Direct Manipulation of Butterfly Subdivision Surfaces: A Physics-based Approach

Chhandomay Mandal*, Hong Qin† and Baba C. Vemuri*

*Authors' address: Department of Computer and Information Science and Engineering, University of Florida, Gainesville, Florida, 32611. Email:{cmandal | vemuri}@cise.ufl.edu.

†Author's address : Department of Computer Science, State University of New York at Stony Brook, Stony Brook, New York, 11794. Email: qin@cs.sunysb.edu.

Abstract

Subdivision surfaces are extensively used to model smooth shapes of arbitrary topology. Recursive subdivision on an user-defined initial control mesh generates a visually pleasing smooth surface in the limit. However, users have to carefully select the initial mesh and/or manipulate the control vertex positions at different levels of subdivision hierarchy to satisfy the functional and aesthetic requirements in the smooth limit surface. This modeling drawback results from the lack of direct manipulation tools for the limit surface. In this paper, we integrate techniques from physics-based modeling with geometric subdivision methodology, and present a scheme for direct 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 distribution functions, 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 physics-based butterfly subdivision surface as a set of novel finite elements, and (3) optimal approximation of this new type of finite elements by a collection of existing finite elements subjected to implicit geometric constraints. Our new dynamic model can be sculpted directly by applying synthesized forces, and its equilibrium is characterized by a minimum of the 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 non-rigid motion estimation from large range and volumetric data sets using very few degrees of freedom (control vertices that define the initial polyhedron). This new physics-based model promises a greater potential of subdivision schemes in computer graphics, geometric modeling, and virtual environments.

Keywords

Computer Graphics, Physics-based Modeling, Geometric Modeling, CAGD, Scientific Visualization, Subdivision Surfaces, Deformable Models, Finite Elements, Interactive Techniques.

I. INTRODUCTION

The concept of subdivision is ubiquitous in computer science. In visual computing areas, subdivision surfaces are extensively used to model smooth shapes of arbitrary topology for computer graphics, 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 an 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 the modeler to interactively obtain the desired effects on the smooth surface by kinematically manipulating the vertex positions at various levels of subdivision hierarchy. In this paper, we tackle the challenging problem of direct manipulation of the limit subdivision surface at arbitrary locations/areas and offer a novel solution to this problem by embedding the modified butterfly subdivision scheme in a physics-based modeling framework. As a result, unlike the existing geometric solutions that only allow the operations on control vertices, our methodology and algorithms permit the user to physically modify the shape of subdivision surfaces at desired locations via application of forces. This gives the user an intuitive and natural feeling that is produced while modeling with real clay/play-dough. We will also demonstrate that the proposed model efficiently recovers shapes as well as non-rigid motions from large range and volumetric data sets. Note that, this paper neither proposes a new subdivision technique nor provides a different interpretation of any existing subdivision technique, but *integrates* the advantages of subdivision surface-based and physics-based modeling techniques to solve important theoretical and practical problems. Although the principles of physics-based modeling are well understood by the graphics experts and modeling researchers, this paper will greatly advance the state of the art in physics-based shape modeling due to the following contributions : (1) a hierarchical local

parameterization scheme for (modified) butterfly subdivision surfaces is derived; (2) material properties are assigned to the smooth limit surface in order to embed this geometric model in a physics-based modeling framework; (3) the smooth limit surface is decomposed into a set of novel finite elements; (4) the motion equations that govern the behavior of (modified) butterfly subdivision surfaces are derived; (5) users can sculpt this physics-based model at desired locations and the model responds naturally (as a real-world object would); and (6) algorithms and procedures are developed to best approximate this novel finite element using a collection of existing lower order finite elements. We will address the novelty of the proposed model, detail our contributions, and compare this new model with our previous research results on dynamic subdivision surfaces in later sections. First, we will briefly review the previous work on subdivision surfaces.

A. Background

In [1], Chaikin first introduced the concept of subdivision to the computer 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 namely, (1) approximating subdivision techniques and (2) interpolating subdivision techniques.

Among the approximating schemes, the techniques of Doo and Sabin [2], [3] and Catmull and Clark [4] generalize the idea of obtaining uniform biquadratic and bicubic B-spline patches respectively from a rectangular control mesh. In [4], Catmull and Clark developed a method for recursively generating a smooth surface from a polyhedral mesh of arbitrary topology. The Catmull-Clark subdivision surface, defined by an arbitrary initial mesh, can be reduced to a set of standard B-spline patches except at a finite number of degenerate points. In [5], Loop 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, non-uniform Doo-Sabin and Catmull-Clark surfaces that generalize non-uniform tensor product B-spline surfaces to arbitrary topologies were introduced by Sederberg et al. [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 [12]. Zorin et al. [13] has developed an improved interpolatory subdivision scheme (which we call the *modified* butterfly scheme) that retains the simplicity of the butterfly scheme and results in much smoother surfaces even from irregular initial meshes. These interpolatory subdivision schemes have extensive applications in wavelets on manifolds, multiresolution decomposition of polyhedral surfaces and multiresolution editing. A variational approach for interpolatory refinement has been proposed by Kobbelt [14], [15] and by Kobbelt and Schröder [16]. In this approach, the vertex positions in the refined mesh at each subdivision step are obtained by solving an optimization problem. Therefore, these schemes are global, i.e., every new vertex position depends on all the vertex positions of the coarser level mesh. The local refinement property which makes the subdivision schemes attractive for implementation in the computer graphics applications is not retained in the variational approach.

The derivation of various mathematical properties of the smooth limit surface generated by the subdivision algorithms is rather complex. Doo and Sabin [17] first analyzed the smoothness behavior of the limit surface using the Fourier transform and an eigen-analysis of the subdivision matrix. Ball and Storry [18], [19] and Reif [20] further extended Doo and Sabin's prior work on continuity properties of subdivision surfaces by deriving various necessary and sufficient conditions on smoothness for different subdivision schemes. Specific subdivision schemes were analyzed by Schweitzer [21], Habib and Warren

[22], Peters and Reif [23] and Zorin [24]. Most recently, Stam [25] presented a method for exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values.

B. Shape modeling using the physics-based subdivision-surface model

Although recursive subdivision surfaces are powerful for representing smooth geometric shapes of arbitrary topology, they constitute a purely geometric representation, and furthermore, conventional geometric modeling with subdivision surfaces may be difficult for representing highly 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 (most 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 non-intuitive and laborious in general. In addition, it may not be enough to obtain the most "fair" surface that interpolates a set of (ordered or unorganized) data points. A certain number of local features such as bulges or inflections may be strongly desired while requiring geometric objects to satisfy global smoothness constraints in geometric modeling and computer graphics applications. In contrast, physics-based modeling provides a superior approach to shape modeling that can overcome most of the limitations associated with traditional geometric modeling approaches. Free-form deformable models governed by the laws of continuum mechanics are of particular 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 easy 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 and visualization in Terzopou-

los et al. [26] and further developed by Terzopoulos and Fleischer [27], Pentland and Williams [28], Metaxas and Terzopoulos [29] and Vemuri and Radisavljevic [30]. Celniker and Gossard [31] developed a system for interactive free-form design based on the finite element optimization of energy functionals proposed in [27]. Bloor and Wilson [32], [33], Celniker and Welch [34] and Welch and Witkin [35] 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 [36], [37], [38] 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. In [10], DeRose et al. 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 can not solve the modeling goal we are trying to achieve. Previously we introduced a dynamic Catmull-Clark subdivision surface model [39], [40] 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 discussed in detail later in Section I-D. In this paper, 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 straight-forward manner. The model can be initialized interactively by an

user-defined control mesh and is amenable to further sculpting via direct application of synthesized forces to any region of object surface. The formulation and implementation details are discussed in subsequent sections.

C. Shape and motion estimation using the physics-based subdivision-surface model

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 [41], another important application of the dynamic subdivision surfaces is in non-rigid shape and motion 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 [30], [38], [41], [42], [43], [44], [45], [46], [47]. Another important criterion in the design of shape models 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 [30], [43], [48], [49], [50] or adaptive size [44], [46], [51], [52], [53], [54] grids. The models with fixed grid size generally use less 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 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 considerable amount of details have more or less a C^1 surface. Under these circumstances, the finite element approaches as in [43], [46] need a *large* number of degrees of freedom for deriving a smooth and accurate representation. In addition, they

can not 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 [39], [40], [41] offered an elegant 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.

D. Contributions

In this section, we summarize the contributions of the present work along with the advantages of the proposed dynamic modified butterfly subdivision surface model over the dynamic Catmull-Clark subdivision surface model [39], [40]. The primary contributions of the present work are as follows.

- We systematically derive a local parameterization scheme for the modified butterfly subdivision surfaces in a hierarchical style, 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.
- We use modified butterfly subdivision techniques to create a surface model that incorporates mass and damping distribution functions, 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 presented 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 this force application. This 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 element 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 optimal in the sense that it can fall within any user-specified error tolerance.

Although the long-term goals inherent in the previously-developed dynamic Catmull-Clark model [39], [40] are the same as our current endeavor of deriving dynamic modified butterfly subdivision scheme, the research presented in this paper achieves them in a much more elegant fashion. First of all, 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 can not be readily generalized to other approximating subdivision schemes in a straight-forward 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 plate energy of Catmull-Clark subdivision surface diverges as shown in [7], hence various case analysis need to be performed to derive the internal energy of the dynamic Catmull-Clark subdivision surface model. The internal deformation energy of the dynamic butterfly scheme can be derived in an unified fashion for a spectrum 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 [48] is computationally expensive. 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. Lastly, it has been empirically found that the recovered shape is more compact (less number of degrees of freedom) when using the proposed model in comparison with our earlier reported model using the same data set and model-fitting criteria.

