Dynamic Subdivision Surfaces For Natural Terrain Modeling

Chhandomay Mandal
Dept. of CISE
University of Florida
Gainesville, FL 32611

Hong Qin
Dept. of CS
State University of New York
Stony Brook, NY 11794

Baba C. Vemuri
Dept. of CISE
University of Florida
Gainesville, FL 32611

Abstract

We propose a novel technique for synthesizing realistic terrain models from elevation data by using a new dynamic finite element method-based subdivision surface model for surface reconstruction and a variant of the successive random addition method for fractal surface synthesis. The subdivision surface model is based on the butterfly subdivision scheme, a popular technique for generating smooth surfaces of arbitrary topology, and is embedded in a physics-based modeling paradigm. The initialized model deforms under the influence of the synthesized data forces, at the same time Gaussian noise is added at different levels of subdivision so that a natural-looking surface is reconstructed. We introduce a noise addition technique in the butterfly subdivision scheme to generate fractals and describe a novel technique for locally parameterizing the surface generated by the subdivision algorithm. We also introduce physical quantities required to develop the dynamic subdivision surface model and derive the governing dynamic differential equation using Lagrangian mechanics and the finite element method. Our experiments with real terrain data demonstrate the efficacy of the proposed technique for modeling natural terrains from elevation data.

1 Introduction

Natural terrain modeling from elevation data is an important application area in computer vision and graphics. Roughness is an essential characteristic of the natural terrains and hence traditional surface reconstruction methods using smoothness constraints do not yield the desired results. Fournier et al. [1] first proposed a random displacement technique for synthesizing fractal surfaces which was later modified by Saupe [2] in his successive random addition scheme of generating fractals. Yokoya et al. [3] improved these schemes by adding data constraints. Szeliski and Terzopoulos [4] proposed constrained fractal surfaces using a Gibbs sampler algorithm which was later improved by Vemuri et al. [5]. Arakawa and Krotkov [6] refined the original Gibbs sampler technique by redefining the temperature (control) parameter to obtain better control of roughness in the fitted (constrained) fractal surface. However, the (CPU) execution times reported in their work are very high, thus making their scheme unattractive for many applications.

All the techniques for natural terrain modeling mentioned above usually needs a grid of very large size to model realistic terrains, especially with irregularly spaced data. Rümelin [7] developed an interesting fractal interpolation algorithm which can generate interpolating fractal surfaces for regularly or irregularly spaced data. However, this scheme is not computationally practical for large problems as pointed out in [5].

In this paper, we propose a novel technique for natural terrain modeling using a new dynamic subdivision surface model in conjunction with a variant of the successive random addition method for fractal surface synthesis. Subdivision surfaces are widely used in computer graphics for modeling surfaces of (known) arbitrary geometry/topology. To model a surface of a known topology, first an arbitrary polygonal mesh of the same genus is chosen as the initial mesh a.k.a. control mesh. Next, a fixed set of rules are applied recursively to refine the initial mesh. After each step of subdivision, a finer polygonal mesh with more vertices, edges and faces is obtained. In the limit, a smooth surface of the same genus as of the initial mesh is obtained. It may be noted that *the limit surface is described by the same degrees of freedom as in the initial mesh*, thereby offering a very compact representation of a potentially complicated shape. In [8], we embedded a specific subdivision scheme in a physics-based modeling paradigm and the resulting dynamic subdivision surface model was used to recover complex smooth shapes of arbitrary topology from large range and volume data sets using *very few* degrees of freedom.

