Smooth Subdivision Surfaces Based on Triangles

by

Charles Teorell Loop

A thesis submitted to the faculty of
the University of Utah
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Mathematics

The University of Utah

August 1987

Abstract

An algorithm for generating a smooth surface from an irregular mesh of triangles is presented. The method is based on a recursive subdivision process that refines the mesh into a piecewise linear approximation of a smooth surface. The rules which govern the mesh refinement are based on well known properties of B-spline curves as well as more recent results from multivariate spline theory. A careful analysis of the smoothness of the resulting surface is made. This analysis reveals that a surface generated by this method is curvature continuous except at a fixed number of extraordinary points corresponding to mesh vertices. At these points it is found that the surface has a well defined tangent plane. An explicit formulation of this tangent plane is given and the issue of curvature continuity at these points is also discussed. Finally, a pseudo code implementation of the algorithm is given which allows the surface to be treated as a collection of individual patches.

Contents

List of Figures

Acknowledgment

I have received inspiration, encouragement, and countless hours of consultation from many generous and talented people. My deepest thanks go to individuals associated with the Mathematics department at the University of Utah: my advisors Robert Barnhill and Gerald Farin; Paul Arner, Bruce Piper, and Mark Watkins. Others whose help has been invaluable include: Tom Jensen, Allyn Rockwood, and Tony De Rose.

Chapter 1

Introduction

The purpose of this thesis is to present an algorithm that allows a designer to create a sculptured smooth surface. This surface is defined and manipulated by a structured set of *control points*. The shape of the surface is determined by the positions of the control points in space. A designer need only understand the relationship between the control points and the surface, and not the mathematics of the underlying implementation. This is one of the fundamental paradigms of Computer Aided Geometric Design (CAGD) [Boehm, Farin, and Kahmann 1984].

Many techniques have been developed that exhibit this behavior. At issue in this thesis is underlying structure of the control points. Most existing surface design schemes require that the control points take on a *regular* structure. This shall be described shortly. A regular structure can be quite restrictive from a design point of view. This thesis will present a method for designing a smooth surface that is not encumbered by this restriction, thereby giving a designer more freedom to create complex shapes.

In most design schemes the underlying shape is represented in a piecewise fashion by a polynomial basis. Piecewise polynomial means that a curve or surface is represented by a collection of individual polynomial *segments* or *patches*. Control point schemes are generally either interpolating or approximating. The choice of which scheme to use depends upon the application. For all applications, the scheme

should be *coordinate free*. This means that the relationship between the control points and the shape is independent of any coordinate system. The shape defined by an interpolating scheme will pass through all of the control points. This type of scheme is well suited for *representation*, i.e., if the control points are known to belong to an existing shape. Interpolating methods may not be the natural choice for design because they generally lack *local control*. This means that modifying the shape by moving a control point will affect the entire shape. Local control is important from a design point of view. A designer wants to be sure that moving a control point is not going to affect some part of the shape that is already deemed correct. In order to obtain local control, interpolation is sacrificed. This leaves schemes which approximate the control points.

Several such approximating schemes exist; the simplest and best known are B-splines (B for basic). The control points of a B-spline scheme are known as *de Boor points*. For B-spline curves, the de Boor points are an ordered set that forms the *de Boor polygon*. These points can be specified by a designer. The resulting curve is a smooth approximation to the de Boor polygon. Figure 1.1 illustrates the relationship between the de Boor points and a B-spline curve. B-spline schemes exhibit the important local control property. The arrow in Figure 1.1 represents the displacement of a single de Boor point, the local effect on the curve has been shaded.

The classic approach of extending B-spline curves to surfaces involves taking a *tensor product* of B-spline curves. This can loosely be described as taking a series of parallel curves in one direction, and *sweeping* these with another curve in the perpendicular direction to form a surface. Similarly, a series of parallel de Boor polygons may be connected in the perpendicular direction to form a *de Boor net*. From this construction, the de Boor net takes on a *rectangular* structure. Figure 1.2 illustrates the de Boor net and resulting tensor product B-spline surface. The

