Catmull-Clark Subdivision Surfaces:
an introduction to one-piece representation

Fuhua (Frank) Cheng
University of Kentucky
Lexington, Kentucky

To my parents, Lin-huang and Ko-su.

Table of Contents

List of Tables

List of Figures

Preface

Subdivision surfaces are powerful shape representation scheme for applications in graphical modeling, animation and CAD/CAM because they can model/represent complex shape of arbitrary topology with only one surface. However, subdivision surfaces did not receive much attention from the CAD/CAM industries for almost 20 years because of two reasons. First, it was not known until 1998 that subdivision surfaces can be parametrized [24]. Without a parametric representation, it is essentially impossible for a CAD/CAM system to include subdivision surfaces as a free-form surface modeling tool because of problems with standard operations such as picking, rendering and texture mapping [24]. The second problem is with hardware. Subdivision surfaces are typically generated through recursive meshing. The complexity of the meshing process grows exponentially with respect to the recursive subdivision level. This made generation and rendering of subdivision surfaces on an ordinary workstation impossible in the 80s and early 90s because of lacking enough memory for the recursive mesh refining process.

Things have changed over the past few years. With the parametrization technique of subdivision surfaces becoming available [24] and with the fact that non-uniform B-spline and NURBS surfaces are special cases of subdivision surfaces becoming known [22], we now know that subdivision surfaces cover both *parametric forms* and *discrete forms*. Since parametric forms are good for design and representation and discrete forms are good for machining and tessellation (including FE mesh generation) [34], we finally have a representation scheme good for all graphics and CAD/CAM applications. With powerful PCs that carry almost unlimited memory available everywhere, computation and rendering of subdivision surfaces are no longer a problem either. The era of subdivision surfaces is finally here. Actually, subdivision surfaces have already been used as primitives in several commercial systems such as Alias|Wavefront's *Maya*, Pixar's *Renderman*, Nichiman's *Mirai*, and Microspace' *Lightwave 3D* [7].

The objective of this book is to present general properties of subdivision surfaces and related geometric algorithms and modeling techniques. These algorithms and technologies are important because they are the building blocks of many subdivision surface based modeling operations and, hence, are needed by any of the CAD/CAM systems that intends to include subdivision surfaces as the next generation surface representation for CAD/CAM applications.

The arrangement of the book is as follows. In the first chapter, we will ... IN the second chapter, ...

Fuhua (Frank) Cheng
Lexington, Kentucky
December 15, 2008

Chapter 1

Introduction

1.1 Motivation

Imagine we have a *one-piece representation scheme*, i.e., we can represent any object with only one surface, no matter how complicated the object's topology or shape (see Figure 1.3(d) for an example). What does this mean? This means *modeling*, *data storage*, *rendering*, and *animation* of objects will all become easier and more efficient. For example, to build a representation of a complicated object, it is no longer necessary to painfully decompose the object into simpler components. We can go directly for a representation of the object instead of building representations of the components first and then combining these representations through union operation or a *constructive solid geometry* (CSG) structure to get a representation of the object. Hence, the number of parts in the final representation is always the minimum: one.

Traditional surface representation schemes, such as *B-spline surfaces* or *non-uniform rational B-spline* (NURBS) *surfaces*, can not achieve the goal of one-piece representation. This is because the topology of the parameter space of such a surface is rectangular. It is not even possible to use such a surface to represent a closed object. Actually, any surface representation scheme whose parameter space has a fixed topological sturcture can not represent a closed object with only one surface. To make one-piece representation possible, one must have a surface scheme whose parameter space can have an arbitrary topological structure. This was the background the concept of *Catmull-Clark subdivision surface* was developed.

Catmull and Clark noticed that the subdivision process of a uniform bicubic B-spline surface can be generalized [1]. The generalized subdivision process works for control mesh of any topology. By iteratively repeating this subdivision scheme, one can get a *limit surface* of any shape. The topological structure of the limit surface's parameter space is the same as the topology of the control mesh. One thus gets a surface scheme whose parameter space can have an arbotrary topology. In the following, we will first review subdivision scheme of uniform bicubic B-spline surfaces and then show generalization of this subivision scheme to get Catmull-Clark subidivison scheme.

1.2 Subdivision of Bicubic B-Spline Patch

Given a set of sixteen control points $\mathbf{P}_{i,j}$, $1 \leq i, j \leq 4$, a bicubic B-spline patch is define by

$$\mathbf{S}(u, v) \;=\; UCGC^tV^t, \qquad 0 \leq u, v \leq 1,$$

where

$$C \;=\; \frac{1}{6}\begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}$$

is the B-spline coefficient matrix for cubics,

$$G \;=\; \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \mathbf{P}_{13} & \mathbf{P}_{14} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \mathbf{P}_{23} & \mathbf{P}_{24} \\ \mathbf{P}_{31} & \mathbf{P}_{32} & \mathbf{P}_{33} & \mathbf{P}_{34} \\ \mathbf{P}_{41} & \mathbf{P}_{42} & \mathbf{P}_{43} & \mathbf{P}_{44} \end{bmatrix}$$

is the control point matrix, and

$$U \;=\; [1,\; u,\; u^2,\; u^3] \qquad \text{and} \qquad V \;=\; [1,\; v,\; v^2,\; v^3]$$

are the primitive basis vectors.

If a mid-point subdivision is performed on the above patch, one gets four subpatches, corresponding to the four quadrants of the unit square, respectively. Consider the subpatch defined on the quadrant $[0, \frac{1}{2}] \times [0, \frac{1}{2}]$:

$$\mathbf{S}(u/2, v/2) \;=\; UDCGC^tD^tV^t, \qquad 0 \leq u, v \leq 1, \tag{1.1}$$

where

$$D \;=\; \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & 1/8 \end{bmatrix}.$$

This subpatch, as a uniform bicubic B-spline patch by itself, can also be expressed as

$$\mathbf{S}(u, v) \;=\; UCG_1C^tV^t, \qquad 0 \leq u, v \leq 1, \tag{1.2}$$

with G_1 being its control point matrix

$$G_1 = \begin{bmatrix} \mathbf{P}_{11}^1 & \mathbf{P}_{12}^1 & \mathbf{P}_{13}^1 & \mathbf{P}_{14}^1 \\ \mathbf{P}_{21}^1 & \mathbf{P}_{22}^1 & \mathbf{P}_{23}^1 & \mathbf{P}_{24}^1 \\ \mathbf{P}_{31}^1 & \mathbf{P}_{32}^1 & \mathbf{P}_{33}^1 & \mathbf{P}_{34}^1 \\ \mathbf{P}_{41}^1 & \mathbf{P}_{42}^1 & \mathbf{P}_{43}^1 & \mathbf{P}_{44}^1 \end{bmatrix}$$

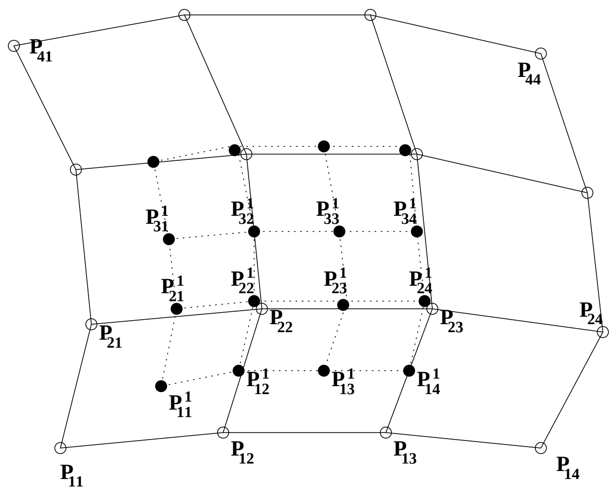


Figure 1.1: Relationship between old control points (circles) and new control points (solid circles) for a bicubic B-Spline patch after a midpoint subdivision.

