

Dynamic Modeling of Butterfly Subdivision Surfaces

Chhandomay Mandal, Hong Qin, *Member, IEEE*, and Baba C. Vemuri, *Senior Member, IEEE*

Abstract—In this paper, we develop integrated techniques that unify physics-based modeling with geometric subdivision methodology and present a scheme for dynamic manipulation of the smooth limit surface generated by the (modified) butterfly scheme using physics-based “force” tools. This procedure-based surface model obtained through butterfly subdivision does not have a closed-form analytic formulation (unlike other well-known spline-based models) and, hence, poses challenging problems to incorporate mass and damping distributions, internal deformation energy, forces, and other physical quantities required to develop a physics-based model. Our primary contributions to computer graphics and geometric modeling include: 1) a new hierarchical formulation for locally parameterizing the butterfly subdivision surface over its initial control polyhedron, 2) formulation of dynamic butterfly subdivision surface as a set of novel finite elements, and 3) approximation of this new type of finite elements by a collection of existing finite elements subject to implicit geometric constraints. Our new physics-based model can be sculpted directly by applying synthesized forces and its equilibrium is characterized by the minimum of a deformation energy subject to the imposed constraints. We demonstrate that this novel dynamic framework not only provides a direct and natural means of manipulating geometric shapes, but also facilitates hierarchical shape and nonrigid motion estimation from large range and volumetric data sets using very few degrees of freedom (control vertices that define the initial polyhedron).

Index Terms—Dynamic modeling, physics-based geometric design, geometric modeling, CAGD, subdivision surfaces, deformable models, finite elements, interactive techniques.



1 INTRODUCTION

IN visual computing areas, subdivision surfaces are extensively employed to model smooth shapes of arbitrary topology for interactive graphics, computer animation, and geometric design applications. A typical 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-defined initial control mesh. Despite the presence of a variety of subdivision schemes in the computer graphics and geometric modeling literature, there is no direct and natural way of manipulating the limit surface. The current state-of-the-art only permits modelers to interactively obtain the desired effects on the smooth surface by kinematically manipulating the vertex positions at various levels of subdivision hierarchy. This paper tackles the challenging problem of direct manipulation of the limit subdivision surface at arbitrary locations/areas and offers a novel solution to this problem by embedding the modified butterfly subdivision scheme in a physics-based modeling framework. Consequently, our methodology and algorithms permit the user to physically modify the shape of subdivision surfaces at desired

locations via application of forces, augmenting the conventional geometric solutions that only allow the operations on control vertices. This provides the user an intuitive and natural feeling that is produced while modeling with real clay/play-dough. We also demonstrate that the proposed model efficiently recovers static and dynamic shapes from large range and volumetric data sets. Our motivation is to *integrate* the advantages of subdivision surface-based and physics-based modeling techniques to solve important theoretical and practical problems. In particular, the existing “direct” editing tools for subdivision surfaces provide direct manipulation of control points, whereas we provide interaction tools that directly manipulate the smooth limit surface instead of the control points. First, we will briefly review the previous work on subdivision surfaces.

2 BACKGROUND REVIEW

Chaikin [1] first introduced the concept of subdivision to the graphics community for generating a smooth curve from a given control polygon. During the last two decades, a wide variety of subdivision schemes for modeling smooth surfaces of arbitrary topology have been derived following Chaikin’s pioneering work on curve generation. In general, these subdivision schemes can be categorized into two distinct classes: 1) approximating subdivision techniques and 2) interpolating subdivision techniques.

Among the approximating schemes, the techniques of Doo [2] and Sabin [3] and Catmull and Clark [4] generalize the idea of obtaining uniform biquadratic and bicubic B-spline patches, respectively, from a rectangular control mesh. Catmull and Clark [4] developed a method for

- C. Mandal is with Sun Microsystems Inc., 5 Omni Way, UCHL05-105, Chelmsford, MA 01851. E-mail: chhandomay.mandal@sun.com.
- H. Qin is with the Department of Computer Science, State University of New York at Stony Brook, Stony Brook, NY 11794-4400. E-mail: qin@cs.sunysb.edu.
- B.C. Vemuri is with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611. E-mail: vemuri@cise.ufl.edu.

Manuscript received 11 Apr. 2000; revised 23 May 2000; accepted 30 May 2000.

For information on obtaining reprints of this article, please send e-mail to: tcvg@computer.org, and reference IEEECS Log Number 111876.

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 [5] presented a similar subdivision scheme based on the generalization of quartic triangular B-splines for triangular meshes. Hoppe et al. [6] extended his work to produce piecewise smooth surfaces with selected discontinuities. Halstead et al. [7] proposed an algorithm to construct a Catmull-Clark subdivision surface that interpolates the vertices of a mesh of arbitrary topology. Peters and Reif [8] proposed a simple subdivision scheme for smoothing polyhedra. Most recently, nonuniform Doo-Sabin and Catmull-Clark surfaces that generalize nonuniform tensor product B-spline surfaces to arbitrary topologies were introduced by Sederberg et al. [9]. All the schemes mentioned above generalize recursive subdivision schemes for generating limit surfaces with a known parameterization. Various issues involved with character animation using these approximating subdivision schemes were discussed at length by DeRose et al. [10].

The most well-known interpolation-based subdivision scheme is the “butterfly” algorithm proposed by Dyn et al. [11]. Butterfly subdivision method, like other subdivision schemes, makes use of a small number of neighboring vertices for subdivision. It requires simple data structures and is extremely easy to implement. However, it needs a topologically regular setting of the initial (control) mesh in order to obtain a smooth C^1 limit surface. A variant of this scheme with better smoothness properties can be found in Dyn et al. [12]. Zorin et al. [13] have 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 decomposition of polyhedral surfaces, and multiresolution editing.

The derivation of various mathematical properties of the smooth limit surface generated by the subdivision algorithms is rather complex. Doo and Sabin [14] first analyzed the smoothness behavior of the limit surface using the Fourier transform and an eigen-analysis of the subdivision matrix. Ball and Storry [15], [16] and Reif [17] 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. Micchelli and Prautzsch [18] discussed the mathematical properties such as convergence, continuity, and differentiability for curves derived via uniform subdivision. Cavaretta et al. [19] presented a detailed and systematic study of stationary subdivision algorithms in a multidimensional (\mathcal{R}^n) setting. Algebraic properties and numerical issues such as necessary and sufficient conditions for the convergence of subdivision algorithms and the smoothness of the limiting functions are derived and analyzed using mathematical tools of Fourier and functional analysis. Specific subdivision schemes were analyzed by Schweitzer [20], Habib and Warren [21], Peters and Reif [22], and Zorin [23]. Most recently, Stam [24]

presented a method for exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values.

A variational approach for subdivision surfaces has been proposed by Kobbelt [25], [26] and by Kobbelt and Schröder [27]. In this approach, the goal is to find appropriate subdivision rules that minimize certain energy functionals and 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. Weimer and Warren [28] have derived a family of subdivision schemes for surfaces that approximate the behavior of thin plate splines. However, their subdivision rules are only applicable to regular grids with rectangular structures that are equivalent to tensor-product splines defined on \mathcal{R}^2 . Our physics-based approach significantly differs from the existing variational techniques because we do not intend to seek a set of subdivision rules minimizing certain energy functionals. Instead, we impose physical properties on the smooth limit surface (generated by the fixed set of subdivision rules) to make it behave in a physically plausible way in response to the application of physics-based force tools.

In recent years, researchers have proposed subdivision-based modeling tools for hierarchical and variational operations. Kurihara [29] has proposed a hierarchical editing method for subdivision surfaces generated by Doo and Sabin’s scheme. The method is essentially based on hierarchical B-splines and allows users to directly edit discrete control points (and not the limit surface) within a subdivision hierarchy. Pulli and Lounsbery [30] have proposed a similar system which supports hierarchical and direct editing of control points used to generate subdivision limit surface. Zorin et al. [31] have presented an interactive system that supports multiresolution editing on control points and meshes with the help of synthesis and analysis filters. The fine meshes they use are obtained after several level of Loop’s subdivision. Therefore, their meshes must maintain subdivision connectivity. Kobbelt et al. [32] have generalized their techniques to arbitrary triangular meshes without requiring subdivision connectivity. They introduced the concept of discrete fairing and allowed energy functionals to be defined over the discrete setting of triangular meshes and control points (polygons) at various levels of subdivision can be smoothed using these discrete energy functionals. Note that these existing subdivision surface editing tools support direct manipulation of the control points governing the limit surface, whereas our methodology permits direct sculpting on the limit surface itself.

3 RATIONALE

Although recursive subdivision surfaces are powerful for representing smooth geometric shapes of arbitrary topology, they constitute a purely geometric representation. In addition, conventional geometric modeling with subdivision surfaces may be difficult for representing extremely

complicated objects. For example, modelers are faced with the tedium of indirect shape modification and refinement through time-consuming operations on a large number of (often irregular) control vertices when using typical subdivision surface-based modeling schemes. Despite the advent of advanced 3D graphics interaction tools, these indirect geometric operations remain nonintuitive and laborious in general. In contrast, physics-based modeling provides a shape modeling approach that can overcome most of the limitations associated with traditional geometric modeling techniques. Free-form deformable models governed by the laws of continuum mechanics are of particular interest in this context. These dynamic models respond to externally applied forces in a very intuitive manner. The dynamic formulation marries the model geometry with time, mass, damping, and constraints via a force balance equation. Dynamic models produce smooth, natural motions which are intuitive to control. In addition, they facilitate interaction—especially direct manipulation of complex geometries. Furthermore, the equilibrium state of the model is characterized by a minimum of the deformation energy of the model subject to the imposed constraints. The deformation energy functionals can be formulated to satisfy local and global modeling criteria and geometric constraints relevant to shape design can also be imposed. 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 computer graphics by Terzopoulos et al. [33] and further developed by Terzopoulos and Fleischer [34], Pentland and Williams [35], Metaxas and Terzopoulos [36], and Vemuri and Radisavljevic [37]. Celniker and Gossard [38] developed a system for interactive free-form design based on the finite element optimization of energy functionals proposed in [34]. Bloor and Wilson [39], [40], Celniker and Welch [41], and Welch and Witkin [42] proposed deformable B-spline curves and surfaces which can be designed by imposing the shape criteria via the minimization of the energy functionals subject to hard or soft geometric constraints through Lagrange multipliers or penalty methods. Qin and Terzopoulos [43], [44], [45] developed dynamic NURBS (D-NURBS) which are very sophisticated 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, hence intuitively predictable, motion and shape variation.

A severe limitation of the existing deformable models, including D-NURBS, is that they are defined on a rectangular parametric domain. Hence, it can be very difficult to model surfaces of arbitrary genus using these models. DeRose et al. [10] assigned material properties to control meshes at various subdivision levels in order to simulate cloth dynamics using subdivision surfaces. Note that they assign physical properties on the control meshes at various levels of subdivision and not on the limit surface itself and, hence, cannot solve the modeling goal we are trying to achieve. Previously, we introduced a dynamic Catmull-Clark subdivision surface model [46], [47] which

combined the benefits of subdivision surfaces for modeling arbitrary topology, as well as that of dynamic splines for interactive shape manipulation, by applying synthesized forces. The dynamic (modified) butterfly subdivision surface model formulated and developed in this paper aims to achieve the same long-term objective, i.e., a formal mechanism of allowing the modeler to directly and intuitively manipulate the smooth limit surface of arbitrary topology as if they were seamlessly sculpting a piece of real-world “clay.” However, this new model is superior to our previously reported research in several significant aspects which will be detailed in Section 4. In particular, we derive a novel technique for locally parameterizing the smooth limit surface generated by the modified butterfly subdivision surface algorithm which embeds the proposed model in a dynamic framework in a straightforward manner. The model can be initialized interactively by a user-defined control mesh and is amenable to further sculpting via direct application of synthesized forces to any region of object surface.

The dynamic subdivision surface model has been developed primarily for modeling arbitrary (known) topology where modelers can directly manipulate the limit surface by applying synthesized forces in an intuitive fashion. However, as we have shown in our earlier work [48], another important application of the dynamic subdivision surfaces is in nonrigid shape reconstruction/recovery. Accurate shape recovery requires distributed parameter models which typically possess a large number of degrees of freedom. On the other hand, efficient shape representation imposes the requirement of geometry compression, i.e., models with fewer degrees of freedom. These requirements are conflicting and numerous researchers have been seeking to strike a balance between these contradicting requirements [37], [45], [48], [49], [50], [51], [52], [53], [54]. Another important criterion in shape design is that the initialization of the model during the shape recovery process should not be restricted to globally parameterized input meshes since it may be infeasible to globally parameterize shapes of arbitrary topology. A physics-based model best satisfying the above mentioned criteria is an ideal candidate for a solution to the shape recovery problem for obvious reasons.

Deformable models, which come in many varieties, have been used to solve the problem in the physics-based modeling paradigm. These models involve the use of either fixed size [37], [50], [55], [56], [57] or adaptive size [51], [53], [58], [59], [60], [61] grids. The models with fixed grid size generally use a fewer number of degrees of freedom for representation at the cost of accuracy of the recovered shape. On the other hand, models using adaptive grids generally need a large number of degrees of freedom to recover the shapes accurately. Note that the shapes being recovered from the image data are smooth in most of the medical applications, i.e., the anatomical structures, even with a considerable amount of details, have more or less a C^1 surface. Under these circumstances, the finite element approaches as in Cohen and Cohen [50] and McNerney and Terzopoulos [53] need a *large* number of degrees of freedom for deriving a smooth and accurate representation. In addition, they cannot represent shapes with known arbitrary

topology. Moreover, almost all of these schemes require a globally parameterized mesh, as their input which may be infeasible at times.

Our previous dynamic subdivision surface model [46], [47], [48] offered a novel solution to the above-mentioned problem as it could recover complex shapes in a hierarchical fashion using very few degrees of freedom without requiring parameterized input mesh. However, the model proposed in this paper outperforms the previous one in the compactness of the model representation. We will show experimental results in support of this claim. We will also demonstrate the potential of this model in motion tracking and visualization of a dynamically deforming shape from a time sequence of volumetric data sets. Like the previous model, the dynamic modified butterfly subdivision surface model also deforms under the influence of synthesized forces to fit the underlying shape in the given range or volumetric data set via the principle of energy minimization.

4 CONTRIBUTIONS

We summarize our contributions along with the advantages of the proposed dynamic modified butterfly subdivision surface model over the dynamic Catmull-Clark subdivision surface model [46], [47]. The primary contributions are as follows:

- We systematically derive a local parameterization scheme for the modified butterfly subdivision surfaces in a hierarchical fashion and, subsequently, the initial control polyhedron can be viewed as the parametric domain of the physics-based smooth limit surface.
- We treat the dynamic subdivision surface in the limit as a “real” physical object and represent the smooth limit surface as a set of novel finite elements whose shape (basis) functions are derived using the modified butterfly subdivision scheme. We envision that this new finite element method will prove to be useful not only in the areas of computer graphics and geometric design, but also in engineering analysis.
- With the aid of the modified butterfly subdivision technique, we create a surface model that incorporates mass and damping distributions, internal deformation energy, forces, and other physical quantities. We also systematically formulate the motion equations for this (modified) butterfly subdivision surface whose degrees of freedom are the collection of initial user-specified control vertices. Therefore, the advantages of both the physics-based modeling philosophy and the geometric subdivision schemes are assimilated within a single unified framework.
- Users will be able to manipulate this physics-based model in an arbitrary region and the model responds naturally (just like the real-world object) to various force distributions. The shape deformation is quantitatively characterized by the time-varying

displacement of control points that uniquely define the geometry of the limit surface.

