Dynamic Catmull-Clark Subdivision Surfaces

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

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

Abstract

Recursive subdivision schemes have been extensively used in computer graphics, computer-aided geometric design and scientific visualization for modeling smooth surfaces of arbitrary topology. Recursive subdivision generates a visually pleasing smooth surface in the limit from an initial user-specified polygonal mesh through the repeated application of a fixed set of subdivision rules. In this paper, we present a new dynamic surface model based on the Catmull-Clark subdivision scheme, a popular technique for modeling complicated objects of arbitrary genus. Our new dynamic surface model inherits the attractive properties of the Catmull-Clark subdivision scheme as well as those of the physics-based models. This new model provides a direct and intuitive means of manipulating geometric shapes, and an efficient hierarchical approach for recovering complex shapes from large range and volume data sets using very few degrees of freedom (control vertices). We provide an analytic formulation and introduce the "physical" quantities required to develop the dynamic subdivision surface model which can be interactively deformed by applying synthesized forces. The governing dynamic differential equation is derived using Lagrangian mechanics and the finite element method. Our experiments demonstrate that this new dynamic model has a promising future in computer graphics, geometric shape design and scientific visualization.

Keywords

Computer Graphics, CAGD, Visualization, Subdivision Surfaces, Deformable Models, Dynamics, Finite Elements, Interactive Techniques.

I. INTRODUCTION

Generating smooth surfaces of arbitrary topology is a grand challenge in geometric modeling, computer graphics and scientific visualization. The recursive subdivision scheme is well suited for this purpose. In [1], Chaikin introduced the idea 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 in geometric modeling after

Chaikin's pioneering work on curve generation. A recursive subdivision algorithm typically generates a smooth surface which is the limit of a sequence of recursively refined polyhedral surfaces based on an user-defined initial control mesh. At each step of the subdivision, a *finer* polyhedral surface with more vertices and faces will be constructed from the previous one via a refinement process. In general, these subdivision schemes can be categorized into two distinct classes namely, (1) approximating subdivision techniques and (2) interpolating subdivision techniques, which are discussed briefly in the next section.

A. Background

Among the approximating schemes, the techniques of Doo and Sabin [2], [3] and Catmull and Clark [4] generalize the idea of obtaining 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 non-rectangular mesh, can be reduced to a set of standard B-spline patches except at a finite number of degenerate points which are also known as *extraordinary* points. The extraordinary points on the limit surface correspond to the *extraordinary* vertices (vertices whose degree is not equal to four) in the control mesh. 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. Recently, Peters and Reif [8] have proposed a simple subdivision scheme for smoothing polyhedra. All the schemes mentioned above generalize recursive subdivision schemes for generating limit surfaces with known parameterization.

The most well-known interpolation-based subdivision scheme is the "butterfly" algorithm proposed by Dyn et al. [9]. 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 [10]. Zorin et al. [11] has developed an improved interpolatory subdivision 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 wide applications in wavelets on manifolds, multiresolution decomposition of polyhedral surfaces and multiresolution editing. A variational approach for interpolatory refinement has been proposed by Kobbelt [12], [13] and by Kobbelt and Schröder [14]. 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 thereby destroying the local refinement property which makes the subdivision schemes attractive for implementation in the computer graphics applications.

The mathematical derivation of the smooth limit surface generated by the subdivision algorithms is rather complex. Doo and Sabin [15] first analyzed the smoothness behavior of the limit surface using Fourier transforms and an eigen-analysis of the subdivision matrix. Ball and Storry [16], [17] and Reif [18] 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. Most recently, specific schemes were analyzed by Schweitzer [19], Habib and Warren [20], Peters and Reif [21] and Zorin [22].

B. Motivation

Although recursive subdivision surfaces are extremely powerful to represent 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 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 making geometric objects satisfy global smoothness requirements 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 physical laws are of particular interest in this context. These models respond dynamically to applied forces in a very intuitive manner. Time is fundamental to the dynamic formulation. The dynamic formulation marries the geometry with time, mass, force, and constraint. Dynamic models produce smooth, natural motions which are familiar and easy to control. In addition, they facilitate interaction – especially direct manipulation of complex geometric models. Furthermore, the equilibrium state of the model is characterized by a minimum of the potential energy of the model subject to imposed constraints. The potential 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 modeling capabilities in an elegant formulation which grounds everything in real-world physics.

Free-form deformable models were first introduced to computer graphics and visualization in Terzopoulos et al. [23] and further developed by Terzopoulos and Fleischer [24], Pentland and Williams [25], Metaxas and Terzopoulos [26] and Vemuri and Radisavljevic [27]. Celniker and Gossard [28] developed a system for interactive free-form design based on the finite element optimization of energy functionals proposed in [24]. Bloor and Wilson [29], [30], Celniker and Welch [31] and Welch and Witkin [32] 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. Recently, Qin and Terzopoulos [33], [34], [35] have 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 parametric domain. Hence, it can be very difficult to model surfaces of arbitrary genus using these models. In this paper, we develop a dynamic generalization of a recursive subdivision scheme based on Catmull-Clark subdivision surfaces. Our new dynamic model combines the benefits of subdivision surfaces for modeling arbitrary topology as well as those of the dynamic splines for direct and interactive shape manipulation by applying synthesized forces. Note that, the derivation of our dynamic subdivision surface poses a significant technical challenge because of the fact that no closed-form parameterization of the limit surface exists near the extraordinary points. The primary contribution of this paper is to develop a dynamic framework based on subdivision schemes for directly manipulating the smooth limit surfaces of arbitrary topology, so that the users do not have to be concerned about how to position and/or refine control vertices. We present the details of our formulation in a later section.

The dynamic Catmull-Clark subdivision surface has been developed primarily for modeling arbitrary topology. However, another important application of the developed model is in shape recovery. In a typical shape reconstruction application, we need to recover shapes of arbitrary topology from large data sets. Physics-based models are often used for this purpose. However, the model used for fitting should be able to recover the shape accurately. At the same time the number of degrees of freedom for model representation should be kept low. Another important criterion is that the model initialization should not be restricted to parameterized input meshes since it is infeasible to globally parameterize shapes of arbitrary topology over a rectangular domain. A physics-based model satisfying the aforementioned criteria is a good candidate for a solution to the shape recovery problem.

Physics-based deformable models used to solve shape recovery problem involve either fixed size [27], [36], [37], [38], [39] or adaptive size [40], [41], [42], [43], [44], [45] grids. The models with fixed grid size generally use fewer degrees of freedom for representation, but the accuracy of the recovered shape is lacking in many cases. On the other hand, the number of degrees of freedom used for shape representation is generally very high, and computationally expensive ad hoc schemes are used in models with adaptive grid size methods. However, the recovered shapes are accurate. The hierarchical shape representation using locally adaptive finite elements discussed in [42] can efficiently represent the shape of an object of genus zero with a small number of nodal points, but this scheme can not be easily extended to cope with arbitrary shapes. The balloon model for describing the shape of complex objects [40] also adapts the mesh surface to local surface shapes and is purely driven by an applied inflation force towards the object surface from the interior of the object. This scheme involves a large number of nodal points for representing complex shapes. Moreover, *all the existing models using either a fixed or an adaptive grid size require a globally parameterized mesh as their input.*

The proposed model solves the shape recovery problem very effectively as it can recover shapes from large range and volume data sets using very few degrees of freedom (control vertices) for its representation and can cope with any arbitrary input mesh, not necessarily parameterized, with an arbitrary number of extraordinary points. The initialized model deforms under the influence of synthesized forces to fit the data set by minimizing its energy. Once the approximate shape is recovered, the model is further subdivided automatically and a better approximation to the input data set is achieved using more degrees of freedom. The process of subdivision after achieving an approximate fit is continued till a prescribed error criteria for fitting the data points is achieved.

In a nutshell, the dynamic Catmull-Clark subdivision surface model can be efficiently used to model arbitrary topology where modelers can directly manipulate the smooth limit surface in an intuitive fashion by applying synthesized forces, and to recover the underlying shapes from large range and volume data sets.

C. Overview

The rest of the paper is organized as follows: Section II presents the detailed formulation of the dynamic Catmull-Clark subdivision surfaces. The implementation details are provided in Section III. Experimental results can be found in IV. Finally, we make concluding remarks and point out future directions of research in Section V.

II. FORMULATION

In this section we present a systematic formulation of our new dynamic model based on Catmull-Clark subdivision surfaces. First, we briefly review the Catmull-Clark subdivision scheme. Then, we demonstrate how to assign a bicubic patch in the limit surface to a non-boundary face in a rectangular setting. We further generalize this idea to assign the infinite number of bicubic patches in the limit surface to faces that are in the vicinity of an extraordinary point/vertex. Next, we formulate a closed-form analytical representation of the smooth limit surface which can be viewed as a function of its (initial) polyhedral control vertices. Finally, we introduce physical quantities in our dynamic model to derive its motion equation.