(see Figure 1.1 for the relationship between $\mathbf{P}_{ij}$ and $\mathbf{P}_{ij}^1$). Eq. (1.1) and eq. (1.2) represent the same subpatch. They equal to each other for arbitrary u and v if and only if

$$DCGC^tD^t = CG_1C^t .$$

Hence, control points of the subpatch are related to the original control points by the expression

$$G_1 = [C^{-1}DC]G[C^tD^tC^{-t}] = HGH^t \tag{1.3}$$

where

$$C^{-1} = \frac{1}{3}\begin{bmatrix} 3 & -3 & 2 & 0 \\ 3 & 0 & -1 & 0 \\ 3 & 3 & 2 & 0 \\ 3 & 6 & 11 & 18 \end{bmatrix} .$$

By carrying out the matrix multiplications, we have

$$H = \frac{1}{8}\begin{bmatrix} 4 & 4 & 0 & 0 \\ 1 & 6 & 1 & 0 \\ 0 & 4 & 4 & 0 \\ 0 & 1 & 6 & 1 \end{bmatrix} .$$

H is called the *splitting matrix* [1]. The new control points are classifed into three categories by Catmull and Clark: *face points*, *edge points*, and *vertex points*, according to their locations with respect to the original control mesh. A new control point is called a *face point* if it is located at the center of an original mesh face, such as $\mathbf{P}^1_{11}$ or $\mathbf{P}^1_{13}$. A new control point is called an *edge point* if it is located near the midpoint of an original mesh edge, such as $\mathbf{P}^1_{12}$ or $\mathbf{P}^1_{23}$. A new control point is called a *vertex point* if it is located near a vertex of the original mesh, such as $\mathbf{P}^1_{22}$ or $\mathbf{P}^1_{24}$. There are four face points, four vertex points and eight edge points in the control mesh of the subpatch. There is an edge between a face point and each of its adjacent edge points, and an edge between a vertex point and each of its adjacent edge points as well.

Carrying out the algebra of (1.3) gives us expressions of these points. A face point is the average of the vertices of the face that it associates with [1]. For example, $\mathbf{P}^1_{11}$ is given by

$$\mathbf{P}^1_{11} = \frac{\mathbf{P}_{11} + \mathbf{P}_{12} + \mathbf{P}_{21} + \mathbf{P}_{22}}{4} . \tag{1.4}$$

An edge point is the average of the midpoint of the edge that it associates with and the average of the new face points of the faces sharing the edge [1]. For example, $\mathbf{P}^1_{12}$ is given by

$$\mathbf{P}^1_{12} = \frac{\frac{\mathbf{P}_{12}+\mathbf{P}_{22}}{2} + \frac{\mathbf{P}^1_{11}+\mathbf{P}^1_{13}}{2}}{2} \tag{1.5}$$

where $\mathbf{P}^1_{11}$ and $\mathbf{P}^1_{13}$ are face points of the faces that share the edge $\mathbf{P}_{12}\mathbf{P}_{22}$. A vertex point is a linear combination of adjacent face points, midpoints of adjacent edges and the associated vertex [1]. For examples, $\mathbf{P}^1_{22}$ is given by

$$\mathbf{P}^1_{22} = \frac{\mathbf{F}}{4} + \frac{2\mathbf{E}}{4} + \frac{\mathbf{P}_{22}}{4} \tag{1.6}$$

where $\mathbf{F}$ is the average of adjacent face points

$$\mathbf{F} = \frac{\mathbf{P}^1_{11} + \mathbf{P}^1_{13} + \mathbf{P}^1_{31} + \mathbf{P}^1_{33}}{4}$$

and $\mathbf{E}$ is the average of midpoints of adjacent edges

$$\mathbf{E} = \left[\frac{\mathbf{P}_{12} + \mathbf{P}_{22}}{2} + \frac{\mathbf{P}_{21} + \mathbf{P}_{22}}{2} + \frac{\mathbf{P}_{23} + \mathbf{P}_{22}}{2} + \frac{\mathbf{P}_{32} + \mathbf{P}_{22}}{2}\right] /4 .$$

Each of the remaining points in G_1 satisfies an expression similar to one of (1.4), (1.5) or (1.6).

Note that the subdivision process does not generate edge points for boundary edges, and no vertex points for boundary vertices either.

1.3 Catmull-Clark Subdivision Surfaces

Once we know that new control points should be classified into three categories and there is a specific expression for each category, then generalizing the above subdivision process to control meshes of arbitrary topologies becomes straightforward. Given a control mesh with arbitrary topology, new vertices are generated according to the following rules:

- New face points - a face point is generated for each face of the given mesh; the new face point is the average of the vertices of the face.
- New edge points - an edge point is generated for each interior edge of the given mesh; the new edge point is the average of the midpoint of the edge with the average of the two new face points of the faces sharing the edge.
- New vertex points - a vertex point is generated for each interior vertex of the given mesh; the new vertex point is a linear combination of adjacent face points, midpoints of adjacent edges and the vertex, as follows:

$$\bar{\mathbf{V}} = \frac{\mathbf{F}}{n} + \frac{2\mathbf{E}}{n} + \frac{(n-3)\mathbf{V}}{n} \tag{1.7}$$

  where $\bar{\mathbf{V}}$ is the new vertex point, $\mathbf{V}$ is the old vertex, n is the number of adjacent edges of $\mathbf{V}$, $\mathbf{F}$ is the average of the n adjacent new face points, and $\mathbf{E}$ is the average of the midpoints of the n adjacent edges of $\mathbf{V}$.

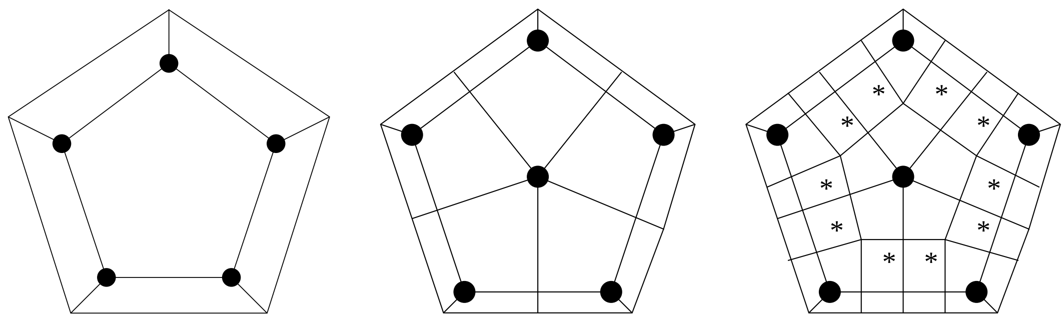
After all the new vertices have been generated, new edges are formed as follows:

- Each new face point is connected to each of its adjacent new edge points.
- Each new vertex point is connected to each of its adjacent new edge points.

