Constrained Scaling of Catmull-Clark Subdivision Surfaces

Shuhua Lai, Shiping Zou, Fuhua (Frank) Cheng
Graphics & Geometric Modeling Lab, Department of Computer Science
University of Kentucky, Lexington, Kentucky 40506-0046

Abstract. A method to scale a Catmull-Clark subdivision surface while holding the shape and size of specific features (sub-structures) unchanged is presented. The basic idea of the method, *fix-and-stretch*, is similar to a previous approach for trimmed NURBS surfaces [20], i.e., the new surface is formed by fixing selected regions of the given subdivision surface that contain the features, scaling and stretching the remaining part; the goal is to ensure that the resulting surface reflects the shape and curvature distribution of the unconstrainedly scaled version of the given surface. However, the stretching process, the core of the entire process, is more complicated because of the complexity of a subdivision surface's topology. The major contributions of the paper include new strain energy computation techniques and energy optimization techniques for regions around extra-ordinary points. The new method is more powerful than the previous method in that it can handle more complicated shapes and, consequently, can be used for more challenging applications. Test results on several mechanical parts that can not be represented by trimmed NURBS surfaces are included.

Keywords: constrained scaling, subdivision surfaces, constrained deformation, trimmed NURBS surfaces, strain energies, features

1 Introduction

Constrained shape modification refers to the process of altering the shape of an object while holding certain features unchanged. The altering process may involve scaling and/or deformation. This area has attracted the interest of industry recently in that it provides a possibility to reshape an existing model without affecting certain important features and, consequently, avoid expensive redesign process.

Constrained shape deformation has been studied for a while. Topics that have been considered include axial deformation and free-form deformation. Both of which have been thoroughly investigated [1, 3, 4, 5, 6, 9, 10, 12, 15, 16, 17, 18] and some of the techniques have already been implemented by popular CAD/CAM software packages such as ACIS and CATIA.

Constrained shape scaling, as a design tool, has not been studied so extensively yet. The only known results are two techniques for trimmed NURBS surfaces. Features are defined by trimming curves. In the first case [19], an attach-and-deform based approach is used. The new surface is formed by scaling the given NURBS surface according to the scaling requirement and then attaching the original features to the scaled surface at appropriate locations. The attaching process requires a minor deformation of the scaled surface to ensure complete attachment. The second case [20] is fix-and-stretch based. The new surface is formed by fixing selected regions of the given trimmed NURBS surface that contain the trimming curves while scaling/stretching the remaining part of the surface to reach certain boundary condition. The stretching process of the second approach and the deformation process of the first approach are both performed as an energy optimization process to ensure that the resulting surface reflects the shape and curvature distribution of the unconstrainedly scaled version of the given surface. The second method is more robust (can tolerate bigger scaling factors) while the first method is more efficient. These methods can only consider features not intersecting the boundary of the surface.

In this paper, we consider constrained scaling for Catmull-Clark subdivision surfaces. *Subdivision surfaces*, with their capability in modeling/representing complex shape of arbitrary topology [7] and covering both parametric and discrete forms [13, 14], are replacing NURBS surfaces as the next representation scheme in geometric modeling and CAD/CAM. However, they are not used as a major surface representation in CAD/CAM systems yet because of lacking necessary geometric algorithms and modeling techniques in shape design. The work developed in this paper will fill up some gap in that direction and, consequently, will help the process of making subdivision surfaces the next generation surface representation for CAD/CAM applications.

The concept of *fix-and-stretch* [20] is followed in this paper. However, the stretching process, performed as an energy optimization process, is more complicated because of the complexity of a subdivision surface's topology. Efficient energy computation techniques and optimization techniques have been developed to deal with topological complexity of a subdivision surface. The new method is more powerful than the previous method [20] in that it can handle more complicated shapes and, consequently, can be used for more challenging applications.

The remaining part of the paper is arranged as follows. A formal description of the problem is given in Section 2. The basic idea of the proposed method is presented in Section 3. Techniques needed in constructing the new surface are described in Sections 4-8. Test results of the proposed method are shown in Section 9. Concluding remarks are given in Section 10.

2 Problem Formulation

The problem of **constrained scaling of Catmull-Clark subdivision surfaces** can be described as follows: Given a Catmull-Clark subdivision surface $\mathbf{S}$ and a set of *features* $\mathbf{C}_i$ $(i = 1, 2, ..., q)$ on the surface, construct a new surface $\bar{\mathbf{S}}$ whose representation is a scaled version of the given surface $\mathbf{S}$, but carries all the original features $\mathbf{C}_i$.

Given a control mesh, a *Catmull-Clark subdivision surface* (CCSS) is generated by recursively refining the mesh [2, 8]. Each mesh refining step involves the construction of three new types of points: *face points*, *edge points* and *vertex points*. New points are connected to form a new control mesh. These control meshes converge to a limit surface. The limit surface is called a *subdivision surface* because the mesh refining process is a generalization of the uniform B-spline surface *subdivision technique*. Therefore, CCSSs include uniform B-spline surfaces and piecewise Bézier surfaces as special cases. It is also known recently that CCSSs include non-uniform B-spline surfaces and NURBS surfaces as special cases [13]. CCSSs can model/represent complex shape of arbitrary topology because there is no limit on the shape and topology of the control mesh of a CCSS [7]. See Figure 1(d) for the representation of a ventilation control component with a single CCSS. The initial control mesh of the surface and the control mesh after one refinement and two refinements are shown in (a), (b) and (c), respectively. The ventilation control component is a solid with seventeen holes (handles). It can not be represented by a single trimmed B-spline or NURBS surface.

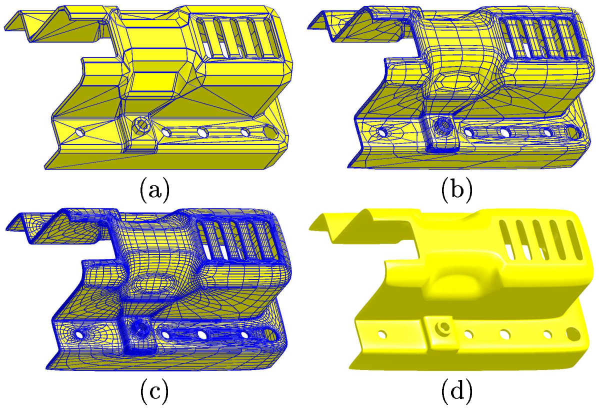

Figure 1: (a) Initial control mesh, (b) control mesh after one refinement, (c) after two refinements, and (d) limit surface of a ventilation control component.

A *feature* is a sub-structure defined by a set of connected patches of the surface. Therefore, each $\mathbf{C}_i$ of $\mathbf{S}$ is defined by a set of control points with related topological information whose limit surface defines a sub-structure of the surface. For instance, each hole in the above figure can be regarded as a feature. Features do not intersect each other and they do not intersect the boundary of the surface if the surface is not closed.