A. Catmull-Clark subdivision surfaces

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

- For each face, introduce a new face point which is the average of all the old vertices defining the face.
- For each (non-boundary) edge, introduce a new edge point which is the average of the following four points: two old vertices defining the edge and two new face points of the faces adjacent to the edge.
- For each (non-boundary) vertex V, introduce a new vertex whose position is $\frac{F}{n} + \frac{2E}{n} + \frac{(n-3)V}{n}$, where F is the average of the new face vertices of all faces adjacent to the old vertex V, E is the average of the midpoints of all edges incident on the old vertex V and n is the number of the edges incident on the vertex.
- Form new edges by connecting each new face point to the new edge points of the edges defining the old face and by connecting each new vertex point to the new edge points of all old edges incident on the old vertex point.
- Define new faces as those enclosed by new edges.

The most important property of Catmull-Clark subdivision surfaces is that a smooth surface can be generated from a control mesh of arbitrary topology. Therefore, this subdivision scheme is extremely valuable for modeling various complicated geometric objects of arbitrary topology. Catmull-Clark subdivision surfaces include standard bicubic B-spline surfaces as their special case (i.e., the limit surface is a tensor-product B-spline surface for a rectangular mesh with all non-boundary vertices of degree 4). In addition, the aforementioned subdivision rules generalize the recursive bicubic B-spline patch subdivision algorithm. For non-rectangular meshes, the limit surface converges to a bicubic B-spline surface except at a finite number of extraordinary points. Note that, after the first subdivision, all faces are quadrilaterals, hence all new vertices created subsequently will have four incident edges. The number of extraordinary points on the limit surface is a constant, and is equal to the number of extraordinary vertices in the refined mesh obtained after applying one step of the Catmull-Clark subdivision on the initial mesh. The limit surface is curvature-continuous everywhere except at extraordinary vertices, where only tangent plane continuity is achieved. In spite of the popularity of Catmull-Clark subdivision surfaces for representing complex geometric shapes of arbitrary topology, these subdivision surfaces may not be easily parameterizable and deriving a closed-form analytic formulation of the limit surface can be very difficult. These deficiencies preclude their immediate pointwise manipulation and hence may restrain the applicability of these schemes. We develop a new dynamic model based on Catmull-Clark subdivision surfaces which offers a closed-form analytic formulation and allows users to manipulate the model directly and intuitively.

To develop the dynamic model which treats the limit smooth surface as a function of its control mesh in a hierarchical fashion, we need to update control vertex positions continually at any given level. However, all the vertices introduced through subdivision are obtained as an affine combination of control vertex positions of the initial mesh. Therefore, we can control the dynamic behavior of the limit surface by formulating the dynamic model on the initial mesh itself, the only exception being the case when the initial mesh has non-rectangular faces. This problem can be circumvented by taking the mesh obtained through one step of subdivision as the initial mesh. To define the limit surface using the vertices of the initial mesh, the enumeration of the bicubic patches in the limit surface is necessary. In the next two sections, we present a scheme of assigning the bicubic patches to various faces of the initial mesh. It may be noted that an additional subdivision step may be needed in some cases to isolate the extraordinary points and the resulting mesh is treated as the initial mesh (a typical example is when the initial mesh is a tetrahedron).

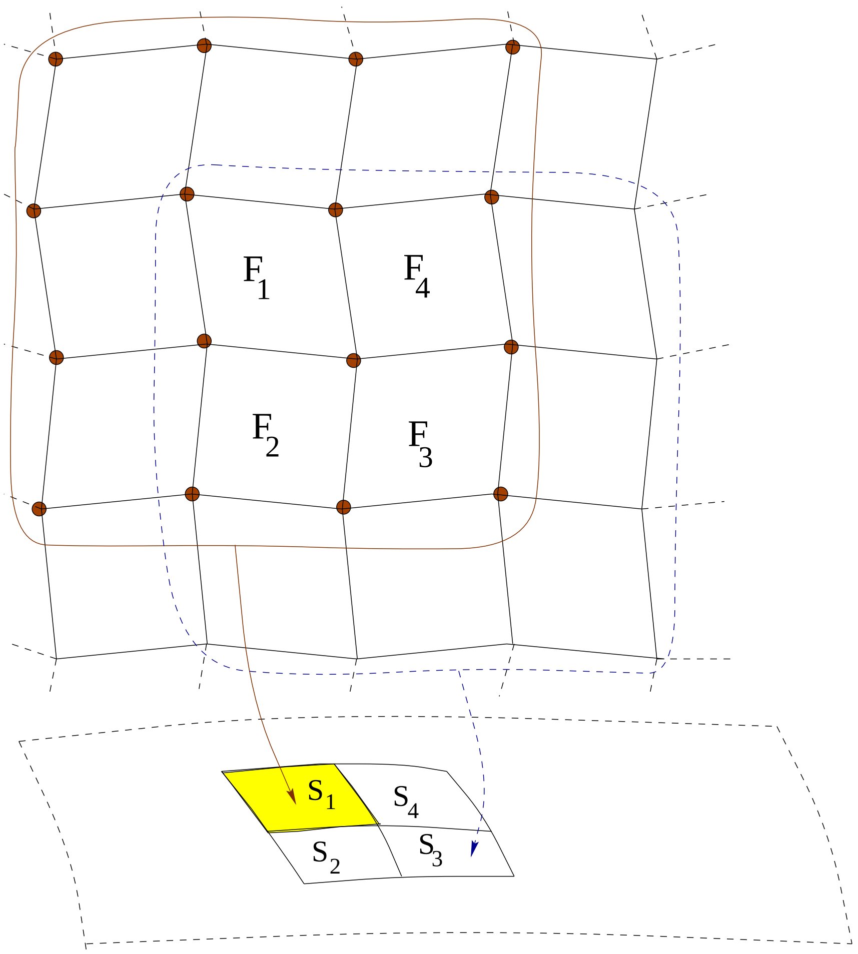


Fig. 1. A rectangular mesh and its limit surface consisting of 4 bicubic surface patches.

B. Assigning patches to regular faces

In Fig.1, a rectangular control mesh is shown along with the bicubic B-spline surface (4 patches) in the limit after an infinite number of subdivision steps. Note that, each of the bicubic patches in the limit surface is defined by a rectangular face with each vertex of degree four, thereby accounting for 16 control points (from its 8 connected neighborhood) needed to define a bicubic surface patch in the limit. Therefore, for each rectangular face in the initial mesh with a degree of 4 at each vertex, the corresponding bicubic surface patch can be assigned to it in a straight forward way. In Fig.1, the surface patches S_1, S_2, S_3 and S_4 are assigned to face F_1, F_2, F_3 and F_4 respectively. The 16 control points for the patch S_1, corresponding to face F_1, are highlighted in Fig.1. Note that, the initial control mesh can be viewed as the parametric domain of the limit surface. Therefore, face F_1 can be thought of as the portion

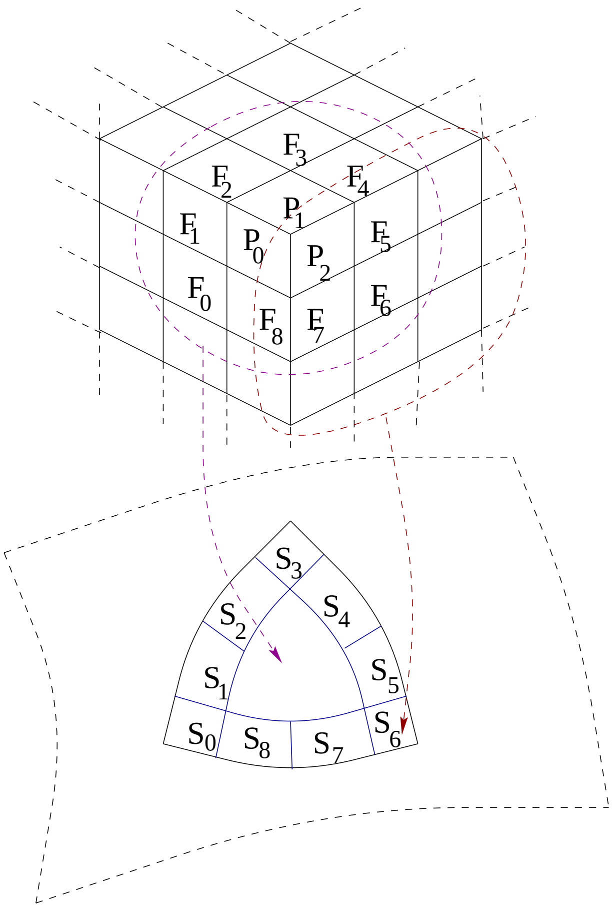


Fig. 2. A mesh with an extraordinary point of degree 3 and its limit surface.

of the parametric domain over which the patch S_1 is defined, i.e., has non-zero values. Nevertheless, each rectangular face (e.g. F_1) can be parametrically defined over $[0,1]^2$, and hence, all bicubic B-spline patches defined by 16 control points are locally parameterized over $[0,1]^2$.

C. Assigning patches to irregular faces

In Fig.2, a mesh containing an extraordinary point of degree 3 and its limit surface are shown. The faces $F_0, F_1, \ldots, F_8$ are assigned to bicubic patches $S_0, S_1, \ldots, S_8$ respectively (as they all have vertices of degree 4) following the aforementioned scheme. However, the central smooth surface enclosed by the patches $S_0, S_1, \ldots, S_8$ consists of infinite number of bicubic patches converging to a point in the limit. We need to develop a recursive way of enumerating these bicubic patches and assigning them to various faces at different levels in order to develop the dynamic subdivision surface model.

