Dynamic Smooth Subdivision Surfaces for Data Visualization

Chhandomay Mandal Hong Qin Baba C. Vemuri

Department of Computer & Information Science and Engineering
University of Florida
Gainesville, FL 32611, USA.
E-mail: cmandal—qin—vemuri@cise.ufl.edu

Abstract

Recursive subdivision schemes have been extensively used in computer graphics 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, which is a very popular method to model complicated objects of arbitrary genus because of many of its nice properties. Our new dynamic surface model inherits the attractive properties of the Catmull-Clark subdivision scheme as well as that of the physics-based modeling paradigm. This new model provides a direct and intuitive means of manipulating geometric shapes, a fast, robust, and hierarchical approach for recovering complex geometric shapes from range and volume data 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 in real time. The governing dynamic differential equation is derived using Lagrangian mechanics and a finite element discretization. Our experiments demonstrate that this new dynamic model has a promising future in computer graphics, geometric shape design and scientific visualization.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling; I.3.8 [Computer Graphics]: Miscellaneous.

Keywords: Visualization, Subdivision Surfaces, Dynamics, Finite Elements, Interactive Techniques.

1 INTRODUCTION

Generating smooth surfaces of arbitrary topology is a grand challenge in geometric modeling, visualization, and computer graphics. The recursive subdivision scheme which generates a smooth surface as the limit of a sequence of recursively refined polyhedral surfaces based on a user-defined initial control mesh, is well suited for the purpose. At each step of the subdivision, a *smoother* polyhedral surface with more vertices and faces will be constructed from the previous one via a refinement process (also called "chopping corners"). In general, subdivision schemes can be categorized into two distinct classes namely, (1) approximating subdivision methods, and (2) interpolating subdivision methods.

Among the approximating schemes, the techniques of Doo and Sabin [6] and Catmull and Clark [3] generalize the idea of obtaining biquadratic and bicubic B-spline patches from rectangular control meshes. In [3], 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 *extraordinary* points, where the in-degree of the vertex in the mesh is not equal to four. Doo and Sabin [6] further analyzed the smoothness behavior of the limit surface near extraordinary points using Fourier transforms and an eigenvalue analysis of the subdivision matrix. Ball and Storry [1] and Reif [14] further extended the prior work on continuity properties of subdivision surfaces by deriving various necessary and sufficient conditions on smoothness for different subdivision schemes. In [10], Loop presented a similar subdivision scheme based on the generalization of quartic triangular B-splines for triangular meshes. Halstead et al. [11] proposed an algorithm to construct a Catmull-Clark subdivision surface that interpolates the vertices of a mesh of arbitrary topology. The most well-known interpolation-based subdivision scheme is the "butterfly" algorithm proposed by Dyn et al. [5]. Butterfly subdivision method 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 for the initial polygonal meshes in order to obtain a smooth limit surface. Recently, Zorin et al. [13] further developed an improved interpolatory subdivision scheme that can retain the simplicity of the butterfly scheme and result in much smoother surfaces from initial polygonal meshes that are irregular.

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 infeasible 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 spline-based modeling schemes. 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 ("roughness") may be strongly desired while making geometric objects satisfy global smoothness requirements in scientific visualization and biomedical 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 are of particular interest in this context. The behavior of dynamic deformation is governed by physical laws. Deformable models will respond dynamically to applied forces in a very intuitive manner. 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 impose geometric constraints relevant to shape design.

Free-form deformable models were first introduced to computer graphics and visualization in Terzopoulos et al. [8]. Bloor and Wilson [2], Celniker and Welch [4] and Welch and Witkin [17] 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 [16] 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 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 is almost impossible to model surfaces of arbitrary genus using these models. In this paper, we develop a dynamic generalization of recursive subdivision schemes based on Catmull-Clark subdivision surfaces. Our new dynamic model combines the benefits of subdivision surfaces for modeling arbitrary topology as well as the dynamic splines for direct interactive manipulation of shapes 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 parametrization of the limit surface exists near the extraordinary points. We present the details of our formulation in a later section. We conduct various experiments that demonstrates the fact that our new dynamic surface model is extremely useful in computer graphics and visualization. The model can recover shapes from large range and volume data sets using very few degrees of freedom for its representation. The initialized model deforms under the influence of synthesized forces to fit the data set by minimizing its energy according to the laws of physics. When the approximate shape is recovered, the model can be further subdivided to achieve a better approximation to the input data set using higher degrees of freedom.