If the given scaling factors in the x, y and z directions are S_x, S_y and S_z, respectively, then the new surface $\bar{\mathbf{S}}$ is expected to be as close to $\hat{\mathbf{S}} = T_s\mathbf{S}$ as possible, where $\hat{\mathbf{S}}$ is the unconstrainedly scaled version of $\mathbf{S}$ by T_s, a scaling matrix with scaling factors S_x, S_y and S_z. The requirement that the new surface carries all the original features $\mathbf{C}_i$ means that $\mathbf{C}_i$ are also features of the new surface $\bar{\mathbf{S}}$, subject to some translation and rotation.

3 Basic Idea

In general, due to change of curvature distribution after a scaling process, it is not possible for the new surface $\bar{\mathbf{S}}$ to have exactly the same shape and dimension as the unconstrainedly scaled surface $\hat{\mathbf{S}} = T_s\mathbf{S}$ while carrying all the original features. An approximation method has to be used to construct $\bar{\mathbf{S}}$. In this work, the new surface will be constructed using a *fix-and-stretch* based approach [20].

The main idea of this approach is to fix regions of the given subdivision surface that contain the features while scaling and stretching the remaining part of the surface until some conditions are reached (see Figure 2). The surface is divided into three parts, the features (region I of the smaller ellipse in Figure 2), neighboring regions of the features (region II in Figure 2), and the remaining part (region III in Figure 2). The features that need to be fixed during the scaling and stretching process have to be transformed to appropriate locations first (see region I of Figure 2). Region III is simply scaled using the given scaling factors. Region II is stretched to provide a smooth connection between the relocated features and the scaled region III.

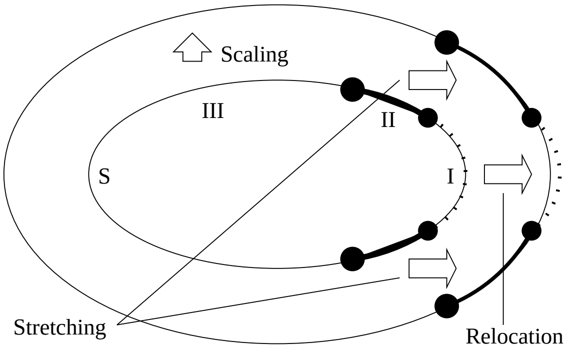

Figure 2: Basic idea of the fix-and-stretching approach.

The stretching process ensures that the shape and curvature distribution of $\bar{\mathbf{S}}$ are as close to those of the unconstrainedly scaled version of the given CCSS, $\hat{\mathbf{S}}$, as possible, while carrying all the original features $\mathbf{C}_i$. This is achieved by minimizing a shape-preserving objective function defined on the difference of these two surfaces on neighboring regions of the features. Efficient techniques have been developed for the energy evaluation process and the optimization process. The stretching process does not change the topological structure of the subdivision surface. Therefore, the resulting surface $\bar{\mathbf{S}}$ is again a CCSS.

The main steps of our approach are shown below.

1. Surface partitioning
2. Feature relocation
3. Setting up shape-preserving objective function
4. Energy evaluation
5. Energy optimization

The last two steps are technically more difficult. But the first two steps are functionally more important because they determine the outcome of the stretching process. Details of these steps are given subsequently.

4 Surface Partitioning

This step first subdivides the surface three times to ensure the precision of the energy computation process (Section 7), then constructs a *neighboring region* for each specified feature $\mathbf{C}_i$ of the surface. These regions effectively divide the control vertices of $\mathbf{S}$ into three groups: *Type-I*, *Type-II* and *Type-III*. Type-I vertices are control vertices of patches that contain the features. Type-II vertices are vertices in the neighboring regions of the features. Type-III vertices are the remaining vertices. Type-III vertices will be unconstrainedly scaled using the given scaling factors; Type-I vertices will be move to new locations through a translation and a rotation. The optimization process is performed on Type-II vertices only. The construction of the neighboring regions for the features is tricky. A small neighboring region will make the stretching process more efficient, but at the cost of getting sharp turns around boundaries of the features. A large neighboring region will give smoother curvature distribution around the boundaries of the features, but might generate sharp edges or corners around the boundaries of Type-III region (actually, if Type-III region is empty, one can not perform constrained scaling at all, due to lack of boundary conditions). Here the neighboring region is obtained by expanding outward uniformly (for a fixed number of patches) for each feature edge. The width of the band depends on the complexity of the features and the entire object. A guideline is to expand to an area where the curvature variation of the surface is small.

5 Relocating Features

This step is to move each feature of $\mathbf{S}$, $\mathbf{C}_i$, to an appropriate location that is not only as close to the unconstrainedly scaled surface $\hat{\mathbf{S}} = T_s\mathbf{S}$ as possible but also with an appropriate orientation. The first requirement is to ensure that the Euclidean distance between the new surface $\bar{\mathbf{S}}$ and the unconstrainedly scaled version $\hat{\mathbf{S}}$ around the features is as small as possible. The second requirement is to ensure that the curvature difference between the new surface $\bar{\mathbf{S}}$ and the unconstrainedly scaled version $\hat{\mathbf{S}}$ around the features is small as well.

To achieve the second requirement, for each feature $\mathbf{C}_i$ of $\mathbf{S}$, a least squares method is used to find a plane P_i whose distance to the boundary points of the feature is a minimum. For each scaled feature $T_s\mathbf{C}_i$ of $\hat{\mathbf{S}}$, a least squares method is used to find a plane $P_i^{'}$ whose distance to the boundary points of the scaled feature is a minimum. A rotation is then performed on $\mathbf{C}_i$ to make the normal of P_i the same as that of $P_i^{'}$.

After that, a least squares method is used to compute a displacement vector $\mathbf{\Delta}_i$ for each (rotated) feature $\mathbf{C}_i$

so that by adding $\boldsymbol{\Delta}_i$ to the control points of the (rotated) feature $\mathbf{C}_i$, one would move the (rotated) feature to a location satisfying the first requirement. $\boldsymbol{\Delta}_i$ is computed by minimizing the following summation

$$\sum_j |(\mathbf{V}_{i,j} + \boldsymbol{\Delta}_i) - \mathbf{V}^{'}_{i,j}|$$

where $\mathbf{V}_{i,j}$ are boundary points of the feature $\mathbf{C}_i$ and $\mathbf{V}^{'}_{i,j}$ are boundary points of the scaled feature $T_s\mathbf{C}_i$.