New faces are then defined as those enclosed by new edges. This subdivision scheme was developed by Catmull and Clark [1] and is called the *Catmull-Clark subdivision scheme* or *Catmull-Clark subdivision.*

The results of applying the Catmull-Clark subdivision one time and two times to a simple non-rectangular topology (Figure 1.2(a)) are shown in Figures 1.2(b) and 1.2(c), respectively.

By iteratively repeating the Catmull-Clark subdivision process on a given control mesh, one gets an infinite sequence of refined control meshes. These control meshes converges to a limit surface. That surface is called a Catmull-Clark subdivision surface. An example of such a surface is shown in Figure 1.3(d). The given control mesh is shown in Figure 1.3(a). Control meshes after one subdivision and two subdivisions are shown in Figures 1.3(b) and 1.3(c), respectively.



(a) Given topology (b) After one subdivision (c) After two subdivisions

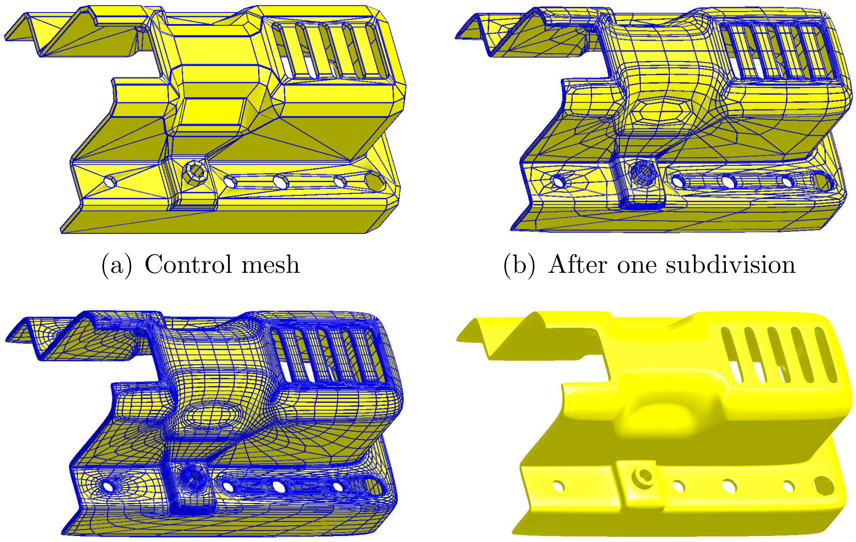
Figure 1.2: Results of applying Catmull-Clark subdivision to a simple non-rectangular topology.

1.4 General Properties

Catmull-Clark subdivision process has several important properties. First, note that for an open mesh, one does not get an edge point for a boundary edge, nor a vertex point for a boundary vertex. Hence, one does not get a limit surface patch for a boundary face. One gets a limit surface patch only for each interior face. Boundary faces merely assist in defining the slope and curvature of the limit surface patches of adjacent interior faces.

Second, since each new face is formed by a face point, a vertex point and two edge points, each new face is always four-sided. But a similar situation does not hold for new vertices. Each new edge point has four adjacent edges, but the number of adjacent edges of a new vertex point depends on the number of adjacent edges of the old vertex, and the number of adjacent edges of a new face point depends on the number of edges of the old face. Following Catmull and Clark's terminology [1], we call an interior mesh vertex an *extra-ordinary point* if the number of adjacent edges is not four. The number of adjacent edges of an extra-ordinary point is also called the *valence* of the point. An example of an extra-ordinary point $\mathbf{V}$ of valence six is shown in Figure 1.4. If the number of extra-ordinary points of a given mesh is m and the number of non-four-sided interior faces is n, then after one Catmull-Clark subdivision, the number of new extra-ordinary points will be $m + n$. The number of extra-ordinary points remains constant after that point no matter how many times of Catmull-Clark subdivision are performed subsequently. This follows from the fact that after one Catmull-Clark subdivision all faces are four-sided, hence all new vertices created subsequently will have four adjacent edges except those correspond to the old extra-ordinary points. All of these properties can also be visually verified with the results shown in Figures 1.2(b) and 1.2(c).

Now look at Figure 1.2(c). Ten faces are marked with a '*'. Each of these faces has associated with it a set of 16 points that lie on a rectangular grid, as with the standard bicubic B-spline patches. Since the Catmull-Clark subdivision process is a generalization of the subdivision of uniform bicubic B-spline surface, we will get a standard bicubic B-spline patch for each of these faces. Therefore, ten portions of the



(a) Control mesh (b) After one subdivision

(c) After two subdivisions (d) Limit surface

Figure 1.3: A ventilation control component represented by a single Catmull-Clark subdivision surface.

final limit surface are defined. If we perform more iterations of the Catmull-Clark subdivision more portions of the final limit surface will be defined. Like standard bicubic B-spline surfaces, those portions of the limit surface have continuous first and second derivatives. Eventually, every point of the limit surface is a bicubic B-spline surface point except vertices corresponding to the extra-ordinary points. Hence, the limit surface is everywhere C^2-continuous except at vertices corresponding to the extra-ordinary points.

If the adjacent vertices of an extra-ordinary point $\mathbf{V}$ are labeled as in Figure 1.4 then, according to eq. (1.7), new location of the extra-ordinary point is

$$\bar{\mathbf{V}} \;=\; \left(1 - \frac{7}{4n}\right)\mathbf{V} + \frac{3}{2n}\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{E}_i\right) + \frac{1}{4n}\left(\frac{1}{n}\sum_{i+1}^{n}\mathbf{F}_i\right) \tag{1.8}$$

where n is the valence of $\mathbf{V}$. Since the sum of the coefficients on the right side equals one, this leads to a more general definition of the new extra-ordinaty point:

$$\bar{\mathbf{V}} \;=\; \alpha_n\mathbf{V} + \beta_n\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{E}_i\right) + \gamma_n\left(\frac{1}{n}\sum_{i+1}^{n}\mathbf{F}_i\right) \tag{1.9}$$

where α_n, β_n and γ_n are non-negative numbers whose sum equals one [5].

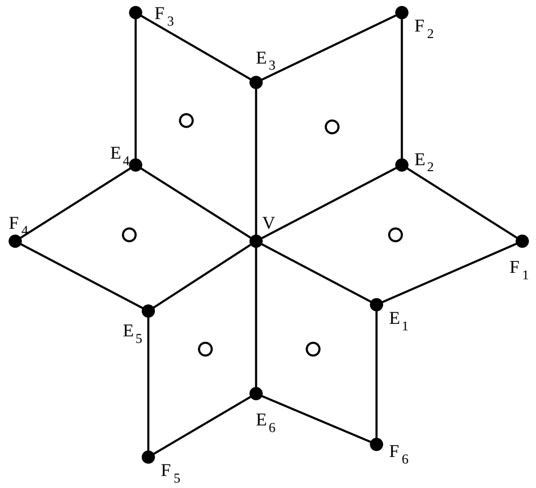


Figure 1.4: Adjacent vertices (solid circles) of an extra-ordinary point **V** and their labels. Hollow circles represent new face points generated for the adjacent faces of **V**.