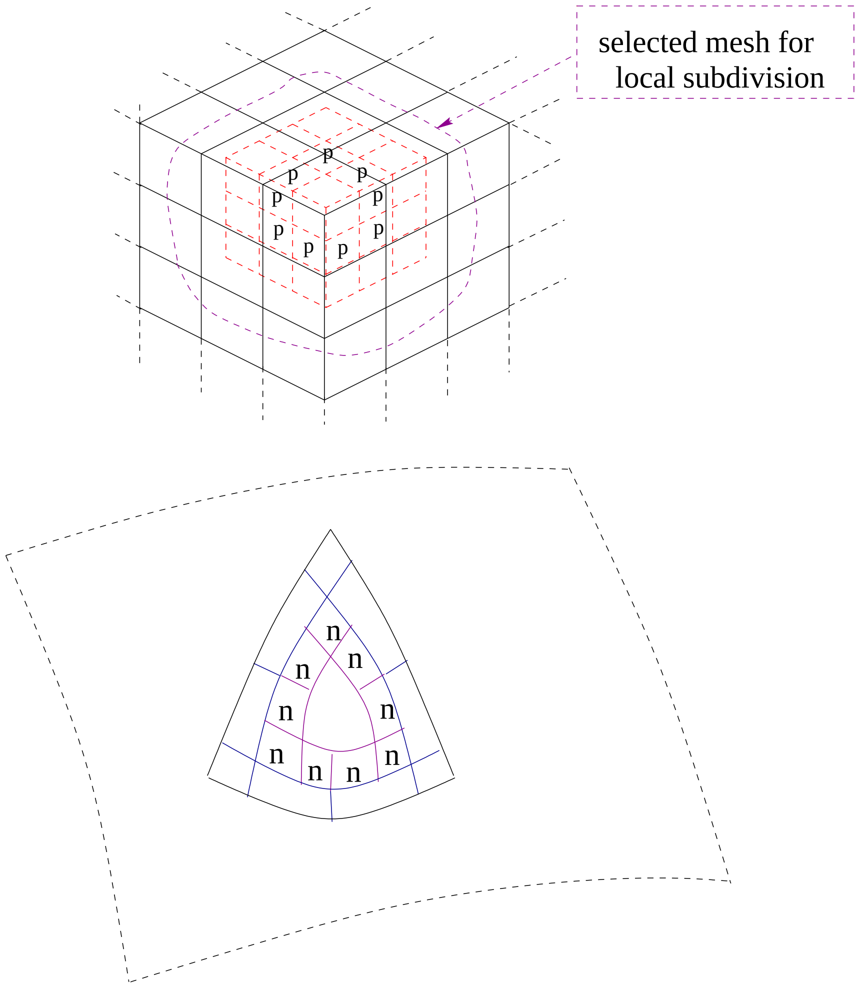


Fig. 3. Local subdivision around the extraordinary point and the limit surface.

The idea of enumerating the bicubic patches corresponding to faces having an extraordinary vertex is shown in Fig.3 where a local subdivision of the mesh consisting of faces $F_0, F_1, ..., F_8, P_0, P_1, P_2$ (and not the other boundary faces) of Fig.2 is carried out. Topologically, the resulting local subdivision mesh (shown as dotted mesh) is exactly the same as the mesh in Fig.2 and hence exactly the same number of bicubic patches can be assigned to its faces with vertices of degree 4 as is evident from Fig.3 (the new faces and the corresponding patches are marked by "p" and "n" respectively). This process of local subdivision and assignment of bicubic patches around an extraordinary point can be carried out recursively and in the limit, the enclosed patch corresponding to faces sharing the extraordinary point will converge to a point. However, there is no need to carry out an infinite number of subdivision steps. This description is for formulation purposes only and the exact implementation will be detailed in a later section.

D. Kinematics of the limit surface

In this section we develop the mathematics for the kinematics of the limit surface via illustrative examples and then present the generalized formulas. We start the illustration with a single bicubic B-spline patch which is obtained as the limiting process of the Catmull-Clark subdivision algorithm applied to an initial 4 by 4 rectangular control mesh. Let $\mathbf{s}_p(u,v)$, where $(u,v) \in [0,1]^2$, denote this bicubic B-spline patch which can be expressed analytically as

$$\mathbf{s}_p(u,v) = (x(u,v), y(u,v), z(u,v))^T = \sum_{i=0}^{3}\sum_{j=0}^{3} \mathbf{d}_{i,j} B_{i,4}(u) B_{j,4}(v) \tag{1}$$

where $\mathbf{d}_{i,j}$ represents a 3-dimensional position vector at the (i,j)th control point location and $B_{i,4}(u)$ and $B_{j,4}(v)$ are the cubic B-spline basis functions. The subscript p on $\mathbf{s}$ denotes the patch under consideration. Expressing eqn.1 in a generalized coordinate system we have

$$\mathbf{s}_p = \mathbf{J}_p \mathbf{q}_p \tag{2}$$

where $\mathbf{J}_p$ is a transformation matrix storing the basis functions of a bicubic B-spline patch, and is of size $(3, 48)$. Vector $\mathbf{q}_p$ is the concatenation of all control points defining a B-spline patch in 3D. Note that in the concatenation of the control points, each control point has an (x, y, z) component. For example, the (x, y, z) components of the control point (i, j) correspond to positions $3k, 3k+1, 3k+2$ - where, $k = 4i + j$ - respectively in the vector $\mathbf{q}_p$. We can express the entries of $\mathbf{J}_p$ explicitly in the following way: $\mathbf{J}_p(0,k) = \mathbf{J}_p(1,k+1) = \mathbf{J}_p(2,k+2) = B_{i,4}(u)B_{j,4}(v)$ and $\mathbf{J}_p(0,k+1) = \mathbf{J}_p(0,k+2) = \mathbf{J}_p(1,k) = \mathbf{J}_p(1,k+2) = \mathbf{J}_p(2,k) = \mathbf{J}_p(2,k+1) = 0$.

D.1 *Limit surface with many bicubic patches from a rectangular initial mesh*

Now let's consider a limit surface consisting of many bicubic surface patches obtained after applying an infinite number of subdivision steps to a rectangular initial mesh. For example, let the limit surface of Fig.1 be $\mathbf{s}_m$ which can be written as

$$\mathbf{s}_m(u,v) = \mathbf{s}_{m_1}(2u, 2v) + \mathbf{s}_{m_2}(2(u-\frac{1}{2}), 2v) + \mathbf{s}_{m_3}(2(u-\frac{1}{2}), 2(v-\frac{1}{2})) + \mathbf{s}_{m_4}(2u, 2(v-\frac{1}{2})) \tag{3}$$

where $\mathbf{s}_{m_1}(2u,2v) = \mathbf{s}_m(u,v)$ for $0 \le u, v \le \frac{1}{2}$, and 0 otherwise. Similarly, $\mathbf{s}_{m_2}, \mathbf{s}_{m_3}$ and $\mathbf{s}_{m_4}$ are also equal to $s_m(u,v)$ for an appropriate range of values of u, v and 0 outside. It may be noted that $\mathbf{s}_{m_1}, \mathbf{s}_{m_2}, \mathbf{s}_{m_3}, \mathbf{s}_{m_4}$ correspond to patches S_1, S_2, S_3, S_4 respectively in Fig.1. Rewriting eqn.3 in generalized coordinates we have

$$\mathbf{s}_m = \mathbf{J}_1\mathbf{q}_1 + \mathbf{J}_2\mathbf{q}_2 + \mathbf{J}_3\mathbf{q}_3 + \mathbf{J}_4\mathbf{q}_4 = \sum_{i=1}^{4} \mathbf{J}_i\mathbf{q}_i \tag{4}$$

where $\mathbf{J}_i$s are the transformation matrices of size $(3, 48)$ and $\mathbf{q}_i$s are the (x,y,z) component concatenation of a subset of the control points of $\mathbf{s}_m$ defining $\mathbf{s}_{m_i}$, $i = 1, 2, 3$ and 4. A more general expression for $\mathbf{s}_m$ is

$$\mathbf{s}_m = \mathbf{J}_1\mathbf{A}_1\mathbf{q}_m + \mathbf{J}_2\mathbf{A}_2\mathbf{q}_m + \mathbf{J}_3\mathbf{A}_3\mathbf{q}_m + \mathbf{J}_4\mathbf{A}_4\mathbf{q}_m = \sum_{i=1}^{4} \mathbf{J}_i\mathbf{A}_i\mathbf{q}_m = \mathbf{J}_m\mathbf{q}_m. \tag{5}$$

Where, $\mathbf{q}_m$ is the 75-component vector of 3D positions of the 25 vertex control mesh defining the limit surface $\mathbf{s}_m$. Matrices $\mathbf{A}_i, 1 \leq i \leq 4$, are of size $(48, 75)$, each row consisting of a single nonzero entry $(= 1)$ and the $(3, 75)$-sized matrix $\mathbf{J}_m = \sum_{i=1}^{4} \mathbf{J}_i\mathbf{A}_i$.