6 Setting Up Shape-Preserving Objective Function

A *shape-preserving objective function* is used to determine type two control points of the new surface $\bar{\mathbf{S}}$ in an optimization process. The object function is defined as an energy function of the displacement function

$$\mathbf{Q} = \bar{\mathbf{S}} - T_s\mathbf{S} \tag{1}$$

for region II, so that by minimizing the energy of the displacement function $\mathbf{Q}$, one can minimize the shape change of the new surface in region II. The energy function is defined as follows:

$$E(\mathbf{Q}) = \alpha E_b + \beta E_{st} + \gamma E_{sp} \tag{2}$$

where E_b, E_{st} and E_{sp} are *bending strain energy*, *stretching strain energy* and *spring potential energy*[11, 21] of $\mathbf{Q}$,

$$\begin{array}{rcl} E_b & = & \frac{1}{2}\int\int_D[(\mathbf{Q}_{uu}+\mathbf{Q}_{vv})^2 \\ & & \quad - 2(\mathbf{Q}_{uu}\mathbf{Q}_{vv} - \mathbf{Q}_{uv}^2)]dudv \\ E_{st} & = & \frac{1}{2}\int\int_D[(\mathbf{Q}_u^2+\mathbf{Q}_v^2)+2(\mathbf{Q}_u\mathbf{Q}_v)]dudv \\ E_{sp} & = & \frac{1}{2}\int\int_D \mathbf{Q}^2dudv \end{array} \tag{3}$$

and α, β and γ are weights to be determined. The values of the weights α, β and γ in (2) are set one here. This leads to the minimization of the *average* energy of E_b, E_{st} and E_{sp}. A more complicated setting of these weights is considered in [20].

7 Energy Evaluation

The energy items required in the shape-preserving objective function (2) are evaluated here for patches contained in region II. We present compact energy forms for patches not adjacent to an extra-ordinary point first.

A patch $\mathbf{S}$ not adjacent to an extra-ordinary point is a regular bicubic B-spline surface patch and can be express as

$$\mathbf{S}(u,v) = U\ M\ G\ M^T\ V^T$$

where U is the u-parameter vector, V is the v-parameter vector, M is the B-spline coefficient matrix and G is the 4×4 control point matrix. All the vectors used in this paper are row vectors. The spring energy, stretching energy and bending energy of such a patch $\mathbf{S}$ can be expressed as follows:

$$\begin{array}{rcl} E_{sp}(\mathbf{S}) & = & \frac{1}{2}\sum_{i=0}^{3}(D_iMGM^TDMG^TM^TY_i^T), \\ E_{st}(\mathbf{S}) & = & \frac{1}{2}\sum_{i=0}^{3}(A_iMGM^TDMG^TM^TY_i^T) \\ & & +\frac{1}{2}\sum_{i=0}^{3}(D_iMGM^TAMG^TM^TY_i^T) \\ & & +\sum_{i=0}^{3}(B_iMGM^TBMG^TM^TY_i^T), \\ E_b(\mathbf{S}) & = & \frac{1}{2}\sum_{i=0}^{3}(C_iMGM^TDMG^TM^TY_i^T) \\ & & +\frac{1}{2}\sum_{i=0}^{3}(D_iMGM^TCMG^TM^TY_i^T) \\ & & +\sum_{i=0}^{3}(A_iMGM^TAMG^TM^TY_i^T), \end{array} \tag{4}$$

respectively, where

$$\begin{array}{rcl} A & = & \begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 4/3 & 3/2 \\ 0 & 1 & 3/2 & 9/5 \end{bmatrix}, \\ B & = & \begin{bmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 1/2 & 2/3 & 3/4 \\ 0 & 1/3 & 1/2 & 3/5 \\ 0 & 1/4 & 2/5 & 1/2 \end{bmatrix}, \\ C & = & \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 6 \\ 0 & 0 & 6 & 12 \end{bmatrix}, \\ D & = & \begin{bmatrix} D_0 \\ D_1 \\ D_2 \\ D_3 \end{bmatrix} = \begin{bmatrix} 1 & 1/2 & 1/3 & 1/4 \\ 1/2 & 1/3 & 1/4 & 1/5 \\ 1/3 & 1/4 & 1/5 & 1/6 \\ 1/4 & 1/5 & 1/6 & 1/7 \end{bmatrix}, \\ Y & = & \begin{bmatrix} Y_0 \\ Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{array} \tag{5}$$

The proof of (4) is shown in Appendix A. If we define $f(U, G_1, G_2, X, V)$ as follows:

$$f(U, G_1, G_2, X, V) = UMG_1M^TXMG_2^TM^TV^T \tag{6}$$

then the total energy of $\mathbf{S}$ can be expressed as:

$$\begin{array}{rcl} E & = & E_{sp} + E_{st} + E_b \\ & = & \sum f((A_i + C_i + D_i)/2, G, G, D, Y_i) \\ & & +\sum f((2A_i + D_i)/2, G, G, A, Y_i) \\ & & +\sum f(B_i, G, G, B, Y_i) \\ & & +\sum f(D_i/2, G, G, C, Y_i). \end{array} \tag{7}$$

The total energy of an extra-ordinary patch (a patch that has an extra-ordinary vertex) has to be computed with care. The current energy definition of a regular 4×4 bicubic B-spline surface patch $\mathbf{S}$ is not divisible, i.e., the energy of $\mathbf{S}$ is not equal to the sum of the energies of its four subpatches $(\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3, \mathbf{S}_4)$ after a subdivision. Hence, to make the energy items defined in (3) comparable, they must be calculated to the same subdivision level. This requirement, however, presents a problem for extra-ordinary patches.

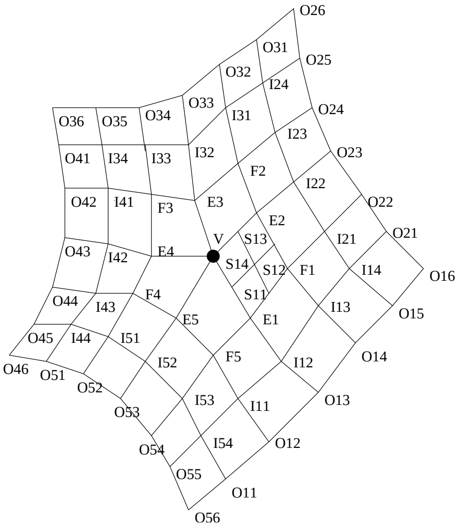

Figure 3: A Catmull-Clark subdivision on $\mathbf{S}$ generates three regular bicubic B-spline patches $\mathbf{S}_{11}$, $\mathbf{S}_{12}$ $\mathbf{S}_{13}$.