- We develop algorithms and procedures which approximate our novel finite elements using a collection of linear and/or bilinear finite elements subject to the implicit geometric constraints enforced by the butterfly subdivision rules. This hierarchically structured approximation is capable of achieving any user-specified error tolerance.

Although the long-term goals inherent in the previously-developed dynamic Catmull-Clark model [46], [47] are the same as in our current endeavor of deriving the dynamic modified butterfly subdivision scheme, the research presented in this paper achieves them in a much more elegant fashion. First, the finite element implementation of the dynamic Catmull-Clark subdivision surface is specific to the subdivision technique involved where a diversity of complicated finite elements must be employed in order to account for the special cases and cannot be readily generalized to other approximating subdivision schemes in a straightforward way. However, the finite element techniques developed in this paper can be easily generalized to other interpolatory (as well as approximating) subdivision schemes involving triangular (or n-sided) meshes. Second, for some specific cases, the thin-plate energy of Catmull-Clark subdivision surface diverges as shown in [7], hence various case analysis needs to be performed to derive the internal energy of the dynamic Catmull-Clark subdivision surface model. In contrast, the internal deformation energy of the dynamic butterfly scheme can be derived in an unified fashion for a variety of scenarios. Third, for both models, we need to derive the closest point on the limit surface from a given point in 3D for force applications. The calculation overhead involved in this process is significantly less for the butterfly case as it is an interpolatory scheme where all vertices at various levels of subdivision lie on the limit surface and the search space can be reduced rapidly in a hierarchical fashion. The situation is quite different for Catmull-Clark subdivision scheme since it is an approximating scheme and the technique used for finding the closest model point [55] is computationally expensive. Note that, even though the limiting points of the control vertices can be derived using the approach proposed by Halstead et al. [7], it produces significant overhead to compute these limiting positions for nearest point computations each time the model is updated. The force application is vital to any physics-based model and, hence, the adopted computationally inexpensive method to find the closest point for the proposed scheme in this paper has very significant advantages over the previous model. Finally, it has been empirically found that the recovered shape is more compact (fewer number of degrees of freedom) when using the proposed model in comparison with our earlier reported model for the same data sets and model-fitting criteria.

5 PHYSICS-BASED FORMULATION

We shall provide a systematic formulation of the dynamic subdivision surface model. First, we briefly review the

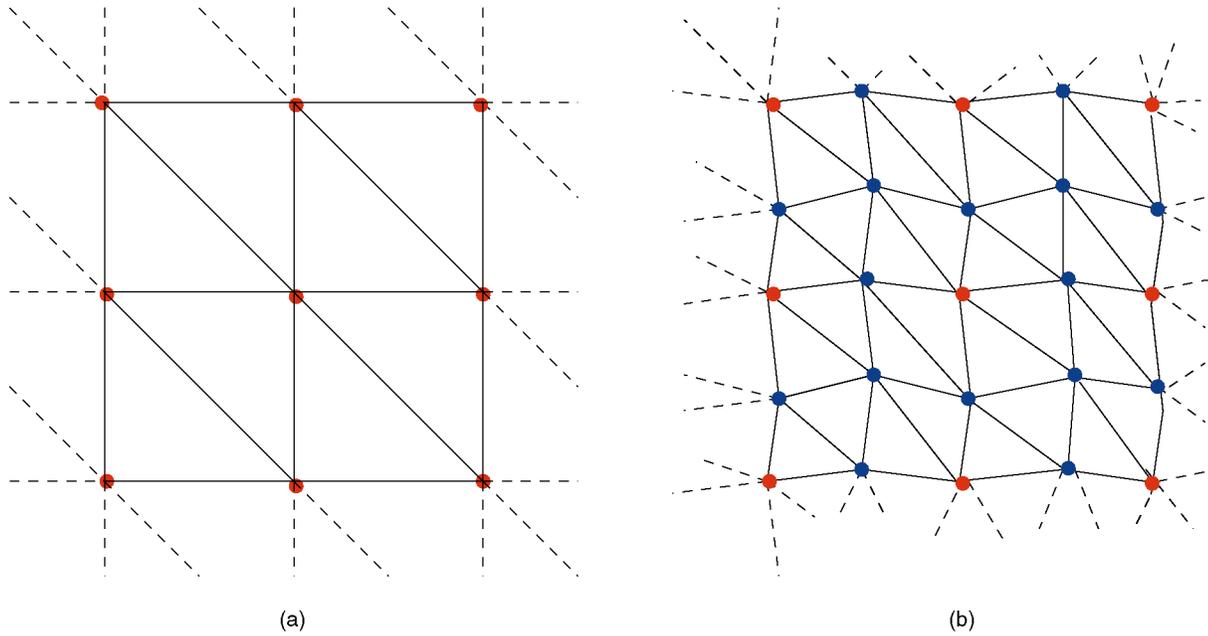


Fig. 1. (a) The control polygon with triangular faces; (b) mesh obtained after one subdivision step.

modified butterfly subdivision scheme. Next, we introduce a local parameterization scheme which will facilitate the formulation of the smooth limit surface as a function of the control point positions defining the initial mesh. This parameterization scheme is then used to derive the dynamic model. Note that, these techniques can be generalized to define and construct a generic dynamic framework for other triangle-based subdivision surface schemes as well.

5.1 Butterfly Subdivision Geometry

The butterfly subdivision scheme [11] starts with an initial triangular mesh which is also known as the control mesh. The vertices of the control mesh are known as the control points. 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. 1b is obtained by subdividing the initial mesh shown in Fig. 1a once. Note that, all the newly introduced vertices (marked in blue) corresponding to the edges in the original mesh have valence (degree) 6, whereas the position and valence of the original vertices (marked in red) do not change in the subdivided mesh.

In the original butterfly scheme, the new vertices corresponding to the edges in the previous level are obtained using an eight-point stencil, as shown in Fig. 2a. The name of the scheme originated from the “butterfly”-like configuration of the contributing vertices. The weighing factors for different contributing vertex positions are shown in Fig. 2b. The vertex e_{12}^{j+1} in the $j + 1$ th level of subdivision, corresponding to the edge connecting vertices v_1^j and v_2^j at level j is obtained by

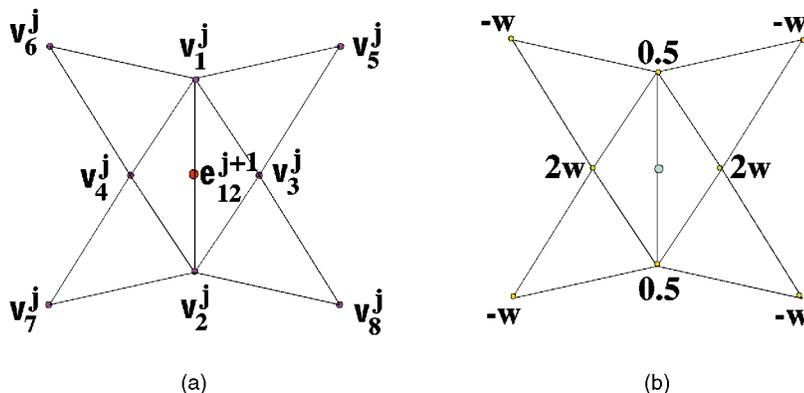


Fig. 2. (a) The contributing vertices in the j th level for the vertex in the $j + 1$ th level corresponding to the edge between v_1^j and v_2^j ; (b) the weighing factors for different vertices.

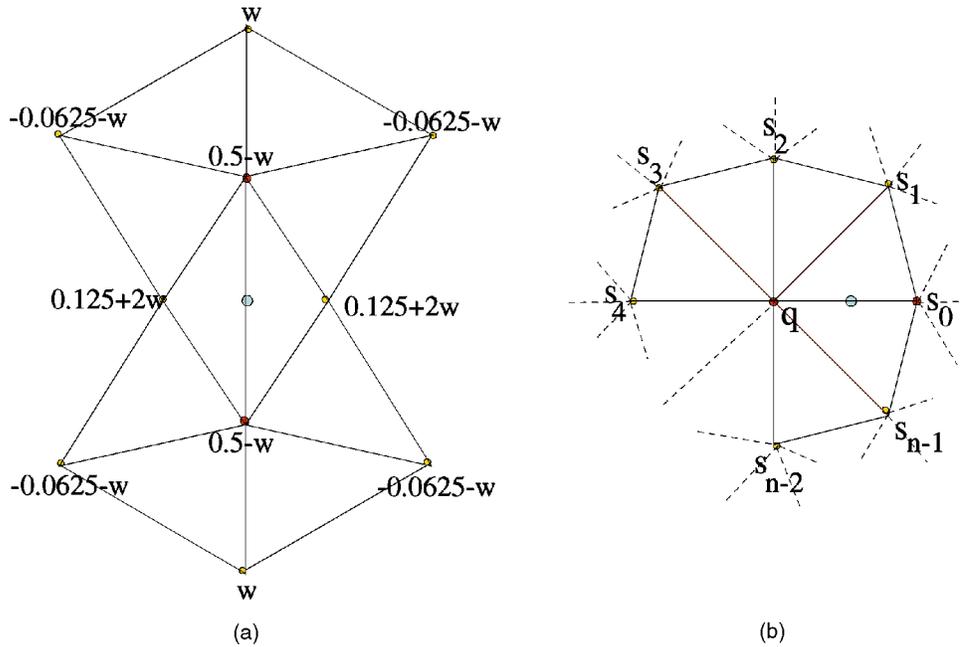


Fig. 3. (a) The weighing factors of contributing vertex positions for an edge connecting two vertices of valence 6; (b) the corresponding case when one vertex is of valence n and the other is of valence 6.

$$\mathbf{e}_{12}^{j+1} = 0.5(\mathbf{v}_1^j + \mathbf{v}_2^j) + 2w(\mathbf{v}_3^j + \mathbf{v}_4^j) - w(\mathbf{v}_5^j + \mathbf{v}_6^j + \mathbf{v}_7^j + \mathbf{v}_8^j), \quad (1)$$

where $0 \leq w \leq 1$, and \mathbf{v}_i^j denotes the position of the i th vertex at the j th level.

The butterfly subdivision scheme produces a smooth C^1 surface in the limit except at the *extraordinary* points corresponding to the *extraordinary* vertices (vertices with valence not equal to 6) in the initial mesh [13]. All the vertices introduced through subdivision have valence 6 and, therefore, the number of extraordinary points in the smooth limit surface equals the number of extraordinary vertices in the initial mesh. Recently, this scheme has been modified by Zorin et al. [13] to obtain better smoothness properties at the extraordinary points. In Zorin et al. [13], all the edges have been categorized into three classes: 1) edges connecting two vertices of valence 6 (a 10 point stencil, as shown in Fig. 3a, is used to obtain the new vertex positions corresponding to these edges), 2) edges connecting a vertex of valence 6 and a vertex of valence $n \neq 6$ (the corresponding stencil to obtain new vertex position is shown in Fig. 3b, where $q = .75$ is the weight associated with the vertex of valence $n \neq 6$ and

$$s_i = (0.25 + \cos(2\pi i/n) + 0.5\cos(4\pi i/n))/n, \\ i = 0, 1, \dots, n-1,$$

are the weights associated with the vertices of valence 6), and 3) edges connecting two vertices of valence $n \neq 6$. The last case cannot occur except in the initial mesh as all the newly introduced vertices are of valence 6 and the new vertex position in this last case is obtained by averaging the positions obtained through the use of stencil shown in Fig. 3b at each of those two extraordinary vertices.

5.2 Parameterization of the Limit Surface

Often, the smooth limit surface defined by the recursive subdivision process is of arbitrary topology where a globally planar parameterization may not be possible. Nevertheless, we can locally parameterize the limit surface over the domain defined by the initial mesh following an approach similar to the one described in Lounsbery et al. [62]. The idea is to track any arbitrary point on the initial mesh across the meshes obtained via the subdivision process (see Fig. 4 and Fig. 5) 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 mesh consisting of a set of triangular faces. The recursive application of the subdivision rules smooths out each triangular face and, in the limit, a smooth surface is obtained which can also be considered as a collection of smooth triangular patches. The subdivision process and the triangular decomposition of the limit surface is depicted in Fig. 4. 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, we can express the limit surface s as

$$\mathbf{s} = \sum_{k=1}^n \mathbf{s}_k, \quad (2)$$

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

The parameterization process is best explained through the following example: We choose a simple planar mesh, shown in Fig. 5a, as the initial mesh. An arbitrary point \mathbf{x} inside the triangular face abc is tracked over the meshes obtained through subdivision. The vertices in the initial mesh are drawn in black in Fig. 5. After one step of

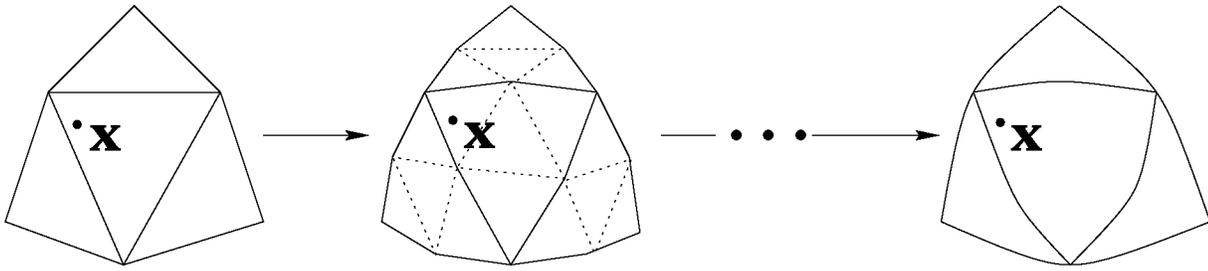


Fig. 4. The smoothing effect of the subdivision process on the triangles of the initial mesh.

