$ is the smooth quadrilateral patch

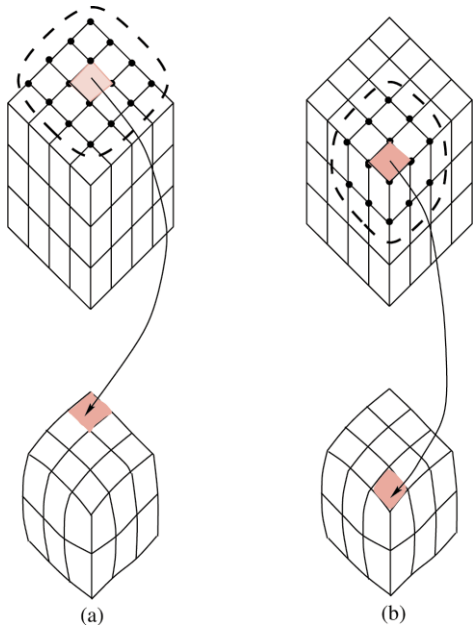


Fig. 9. (a) The marked 16 control vertices define the shaded quadrilateral patch associated with the shaded regular face in the control mesh. (b) The marked 14 control vertices define the shaded quadrilateral patch associated with the shaded irregular face in the control mesh.

corresponding to the lth non-boundary quadrilateral face in the control mesh. Each of these quadrilateral patches can be parameterized over the corresponding non-boundary quadrilateral face in the control mesh. However, since a quadrilateral face can easily be reparameterized over a $[0,1]^2$ domain, each quadrilateral patch is locally parameterized over $[0,1]^2$.

The non-boundary quadrilateral faces are of two types: (a) faces having no extraordinary vertices (dubbed as "regular" faces in Refs. [15,16,21], marked as κ in Fig. 8(a)) and (b) faces with one extraordinary vertex (dubbed as "irregular" faces in Refs. [15,16,21], marked as ζ in Fig. 8(a)). If there are $m$regular and $n-m$ irregular faces, then Eq. (15) can be rewritten as

$$\mathbf{s} = \sum_{i=1}^{m} \mathbf{s}_i + \sum_{j=1}^{n-m} \mathbf{s}_j, \tag{16}$$

where $\mathbf{s}_i$ is the quadrilateral patch corresponding to the ith regular face and $\mathbf{s}_j$ is the quadrilateral patch corresponding to the jth irregular face.

The quadrilateral patch in the limit surface corresponding to each regular face is a bicubic B-spline patch, which is defined over $[0,1]^2$. The set of control vertices defining this bicubic B-spline patch can be obtained using the adjacent face information. Therefore, the quadrilateral patches in the smooth limit surface corresponding to the regular faces in the control mesh can be easily expressed analytically, which are essentially bicubic B-spline patches defined by 16 control vertices over a $[0,1]^2$ domain. The analytic expression for the quadrilateral patch corresponding to the regular face i is given by

$$\mathbf{s}_i = \mathbf{J}_b(u,v)\mathbf{p}_i = (\mathbf{J}_b(u,v)\mathbf{A}_i)\mathbf{p} = \mathbf{J}_i(u,v)\mathbf{p}, \tag{17}$$

where $0 \le u,v \le 1$, $\mathbf{J}_b(u,v)$ is the collection of the bicubic B-spline basis functions, $\mathbf{p}_i$ the concatenation of the 16 control vertex positions defining the bicubic B-spline patch, $\mathbf{A}_i$ the selection matrix which when multiplied with $\mathbf{p}$, the concatenation of all the control vertex positions defining the smooth limit surface, selects the corresponding set of control vertices, and $\mathbf{J}_i(u,v) = \mathbf{J}_b(u,v)\mathbf{A}_i$.

By contrast, the analytic expression of the quadrilateral patches corresponding to the irregular faces in the control mesh was difficult to derive, and hence an alternative approach was taken in Refs. [15,16,21]. However, very recently an efficient scheme for evaluating Catmull–Clark subdivision surfaces at arbitrary parameter values has been proposed by Stam [25]. The proposed approach, involving eigen-analysis of the subdivision matrix, leads to an analytic expression of the quadrilateral patches which are parameterized over an irregular face in the control mesh, and hence over $[0,1]^2$ after reparameterization. Following the scheme developed by Stam [25], the quadrilateral patch corresponding to the irregular face j is given by

$$\mathbf{s}_j = \mathbf{J}_{d_k}(u,v)\mathbf{p}_j = (\mathbf{J}_{d_k}(u,v)\mathbf{A}_j)\mathbf{p} = \mathbf{J}_j(u,v)\mathbf{p}, \tag{18}$$

where $0 \le u,v \le 1$ as before. $\mathbf{J}_{d_k}(u,v)$ is the collection of basis functions for the corresponding quadrilateral patch in the smooth limit surface. The subscript d_k is used to denote the fact that the irregular face has an extraordinary vertex of degree k. The detailed derivation and the analytic expressions of these basis functions involving the eigenvalues and eigenvectors of the subdivision matrix can be found in Ref. [25]. The other symbols used in Eq. (18) have the usual meaning: $\mathbf{p}_j$ is the concatenation of the $2k+8$ control vertices defining the quadrilateral patch in the limit surface, $\mathbf{p}$ the concatenation of all the control vertex positions defining the smooth limit surface, $\mathbf{A}_j$ the selection matrix which when multiplied with $\mathbf{p}$ selects the corresponding set of control vertices, and $\mathbf{J}_j(u,v) = \mathbf{J}_{d_k}(u,v)\mathbf{A}_j$.