The dynamic subdivision surface model proposed in [8] relies on the nature of the specific subdivision rules used. The approach taken in this paper is much more general in the sense that it can be used with *any* type of subdivision schemes. However, we choose the butterfly scheme [9] to demonstrate the concept. This scheme has the added advantage of an interpolatory subdivision scheme where the limit surface *interpolates* the initial mesh whereas in [8], the limit surface only *approximates* the initial mesh. Once we embed the chosen subdivision surface model into physics-based paradigm, the initialized model deforms under the influence of synthesized forces to fit the elevation data via the principle of energy minimization. The interpolatory feature of the butterfly subdivision scheme reduces the computational overhead involved in propagating the external forces from the limit surface to the initial control mesh. Recalling the fact that the limit surface in any subdivision scheme is a function of the degrees of freedom of the initial mesh, once an approximate surface is recovered from the data, the model adopts a new initial mesh which is obtained via a subdivision of the original initial mesh. Note that this new initial mesh and the original initial mesh have the same limit surface, but the new initial mesh has more degrees of freedom thereby facilitating the recovery of the local features in the surface. This process is continued till a prescribed error criteria for fitting the data points is achieved.

The limit surface of any subdivision scheme is smooth, whereas the natural terrains are not. The dynamic subdivision surface model, without any modifications, recovers a smooth surface from elevation data. To recover a natural terrain, we perturb the new vertex positions of the dynamic model at various levels of subdivision in a similar fashion as in the successive random addition method [2]. The variance of the added Gaussian noise at different subdivision levels controls the roughness of the final reconstructed natural terrain. The proposed model can synthesize natural terrains from elevation data using *very few* degrees of freedom. The effectiveness of the proposed technique is illustrated by successful experiments with real elevation data sets in a later section.

2 Formulation

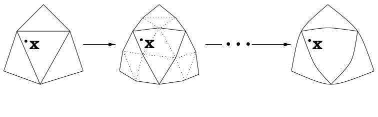

Figure 1: The smoothing effect of the subdivision process on the triangles of the initial mesh.

The butterfly subdivision scheme [9], like many other subdivision schemes used in computer graphics applications, starts with an initial triangular mesh which is also known as control mesh. The vertices of the control mesh are known as control points. In each step of subdivision, the initial (control) mesh is refined through the transformation of each triangular face into a patch with four triangular faces as illustrated in the Fig.1. After one step of refinement, the new mesh in the finer level retains the vertices of each triangular face in the previous level and hence, interpolates the coarser mesh in the previous level. In addition, every edge in each triangular face is spilt by adding a new vertex whose position is obtained by an affine combination of the neighboring vertex positions in the coarser level as shown in Fig.2(a). The name, butterfly subdivision, originated from the "butterfly"-like configuration of the contributing vertices. The weighting factors for different contributing vertex positions are shown in Fig.2(b). The vertex $\mathbf{e}_{12}^{j+1}$ in the $j+1$-th level of subdivision, corresponding to the edge connecting vertices $\mathbf{v}_1^j$ and $\mathbf{v}_2^j$ at level j, is obtained by $\mathbf{e}_{12}^{j+1} = 0.5(\mathbf{v}_1^j+\mathbf{v}_2^j)+2w(\mathbf{v}_3^j+\mathbf{v}_4^j)-w(\mathbf{v}_5^j+\mathbf{v}_6^j+\mathbf{v}_7^j+\mathbf{v}_8^j)$, where $0 \leq w \leq 1$, and $\mathbf{v}_i^j$ denotes the position of the i-th vertex at the j-th level.

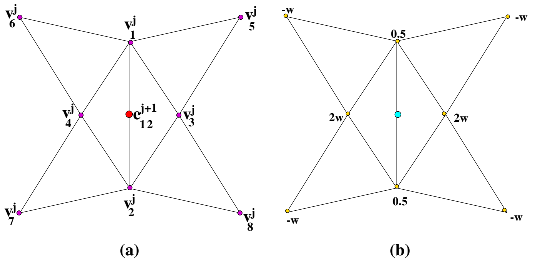

Figure 2: (a) The contributing vertices in the j-th level for the vertex in the j+1-th level corresponding to the edge between $\mathbf{v}_1^j$ and $\mathbf{v}_2^j$; (b) the weighing factors for different vertices.