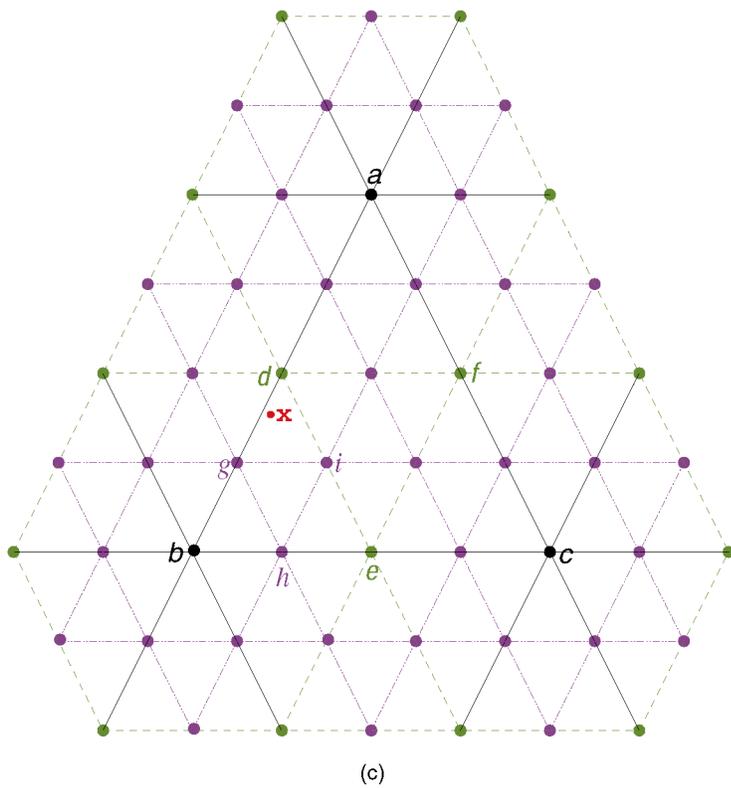
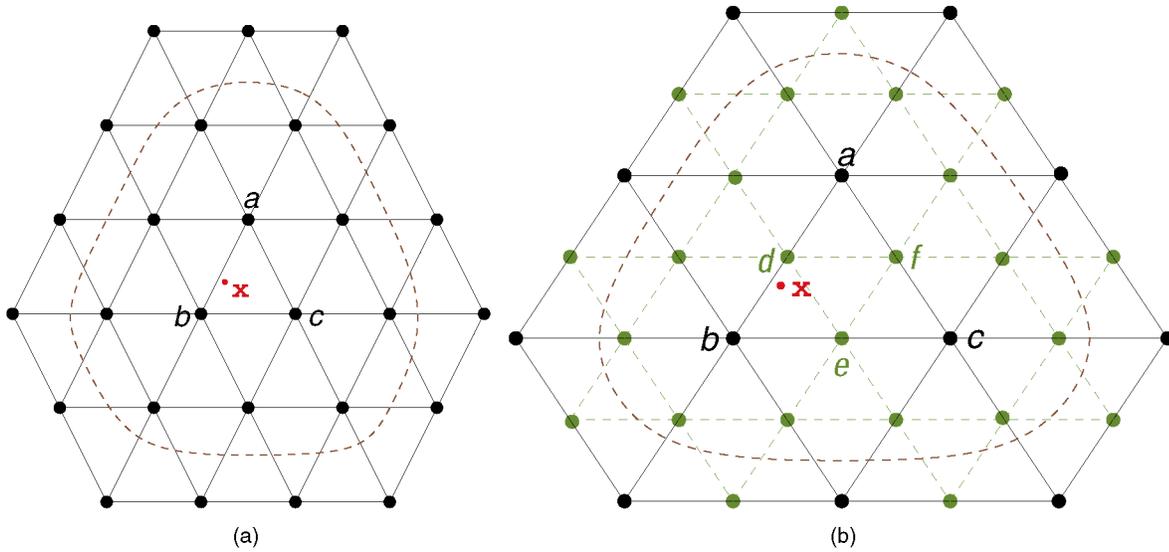


Fig. 5. Tracking a point x through various levels of subdivision: (a) initial mesh, (b) the selected section (enclosed by dotted lines) of the mesh in (a) after one subdivision step, (c) the selected section of the mesh in (b) after another subdivision step.

subdivision, the initial mesh is refined by addition of new vertices, which are colored green. Another subdivision step on this refined mesh leads to a finer mesh with the introduction of magenta-colored new vertices. Note 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 \mathbf{a} , \mathbf{b} , and \mathbf{c} due to the local nature of the subdivision process.* For example, the vertex \mathbf{d} , introduced after the first subdivision step, can be obtained using the 10-point stencil shown in Fig. 3a on the edge ab . All the contributing vertices in the initial mesh are within the 1-neighborhood of the vertices \mathbf{a} and \mathbf{b} . A 10-point stencil can be used again in the next subdivision step on the edge db to obtain the vertex \mathbf{g} . Some of the contributing vertices at this level of subdivision, for example, the (green-colored) 1-neighbors of the vertex \mathbf{b} (except \mathbf{d} and \mathbf{e}) in Fig. 5b depend on some vertices in the initial mesh which are within the 2-neighborhood of the vertices \mathbf{a} , \mathbf{b} , and \mathbf{c} in the initial mesh.

In the rest of the discussion, superscripts are used to indicate the subdivision level. For example, \mathbf{v}_{uvw}^j 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 j th level of subdivision. Let \mathbf{v}_{abc}^0 be the collection of vertices in the initial mesh which are within the 2-neighborhood of the vertices \mathbf{a} , \mathbf{b} , and \mathbf{c} (marked black in Fig. 5a). Let the number of such vertices be r . Then, the vector \mathbf{v}_{abc}^0 , which is the concatenation of the (x, y, z) positions for all the r vertices, is of dimension $3r$. These r vertices control the smooth triangular patch in the limit surface corresponding to the triangular face abc in the initial mesh. Now, there exist four $(3r \times 3r)$ subdivision matrices $(\mathbf{A}_{abc})_t$, $(\mathbf{A}_{abc})_l$, $(\mathbf{A}_{abc})_r$, and $(\mathbf{A}_{abc})_m$ such that:

$$\begin{aligned} \mathbf{v}_{adf}^1 &= (\mathbf{A}_{abc})_t \mathbf{v}_{abc}^0, \\ \mathbf{v}_{bed}^1 &= (\mathbf{A}_{abc})_l \mathbf{v}_{abc}^0, \\ \mathbf{v}_{cfe}^1 &= (\mathbf{A}_{abc})_r \mathbf{v}_{abc}^0, \\ \mathbf{v}_{def}^1 &= (\mathbf{A}_{abc})_m \mathbf{v}_{abc}^0, \end{aligned} \quad (3)$$

where the subscripts t , l , r , and m denote top, left, right, and middle triangle positions, respectively (indicating the relative position of the *new* triangle with respect to the *original* triangle), and \mathbf{v}_{adf}^1 , \mathbf{v}_{bed}^1 , \mathbf{v}_{cfe}^1 , and \mathbf{v}_{def}^1 are the concatenation of the (x, y, z) positions for the vertices in the 2-neighborhood of the corresponding triangle in the newly obtained subdivided mesh. The new vertices in this level of subdivision are marked green in Fig. 5b. 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 (3) are the same. This concept is further illustrated in Fig. 6.

Carrying out one more level of subdivision, along with the old vertices, we get a new set of vertices, which are marked in magenta in Fig. 5c. Adopting a similar approach as in the derivation of (3), we obtain

$$\begin{aligned} \mathbf{v}_{dgi}^2 &= (\mathbf{A}_{bed})_t \mathbf{v}_{bed}^1 \\ \mathbf{v}_{bhg}^2 &= (\mathbf{A}_{bed})_l \mathbf{v}_{bed}^1 \\ \mathbf{v}_{eih}^2 &= (\mathbf{A}_{bed})_r \mathbf{v}_{bed}^1 \\ \mathbf{v}_{ghi}^2 &= (\mathbf{A}_{bed})_m \mathbf{v}_{bed}^1. \end{aligned} \quad (4)$$

The relative position of the triangular face dgi in Fig. 5c with respect to the triangular face bed is topologically the same as of the triangular face adf in Fig. 5b with respect to the triangular face abc . Therefore, we can write $(\mathbf{A}_{bed})_t = (\mathbf{A}_{abc})_t$. Using similar reasoning, (4) can be rewritten as

$$\begin{aligned} \mathbf{v}_{dgi}^2 &= (\mathbf{A}_{bed})_t \mathbf{v}_{bed}^1 = (\mathbf{A}_{abc})_t \mathbf{v}_{bed}^1 \\ \mathbf{v}_{bhg}^2 &= (\mathbf{A}_{bed})_l \mathbf{v}_{bed}^1 = (\mathbf{A}_{abc})_l \mathbf{v}_{bed}^1 \\ \mathbf{v}_{eih}^2 &= (\mathbf{A}_{bed})_r \mathbf{v}_{bed}^1 = (\mathbf{A}_{abc})_r \mathbf{v}_{bed}^1 \\ \mathbf{v}_{ghi}^2 &= (\mathbf{A}_{bed})_m \mathbf{v}_{bed}^1 = (\mathbf{A}_{abc})_m \mathbf{v}_{bed}^1. \end{aligned} \quad (5)$$

Combining (3) and (5), we get

$$\begin{aligned} \mathbf{v}_{dgi}^2 &= (\mathbf{A}_{abc})_t (\mathbf{A}_{abc})_l \mathbf{v}_{abc}^0, \\ \mathbf{v}_{bhg}^2 &= (\mathbf{A}_{abc})_l (\mathbf{A}_{abc})_l \mathbf{v}_{abc}^0, \\ \mathbf{v}_{eih}^2 &= (\mathbf{A}_{abc})_r (\mathbf{A}_{abc})_l \mathbf{v}_{abc}^0, \\ \mathbf{v}_{ghi}^2 &= (\mathbf{A}_{abc})_m (\mathbf{A}_{abc})_l \mathbf{v}_{abc}^0. \end{aligned} \quad (6)$$

Let \mathbf{x} be a point with barycentric coordinates $(\alpha_{abc}^0, \beta_{abc}^0, \gamma_{abc}^0)$ inside the triangular face abc . When the initial mesh is subdivided, \mathbf{x} becomes a point inside the triangular face bed with barycentric coordinates $(\alpha_{bed}^1, \beta_{bed}^1, \gamma_{bed}^1)$. Another level of subdivision causes \mathbf{x} to be included in the triangular face dgi with barycentric coordinates $(\alpha_{dgi}^2, \beta_{dgi}^2, \gamma_{dgi}^2)$. Let \mathbf{s}_{abc}^j 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}_{abc}^0 can be written as

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

where the subscripts x , y , and z indicate the x , y , and z coordinates, respectively, of the corresponding vertex position. The expressions for \mathbf{v}_{bed}^1 and \mathbf{v}_{dgi}^2 can also be written in a similar manner. Next, we construct the matrix \mathbf{B}_{abc}^0 as follows:

$$\mathbf{B}_{abc}^0(\mathbf{x}) = \begin{bmatrix} \overbrace{\alpha_{abc}^0, \beta_{abc}^0, \gamma_{abc}^0, 0, \dots, 0}^r, \overbrace{0, \dots, 0}^r, \overbrace{0, \dots, 0}^r \\ \overbrace{0, \dots, 0}^r, \overbrace{\alpha_{abc}^0, \beta_{abc}^0, \gamma_{abc}^0, 0, \dots, 0}^r, \overbrace{0, \dots, 0}^r \\ \overbrace{0, \dots, 0}^r, \overbrace{0, \dots, 0}^r, \overbrace{\alpha_{abc}^0, \beta_{abc}^0, \gamma_{abc}^0, 0, \dots, 0}^r \end{bmatrix}.$$

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

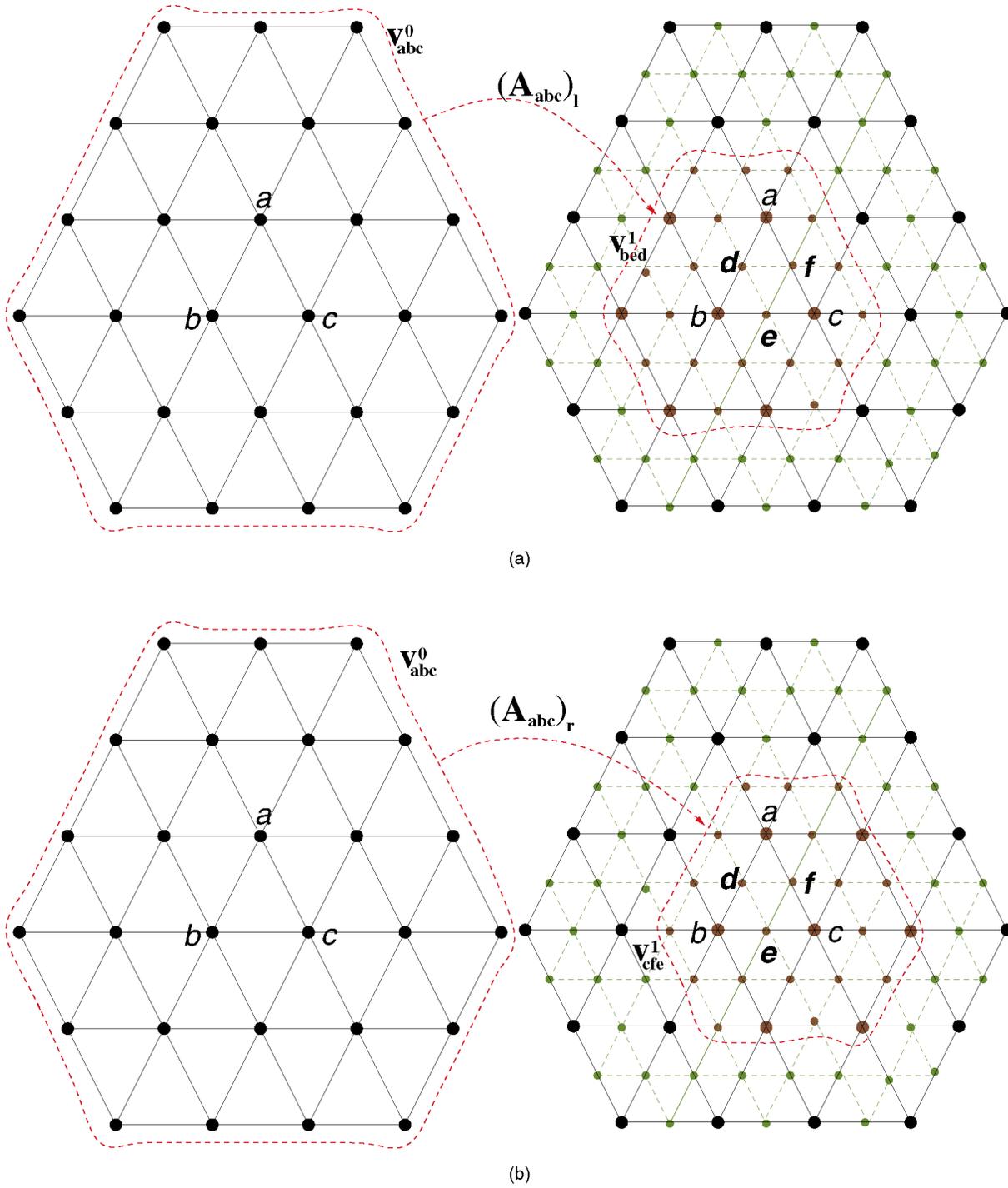


Fig. 6. Different set of control points at a subdivided level obtained by applying different subdivision matrices on a given set of control points in a coarser mesh.