It may be noted that the number of control vertices in the initial mesh defining a quadrilateral patch in the smooth limit surface is $2k+8$, where $k=4$ in case the associated quadrilateral face in the control mesh is regular, or $k=$ degree of the extraordinary vertex if the associated quadrilateral face is irregular. For example, the shaded quadrilateral patch is associated with the shaded regular face in Fig. 9(a), and the 16 control vertices defining this patch (which is actually a bicubic B-spline patch) are marked. Similarly, the shaded quadrilateral patch is associated with the shaded irregular face in Fig. 9(b), and the 14 control vertices defining this patch are highlighted. Now an expression of the smooth limit surface can be formulated. Using Eqs. (16)–(18), it

can be shown that

$$\mathbf{s} = \sum_{i=1}^{m} \mathbf{J}_i\mathbf{p} + \sum_{j=1}^{n-m} \mathbf{J}_j\mathbf{p} \quad = \left(\sum_{i=1}^{m} \mathbf{J}_i + \sum_{j=1}^{n-m} \mathbf{J}_j\right)\mathbf{p} \quad = \mathbf{J}\mathbf{p}, \tag{19}$$

where

$$\mathbf{J} = \left(\sum_{i=1}^{m} \mathbf{J}_i + \sum_{j=1}^{n-m} \mathbf{J}_j\right).$$

Note that even though the initial mesh serves as the parametric domain of the smooth limit surface, each quadrilateral face in the initial mesh and consequently the smooth limit surface can be defined over a $[0,1]^2$ domain.

Once an analytic expression of the smooth limit surface of Catmull–Clark subdivision is derived, we then develop the dynamic model by considering the control vertex positions as time-varying variables. The velocity of the surface model can be expressed as $\dot{\mathbf{s}}(\mathbf{x}, u, v) = \mathbf{J}(\mathbf{x}, u, v)\dot{\mathbf{p}}$, where an overstruck dot denotes a time derivative and $x \in S^0$, S^0 being the domain by the initial mesh.

5.3. Finite element implementation

The smooth limit surface of Catmull–Clark subdivision comprises a collection of quadrilateral patches. Each quadrilateral patch is considered as a finite element. Therefore, within the unified framework the limit surface can be decomposed into a collection of single type of finite elements rather than *two* different types as in Refs. [15,16,21]. Our new FEM technique significantly simplifies the data structure and system architecture. Consequently, more efficient algorithms for finite-element assembly, dynamic simulation, etc. can be devised using this unified approach. The motion equation of the dynamic model is same as that of the dynamic model of butterfly-based subdivision:

$$\mathbf{M}\ddot{\mathbf{p}} + \mathbf{D}\dot{\mathbf{p}} + \mathbf{K}\mathbf{p} = \mathbf{f}_\mathbf{p}, \tag{20}$$

where $\mathbf{f}_\mathbf{p}$ is the generalized force vector and $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{K}$ are the mass, damping and stiffness matrices of the model. The expressions of the mass, damping and stiffness matrices for a quadrilateral element (which is a bicubic B-spline) can be written as

$$\mathbf{M}_e = \int_0^1 \int_0^1 \mu \mathbf{J}_b^{\mathrm{T}} \mathbf{J}_b \, \mathrm{d}u \, \mathrm{d}v, \tag{21}$$

$$\mathbf{D}_e = \int_0^1 \int_0^1 \gamma \mathbf{J}_b^{\mathrm{T}} \mathbf{J}_b \, \mathrm{d}u \, \mathrm{d}v, \tag{22}$$

and

$$\begin{aligned}\mathbf{K}_e = \int_0^1 \int_0^1 (&\alpha_{11}\{(\mathbf{J}_b)_u\}^{\mathrm{T}}\{(\mathbf{J}_b)_u\} + \alpha_{22}\{(\mathbf{J}_b)_v\}^{\mathrm{T}}\{(\mathbf{J}_b)_v\} \\ &+ \beta_{11}\{(\mathbf{J}_b)_{uu}\}^{\mathrm{T}}\{(\mathbf{J}_b)_{uu}\} + \beta_{12}\{(\mathbf{J}_b)_{uv}\}^{\mathrm{T}}\{(\mathbf{J}_b)_{uv}\} \\ &+ \beta_{22}\{(\mathbf{J}_b)_{vv}\}^{\mathrm{T}}\{(\mathbf{J}_b)_{vv}\}) \, \mathrm{d}u \, \mathrm{d}v\end{aligned} \tag{23}$$

where $\mathbf{J}_b$ is the bicubic B-spline basis matrix, $\mu(u,v)$ the mass density, $\gamma(u,v)$ the damping density, $\alpha_{ii}(u,v)$ and $\beta_{ii}(u,v)$ are the tension and rigidity functions, respectively. The subscript u and v denote partial derivatives with respect to u and v, respectively. The subscript e is used to indicate elemental matrices which are of size (16,16). Note that, the mass, damping and stiffness matrices for these elements can be evaluated analytically, provided the material properties (e.g. mass, damping, rigidity and bending distributions) have analytic expressions. In some cases, these distribution functions can be assumed to be constant to simplify the matter.

The mass damping and stiffness matrices for the quadrilateral elements which are not bicubic B-splines (corresponding to the irregular faces) can also be expressed analytically by simply replacing the matrix $\mathbf{J}_b$ in Eqs. (21)–(23) with the matrix $\mathbf{J}_{d_k}$ (refer to Eq. (18)), where k denotes the degree of the extraordinary vertex associated with the corresponding irregular face. These elemental matrices are of size $(2k + 8, 2k + 8)$. The generalized force vector for these elements can also be determined in a similar fashion. It may be noted that the limits of integration need to be chosen carefully for elemental stiffness matrices as the second derivative diverges near the extraordinary points for Catmull–Clark subdivision surfaces.