2.1 "Natural" Surface Generation

The original butterfly subdivision scheme produces a smooth surface in the limit. However, we are trying to recover natural terrains, which are rough. Therefore, we need to modify the butterfly subdivision scheme such that the limit surface looks like "natural" terrain. We adopt a similar technique as in the successive random addition method [2] for generating fractal surfaces. In this successive random addition technique, an equally spaced rectangular grid is refined by interpolating the midpoints of each rectangular cell (thereby dividing each rectangular cell into four rectangular cells) and then all grid positions are perturbed by addition of a Gaussian noise. This process is carried out recursively to obtain a fractal surface whose roughness is controlled by the variance of the added Gaussian noise at different refinement levels. *We have modified the butterfly subdivision scheme by perturbing the vertex positions at various levels of subdivision by addition of a Gaussian noise whose variance controls the roughness of the resulting limit surface.* This process of adding Gaussian noise is similar to that of the successive random addition method mentioned above, the difference being the vertex positions obtained using butterfly subdivision rules are perturbed instead of grid points obtained through midpoint interpolation.

2.2 Local Parameterization

The limit surface defined by the recursive subdivision process is of arbitrary topology where a global parameterization may not be possible. However, we can locally parameterize the limit surface over the domain defined by the initial mesh. The idea is to track any arbitrary point on the initial mesh across the meshes obtained via the subdivision process with Gaussian noise addition, so that a correspondence can be established between the point being tracked in the initial mesh and its mapping on the limit surface. We note that the limit surface can be represented by same number of triangular patches as that of the triangular faces in the initial mesh (see Fig.1). Therefore, the limit surface $\mathbf{s}$ can be expressed as

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

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

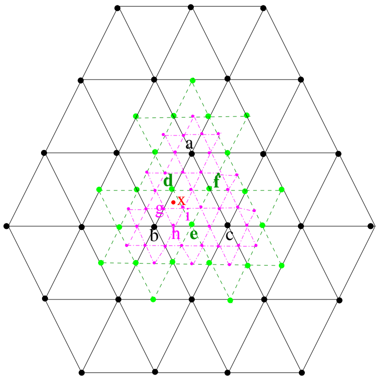

Figure 3: Tracking a point x through various levels of subdivision.

We are now ready to describe the parameterization of the limit surface over the initial mesh. The process is best explained through an example. We choose a simple planar mesh shown in Fig.3 as the initial mesh. An arbitrary point $\mathbf{x}$ inside the triangular face abc is tracked over the meshes obtained through subdivision. It may be noted that *any point inside the triangular patch in the limit surface corresponding to the face abc in the initial mesh depends only on the vertices in the initial mesh which are within the 2-neighborhood of the vertices* $\mathbf{a}, \mathbf{b}$ *and* $\mathbf{c}$ *due to the local nature of the subdivision process.*