The rest of the paper is organized as follows: Section 2 presents the detailed formulation of the dynamic Catmull-Clark subdivision surfaces. The implementation details are provided in Section 3. Experimental results are presented in Section 4. Finally, we conclude in Section 5.

2 FORMULATION

In this section we present a systematic formulation of our new dynamic model based on Catmull-Clark subdivisions. 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 topologically 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 limit smooth surface which can be viewed as a function of its (initial) polyhedral control vertices. Finally, we introduce physical quantities into our dynamic model in order to derive its motion equation.

2.1 Catmull-Clark subdivision surfaces

Catmull-Clark subdivision scheme, like any other subdivision scheme, starts with a user-defined mesh of arbitrary topology, refines the initial mesh by adding new vertices, edges and faces with each step of subdivision which follows 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, introduce a new face point obtained from the average $\frac{F}{n} + \frac{2E}{n} + \frac{(n-3)V}{n}$, where F is the average of the new face points of all faces adjacent to the old vertex point, E is the average of the midpoints of all edges incident on the old vertex 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.

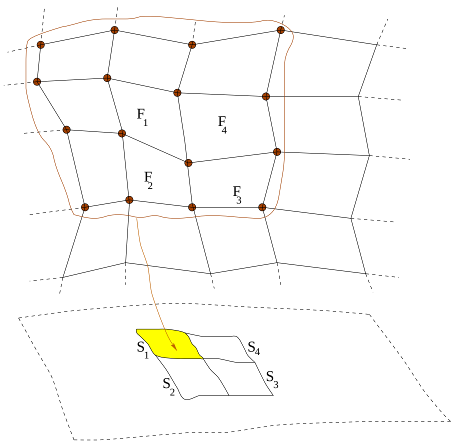

Figure 1: A rectangular mesh and its limit surface consisting of 4 bicubic surface patches.

The most important property of Catmull-Clark subdivision surfaces is that the smooth surface can be generated from control meshes 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 control point mesh). 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 surfaces

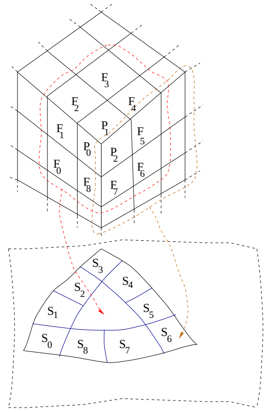

Figure 2: A mesh with an extraordinary point of valence 3 and its limit surface.

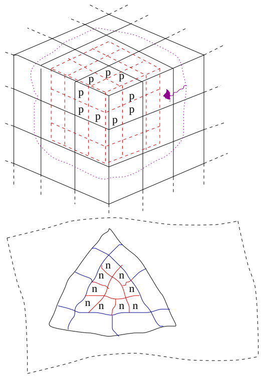

Figure 3: Local subdivision around the extraordinary point and the limit surface.

remains a constant which is determined by the refined meshes after one subdivision. 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 are not parametrizable and lack closed-form analytic formulations. 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 offer modelers 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. The 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 subsections, we present a scheme of assigning the bicubic patches to various faces of the initial mesh. It may be noted that one additional subdivision step may be needed in some cases to isolate the extraordinary points and treat the obtained mesh as the initial mesh (one typical example is when the initial mesh is a tetrahedron).

2.2 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 valence 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.