$$\begin{aligned}
 s_{abc}^0(\mathbf{x}) &= \mathbf{B}_{abc}^0(\mathbf{x})\mathbf{v}_{abc}^0, \\
 s_{abc}^1(\mathbf{x}) &= \mathbf{B}_{bed}^1(\mathbf{x})\mathbf{v}_{bed}^1 = \mathbf{B}_{bed}^1(\mathbf{x})(\mathbf{A}_{abc})_l\mathbf{v}_{abc}^0, \\
 s_{abc}^2(\mathbf{x}) &= \mathbf{B}_{dgi}^2(\mathbf{x})\mathbf{v}_{dgi}^2 = \mathbf{B}_{dgi}^2(\mathbf{x})(\mathbf{A}_{abc})_t\mathbf{v}_{bed}^1 \\
 &= \mathbf{B}_{dgi}^2(\mathbf{x})(\mathbf{A}_{abc})_t(\mathbf{A}_{abc})_l\mathbf{v}_{abc}^0.
 \end{aligned} \tag{7}$$

$$\begin{aligned}
 s_{abc}^j(\mathbf{x}) &= \mathbf{B}_{uvw}^j(\mathbf{x}) \overbrace{(\mathbf{A}_{abc})_m \dots (\mathbf{A}_{abc})_t (\mathbf{A}_{abc})_l}^j \mathbf{v}_{abc}^0 \\
 &= \mathbf{B}_{uvw}^j(\mathbf{x})(\mathbf{A}_{abc}^j)\mathbf{v}_{abc}^0 \\
 &= \mathbf{B}_{abc}^j(\mathbf{x})\mathbf{v}_{abc}^0,
 \end{aligned} \tag{8}$$

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

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), $(\mathbf{A}_{abc}^j) = (\mathbf{A}_{abc})_m \dots (\mathbf{A}_{abc})_t (\mathbf{A}_{abc})_l$ and

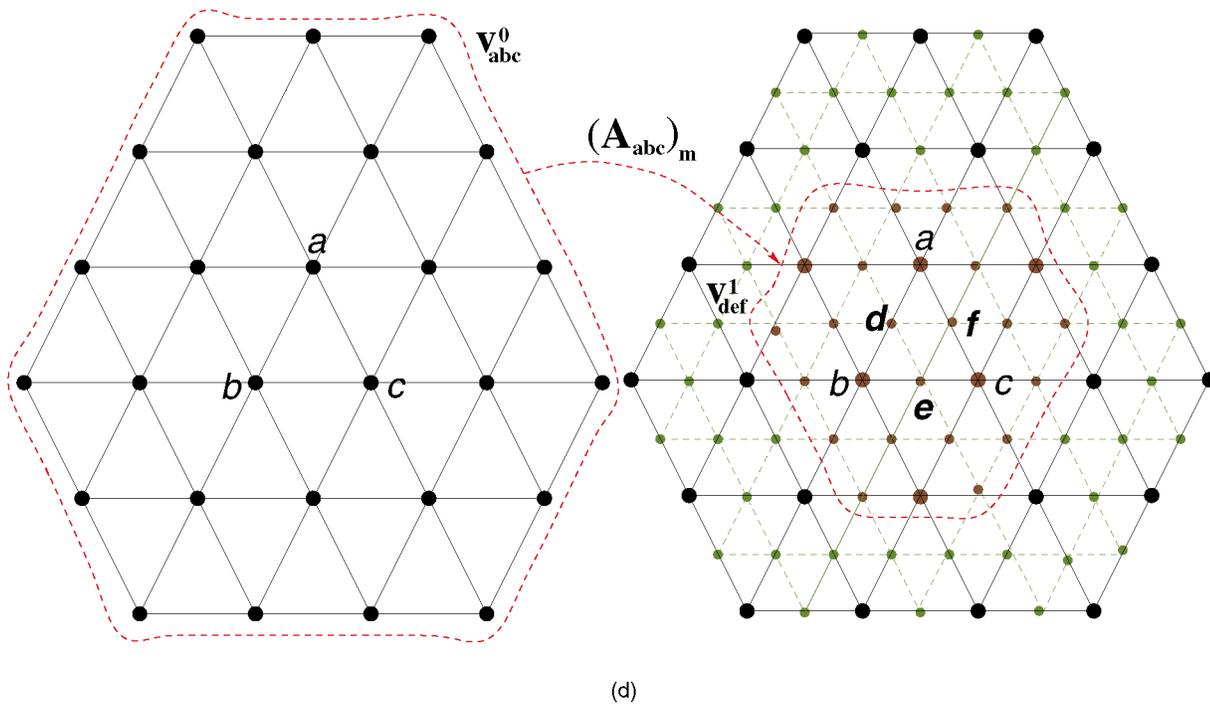
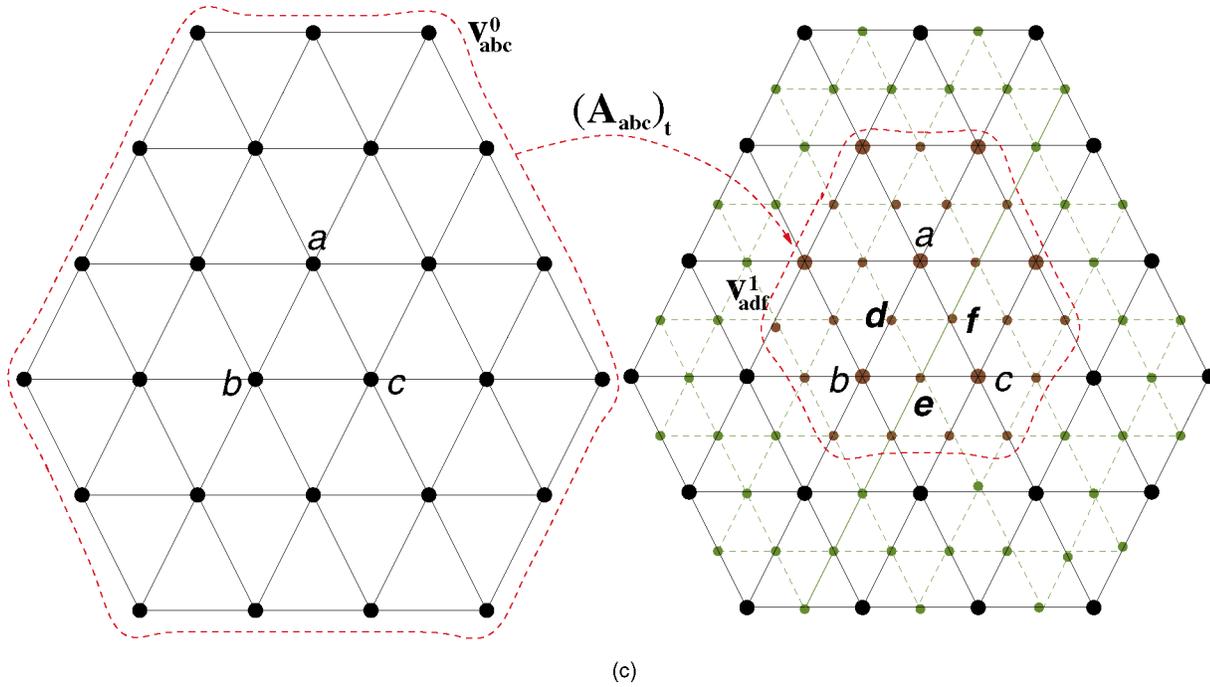


Fig. 6. Continued.

$\mathbf{B}_{abc}^j(\mathbf{x}) = \mathbf{B}_{uvw}^j(\mathbf{x})(\mathbf{A}_{abc}^j)$. Note that the sequence of applying $(\mathbf{A}_{abc})_{tr}$, $(\mathbf{A}_{abc})_{lr}$, $(\mathbf{A}_{abc})_{rr}$, and $(\mathbf{A}_{abc})_m$ depends on the triangle inside which the tracked point \mathbf{x} falls after each subdivision step. Finally, we complete the local parameterization process and obtain

$$\mathbf{s}_{abc}(\mathbf{x}) = \left(\lim_{j \rightarrow \infty} \mathbf{B}_{abc}^j(\mathbf{x}) \right) \mathbf{v}_{abc}^0 = \mathbf{B}_{abc}(\mathbf{x}) \mathbf{v}_{abc}^0, \quad (9)$$

where \mathbf{B}_{abc} is the collection of basis functions at the vertices of \mathbf{v}_{abc}^0 . Note 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 $(\mathbf{A}_{abc})_{tr}$, $(\mathbf{A}_{abc})_{lr}$, $(\mathbf{A}_{abc})_{rr}$, and $(\mathbf{A}_{abc})_m$ sums to one. The largest eigenvalue of such matrices is 1 and, therefore, the limit in (9) exists. Now, if we assume that the triangular face abc is the k th face in the initial mesh, then (9) can be rewritten as

$$\mathbf{s}_k(\mathbf{x}) = \mathbf{B}_k(\mathbf{x}) \mathbf{v}_k^0 = \mathbf{B}_k(\mathbf{x}) \mathbf{A}_k \mathbf{p}, \quad (10)$$

where \mathbf{p} is the concatenation of the (x, y, z) positions of all the vertices in the initial mesh and the matrix \mathbf{A}_k , when postmultiplied by \mathbf{p} , selects the vertices \mathbf{v}_k^0 controlling 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 $3t$, \mathbf{A}_k is a $(3r \times 3t)$ matrix, and $\mathbf{B}_k(\mathbf{x})$ is a $(3 \times 3r)$ matrix.

Combining (2) and (10), we get

$$\mathbf{s}(\mathbf{x}) = \left(\sum_{k=1}^n \mathbf{B}_k(\mathbf{x}) \mathbf{A}_k \right) \mathbf{p}; = \mathbf{J}(\mathbf{x}) \mathbf{p}, \quad (11)$$

where the $(3 \times 3t)$ matrix \mathbf{J} is the collection of basis functions for the corresponding vertices in the initial mesh. The basis function (at various resolutions) corresponding to a vertex of degree 5 is shown in Fig. 7. The vector \mathbf{p} is also known as the degrees of freedom (DOF) vector of the smooth limit surface \mathbf{s} .

5.3 Kinematics and Dynamics

We now treat the vertex positions in the initial mesh defining the smooth limit surface \mathbf{s} as a function of time in order to embed the modified butterfly subdivision model in a dynamic framework. The velocity of this surface model can be expressed as

$$\dot{\mathbf{s}}(\mathbf{x}, \mathbf{p}) = \mathbf{J}(\mathbf{x}) \dot{\mathbf{p}}, \quad (12)$$

where an overstruck dot denotes a time derivative and $\mathbf{x} \in S^0$, S^0 being the domain defined by the initial mesh. The physics of the dynamic subdivision surface model is based on the work-energy relationship of Lagrangian dynamics [63] and is formulated in an analogous way to that in [46], [47].

In an abstract physical system, let $p_i(t)$ be a set of generalized coordinates which are functions of time and are assembled into the vector \mathbf{p} . Let $f_i(t)$ be the generalized force assembled into the vector \mathbf{f}_p and acting on p_i . Then, the Lagrangian equation of motion can be expressed as

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{p}_i} - \frac{\partial T}{\partial p_i} + \frac{\partial F}{\partial \dot{p}_i} + \frac{\partial U}{\partial p_i} = f_i, \quad (13)$$

where T , F , and U are the kinetic, dissipation, and potential energy, respectively.

Let μ be the mass density function of the surface. Then, the kinetic energy of the surface is

$$T = \frac{1}{2} \int_{\mathbf{x} \in S^0} \mu(\mathbf{x}) \dot{\mathbf{s}}^T(\mathbf{x}) \dot{\mathbf{s}}(\mathbf{x}) d\mathbf{x} = \frac{1}{2} \dot{\mathbf{p}}^T \mathbf{M} \dot{\mathbf{p}}, \quad (14)$$

where (using (12)) $\mathbf{M} = \int_{\mathbf{x} \in S^0} \mu(\mathbf{x}) \mathbf{J}^T(\mathbf{x}) \mathbf{J}(\mathbf{x}) d\mathbf{x}$ is the $(3t \times 3t)$ mass matrix. Similarly, let γ be the damping density function of the surface. The dissipation energy is

$$F = \frac{1}{2} \int_{\mathbf{x} \in S^0} \gamma(\mathbf{x}) \dot{\mathbf{s}}^T(\mathbf{x}) \dot{\mathbf{s}}(\mathbf{x}) d\mathbf{x} = \frac{1}{2} \dot{\mathbf{p}}^T \mathbf{D} \dot{\mathbf{p}}, \quad (15)$$

where $\mathbf{D} = \int_{\mathbf{x} \in S^0} \gamma(\mathbf{x}) \mathbf{J}^T(\mathbf{x}) \mathbf{J}(\mathbf{x}) d\mathbf{x}$ is the $(3t \times 3t)$ damping matrix. The potential energy of the smooth limit surface can be expressed as

$$U = \frac{1}{2} \mathbf{p}^T \mathbf{K} \mathbf{p}, \quad (16)$$

where the $(3t \times 3t)$ stiffness matrix \mathbf{K} is obtained by assigning various internal energies to a discretized approximation of the limit surface and is detailed in Section 6.

Using the expressions for the kinetic, dissipation, and potential energy in (13), we get the motion equation given by

$$\mathbf{M} \ddot{\mathbf{p}} + \mathbf{D} \dot{\mathbf{p}} + \mathbf{K} \mathbf{p} = \mathbf{f}_p. \quad (17)$$

The generalized force vector \mathbf{f}_p , which can be obtained through the principle of virtual work [63], is expressed as

$$\mathbf{f}_p = \int_{\mathbf{x} \in S^0} \mathbf{J}^T(\mathbf{x}) \mathbf{f}(\mathbf{x}, t) d\mathbf{x}. \quad (18)$$

We can apply various types of forces on the smooth limit surface and the limit surface would evolve over time, according to (17), to obtain an equilibrium position characterized by a minimum of the total model energy.

It may be noted that our newly developed butterfly finite elements are different from conventional finite elements, which are primarily founded upon polynomials. In addition, our finite element analysis and synthesis techniques are different from the traditional finite element method in which the unknowns are displacement distribution with respect to the initial undeformed shape. However, consider $\mathbf{d}(\mathbf{x}, \mathbf{p})$ as a companion displacement function over the limit geometry $\mathbf{s}(\mathbf{x}, \mathbf{p})$, all the finite element procedures explained in this paper will be amenable to the conventional finite element analysis for mechanical engineering applications without extra difficulties. Therefore, our formulation and accompanying algorithms are useful for geometric design, dynamic sculpting, as well as traditional analysis on mechanical properties.

5.4 Hierarchical Structure and Multilevel Dynamics

We have developed a dynamic framework where the smooth limit surface evolves over time in response to the applied forces. The entire process can be described as follows: Given an initial mesh, a smooth surface is obtained in the limit. Users can directly apply synthesized forces to this smooth limit surface to enforce various functional and aesthetic constraints. This direct manipulation is then transferred back as virtual forces acting on the initial mesh through a transformation matrix (18) and the initial mesh (as well as the underlying smooth limit surface) deforms continuously over time until an equilibrium position is obtained. The deformation of the surface in response to the applied forces is governed by the motion equation (17). Within our physics-based modeling framework, the limit surface evolves as a consequence of the evolution of the initial mesh. One can apply various types of forces on the limit surface to obtain a desired effect, but the possible level of details appearing in a shape that can be obtained through evolution is constrained by the number of control vertices in the initial mesh. It might be necessary to increase the number of control vertices in the initial mesh in order to obtain a detailed shape through this evolution process.