Let $\mathbf{V}$ be an extra-ordinary point of valence n. Let $\mathbf{S}$ be a patch adjacent to $\mathbf{V}$ and the control point vector of $\mathbf{S}$ be N. N contains $2n + 8$ points (see Figure 3 for an example when $n = 5$. The control point vector of $\mathbf{S}$ in this case is shown below).

$$\begin{aligned} N = \{\ & \mathbf{V}, \mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3, \mathbf{E}_4, \mathbf{E}_5, \mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \\ & \mathbf{F}_4, \mathbf{F}_5, \mathbf{I}_{11}, \mathbf{I}_{12}, \mathbf{I}_{13}, \mathbf{I}_{14}, \mathbf{I}_{21}, \mathbf{I}_{22}, \mathbf{I}_{23}\ \} \end{aligned} \tag{8}$$

The energies of $\mathbf{S}$ can not be calculated using the above formulas directly because the control point set of $\mathbf{S}$ is not a 4×4 grid. If we perform a Catmull-Clark subdivision on $\mathbf{S}$, three of the resulting subpatches, $\mathbf{S}_{11}$, $\mathbf{S}_{12}$ and $\mathbf{S}_{13}$, are regular bicubic B-spline patches (see Figure 3) and, consequently, can be evaluated using the above formulas. If we perform a Catmull-Clark subdivision on the new extra-ordinary subpatch $\mathbf{S}_{14}$, one gets three more regular bicubic B-spline patches, $\mathbf{S}_{21}$, $\mathbf{S}_{22}$, $\mathbf{S}_{23}$, and a new extra-ordinary subpatch $\mathbf{S}_{24}$. By iteratively repeat this process, one gets a sequence of regular bicubic B-spline patches $\{\ \mathbf{S}_{L1}, \mathbf{S}_{L2}, \mathbf{S}_{L3} \mid L = 1, 2, 3, \ldots\}$. But we can not use the sum of their energies to represent the energy of the extra-ordinary patch because these subpatches are of different subdivision levels. To overcome this problem, note that for a regular (4×4) bicubic B-spline surface patch $\mathbf{S}_{L-1}$ and its four subpatches $(\mathbf{S}_{L1}, \mathbf{S}_{L2}, \mathbf{S}_{L3}, \mathbf{S}_{L4})$ after a subdivision, we have the following result for their energies:

$$\lim_{L \to \infty} \frac{E(S_{L-1})}{E(S_{L1}) + E(S_{L2}) + E(S_{L3}) + E(S_{L4})} = \frac{1}{4}.$$

This follows from the observation that when L is sufficiently large, $\mathbf{S}_{L-1}$ and its subpatches all have the same height. On the other hand, they all have the same parameter space $[0, 1] \times [0, 1]$. Therefore, the energy of each subpatch is the same as $\mathbf{S}_{L-1}$. The ratio converges to $1/4$ quite fast. According to our experiments, the ratio is already close enough to $1/4$ when $L = 3$ (see Figure 4 for the rates of convengence on ten randomly chosen 4×4 patches). This shows that energies of subpatches from different levels can be considered together if the energy of a subpatch is divided by 4^L first where L $(L \geq 3)$ is the subdivision level of the subpatch. Therefore, the total energy of $\mathbf{S}$ can be expressed as

$$E(\mathbf{S}) = \sum_{L=1}^{\infty} \frac{E(\mathbf{S}_{L1}) + E(\mathbf{S}_{L2}) + E(\mathbf{S}_{L3})}{4^L} \tag{9}$$

with $E(\mathbf{S}_{Ld}) = E_{sp}(\mathbf{S}_{Ld}) + E_{st}(\mathbf{S}_{Ld}) + E_b(\mathbf{S}_{Ld})$, $d = 1, 2, 3$.

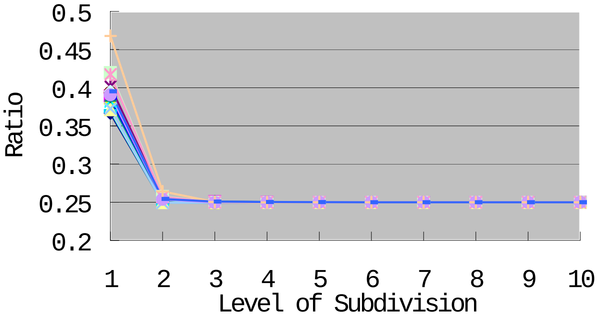

Figure 4: Rates of convergence of eq. (9) on ten randomly chosen 4×4 patches.

The control point matrix G_{Ld} of each $\mathbf{S}_{Ld}$ can be computed as follows:

$$G_{L,d} = \sum_{j=0}^{3} P_{dj} \bar{S} S^{L-1} N^T Y_j^T \tag{10}$$

where S is the $K \times K$ Catmull-Clark subdivision matrix $(K = 2n + 8)$, $\bar{S}$ is the $(K + 9) \times (K + 9)$ extended

Catmull-Clark subdivision matrix [14], P_{dj} is a $4\times(K+9)$ *picking matrix* and Y_j are defined in (5). Each row of the picking matrix P_{dj} picks an appropriate entry from $\bar{S}S^{L-1}N^T$ and P_{dj} depends on d and j only. The proof of (10) is shown in Appendix B. The computation of each of the three energies involved in (9) requires the evaluation of the function f defined in (6) for each G_{Ld}. With these values, the energy $\mathrm{E}(\mathbf{S})$ of the extraordinary patch $\mathbf{S}$ can then be expressed as

$$\begin{array}{rcl} E(\mathbf{S}) & = & \sum_{d=1}^{3}\sum_{i=0}^{3}\sum_{0\le j,k\le 3}\{ \\ & & h_{jk}((A_i+C_i+D_i)/2, D, d)R_N\bar{h}_k(Y_i,d) \\ & & +h_{jk}((2A_i+D_i)/2, A, d)R_N\bar{h}_k(Y_i,d) \\ & & +h_{jk}(B_i, B, d)R_N\bar{h}_k(Y_i,d) \\ & & +h_{jk}(D_i/2, C, d)R_N\bar{h}_k(Y_i,d)\} \end{array} \tag{11}$$

where

$$\begin{array}{l} h_{jk}(U,X,d) = (Y_jM^TXMY_k^T)UMP_{dj}\bar{S}, \\ \bar{h}_k(V,d) = \bar{S}^T(P_{dk})^TM^TV^T, \\ R_N = \sum_{L=1}^{\infty} S^{L-1}N^TN(S^{L-1})^T \ / \ 4^L. \end{array} \tag{12}$$

R_N satisfies the following condtion.

$$R_N - \tfrac{1}{4}SR_NS^T = \tfrac{1}{4}N^TN \tag{13}$$

The proof of (11) is shown in Appendix C.

8 Energy Optimization

This step is to minimize the energy of the displacement function $\mathbf{Q} = \bar{\mathbf{S}} - T_s\mathbf{S}$ to find new locations of Type-II vertices. In the following, for the sake of convenience, we shall also use $E(G)$ to denote the energy of a surface whose control mesh is G. The energy of a CCSS is the sum of its patches' energies. Therefore, the energy of $\mathbf{Q}$ can be expressed as

$$E(\bar{G}-\hat{G}) = \sum_i E(\bar{G}_i - \hat{G}_i)$$

where $\bar{G}$ and $\hat{G}$ are the control meshes of $\bar{\mathbf{S}}$ and $T_s\mathbf{S}$, respectively, and $\bar{G}_i$ and $\hat{G}_i$ are the control meshes of patch i of $\bar{\mathbf{S}}$ and $T_s\mathbf{S}$, respectively. $\bar{G}$ is unkown. To minimize E, we set partial derivative of E with respect to each control point in region II to zero,

$$\frac{\partial E(\bar{G}-\hat{G})}{\partial \mathbf{P}_k} = \sum_i \frac{\partial E(\bar{G}_i-\hat{G}_i)}{\partial \mathbf{P}_k} = 0, \qquad \mathbf{P}_k \in \bar{G} \tag{14}$$

Each $E(\bar{G}_i - \hat{G}_i)$ is a quadratic expression, so equation (14) is a system of linear equations $Ax = b$ where x is the vector of unknows $\mathbf{P}_k$. By solving this system, we get new location of each point in region II. The point is how to efficiently find A and b. It is sufficient to show the process for a patch of $\mathbf{Q} = \bar{\mathbf{S}} - T_s\mathbf{S}$ only. For the sake of notation simplicity, we shall use G and G_1, instead of $\bar{G}_i$ and $\hat{G}_i$, to represent the control meshes of a patch of $\bar{\mathbf{S}}$ and $T_s\mathbf{S}$ in region II, respectively.