In the rest of the formulation, superscripts are used to indicate the subdivision level. For example, $\mathbf{v}^j_{uvw}$ denotes the collection of vertices at level j which controls the patch in the limit surface corresponding to the triangular face uvw at the j-th level of subdivision. Let $\mathbf{v}^0_{abc}$ be the collection of vertices in the initial mesh which are within the 2-neighborhood of the vertices $\mathbf{a}, \mathbf{b}$ and $\mathbf{c}$ (marked black in Fig.3). Let the number of such vertices be r. Then, the vector $\mathbf{v}^0_{abc}$, which is the concatenation of the (x, y, z) positions for all the r vertices, is of dimension $3r$. These r vertices control the triangular patch in the limit surface corresponding to the triangular face abc in the initial mesh. Now, there exists $(3r \times 3r)$ subdivision matrices $(\mathbf{A}_{abc})_t$, $(\mathbf{A}_{abc})_l$, $(\mathbf{A}_{abc})_r$ and $(\mathbf{A}_{abc})_m$ such that $\mathbf{v}^1_{adf} = (\mathbf{A}_{abc})_t \mathbf{v}^0_{abc}$, $\mathbf{v}^1_{bed} = (\mathbf{A}_{abc})_l \mathbf{v}^0_{abc}$, $\mathbf{v}^1_{cfe} = (\mathbf{A}_{abc})_r \mathbf{v}^0_{abc}$, and $\mathbf{v}^1_{def} = (\mathbf{A}_{abc})_m \mathbf{v}^0_{abc}$, where the subscripts t, l, r and m stand for top, left, right and middle respectively (indicating the relative position of the *new* triangle with respect to the *original* triangle), and $\mathbf{v}^1_{adf}, \mathbf{v}^1_{bed}, \mathbf{v}^1_{cfe}$ and $\mathbf{v}^1_{def}$ are the concatenation of the (x, y, z) positions for the vertices in the 2-neighborhood of the corresponding triangle in the newly obtained subdivided mesh. The new vertices in this level of subdivision are marked green in Fig.3. The 2-neighborhood configuration of the vertices in the newly obtained triangles is exactly the same as that of the original triangle, hence local subdivision matrices are square and the vector dimensions are the same. Carrying out one more level of subdivision, along with the old vertices, we get a new set of vertices which are marked in magenta in Fig.3. Adopting a similar approach, we obtain $\mathbf{v}^2_{dgi} = (\mathbf{A}_{bed})_t \mathbf{v}^1_{bed} = (\mathbf{A}_{abc})_t \mathbf{v}^1_{bed}$, $\mathbf{v}^2_{bhg} = (\mathbf{A}_{bed})_l \mathbf{v}^1_{bed} = (\mathbf{A}_{abc})_l \mathbf{v}^1_{bed}$, $\mathbf{v}^2_{eih} = (\mathbf{A}_{bed})_r \mathbf{v}^1_{bed} = (\mathbf{A}_{abc})_r \mathbf{v}^1_{bed}$, $\mathbf{v}^2_{ghi} = (\mathbf{A}_{bed})_m \mathbf{v}^1_{bed} = (\mathbf{A}_{abc})_m \mathbf{v}^1_{bed}$.

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

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

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

$$\mathbf{B}^0_{abc} = \left[\begin{array}{c} \overbrace{\alpha^0_{abc}, \beta^0_{abc}, \gamma^0_{abc}, 0, \ldots, 0}^{r}, \overbrace{0, \ldots, 0}^{r}, \overbrace{0, \ldots, 0}^{r} \\ \overbrace{0, \ldots, 0}^{r}, \overbrace{\alpha^0_{abc}, \beta^0_{abc}, \gamma^0_{abc}, 0, \ldots, 0}^{r}, \overbrace{0, \ldots, 0}^{r} \\ \overbrace{0, \ldots, 0}^{r}, \overbrace{0, \ldots, 0}^{r}, \overbrace{\alpha^0_{abc}, \beta^0_{abc}, \gamma^0_{abc}, 0, \ldots, 0}^{r} \end{array} \right]$$

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

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

Proceeding in a similar fashion, we can derive an expression for $\mathbf{s}^j_{abc}(\mathbf{x})$, the j-th level approximation of $\mathbf{s}_{abc}(\mathbf{x})$. It can be shown that the expression for $\mathbf{s}^j_{abc}(\mathbf{x})$ can be simplified and written as

$$\begin{aligned} \mathbf{s}^j_{abc}(\mathbf{x}) &= \mathbf{B}^j_{uvw}(\mathbf{x})(\mathbf{A}^j_{abc})\mathbf{v}^0_{abc} \\ &= \mathbf{B}^j_{abc}(\mathbf{x})\mathbf{v}^0_{abc}, \end{aligned} \qquad (2)$$

where $\mathbf{x}$ is inside the triangular face uvw at level j and $\mathbf{B}^j_{abc}(\mathbf{x}) = \mathbf{B}^j_{uvw}(\mathbf{x})(\mathbf{A}^j_{abc})$. Finally, we can complete the local parameterization process by writing

$$\mathbf{s}_{abc}(\mathbf{x}) = (\lim_{j \to \infty} \mathbf{B}^j_{abc}(\mathbf{x}))\mathbf{v}^0_{abc} = \mathbf{B}_{abc}(\mathbf{x})\mathbf{v}^0_{abc}. \qquad (3)$$