2.3 Assigning patches to irregular faces

In Fig.2, a mesh containing an extraordinary point of valence 3 and its limit surface are shown. The faces $F_0, F_1, \ldots, F_8$ are assigned to bicubic patches $S_0, S_1, ...S_8$ respectively (as they all have vertices of valence 4) following the aforementioned scheme. However, the smooth surface enclosed by the patches $S_0, S_1, ...S_8$ 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.

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 valence 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.

2.4 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 $\mathbf{s}_p(u,v), (u,v) \in [0,1]^2$, denote this bicubic B-spline patch which can be expressed analytically as

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

where $\mathbf{d}_{i,j}$ represents a 3-dimensional position vector at the (i,j)th control point location and $B_{i,4}(u)$,$B_{j,4}(v)$ 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

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

where $\mathbf{J}_p$ is the standard Jacobian matrix of a bicubic B-spline patch, and is of size is $(3, 48)$. Vector $\mathbf{q}$ 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 $\mathbf{q}$. 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 $\mathbf{J}_p(0, k+1) = \mathbf{J}_p(0, k+2) = \mathbf{J}_p(1, k) = \mathbf{J}_p(1, k+2) = \mathbf{J}_p(2, k) = \mathbf{J}_p(2, k+1) = 0$.

2.4.1 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

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

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 Jacobian 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

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

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 $\mathbf{A}_i, 1 \le i \le 4$, are of size $(48, 75)$, each row consisting of a single nonzero entry $(= 1)$ and the $(3, 75)$-sized matrix $\mathbf{J}_m = \sum_{i=1}^{4} \mathbf{J}_i\mathbf{A}_i$.

2.4.2 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 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 M be a, and let l of these be extraordinary vertices. Let us assume that the number of faces in the initial mesh are b, and that k of these have vertices with valence 4 (here onwards termed a "normal face") and each of the remaining $(b - k)$ faces have one of the l extraordinary vertices (here onwards termed a "special face"). Let $\mathbf{p}$ be the $3a = N$ dimensional vector containing the control vertex positions in 3D. Using the formulations in subsections 2.2 and 2.3, 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. 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 subscripts 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 subsection 2.3. 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 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 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

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

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

$$\begin{aligned}\mathbf{s}_j &= \{{}^s\mathbf{J}_j\}_1\{{}^s\mathbf{B}_j\}_1\{{}^s\mathbf{p}_j\}_0 + \{{}^s\mathbf{J}_j\}_2\{{}^s\mathbf{B}_j\}_2\{{}^s\mathbf{C}_j\}_1\{{}^s\mathbf{p}_j\}_0 \\ &+ \{{}^s\mathbf{J}_j\}_3\{{}^s\mathbf{B}_j\}_3\{{}^s\mathbf{C}_j\}_1^2\{{}^s\mathbf{p}_j\}_0 + \{\mathbf{s}_j\}_3\end{aligned} \tag{10}$$

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

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

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

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

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 ${}^s\mathbf{p}_j = \{{}^s\mathbf{p}_j\}_0$. 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 [11]. 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,

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

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

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

2.5 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

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

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 [7] and is formulated in an analogous way to that in [16].

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 applied force acting on p_i and these f_i's are assembled into the vector $\mathbf{f}_p$. The Lagrangian equation of motion can then be expressed as

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

Let μ be the mass density function of the surface. Then

$$\mathbf{M} = \int\int \mu\mathbf{J}^T\mathbf{J}dudv \qquad (18)$$

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

$$\mathbf{D} = \int\int \gamma\mathbf{J}^T\mathbf{J}dudv \qquad (19)$$

where γ is the damping density.