We first consider the case of a 4×4 patch. Let

$$\begin{array}{l} F(U,G,G_1,X,V) \\ \quad = UM(G-G_1)M^TXM(G-G_1)^TM^TV^T \\ \quad = f(U,G,G,X,V) - f(U,G_1,G,X,V) \\ \qquad - f(U,G,G_1,X,V) + f(U,G_1,G_1,X,V) \end{array} \tag{15}$$

G_1 is fixed and G is unknown. For each $\mathbf{P}_{ij}$ of G, we have

$$\begin{array}{l} \partial F(U,G,G_1,X,V)/\partial \mathbf{P}_{ij} \\ \quad = f(U,G',G,X,V) + f(U,G,G',X,V) \\ \qquad - f(U,G_1,G',X,V) - f(U,G',G_1,X,V) \end{array} \tag{16}$$

where G' is the derivative of G with respect to $\mathbf{P}_{ij}$, i.e.

$$(G')_{mn} = \begin{cases} 1, & if\ (m,n) = (i,j) \\ 0, & otherwise. \end{cases}$$

Since $F(U,G,G_1,X,V)$ is a quadratic polynomial in $\mathbf{P}_{ij}$, the partial derivative of $F(U,G,G_1,X,V)$ is linear in $\mathbf{P}_{ij}$ and, consequently, can be put in the following form

$$\frac{\partial F(U,G,G_1,X,V)}{\partial \mathbf{P}_{ij}} = \sum_{m,n} \Psi_{mn}\mathbf{P}_{mn} + \Theta \tag{17}$$

where Θ and Ψ_{mn} are constants. Θ can be obtained by setting all $\mathbf{P}_{mn}$ to zero, i.e., setting $G = G_0$ in (17) where

$$(G_0)_{mn} = \begin{cases} 0, & \mathbf{P}_{mn}\ is\ a\ variable\ in\ G \\ P_{mn}, & \mathbf{P}_{mn}\ is\ a\ constant\ in\ G. \end{cases} \tag{18}$$

Ψ_{mn}, the coefficient of $\mathbf{P}_{mn}$, can be obtained by setting all other variable $\mathbf{P}_{ij}$ to zero, setting $\mathbf{P}_{mn}$ to one, and then subtracting Θ from the result. Θ and Ψ_{mn} can be expressed as follows.

$$\Theta(U,G,G_1,X,V) = \frac{\partial F(U,G_0,G_1,X,V)}{\partial \mathbf{P}_{ij}}, \tag{19}$$

$$\Psi_{mn}(U,G,G_1,X,V) = \frac{\partial F(U,G_0+G',G_1,X,V)}{\partial \mathbf{P}_{ij}} - \Theta \tag{20}$$

where G_0 is defined in (18).

Therefore, following eq. (7), the energy of the displacement function for a 4 × 4 patch is

$$\begin{aligned} &E(G - G_1) = \\ &\quad \textstyle\sum_{i=0}^{3} F((A_i + C_i + D_i)/2, G, G_1, D, Y_i) \\ &\quad + \textstyle\sum_{i=0}^{3} F((2A_i + D_i)/2, G, G_1, A, Y_i) \\ &\quad + \textstyle\sum_{i=0}^{3} F(B_i, G, G_1, B, Y_i) \\ &\quad + \textstyle\sum_{i=0}^{3} F(D_i/2, G, G_1, C, Y_i). \end{aligned} \tag{21}$$

Then the partial derivative of $E(G - G_1)$ can be put in a linear form, similar to eq. (17), as follows:

$$\frac{\partial E(G - G_1)}{\partial \mathbf{P}_{ij}} = \sum_{m,n} \xi_{mn} P_{mn} + \zeta$$

where the constant term is

$$\begin{aligned} \zeta = &\textstyle\sum_{i=0}^{3} \Theta((A_i + C_i + D_i)/2, G, G_1, D, Y_i) \\ &+ \textstyle\sum_{i=0}^{3} \Theta((2A_i + D_i)/2, G, G_1, A, Y_i) \\ &+ \textstyle\sum_{i=0}^{3} \Theta(B_i, G, G_1, B, Y_i) \\ &+ \textstyle\sum_{i=0}^{3} \Theta(D_i/2, G, G_1, C, Y_i) \end{aligned} \tag{22}$$

and the coefficient of $\mathbf{P}_{mn}$ is

$$\begin{aligned} \xi_{mn} = &\textstyle\sum_{i=0}^{3} \Psi_{mn}((A_i + C_i + D_i)/2, G, G_1, D, Y_i) \\ &+ \textstyle\sum_{i=0}^{3} \Psi_{mn}((2A_i + D_i)/2, G, G_1, A, Y_i) \\ &+ \textstyle\sum_{i=0}^{3} \Psi_{mn}(B_i, G, G_1, B, Y_i) \\ &+ \textstyle\sum_{i=0}^{3} \Psi_{mn}(D_i/2, G, G_1, C, Y_i). \end{aligned} \tag{23}$$

We now consider the case of an extraordinary patch. From equation (13), we have

$$R_{(N-N_1)} - \frac{1}{4} S R_{(N-N_1)} S^T = \frac{1}{4}(N - N_1)^T (N - N_1)$$

R_{N-N_1} is an implicit function of $(N - N_1)^T(N - N_1)$. N_1 is fixed and N is unknonw. If $\mathbf{P}_k$ is a variable in N then

$$\begin{aligned} &R'((N - N_1)^T (N - N_1))' \\ &-\tfrac{1}{4} S R'((N - N_1)^T (N - N_1))' S^T \\ &= \tfrac{1}{4}((N - N_1)^T (N - N_1))' \end{aligned} \tag{24}$$

where

$$\begin{aligned} &((N - N_1)^T (N - N_1))' \\ &= (N')^T (N - N_1) + (N - N_1)^T (N') \end{aligned} \tag{25}$$

and R' and N' are the derivatives of $R_{(N-N_1)}$ and N with respest to $\mathbf{P}_k$, respectively. This is a system of linear equations. Assume the solution of equation (24) is $R'_{(N-N_1)} = g(N, N_1, N')$, where N' is defined as follows:

$$(N')_m = \begin{cases} 1, & if\ m = k \\ 0, & otherwise. \end{cases}$$

Then the partial derivative of the energy of an extraordinary patch can be put in the following form