Even though an analytical expression for a *non-B-spline* quadrilateral element in the limit surface exists, it is cumbersome to actually evaluate the elemental matrix expressions. Numerical integration using Gaussian quadrature may be used to obtain approximations of these elemental matrices. However, in this paper, an approach similar to the FEM procedure presented in Section 4 is utilized because of its simplicity and effectiveness. An approximation of the smooth limit surface is obtained by refining the initial control mesh j times, and a spring–mass system is developed on this jth approximation level in a similar fashion as in Section 4.3. The physical matrices of this system are then used as an approximation to the actual physical matrices. This approximation has been found to be very efficient for implementation purposes.

6. Dynamic loop subdivision surfaces

Loop's subdivision scheme starts with a triangular control mesh and generates a smooth surface with triangular patches in the limit. It is an approximating subdivision scheme which generalizes recursive subdivision schemes for

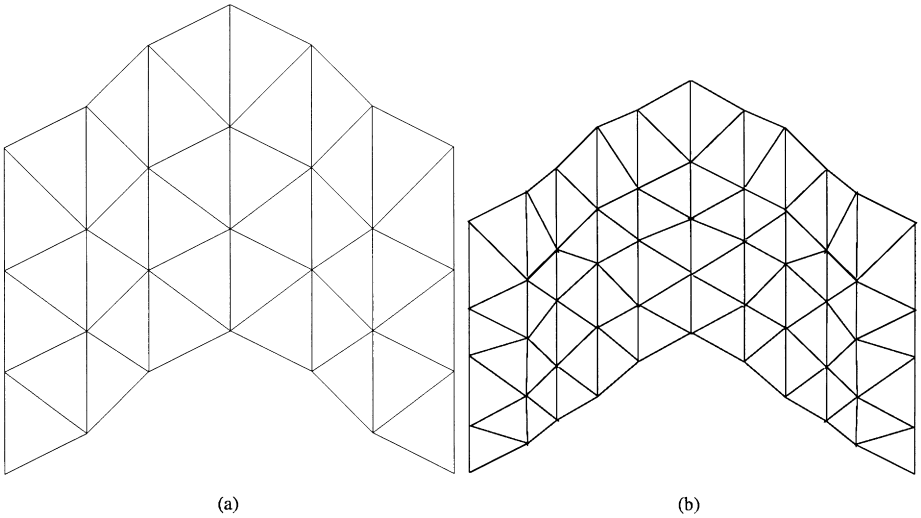


Fig. 10. (a) The control polygon with triangular faces. (b) Mesh obtained after one subdivision step using Loop's subdivision rules.

obtaining C^2 quartic triangular B-spline patches in a regular setting. In each step of Loop subdivision, each (non-boundary) triangular face is refined into 4 triangular faces using the following rules:

1. For each (non-boundary) vertex V of degree n, a new vertex point is introduced. The position of this newly introduced vertex point is given by $(\alpha(n)\mathbf{v} + \mathbf{v}_1 + \cdots + \mathbf{v}_n)/(\alpha(n) + n)$, where $\mathbf{v}$ is the position vector of vertex V, $\mathbf{v_1}, \ldots, \mathbf{v_n}$ are the vertex positions of the n vertices connected to vertex V $\alpha(n) = (n(1 - \beta(n)))/\beta(n)$ and $\beta(n) = 5/8 - (3 + 2\cos(2\pi/n))^2/64$.
2. For each (non-boundary) edge E, a new edge point is introduced. Let E be the connecting edge between vertices V_1 and V_2, and is shared by faces F_1 and F_2. If F_1 and F_2 have vertices V_{F_1} and V_{F_2}, respectively (apart from V_1 and V_2), then the position of the newly introduced edge point is given by $(3(\mathbf{v}_1 + \mathbf{v}_2) + \mathbf{v}_{\mathbf{F}_1} + \mathbf{v}_{\mathbf{F}_2})/8$, where $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_{\mathbf{F}_1}$ and $\mathbf{v}_{\mathbf{F}_2}$ are the position vector of the vertex V_1, V_2, V_{F_1} and V_{F_2}, respectively.
3. New edges are formed by connecting each new vertex point to the new edge points corresponding to the edges incident on the old vertex, and by connecting each new edge point to the new edge points of the other edges in the two faces, which shared the original edge.
4. New faces are defined as faces enclosed by the new edges.

Examples of refining an initial mesh using Loop's subdivision rules are shown in Figs. 10 and 11. These subdivision rules ensure tangent plane continuity of the limit surface even in a irregular setting, i.e. when the triangular control mesh has extraordinary vertices whose degree is not equal to 6. A detailed discussion on how to obtain positions and normals in the smooth limit surface generated by the Loop subdivision scheme can be found in Hoppe et al. [10].

6.1. Local parameterization

The limit surface obtained via Loop's subdivision scheme can be locally parameterized easily. This local parameterization scheme is very similar in nature to the one described for Catmull–Clark subdivision scheme in the previous section. For Loop's scheme, the smooth limit surface consists of triangular patches and the number of these triangular patches is the same as the number of non-boundary triangular faces in the control mesh. Therefore, each of the triangular patch in the limit can be locally parameterized

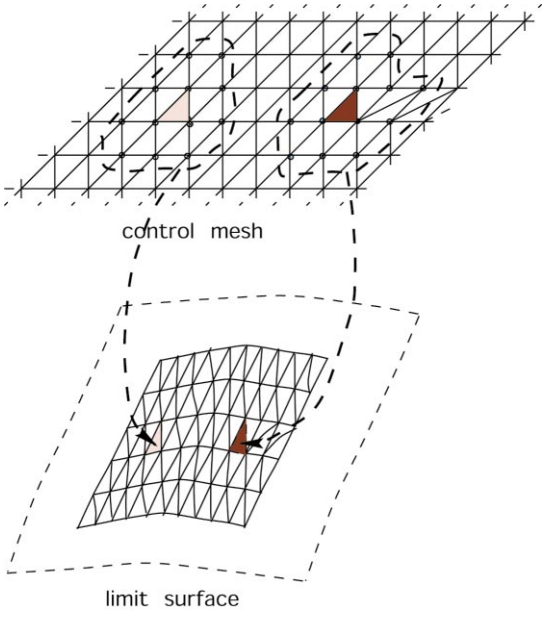


Fig. 11. An initial mesh and the corresponding limit surface obtained using Loop's subdivision rules. The domains of the shaded triangular patches in the limit surface are the corresponding triangular faces in the initial mesh. The encircled vertices are the control vertices for the corresponding triangular patch in the limit surface.