A *thin-plate-under-tension* energy model [15] is used to compute the elastic potential energy of the dynamic subdivision surface. The corresponding expression for the stiffness matrix $\mathbf{K}$ is

$$\begin{aligned}\mathbf{K} &= \int\int(\alpha_{11}\mathbf{J}_u^T\mathbf{J}_u + \alpha_{22}\mathbf{J}_v^T\mathbf{J}_v + \beta_{11}\mathbf{J}_{uu}^T\mathbf{J}_{uu} \\ &\quad + \beta_{12}\mathbf{J}_{uv}^T\mathbf{J}_{uv} + \beta_{22}\mathbf{J}_{vv}^T\mathbf{J}_{vv})dudv \qquad (20)\end{aligned}$$

where the subscripts on $\mathbf{J}$ denote the parametric partial derivatives. The α_{ii} and β_{ij}s are elasticity functions controlling local tension and rigidity in the two parametric coordinate directions.

The generalized force vector $\mathbf{f}_p$ can be obtained through the principle of virtual work [7] done by the applied force distribution $\mathbf{f}(u, v, t)$ and can be expressed as

$$\mathbf{f}_p = \int\int \mathbf{J}^T\mathbf{f}(u, v, t)dudv \qquad (21)$$

2.5.1 Multilevel Dynamics

Our dynamic Catmull-Clark surface model can be subdivided globally to increase the number of vertices (control points) of the model. Therefore, we need to derive the equation of motion for this subdivided model involving larger number of vertices. We need to recompute the mass, damping and stiffness matrices for this level. However, the equation of motion as given by Eqn. 17 remains valid, only the dimensionality of $\mathbf{M}$, $\mathbf{D}$, $\mathbf{K}$, $\mathbf{p}$ and $\mathbf{f}_p$ changes in the newly obtained subdivided level.

3 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 obtained in general. However, an efficient numerical implementation can be obtained using finite element analysis techniques [9]. For the dynamic subdivision surface model, two types of finite elements are considered - 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). In the current implementation, the $\mathbf{M}$, $\mathbf{D}$ and $\mathbf{K}$ matrices for each individual normal and special elements are calculated and they can be assembled into the global $\mathbf{M}$, $\mathbf{D}$ and $\mathbf{K}$ 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 subsections.

3.1 Data Structures

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 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. At this point, let's take a closer look at normal and special element data structures and computation of the corresponding local $\mathbf{M}, \mathbf{D}$ and $\mathbf{K}$ matrices.

3.1.1 Normal Elements

Each normal element is a bicubic surface patch and hence defined by 16 vertices (from the 8-connected neighborhood of the corresponding normal face). 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 need 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.

3.1.2 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 2.3. Let us now consider some arbitrary bicubic patch of the special element in some level j. The mass matrix $\mathbf{M}_s$ of this patch can be written as

$$\mathbf{M}_s = \mathbf{\Omega}_s^T \mathbf{M}_p \mathbf{\Omega}_s \qquad (22)$$

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 an exactly similar fashion. Now, these mass, damping and stiffness matrices can be assembled to form the mass, damping and stiffness matrices of the special element. As mentioned earlier, the infinite series summation is convergent. However, it has been found that the contribution from bicubic patches at higher level of subdivision to the mass, damping and stiffness matrices becomes negligible and in the implementation, the local subdivision is carried out until the contribution is small enough to ignore in the formation of the global mass, damping and stiffness matrices.

3.2 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 implemented easily as well.

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

$$\mathbf{f}(u, v, t) = \int\int k(\mathbf{d}_0 - \mathbf{s}(u, v, t))\delta(u - u_0, v - v_0)dudv \quad (23)$$

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 the disk.

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 [12] 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

$$\mathbf{f}(x, y, z) = k \frac{\nabla P(x, y, z)}{\| \nabla P(x, y, z) \|} \qquad (24)$$

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 with the image-based forces by placing points near the region of interest in the slices of the 3D image data.