E. Overview

The rest of the paper is organized as follows: Section II presents the detailed formulation of the dynamic modified butterfly subdivision surfaces. The implementation details are presented in Section III followed by experimental results in Section IV. Finally, we make the concluding remarks in Section V.

II. Formulation

In this section, we provide a systematic formulation of the dynamic subdivision surface model. First, we briefly review the 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. It may be noted that these techniques can be generalized to define and construct a generic dynamic framework for other triangle-based subdivision surface schemes as well, but we will focus only on the modified butterfly subdivision technique in this paper.

A. The (modified) butterfly subdivision scheme

The butterfly subdivision scheme [11], like many other subdivision schemes used in geometric design literature/applications, 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

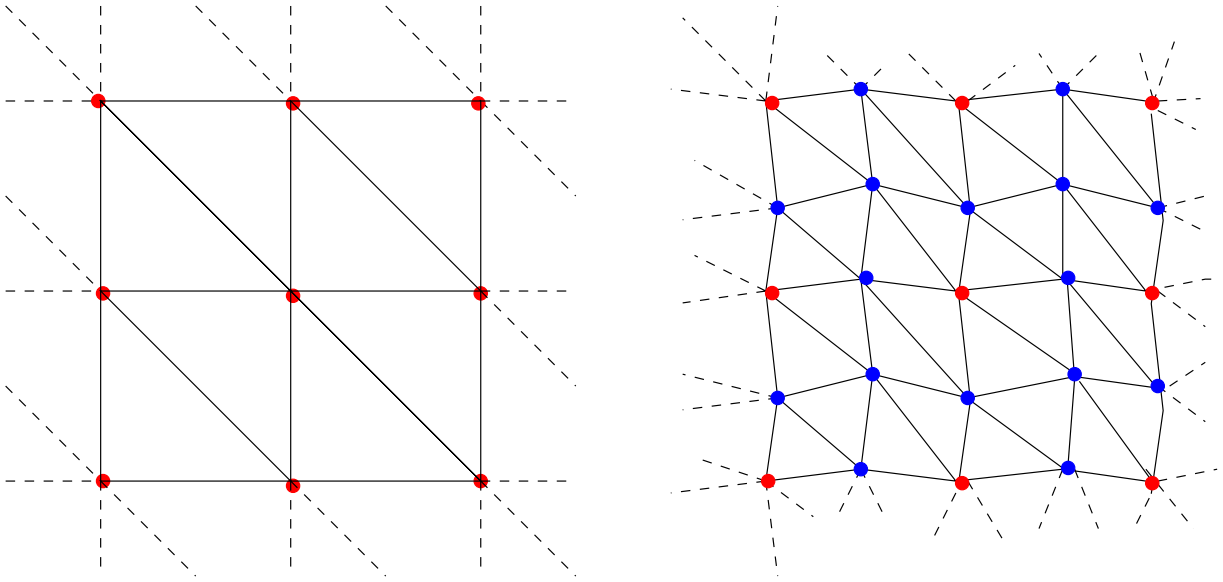


Fig. 1. (a) The control polygon with triangular faces; (b) Mesh obtained after one subdivision step.

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 spilt by adding a new vertex whose position is obtained by an affine combination of the neighboring vertex positions in the coarser level. For instance, the mesh in Fig.1(b) is obtained by subdividing the initial mesh shown in Fig.1(a) once. It may be noted 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.2(a). 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.2(b). The vertex $\mathbf{e}_{12}^{j+1}$ in the $j+1$-th level of subdivision, corresponding to the edge connecting vertices $\mathbf{v}_1^j$ and $\mathbf{v}_2^j$ at level j, is obtained by

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

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

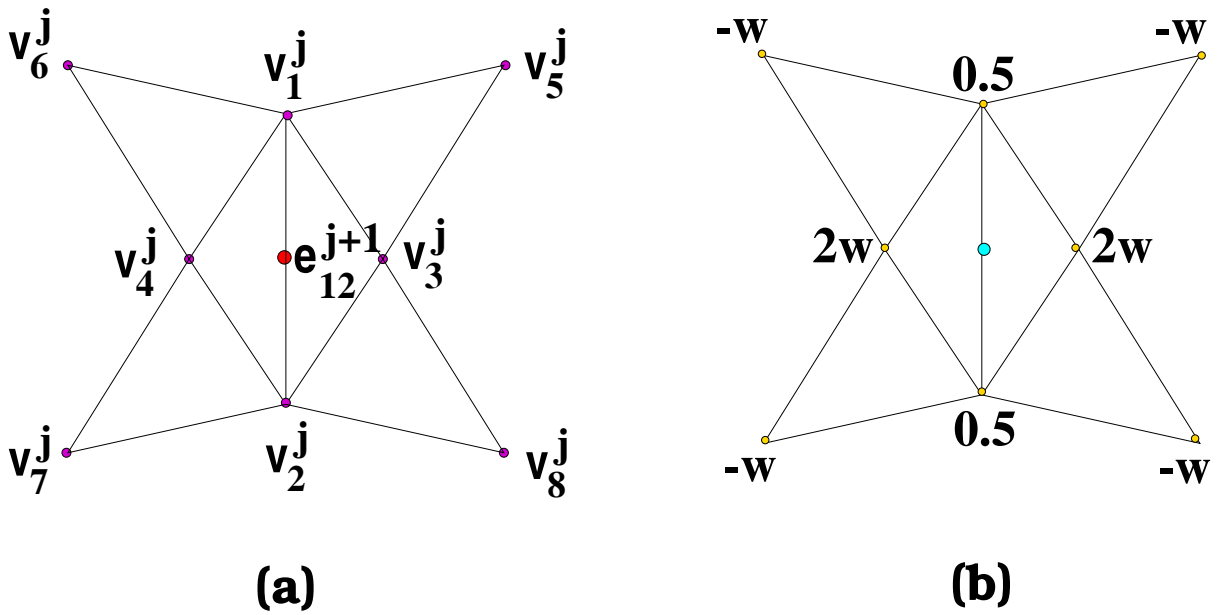


Fig. 2. (a) The contributing vertices in the j-th level for the vertex in the j+1-th level corresponding to the edge between $\mathbf{v}_1^j$ and $\mathbf{v}_2^j$; (b) the weighing factors for different vertices.

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 [13], all the edges have been categorized into three classes : (i) edges connecting two vertices of valence 6 (a 10 point stencil, as shown in Fig.3(a), is used to obtain the new vertex positions corresponding to these edges), (ii) 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.3(b), where $q = .75$ is the weight associated with the vertex of valence $n \neq 6$, and $s_i = (0.25 + cos(2\pi i/n) + 0.5cos(4\pi i/n))/n$, $i = 0, 1, \ldots, n-1$, are the weights associated with the vertices of valence 6), and (iii) edges connecting two vertices of valence $n \neq 6$. The last case can not occur except in the initial mesh as 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.3(b) at each of those two extraordinary vertices.

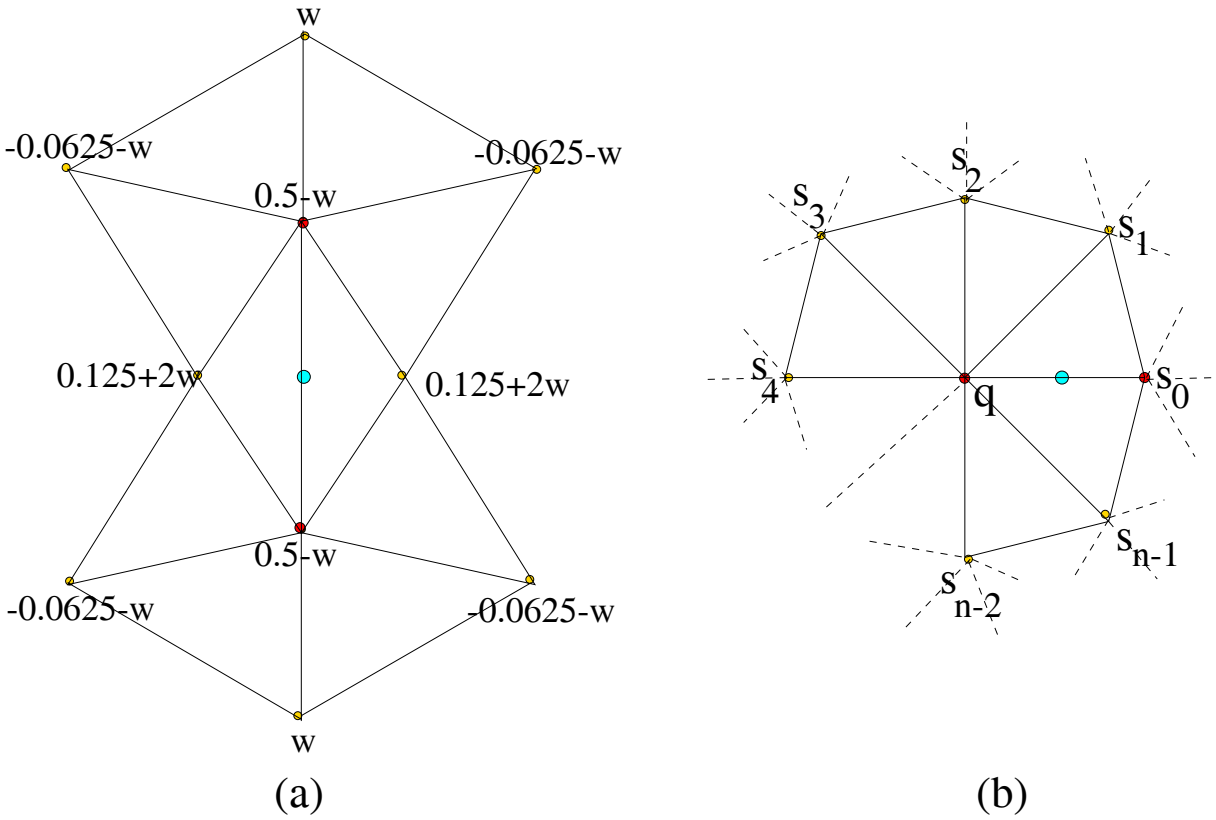


Fig. 3. (a) The weighing factors of contributing vertex positions for an edge connecting two vertices of valence 6; (b) the corresponding case when one vertex is of valence n and the other is of valence 6.