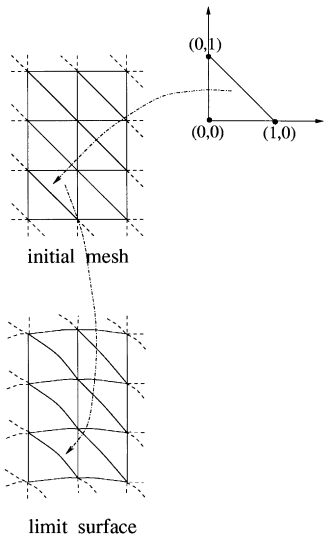


Fig. 12. Each triangular patch in the limit surface can be associated with a non-boundary triangular face in the initial mesh, which in turn can be parameterized over a triangle with vertices at (0,0), (1,0) and (0,1).

over the corresponding triangular face in the control mesh. It may be noted that each triangular face in the control mesh can be parameterized over a triangular domain whose vertices are located at (0, 0), (0, 1) and (1, 0), and hence each triangular patch and consequently the smooth limit surface can be defined over this domain (refer Fig. 12).

The triangular patches in the smooth limit surface are of two types. For a non-boundary triangular face in the control mesh with no extraordinary vertices (i.e. with three vertices of degree 6), the corresponding triangular patch in the limit surface is a particular type of triangular B-spline (the three-direction quartic box spline) whose analytic expression is easy to obtain. This triangular B-spline patch is controlled by 12 vertices as shown in Fig. 11 (the set of enclosed vertices in the left-hand side). The triangular patch in the limit surface corresponding to a non-boundary triangular face in the control mesh with one extraordinary vertex can also be expressed analytically using the schemes proposed by Stam [26]. This triangular patch is controlled by $n + 6$ vertices in the control mesh where n is the degree of the extraordinary vertex. The set of control vertices for a triangular patch of the later type is shown in the right-hand side of Fig. 11. Therefore, each triangular patch in the limit surface can be expressed analytically, and an expression for the limit surface similar to Eq. (19) can be obtained.

Once an expression for the limit surface using Loop's subdivision is obtained, the dynamic model can be developed following an exactly similar procedure described for Catmull–Clark subdivision scheme in the previous section. This is mainly because of the fact that both subdivision schemes have an analytical representation of the limit surface. Furthermore, the motion equation of the dynamic Loop subdivision model can also be derived in a similar fashion.

6.2. Finite element implementation

The implementation of the dynamic framework for Loop subdivision scheme using the unified approach treats each triangular patch in the limit surface as a finite element. Each triangular patch has an analytic expression, and hence the elemental physical matrices and the generalized force vector can be derived analytically. The derivation of an exact expression for elemental matrices is cumbersome for the triangular patches corresponding to the triangular faces with an extraordinary vertex, and numerical integration using Gaussian quadrature may be used for deriving an approximation. However, a practical alternative for implementation is to subdivide the control mesh j times using Loop's subdivision rules, and to build a spring–mass system on this jth level approximation as has been done for the dynamic modified butterfly subdivision model in Section 4.3. The physical matrices of this spring–mass system provide an approximation of the original physical matrices, and it works well in practice.

7. The unified approach for any subdivision scheme

The dynamic framework for modified butterfly and Catmull–Clark subdivision scheme can be generalized to any subdivision scheme. The key observation is that the smooth limit surface can be viewed as a collection of a single type finite elements. Because of the nature of recursive refinement, any subdivision-based scheme essentially defines a "natural" correspondence which leads to a local parameterization of the smooth limit surface. The unique type of the associated finite element results from the local parameterization scheme. This is evident from the triangular finite element patches developed for the modified butterfly subdivision scheme and from the quadrilateral finite element patches developed for Catmull–Clark subdivision scheme. We will present a general outline on how to provide a dynamic framework for interpolatory and approximating subdivision schemes.

7.1. Interpolatory subdivision schemes

Most of the interpolatory subdivision schemes are obtained by modifying the butterfly subdivision scheme [7]. Therefore, the framework for the modified butterfly subdivision scheme in Section 4 and its principles can be applied to other interpolatory subdivision schemes. The only difference is that the basis functions as well as the set of control vertices of arbitrary patch in the limit surface depend on the chosen interpolatory subdivision rules. It may also be noted that unlike the approximating schemes, the physical matrices cannot be obtained analytically as the basis functions corresponding to interpolatory subdivision schemes do not have any analytic expressions in general. Even though these matrices can be obtained via numerical integration, the point–mass system connected by springs as

developed in Section 4 is more preferable for implementation purposes because of efficiency reasons.