D.2 *Limit surface with many bicubic patches from an arbitrary initial mesh*

The stage is now set to define the limit surface $\mathbf{s}$ using the vertices of initial mesh $\mathcal{M}$ for any arbitrary topology, assuming all faces are rectangular and no face contains more than one extraordinary point as its vertex (i.e., extraordinary points are isolated). As mentioned earlier, if these assumptions are not satisfied, one or two steps of global subdivision may be required and the resulting mesh can be treated as the initial mesh. Let the number of vertices in the initial mesh $\mathcal{M}$ be a, and let l of these be the extraordinary vertices. Let us assume that the number of faces in the initial mesh are b, and that k of these have vertices with degree 4 (henceforth termed a "normal face") and each of the remaining $(b - k)$ faces have one of the l extraordinary vertices (henceforth termed a "special face"). Let $\mathbf{p}$ be the $3a = N$ dimensional vector containing the control vertex positions in 3D. Using the formulations in Section II-B and Section II-C, the smooth limit surface can be expressed as

$$\mathbf{s} = \sum_{i=1}^{k} \mathbf{n}_i + \sum_{j=1}^{l} \mathbf{s}_j \tag{6}$$

where $\mathbf{n}_i$ is a single bicubic patch assigned to each of the normal faces and $\mathbf{s}_j$ is a collection of infinite number of bicubic patches corresponding to each of the extraordinary points. As $\mathbf{s}$ is a surface defined over the faces of the initial control mesh, each $\mathbf{n}_i$ can be viewed as a bicubic patch defined over the corresponding regular face in the initial control mesh. Similarly, each $\mathbf{s}_j$ can be defined over the corresponding irregular faces in the initial mesh (refer to Fig.1–Fig.3 for the detailed description on parametric domains of subdivision surfaces). For the simplicity of the following mathematical derivations on our dynamic model, from now on we will not explicitly provide the parametric domain in our formulation. Employing the same approach taken before to derive eqn.5, it can be shown that

$$\sum_{i=1}^{k} \mathbf{n}_i = \sum_{i=1}^{k} ({}^n\mathbf{J}_i)({}^n\mathbf{p}_i) = (\sum_{i=1}^{k} ({}^n\mathbf{J}_i)({}^n\mathbf{A}_i))\mathbf{p} = ({}^n\mathbf{J})\mathbf{p} \tag{7}$$

where ${}^n\mathbf{J}_i, {}^n\mathbf{p}_i$ and ${}^n\mathbf{A}_i$ are the equivalent of $\mathbf{J}_i, \mathbf{p}_i$ in eqn.4 and $\mathbf{A}_i$ in eqn.5 respectively. The pre-superscript n is used to indicate that these mathematical quantities describe bicubic patch in the limit surface corresponding to normal faces.

We will use the following **notational convention** for describing various mathematical quantities used in the derivation of the expression for a collection of infinite number of bicubic patches around an extraordinary vertex. The pre-superscript s is used to represent a collection of bicubic patches around an extraordinary vertex, the subscript j is used to indicate the j-th extraordinary point, the post-superscript represents the exponent of a mathematical quantity and the level indicator (to represent various levels of subdivision in the local control mesh around an extraordinary vertex) is depicted via a subscript on the curly braces.

The expression for $\mathbf{s}_j$ is derived using the recursive nature of local subdivision around an extraordinary vertex as shown in Section II-C. First, $\mathbf{s}_j$ can be expressed as

$$\mathbf{s}_j = \{{}^s\mathbf{J}_j\}_1 \{{}^s\mathbf{p}_j\}_1 + \{\mathbf{s}_j\}_1 \tag{8}$$

where the first term of eqn.8 is the generalized coordinate representation of the bicubic B-spline patches corresponding to the normal faces of the new local subdivision mesh obtained after one subdivision step on the local control mesh (similar to those patches marked n in Fig.3). $\{\mathbf{s}_j\}_1$ represents the rest of the infinite bicubic B-spline patches surrounding the extraordinary point (similar to the central patch enclosed by patches marked n in Fig.3). The vertices in the newly obtained local subdivision mesh $\{{}^s\mathbf{p}_j\}_1$ can be expressed as a linear combination of a subset of the vertices of the initial mesh $\mathcal{M}$ (which will contribute to the local subdivision) following the subdivision rules. We can name this subset of initial control vertices $\{{}^s\mathbf{p}_j\}_0$. Furthermore, there exists a matrix $\{{}^s\mathbf{B}_j\}_1$ of size $(3c, 3d)$, such that $\{{}^s\mathbf{B}_j\}_1 \{{}^s\mathbf{p}_j\}_0 = \{{}^s\mathbf{p}_j\}_1$ where $\{{}^s\mathbf{p}_j\}_1$ and $\{{}^s\mathbf{p}_j\}_0$ are vectors of dimension $3c$ and $3d$ respectively. Applying the idea of recursive local subdivision again on $\{\mathbf{s}_j\}_1$, $\mathbf{s}_j$ can be further expanded as

$$\mathbf{s}_j = \{{}^s\mathbf{J}_j\}_1 \{{}^s\mathbf{B}_j\}_1 \{{}^s\mathbf{p}_j\}_0 + \{{}^s\mathbf{J}_j\}_2 \{{}^s\mathbf{B}_j\}_2 \{{}^s\tilde{\mathbf{p}}_j\}_1 + \{\mathbf{s}_j\}_2 \tag{9}$$

In the above derivation, $\{{}^s\tilde{\mathbf{p}}_j\}_1$ is a vector of dimension $3d$, comprising of a subset of the vertices defining the $3c$ dimensional vector $\{{}^s\mathbf{p}_j\}_1$. Note that, $\{{}^s\tilde{\mathbf{p}}_j\}_1$ has the same structure as $\{{}^s\mathbf{p}_j\}_0$, therefore, there exists a $(3d, 3d)$ matrix $\{{}^s\mathbf{C}_j\}_1$ such that $\{{}^s\mathbf{C}_j\}_1\{{}^s\mathbf{p}_j\}_0 = \{{}^s\tilde{\mathbf{p}}_j\}_1$. Each subdivision of a local mesh with d vertices creates a new local mesh with c vertices which contributes a fixed number of bicubic B-spline patches. So, if we proceed one step further, we obtain

$$\mathbf{s}_j = \{{}^s\mathbf{J}_j\}_1\{{}^s\mathbf{B}_j\}_1\{{}^s\mathbf{p}_j\}_0 + \{{}^s\mathbf{J}_j\}_2\{{}^s\mathbf{B}_j\}_2\{{}^s\mathbf{C}_j\}_1\{{}^s\mathbf{p}_j\}_0 + \{{}^s\mathbf{J}_j\}_3\{{}^s\mathbf{B}_j\}_3\{{}^s\mathbf{C}_j\}_1^2\{{}^s\mathbf{p}_j\}_0 + \{\mathbf{s}_j\}_3 \quad (10)$$

Because of the intrinsic property of the local recursive subdivision around the extraordinary point, we have $\{{}^s\mathbf{J}_j\}_1 = \{{}^s\mathbf{J}_j\}_2 = \ldots = \{{}^s\mathbf{J}_j\}_n = \ldots = \{{}^s\mathbf{J}_j\}_\infty$. In addition, the subdivision rules remain the same throughout the refinement process, we also have $\{{}^s\mathbf{B}_j\}_1 = \{{}^s\mathbf{B}_j\}_2 = \ldots = \{{}^s\mathbf{B}_j\}_n = \ldots = \{{}^s\mathbf{B}_j\}_\infty$. So, we can further simplify the above equations leading to

$$\begin{aligned}\mathbf{s}_j &= \{{}^s\mathbf{J}_j\}_1\{{}^s\mathbf{B}_j\}_1\{{}^s\mathbf{p}_j\}_0 + \{{}^s\mathbf{J}_j\}_1\{{}^s\mathbf{B}_j\}_1\{{}^s\mathbf{C}_j\}_1\{{}^s\mathbf{p}_j\}_0 + \{{}^s\mathbf{J}_j\}_1\{{}^s\mathbf{B}_j\}_1\{{}^s\mathbf{C}_j\}_1^2\{{}^s\mathbf{p}_j\}_0 + \ldots \\ &= \{{}^s\mathbf{J}_j\}_1\{{}^s\mathbf{B}_j\}_1(\sum_{i=0}^{\infty}\{{}^s\mathbf{C}_j\}_1^i)\{{}^s\mathbf{p}_j\}_0 \qquad (11)\end{aligned}$$