The number of control vertices defining the same smooth limit surface can be increased by simply replacing the initial mesh with a mesh obtained after one subdivision step applied to the initial mesh. This new mesh has a greater

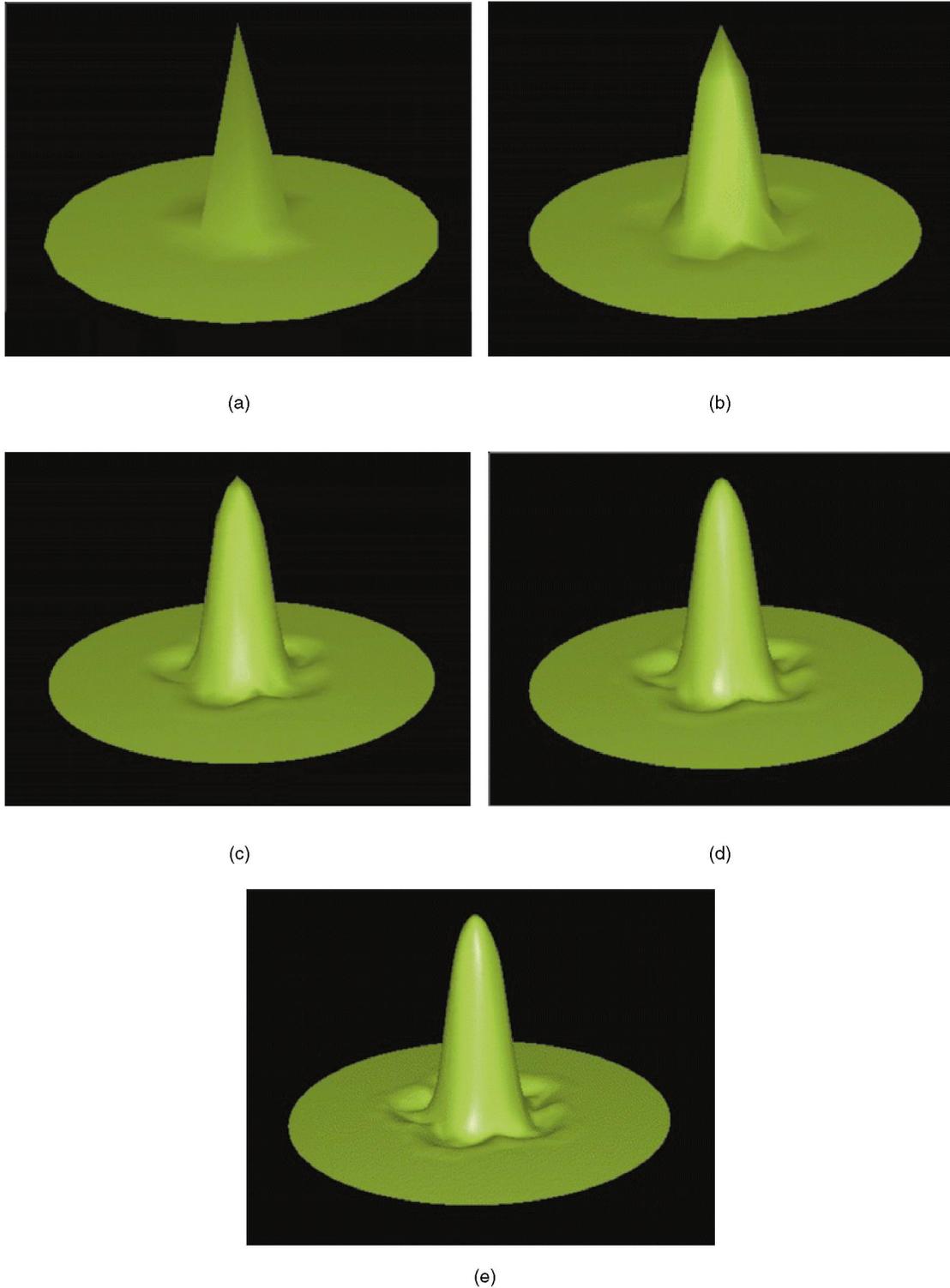


Fig. 7. The butterfly basis function corresponding to a control vertex of degree 5 at various resolution: (a) level 0, (b) level 1, (c) level 2, (d) level 3, and (e) level 4.

number of vertices, but defines the same limit surface. For example, after one step of modified butterfly subdivision, the initial degrees of freedom \mathbf{p} (refer to (11) and (12)) in the dynamic system will be replaced by a larger number of degrees of freedom \mathbf{q} , where $\mathbf{q} = \mathbf{A}\mathbf{p}$ (and, therefore, $\dot{\mathbf{q}} = \mathbf{A}\dot{\mathbf{p}}$). \mathbf{A} is a global subdivision matrix of size

$(3s \times 3t)$ whose entries are uniquely determined by the weights used in the modified butterfly subdivision scheme (see Section 5.1 for the weights). Thus, \mathbf{p} , expressed as a function of \mathbf{q} , can be written as

$$\mathbf{p} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{q} = \mathbf{A}^\dagger \mathbf{q}, \quad (19)$$

where $\mathbf{A}^\dagger = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$. Therefore, we can rewrite (11) and (12) as

$$\mathbf{s}(\mathbf{x}) = (\mathbf{J}(\mathbf{x}) \mathbf{A}^\dagger) \mathbf{q} \quad (20)$$

and

$$\dot{\mathbf{s}}(\mathbf{x}, \mathbf{q}) = (\mathbf{J}(\mathbf{x}) \mathbf{A}^\dagger) \dot{\mathbf{q}}, \quad (21)$$

respectively. Note that the new transformation matrix $(\mathbf{J}(\mathbf{x}) \mathbf{A}^\dagger)$ can be computed directly from \mathbf{q} as well with the help of new associated parameterization because the new control mesh with more degrees of freedom produces the same limit surface. Now, we can easily derive the equation of motion for this new subdivided model involving a larger number of control vertices, namely \mathbf{q} . We need to recompute the mass, damping, and stiffness matrices for this “finer” level. The structure of the motion equation as given by (17) remains unchanged, but the dimensionality and the entries of \mathbf{M} , \mathbf{D} , \mathbf{K} , \mathbf{p} and \mathbf{f}_p change correspondingly in this newly obtained subdivided level. In particular, the motion equation, explicitly expressed as a function of \mathbf{q} , can be written as

$$\mathbf{M}_q \ddot{\mathbf{q}} + \mathbf{D}_q \dot{\mathbf{q}} + \mathbf{K}_q \mathbf{q} = \mathbf{f}_q, \quad (22)$$

where $\mathbf{M}_q = \int_{\mathbf{x} \in S^1} \mu(\mathbf{x}) (\mathbf{A}^\dagger)^T \mathbf{J}^T(\mathbf{x}) \mathbf{J}(\mathbf{x}) \mathbf{A}^\dagger d\mathbf{x}$, S^1 being the domain defined by the newly obtained subdivided mesh. The derivation of \mathbf{D}_q , \mathbf{K}_q , and \mathbf{f}_q follow suit.

Note that further increase in the number of control vertices, if necessary, can be obtained via another level of subdivision. Therefore, multilevel dynamics is achieved through recursive subdivision on the initial set of control vertices. Users can interactively choose any subdivided mesh as the control mesh for the dynamic model, depending on their needs. Alternatively, the system can automatically determine the most suitable control mesh for certain applications based on an application-specific criteria.

6 FINITE ELEMENT IMPLEMENTATION

In this section, we describe the implementation of this physical model using the finite element method. Previously, in Section 5, we pointed out that the smooth limit surface obtained by the recursive application of the modified butterfly subdivision rules can be represented by a set of smooth triangular patches, each of which is represented by a finite element. The shape (basis) function of this finite element is obtained by smoothing a hat function through repeated application of the modified butterfly subdivision rules. The number of finite elements in the smooth limit surface is equal to the number of triangular faces in the initial mesh as mentioned earlier (refer to Fig. 4 and Fig. 8). We now provide a detailed discussion on how to derive the mass, damping, and stiffness matrices for these elements. These elemental matrices can be assembled to obtain the global physical matrices \mathbf{M} , \mathbf{D} , and \mathbf{K} , and a numerical solution to the governing second-order differential equation as given by (17) can be obtained using finite element analysis techniques [64]. We use the same example as in Section 5 (refer to Fig. 5) to develop the related concepts. The concept of elements, along with the control vertices and

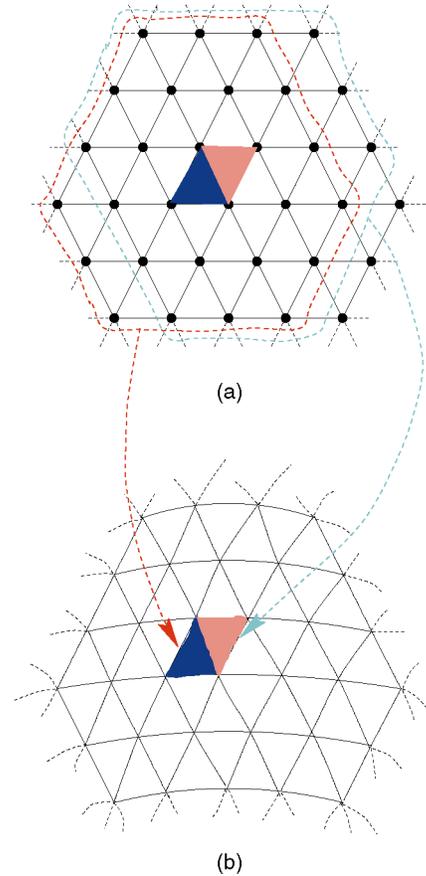


Fig. 8. (a) An initial mesh and (b) the corresponding limit surface. The domains of the shaded elements in the limit surface are the corresponding triangular faces in the initial mesh. The encircled vertices in (a) are the degrees of freedom for the corresponding element.

their corresponding domains, is further illustrated in Fig. 8. We will now show how to derive the mass, damping and stiffness matrices for the element corresponding to the triangular face abc in Fig. 5. The derivations hold for any element in general.

6.1 Elemental Mass and Damping Matrices

The mass matrix for the element given by \mathbf{s}_{abc} (9) can be written as

$$\mathbf{M}_{abc} = \int_{\mathbf{x} \in \mathbf{s}_{abc}} \mu(\mathbf{x}) \{\mathbf{B}_{abc}(\mathbf{x})\}^T \{\mathbf{B}_{abc}(\mathbf{x})\} d\mathbf{x}. \quad (23)$$

However, from (9), we know that the basis functions corresponding to the vertices in \mathbf{v}_{abc}^0 which are stored as entries in \mathbf{B}_{abc} are obtained as a limiting process. These basis functions do not have any analytic form, hence, computing the inner product of such basis functions as needed in (23) is a challenging problem. In [62], an outline is provided on the computation of these inner products without performing any integration. In this paper, we develop a different and simpler approach to solve this problem. The smooth triangular patch in the limit surface corresponding to the face abc in the initial mesh is approximated 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

four triangular faces). Then, the mass matrix can be expressed as

$$\mathbf{M}_{abc} = \sum_{i=1}^{4^j} \int_{\mathbf{x} \in \Delta_i} \mu(\mathbf{x}) \{\mathbf{B}_{abc}^j(\mathbf{x})\}^T \{\mathbf{B}_{abc}^j(\mathbf{x})\} dx. \quad (24)$$

The j th level approximation of the corresponding basis functions can be explicitly evaluated (refer to (8) for an expression of \mathbf{B}_{abc}^j). An important point to note is that (8) involves several matrix multiplications and, hence, can be very expensive to evaluate. However, the matrix $(\mathbf{A}_{abc}^j) (= (\mathbf{A}_{abc})_m \cdots (\mathbf{A}_{abc})_t (\mathbf{A}_{abc})_i)$ in (8) encodes how vertices in the 2-neighborhood of the triangular face uvw at level j are related to the vertices in the 2-neighborhood of the triangular face abc in the initial mesh. In the implementation, we keep track of how a new vertex is obtained from the contributing vertices in its immediate predecessor level. If we move up from level j to level 0, we get the information stored in (\mathbf{A}_{abc}^j) without forming any local subdivision matrices and thus avoiding subsequent matrix multiplications.

By choosing a sufficiently high value of j , we achieve a reasonably good approximation of the elemental mass matrices. We eliminate the computations involved in evaluating the integrals in (24) by using the discrete mass density function, which has nonzero values only at the vertex positions of the j th subdivision level mesh. Therefore, the approximation of the mass matrix for the element can be written as

$$\mathbf{M}_{abc} = \sum_{i=1}^k \mu(\mathbf{v}_i^j) \{\mathbf{B}_{abc}^j(\mathbf{v}_i^j)\}^T \{\mathbf{B}_{abc}^j(\mathbf{v}_i^j)\}, \quad (25)$$

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 implementation purposes. The elemental damping matrix can be obtained in a similar fashion.

6.2 Elemental Stiffness Matrix

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. We assign spring-like energy, because of its simplicity and efficient computation, to the approximated model. The energy at the j th level of approximation can be defined as

$$E_{abc} \approx E_{abc}^j = \frac{1}{2} \sum_{\Omega} \frac{k_{lm} (|\mathbf{v}_l^j - \mathbf{v}_m^j| - \ell_{lm})^2}{|\mathbf{v}_l^j - \mathbf{v}_m^j|^2} (\mathbf{v}_l^j - \mathbf{v}_m^j)^2, \quad (26)$$

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, 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

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

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 $\mathbf{K}_{abc}^j = (\mathbf{A}_{abc}^j)^T (\mathbf{K}_{abc}^0) (\mathbf{A}_{abc}^j)$. There is no theoretical way of assigning the natural length ℓ_{lm} . We use the initialized model as the natural (rest) shape and, therefore, the distance between two vertices of the initialized model is used as the natural length of the spring connected between those vertices. Note that this approach is applicable for modeling isotropic, as well as anisotropic 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 employed to describe a wide range of material and physical behaviors such as linear elastic deformation and/or nonlinear plastic deformation.

6.3 Force Distribution

The force $\mathbf{f}(\mathbf{x}, t)$ in (18) represents the net effect of all externally applied forces. The current implementation supports spring, inflation as well as image-based forces. However, other types of forces, like repulsion forces, gravitational forces, etc., can easily be implemented.

To apply spring forces, a spring of stiffness k can be connected from a point \mathbf{d}_0 to a point \mathbf{x}_0 on the limit surface (or to the j th level approximation mesh), the net applied spring force being

$$\mathbf{f}(\mathbf{x}, t) = \int_{\mathbf{x} \in S^j} k(\mathbf{d}_0 - \mathbf{s}(\mathbf{x}, t)) \delta(\mathbf{x} - \mathbf{x}_0) dx, \quad (27)$$

where δ is the unit impulse function implying $\mathbf{f}(\mathbf{x}_0, t) = k|\mathbf{d}_0 - \mathbf{s}(\mathbf{x}_0, t)|$ and vanishes elsewhere on the surface. However, the δ function can be replaced with a smooth kernel to spread the force over a greater portion of the surface. The spring forces can be applied interactively using the computer mouse or the points from which forces need to be applied can be read in from a file.