In the above equation, $\mathbf{B}_{abc}$ is the collection of basis functions at the vertices of $\mathbf{v}^0_{abc}$. It may also be noted that the butterfly subdivision scheme is a stationary subdivision process, and hence new vertex positions are obtained by affine combinations of nearby vertices. This guarantees that each row of the matrices $(\mathbf{A}_{abc})_t, (\mathbf{A}_{abc})_l, (\mathbf{A}_{abc})_r$ and $(\mathbf{A}_{abc})_m$ sums to one. The largest eigenvalue of such matrices is 1 and therefore the limit in Eqn.3 exists. Now, if we assume that the triangular face abc is the k-th face in the initial mesh, then Eqn.3 can be rewritten as

$$\mathbf{s}_k(\mathbf{x}) = \mathbf{B}_k(\mathbf{x})\mathbf{v}^0_k = \mathbf{B}_k(\mathbf{x})\mathbf{A}_k\mathbf{p}, \qquad (4)$$

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

Combining Eqn.1 and Eqn.4, we get

$$\mathbf{s}(\mathbf{x}) = (\sum_{k=1}^{n} \mathbf{B}_k(\mathbf{x})\mathbf{A}_k)\mathbf{p} = \mathbf{J}(\mathbf{x})\mathbf{p}, \qquad (5)$$

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

We now treat the vertex positions in the initial mesh defining the limit surface $\mathbf{s}$ as a function of time in order to develop the new dynamic butterfly subdivision model. The velocity of the surface model can be expressed as $\dot{\mathbf{s}}(\mathbf{x}, \mathbf{p}) = \mathbf{J}(\mathbf{x})\dot{\mathbf{p}}$, where an overstruck dot denotes a time derivative and $\mathbf{x} \in S^0$, S^0 being the domain defined by the initial mesh.

3 Finite Element Implementation

We have already pointed out in Section 2 that the limit surface obtained by the recursive application of the butterfly subdivision rules along with Gaussian noise addition can be represented by triangular patches. We consider each patch in the limit surface as an element. The number of such patches is equal to the number of triangular faces in the initial mesh as mentioned earlier. The governing motion equation of the dynamic model is given by

$$\mathbf{M}\ddot{\mathbf{p}} + \mathbf{D}\dot{\mathbf{p}} + \mathbf{K}\mathbf{p} = \mathbf{f}_p, \qquad (6)$$

where $\mathbf{f}_p$ is the generalized force vector and $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{K}$ are the mass, damping and stiffness matrices of the model. We use the same example as in Section 2 (refer Fig.3) to develop the mass, damping and stiffness matrices for the element corresponding to the triangular face abc in Fig.3. The derivations hold for any element in general.

The mass matrix for the element given by $\mathbf{s}_{abc}$ (Eqn.3) can be written as

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

However, in this paper, we develop a simplified method for the elemental mass matrix derivation. The triangular patch in the limit surface corresponding to the face *abc* in the initial mesh is approximated by a triangular mesh with 4^j faces obtained after j levels of subdivision of the original triangular face *abc* (each subdivision step splits one triangular face into 4 triangular faces). In addition, we choose a discrete mass density function which has non-zero values only at the vertex positions of the j-th subdivision level mesh. Then the mass matrix can be expressed as

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

where k is the number of vertices in the triangular mesh with 4^j faces. This approximation has been found to be very effective and efficient for implementation purposes. The elemental damping matrix can be obtained in an exactly similar fashion.