3.3 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 backward differences

$$\ddot{\mathbf{p}}(t + \Delta t) = \frac{\mathbf{p}(t + \Delta t) - 2\mathbf{p}(t) + \mathbf{p}(t - \Delta t)}{\Delta t^2} \qquad (25)$$

and

$$\dot{\mathbf{p}}(t + \Delta t) = \frac{\mathbf{p}(t + \Delta t) - \mathbf{p}(t - \Delta t)}{2\Delta t} \qquad (26)$$

Using Eqn. 17, 25 and 26, the discrete equation of motion is obtained as

$$(2\mathbf{M} + \mathbf{D}\Delta t + 2\Delta t^2\mathbf{K})\mathbf{p}(t + \Delta t) =$$

$$2\Delta t^2\mathbf{f}_p(t + \Delta t) + (\mathbf{D}\Delta t - 2\mathbf{M})\mathbf{p}(t - \Delta t) + 4\mathbf{M}\mathbf{p}(t) \quad (27)$$

This linear system of equations is solved iteratively between each time step using the conjugate gradient method.

3.4 Model Subdivision

The initialized model grows dynamically according to the equation of motion (Eqn. 17) and when an equilibrium is achieved at a given level of subdivision, the model can be subdivided, if necessary, according to the Catmull-Clark subdivision rules to increase the number of vertices (control points) and a better fit to the data can be achieved. Currently the error of fit criteria is based on distance between the data points and the points on the limit surface where the corresponding springs are attached. However, other types of error criterion can also be defined and used in this context. For example, 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 model can be subdivided further until the model energy is sufficiently small and the change in energy between successive iterations becomes less than a pre-specified tolerance.

4 RESULTS

The proposed dynamic subdivision surface can be used to represent a wide variety of shapes with arbitrary genus. In this section we demonstrate the power of our modeling scheme via model fitting examples to a variety of data sets of varying degree of complexity. In all the experiments, normal elements are shaded yellow, while special elements are colored green.

In Fig.4(a) an open limit surface defined by an initial mesh of 61 vertices and 45 faces is shown. The mesh has one extraordinary point of valence 5. The limit surface is acted upon by spring forces as shown in Fig.4(b). An intermediate stage of model evolution is shown in Fig.4(c) and the final fitted model is depicted in

Fig.4(d). It may be noted that the model controlled by the initial mesh reached a local minimum without fitting the points exactly. In order to obtain an exact fit (Fig.4(d)), the control mesh is subdivided once thereby increasing the degrees of freedom (control vertices) of the underlying model. Thus the dynamics can be applied in a hierarchical fashion. Our model can be used to obtain a very fast approximate fitting with fewer number of vertices and an exact fit after more subdivision steps as needed.

In the next experiment, we show the fitting process using spring forces with a closed surface of genus two(Fig.5). The smooth surface is controlled by an initial mesh of 544 faces and 542 vertices, 8 of them being extraordinary points of valence 5. In this experiment, the model has sufficient degrees of freedom and fitted the data points exactly without needing further subdivision of its control mesh.

In Fig.6, we present shape recovery from range data using our model. The model, with 96 faces and 98 vertices, 8 of them being extraordinary vertices of valence 3, was initialized inside a set of 1,779 range data points. The final fitted model has a control polygon of 384 faces with 386 vertices and thus the shape is recovered using a very few number of control points in comparison with the number of data points in the range data set.

In the last experiment, we fit the model to a cerebellum (a cortical structure in brain) given an input of 30 sagittal slices from a MR brain scan. Fig.7(a) depicts a slice from this MRI scan and a slice of the 3D model initialized inside the region of interest is shown in Fig.7(b). Image gradient-based forces are applied to the model and the model deforms under the influence of these forces until maximal conformation to the boundaries of of the desired cerebellum shape. The final fitted model is shown in Fig.7(c). An arbitrary 3D view of the fitted model is shown in Fig.7(d).

5 CONCLUSIONS

In this paper, a dynamic generalization of the Catmull-Clark subdivision surfaces is presented which has numerous applications in geometric modeling and visualization. Apart from providing a direct intuitive way of manipulating shapes, it provides a fast as well as robust way of visualizing large range and volume data sets using very few degrees of freedom. We have presented an analytic formulation of the subdivision scheme, 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 very sharp edges. Also, the initialization is interactive; ideally, initialization should be done automatically on the basis of the input data set. Our future efforts will focus on addressing these issues.