7.2. Approximating subdivision schemes

The unified approach for a dynamic model of Catmull–Clark subdivision can be generalized for other approximating subdivision schemes as well. This generalized approach involves three steps:

1. The limit surface obtained via an approximating subdivision scheme can be expressed as a collection of smooth patches which can be locally parameterized over a corresponding face in the control mesh. Each patch is *n*-sided if it is locally parameterized over an *n*-sided face. Analytic expressions for each of these patches can be derived even in the presence of extraordinary vertices in the control mesh, and hence an expression of the limit surface can be obtained.
2. Once an expression of the limit surface is obtained, the dynamic framework can be developed by considering control vertex positions as a function of time. The corresponding motion equation can be derived.
3. Each patch in the limit surface is treated as a finite element in implementation. The elemental mass, damping and stiffness matrices along with the generalized force vector can be obtained by either analytic or numerical integration. Alternatively, the control mesh can be subdivided j times to obtain an approximation of the smooth limit surface, and a spring–mass system can be developed on this approximation mesh. The physical matrices of this system provide an approximation to the original physical matrices and works well in practice.

8. Solid modeling applications

The proposed FEM-based dynamic subdivision models can be used to represent a wide variety of smooth shapes with arbitrary genus. The smooth limit object can be sculpted by applying synthesized forces in a direct and intuitive way in shape design applications for solid modeling. The underlying shape from a cloud of 3D points can also be recovered hierarchically using our FEM models. For data fitting applications, springs are attached to the initialized model from the data points in 3D, and the initialized model evolves dynamically according to the equation of motion subject to the applied spring forces and various geometric constraints. When an optimal fit to the given data set is achieved, the number of control vertices can be increased by replacing the original initial mesh by a new initial mesh obtained by applying a single subdivision step. This increases the number of degrees of freedom to represent the same limit surface and a new equilibrium position for the model with a much better fit to the given data set can be achieved. The fitting-error criteria for the discrete data can be computed according to distance between the data points and the points on the limit surface where the corresponding springs are attached. We now demonstrate modeling and data fitting examples using our dynamic FEM model.

In a shape modeling application, the user can specify any mesh as the initial (control) mesh, and the corresponding limit surface can be sculpted interactively by applying synthesized forces. In Fig. 13, we show several initial surfaces obtained from different control meshes and the corresponding modified surfaces after interactive sculpting. To change the shape of an initial surface, the user can attach springs from different points in 3D to the nearest points on the limit surface such that the limit surface deforms towards these points to generate the desired shape. It may be noted that the user can specify these data points in several ways—directly in 3D, on a 2D plane at a fixed height (using mouse input) or from a file containing (x,y,z) coordinates of the points in 3D. Also, the distance between control vertices of the initialized mesh is used as the natural (rest) length of the spring attached between those vertices. When the model deforming under the influence of spring forces reaches an equilibrium, the control mesh can be subdivided to obtain another control mesh with more degrees of freedom for the same smooth limit surface if the error is unacceptable. For modeling purposes, error is defined as the maximum distance between a data point and the nearest point on the limit surface expressed as a percentage of the diameter of the smallest sphere enclosing all the data points. The time needed for the initialized model to deform into the final shape depends on the number of degrees of freedom of the model as well as on the number of data points exerting force on the model. Generally speaking, less number of degrees of freedom leads to faster deformation (a smaller system of equations is solved). For the examples shown in Fig. 13, the deformations took approximately 30–45 s under normal system load on a Ultra-SPARC 30 machine. A small time step is used for stability, and one conjugate gradient iteration was necessary between each Euler step.

The initial mesh of the smooth surface shown in Fig. 13(a) has 125 faces and 176 vertices (degrees of freedom), which is deformed to the smooth shape shown in Fig. 13(c) by interactive spring force application. The initial mesh of the closed solid shape in Fig. 13(e) has 24 faces and 14 vertices. This solid shape is deformed to the shape shown in Fig. 13(g). The one hole torus in Fig. 13(i) and the corresponding modified shape in Fig. 13(k) have initial meshes with 64 faces and 32 vertices. A two hole torus with a control mesh of 272 faces and 134 vertices, shown in Fig. 13(m), is dynamically sculpted to the shape shown in Fig. 13(o).

We have also performed several experiments testing the applicability of our model to recover the underlying shapes from a cloud of points in 3D. In all the experiments, the initialized dynamic model has a control mesh comprising of 24 triangular faces and 14 vertices whereas the control mesh of the fitted model has 384 triangular faces and 194 vertices.