It may be noted that the real force distribution is applied to the finest level and then transferred back to the initial DOFs on the coarsest resolution through the use of Jacobian matrix and the principle of virtual work. The computational expense to figure out the surface point on which the force is applied is $O(n)$, where n is the number of DOFs. We first figure out the nearest DOF in the coarsest level from the point of force application and then recursively find out the nearest point in the next higher resolution level by searching only the 2-neighbors of the previously selected nearest point. The recursive iteration is stopped at the highest resolution level used in the implementation. The

computational cost involved in applying the force back on the initial DOFs is also $O(n)$, as this effectively involves sparse matrix multiplication that does not need to be explicitly carried out.

To recover shapes from 3D image data, we synthesize image-based forces. A 3D edge detection is performed on a volume data set using the 3D Monga-Deriche (MD) operator [65] to produce a 3D potential field $P(x, y, z)$, which we use as an external potential for the model. The force distribution is then computed as

$$\mathbf{f}(x, y, z) = \lambda \frac{\nabla P(x, y, z)}{\|\nabla P(x, y, z)\|}, \quad (28)$$

where λ controls the strength of the force. The applied force on each vertex at the j th approximation level is computed by trilinear interpolation for evaluating (18) in Cartesian coordinates. More sophisticated image-based forces which incorporate region-based information, such as gradients of a thresholded fuzzy voxel classification, can also be used to yield better and more accurate shape recovery. Note that we can apply spring forces in addition with the image-based forces by placing points on the boundary of the region of interest in each slices of the 3D volume (MR, CT, etc.) image data.

6.4 Discrete Dynamic Equation

The differential equation given by (17) is integrated through time by discretizing the time derivative of \mathbf{p} over time steps Δt . The state of the dynamic subdivision surface at time $t + \Delta t$ is integrated using prior states at time t and $t - \Delta t$. An implicit time integration method is used in the current implementation, where discrete derivatives of \mathbf{p} are calculated using

$$\ddot{\mathbf{p}}(t + \Delta t) = \frac{\mathbf{p}(t + \Delta t) - 2\mathbf{p}(t) + \mathbf{p}(t - \Delta t)}{\Delta t^2} \quad (29)$$

and

$$\dot{\mathbf{p}}(t + \Delta t) = \frac{\mathbf{p}(t + \Delta t) - \mathbf{p}(t - \Delta t)}{2\Delta t}. \quad (30)$$

The elemental mass, damping, and stiffness matrices can be assembled to get the global mass, damping, and stiffness matrix for the smooth subdivision surface model. However, we do not assemble these global sparse matrices explicitly for efficiency reasons. For the time-varying stiffness matrix, we recompute \mathbf{K} at each time step. Using (17), (29), and (30), the discrete equation of motion is obtained as

$$(2\mathbf{M} + \mathbf{D}\Delta t + 2\Delta t^2\mathbf{K})\mathbf{p}(t + \Delta t) = 2\Delta t^2\mathbf{f}_p(t + \Delta t) + (\mathbf{D}\Delta t - 2\mathbf{M})\mathbf{p}(t - \Delta t) + 4\mathbf{M}\mathbf{p}(t). \quad (31)$$

This linear system of equations is solved iteratively between each time step using the conjugate gradient method [66], [67].

6.5 Physics-Based Subdivision

The initialized model grows dynamically according to the equation of motion (17). The degrees of freedom of the initialized model is equal to the number of control vertices in the initial mesh as mentioned earlier. When an

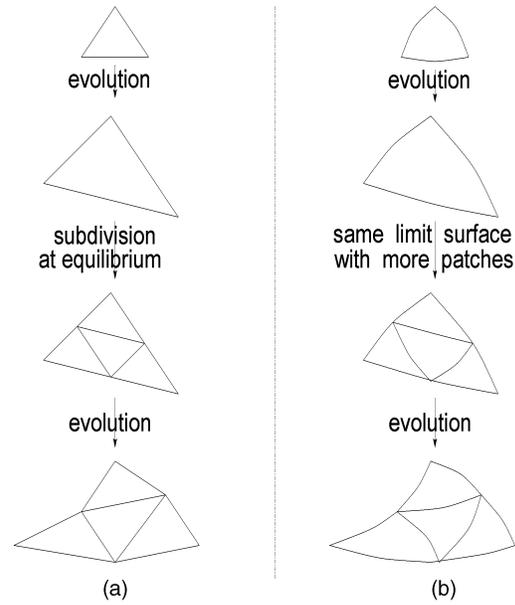


Fig. 9. Model subdivision to increase the degrees of freedom: (a) evolution of the initial mesh and (b) the corresponding limit surface evolution perceived by the user.

equilibrium is achieved for the model, the number of control vertices can be increased by replacing the original initial mesh by a new initial mesh obtained through one step of butterfly subdivision. This increases the number of degrees of freedom to represent the same (deformed) smooth limit surface and a new equilibrium position for the model can be obtained. This process is depicted schematically in Fig. 9. Model subdivision might be needed to obtain a very localized effect on a smooth limit surface. For a shape recovery application, one may start with a very simple initial model and, when an approximate shape is recovered, the degrees of freedom can be increased to obtain a new equilibrium position for the model with a better fit to the given data set. The error of fit criteria for the discrete data is based on distance between the data points and the points on the limit surface where the corresponding springs are attached. In the context of image-based forces, if the model energy does not change between successive iterations, indicating an equilibrium for the given resolution, the degrees of freedom for the model can be increased by the above-mentioned replacement scheme until the model energy is sufficiently small and the change in model energy between successive iterations becomes less than a prespecified tolerance.

7 RESULTS AND DISCUSSION

Our dynamic butterfly subdivision surface model can be used to represent a wide variety of smooth shapes with arbitrary genus. The smooth limit surface can be sculpted by applying synthesized forces in a direct and intuitive way in shape design applications. The underlying shape in a range or volume data set can also be recovered hierarchically using our dynamic butterfly subdivision surface model. To recover shapes from a given set of points in 3D, the existing subdivision surface-based techniques resort

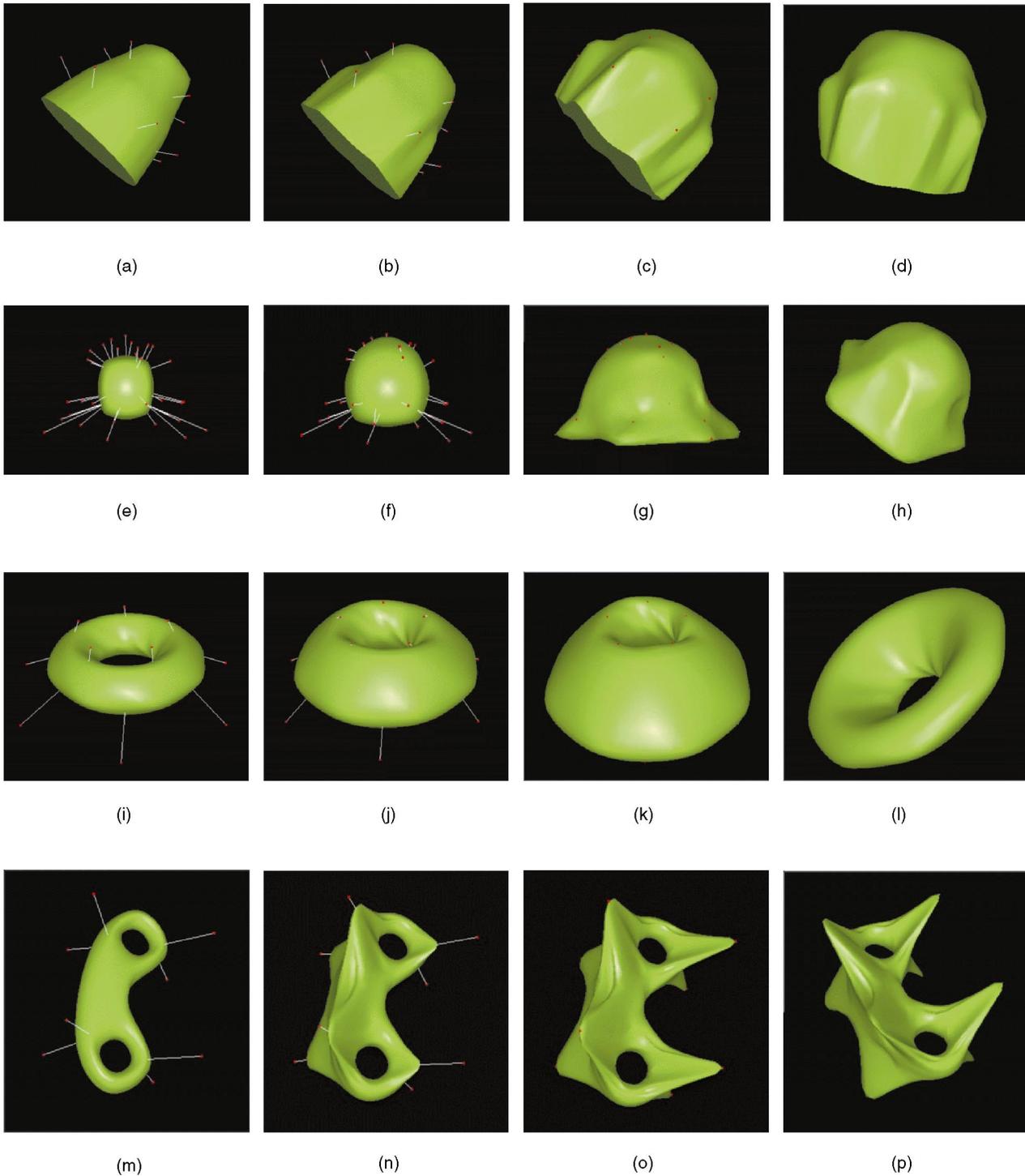


Fig. 10. **First column:** Initial shapes along with attached springs for deformation. **Second column:** Deformation of initial shapes due to spring forces. **Third column:** The final deformed shape. **Fourth column:** Another view of the final deformed shape.

to complex techniques to derive a mesh for the underlying shape and then, typically, mesh optimization techniques are used to obtain a compact representation of the same. Our model recovers the shape from a set of points in an efficient hierarchical way—any simple mesh of the same topology can be used as an initial mesh which will evolve over time to fit the given data and, depending on the error of fit achieved, it will automatically refine itself until the

prescribed error of fit is obtained. We would also like to point out that, in all our fitting examples, the final fit is *relatively* insensitive to the model initialization as is evident from the examples. Note that the initialization is very far from the true/expected solution. This is not so in the case of conventional geometry-based fitting algorithms. Most of these fitting methods rely on a nonlinear least squares technique or a quasi-Newton type numerical solver. These

TABLE 1
Editing Example Parameters

Example Surface	Control mesh of initial smooth surface has				Control mesh of final smooth surface has				Initial Error	Final Error
	Face Count	Vertex Count	Force Constant	Time Step	Face Count	Vertex Count	No. of Time Updates	No. of times DOFs increased		
Open	125	76	4000	0.01	125	76	48	0	15.27%	2.15%
Genus 0	24	14	1000	0.01	384	194	115	2	33.02%	2.82%
Genus 1	64	32	10000	0.01	64	32	56	0	19.60%	1.12%
Genus 2	272	134	1000	0.01	272	134	95	0	27.78%	2.93%

methods, although known to be quite stable in general, require a good model initialization for the fitting problem. In the dynamic model, it is possible for the mass term to provide enough momentum to push the model out of undesirable local minima during the fitting. This in turn can increase the convergence range of the fitting algorithm. In the rest of this section, we elaborate on these points via examples.

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. 10, 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. Note 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 two control vertices of the initialized mesh is used as the natural (rest) length of the spring attached between those vertices.

The various parameter values used in editing examples of Fig. 10 are tabulated in Table 1. Note that no theory exists in guiding us to choose a set of parameter values, like force constant, time step, etc., in the physics-based modeling paradigm. The chosen set of values is found to work well for the examples shown and it is determined in a trial and

error fashion. The magnitude of the force applied to the smooth limit surface is proportional to the length of the springs attached. 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, fewer number of degrees of freedom leads to faster deformation (a smaller system of equations is solved). For the examples shown in Fig. 10, the deformations took approximately 30-45 seconds under normal system load on an 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 error, final error, number of time updates, and subdivision steps required are listed in Table 1 for the examples shown in Fig. 10.

We have performed several experiments testing the applicability of our model to recover the underlying shapes in range and volume data sets. We use the distance between two control vertices of the initialized mesh as the natural (rest) length of the spring attached between those vertices (as in the editing examples). In all the fitting experiments,

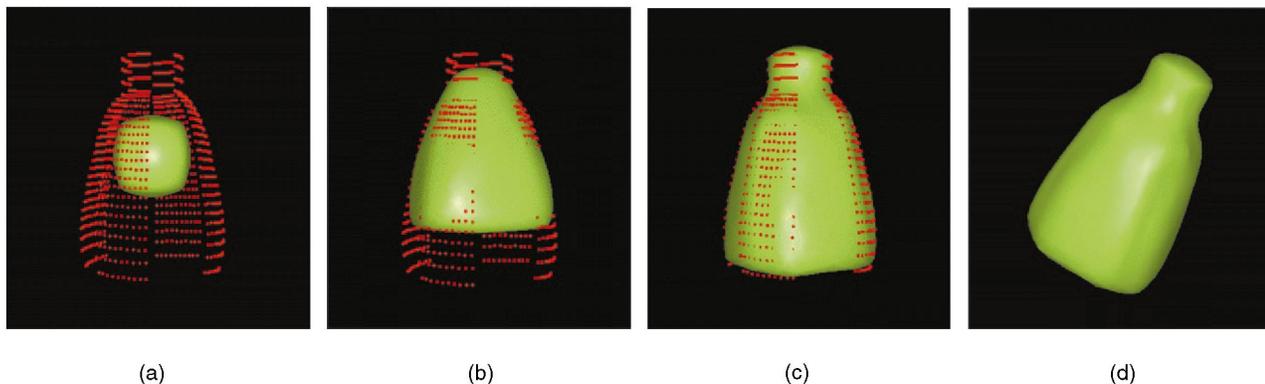


Fig. 11. (a) Range data of a bulb along with the initialized model, (b) model deformation, (c) the fitted dynamic butterfly subdivision model, and (d) visualization of the shape from another view point.

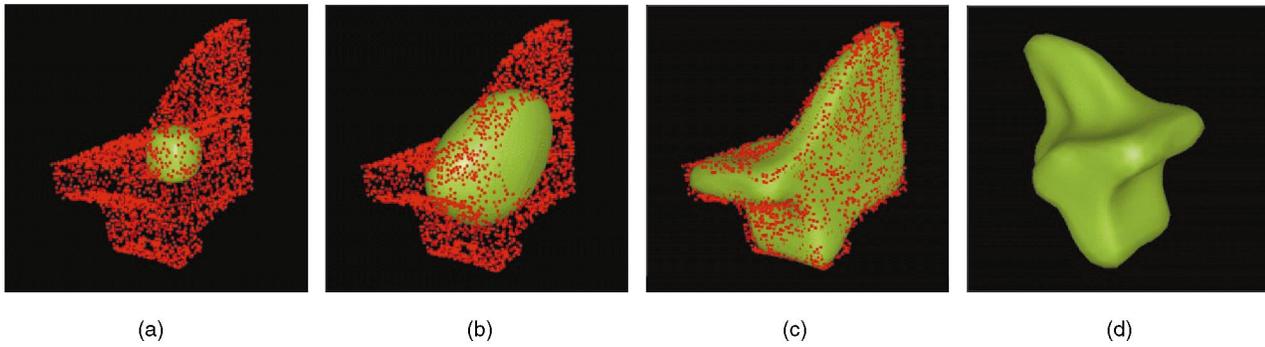


Fig. 12. (a) Range data of a mechanical part along with the initialized model, (b) model deformation, (c) the fitted dynamic butterfly subdivision model, and (d) visualization of the shape from another view point.