B. Local parameterization of the limit surface

Oftentimes, the smooth limit surface defined by the recursive subdivision process is of arbitrary topology where a global parameterization may not be possible. Nevertheless, we can locally parameterize the limit surface over the domain defined by the initial mesh following a similar approach described in [55]. The idea is to track any arbitrary point on the initial mesh across the meshes obtained via the subdivision process (see Fig.4 and 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 smoothes 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

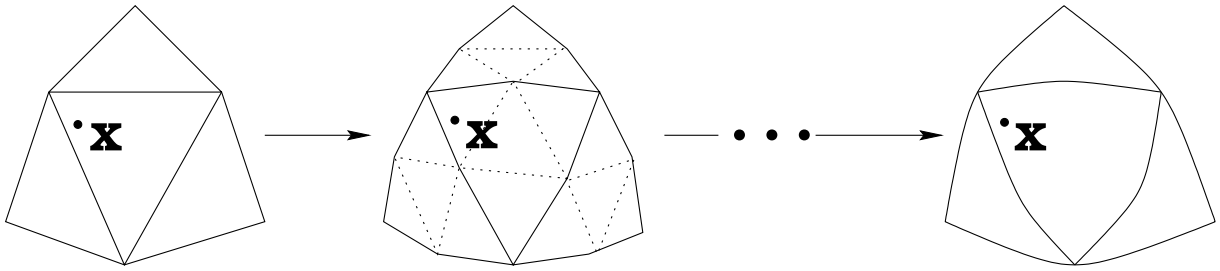


Fig. 4. The smoothing effect of the subdivision process on the triangles of the initial mesh.

as that of the triangular faces in the initial mesh. Therefore, we can express the limit surface $\mathbf{s}$ as

$$\mathbf{s} = \sum_{k=1}^{n} \mathbf{s}_k, \tag{2}$$

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.

We are now ready to describe the parameterization of the limit surface over the initial mesh. The process is best explained through the following example. We choose a simple planar mesh shown in Fig.5(a) 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 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 introduction of magenta colored new vertices. It may be noted that *any point inside the smooth triangular patch in the limit surface corresponding to the face abc in the initial mesh depends only on the vertices in the initial mesh which are within the 2-neighborhood of the vertices* $\mathbf{a}, \mathbf{b}$ *and* $\mathbf{c}$ *due to the local nature of the subdivision process.* For example, the vertex $\mathbf{d}$, introduced after first subdivision step, can be obtained using the 10 point stencil shown in Fig.3(a) 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.5(b), 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 formulation, superscripts are used to indicate the subdivision level. For example,

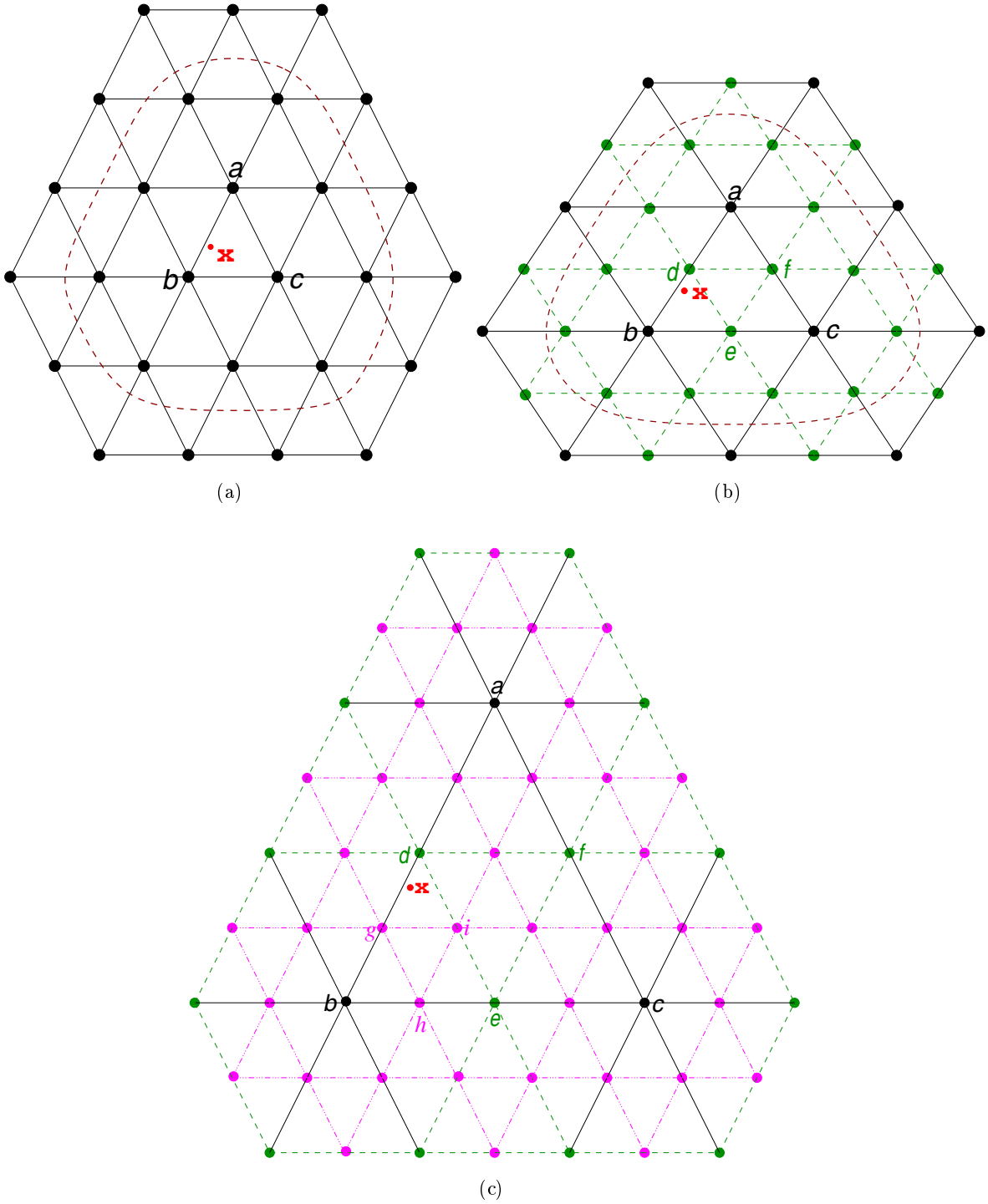


Fig. 5. Tracking a point **x** through various levels of subdivision : (a) initial mesh, (b) the selected section (enclosed by dotted lines) of the mesh in (a), after one subdivision step, (c) the selected section of the mesh in (b), after another subdivision step.

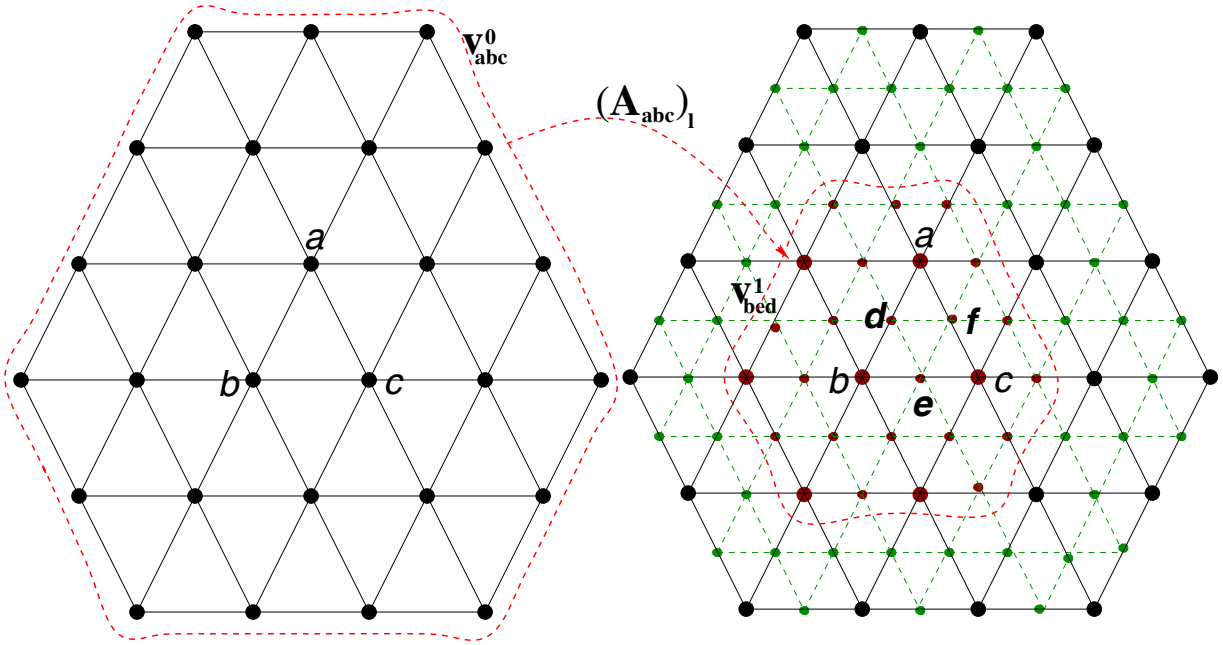
$\mathbf{v}^j_{uvw}$ denotes the collection of vertices at level j which control the smooth patch in the limit surface corresponding to the triangular face uvw at the j-th level of subdivision. Let $\mathbf{v}^0_{abc}$ be the collection of vertices in the initial mesh which are within the 2-neighborhood of the vertices $\mathbf{a}, \mathbf{b}$ and $\mathbf{c}$ (marked black in Fig.5(a)). Let the number of such vertices be r. Then, the vector $\mathbf{v}^0_{abc}$, 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 exists 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