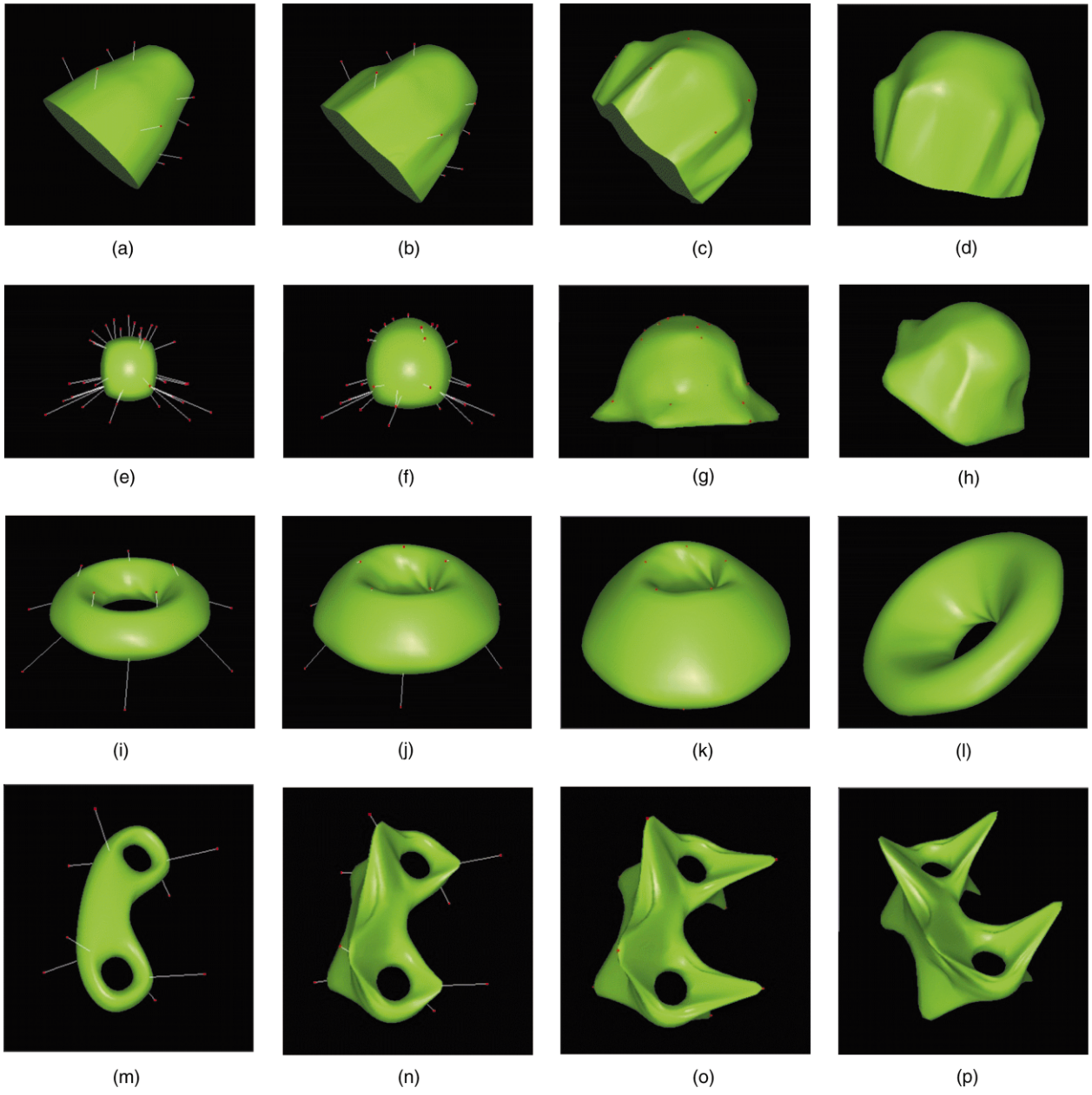


Fig. 13. *First column*: Initial shapes along with attached springs for deformation. *Second column*: Deformation of initial shapes due to spring forces. *Third column*: The final deformed shape. *Fourth column*: Another view of the final deformed shape.

It may be noted that once an optimal shape defined by a fixed number of control vertices (determined by subdivision levels) is recovered, the limit smooth model is capable of refining itself in accordance with the data-fitting criteria, thereby increasing the degrees of freedom of the recovered shape only when necessary. For the fitting-error (defined as the maximum distance between a data point and the nearest point on the limit surface expressed as a percentage of the diameter of the smallest sphere enclosing the object) of approximately 3%, the initialized model is refined twice. The data-fitting examples are shown in Fig. 14. In the first data fitting experiment, range data acquired from multiple views of a light bulb is used and the model was initialized inside the 1000 data points (Fig. 14(a)). The fitted dynamic model is shown in Fig. 14(b). In the next experiment, the shape of a mechanical part is recovered from a range data-set containing 2031 data points (Fig. 14(c) and (d)). We also recover the shape of a human head from the data set as shown in Fig. 14(e). The head data set has 1779 3D points. The time of dynamic evolution for the fitting of range data-sets used in the experiments is approximately 3 min on a Ultra-SPARC 30 workstation. It may be noted that the final shape with a very low error tolerance is recovered using a very few number of control points in comparison

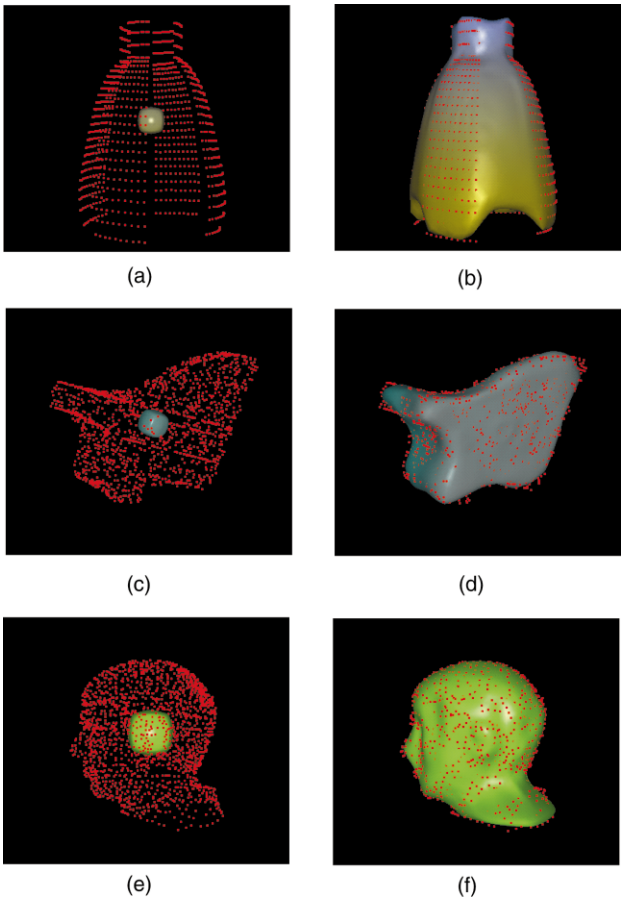


Fig. 14. (a), (c) and (e): Collection of points in 3D along with the initialized model; (b), (d) and (f): the corresponding fitted dynamic subdivision surface model.

to the large number of data points present in the original range data set.