We can rewrite $\mathbf{s}_j$ as

$$\mathbf{s}_j = ({}^s\mathbf{J}_j)({}^s\mathbf{p}_j) \quad (12)$$

where ${}^s\mathbf{J}_j = \{{}^s\mathbf{J}_j\}_1\{{}^s\mathbf{B}_j\}_1(\sum_{i=0}^{\infty}\{{}^s\mathbf{C}_j\}_1^i)$ and ${}^s\mathbf{p}_j = \{{}^s\mathbf{p}_j\}_0$. The idea of local recursive subdivision around an extraordinary point is illustrated in Fig.4. Note that, each vertex position in the subdivided mesh is obtained by an affine combination of some vertices in the previous level and hence any row of $\{{}^s\mathbf{C}_j\}_1$ sums to 1. The largest eigenvalue of such a matrix is 1 and it can be shown that the corresponding infinite series is convergent following a similar approach as in [7]. The rest of the derivation leading to an expression for $\mathbf{s}$ is relatively straight forward. Using the same approach used to derive the eqn.7, it can be shown that

$$\sum_{j=1}^{l}\mathbf{s}_j = \sum_{j=1}^{l}({}^s\mathbf{J}_j)({}^s\mathbf{p}_j) = (\sum_{j=1}^{l}({}^s\mathbf{J}_j)({}^s\mathbf{A}_j))\mathbf{p} = ({}^s\mathbf{J})\mathbf{p} \quad (13)$$

From eqn.6, 7 and 13,

$$\mathbf{s} = ({}^n\mathbf{J})\mathbf{p} + ({}^s\mathbf{J})\mathbf{p} \quad (14)$$

Let $\mathbf{J} = ({}^n\mathbf{J}) + ({}^s\mathbf{J})$, and hence

$$\mathbf{s} = \mathbf{J}\mathbf{p} \quad (15)$$

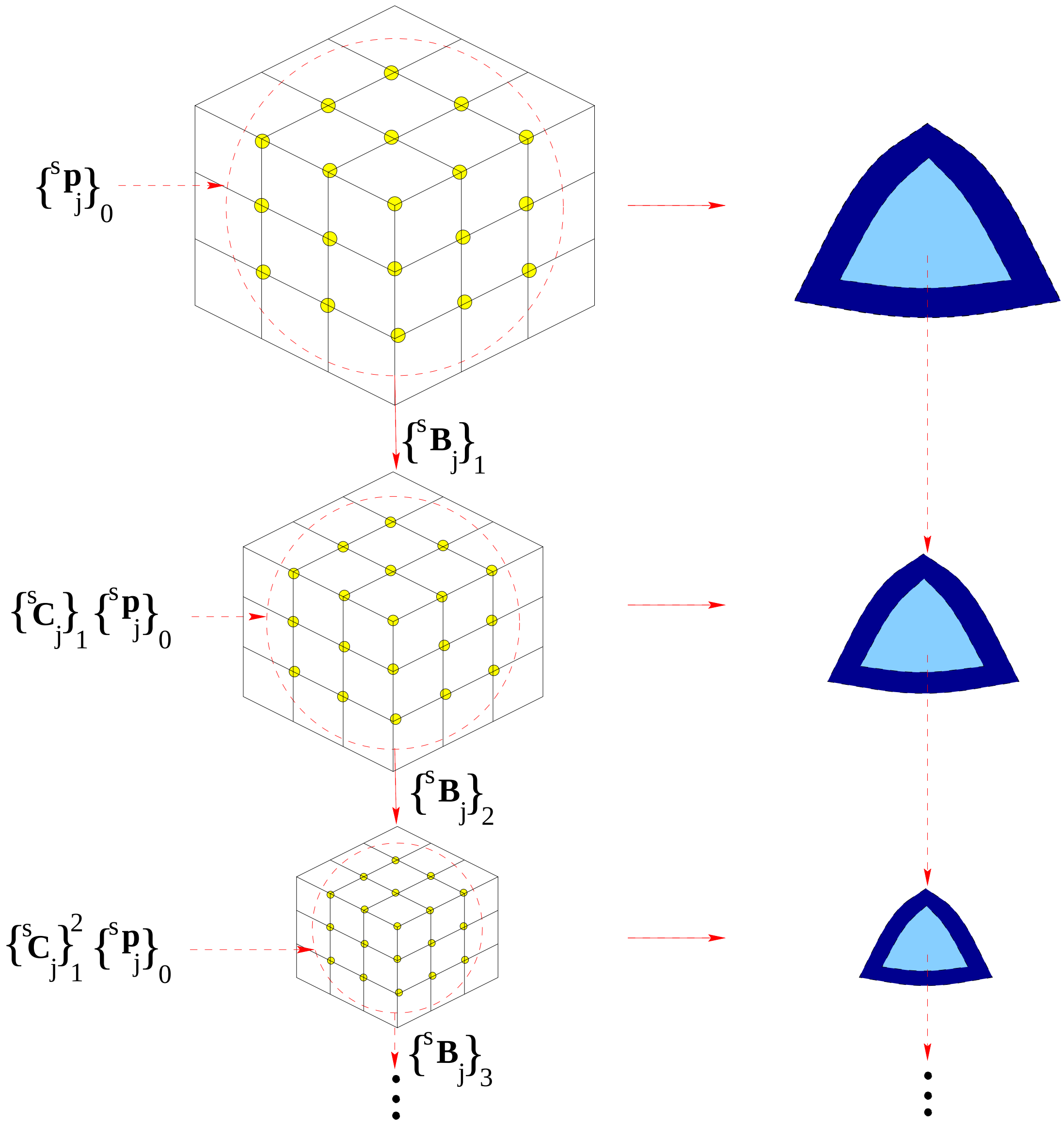


Fig. 4. Local subdivision around the extraordinary point and the corresponding patches in the limit surface from different levels of subdivision.

E. Dynamics

We now treat the control point positions (alternatively, the vertex positions in the initial mesh) defining the limit surface $\mathbf{s}$ as a function of time in order to develop our new dynamic model. The velocity of the surface model can be expressed as

$$\dot{\mathbf{s}}(u, v, \mathbf{p}) = \mathbf{J}\dot{\mathbf{p}} \tag{16}$$

where an overstruck dot denotes a time derivative. The physics of the dynamic subdivision surface model is based on the work-energy version of Lagrangian dynamics [46] and is formulated in an analogous way to the dynamic framework presented in [35].

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. The Lagrangian equation of motion can then be expressed as

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

Let $\mu(u,v)$ be the mass density function of the surface. Then

$$\mathbf{M} = \int\int \mu \mathbf{J}^T \mathbf{J} du dv \tag{18}$$

is an $N \times N$ mass matrix. Similarly the expression for damping matrix is

$$\mathbf{D} = \int\int \gamma \mathbf{J}^T \mathbf{J} du dv \tag{19}$$

where $\gamma(u,v)$ is the damping density.

A *thin-plate-under-tension* energy model [47] is used to compute the elastic potential energy of the dynamic subdivision surface. The corresponding expression for the stiffness matrix $\mathbf{K}$ is

$$\mathbf{K} = \int\int (\alpha_{11}\mathbf{J}_u^T\mathbf{J}_u + \alpha_{22}\mathbf{J}_v^T\mathbf{J}_v + \beta_{11}\mathbf{J}_{uu}^T\mathbf{J}_{uu} + \beta_{12}\mathbf{J}_{uv}^T\mathbf{J}_{uv} + \beta_{22}\mathbf{J}_{vv}^T\mathbf{J}_{vv}) du dv \tag{20}$$

where the subscripts on $\mathbf{J}$ denote the parametric partial derivatives. The $\alpha_{ii}(u,v)$ and $\beta_{ij}(u,v)$s are elasticity functions controlling local tension and rigidity in the two parametric coordinate directions. Note that for certain parameterizations, the thin-plate energy expression may diverge at extraordinary points on the limit surface for Catmull-Clark subdivision scheme as shown in [7]. Several methods have been suggested in [7] to overcome the problem of the divergent series. However, we circumvent the problem by setting the corresponding rigidity coefficients to be zero at these points. Therefore, the thin-plate energy at extraordinary points is zero. The effect is negligible to the overall thin-plate energy as *very few* extraordinary points are present in the smooth limit surface.

The generalized force vector $\mathbf{f}_p$ can be obtained through the principle of virtual work [46] done by the applied force distribution $\mathbf{f}(u,v,t)$ and can be expressed as

$$\mathbf{f}_p = \int\int \mathbf{J}^T \mathbf{f}(u,v,t) du dv \tag{21}$$

E.1 *Multilevel Dynamics*

Our dynamic Catmull-Clark surface model can be subdivided globally to increase the number of vertices (control points) of the model. For example, after one step of global subdivision, the initial degrees of freedom $\mathbf{p}$ (refer to eqn.15 and eqn.16) 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 (M, N) whose entries are uniquely determined by Catmull-Clark subdivision rules (see Section II-A for the details about the rules). Thus, $\mathbf{p}$, expressed as a function of $\mathbf{q}$, can be written as

$$\mathbf{p} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{q} = \mathbf{B}\mathbf{q} \tag{22}$$

where $\mathbf{B} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$. Therefore, we can rewrite eqn.15 and eqn.16 as

$$\mathbf{s} = (\mathbf{J}\mathbf{B})\mathbf{q} \tag{23}$$

and

$$\dot{\mathbf{s}}(u, v, \mathbf{q}) = (\mathbf{J}\mathbf{B})\dot{\mathbf{q}} \tag{24}$$

respectively. Now we need to 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

$$\mathbf{M}_q\ddot{\mathbf{q}} + \mathbf{D}_q\dot{\mathbf{q}} + \mathbf{K}_q\mathbf{q} = \mathbf{f}_q \tag{25}$$

where $\mathbf{M}_q = \int\int \mu\mathbf{B}^T\mathbf{J}^T\mathbf{J}\mathbf{B}dudv$ and the derivation of $\mathbf{D}_q$, $\mathbf{K}_q$ and $\mathbf{f}_q$ follow suit.