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

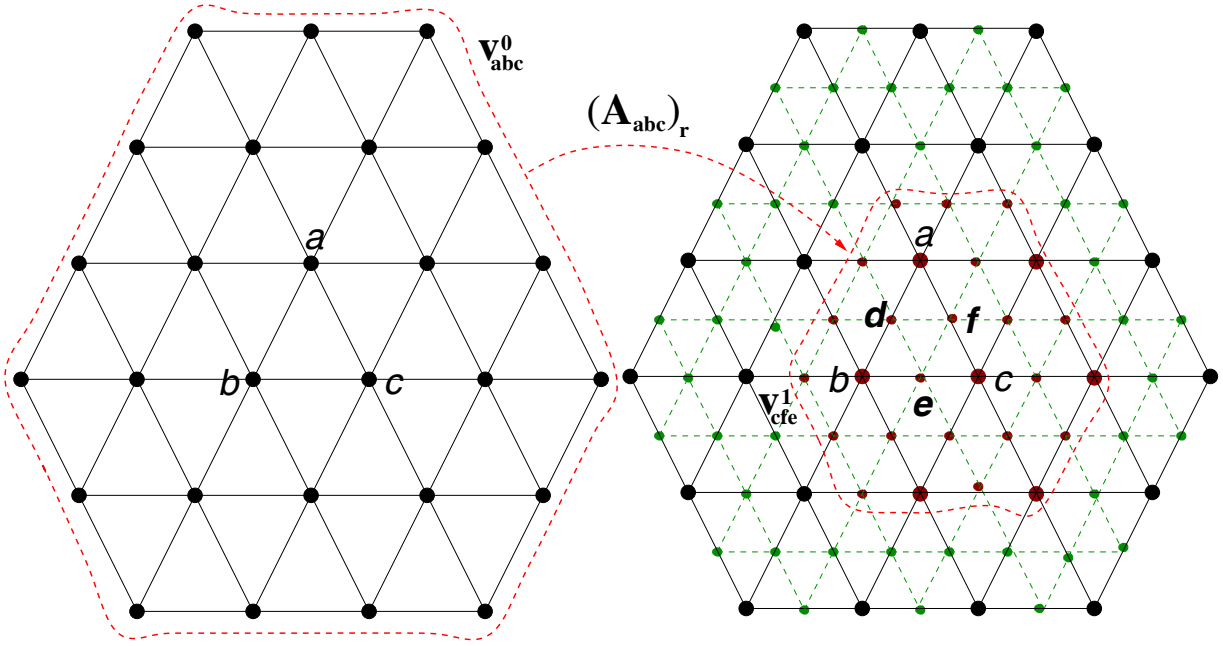
where the subscripts t, l, r and m denote top, left, right and middle triangle positions respectively (indicating the relative position of the *new* triangle with respect to the *original* triangle), and $\mathbf{v}^1_{adf}, \mathbf{v}^1_{bed}, \mathbf{v}^1_{cfe}$ and $\mathbf{v}^1_{def}$ are the concatenation of the (x, y, z) positions for the vertices in the 2-neighborhood of the corresponding triangle in the newly obtained subdivided mesh. The new vertices in this level of subdivision are marked green in Fig.5(b). The 2-neighborhood configuration of the vertices in the newly obtained triangles is exactly the same as that of the original triangle, hence local subdivision matrices are square and the vector dimensions on both sides of Eqn.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.5(c). Adopting a similar approach as in the derivation of Eqn.3, we obtain

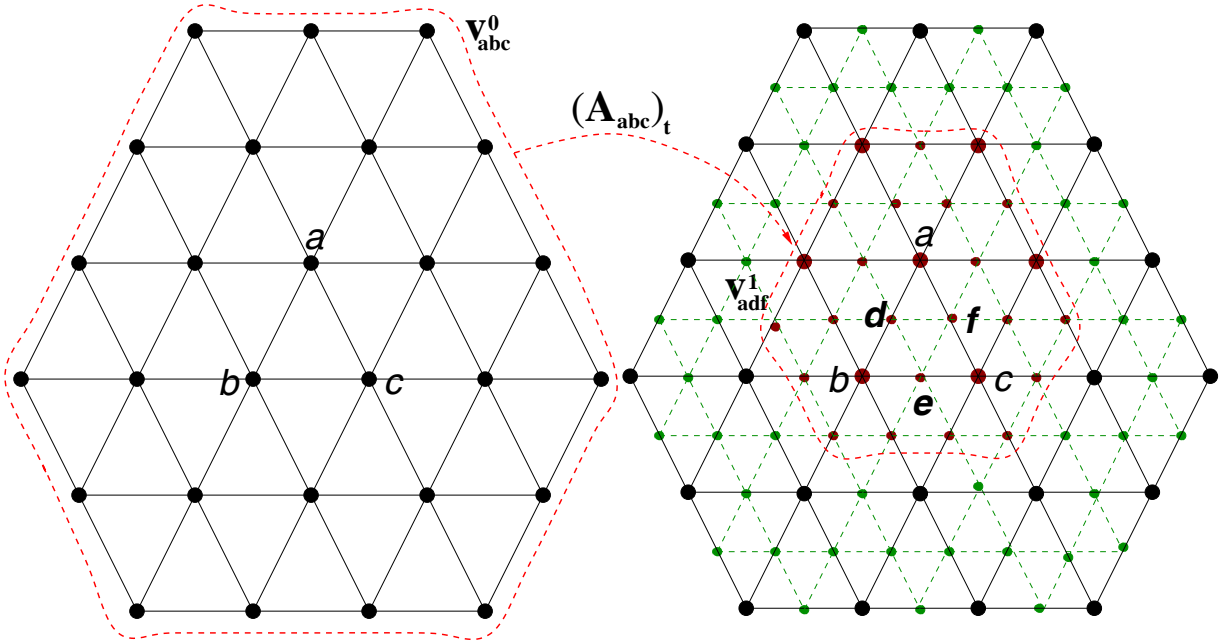
$$\begin{aligned} \mathbf{v}^2_{dgi} &= (\mathbf{A}_{bed})_t \mathbf{v}^1_{bed} \\ \mathbf{v}^2_{bhg} &= (\mathbf{A}_{bed})_l \mathbf{v}^1_{bed} \\ \mathbf{v}^2_{eih} &= (\mathbf{A}_{bed})_r \mathbf{v}^1_{bed} \\ \mathbf{v}^2_{ghi} &= (\mathbf{A}_{bed})_m \mathbf{v}^1_{bed} \end{aligned} \tag{4}$$



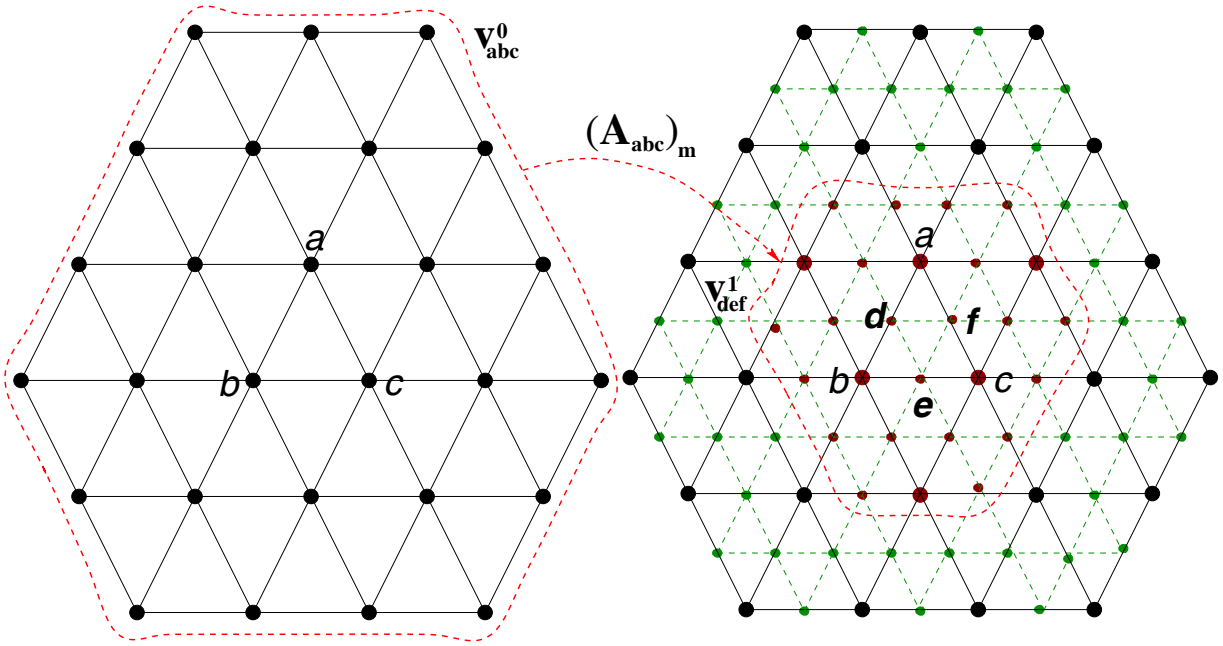
(a)



(b)



(c)



(d)

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.