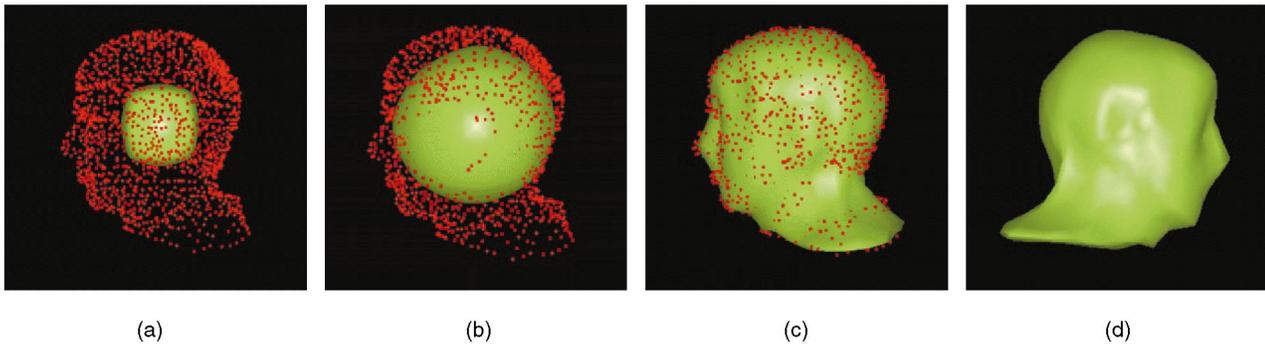


Fig. 13. (a) Range data of a head along with the initialized model, (b) model deformation, (c) the fitted dynamic butterfly subdivision model, and (d) visualization of the shape from another view point.

the initialized 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. Once an approximate shape is recovered, the model is refined depending on the data-fitting criteria, thereby increasing the degrees of freedom of the recovered shape only when necessary. For an error in fit of approximately 3 percent, the initialized model is refined twice, following the technique described in Section 6.5. Also, the limit surface of any control mesh (of the desired genus) can be used as the initialized model. However, an initial mesh with few degrees of freedom usually performs better in terms of recovering a compact representation of the underlying shape. The time needed to recover the shapes is approximately 3-4 minutes under normal system load on an Ultra-SPARC 30 machine. Note that the data sets we had for fitting examples do not contain high resolution features.

In the first shape recovery experiment, we depict the laser range data acquired from multiple views of a light bulb in Fig. 11a. Prior to applying our algorithm, the data were transformed into a single reference coordinate system. The model was initialized inside the 1,000 range data points on the surface of the bulb. The fitted dynamic (modified) butterfly subdivision surface model is shown in Fig. 11b and Fig. 11c. In the next experiment, the shape of a mechanical part is recovered from a range data set having 2,031 data points (Fig. 12). We also recover the shape of a human head from a range data set, as shown in Fig. 13. The head range data set has 1,779 points in 3D. Note that the final shape, with a very low error tolerance, is recovered

using very few number of control points in comparison to the number of data points present in the original range data set.

We also present the recovery of the caudate nucleus (a cortical structure in the human brain) from 64 MRI slices, each of size 256×256 in our next experiment. An expert neuroscientist (sparsely) placed points on the boundary of the shape of interest in each MRI slice. Fig. 14a depicts the sparse data points (placed in each of the slices depicting the boundary of the shape of interest) in 3D along with the initialized model. Note that points had to be placed on the boundary of the caudate nucleus due to lack of image gradients delineating the caudate from the surrounding tissue in parts of the image. Continuous image-based forces, as well as spring forces, are applied to the model and the model deforms under the influence of these forces until maximal conformation to the data is achieved. The final fitted model is shown in Fig. 14c. Note that the recovered shape in [48], using our previous dynamic subdivision surface model for the same data set, has 386 degrees of freedom and, therefore, we achieve a factor of 2 improvement in the number of degrees of freedom required to represent the model in this particular example.

The various parameter values used for model fitting are tabulated in Table 2. As in the editing examples, the parameter values, like time step, force constant etc., are chosen by a trial and error basis. We also plot the number of time updates versus error in fit in Fig. 15. When an equilibrium is reached for a given number of degrees of freedom, a new control mesh with more degrees of freedom is obtained for the same smooth limit surface via subdivision.

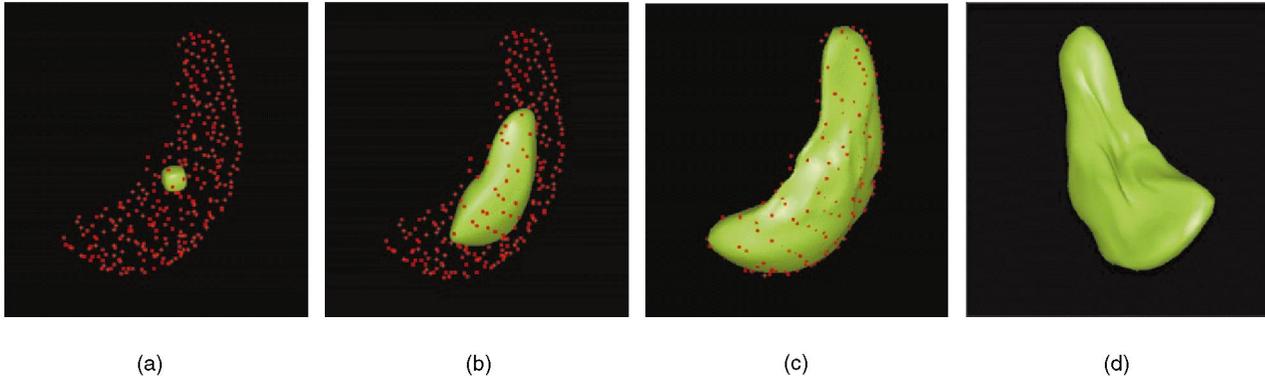


Fig. 14. (a) Data points (from all slices) in 3D along with the initialized model, (b) model deformation, (c) the fitted dynamic butterfly subdivision model, and (d) visualization of the shape from another view point.

TABLE 2
Fitting Example Parameters

Example	Control mesh of initial smooth surface has				Control mesh of final smooth surface has				Initial Error	Final Error
	Face Count	Vertex Count	Force Constant	Time Step	Face Count	Vertex Count	No. of Time Updates	No. of times DOFs increased		
Bulb	24	14	100	0.01	384	194	66	2	32.83%	3.33%
Anvil	24	14	10	0.01	384	194	150	2	39.87%	4.07%
Head	24	14	100	0.005	384	194	125	2	36.20%	6.59%
Caudate	24	14	100	0.01	384	194	120	2	45.67%	3.36%

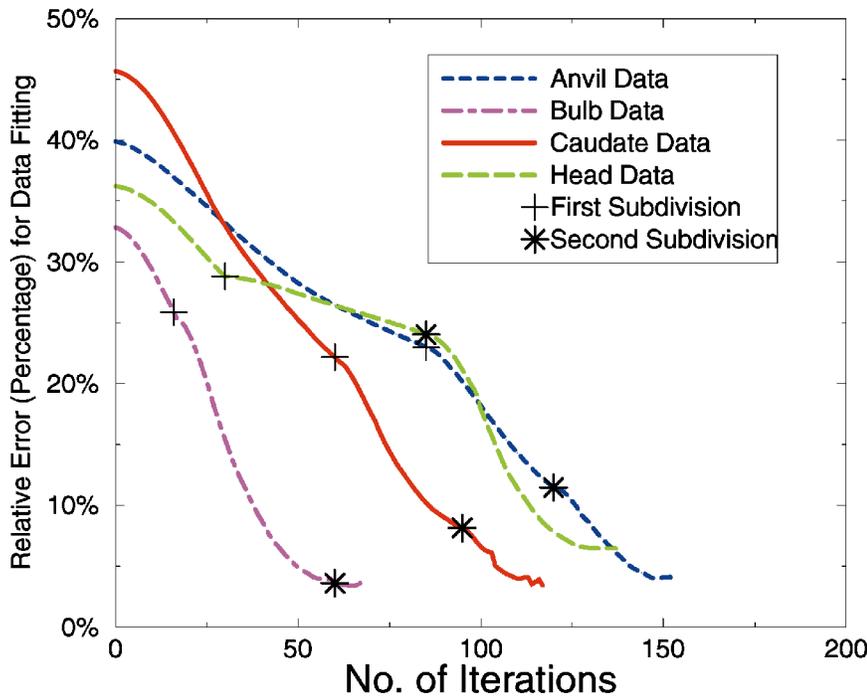


Fig. 15. Number of time updates vs. relative error plot for the fitting examples.

The time update points when subdivision became necessary are depicted in Fig. 15 for the fitting examples shown in this paper. Note that the user always perceives evolution of a smooth surface, but the amount of details that can be recovered depends on the degrees of freedom of the governing control mesh.

In the last experiment, we animate the motion of the left-ventricular chamber of a canine heart over a complete cardiac cycle. The data set comprised of 16 3D CT images, with each volume image having 118 slices of 128×128 pixels. First, we have recovered the shape from one data set using image-based (gradient), as well as

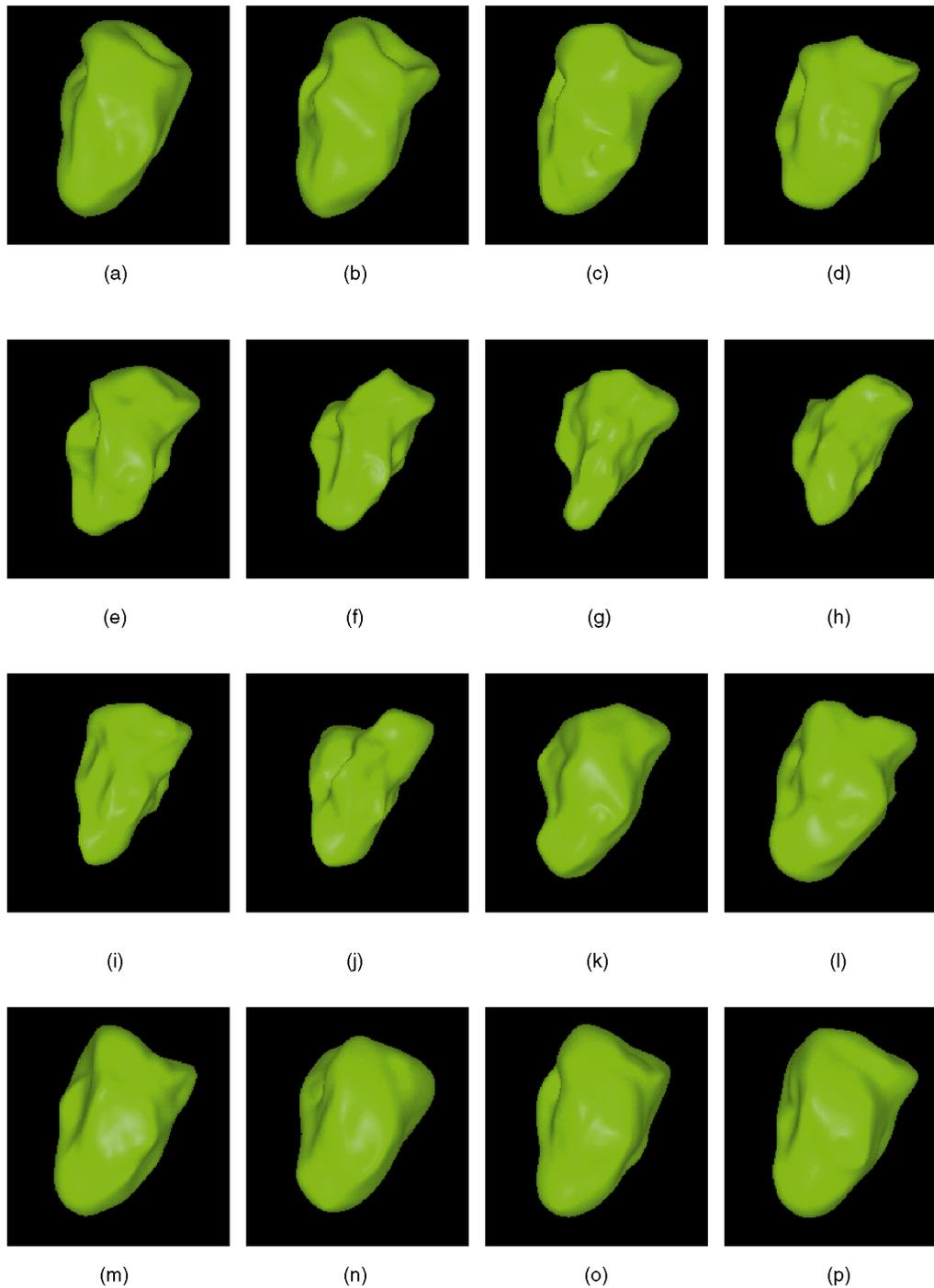


Fig. 16. Snapshots from the animation of canine heart motion over a cardiac cycle using the dynamic butterfly subdivision model.

point-based forces. Once the shape is recovered from one data set, this fitted model is used as an initialization for the next data set to track the shape of interest. The snapshots from motion tracking are shown in Fig. 16 for the 16 volume data sets. It may be noted that the control mesh describing the smooth surfaces shown in Fig. 16 has only 384 triangular faces with a total of 194 vertices, as mentioned earlier. This experiment clearly shows that our model can be used to track a shape of interest from a set of time dependent volume data sets in an efficient manner. *Note that no other existing purely geometric*

subdivision surface technique can be used with (time varying) continuous data sets.

8 CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a novel finite element method to derive and analyze the new dynamic model based on the modified butterfly subdivision surface scheme. The new physics-based surface model provides a direct and intuitive way of manipulating smooth shapes of arbitrary topology and is very useful for directly extracting

and visualizing shapes of interest in large range and volume data sets. The proposed model has also been used successfully for nonrigid motion tracking from a temporal sequence of volume data sets. We have developed a hierarchical local parameterization of the subdivision scheme which is critical to the formulation of our dynamic model. We have combined material properties with geometric entities, formulated the motion equations for our dynamic model, and incorporated the advantages of free-form deformable models into conventional subdivision schemes. Further, we have introduced a hierarchical dynamic control for various applications. Our experiments indicate a promising future for the proposed model in computer graphics, geometric modeling, and scientific visualization. Also, the finite element techniques proposed in this paper should be of great interest to the engineering design and analysis community as well.