Chapter 2

Background

2.1 A Brief History

The concept of generating a surface through mesh refinement has its root in a curve generation technique developed by Chaikin [?]. In his approach, a curve is generated by recursively cutting off corners of a given polygon. Each recursive cutting cycle generates two new points on each leg of the polygon. If there are $n+1$ vertices $\mathbf{P}_i^j$, $i = 0, 1, ..., n$, after the jth recursive cutting cycle, then the two new points generated on the polygon leg $\mathbf{P}_i^j\mathbf{P}_{i+1}^j$ are defined as follows:

$$\mathbf{P}_{2i}^{j+1} = \frac{3}{4}\mathbf{P}_i^j + \frac{1}{4}\mathbf{P}_{i+1}^j; \qquad \mathbf{P}_{2i+1}^{j+1} = \frac{1}{4}\mathbf{P}_i^j + \frac{3}{4}\mathbf{P}_{i+1}^j.$$

This process generates a uniform, quadratic B-spline curve as this corner-cutting process is nothing but the quadratic B-spine subdivision process. The concept of B-spline subdivision is actually a generalization of Chaikin's algorithm (see [?] for the corresponding refinement equation).

Following Chaikin's work, a variety of subdivision schemes for curves and surfaces have been proposed during the past two decades. For instance, a 4-point subdivision scheme proposed by Dyn, Levin and Gregory [?] can generate a subdivision curve to interpolate given data points. New points for each leg of the refined control polygon are defined by

$$\mathbf{P}_{2i}^{j+1} = \mathbf{P}_i^j; \qquad \mathbf{P}_{2i+1}^{j+1} = \frac{8+\omega}{16}(\mathbf{P}_i^j + \mathbf{P}_{i+1}^j) - \frac{\omega}{16}(\mathbf{P}_{i-1}^j + \mathbf{P}_{i+2}^j)$$

where $0 < \omega < 2(\sqrt{5}-1)$, to ensure convergence of the refined mesh. The standard value is $\omega = 1$ which has an order three precision.

Refining (subdivision) schemes for subdivision surfaces can be classified into two categories: (1) *approximating techniques*, and (2) *interpolating techniques*. Two typical subdivision schemes in the first category are Doo and Sabin's scheme [?] and Catmull and Clark's scheme [1]. Doo and Sabin's scheme generates a surface by recursively cutting off corners and edges of a given **rectangular mesh** as follows:

1. For every vertex V_i of the current mesh P, a new vertex V_i', called an *image*, is generated on each face adjacent to V_i.

2. For each face F_i of P, a new face, called an *F-face*, is constructed by connecting the *image* vertices V_i's generated in Step 1.

3. For each edge E_i common to two faces F_i and F_i', a new 4-sided face, called an *E-face*, is constructed by connecting the *images* of the end vertices of E_i on the faces F_i and F_i'.

4. For each vertex V_i, where n faces meet, a new face, called a *V-face*, is constructed by connecting the images of V_i on the faces meeting at V_i.

This subdivision scheme generates a uniform biquadratic B-spline surface. Catmull and Clark's scheme [1] is similar to the Doo-Sabin scheme, but is based on tensor product bicubic B-spline. The surface generated by this scheme is C^2 continuous everywhere except at some extraordinary points where it is C^1 continuous. Catmull and Clark's scheme can work on meshes of arbitrary topology. Loop [3] has presented a similar subdivision scheme based on generalization of quartic three-direction Box-splines for triangular meshes. Peters and Reif [?] and Habib and Warren [?] independently introduced schemes that generalize quadratic 4-direction Box Splines on irregualr meshes. Subdivision schemes that can generate surfaces with sharp features [14] or fractionally sharp features [?] have also been proposed. Recently, it is even possible to generate features such as cusps, creases, and darts through the introduction of non-uniform subdivision surfaces [22]. A new subdivision scheme that can produce triangular meshes with small number of vertices is proposed by Kobbelt [?].

The first interpolating scheme for subdivision surfaces was presented by Dyn, Levin and Gregory [?]. The technique, called a *butterfly scheme*, requires a topologically regular setting of the initial (control) mesh to produce a C^1 limit surface. Zorin *et al* [?] and Kobbelt [?] have both developed improved interpolating schemes recently. Kobbelt's scheme is a simple extension of the 4-point interpolating subdivision [?]. Zorin *et al*'s scheme retains the simplicity of the butterfly scheme and results in much smoother surfaces even from irregular initial meshes. These interpolating subdivision schemes also find applications in wavelets on manifolds, multiresolution decomposition of polyhedral surfaces, and multiresolution editing.

Some of the mathematical properties of subdivision surfaces have been studied before. For instance, Doo and Sabin have studied the smoothness behavior of their subdivision surfaces through Fourier transformations and eigen-value analysis of the subdivision matrix [15]. Ball and Storry [5][6] and Reif [18] extended Doo and Sabin' work by deriving various necessary and sufficient smoothness conditions for different subdivision schemes. Specific subdivision schemes have also been analyzed by several other people [?][?][?][?][?]. Nevertheless, most of the geometric algorithms and modeling technologies required in subdivision surface based modeling operations are not well studied yet. Four of these areas are especially critical to the design community.

2.2 Surface Tessellation

Given a surface, a major concern in both finite element analysis (FEM) and surface rendering is the generation of an approximating mesh of the given surface (within a given error tolerance) with as few nodes as possible. The approximating mesh is used to analyze the physical performance of the surface or in the scan conversion process of the surface. Smaller number of nodes in the approximating mesh is preferred because it makes the analysis process and the rendering process both more efficient. This process of generating an approximating mesh for a given surface, called *surface tessellation*, has been extensively studied for parametric surfaces [?][?]. It has not been well studied for subdivision surfaces yet.

To generate a good approximating mesh for a subdivision surface, one needs to be able to (1) estimate the error between the control mesh (or, an approximating mesh) and the limit (subdivision) surface, (2) determine the level (depth) of recursive subdivision needed to reach a required precision, and (3) adaptively tessellate the faces of the initial control mesh so that an approximating mesh that is just good enough for the specified precision and yet satisfying the crack-free requirement can be constructed. Existing subdivision schemes can not be used directly in the tessellation process because they lack the so-called *adaptive capability*; they would subdivide all the faces of a mesh even if only one of them does not satisfy the precision requirement and, consequently, would generate approximating meshes with too many nodes (see Figure 2.1(c) for excessively generated nodes in flat regions of a rocker arm with only two levels of subdivision).

The first adaptive scheme for subdivision surfaces is proposed by Kobbelt [?] for Catmull-Clark subdivision surfaces. The method is performed on a trial-and-error basis and only works for the so-called *balanced nets* which, in addition, have to satisfy some other constraints such as *even critical edges*. A few more general schemes appeared recently for *interpolatory* $\sqrt{3}$*-subdivision surfaces* [?], $\sqrt{3}$*-subdivision surfaces* [?], and *modified butterfly subdivision surfaces* [?]. But they work for triangular control meshes only. Another problem with all the above adaptive schemes is that none of them use the error criterion most commonly used in mechanical part design, i.e., the error between the approximating mesh and the limit surface.

We have worked in all these three areas: *error estimation* [?], *subdivision level (depth) computation* [?], and *adaptive mesh generation* [?][?]. However, the techniques developed for B-spline and NURBS surfaces can not be used for subdivision surface directly because the parameter space of a subdivision surface in general is not rectangular or triangular; it can be of any shape. New techniques have to be developed for each of these areas.

(a) Control mesh (b) Limit surface (c) Approximating mesh

Figure 2.1: An example of uniform subdivision of the control mesh of rocker arm.