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

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

Combining Eqn.3 and Eqn.5, we get

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

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

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

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

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

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

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

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

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

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

$$\mathbf{s}_{abc}(\mathbf{x}) = (\lim_{j\to\infty} \mathbf{B}^j_{abc}(\mathbf{x}))\mathbf{v}^0_{abc} = \mathbf{B}_{abc}(\mathbf{x})\mathbf{v}^0_{abc}. \tag{9}$$

In the above equation, $\mathbf{B}_{abc}$ is the collection of basis functions at the vertices of $\mathbf{v}^0_{abc}$. It may also be noted that the modified butterfly subdivision scheme is a stationary subdivision process, and hence new vertex positions are obtained by affine combinations of nearby vertices. This guarantees that each row of the matrices $(\mathbf{A}_{abc})_t, (\mathbf{A}_{abc})_l, (\mathbf{A}_{abc})_r$ and $(\mathbf{A}_{abc})_m$ sums to one. The largest eigenvalue of such matrices is 1 and therefore the limit in Eqn.9 exists. Now, if we assume that the triangular face abc is the k-th face in the initial mesh, then Eqn.9 can be rewritten as

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

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 post-multiplied by $\mathbf{p}$, selects the vertices $\mathbf{v}^0_k$ 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 Eqn.2 and Eqn.10, we get

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

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

C. Dynamics

In the previous section, we have derived an expression of the smooth limit surface obtained via infinite subdivision steps as a function of the control vertex positions in the initial mesh. 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

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

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 [56] and is formulated in an analogous way to that in [39], [40].

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

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

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

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

where (using Eqn.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

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

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

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

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 III.

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

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

The generalized force vector $\mathbf{f}_p$, which can be obtained through the principle of virtual work [56], is expressed as

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

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

C.1 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 (Eqn.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 (Eqn.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 more 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 Eqn.11 and Eqn.12) in the dynamic system will be replaced by a larger number of degrees of freedom $\mathbf{q}$, where $\mathbf{q} = \mathbf{A}\mathbf{p}$. $\mathbf{A}$ 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 II-A for the weights). Thus, $\mathbf{p}$, expressed as a function of $\mathbf{q}$, can be written as

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

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

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

and

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

respectively. Now we 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 Eqn.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

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

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.

It may be noted 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.

III. Finite Element Implementation

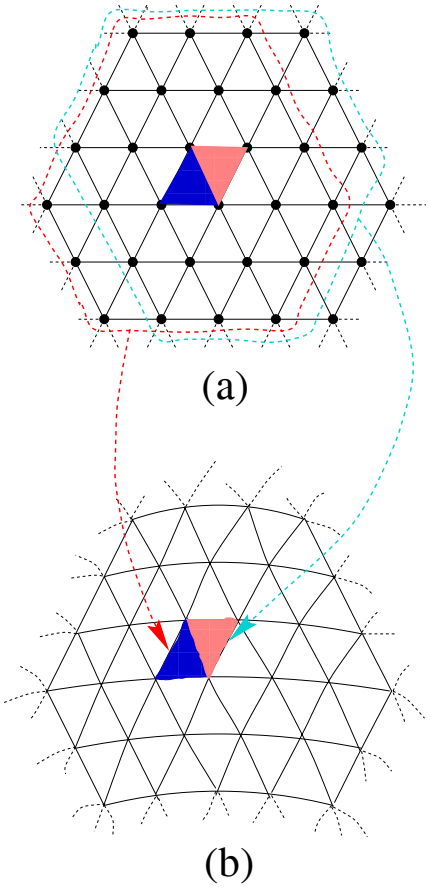


Fig. 7. (a) An initial mesh, and (b) the corresponding limit surface. The domains of the shaded elements in the limit surface are the corresponding triangular faces in the initial mesh. The encircled vertices in (a) are the degrees of freedom for the corresponding element.

In the previous section, we have expressed the smooth limit surface as a function of the control vertex positions in the initial mesh, and have assigned mass and damping distribution, internal deformation energy and forces to the limit surface to develop the corresponding physical model. In this section, we describe the implementation of this physical model using the finite element method.

In Section II 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 Fig.4 and 7). 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 Eqn.17 can be obtained using finite element analysis techniques [57]. We use the same example as in Section II (refer Fig.5) to develop the related concepts. The concept of elements along with the control vertices and their corresponding domains is further illustrated in Fig.7.

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.

A. Elemental mass and damping matrices

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

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

However, from Eqn.9 we know that the basis functions corresponding to the vertices in $\mathbf{v}^0_{abc}$ 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 Eqn.23 is a challenging problem. In [55], an outline is provided on the computation of these inner products without performing any integration. In this paper, we develop a different yet 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 4 triangular faces). Then the mass matrix can be expressed as

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

The j-th level approximation of the corresponding basis functions can be explicitly evaluated (refer Eqn.8 for an expression of $\mathbf{B}^j_{abc}$). An important point to note is that Eqn.8 involves several matrix multiplications and hence can be very expensive to evaluate. However, the matrix $(\mathbf{A}^j_{abc})(= (\mathbf{A}_{abc})_m \ldots (\mathbf{A}_{abc})_t (\mathbf{A}_{abc})_l)$ in Eqn.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 $(\mathbf{A}^j_{abc})$ 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 Eqn.24 by using discrete mass density function which has non-zero 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

$$\mathbf{M}_{abc} = \sum_{i=1}^{k} \mu(\mathbf{v}^j_i)\{\mathbf{B}^j_{abc}(\mathbf{v}^j_i)\}^T\{\mathbf{B}^j_{abc}(\mathbf{v}^j_i)\}, \tag{25}$$

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.

B. Elemental stiffness matrix

We assign the internal energy to each element in the limit surface, thereby defining the internal energy of the smooth subdivision surface model. We take a similar approach as in the derivation of the elemental mass and damping matrix and assign the internal energy to a j-th level approximation of the element.

In this paper, we use three types of internal energy – tension, stiffness and spring energy. For the examples used throughout the paper, this energy at the j-th level of approximation can be written as

$$E_{abc} \approx E^j_{abc} = (E^j_{abc})_t + (E^j_{abc})_{st} + (E^j_{abc})_{sp}, \tag{26}$$

where the subscripts t, st and sp denote tension, stiffness and spring respectively.

The expression for the tension energy, which is essentially equivalent to the first order strain (membrane) energy [58], is

$$\begin{aligned}(E^j_{abc})_t &= \frac{1}{2}k_t \sum_{\Omega} |\mathbf{v}^j_l - \mathbf{v}^j_m|^2 \\ &= \frac{1}{2}k_t\{\mathbf{v}^j_{abc}\}^T(\mathbf{K}^j_{abc})_t\{\mathbf{v}^j_{abc}\},\end{aligned} \tag{27}$$

where k_t is a constant, $\mathbf{v}^j_l$ and $\mathbf{v}^j_m$, 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, and $\mathbf{v}^j_{abc}$ is 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.

Similarly, the expression for stiffness energy, which is equivalent to the second order strain (thin plate) energy [58], can be written as

$$\begin{aligned}(E^j_{abc})_{st} &= \frac{1}{2}k_{st}\sum_{\Omega}|\mathbf{v}^j_l - 2\mathbf{v}^j_m + \mathbf{v}^j_n|^2 \\ &= \frac{1}{2}k_{st}\{\mathbf{v}^j_{abc}\}^T(\mathbf{K}^j_{abc})_{st}\{\mathbf{v}^j_{abc}\},\end{aligned} \tag{28}$$

where $\mathbf{v}^j_l$, $\mathbf{v}^j_m$ and $\mathbf{v}^j_n$ are the three vertices of a triangular face. The summation involves three terms corresponding to each triangular face, and is over all the triangular faces in the mesh at the j-th level of subdivision.

Finally, we add a spring energy term which is given by

$$\begin{aligned}(E^j_{abc})_{sp} &= \frac{1}{2}\sum_{\Omega}\frac{(k_{lm})_{sp}(|\mathbf{v}^j_l - \mathbf{v}^j_m| - \ell_{lm})}{|\mathbf{v}^j_l - \mathbf{v}^j_m|}(\mathbf{v}^j_l - \mathbf{v}^j_m) \\ &= \frac{1}{2}\{\mathbf{v}^j_{abc}\}^T(\mathbf{K}^j_{abc})_{sp}\{\mathbf{v}^j_{abc}\},\end{aligned} \tag{29}$$

where $(k_{lm})_{sp}$ is the spring constant, Ω is the domain as in Eqn.27 and ℓ_{lm} is the natural length of the spring connected between $\mathbf{v}^j_l$ and $\mathbf{v}^j_m$. It may be noted that the entries in $(\mathbf{K}^j_{abc})_{sp}$ depend on the distance between the connected vertices and hence, $(\mathbf{K}^j_{abc})_{sp}$, unlike other elemental matrices, is a function of time which needs to be recomputed in each time step.

Combining the expressions for tension, stiffness and spring energy, we get

$$\begin{aligned}E^j_{abc} &= \frac{1}{2}\{\mathbf{v}^j_{abc}\}^T\{k_t(\mathbf{K}^j_{abc})_t + k_{st}(\mathbf{K}^j_{abc})_{st} + (\mathbf{K}^j_{abc})_{sp}\}\{\mathbf{v}^j_{abc}\} \\ &= \frac{1}{2}\{\mathbf{v}^j_{abc}\}^T(\mathbf{K}^j_{abc})\{\mathbf{v}^j_{abc}\} \\ &= \frac{1}{2}\{(\mathbf{A}^j_{abc})\{\mathbf{v}^0_{abc}\}\}^T(\mathbf{K}^j_{abc})(\mathbf{A}^j_{abc})\{\mathbf{v}^0_{abc}\} \\ &= \frac{1}{2}\{\mathbf{v}^0_{abc}\}^T\{(\mathbf{A}^j_{abc})^T(\mathbf{K}^j_{abc})(\mathbf{A}^j_{abc})\}\{\mathbf{v}^0_{abc}\},\end{aligned} \tag{30}$$

where $(\mathbf{A}^j_{abc})$ and $\mathbf{v}^0_{abc}$ are same as in Eqn.8. Therefore, the expression for the elemental stiffness matrix is given by

$$\mathbf{K}_{abc} = (\mathbf{A}^j_{abc})^T(\mathbf{K}^j_{abc})(\mathbf{A}^j_{abc}). \tag{31}$$

It may be noted that the matrix multiplications for constructing $\mathbf{K}_{abc}$ are avoided in the implementation by following the same technique described in Section III-A.