9. Conclusions

In this paper, we have presented a new FEM-based dynamic framework where a single type of subdivision-based finite elements are used to represent the smooth limit surface generated by any subdivision scheme. The primary objective is to integrate physics-based modeling techniques with geometric subdivision methodology for the interactive sculpting and direct manipulation of the limit surface of prevalent subdivision schemes. We have proposed a unified approach and demonstrated how to transform any subdivision scheme into our dynamic modeling framework. Modelers can *physically* sculpt *virtual* objects defined through arbitrary procedure-based subdivision techniques in a natural and intuitive manner within the proposed dynamic framework. Users can also directly enforce various functional and aesthetic requirements on the limit surface without the need to explicitly manipulate the control vertices. Furthermore, this dynamic framework permits physics-based models to be refined adaptively in a hierarchical fashion which is an intrinsic feature of subdivision geometry. Our experiments have demonstrated the applicability of the new unified FEM-based framework in solid modeling and data fitting applications. This unified method will offer a greater potential for popular subdivision techniques in solid and geometric modeling, interactive graphics, finite element analysis, and engineering design applications.

At present, we are planning to pursue several research directions based on our current work. First, the capability for adaptive local subdivision of the control mesh may be more attractive to modelers in certain applications such as feature description and manipulation. In general, adaptive local refinement in the regions of interest is a non-trivial matter. Second, novel algorithms that support automatic modification of complex topology and/or automatic change of subdivision rules should be developed for our FEM-based modeling framework. Third, a wide variety of constraint imposition techniques as well as force-based sculpting toolkits should be investigated in order to further enhance the functionalities of our modeling system.

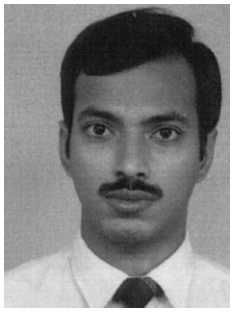
Acknowledgements

This research was supported in part by the NSF CAREER award CCR-9896123, the NSF grant DMI-9896170, and a research grant from Ford Motor Company to H. Qin; the NSF grant IIS-9811042 and the NIH grant ROI-RR13197 to B.C. Vemuri. We wish to acknowledge Drs Hughes Hoppe and Kari Pulli for the data sets.

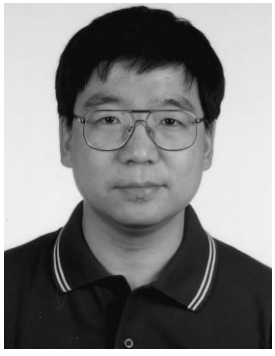
References

[1] Ball AA, Storry DJT. An investigation of curvature variations over recursively generated B-spline surfaces. ACM Transactions on Graphics 1990;9(4):424–37.
[2] Catmull E, Clark J. Recursively generated B-spline surfaces on arbitrary topological meshes. Computer Aided Design 1978;10(6):350–5.
[3] Chaikin GM. An algorithm for high speed curve generation. Computer Vision, Graphics and Image Processing 1974;3(4):346–9.
[4] DeRose T, Kass M, Truong T. Subdivision surfaces in character animation. In: Computer Graphics Proceedings, ACM SIGGRAPH, Annual Conference Series, July 1998. p. 85–94.
[5] Doo D. A subdivision algorithm for smoothing down irregularly shaped polyhedrons. In: Proceedings on Interactive Techniques in Computer Aided Design, 1978. p. 157–65.
[6] Doo D, Sabin M. Analysis of the behavior of recursive division surfaces near extraordinary points. Computer Aided Design 1978; 10(6):356–60.
[7] Dyn N, Levin D, Gregory JA. A butterfly subdivision scheme for surface interpolation with tension control. ACM Transactions on Graphics 1990;9(2):160–9.
[8] Habib A, Warren J. Edge and vertex insertion for a class of C1 subdivision surfaces. Computer Aided Geometric Design 2000 (in press).
[9] Halstead M, Kass M, DeRose T. Efficient, fair interpolation using Catmull–Clark surfaces. In: Computer Graphics Proceedings, ACM SIGGRAPH, Annual Conference Series, August 1993. p. 35–44.