It may be noted that further subdivision, if necessary, can be carried out in a similar fashion. Therefore, multilevel dynamics is achieved through recursive subdivision on the initial set of control vertices. Users can interactively choose the level of detail representation of the dynamic model as appropriate for their modeling and design requirements. Alternatively, the system can automatically determine the level of subdivision most suitable for an application depending on some application-specific criteria.

III. FINITE ELEMENT IMPLEMENTATION

The evolution of the generalized coordinates for our new dynamic surface model can be determined by the second-order differential equation as given by eqn.17. An analytical solution of the governing differential equation can not be derived in general. However, an efficient numerical implementation can be obtained using the finite element method [48]. The limit surface of the dynamic Catmull-Clark subdivision model is a collection of bicubic patch elements. We use two types of finite elements for this purpose - normal elements (bicubic patches assigned to the normal faces of the initial mesh) and special elements (collection of infinite number of bicubic patches assigned to each extraordinary vertex of the initial mesh). The shape functions for the normal element are the basis functions of the bicubic surface patch, whereas the shape functions for the special element are the collection of basis functions corresponding to the bicubic patches in the special element. In the current implementation, the **M**, **D** and **K** matrices for each individual normal and special elements are calculated and they can be assembled into the global **M**, **D** and **K** matrices that appear in the corresponding discrete equation of motion. In practice, we never assemble the global matrices explicitly in the interest of time performance. The detailed implementation is explained in the following sections.

A. Data Structures

The limit surface of the dynamic Catmull-Clark subdivision model is a collection of bicubic patches, and hence requires us to keep track of the control polygons defining such patches. We can get the control polygons for the normal elements in the limit surface from the initial control mesh itself. However, we need to locally subdivide the initial control mesh around the extraordinary vertices to obtain the control polygons of the bicubic patches in the special elements on the limit surface.

A subdivision surface defined by a control mesh at any level is designed as a class which has a pointer to its parent mesh, a set of pointers to its offspring meshes (arising out of local subdivision around the extraordinary vertices at that level), a list of faces, edges, vertices and normal elements . Face, edge, vertex and normal elements are, in turn, classes which store all the connectivity and other information needed to either enumerate all the patches or locally subdivide around an extraordinary vertex in that

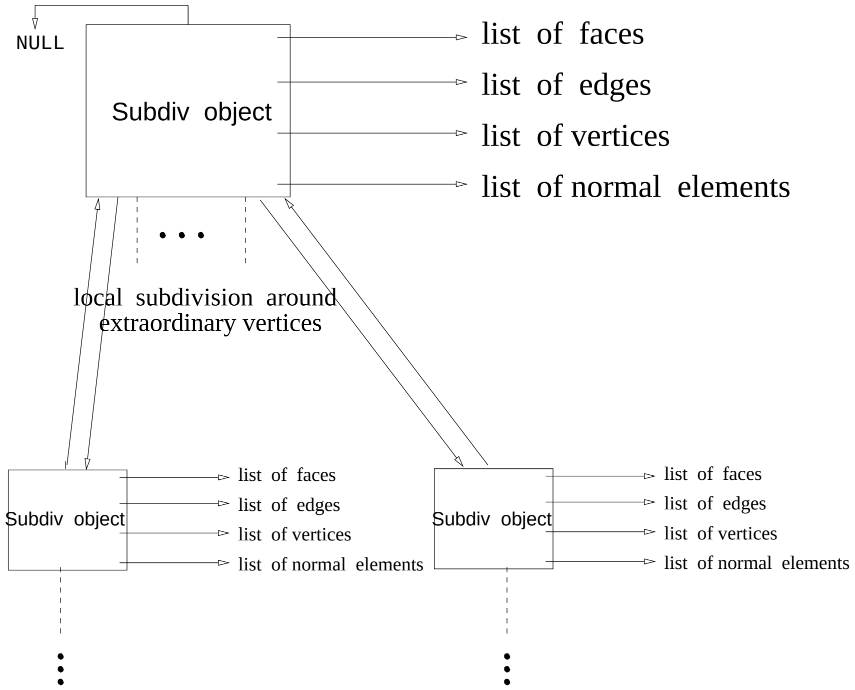


Fig. 5. The data structure used for dynamic subdivision surface implementation.

level. The implementation takes the initial mesh as the base subdivision surface object (with its parent pointer set to NULL) and locally subdivides the initial mesh upto a user-defined maximum level around each extraordinary vertex to create offspring objects at different levels (Fig.5). At this point, let's take a closer look at the normal and special element data structures and computation of the corresponding local $\mathbf{M}, \mathbf{D}$ and $\mathbf{K}$ matrices.

A.1 *Normal Elements*

Each normal element is a bicubic surface patch and hence is defined by 16 vertices (from the 8-connected neighborhood of the corresponding normal face) in the initial control mesh. Each normal element keeps a set of pointers to those vertices of the initial mesh which act as control points for the given element. For a normal element, the mass, damping and stiffness matrices are of size $(16, 16)$ and can be computed exactly by carrying out the necessary integrations analytically. The matrix $\mathbf{J}$ in eqn.18, 19 and 20 needs to be replaced by $\mathbf{J}_p$ (of eqn.2) for computation of the local $\mathbf{M}, \mathbf{D}$ and $\mathbf{K}$ matrices respectively of the corresponding normal element.

A.2 *Special Elements*

Each special element consists of an infinite number of bicubic patches in the limit. We have already described a recursive enumeration of the bicubic patches of a special element in Section II-C. Let us now consider an arbitrary bicubic patch of the special element in some level j. The mass matrix $\mathbf{M}_s$ of this patch can be written as

$$\mathbf{M}_s = \mathbf{\Omega}_s^T \mathbf{M}_p \mathbf{\Omega}_s \tag{26}$$

where $\mathbf{M}_p$ is the normal element mass matrix (scaled by a factor of $\frac{1}{4^j}$ to take into account of the area shrinkage in bicubic patches at higher level of subdivision) and $\mathbf{\Omega}_s$ is the transformation matrix of the control points of that arbitrary patch from the corresponding control points in the initial mesh. The damping and stiffness matrices for the given bicubic patch can be derived in a similar fashion. These mass, damping and stiffness matrices from various levels of (local) subdivision can then be assembled to form the mass, damping and stiffness matrices of the special element. It may be noted that the stiffness energy due to plate terms diverges at the extraordinary points on the limit surface. We solve the problem using a slightly different approach than the one used in [7]. When the area of the bicubic patch obtained via local subdivision of the initial mesh around an extraordinary vertex becomes smaller than the display resolution, the contribution from such a bicubic patch is ignored in computing the physical matrices of the corresponding special element. The number of extraordinary points in the limit surface is *very few*, and the above mentioned approximation is found to work well in practice.

B. Force Application

The force $\mathbf{f}(u, v, t)$ in eqn.21 represents the net effect of all 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 be easily implemented.

To apply spring forces, a spring of stiffness k can be connected from a point $\mathbf{d}_0$ to a point (u_0, v_0) on the limit surface, the net applied spring force being

$$\mathbf{f}(u, v, t) = \int\int k(\mathbf{d}_0 - \mathbf{s}(u, v, t))\delta(u - u_0, v - v_0)du dv \tag{27}$$

where δ is the unit impulse function implying $\mathbf{f}(u_0, v_0, t) = k(\mathbf{d}_0 - \mathbf{s}(u_0, v_0, t))$ and vanishes elsewhere in 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 a mouse button 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 Gaussian smoothed volume data set using the 3D Monga-Deriche(MD) operator [49] 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

$$\mathbf{f}(x, y, z) = k \frac{\bigtriangledown P(x, y, z)}{\| \bigtriangledown P(x, y, z) \|} \tag{28}$$

where k controls the strength of the force. The applied force on each element is computed using Gaussian quadrature for evaluating eqn.21 in Cartesian coordinates. It may be noted that we can apply spring forces in addition to the image-based forces by interactively placing a sparse set of points along the boundary of the region of interest in the slices of the 3D image data and using the distance from these points to the model as described above.

C. 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

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

and

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

Using eqns.17, 29 and 30, the discrete equation of motion is obtained as

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

This linear system of equations is solved iteratively between each time step using the conjugate gradient algorithm [50]. For a first order system with no mass, the above equation reduces to

$$(\mathbf{D} + 2\Delta t \mathbf{K})\mathbf{p}(t + \Delta t) = 2\Delta t \mathbf{f}_p(t + \Delta t) + \mathbf{D}\mathbf{p}(t - \Delta t) \tag{32}$$

which gives a faster convergence.