2.3 Automatic Fairing

Automatic fairing refers to the process of detecting and removing local irregularities of a surface automatically. Curvature plots have been frequently used to analyze the quality of a surface. Commonly used curvature measures include *Gaussian*, *mean*, and *principal curvatures* as well as *normal curvatures* along given directions. *Isophotes* [?], *reflection lines* [?, ?] and, more recently, *highlight lines* [?, ?] have also been used in assessing the quality of a surface. These techniques prove to be more effective and are becoming more popular recently, especially in automotive body surface design, because they are more intuitive to understand and easier to compute. The smoothness of a surface is measured using indicators such as parametric or geometric continuity.

Several papers analyzing parametric and geometric continuity of subdivision surfaces have been published (see, e.g., [?, 5, 18]). They all concentrate on analyzing the subdivision scheme, instead of the layout of the control points, of the subdivision surface. The latter is actually more important because a well-designed control point net is likely to bring out a higher order of continuity.

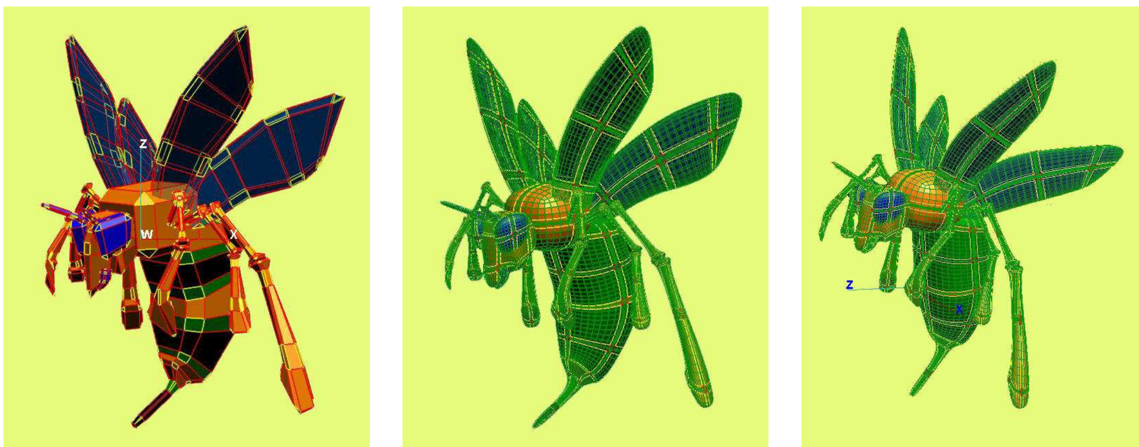
Using diffusion and curvature flow, Desbrun, Meyer, Schröder and Barr [?] have presented a method for removing undesirable noises and uneven edges from irregularly triangulated data. A problem with this approach is that while removing vertices and edges, one might also remove important data “underneath” the “noises”. For instance, the “noises” could be introduced by numerical error in the input phase but are within the tolerance level, therefore, the information carried underneath the noises should still be acceptable. A better approach would be to perturb the points or edges to achieve the goal of shape fairing, instead of removing points or edges. However, no paper has been published on constructing a new limit (subdivision) surface with higher parametric or geometric smoothness but with minimum distance from the original limit (subdivision) surface.

Fairing techniques based on modifying reflection or highlight lines have also been proposed [?][?][?][?]. They all heavily rely on the designers to visually identify the irregular regions and to fix them manually by correcting the control points of the surface. This is an experience-based, trial-and-error, and time-consuming process. The complexity of the problem for subdivision surfaces would make the situation even worse, likely to exceed what the human being can cope with, because the topology of a subdivision space is usually much more complicated than that of a parametric surface. One needs the capability of automatic detection and correction of local irregularities for subdivision surfaces. One also needs an approach different from the highlight line model because identifying surface normals that intersect the light source for a subdivision surface is too costly a process for an interactive design environment. A newly developed surface smoothness evaluation model by us, called the *shadowgraph line model*, will be considered here. This model has an analytical representation for each shadowgraph line. Therefore, there is no cost in getting a representation for a shadowgraph line at all.

2.4 Shape design

The design of a subdivision surface involves (1) the design, and (2) fine tuning of the control mesh. The only known technique in the first area is the work of Levin [52] which uses a combined subdivision scheme to construct a subdivision surface to interpolate a given net of curves. This is an important work because it points out a better approach for subdivision surface shape design (a parallel work for parametric surfaces can be found in [?]). However, properties of Levin's surface are not known yet and it is not a good idea to include too many new subdivision schemes in a modeling system. It is preferred to have similar interpolation techniques using existing subdivision schemes so that the trimming process can be handled with efficiency (see next section for the justification).

The only known technique in the second area is the work by Miura, Wang and Cheng [?] which provides the user with a tangent manipulation technique to fine tune the shape of a subdivision surface. An example is shown in Figure ?? where a set of Doo-Sabin surfaces are deformed using the tangent vector blending technique and the resulting Doo-Sabin surfaces in non-uniform form are shown in (b). For comparison purpose, the original Doo-Sabin surfaces in non-uniform form are shown in (c). The advantage of this approach is that through the manipulation of the tangent vectors, one can directly manipulate the curvature and variation of curvature of the surface. The disadvantage is that it could be too laborious for subdivision surfaces with complex topology. Note that while it is necessary to provide the user with the capability of direct control point or tangent vector manipulation, it is essential that the user can manipulate the shape of the surface directly (such as dragging a point of the surface to a new location), leaving the time-consuming job of finding the new locations of the control points to the system, so that the fine tuning process of shape design can be carried out more efficiently.



(a) Corresponding control mesh (b) fine tuned Doo-Sabin surfaces in non-uniform form (c) original Doo-Sabin surfaces in non-uniform form

Figure 2.2: An example of fine tuning of the control mesh.

2.5 Surface Trimming

NURBS surface intersection, even up to today, is still considered the most difficult problem and one of the weaker links in even high end commercial CAD systems [17][**?**]. The subdivision surface intersection problem would be even more difficult because of the irregularity of the topology of a subdivision surface. The main difficulty is the development of a reliable and efficient computation (marching) process.

An algorithm for calculating the trimming curves of two Loop's subdivision surfaces is proposed by Litke, Levin and Schröeder [17] recently. The algorithm can guarantee exact interpolation of the trimming curves. This is achieved by introducing a new type of surfaces, called *combined surfaces*, to approximate the trimmed surfaces. A problem with this approach is that the inclusion of a new surface type in a CAD system with m surface representation schemes requires m more functions to implement the surface intersection problem. It is preferred to keep the number of surface representation schemes low in a CAD system.

Biermann, Kristjansson and Zorin [8] have presented a new method to approximate Boolean operations on free-form solids. The result of a Boolean operation is approximated by a multiresolution surface. The work pays more attention to efficiency and robustness than to precision and, consequently, is more suitable for applications where precision modeling is not required, such as animation. For applications in CAD/CAM, however, one needs to pay more attention to precision and robustness than to efficiency.

Bibliography

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

[2] D. Doo and M. A. Sabin. Behaviour of recursive subdivision surfaces near extraordinary points. *Computer-Aided Design*, 10:356–360, 1978.

[3] C.T. Loop, Smooth subdivision surfaces based on triangles, *M.S. Thesis*, Department of Mathematics, University of Utah, Salt Lake City, 1987.

[4] KOBBELT, L. $\sqrt{3}$ Subdivision, *Proceedings of SIGGRAPH 2000*, pp. 103–112, July, 2000.

[5] Ball AA, Storry DJT, Conditions for tangent plane continuity over recursively generated B-spline surfaces, *ACM Transactions on Graphics*, 1988, 7(2): 83-102.