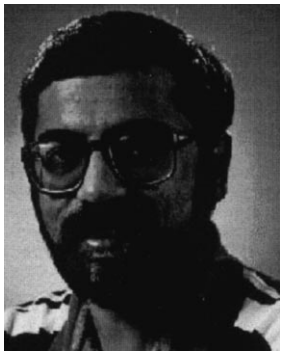
[10] Hoppe H, DeRose T, Duchamp T, Halstead M, Jin H, McDonald J, Schweitzer J, Stuetzle W. Piecewise smooth surface reconstruction. In: Computer Graphics Proceedings, ACM SIGGRAPH, Annual Conference Series, July 1994. p. 295–302.
[11] Kobbelt L. Interpolatory refinement by variational methods. In: Chui C, Schumaker L, editors. Approximation theory VIII, Wavelets and multilevel approximation, vol. 2. Singapore: World Scientific, 1995. p. 217–24.
[12] Kobbelt L. A variational approach to subdivision. Computer-Aided Geometric Design 1996;13:743–61.
[13] Kobbelt L, Schröder P. Constructing variationally optimal curves through subdivision, Tech. Rep. CS-TR-97-05, California Institute of Technology Computer Science Department Technical Report, 1997.
[14] Loop C. Smooth subdivision surfaces based on triangles. MS thesis, Department of Mathematics, University of Utah, 1987.
[15] Mandal C. A dynamic framework for subdivision surfaces. PhD thesis, University of Florida, Gainesville, 1998, Available as UF-CISE-TR-98-022 from http://www.cise.ufl.edu/tech-reports/tech-reports/tr98-abstract.shtml
[16] Mandal C, Qin H, Vemuri BC. Dynamic smooth subdivision surfaces for data visualization. In: IEEE Visualization'97 Conference Proceedings, Phoenix, AZ, October 1997. p. 371–7.
[17] Mandal C, Qin H, Vemuri BC. Direct manipulation of butterfly subdivision surfaces: a physics-based approach. Tech. Rep. CISE-TR-98-009, University of Florida, 1998.
[18] Peters J, Reif U. The simplest subdivision scheme for smoothing polyhedra. ACM Transactions on Graphics 1997;16(4):420–31.
[19] Peters J, Reif U. Analysis of generalized B-spline subdivision algorithms. SIAM Journal of Numerical Analysis 2000 (in press) available at ftp://ftp.cs.purdue.edu/pub/jorg/9697abg.ps.Z.
[20] Qin H, Terzopoulos D. D-NURBS: a physics-based framework for geometric design. IEEE Transactions on Visualization and Computer Graphics 1996;2(1):85–6.
[21] Qin H, Mandal C, Vemuri BC. Dynamic Catmull–Clark subdivision surfaces. IEEE Transactions on Visualization and Computer Graphics 1998;4(3):215–29.
[22] Reif U. A unified approach to subdivision algorithms near extraordinary points. Computer Aided Geometric Design 1995;12(2):153–74.
[23] Schweitzer JE. Analysis and Application of Subdivision Surfaces. PhD thesis, University of Washington, Seattle, 1996.
[24] Sederberg TW, Zheng J, Sewell D, Sabin M. Non-uniform recursive subdivision surfaces. In: Computer Graphics Proceedings, ACM SIGGRAPH, Annual Conference Series, July 1998. p. 387–94.
[25] Stam J. Exact evaluation of Catmull–Clark subdivision surfaces at arbitrary parameter values. In: Computer Graphics Proceedings, ACM SIGGRAPH, Annual Conference Series, July 1998. p. 395–404.
[26] Stam J. Evaluation of Loop subdivision surfaces. In: Computer Graphics Proceedings CDROM, ACM SIGGRAPH, Annual Conference Series, July 1998.
[27] Terzopoulos D, Platt J, Barr A, Fleischer K. Elastically deformable models. In: Computer Graphics Proceedings, ACM SIGRAPH, Annual Conference Series, 1987. p. 205–14.
[28] Zorin D, Schröder P, Sweldens W. Interpolating subdivision for meshes with arbitrary topology. In: Computer Graphics Proceedings, ACM SIGGRAPH, Annual Conference Series, August 1996. p. 189–92.
[29] Zorin D. Smoothness of stationary subdivision on irregular meshes, Constructive Approximation, submitted for publication, available as Stanford Computer Science Lab Tech. Rep. CSL-TR-98-752, 1998.



Chhandomay Mandal is a senior member of technical staff at Sun Microsystems, Inc. He received the BTech (Hons.) degree in electronics and Electrical Communication Engineering from Indian Institute of Technology, Kharagpur, India, in 1995, and the MS and PhD degrees in Computer and Information Science and Engineering from the University of Florida, Gainesville in 1997 and 1998, respectively. He received the best senior project award from the ECE Dept, IIT-KGP in 1995, and special mention for academic achievements from the University of Florida during the academic years 1995–1998. His research interests include computer graphics and modeling, operating systems, and computer architecture.



Hong Qin is an Assistant Professor of Computer Science at State University of New York at Stony Brook, where he is also a member of SUNYSB Center for Visual Computing. He received his BS (1986) degree and his MS degree (1989) in Computer Science from Peking University in Beijing, P.R. China. He received his PhD (1995) degree in Computer Science from the University of Toronto. During 1989–1990, he was a research scientist at North-China Institute of Computing Technologies. During 1990–1991, he was a PhD candidate in Computer Science at the University of North Carolina at Chapel Hill. During 1996–1997, he was an Assistant Professor of Computer and Information Science and Engineering at the University of Florida. He received the Honor Student Award from 1983 to 1985 and the Best Graduate Award in 1986 from Peking University. During his years at the University of Toronto, he received a University of Toronto Open Doctoral Fellowship. In 1997, Dr Qin was awarded NSF CAREER Award from the National Science Foundation (NSF). He is a member of ACM, IEEE and SIAM.



Baba C. Vemuri received the PhD degree in Electrical and Computer Engineering from the University of Texas at Austin in 1987 and joined the Department of Computer and Information Sciences at the University of Florida, Gainesville, where he is currently a professor of Computer and Information Sciences and Electrical Engineering. He was a Visiting Faculty Member at IBM T.J. Watson Research Center, Yorktown Heights, NY, with the Exploratory Computer vision Group in the summer of 1988 and 1992. The academic year 1989–1990, he was Visiting Research Scientist with the German Aerospace Research Institute, DLR Oberpfaffenhofen, Germany. His research interests are in computational vision, modeling for vision and graphics, medical imaging and applied mathematics. In the past several years, his research work has primarily focused on efficient algorithms for 3D shape modeling and recovery from image data. Dr Vemuri received the NSF Research Initiation and the Whitaker Foundation Awards in 1988 and 1994, respectively. He was a recipient of the best paper in computer vision from the IAPR (Norway) in 1992 and subsequently received the best peer reviews for his paper in ECCV'94. Dr Vemuri served as a program committee member for several IEEE conferences and was the chair of the Third IEEE Workshop on Biomedical Image Analysis in 1998. He was also the chairman of the SPIE sponsored conferences on Geometric Methods in Computer Vision in 1991 and 1993, respectively. He is currently an Associate Editor for the IEEE Transactions on Medical Imaging and served as an AE for the Transaction on Pattern Analysis and Machine Intelligence in the past. Dr Vemuri is a senior member of the IEEE Computer Society.