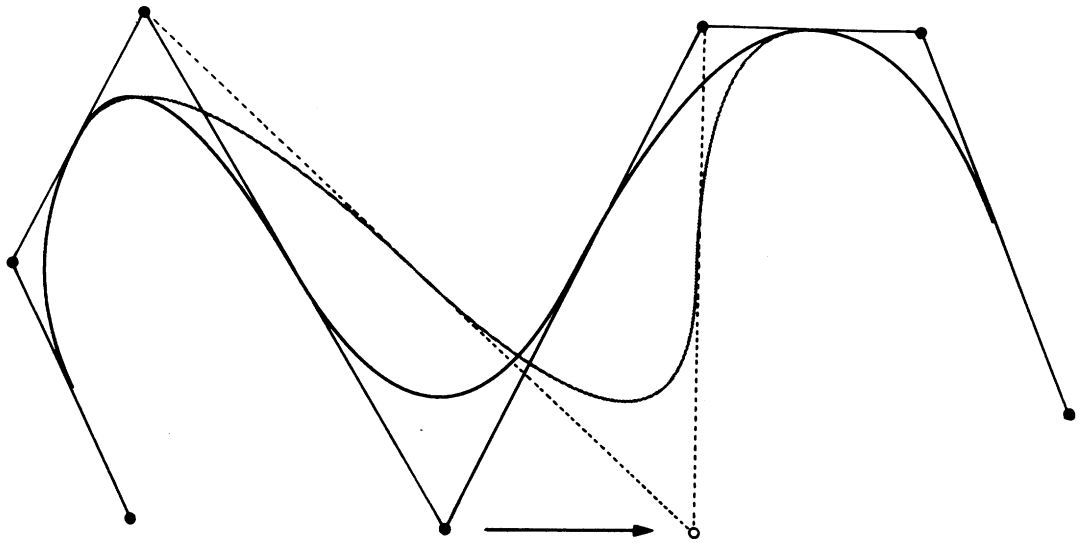

Figure 1.1: A B-spline curve.

tensor product B-spline scheme has local control and produces a smooth surface that approximates the de Boor net.

A more recent addition to the class of smooth, locally controlled, approximating surface schemes are *triangular splines*. Triangular spline surfaces behave much like tensor product B-spline surfaces. A triangular spline surface is defined by a de Boor net that has a regular *triangular* structure. Such a surface is illustrated in Figure 1.3. The properties of triangular splines are of special interest in this thesis. The surface design scheme to be presented is a generalization of triangular splines, where the de Boor net need not be regular.

A variety of techniques exist for computing a representation of the underlying shape of a B-spline scheme. Generally one explicitly evaluates the underlying piecewise polynomials a sufficient number of times to obtain a piecewise linear approximation to the curve or surface. B-spline schemes offer an interesting alternative. Each curve segment may be reparameterized as two or more sub-segments.

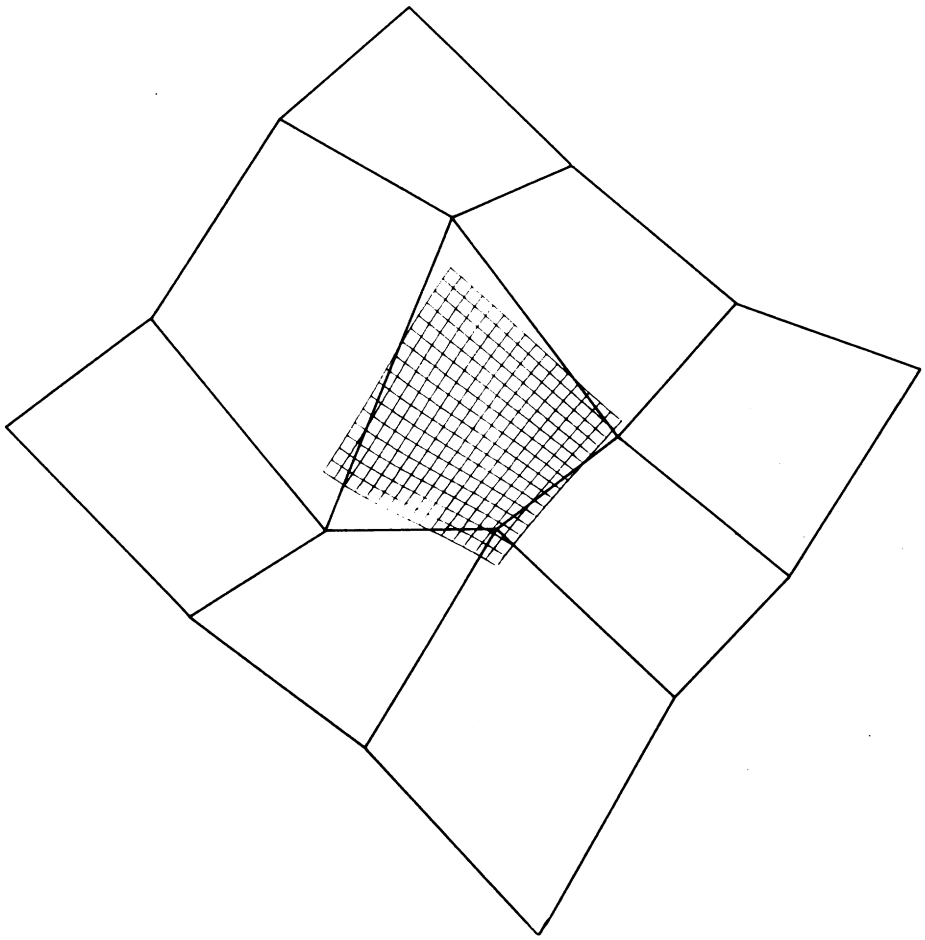

Figure 1.2: A tensor product B-spline surface and associated de Boor net.