[6] Ball AA, Storry DJT, An investigation of curvature variations over recursively generated B-spline surfaces, *ACM Transactions on Graphics*, 1990, 9(4):424-437.

[7] Biermann H, Levin A, Zorin D, Piecewise smooth subdivision surfaces with normal control, Proceedings of SIGGRAPH 2000: 113-120.

[8] Biermann H, Kristjansson D, Zorin D, Approximate Boolean operations on free-form solids, *Proceedings of SIGGRAPH*, 2001: 185-194.

[9] Boier-Martin I, Zorin D, Differentiable Parameterization of Catmull-Clark Subdivision Surfaces, *Eurographics Symposium on Geometry Processing* (2004).

[10] Boullion T, Odell P, *Generalized Inverse Matrices*, New York, Wiley, 1971.

[11] Chen G, Cheng F, Matrix based Subdivision Depth Computation Method for Extra-Ordinary Catmull-Clark Subdivision Surface Patches, *Lecture Notes in Computer Science*, Vol. 4077, Springer, 2006, 545-552.

[12] Cheng F, Yong J, Subdivision Depth Computation for Catmull-Clark Subdivision Surfaces, *Computer Aided Design & Applications 3*, 1-4, 2006.

[13] Cheng F, Chen G, Yong J, Subdivision Depth Computation for Extra-Ordinary Catmull-Clark Subdivision Surface Patches, *Lecture Notes in Computer Science*, Vol. 4035, Springer, 2006, 545-552.

[14] DeRose T, Kass M, Truong T, Subdivision Surfaces in Character Animation, *Proc. of SIGGRAPH*, 1998.

[15] Doo D, Sabin M, Behavior of recursive division surfaces near extraordinary points, *Computer-Aided Design*, 1978, 10(6):356-360.

[16] Halstead M, Kass M, DeRose T, Efficient, fair interpolation using Catmull-Clark surfaces, *Proceedings of SIGGRAPH*, 1993:35-44.

[17] Litke N, Levin A, Schröder P, Trimming for Subdivision Surfaces, *Computer Aided Geometric Design* 2001, 18(5):463-481.

[18] Reif U, A unified approach to subdivision algorithms near extraordinary vertices, *Computer Aided Geometric Design*, 1995, 12(2): 153-174.

[19] Jörg Peters, Ulrich Reif , Analysis of Algorithms Generalizing B-Spline Subdivision, *SIAM Journal of Numerical Analysis*, Vol. 35, No. 2, pp. 728-748, 1998.

[20] Lutterkort D, Peters J, Tight linear envelopes for splines, *Numerische Mathematik 89*, 4, 735-748, 2001.

[21] Peters J, Patching Catmull-Clark Meshes, *Proceedings of SIGGRAPH 2000*, 255-258.

[22] Sederberg TW, Zheng J, Sewell D, Sabin M, Non-uniform recursive subdivision surfaces, *Proceedings of SIGGRAPH*, 1998:19-24.

[23] Smith J, Epps D, Sequin C, Exact Evaluation of Piecewise Smooth Catmull-Clark Surfaces Using Jordan Blocks, http://www.cs.berkeley.edu/~jordans/pubs/ June, 2004.

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

[25] Stam J, Evaluation of Loop Subdivision Surfaces, *SIGGRAPH'99 Course Notes*, 1999.

[26] Peter Schröder, Denis Zorin, Subdivision for Modeling and Animation, *SIGGRAPH'98 Course Notes*, 1998.

[27] Zorin D, Kristjansson D, Evaluation of Piecewise Smooth Subdivision Surfaces, *The Visual Computer*, 2002, 18(5/6):299-315.

[28] Zorin, D., Schröder, P., and Sweldens, W. Interactive Multiresolution Mesh Editing. In *Proceedings of SIGGRAPH 1997*, 259-268.

[29] Joe Warren, Henrik Weimer, Subdivision Methods for Geometric Design: A Constructive Approach. ISBN: 1-55860-446-4, Academic Press, 2002.

[30] Kobbelt, L., Interpolatory subdivision on open quadrilateral nets with arbitrary topology, *Computer Graphics Forum*, Eurographics, V.15, 1996.

[31] D. Zorin, P. Schröder, W. Sweldens, Interpolating Subdivision for meshes with arbitrary topology, *ACM SIGGRAPH*, 1996:189-192.

[32] Dyn,N., Levin, D., and Gregory, J. A., A butterfly subdivision scheme for surface interpolation with tension control, *ACM Transactions on Graphics*, 9, 2 (1990) 160169.

[33] Nasri, A. H., Surface interpolation on irregular networks with normal conditions, *Computer Aided Geometric Design*, 8 (1991), 8996.

[34] Austin SP, Jerard RB, Drysdale RL, Comparison of discretization algorithms for NURBS surfaces with application to numerically controlled machining, *Computer Aided Design* 1997, 29(1): 71-83.

[35] Fuhua (Frank) Cheng, Gang Chen and Junhai Yong, Subdivision Depth Computation for Catmull-Clark Subdivision Surfaces, submitted. www.cs.uky.edu/∼cheng/PUBL/sub_depth.pdf.

[36] Garland M, Heckber P, Surface simplification using quadric error metrics, *Proceedings of SIGGRAPH* 1997:209-216.

[37] Settgast V, Müller K, Fünfzig C, et.al., Adaptive Tesselation of Subdivision Surfaces, In *Computers & Graphics*, 2004, pp:73-78.

[38] Amresh A, Farin G, Razdan A, Adaptive Subdivision Schemes for Triangular Meshes, In *Hierarchical and Geometric Methods in Scientific Visualization*, Springer-Verlag, 2002 pp:319-327.

[39] Wu X, Peters J, An Accurate Error Measure for Adaptive Subdivision Surfaces, In *Shape Modeling International*, 2005

[40] M. Böo, M. Amor, M. Doggett, et.al., Hardware Support for Adaptive Subdivision Surface Rendering, In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware* 2001, pp:33-40.

[41] Müller K, Techmann T, Fellner D, Adaptive Ray Tracing of Subdivision Surfaces *Computer Graphics Forum* Vol 22, Issue 3 (Sept 2003).

[42] Smith J, Séquin C, Vertex-Centered Adaptive Subdivision, www.cs.berkeley.edu/∼jordans/pubs/vertexcentered.pdf.

[43] Isenberg T, Hartmann K, König H, Interest Value Driven Adaptive Subdivision, In *Simulation und Visualisierung*, March 6-7, 2003, Magdeburg, Germany.

[44] Sovakar A, Kobbelt L, API Design for adaptive subdivision schemes. 67-72, *Computers & Graphics*, Vol. 28, No. 1, Feb. 2004.

[45] Rose D, Kada M, Ertl T, On-the-Fly Adaptive Subdivision Terrain. In *Proceedings of the Vision Modeling and Visualization Conference*, Stuttgart, Germany, pp: 87-92, Nov. 2001.

[46] Wu X, Peters J, Interference detection for subdivision surfaces, *Computer Graphics Forum, Eurographics* 23(3):577-585, 2004.

[47] Yong J, Cheng F, Adaptive Subdivision of Catmull-Clark Subdivision Surfaces, *Computer-Aided Design & Applications* 2(1-4):253-261, 2005.

[48] Barsky B A, End conditions and boundary conditions for uniform B-spline curve and surface representation, *Computers in Industry*, 1982, 3(1/2):17-29.