C. Force Application

The force $\mathbf{f}(\mathbf{x}, t)$ in Eqn.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

$$\mathbf{f}(\mathbf{x}, t) = \int_{\mathbf{x} \in S^j} k(\mathbf{d}_0 - \mathbf{s}(\mathbf{x}, t))\delta(\mathbf{x} - \mathbf{x}_0)d\mathbf{x}, \tag{32}$$

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 on 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.

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 [59] 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

$$\mathbf{f}(x, y, z) = \lambda \frac{\bigtriangledown P(x, y, z)}{\| \bigtriangledown P(x, y, z) \|}, \tag{33}$$

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 Eqn.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. It may be noted 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.

D. Discrete Dynamic Equation

The differential equation given by Eqn.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

$$\ddot{\mathbf{p}}(t+\Delta t) = \frac{\mathbf{p}(t+\Delta t) - 2\mathbf{p}(t) + \mathbf{p}(t-\Delta t)}{\Delta t^2}, \tag{34}$$

and

$$\dot{\mathbf{p}}(t+\Delta t) = \frac{\mathbf{p}(t+\Delta t) - \mathbf{p}(t-\Delta t)}{2\Delta t}. \tag{35}$$

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 Eqn.17, Eqn.34 and Eqn.35, the discrete equation of motion is obtained as

$$(2\mathbf{M} + \mathbf{D}\Delta t + 2\Delta t^2\mathbf{K})\mathbf{p}(t+\Delta t) = 2\Delta t^2\mathbf{f}_p(t+\Delta t) + (\mathbf{D}\Delta t - 2\mathbf{M})\mathbf{p}(t-\Delta t) + 4\mathbf{M}\mathbf{p}(t). \tag{36}$$

This linear system of equations is solved iteratively between each time step using the conjugate gradient method [60], [61].

E. Model Subdivision

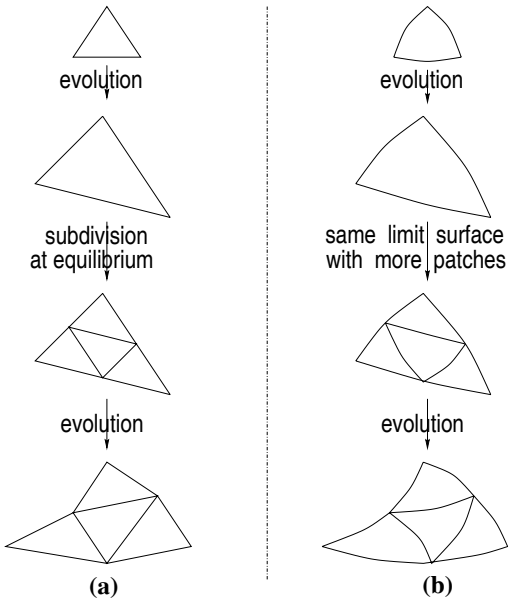


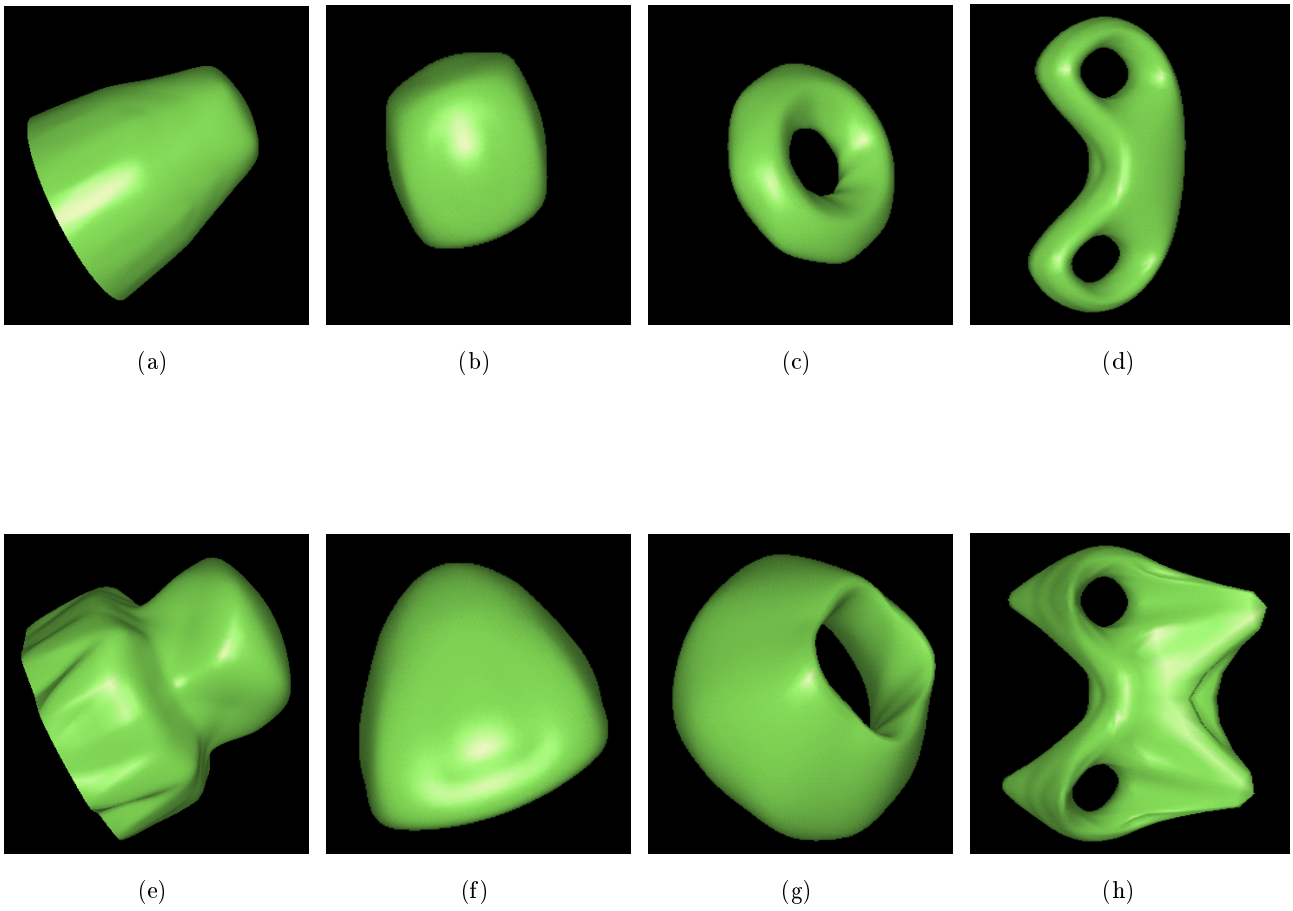
Fig. 8. Model subdivision to increase the degrees of freedom : (a) evolution of the initial mesh, and (b) the corresponding limit surface evolution perceived by the user.

The initialized model grows dynamically according to the equation of motion (Eqn.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 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.8. 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 pre-specified tolerance.

IV. Results

The proposed dynamic (modified) 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 the proposed dynamic (modified) butterfly subdivision surface model. Before illustrating the application results of the proposed model, we would like to point out the advantages of our model over the existing techniques. It is known that shapes of arbitrary topology can be modeled either by explicit patching or by subdivision surfaces. Continuity constraints across patches need to be imposed with care while modeling arbitrary topology using explicit patching techniques. Subdivision surfaces do not face this problem, but one has to manipulate control vertex positions at various levels of subdivision to obtain a desired smooth shape of arbitrary topology using subdivision surfaces. Our dynamic subdivision surface model overcomes this problem by allowing the user to obtain the desired effect on the limit surface by direct sculpting via the application of synthesized forces. To recover shapes from a given set of points in 3D, the existing subdivision surface based tech-



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 9. (a), (b), (c) and (d) : Initial shapes; (e), (f), (g) and (h) : the corresponding modified shapes after interactive sculpting via force application.

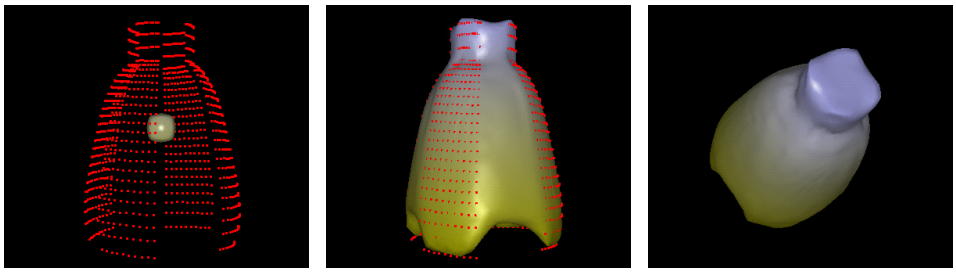
niques resort to complex techniques to derive a mesh for the underlying shape, and then typically mesh optimization techniques are used to obtain a compact representation of the same. Our model recovers the shape from a set of points in an efficient hierarchical way – any simple mesh of the same topology 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 prescribed error of fit is obtained. 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.9,

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. The initial mesh of Fig.9(a) has 125 faces and 76 vertices (degrees of freedom), the initial mesh of the closed cube-like shape in Fig.9(b) has 24 faces and 14 vertices. The one hole torus in Fig.9(c) and the corresponding modified shape in Fig.9(g) have initial meshes with 64 faces and 32 vertices. A two hole torus with a control mesh of 272 faces and 134 vertices, shown in Fig.9(d), is dynamically sculpted to the shape shown in Fig.9(h).