D. Model Subdivision

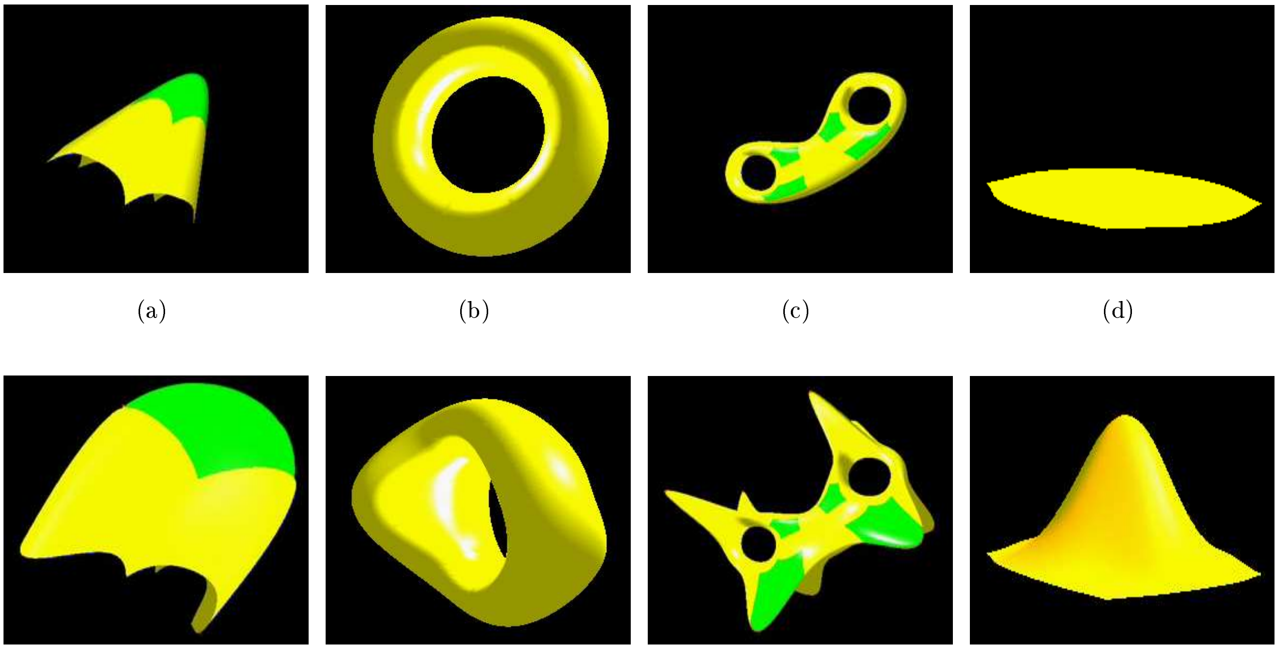
The initialized model grows dynamically according to the equation of motion (eqn.17) and when an equilibrium is achieved, the number of control vertices can be increased by replacing the original initial mesh by a new initial mesh obtained through one step of Catmull-Clark subdivision. This increases the number of degrees of freedom to represent the same smooth limit surface and a new equilibrium position for the model with a better fit to the given data set can be achieved. 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 within a prescribed tolerance between successive iterations thereby 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 energy between successive iterations becomes less than a pre-specified tolerance.

IV. RESULTS

The proposed dynamic Catmull-Clark subdivision surface model can be used to represent a wide variety of shapes with arbitrary genus. The smooth limit surface can be manipulated by applying synthesized forces in a direct and intuitive way in the geometric modeling applications. The underlying shape of a large range or volume data set can also be recovered efficiently using the proposed dynamic Catmull-Clark subdivision surface model. The application results of the proposed model are illustrated in the rest of this section. In all the experiments, the normal elements are shaded yellow, whereas the special elements are colored green.

A. Applications in modeling

In a typical geometric modeling application using dynamic Catmull-Clark subdivision surface model, the user can specify any mesh as the initial (control) mesh, and the corresponding limit surface can be manipulated interactively by applying synthesized forces. In Fig.6, we show several initial surfaces

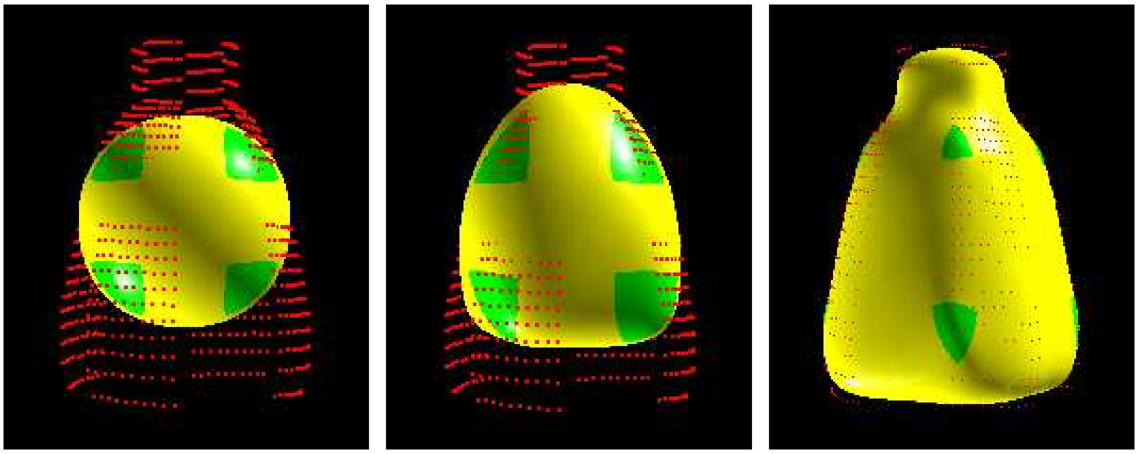


(a) (b) (c) (d)

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

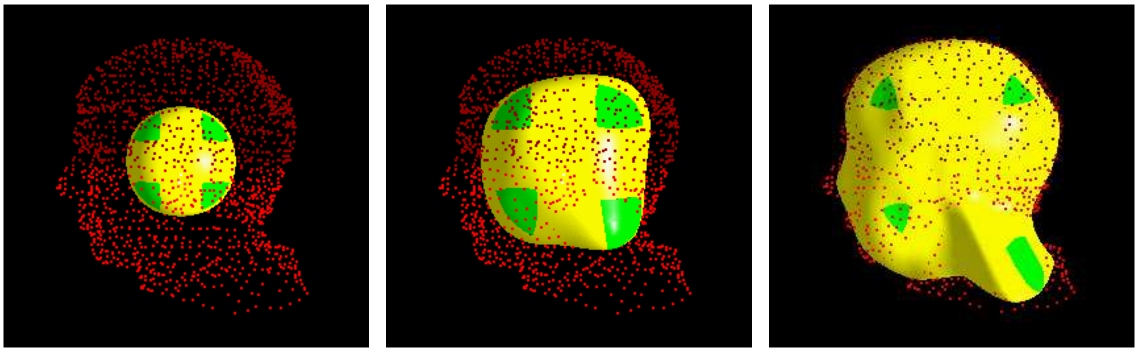
Fig. 6. (a), (b), (c) and (d) : Initial shapes; (e), (f), (g) and (h) : the corresponding modified shapes after interactive force application.

obtained from different control meshes and the corresponding modified surfaces after interactive spring force application. To change the shape of an initial surface, we 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. In Fig.6(a), an open surface defined by an initial mesh of 61 vertices and 45 faces is shown. The mesh has one extraordinary vertex of degree 5. This limit surface is modified by applying spring forces interactively, and the modified surface is depicted in Fig.6(e). A torus, defined by an initial mesh of 32 vertices and 32 faces, and its modified shape is shown in Fig.6(b) and (f) respectively. The initial mesh of the smooth limit surface shown in Fig.6(c) has 544 faces and 542 vertices, 8 of them are extraordinary vertices of degree 5. The limit surface is modified interactively by applying spring forces from various points in 3D and the modified shape is depicted in Fig.6(g). Note that the extent of deformation has been interactively controlled by varying the stiffness of the attached springs. The upper portion of the limit surface has been deformed by applying spring forces of higher magnitude, whereas



(a) (b) (c)

Fig. 7. (a) Range data of a bulb along with the initialized model, (b) an intermediate stage of evolution, and (c) the fitted dynamic Catmull-Clark subdivision surface model.



(a) (b) (c)

Fig. 8. (a) Range data of a head along with the initialized model, (b) an intermediate stage of evolution, and (c) the fitted dynamic Catmull-Clark subdivision surface model.

the lower portion has been modified by applying spring forces of lower magnitude. The spread of the deformation effect is clearly larger in the latter case for obvious reasons. Finally, a flat sheet defined by an initial mesh of 64 faces and 81 vertices, shown in Fig.6(d), is deformed interactively to obtain the hat-like shape shown in Fig.6(h).