[49] Halstead M, Kass M, DeRose T, Efficient, fair interpolation using Catmull-Clark surfaces, *ACM SIGGRAPH*, 1993:35-44.

[50] Kallay, M., Ravani, B., Optimal twist vectors as a tool for interpolating a network of curves with a minimum energy surface, *Computer Aided Geometric Design*, 7,6 (1990):465-473.

[51] Kersey S N, Smoothing and near-interpolatory subdivision surfaces, www.cs.georgiasouthern.edu/faculty/kersey_s/ private/res/siam2003.pdf

[52] Levin A, Interpolating nets of curves by smooth subdivision surfaces, *ACM SIGGRAPH*, 1999, 57-64.

[53] Litke N, Levin A, Schröder P, Fitting subdivision surfaces, *Proceedings of the conference on Visualization* 2001:319-324.

[54] Nasri A H, Sabin M A, Taxonomy of interpolation constraints on recursive subdivision curves, *The Visual Computer*, 2002, 18(4):259-272.

[55] Schaefer S, Warren J, A Factored Interpolatory Subdivision Scheme for Quadrilateral Surfaces, *Curves and Surface Fitting*, 2002, 373-382.

[56] Peters J, C1-surface splines. *SIAM Journal on Numerical Analysis* 1995, 32(2):645-666.

[57] Schaefer S., Warren, J., Zorin, D., Lofting curve networks using subdivision surfaces, *Proc 2004 Eurographics symposium on Geometry processing*, 2004:103-114.

[58] Baker, T. J., Interpolation from a cloud of points, *Proceedings, 12th International Meshing Roundtable*, Sandia National Laboratories, pp.55-63, Sept 2003.

[59] Xunnian Yang, Surface interpolation of meshes by geometric subdivision, *Computer-Aided Design*, 2005, 37(5):497-508.

[60] Kestutis Karciauskas and Jörg Peters, Guided Subdivision, http://www.cise.ufl.edu/research/ SurfLab/papers/05guiSub.pdf, 2005.

[61] Fuhua (Frank) Cheng, Gang Chen and Junhai Yong, Subdivision Depth Computation for Catmull-Clark Subdivision Surfaces, to appear in *Lecture Notes in Computer Science*, Springer, 2006.

[62] Shuhua Lai, Fuhua (Frank) Cheng, Similarity based Interpolation Using Catmull-Clark Subdivision Surfaces, *The Visual Computer* 22,9-11 (October 2006), 865-873.

[63] Shuhua Lai, Fuhua (Frank) Cheng, Inscribed Approximation based Adaptive Tessellation, *International Journal of CAD/CAM*, Vol. 6, No. 1, 2006.

[64] Shuhua Lai, Fuhua (Frank) Cheng, Voxelization of Free-form Solids Represented by Catmull-Clark Subdivision Surfaces, *Lecture Notes in Computer Science*, Vol. 4077, Springer, 2006, pp. 595-601.

[65] Shuhua Lai, Fuhua (Frank) Cheng, Parametrization of Catmull-Clark Subdivision Surfaces and its Applications, *Computer Aided Design & Applications*, 3, 1-4, 2006.

[66] Shuhua Lai, Fuhua (Frank) Cheng, Near-Optimum Adaptive Tessellation of General Catmull-Clark Subdivision Surfaces, *CGI 2006*, *Lecture Notes in Computer Science*, Vol. 4035, Springer, 2006, pp. 562-569.

[67] Shuhua Lai, Fuhua (Frank) Cheng, Texture Mapping on Surfaces of Arbitrary Topology using Norm Preserving based Optimization, *The Visual Computer*, 21(1-8):783-790, 2005.

[68] Shuhua Lai, Fuhua (Frank) Cheng, Adaptive Rendering of Catmull-Clark Subdivision Surfaces, *9th International Conference of Computer Aided Design & Computer Graphics*, 125-130, 2005.

[69] Shuhua Lai, Shiping Zou, Fuhua (Frank) Cheng, Constrained Scaling of Catmull-Clark Subdivision Surfaces, *Computer Aided Design & Applications*, 1(1-4): 7-16, 2004.

[70] David Guinnip, Shuhua Lai and Ruigang Yang. View Dependent Textured Splatting for Rendering Live Scenes, *ACM SIGGRAPH poster*, 2004.

[71] Shuhua Lai, Fuhua (Frank) Cheng, Robust and Error Controllable Boolean Operations on Free-Form Solids Represented by Catmull-Clark Subdivision Surfaces, Submitted.

[72] V. Settgast, K. Müler, Christoph Füfzig et.al., Adaptive Tesselation of Subdivision Surfaces in OpenSG, In *Proceedings of OpenSG Symposium*, 2003, pp:39-47.

[73] Cohen, D. and Kaufman, A., Scan Conversion Algorithms for Linear and Quadratic Objects, in *Volume Visualization*, A. Kaufman, (ed.), IEEE Computer Society Press, Los Alamitos, CA, 1990, 280-301.

[74] Kaufman, A. and Shimony, E., 3D Scan-Conversion Algorithms for Voxel-Based Graphics, *Proc. ACM Workshop on Interactive 3D Graphics*, Chapel Hill, NC, October 1986, 45-76.

[75] Mokrzycki, W., Algorithms of Discretization of Algebraic Spatial Curves on Homogeneous Cubical Grids, *Computers & Graphics*, 12, 3/4 (1988), 477-487.

[76] A. Kaufman, D. Cohen. Volume Graphics. *IEEE Computer*, Vol. 26, No. 7, July 1993, pp. 51-64.

[77] T.A. Galyean and J.F. Hughes. Sculpting: An interactive volumetric modeling technique. *Computer Graphics*, *Proceedings of SIGGRAPH91*, 25(4):267C274, July 1991.

[78] M. W. Jones and R. Satherley. Voxelisation: Modelling for Volume Graphics. In *Vision, Modeling, and Visualization 2000*, IOS Press, pp. 319-326.

[79] E.A. Karabassi, G. Papaioannou, and T. Theoharis. A fast depth-buffer-based voxelization algorithm. *Journal of Graphics Tools*, 4(4):5-10, 1999.

[80] M. Sramek. Gray level voxelisation: a tool for simultaneous rendering of scanned and analytical data. *Proc. the 10th Spring School on Computer Graphics and its Applications*, Bratislava, Slovak Republic, 1994, pp. 159-168.

[81] D. Haumont and N. Warzee. Complete Polygonal Scene Voxelization, *Journal of Graphics Tools*, Volume 7, Number 3, pp. 27-41, 2002.

[82] M.W. Jones. The production of volume data from triangular meshes using voxelisation, *Computer Graphics Forum*, vol. 15, no 5, pp. 311-318, 1996.

[83] S. Thon, G. Gesquiere, R. Raffin, A low Cost Antialiased Space Filled Voxelization Of Polygonal Objects, *GraphiCon 2004*, pp. 71-78, Moscou, Septembre 2004.

[84] Kaufman, A., An Algorithm for 3D Scan-Conversion of Polygons, *Proc. EUROGRAPHICS'87*, Amsterdam, Netherlands, August 1987, 197-208.

[85] Kaufman, A., Efficient Algorithms for 3D Scan-Conversion of Parametric Curves, Surfaces, and Volumes, *Computer Graphics*, 21, 4 (July 1987), 171-179.

[86] M. Sramek and A. Kaufman, Object voxelization by filtering, *IEEE Symposium on Volume Visualization*, pp. 111-118, 1998.