We assign the internal energy to each element in the limit surface, thereby defining the internal energy of the subdivision surface model. We take a similar approach as in the derivation of the elemental mass and damping matrix and assign the internal energy to a j-th level approximation of the element. In this paper, we assign spring energy to the discretized model as its internal energy. For the example used throughout the paper, this energy at the j-th level of approximation can be written as

$$\begin{aligned} E_{abc} \approx E_{abc}^j &= \frac{1}{2}\sum_{\Omega}\frac{k_{lm}(|\mathbf{v}_l^j - \mathbf{v}_m^j| - \ell_{lm})}{|\mathbf{v}_l^j - \mathbf{v}_m^j|}(\mathbf{v}_l^j - \mathbf{v}_m^j) \\ &= \frac{1}{2}\{\mathbf{v}_{abc}^j\}^T(\mathbf{K}_{abc}^j)\{\mathbf{v}_{abc}^j\}, \end{aligned}$$

where k_{lm} is the spring constant, $\mathbf{v}_l^j$ and $\mathbf{v}_m^j$, the l-th and m-th vertex in the j-th level mesh, are in the 1-neighborhood of each other, Ω is the domain defined by all such vertex pairs, ℓ_{lm} is the natural length of the spring connected between $\mathbf{v}_l^j$ and $\mathbf{v}_m^j$, and $\mathbf{v}_{abc}^j$ is the concatenation of the (x,y,z) positions of all the vertices in the j-th subdivision level of the triangular face *abc* in the initial mesh. It may be noted that the entries in $(\mathbf{K}_{abc}^j)$ depend on the distance between the connected vertices and hence, $(\mathbf{K}_{abc}^j)$, unlike other elemental matrices, is a function of time which needs to be recomputed in each time step.

The expression for the internal energy can be rewritten as

$$\begin{aligned} E_{abc}^j &= \frac{1}{2}\{\mathbf{v}_{abc}^j\}^T(\mathbf{K}_{abc}^j)\{\mathbf{v}_{abc}^j\} \\ &= \frac{1}{2}\{(\mathbf{A}_{abc}^j)\{\mathbf{v}_{abc}^0\}\}^T(\mathbf{K}_{abc}^j)(\mathbf{A}_{abc}^j)\{\mathbf{v}_{abc}^0\} \\ &= \frac{1}{2}\{\mathbf{v}_{abc}^0\}^T\{(\mathbf{A}_{abc}^j)^T(\mathbf{K}_{abc}^j)(\mathbf{A}_{abc}^j)\}\{\mathbf{v}_{abc}^0\}, \end{aligned}$$

where $(\mathbf{A}_{abc}^j)$ and $\mathbf{v}_{abc}^0$ are same as in Eqn.2. Therefore, the expression for the elemental stiffness matrix is given by

$$\mathbf{K}_{abc} \;=\; (\mathbf{A}_{abc}^j)^T(\mathbf{K}_{abc}^j)(\mathbf{A}_{abc}^j).$$

The generalized force vector $\mathbf{f}_p$ in Eqn.6 represents the net effect of all externally applied forces. We attach springs from each point in the elevation data set to the nearest point on the model and the exerted point forces deform the initialized model to fit the given data set.

3.1 Model Subdivision.

The initialized model grows dynamically according to the equation of motion (Eqn.6) and when an approximate fit to the given elevation data set is achieved, the number of control vertices can be increased by replacing the original initial mesh by a new initial mesh obtained by applying a single step of the *butterfly* subdivision algorithm. This increases the number of degrees of freedom to represent the same 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.

4 Results

We present two natural terrain synthesis results in this section. The initialized model is deformed by applying spring forces on its limit surface from the discrete data points. At each time step, every control vertex position is perturbed by adding a random noise drawn from a Gaussian distribution. The variance of the Gaussian distribution determines the roughness of the synthesized surface. In both the experiments, the initialized butterfly subdivision model has an initial (control) mesh with 98 triangular faces and 68 control vertices. The ‘natural" looking limit surface of the initialized model is deformed by the forces exerted from the discrete elevation data points. When an approximate fitting is obtained, the model is subdivided to obtain a closer fit using more degrees of freedom (control vertices) of the new initial mesh. The fitted surface has 1568 triangular faces and 841 control vertices in both the experiments. It may be noted that synthesis of same quality natural terrains using the existing techniques requires a large number of grid points (of the order of 10^5) [4, 5, 6, 7] and hence the proposed technique provides a more compact representation of the synthesized terrain. The elevation data values are scaled to fit an unit cube and the variance of added noise is 10^{-4} for the synthesized fractal surface depicted in Fig.4. The corresponding value of noise variance for fractal surfaces depicted in Fig.5(a), (b) and (c) are 10^{-6}, 10^{-5} and 10^{-4} respectively. In the first experiment, 4096 elevation data points are used whereas the second data set comprised of 3099 elevation values. The error in fitting is approximately one percent in both the experiments.