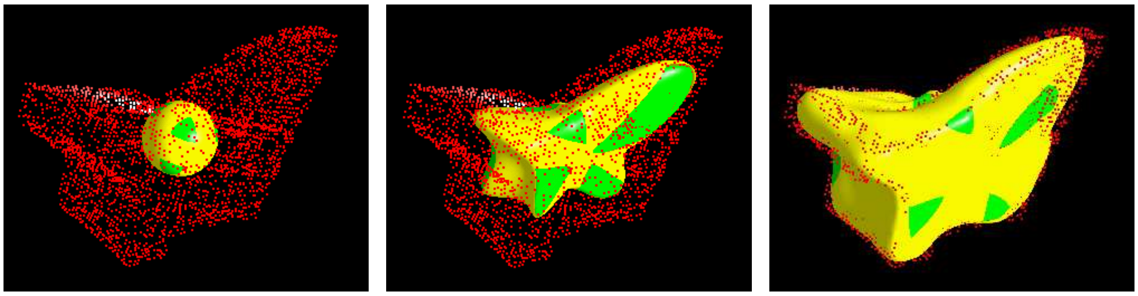

(a) (b) (c)

Fig. 9. (a) Range data of an anvil along with the initialized model, (b) an intermediate stage of evolution, and (c) the fitted dynamic Catmull-Clark subdivision surface model.

B. Applications in shape recovery from range data

The dynamic Catmull-Clark subdivision surface model can recover the underlying shape of a given range data set effectively in a hierarchical fashion. The initialized model deforms under the influence of the spring forces from the range data. When an approximate shape is recovered, a new control mesh can be obtained by one step of Catmull-Clark subdivision on the initial mesh thereby increasing the degrees of freedom to represent the same limit surface, and a better fit to the given range data set can be achieved. It may be noted that the model initialization is interactive, and the initialized model can have any control mesh of the desired genus. However, an initial mesh with few degrees of freedom usually performs better in terms of the compact representation of the underlying shape. In all the experiments in this section, the initialized model had 96 faces and 98 vertices, 8 of them being extraordinary vertices of degree 3. The final fitted model, obtained through one step of subdivision, has a control polygon of 384 faces with 386 vertices. The error in fit, which is defined as the maximum distance between a data point and the nearest point on the limit surface as a percentage of the diameter of the smallest sphere enclosing the object, is approximately 3% in all the experiments with range data. The time of dynamic evolution for fitting the range data sets used in the experiments is approximately 3 minutes in a SGI O_2 workstation.

In Fig.7, we demonstrate the model fitting algorithm applied to laser range data acquired from multiple views of a light bulb. 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.

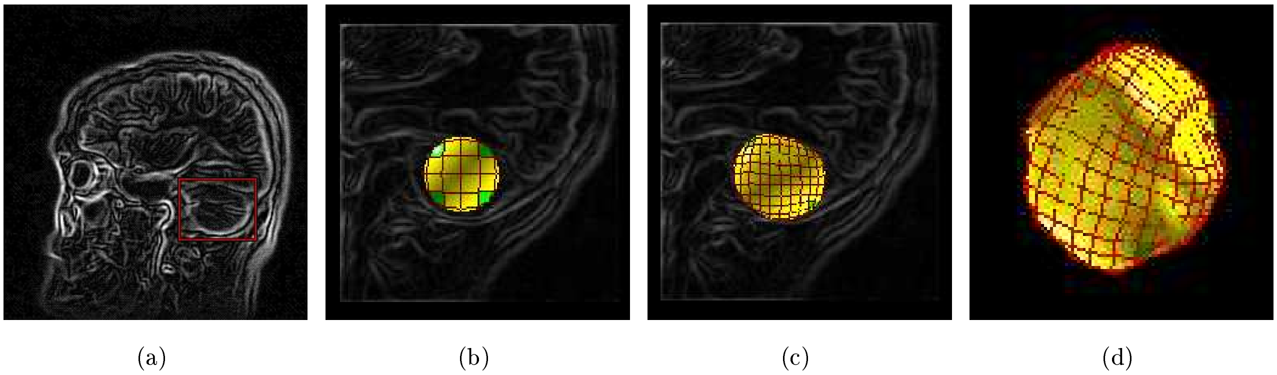


Fig. 10. (a) A slice from a brain MRI, (b) initialized model inside the region of interest superimposed on the slice, (c) the fitted model superimposed on the slice, and (d) a 3D view of the dynamic Catmull-Clark subdivision surface model fitted to the cerebellum.

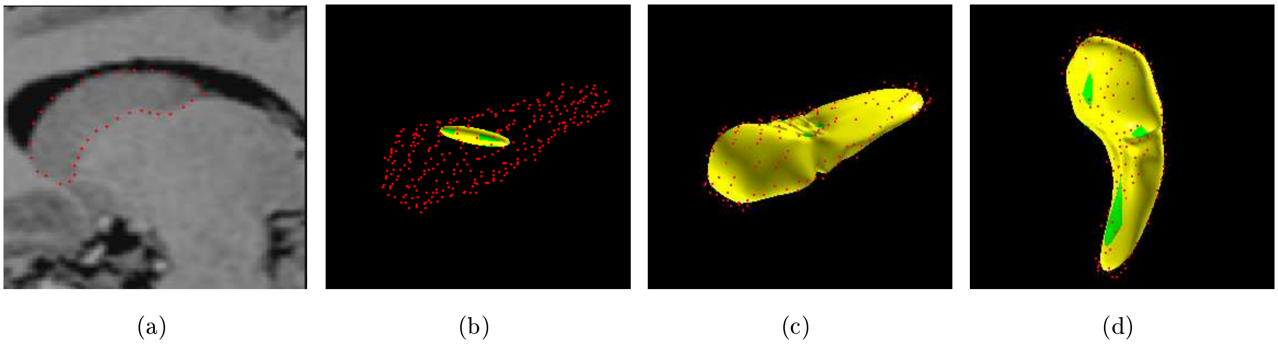


Fig. 11. (a) Data points identifying the boundary of the region of interest (a caudate nucleus) on a MRI slice of human brain, (b) data points (from all the slices) in 3D along with the initialized model, (c) fitted dynamic Catmull-Clark subdivision surface model, and (d) another view of the fitted model.

In the next experiment, the shape of a head is recovered from a range data set as shown in Fig.8. The range data set has 1779 points in 3D. In the last experiment with range data, the dynamic Catmull-Clark subdivision surface model is fitted to an anvil data set (Fig.9). The data set has 2031 data points. It may be noted that the final shape with 3% error tolerance uses very few control points for representation in comparison with the number of data points present in the original range data set.

C. Applications in shape recovery from volume data

The application of the dynamic Catmull-Clark subdivision surface model to anatomical shape recovery from 3D volumetric MRI data is shown in the next two experiments. As in the previous section, the initialized model had 96 faces and 98 vertices, 8 of them being extraordinary vertices of degree 3. The final fitted model, obtained through one step of subdivision, has a control polygon of 384 faces with 386 vertices.

First, the model is fitted to a cerebellum (a cortical structure in brain) given an input of 30 sagittal slices from a MR brain scan. Fig.10(a) depicts a slice from this MRI scan and the model initialization is shown in Fig.10(b). Continuous image based forces are applied to the model and the model deforms under the influence of these forces until maximum conformation to the boundaries of the desired cerebellum shape. The final fitted model is shown in Fig.10(c). A 3D view of the fitted model is depicted in Fig.10(d).

In the next experiment, we present the shape extraction of a caudate nucleus (another cortical structure in human brain) from 64 MRI slices, each of size $(256, 256)$. Fig.11(a) depicts a slice from this MRI scan along with the points placed by an expert neuroscientist on the boundary of the shape of interest. Fig.11(b) depicts the interactively placed sparse set of data points (placed in some of the slices depicting the boundary of the shape of interest) in 3D along with the initialized model. Note that points had to be interactively placed on the boundary of the caudate nucleus in MR slices lacking image gradients which delineate the caudate from the surrounding tissue in 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 maximum conformation to the boundaries of the desired caudate shape. Two arbitrary views of the final fitted model in 3D are shown in Fig.11(c) and (d).

V. CONCLUSIONS

In this paper, a dynamic framework for the Catmull-Clark subdivision surfaces is presented which has numerous applications in geometric modeling, computer graphics and scientific visualization. Apart from providing a direct and intuitive way of manipulating shapes, it facilitates the modeling and shape analysis of objects contained in range and volume data sets using very few degrees of freedom. We have presented an analytic formulation of the subdivision model, incorporated the advantages of free-form deformable models in subdivision scheme, introduced hierarchical dynamic control, and shown the advantages of our model via experiments. However, the current scheme can not recover sharp edges in the data. Also, the initialization is interactive; ideally, initialization should be done automatically on the basis of the input data set. Efficient numerical techniques like preconditioning will speed up the fitting process significantly. Our future efforts will be focussed on addressing these issues.

VI. Acknowledgements

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 also wish to acknowledge Dr. G. Malandain for the edge detection software, Dr. H. Hoppe and Dr. K. Pulli for the range data, and Dr. C.M. Leonard for the brain MRI data.

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] 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.

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

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

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

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

[21] 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.

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

[23] 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.

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

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

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

[33] 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.

[34] 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.

[35] 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.

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

[37] 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.

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

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

[40] 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.

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

[42] 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.

[43] T. McInerney and D. Terzopoulos, "A finite element model for 3d shape reconstruction and nonrigid motion tracking," in *Proceedings of the International Conference on Computer Vision*, Berlin, Germany, May 1993, pp. 518 – 523.

[44] 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.

[45] 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.

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

[47] 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.

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

[49] 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.

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