$$\frac{\partial E(N - N_1)}{\partial \mathbf{P}_k} = \sum_m \xi_m \mathbf{P}_m + \zeta$$

where the constant term and the coefficients of $\mathbf{P}_k$ can be obtained similarly to those obtained in (19) and (20). They can be expressed as follows:

$$\begin{aligned} \zeta = &\textstyle\sum_{d=1}^{3} \sum_{i=0}^{3} \sum_{0 \le j,k \le 3} \{ \\ &h_{jk}((A_i + C_i + D_i)/2, D, d) g(N_0, N_1, N') \bar{h}_k(Y_i, d) \\ &+ h_{jk}((2A_i + D_i)/2, A, d) g(N_0, N_1, N') \bar{h}_k(Y_i, d) \\ &+ h_{jk}(B_i, B, d) g(N_0, N_1, N') \bar{h}_k(Y_i, d) \\ &+ h_{jk}(D_i/2, C, d) g(N_0, N_1, N') \bar{h}_k(Y_i, d) \} \end{aligned} \tag{26}$$

$$\begin{aligned} \xi_m = &\textstyle\sum_{d=1}^{3} \sum_{i=0}^{3} \sum_{0 \le j,k \le 3} \{ \\ &h_{jk}((A_i + C_i + D_i)/2, D, d) g(N_0 + N', N_1, N') \bar{h}_k(Y_i, d) \\ &+ h_{jk}((2A_i + D_i)/2, A, d) g(N_0 + N', N_1, N') \bar{h}_k(Y_i, d) \\ &+ h_{jk}(B_i, B, d) g(N_0 + N', N_1, N') \bar{h}_k(Y_i, d) \\ &+ h_{jk}(D_i/2, C, d) g(N_0 + N', N_1, N') \bar{h}_k(Y_i, d) \} \\ &- \zeta \end{aligned} \tag{27}$$

with N_0 being defined as follows:

$$(N_0)_m = \begin{cases} 0, & if\ N_m\ is\ a\ variable\ in\ G, \\ N_m, & if\ N_m\ is\ a\ fixed\ point\ in\ G. \end{cases}$$

Once we have the ζ and ξ values for each patch and variable point, we combine them to form a linear system $Ax = b$. The new locations of the control points in region II are obtained by solving this system for x.

9 Implementation & Test Results

The proposed approach has been implemented in C++ using OpenGL as the supporting graphics system on an SGI platform. Three test cases are presented here. These include a rocker arm (Figures 5 and 6), a ventilation control component (Figures 7 and 8) and a marker cap (Figure 9). These examples are mechanical parts with holes. Since the boundary representations of these mechanical parts are closed surfaces with handles, none of them can be represented by a single trimmed NURBS surface.

In each example, the original object is shown on the left and its shape after the constrained scaling process is shown on the right. In the first and the third cases (Figures 5, 6 and 9), only one feature is fixed in the constrained scaling process. These features are shown in blue. In the second case (Figures 7 and 8), three

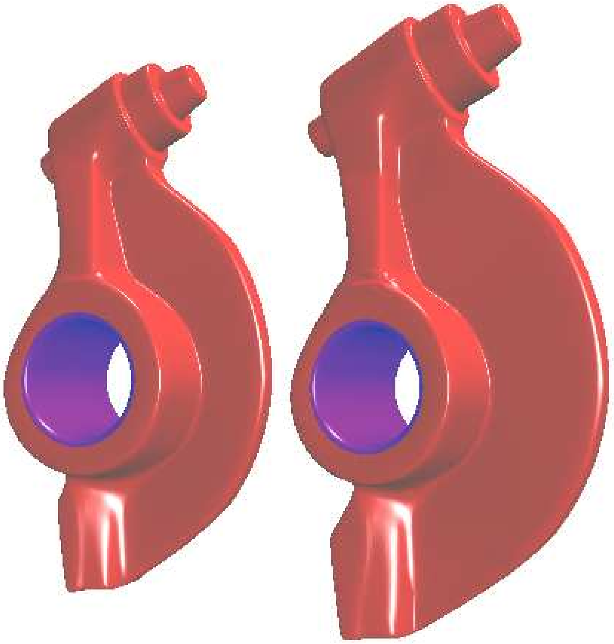

Figure 5: Constrained scaling of a rocker arm: before scaling (left), after sacling (right) with $S_x = 1.3$, $S_y = 1.2$ $S_z = 1.1$.

holes and five holes are fixed in the constrained scaling process, respectively. Those fixed holes are also shown in blue even though they can not be seen due to the direction of projection. It is easy to see that the features remain the same in both shape and dimension after the constrained scaling process in all cases. These examples represent great challenge in practical applications because of the complexity of the shape around the features.

10 Conclusion

A fix-and-stretch based constrained scaling method for CCSSs is presented. The new surface is constructed by fixing the the regions containing the features and stretching the remaining parts until certain conditions are reached. The stretching process is realized through an energy optimization process to ensure the resulting surface reflects the shape and curvature distribution of the unconstrainedly scaled version of the given surface. Efficient energy computation techniques and elegant optimization techniques for regions with extra-ordinary points have been developed to handle the topological complexity of a subdivision surface. The new method is more powerful than the previous method [20] in that it can handle more complicated shapes; it can be used for most free-form objects as long as a CCSS representation of the object is available.

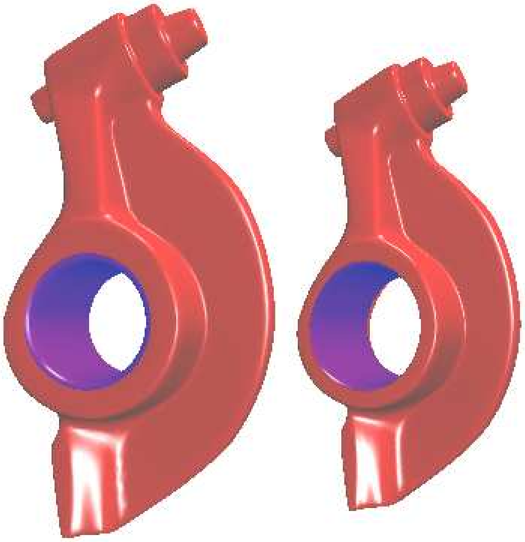

Figure 6: Constrained scaling of a rocker arm: before scaling (left), after sacling (right) with $S_x = 0.8$, $S_y = 0.85$ $S_z = 0.9$.

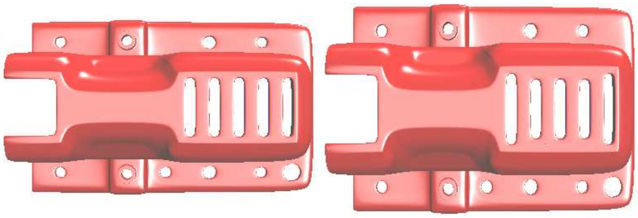

Figure 7: Constrained scaling of a ventilation control component with three features: before scaling (left), after sacling (right) with $S_x = 1.0$, $S_y = 1.2$ $S_z = 1.0$.