[87] S. Fang and H. Chen. Hardware accelerated Voxelisation. *Volume Graphics*, Chapter 20, pp. 301-315. Springer-Verlag, March, 2000.

[88] Beckhaus S., Wind J., Strothotte T., Hardware-Based Voxelization for 3D Spatial Analysis *Proceedings of CGIM '02*, pp. 15-20, August 2002.

[89] Sramek M. Non-binary voxelization for volume graphics, *Proceedings of Spring Conference on Computer Graphics*, 2001. p. 35C51.

[90] J. Baerentzen, Octree-based volume sculpting, *Proc. of. IEEE Visualization*, pages 9C12, 1998.

[91] N. Stolte, A. Kaufman, Efficient Parallel Recursive Voxelization for SGI Challenge Multi-Processor System, *Computer Graphics International*, 1998.

[92] Nilo Stolte, Graphics using Implicit Surfaces with Interval Arithmetic based Recursive Voxelization, *Computer Graphics and Imaging*, pp. 200-205, 2003.

[93] T. Duff, Interval arithmetic and recursive subdivision for implicit functions and constructive solid geometry, *SIGGRAPH*, pp. 131–138, July, 1992.

[94] Lee, Y. T. and Requicha, A. A. G., Algorithms for Computing the Volume and Other Integral Properties of Solids: I-Known Methods and Open Issues; II-A Family of Algorithms Based on Representation Conversion and Cellular Approximation, *Communications of the ACM*, 25, 9 (September 1982), 635-650.

[95] S. Fang and D. Liao. Fast CSG Voxelization by Frame Buffer Pixel Mapping. *ACM/IEEE Volume Visualization and Graphics Symposium 2000 (Volviz'00)*, Salt Lake City, UT, 9-10 October 2000, 43-48.

[96] Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, Markus Gross, Surface Splatting, *SIGGRAPH 2001.*

[97] Cohen Or, D., Kaufman, A., Fundamentals of Surface Voxelization, *Graphical Models and Image Processing*, 57, 6 (November 1995), 453-461.

[98] Jian Huang, Roni Yagel, V. Fillipov and Yair Kurzion, An Accurate Method to Voxelize Polygonal Meshes, *IEEE Volume Visualization'98*, October, 1998.

[99] S. Krishnan and D. Manocha, Computing Boolean Combinations of Solids Composed of Free-form Surfaces, *Proceedings of the 1996 ASME Design for Manufacturing Conference*, August 1996.

[100] R. E. Barnhill, G. Farin, M. Jordan, and B. R. Piper. Surface/surface intersection. *Computer Aided Geometric Design*, 4(1-2):3-16, July 1987.

[101] Christoph Burnikel, Kurt Mehlhorn, and Stefan Schirra. On degeneracy in geometric computations. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp16-23, New York, 1994.

[102] Katrin Dobrindt, Kurt Mehlhorn, and Mariette Yvinec. A complete and efficient algorithm for the intersection of a general and a convex polyhedron. In *Algorithms and data structures*, pp314-324, 1993.

[103] Jack Goldfeather, Jeff P. M. Hultquist, and Henry Fuchs. Fast constructive solid geometry display in the pixel-powers graphics system. *Proceedings of SIGGRAPH 1986*, 20(4):107-116.

[104] Shankar Krishnan and Dinesh Manocha. An efficient surface intersection algorithm based on lower-dimensional formulation. *ACM Transactions on Graphics*, 16(1):74-106, January 1997.

[105] Ari Rappoport and Steven Spitz. Interactive Boolean operations for conceptual design of 3D solids. *Proceedings of SIGGRAPH 97*, pp269-278, 1997.

[106] T. Sederberg and T. Nishita. Geometric hermite approximation of surface patch intersection curves. *Computer Aided Geometric Design*, 8(2):97-114, 1991.

[107] R. Seidel. The nature and meaning of perturbations in geometric computing. *Discrete Comput. Geom.*, 19(1):1-17, 1998.

[108] N. M. Patrikalakis, Surface-to-surface Intersections, *IEEE Computer Graphics and Applications*, 13(1):89-95, 1993.

[109] Bieri, H., Nef,W., Elementary set operations with d-dimensional polyhedra. *Computational Geometry and its Applications*, LNCS 333, Springer-Verlag, 1988, pp. 97-112.

[110] Chazelle, B., An optimal algorithm for intersecting three dimensional convex polyhedra. *SIAM J. Comput.*, 21(4):671-696, 1992.

[111] Wiegand, T.F., Interactive rendering of CSG models. *Computer Graphics Forum*, 15(4):249-261, 1996.

[112] Duoduo Liao, Shiaofen Fang, Fast CSG Voxelization by Frame Buffer Pixel Mapping, *Proceedings of the 2000 IEEE Symposium on Volume Visualization*, pp. 43-48, 2000.

[113] Bart Adams, Philip Dutré, Interactive Boolean operations on surfel-bounded solids, *ACM SIGGRAPH 2003*, pp651-656.

[114] Grossman, J. P., Dally, W. J. Point sample rendering. *Eurographics Rendering Workshop 1998*, pp181-192.

[115] S. G. Mallat, A theory for multiresolution signal decomposition: The wavelet representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11, pp. 674–693, Jul 1989.

[116] Charles K. Chui, An Introduction to Wavelets, (1992), *Academic Press*, San Diego, ISBN 91-58831.

[117] M. Lounsbery, T. DeRose and J. Warren, Multiresolution analysis for surfaces of arbitrary topological type, Transaction on Graphics 16,1, vol. 99, 1997.

[118] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery and W. Stuetzle, Multiresolution analysis of arbitrary meshes, *ACM SIGGRAPH 1995*, pp173-182.

[119] Adam Finkelstein, David H. Salesin, Multiresolution curves, *ACM SIGGRAPH 1994*, pp261-268.

[120] Steven J. Gortler and Michael F. Cohen, Hierarchical and Variational Geometric Modeling with Wavelets, In *Proceedings Symposium on Interactive 3D Graphics*, pp35-42, April 1995.

[121] L. Kobbelt, S. Campagna, J. Vorsatz and H. P. Seidel, Interactive multi-resolution modeling on arbitrary meshes, In *ACM SIGGRAPH 1998*, pp105C114.

[122] R. DeVore, B. Jawerth, and B. Lucier, Image compression through wavelet transform coding, *IEEE Trans. Information Theory*, 38, 2 (1992), pp. 719–746, Special issue on Wavelet Transforms and Multiresolution Analysis.

[123] Dan Piponi, George Borshukov, Seamless texture mapping of subdivision surfaces by model pelting and texture blending, *SIGGRAPH 2000*, pp. 471–478.

[124] L. Kobbelt, T. Bareuther, H. P. Seidel, Multiresolution shape deformations for meshes with dynamic vertex connectivity, *Computer Graphics Forum (Eurographics2000)*, 19(3), C249-C259 (2000).

[125] D. Gonsor and M. Neamtu. Subdivision surfaces - can they be useful for geometric modeling applications?, *Technical Report, Boeing Technical Report 01-011*, Boeing Company, 2001.

[126] Wang H, Qin K, 2004. Estimating Subidivision Depth of Catmull-Clark Surfaces. *J. Comput. Sci. & Technol. 19*, 5, 657-664.

[127] Wu X, Peters J, An Accurate Error Measure for Adaptive Subdivision Surfaces, *Proc. Shape Modeling International 2005*, 1-6.