Several features/enhancements can be incorporated into the proposed model. First, adaptive local subdivision of the control mesh will provide a useful enhancement to our model. Such a subdivision will introduce new degrees of freedom only in the regions needing refinement, making it more desirable in certain situations, such as geometry compression, local feature description, etc. Second, new algorithms that support automatic change of topology and automatic rule changing in order to reverse-engineer sharp features should be investigated. Additionally, various constraint imposition techniques, as well as rigorous finite element analysis of the butterfly shape functions need to be investigated. Novel force-based sculpting toolkits should enhance the functionalities of our dynamic system. We are currently pursuing these directions of research.

ACKNOWLEDGMENTS

This research was supported in part by US National Science Foundation (NSF) CAREER award CCR-9896123 and DMI-9896170 to H. Qin; and NSF grant IIS-9811042 and US National Institutes of Health grant RO1-RR13197 to B.C. Vemuri. We wish to acknowledge Dr. Tim McInerney for his help, Dr. Gregoire Malandain for the edge detection software, Dr. Hughes Hoppe and Dr. Kari Pulli for the range data, Dr. Dmitry Goldgof for the 4D CT images, and Dr. Christina Leonard for brain MRI images.

REFERENCES

- [1] G.M. Chaikin, "An Algorithm for High Speed Curve Generation," *Computer Vision, Graphics, and Image Processing*, vol. 3, no. 4, pp. 346-349, 1974.
- [2] D. Doo, "A Subdivision Algorithm for Smoothing Down Irregularly Shaped Polyhedrons," *Proc. Interactive Techniques in Computer Aided Design*, pp. 157-165, 1978.
- [3] M. Sabin, "The Use of Piecewise Forms for the Numerical Representation of Shape" PhD thesis, Hungarian Academy of Sciences, Budapest, 1976.
- [4] E. Catmull and J. Clark, "Recursively Generated B-Spline Surfaces on Arbitrary Topological Meshes," *Computer Aided Design*, vol. 10, no. 6, pp. 350-355, 1978.
- [5] C. Loop, "Smooth Subdivision Surfaces Based on Triangles," MS thesis, Dept. of Math., Univ. of Utah, 1987.
- [6] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle, "Piecewise Smooth Surface Reconstruction," *Proc., ACM SIGGRAPH*, pp. 295-302, July 1994.
- [7] M. Halstead, M. Kass, and T. DeRose, "Efficient, Fair Interpolation Using Catmull-Clark Surfaces," *Proc., ACM SIGGRAPH*, pp. 35-44, Aug. 1993.
- [8] J. Peters and U. Reif, "The Simplest Subdivision Scheme for Smoothing Polyhedra," *ACM Trans. Graphics*, vol. 16, no. 4, pp. 420-431, Oct. 1997.
- [9] T.W. Sederberg, J. Zheng, D. Sewell, and M. Sabin, "Non-Uniform Recursive Subdivision Surfaces," *Proc., ACM SIGGRAPH*, pp. 387-394, July 1998.
- [10] T. DeRose, M. Kass, and T. Truong, "Subdivision Surfaces in Character Animation," *Proc., ACM SIGGRAPH*, pp. 85-94, July 1998.
- [11] N. Dyn, D. Levin, and J.A. Gregory, "A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control," *ACM Transactions on Graphics*, vol. 9, no. 2, pp. 160-169, Apr. 1990.
- [12] N. Dyn, S. Hed, and D. Levin, "Subdivision Schemes for Surface Interpolation," *Proc. Workshop Computational Geometry*, pp. 97-118, 1993.
- [13] D. Zorin, P. Schröder, and W. Sweldens, "Interpolating Subdivision for Meshes with Arbitrary Topology," *Proc., ACM SIGGRAPH*, pp. 189-192, Aug. 1996.
- [14] D. Doo and M. Sabin, "Analysis of the Behavior of Recursive Division Surfaces Near Extraordinary Points," *Computer Aided Design*, vol. 10, no. 6, pp. 356-360, 1978.
- [15] A.A. Ball and D.J.T. Storry, "Conditions for Tangent Plane Continuity over Recursively Generated B-Spline Surfaces," *ACM Trans. Graphics*, vol. 7, no. 2, pp. 83-102, 1988.
- [16] A.A. Ball and D.J.T. Storry, "An Investigation of Curvature Variations over Recursively Generated B-Spline Surfaces," *ACM Trans. Graphics*, vol. 9, no. 4, pp. 424-437, 1990.
- [17] U. Reif, "A Unified Approach to Subdivision Algorithms Near Extraordinary Points," *Computer Aided Geometric Design*, vol. 12, no. 2, pp. 153-174, 1995.
- [18] C.A. Micchelli and H. Prautzsch, "Uniform Refinement of Curves," *Linear Algebra and Its Applications*, vols. 114/115, pp. 841-870, 1989.
- [19] A.S. Cavaretta, W. Dahmen, and C.A. Micchelli, "Stationary Subdivision," *Memoirs of the Am. Math. Soc.*, vol. 93, no. 453, Sept. 1991.
- [20] J.E. Schweitzer, "Analysis and Application of Subdivision Surfaces," PhD thesis, Univ. of Washington, Seattle, 1996.
- [21] A. Habib and J. Warren, "Edge and Vertex Insertion for a Class of C^1 Subdivision Surfaces," *Computer Aided Geometric Design*, to appear.
- [22] J. Peters and U. Reif, "Analysis of Generalized B-Spline Subdivision Algorithms," *SIAM J. Numerical Analysis*, to appear, available at <ftp://ftp.cs.purdue.edu/pub/jorg/9697agb.ps.Z>.
- [23] D. Zorin, "Smoothness of Stationary Subdivision on Irregular Meshes," *Constructive Approximation*, submitted, available as Stanford Computer Science Lab Technical Report CSL-TR-98-752.
- [24] J. Stam, "Exact Evaluation of Catmull-Clark Subdivision Surfaces at Arbitrary Parameter Values," *Proc. ACM SIGGRAPH*, pp. 395-404, July 1998.
- [25] L. Kobbelt, "Interpolatory Refinement by Variational Methods," *Approximation Theory VIII*, C. Chui and L. Schumaker, eds., pp. 217-224, World Scientific, 1995.
- [26] L. Kobbelt, "A Variational Approach to Subdivision," *Computer-Aided Geometric Design*, vol. 13, pp. 743-761, 1996.
- [27] L. Kobbelt and P. Schröder, "Constructing Variationally Optimal Curves through Subdivision," Technical Report CS-TR-97-05, California Inst. of Technology Computer Science Dept., 1997.
- [28] H. Weimer and J. Warren, "Subdivision Schemes for Thin Plate Splines," *Computer Graphics Forum*, vol. 17, no. 3, pp. 303-313, 1998.
- [29] T. Kurihara, "Interactive Surface Design Using Recursive Subdivision," *Communicating with Virtual Worlds*, N.M. Thalmann and D. Thalmann, eds., pp. 228-243, Springer-Verlag, 1993.
- [30] K. Pulli and M. Lounsbury, "Hierarchical Editing of Subdivision Surfaces," Technical Report UW-CSE-97-04-07, Univ. of Washington Computer Science Dept., 1997.
- [31] D. Zorin, P. Schröder, and W. Sweldens, "Interactive Multi-resolution Mesh Editing," *Proc. ACM SIGGRAPH*, pp. 259-268, Aug. 1997.
- [32] L. Kobbelt, S. Campagna, J. Vorsatz, and H.P. Seidel, "Interactive Multiresolution Modeling on Arbitrary Meshes," *Proc. ACM SIGGRAPH*, pp. 105-114, July 1998.
- [33] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, "Elastically Deformable Models," *Proc. ACM SIGGRAPH*, pp. 205-214, 1987.

- [34] D. Terzopoulos and K. Fleischer, "Deformable Models," *The Visual Computer*, vol. 4, no. 6, pp. 306-331, 1988.
- [35] A. Pentland and J. Williams, "Good Vibrations: Modal Dynamics for Graphics and Animation," *Proc. ACM SIGGRAPH*, pp. 215-222, 1989.
- [36] D. Metaxas and D. Terzopoulos, "Dynamic Deformation of Solid Primitives with Constraints," *Proc. ACM SIGGRAPH*, pp. 309-312, July 1992.
- [37] B.C. Vemuri and A. Radisavljevic, "Multiresolution Stochastic Hybrid Shape Models with Fractal Priors," *ACM Trans. Graphics*, vol. 13, no. 2, pp. 177-207, Apr. 1994.
- [38] G. Celniker and D. Gossard, "Deformable Curve and Surface Finite Elements for Free-Form Shape Design," *Proc. ACM SIGGRAPH*, pp. 257-266, July 1991.
- [39] M.I.J. Bloor and M.J. Wilson, "Representing PDE Surfaces in Terms of B-Splines," *Computer Aided Design*, vol. 22, no. 6, pp. 324-331, 1990.
- [40] M.I.J. Bloor and M.J. Wilson, "Using Partial Differential Equations to Generate Free-Form Surfaces," *Computer Aided Design*, vol. 22, no. 4, pp. 202-212, 1990.
- [41] G. Celniker and W. Welch, "Linear Constraints for Deformable B-Spline Surfaces," *Proc. Symp. Interactive 3D Graphics*, pp. 165-170, 1992.
- [42] W. Welch and A. Witkin, "Variational Surface Modeling," *Proc. ACM SIGGRAPH*, pp. 157-166, July 1992.
- [43] H. Qin and D. Terzopoulos, "Dynamic NURBS Swung Surfaces for Physics-Based Shape Design," *Computer-Aided Design*, vol. 27, no. 2, pp. 111-127, 1995.
- [44] H. Qin and D. Terzopoulos, "D-NURBS: A Physics-Based Framework for Geometric Design," *IEEE Trans. Visualization and Computer Graphics*, vol. 2, no. 1, pp. 85-96, Mar. 1996.
- [45] D. Terzopoulos and H. Qin, "Dynamic NURBS with Geometric Constraints for Interactive Sculpting," *ACM Trans. Graphics*, vol. 13, no. 2, pp. 103-136, Apr. 1994.
- [46] C. Mandal, H. Qin, and B.C. Vemuri, "Dynamic Smooth Subdivision Surfaces for Data Visualization," *Proc. IEEE Visualization '97*, pp. 371-377, Oct. 1997.
- [47] H. Qin, C. Mandal, and B.C. Vemuri, "Dynamic Catmull-Clark Subdivision Surfaces," *IEEE Trans. Visualization and Computer Graphics*, vol. 4, no. 3, Sept. 1998.
- [48] C. Mandal, B.C. Vemuri, and H. Qin, "Shape Recovery Using Dynamic Subdivision Surfaces," *Proc. Int'l Conf. Computer Vision*, pp. 805-810, Jan. 1998.
- [49] E. Bardinet, L.D. Cohen, and N. Ayache, "Superquadrics and Free-Form Deformations: A Global Model to Fit and Track 3D Medical Data," *Proc. Computer Visualization and Pattern Recognition in Medicine*, N. Ayache, ed., pp. 319-326, 1995.
- [50] L.D. Cohen and I. Cohen, "Finite-Element Methods for Active Contour Models and Balloons for 2D and 3D Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1,131-1,147, Nov. 1993.
- [51] E. Koh, D. Metaxas, and N. Badler, "Hierarchical Shape Representation Using Locally Adaptive Finite Elements," *Proc. European Conf. Computer Vision*, J.O. Eklundh, ed., pp. 441-446, 1994.
- [52] F. Leitner and P. Cinquin, "Complex Topology 3D Objects Segmentation," *Model-Based Vision Development and Tools, SPIE Proc.*, pp. 16-26, 1991.
- [53] T. McInerney and D. Terzopoulos, "A Dynamic Finite Element Surface Model for Segmentation and Tracking in Multidimensional Medical Images with Application to Cardiac 4D Image Analysis," *Computerized Medical Imaging and Graphics*, vol. 19, no. 1, pp. 69-83, 1995.
- [54] L.H. Staib and J.S. Duncan, "Boundary Finding with Parametrically Deformable Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1,061-1,075, Nov. 1992.
- [55] D. Terzopoulos and D. Metaxas, "Dynamic 3D Models with Local and Global Deformations: Deformable Superquadrics," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, pp. 703-714, July 1991.
- [56] D. Metaxas and D. Terzopoulos, "Shape and Non-Rigid Motion Estimation through Physics-Based Synthesis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 6, pp. 580-591, June 1993.
- [57] A. Pentland and B. Horowitz, "Recovery of Non-Rigid Motion and Structure," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, pp. 730-742, July 1991.
- [58] Y. Chen and G. Medioni, "Surface Description of Complex Objects from Multiple Range Images," *Proc. Conf. Computer Vision and Pattern Recognition*, pp. 153-158, June 1994.
- [59] W.C. Huang and D. Goldgof, "Adaptive-Size Physically-Based Models for Non-rigid Motion Analysis," *Proc. Conf. Computer Vision and Pattern Recognition*, pp. 833-835, June 1992.
- [60] H. Tanaka and F. Kishino, "Adaptive Mesh Generation for Surface Reconstruction: Parallel Hierarchical Triangulation without Discontinuities," *Proc. Conf. Computer Vision and Pattern Recognition*, pp. 88-94, June 1993.
- [61] M. Vasilescu and D. Terzopoulos, "Adaptive Meshes and Shells: Irregular Triangulation, Discontinuities and Hierarchical Subdivision," *Proc. Conf. Computer Vision and Pattern Recognition*, pp. 829-832, June 1992.
- [62] J.M. Lounsbery, T. DeRose, and J. Warren, "Multiresolution Analysis for Surfaces of Arbitrary Topological Type," *ACM Trans. Graphics*, vol. 16, no. 1, pp. 34-73, Jan. 1997.
- [63] B.R. Gossick, *Hamilton's Principle and Physical Systems*. New York: Academic Press, 1967.
- [64] H. Kardestuncer, *The Finite Element Handbook*. New York: McGraw-Hill, 1987.
- [65] O. Monga and R. Deriche, "3D Edge Detection Using Recursive Filtering," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 28-35, June 1989.
- [66] G.H. Golub and V.H. Van Loan, *Matrix Computations*. The Johns Hopkins Univ. Press, 1989.
- [67] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in C*. Cambridge Univ. Press, 1992.



Chhandomay Mandal received the BTech (hons.) degree in electronics and electrical communication engineering from the 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 is a senior member of the technical staff at Sun Microsystems, Inc., Chelmsford, Massachusetts. 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 1995-98. His research interests include computer graphics and modeling, operating systems, and computer architecture.



Hong Qin received his BS (1986) degree and his MS degree (1989) in computer science from Peking University in Beijing, People's Republic of China. He received his PhD (1995) degree in computer science from the University of Toronto. He is an assistant professor of computer science at State University of New York at Stony Brook (SUNYSB), where he is also a member of the SUNYSB Center for Visual Computing. During 1989-1990, he was a research scientist

at the 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 the US National Science Foundation CAREER Award. He is a member of the 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 the IBM T.J. Watson Research Center, Yorktown Heights, New York, with the

Exploratory Computer Vision Group in the summers of 1988 and 1992. During the academic year 1989-1990, he was a 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 US National Science Foundation 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 at the European Conference on Computer Vision in 1994. 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 associate editor for the *IEEE Transactions on Pattern Analysis and Machine Intelligence* in the past. Dr. Vemuri is a senior member of the IEEE.