The new method can not tolerate scaling factors bigger than 1.4. The safe scaling range provided by this method is between 0.8 to 1.3. This is enough for most of the reshaping tasks in the automotive industry as the scaling factors used in the reshaping process there are usually smaller than 1.2.

References

[1] Borrel, P. and Rappoport, A., Simple Constrained Deformations for Geometric Modeling and Interactive Design, *ACM Tansactions on Graphics*, 13(2), 1994, 137-155.

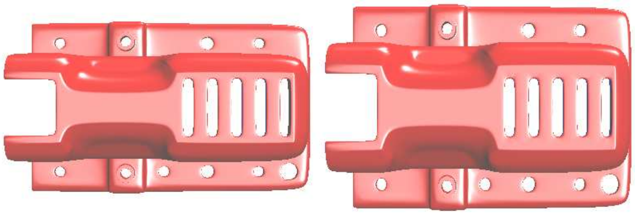

Figure 8: Constrained scaling of a ventilation control component with five features: before scaling (left), after saling (right) with $S_x = 1.0$, $S_y = 1.2$ $S_z = 1.0$.

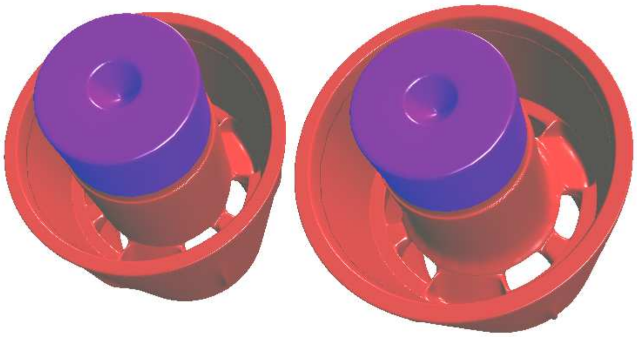

Figure 9: Constrained scaling of a marker cap: before scaling (left), after saling (right) with $S_x = 1.2$, $S_y = 1.2$ $S_z = 1.0$.

[2] Catmull, E., and Clark, J. 1978. Recursively Generated B-spline Surfaces on Arbitrary Topological Meshes. *Computer-Aided Design 10*, 6, 350-355.

[3] Cavendish, J. C. and Marin, S. P., A procedural Feature-Based Approach for Designing Functional Surfaces, In *Topics in Surface Modeling*, Hans Hagen, editor, SIAM 1992, 145-168.

[4] Celniker, G. and Gossard, D., Deformable curve and surface finite elements for free form shape design, *Computer Graphics*, 25(4), 1991, 257-266.

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

[6] Coquillart, S., Extended free form deformation: A Sculpting tool for 3D geometric modeling, *Computer Graphics*, 24(4), 1990, 187-193.

[7] DeRose T, Kass M, Truong T, Subdivision Surfaces in Character Animation, Proceedings of SIGGRAPH, 1998: 85-94.

[8] Doo, D., and Sabin, M. 1978. Behavior of Recursive Division Surfaces near Extraordinary Points. *Computer-Aided Design 10*, 6, 356-360.

[9] Hsu, W. H., Hughes, J. F. and Kaufman, H., Direct manipulation of free-form deformations, *Computer Graphics*, 26(2), 1992, 177-184.

[10] Lamousin, H. J. and Waggenspack, W. N., NURBS-based free-form deformations, *IEEE Transactions on Computer Graphics & Applications*, 14(6), 1994, 59-65.

[11] Mansfield, E. H., The Bending and Stretching of Plates, Machillan, New York, 1964.

[12] Sedbergerg, T. W. and Parry, S. R., Free-form deformation of solid geometric models, *Computer Graphics*, 20(4), 1986, 151-160.

[13] Sederberg TW, Zheng J, Sewell D, Sabin M, Nonuniform recursive subdivision surfaces, Proceedings of SIGGRAPH, 1998:19-24.

[14] Stam J, Exact Evaluation of Catmull-Clark Subdivision Surfaces at Arbitrary Parameter Values. *Proceedings of SIGGRAPH* 1998:395-404.

[15] Terzopoulos, D., Platt, J., Barr, A. and Fleischer, K., Elastically deformable models, *Computer Graphics*, 21(4), 1987, 205-214.

[16] Terzopoulos, D. and Fleischer, K., Deformable models, *Visual Computing*, 4(6), 1988, 306-331.

[17] Terzopoulos, D. and Qin, H., Dynamic NURBS with Geometric Constraints for interactive Sculpting, *ACM Transactions on Graphics*, 13(2), 1994, 103-136.

[18] Welch, W. and Witkin, A., Variational surface modeling, *Computer Graphics*, 26(2), 1992, 157-166.

[19] Zhang, P., Zhang, C. and Cheng, F., Constrainted Shape Scaling of Trimmed NURBS Surfaces, Proc. 1999 ASME Design Theory and Methodology Conference, Las Vegas, Nevada, 1999.

[20] Zhang, C.,Zhang, P. and Cheng, F., Constrainted Scaling of Trimmed NURBS Surfaces Based on Fix-and-Stretch Approach, *Computer-Aided Design*, 33(1), 2001, 103-112.

[21] Ugural, A. C., Ferster, S. K., Advanced Strength and Applied Elasticity, America Elsevier Publishing Co., New York, 1975.

11 Appendix A: Proof of (4)

Let $K = MGM^TV^TVMG^TM^T$. K is a 4×4 matrix. Then E_{sp} can be expressed as

$$\begin{aligned}
E_{sp} &= \tfrac{1}{2}\int\int_D \mathbf{S}^2 dudv \\
&= \tfrac{1}{2}\int\int_D UMGM^TV^TVMG^TM^TU^T dudv \\
&= \tfrac{1}{2}\int\int_D UKU^T du
\end{aligned}$$

$$= \tfrac{1}{2}\int\int_D \begin{bmatrix} K_{00} + K_{01}u + K_{02}u^2 + K_{03}u^3 + \\ K_{10}u + K_{11}u^2 + K_{12}u^3 + K_{13}u^4 + \\ K_{20}u^2 + K_{21}u^3 + K_{22}u^4 + K_{23}u^5 + \\ K_{30}u^3 + K_{31}u^4 + K_{32}u^5 + K_{33}u^6 \end{bmatrix} dudv$$

$$= \tfrac{1}{2}\int_0^1 \begin{bmatrix} K_{00} + K_{01}1/2 + K_{02}1/3 + K_{03}1/4 + \\ K_{10}1/2 + K_{11}1/3 + K_{12}1/4 + K_{13}1/5 + \\ K_{20}1/3 + K_{21}1/4 + K_{22}1/5 + K_{23}1/6 + \\ K_{30}1/4 + K_{31}1/5 + K_{32}1/6 + K_{33}1/7 \end{bmatrix} dv$$

$$= \tfrac{1}{2}\int_0^1 (D_0KY_0^T + D_1KY_1^T + D_2KY_2^T + D_3KY_3^T)dv$$

where D and Y are defined in (5). Since for each $i = 0, 1, 2, 3$, we have

$$\begin{aligned}
&\int_0^1 (D_iKY_i^T)dv \\
&= \int_{[0,1]}(D_iMGM^TV^TVMG^TM^TY_i^T)dv \\
&= D_iMGM^T(\int_{[0,1]}(V^TV)dv)MG^TM^TY_i^T \\
&= D_iMGM^TDMG^TM^TY_i^T
\end{aligned}$$

Hence $E_{sp} = 1/2\sum_{i=0}^3(D_iMGM^TDMG^TM^TY_i^T)$. The other two equations can be proved similarly.