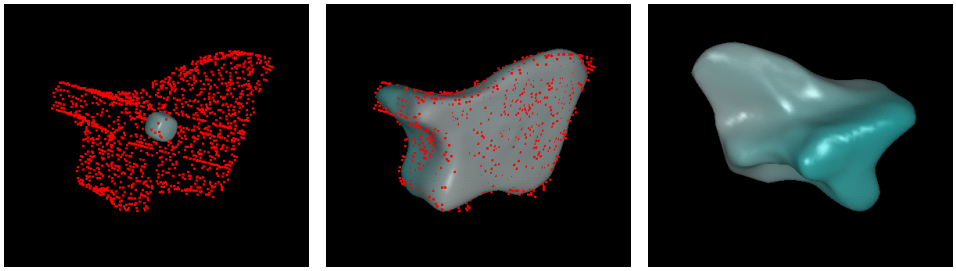
We have performed several experiments testing the applicability of our model to recover the underlying shapes in range and volume data sets. In all the experiments, the initialized model has a control mesh comprising of 24 triangular faces and 14 vertices whereas the control mesh of the fitted model has 384 triangular faces and 194 vertices. It may be noted that 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 (defined as the maximum distance between a data point and the nearest point on the limit surface expressed as a percentage of the diameter of the smallest sphere enclosing the object) of approximately 3%, the initialized model is refined twice following the technique described in Section III-E. 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.

In the first shape recovery experiment, we depict the laser range data acquired from multiple views of a light bulb in Fig.10(a). Prior to applying our algorithm, the data were transformed into a single reference coordinate system. The model was initialized inside the 1000 range data points on the surface of the bulb. The fitted dynamic (modified) butterfly subdivision surface model is shown in Fig.10(b) and (c). In the next experiment, the shape of a mechanical part is recovered from a range data set having 2031 data points (Fig.11). We also recover the shape of a human head from a range data set as shown in Fig.12. The head range data set has 1779 points in 3D. It may be noted 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



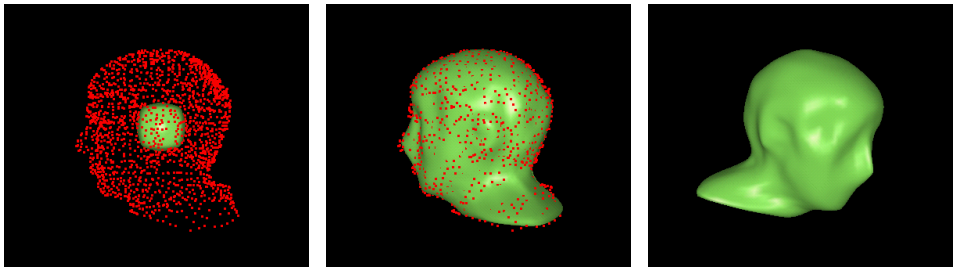
(a) (b) (c)

Fig. 10. (a) Range data of a bulb along with the initialized model, (d) the fitted dynamic butterfly subdivision model, and (c) visualization of the shape from another view point.



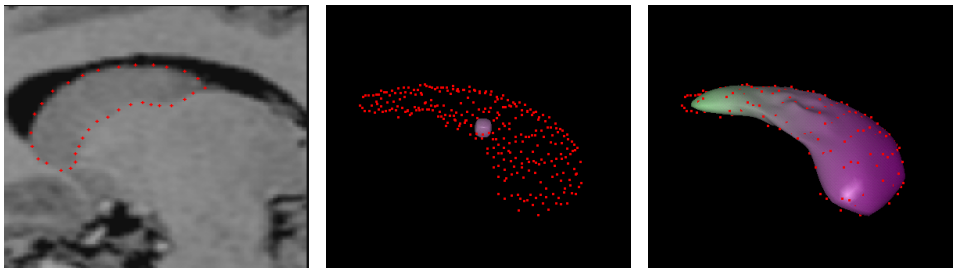
(a) (b) (c)

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



(a) (b) (c)

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



(a) (b) (c)

Fig. 13. (a) Data points identifying the boundary of the caudate nucleus on a MRI slice of human brain, (b) data points (from all slices) in 3D along with the initialized model, and (c) the fitted dynamic butterfly subdivision model.

data points present in the original range data set.

We present the recovery of the caudate nucleus (a cortical structure in human brain) from 64 MRI slices, each of size 256×256 in our next experiment. Fig.13(a) depicts a slice from this MRI scan along with a sparse set of points placed by an expert neuroscientist on the boundary of the shape of interest. Fig.13(b) 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.13(c). We like to point out the fact that the recovered shape in [41] 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.

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 point-based forces. Once the shape is recovered from one data set, this fitted model

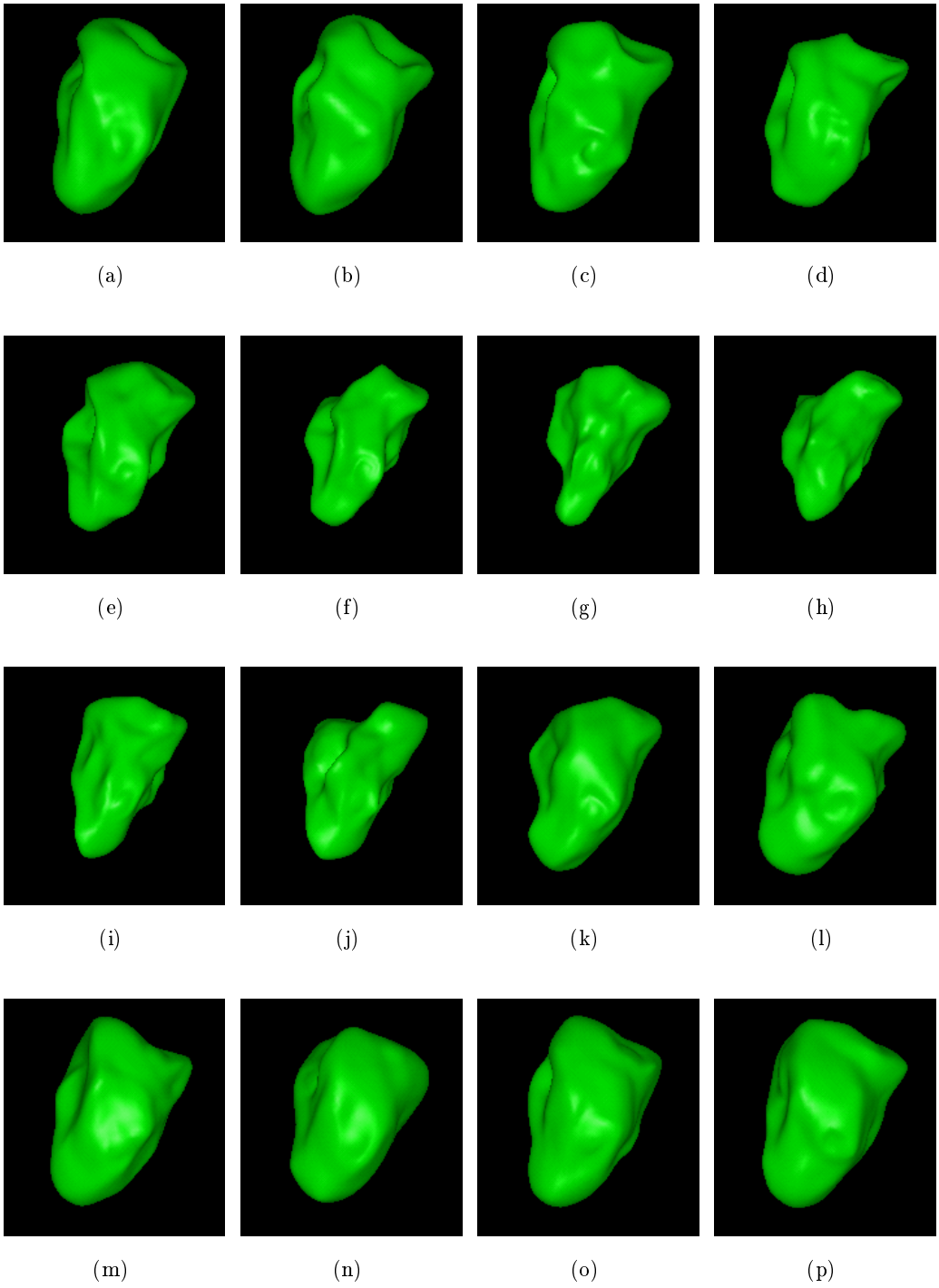


Fig. 14. Snapshots from the animation of canine heart motion over a cardiac cycle using the dynamic butterfly subdivision model.

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.14 for the 16 volume data sets. It may be noted that the control mesh describing the smooth surfaces shown in Fig.14 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.

V. Conclusions

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 the underlying shapes in large range and volume data sets. The proposed model has also been used successfully for non-rigid 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 scheme. Moreover, we have introduced an efficient hierarchical dynamic control for various applications. Our experiments indicate a promising future of the proposed model in computer graphics, geometric modeling and scientific visualization. Furthermore, the finite element techniques proposed in this paper should be of great interest to the engineering design and analysis community as well.