6 ACKNOWLEDGMENT

This research was supported in part by the NSF grant ECS-9210648 to BCV, the NSF career award CCR-9702103 and DMI-9700129 to HQ. We also wish to acknowledge Dr. H. Hoppe and Dr. K. Pulli for the range data and Dr. C.M. Leonard for the brain MRI data.

References

[1] A.A. Ball and D.J.T. Storry. Conditions for tangent plane continuity over recursively generated B-spline surfaces. *ACM Transactions on Graphics*, 7(2):83 – 102, 1988.

[2] M.I.J. Bloor and M.J. Wilson. Representing PDE surfaces in terms of B-splines. *Computer Aided Design*, 22(6):324 – 331, 1990.

[3] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer Aided Design*, 10(6):350 – 355, 1978.

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

[5] N. Dyn, D. Levin and J.A. Gregory. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics*, 9(2):160 – 169, April 1990.

[6] D. Doo and M. Sabin. Analysis of the behavior of recursive division surfaces near extraordinary points. *Computer Aided Design*, 10(6):356 – 360, 1978.

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

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

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

[10] C. Loop. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, Department of Mathematics, 1987.

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

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

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

[14] U. Reif. A unified approach to subdivision algorithms near extraordinary points. *Computer Aided Geometric Design*, (12):153 – 174, 1995.

[15] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4):413–424, 1986.

[16] D. Terzopoulos and H. Qin. Dynamic NURBS with geometric constraints for interactive sculpting. *ACM Transactions on Graphics*, 13(2):103 – 136, April 1994.

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

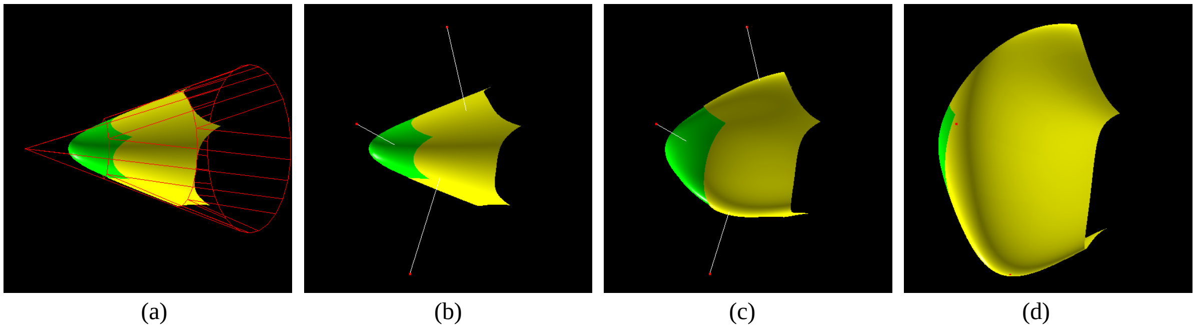
(a) (b) (c) (d)

Figure 4: Fitting the dynamic open surface model to discrete points in 3D : (a) model initialization depicting the associated control mesh, (b) model and data points, (c) intermediate stage of the fitting and (d) the fitted model.

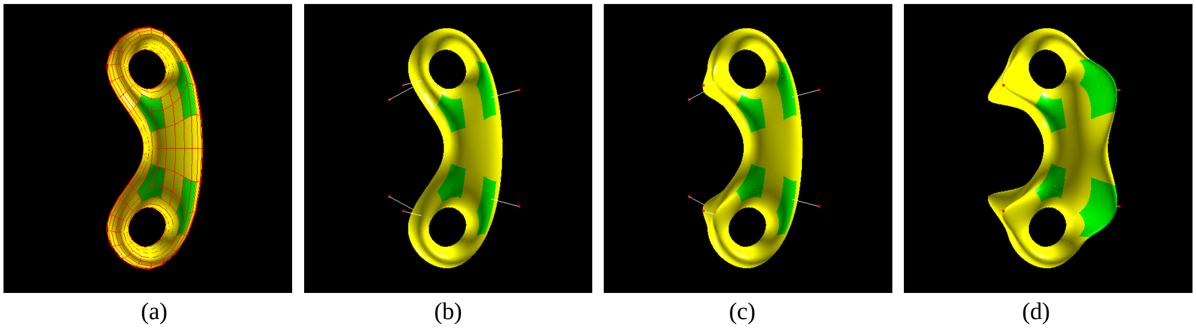
(a) (b) (c) (d)