12 Appendix B: Proof of (10)

Note that if N is the set of $(2n+8)$ control vertices around the extra-ordinary point (n is the degree of the extraordinary point, see Figure 3 for an example of $n = 5$), then $\bar{S}S^{L-1}N^T$ is the set of $(2n+17)$ control vertices around the extraordinary point after L levels of Catmull-Clark subdivision.

Our problem now is to retrieve 4×4 vertices from this vector to form a 4×4 matrix G_{Ld} ($d = 1, 2, 3$). First, we select 4 vertices to form the first column of G_{Ld} from $\bar{S}S^{L-1}N^T$ by using pick matrix P_{d0}, i.e. $C_0^T = P_{d0}\bar{S}S^{L-1}N^T$. C_0^T is a column vector. We extend this column vector to a 4×4 matrix by multiplying it with a row vector Y_0(defined in (5)). The result is a 4×4 matrix: $M_0 = (C_0^T, 0, 0, 0)$. Second, we select another 4 vertices to form the second column of G_{Ld} from $\bar{S}S^{L-1}N^T$ by using pick matrix P_{d1}, i.e. $C_1^T = P_{d1}\bar{S}S^{L-1}N^T$. This is again a column vector. We also extend this vector to a 4×4 matrix by multiplying another row vector Y_1 (defined at (5)). The result is a 4×4 matrix: $M_1 = (0, C_1^T, 0, 0)$. Similarly,we can generate $M_2 = (0, 0, C_2^T, 0)$ and $M_3 = (0, 0, 0, C_3^T)$. Adding them together,we have (10). (Note: P_{dj} is a 4 by (2n+17) matrix, with all the entries equal to zero except one "1" each row. The value of d varies from 1 to 3, indicating which regular bicubic B-spline patch after a subdivision is being considered.)

13 Appendix C: Proof of (11)

First note that

$$\begin{aligned}
&f(U, G_{Ld}, G_{L,d}, X, V) \\
&= UMG_{Ld}M^TXMG_{Ld}^TM^TV^T \\
&= UM(\textstyle\sum_{j=0}^3 P_{dj}\bar{S}S^{L-1}N^TY_j)M^TXM\cdot \\
&\quad (\textstyle\sum_{k=0}^3 P_{dk}\bar{S}S^{L-1}N^TY_k)^TM^TV^T \\
&= \textstyle\sum_{0\leq j,k\leq 3} UM(P_{dj}\bar{S}S^{L-1}N^TY_j)M^TXM\cdot \\
&\qquad (P_{dk}\bar{S}S^{L-1}N^TY_k)^TM^TV^T \\
&= \textstyle\sum_{0\leq j,k\leq 3}(Y_jM^TXMY_k^T)UMP_{dj}\bar{S}\cdot \\
&\qquad S^{L-1}N^TN(S^{L-1})^T \cdot \bar{S}^T(P_{dk})^TM^TV^T \\
&= \textstyle\sum_{0\leq j,k\leq 3} h_{jk}(U, X, d)S^{L-1}N^TN(S^{L-1})^T\bar{h}_k(V, d)
\end{aligned}$$

where $h_{jk}(U, X, d)$ and $\bar{h}_k(V, d)$ are defined in (12). And then

$$\begin{aligned}
&\textstyle\sum_{L=1}^\infty f(U, G_{Ld}, G_{Ld}, X, V) \ / \ 4^L \\
&= \textstyle\sum_{L=1}^\infty \sum_{0\leq j,k\leq 3} h_{jk}(U, X, d)\cdot \\
&\qquad S^{L-1}N^TN(S^{L-1})^T\bar{h}_k(V, d) \ / \ 4^L \\
&= \textstyle\sum_{0\leq j,k\leq 3} h_{jk}(U, X, d)\cdot \\
&\qquad (\textstyle\sum_{L=1}^\infty S^{L-1}N^TN(S^{L-1})^T \ / \ 4^L)\bar{h}_k(V, d) \\
&= \textstyle\sum_{0\leq j,k\leq 3} h_{jk}(U, X, d)R_N\bar{h}_k(V, d)
\end{aligned}$$

where R_N is defined in (12). Note that R_N satisfies (13). But then for each fixed d ($d = 1, 2, 3$), we have

$$\begin{aligned}
&\textstyle\sum_{L=1}^\infty E(\mathbf{S}_{Ld}) \ / \ 4^L \\
&= \textstyle\sum_{L=1}^\infty\{\sum_{i=0}^3 f((A_i + C_i + D_i)/2, G_{Ld}, G_{Ld}, D, Y_i) \\
&\qquad + \textstyle\sum_{i=0}^3 f((2A_i + D_i)/2, G_{Ld}, G_{Ld}, A, Y_i) \\
&\qquad + \textstyle\sum_{i=0}^3 f(B_i, G_{Ld}, G_{Ld}, B, Y_i) \\
&\qquad + \textstyle\sum_{i=0}^3 f(D_i/2, G_{Ld}, G_{Ld}, C, Y_i)\} \ / \ 4^L \\
&= \textstyle\sum_{i=0}^3\{\sum_{L=1}^\infty f((A_i + C_i + D_i)/2, G_{Ld}, G_{Ld}, A, Y_i) \ / \ 4^L \\
&\quad + \textstyle\sum_{L=1}^\infty f((2A_i + D_i)/2, G_{Ld}, G_{Ld}, A, Y_i) \ / \ 4^L \\
&\quad + \textstyle\sum_{L=1}^\infty f(B_i, G_{Ld}, G_{Ld}, B, Y_i) \ / \ 4^L \\
&\quad + \textstyle\sum_{L=1}^\infty f(D_i/2, G_{Ld}, G_{Ld}, C, Y_i) \ / \ 4^L \ \} \\
&= \textstyle\sum_{i=0}^3 \sum_{0\leq j,k\leq 3}\{ \\
&\ h_{jk}((A_i + C_i + D_i)/2, D, d)R_N\bar{h}_k(Y_i, d) \\
&\ + h_{jk}((2A_i + D_i)/2, A, d)R_N\bar{h}_k(Y_i, d) \\
&\ + h_{jk}(B_i, B, d)R_N\bar{h}_k(Y_i, d) \\
&\ + h_{jk}(D_i/2, C, d)R_N\bar{h}_k(Y_i, d)\}.
\end{aligned}$$

The energy of the entire extra-ordinary patch is then simply the sum for $d = 1, 2$, and 3 which is (11).