VI. Acknowledgments

This research was supported in part by the NSF grant ECS-9210648 and the NIH grant RO1-LM05944 to B.C. Vemuri, the NSF CAREER award CCR-9702103 and DMI-9700129 to H. Qin. 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," in *Proceedings on Interactive Techniques in Computer Aided Design*, 1978, pp. 157 – 165.

[3] M. Sabin, *The use of piecewise forms for the numerical representation of shape*, Ph.D. 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*, M.S. thesis, University of Utah, Department of Mathematics, 1987.

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

[7] M. Halstead, M. Kass and T. DeRose, "Efficient, fair interpolation using Catmull-Clark surfaces," in *Computer Graphics Proceedings*, ACM SIGGRAPH, August 1993, Annual Conference Series, pp. 35 – 44.

[8] J. Peters and U. Reif, "The simplest subdivision scheme for smoothing polyhedra," *ACM Transactions on Graphics*, vol. 16, no. 4, pp. 420 – 431, October 1997.

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

[10] T. DeRose, M. Kass and T. Truong, "Subdivision surfaces in character animation," in *Computer Graphics Proceedings*, ACM SIGGRAPH, July 1998, Annual Conference Series, pp. 85 – 94.

[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, April 1990.

[12] N. Dyn, S. Hed and D. Levin, "Subdivision schemes for surface interpolation," in *Workshop in Computational Geometry*, A. C. et al., Ed., pp. 97 – 118. 1993.

[13] D. Zorin, P. Schröder and W. Sweldens, "Interpolating subdivision for meshes with arbitrary topology," in *Computer Graphics Proceedings*, ACM SIGGRAPH, August 1996, Annual Conference Series, pp. 189 – 192.

[14] L. Kobbelt, "Interpolatory refinement by variational methods," in *Approximation Theory VIII*, C. Chui and L. Schumaker, Eds., vol. 2 of *Wavelets and Multilevel Approximation*, World Scientific Publishing Co., 1995, pp. 217 – 224.

[15] L. Kobbelt, "A variational approach to subdivision," *Computer-Aided Geometric Design*, vol. 13, pp. 743 – 761, 1996.

[16] L. Kobbelt and P. Schröder, "Constructing variationally optimal curves through subdivision," Tech. Rep. CS-TR-97-05, California Institute of Technology Computer Science Department Technical Report, 1997.

[17] 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.

[18] A.A. Ball and D.J.T. Storry, "Conditions for tangent plane continuity over recursively generated B-spline surfaces," *ACM Transactions on Graphics*, vol. 7, no. 2, pp. 83 – 102, 1988.

[19] A.A. Ball and D.J.T. Storry, "An investigation of curvature variations over recursively generated B-spline surfaces," *ACM Transactions on Graphics*, vol. 9, no. 4, pp. 424 – 437, 1990.

[20] U. Reif, "A unified approach to subdivision algorithms near extraordinary points," *Computer Aided Geometric Design*, vol. 12 , no. 2, pp. 153 – 174, 1995.

[21] J.E. Schweitzer, *Analysis and Application of Subdivision Surfaces*, Ph.D. thesis, University of Washington, Seattle, 1996.

[22] A. Habib and J. Warren, "Edge and vertex insertion for a class of C^1 subdivision surfaces," *Computer Aided Geometric Design*, to appear.

[23] J. Peters and U. Reif, "Analysis of generalized B-spline subdivision algorithms," *SIAM Journal of Numerical Analysis*, to appear, available at ftp://ftp.cs.purdue.edu/pub/jorg/9697agb.ps.Z.

[24] D. Zorin, "Smoothness of stationary subdivision on irregular meshes," *Constructive Approximation*, submitted, available as Stanford Computer Science Lab Tech. Rep. CSL-TR-98-752.

[25] J. Stam, "Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values," in *Computer Graphics Proceedings*, ACM SIGGRAPH, July 1998, Annual Conference Series, pp. 395 – 404.

[26] D. Terzopoulos, J. Platt, A. Barr and K. Fleischer, "Elastically deformable models," in *Computer Graphics Proceedings*, ACM SIGGRAPH, 1987, Annual Conference Series, pp. 205 – 214.

[27] D. Terzopoulos and K. Fleischer, "Deformable models," *The Visual Computer*, vol. 4, no. 6, pp. 306 – 331, 1988.

[28] A. Pentland and J. Williams, "Good vibrations : Modal dynamics for graphics and animation," in *Computer Graphics Proceedings*, ACM SIGGRAPH, 1989, Annual Conference Series, pp. 215 – 222.

[29] D. Metaxas and D. Terzopoulos, "Dynamic deformation of solid primitives with constraints," in *Computer Graphics Proceedings*, ACM SIGGRAPH, July 1992, Annual Conference Series, pp. 309 – 312.

[30] B.C. Vemuri and A. Radisavljevic, "Multiresolution stochastic hybrid shape models with fractal priors," *ACM Transactions on Graphics*, vol. 13, no. 2, pp. 177 – 207, April 1994.

[31] G. Celniker and D. Gossard, "Deformable curve and surface finite elements for free-form shape design," in *Computer Graphics Proceedings*, ACM SIGGRAPH, July 1991, Annual Conference Series, pp. 257 – 266.

[32] 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.

[33] 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.

[34] G. Celniker and W. Welch, "Linear constraints for deformable B-spline surfaces," in *Proceedings of the Symposium on Interactive 3D Graphics*, pp. 165 – 170. ACM, New York, 1992.

[35] W. Welch and A. Witkin, "Variational surface modeling," in *Computer Graphics Proceedings*, ACM SIGGRAPH, July 1992, Annual Conference Series, pp. 157 – 166.

[36] 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.

[37] H. Qin and D. Terzopoulos, "D-NURBS : A physics-based framework for geometric design," *IEEE Transactions on Visualization and Computer Graphics*, vol. 2, no. 1, pp. 85–96, March 1996.

[38] D. Terzopoulos and H. Qin, "Dynamic NURBS with geometric constraints for interactive sculpting," *ACM Transactions on Graphics*, vol. 13, no. 2, pp. 103 – 136, April 1994.

[39] C. Mandal, H. Qin and B.C. Vemuri, "Dynamic smooth subdivision surfaces for data visualization," in *IEEE Visualization'97 Conference Proceedings*, Phoenix,AZ, October 1997, pp. 371 – 377.

[40] H. Qin, C. Mandal and B.C. Vemuri, "Dynamic Catmull-Clark subdivision surfaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 4, no. 3, September 1998, to appear.

[41] C. Mandal, B.C. Vemuri and H. Qin, "Shape recovery using dynamic subdivision surfaces," in *Proceedings of the International Conference on Computer Vision*, Bombay, India, January 1998, pp. 805 – 810.

[42] E. Bardinet, L.D. Cohen and N. Ayache, "Superquadrics and free-form deformations : A global model to fit and track 3D medical data," in *Lecture Notes in Computer Science, CVRMed'95*, N. Ayache, Ed., Berlin, Germany, 1995, vol. 905, pp. 319 – 326, Springer - Verlag.

[43] L.D. Cohen and I. Cohen, "Finite-element methods for active contour models and balloons for 2D and 3D images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1131 – 1147, November 1993.

[44] E. Koh, D. Metaxas and N. Badler, "Hierarchical shape representation using locally adaptive finite elements," in *Lecture Notes in computer Science, Computer Vision - ECCV 1994*, J.O. Eklundh, Ed., Berlin, Germany, 1994, vol. 800, pp. 441 – 446, Springer-Verlag.

[45] F. Leitner and P. Cinquin, "Complex topology 3D objects segmentation," in *Model-based Vision Development*

and Tools, SPIE Proceedings, Bellingham, WA, 1991, SPIE, vol. 1609, pp. 16 – 26.

[46] 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.

[47] L.H. Staib and J.S. Duncan, "Boundary finding with parametrically deformable models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1061 – 1075, 1992.

[48] D. Terzopoulos and D. Metaxas, "Dynamic 3D models with local and global deformations: Deformable superquadrics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, pp. 703 – 714, July 1991.

[49] D. Metaxas and D. Terzopoulos, "Shape and non-rigid motion estimation through physics-based synthesis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 6, pp. 580 – 591, June 1993.

[50] A. Pentland and B. Horowitz, "Recovery of non-rigid motion and structure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, pp. 730 – 742, July 1991.

[51] Y. Chen and G. Medioni, "Surface description of complex objects from multiple range images," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Seattle,WA, June 1994, pp. 153 – 158.

[52] W.C. Huang and D. Goldgof, "Adaptive-size physically-based models for non-rigid motion analysis," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Urbana-Champaign, IL, June 1992, pp. 833 – 835.

[53] H. Tanaka and F. Kishino, "Adaptive mesh generation for surface reconstruction : Parallel hierarchical triangulation without discontinuities," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, New York City, NY, June 1993, pp. 88 – 94.

[54] M. Vasilescu and D. Terzopoulos, "Adaptive meshes and shells : Irregular triangulation, discontinuities and hierarchical subdivision," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Urbana-Champaign, IL, June 1992, pp. 829 – 832.

[55] J.M. Lounsbery, T. DeRose and J. Warren, "Multiresolution analysis for surfaces of arbitrary topological type," *ACM Transactions on Graphics*, vol. 16, no. 1, pp. 34 – 73, January 1997.

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

[57] H. Kardestuncer, *The Finite Element Handbook*, McGraw-Hill, New York, 1987.

[58] D. Terzopoulos, "Regularization of inverse visual problems involving discontinuities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 4, pp. 413 – 424, 1986.

[59] O. Monga and R. Deriche, "3D edge detection using recursive filtering," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, June 1989, pp. 28 – 35.

[60] G.H. Golub and V.H. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, 1989.

[61] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes in C*, Cambridge University Press, 1992.