Figure 5: Fitting the dynamic closed surface model to discrete points in 3D : (a) model initialization depicting the associated control mesh, (b) model and data points, (c)intermediate stage of the fitting and (d) the fitted model.

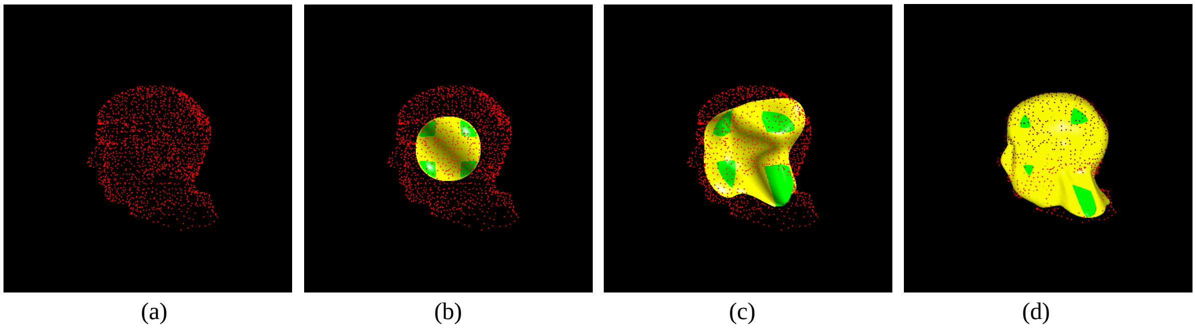
(a) (b) (c) (d)

Figure 6: (a) Range data of a head, (b) data and initialized model, (c) intermediate stage of evolution and (d) the fitted model polygon.

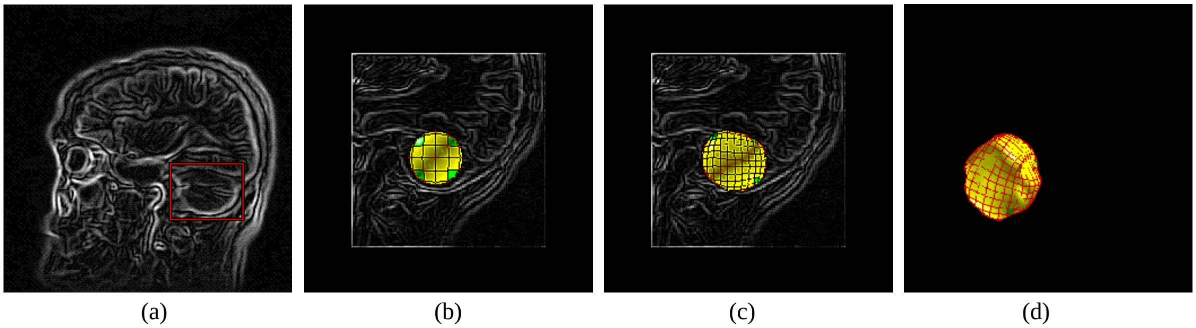
(a) (b) (c) (d)

Figure 7: (a) A slice from a brain MRI along with the region of interest (ROI), (b) a slice of the initialized model inside the ROI superimposed on the data slice, (c) a slice of the fitted model superimposed on the corresponding slice of data and (d) a 3D view of the model fitted to the cerebellum in the ROI.