5 Conclusions

In this paper, we have presented a novel technique for synthesizing natural terrains from elevation data by using a finite element method-based dynamic butterfly subdivision surface model in conjunction with a modified successive random addition technique. We have modified the butterfly subdivision scheme and presented a local parameterization of the subdivision scheme. We have also incorporated the advantages of free-form deformable models in the modified butterfly subdivision scheme and have developed a new type of finite element based on the butterfly subdivision scheme. The efficacy of our model is shown via experiments indicating a promising future of the proposed model in natural terrain modeling.

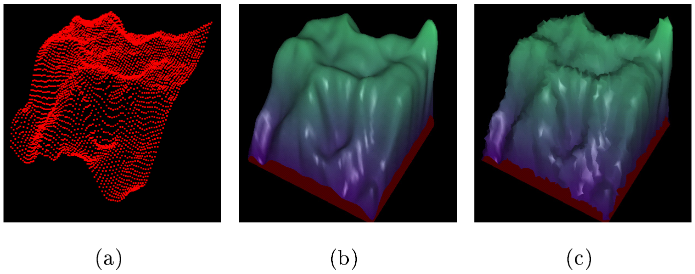

Figure 4: (a) discrete elevation data set (4096 points), (b) fitted dynamic butterfly subdivision surface model with 841 vertices (without noise addition), and (c) fitted dynamic subdivision surface model with 841 vertices when Gaussian noise is added.

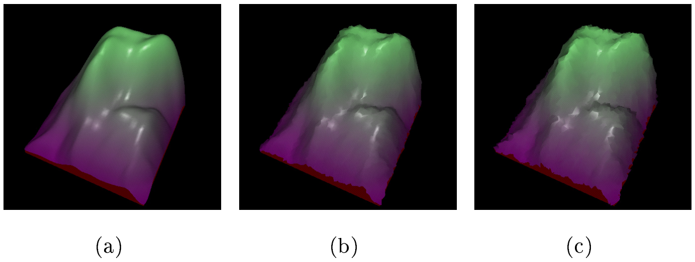

Figure 5: Synthesized natural terrain of different roughness using the dynamic butterfly subdivision surface model with 841 vertices from a data set of 3099 elevation values.

References

[1] A. Fournier, D. Fussel, and L. Carpenter, "Computer rendering of stochastic models," *Communications of the ACM*, vol. 25, no. 6, pp. 371–384, 1982.

[2] D. Saupe, "Algorithms for random fractals," in *The Science of Fractal Images*, H.O. Peitgen and D. Saupe, Eds., pp. 71 – 136. Springer - Verlag, 1988.

[3] N. Yokoya, K. Yamamoto, and N. Funakubo, "Fractal-based analysis and interpolation of 3d natural surfaces and their application to terrain modeling," *Computer Vision, Graphics and Image Processing*, vol. 46, pp. 284–302, 1989.

[4] R. Szeliski and D. Terzopoulos, "From splines to fractals," in *Proceedings of ACM SIGGRAPH'89*, August 1989, pp. 51–60.

[5] B.C. Vemuri, C. Mandal, and S.H. Lai, "A fast Gibbs sampler for synthesizing constrained fractals," *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, no. 4, pp. 337 – 351, October - December 1997.

[6] K. Arakawa and E. Krotkov, "Fractal surface reconstruction for modeling natural terrain," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1993.

[7] W. Rümelin, "Fractal interpolation of random fields of fractional brownian motion," in *Fractal Geometry and Computer Graphics*, J.L. Encarnacao, H.O. Peitgen, G. Sakas and G. Englert, Eds., pp. 122 – 132. Springer-Verlag, 1992.

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

[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.