This reparameterization has the effect of *refining* the de Boor polygon. The refined de Boor polygon is a denser collection of points that is in some sense "closer" to the underlying curve than the original de Boor polygon. Such a reparameterization or ***subdivision*** of the polynomial segments may be done repeatedly, i.e., each sub-segment may be subdivided. After repeated subdivision, the refined de Boor polygon is indistinguishable from the B-spline curve. This same principle may be applied to a surface, resulting in a refined de Boor net that is indistinguishable from the surface. In this thesis, only *binary* subdivision is considered. For curves, this means each polynomial segment is reparameterized as two sub-segments. For

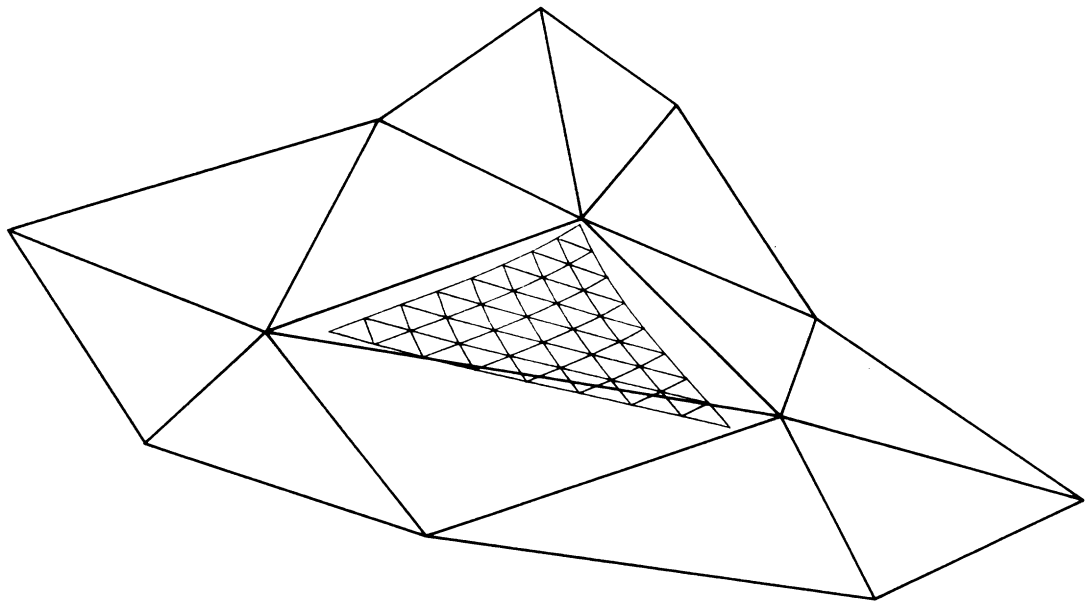

Figure 1.3: A triangular spline surface and associated de Boor net.

surfaces, each polynomial patch is subdivided into four sub-patches.

In principle, refinement of the de Boor net is a *geometric* operation. A set of geometric rules based on the regular structure of the de Boor net governs the refinement process. This geometric approach to subdivision will be adapted to work for irregular or arbitrary control point sets. Such extensions of B-splines to arbitrary topologies are not new. Extending the geometric properties of tensor product B-splines has been considered [Doo and Sabin 1978][Catmull and Clark 1978]. Extending the subdivision rules for triangular splines will be considered in this thesis.

Chapter 2

Binary Subdivision of B-splines

In this chapter, B-spline curves, Tensor product B-spline surfaces, and Triangular spline surfaces will be discussed. Fundamental concepts, properties, and formulas are presented for each. Special attention is paid to deriving *Binary* subdivision formulas.

2.1 Univariate B-splines

A degree r, piecewise polynomial B-spline curve $S^r(u)$ is defined:

$$S^r(u) = \sum_i d_i^r N_i^r(u). \tag{2.1}$$

The vector valued coefficients d_i^r form the *de Boor polygon*. The $N_i^r(u)$ are *normalized* B-splines of degree r defined over a sequence of knots i . This knot sequence forms a partition of the real u-axis. Only the case of equidistant knot spacing is discussed. For simplicity, knot sequences will be derived from the sequence

$$\mathcal{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}.$$

Under these restrictions, (2.1) becomes

$$S^r(u) = \sum_{i \in \mathcal{Z}} d_i^r N^r(u - i). \tag{2.2}$$

The $N^r(u - i)$ can be viewed as translates of a single B-spline $N^r(u)$. $N^r(u)$ has the following properties:

- Partition of unity : $\sum_{i \in Z} N^r(u-i) = 1$
- Positivity : $N^r(u) \geq 0$
- Local support : $N^r(u-i) = 0 \quad \text{if } u \notin [i, i+r+1]$
- Continuity : $N^r(u)$ is $(r-1)$ times continuously differentiable
- Recursion : $N^r(u-i) = \frac{u-i}{r} N^{r-1}(u-i) + \frac{i+r+1-u}{r} N^{r-1}(u-i-1)$

$$\text{where} \quad N^0(u) = \begin{cases} 0, & \text{if } u \in [i, i+1] \\ 1, & \text{otherwise.} \end{cases}$$

Figure 2.1 illustrates some examples.

The first three properties imply the convex hull property, i.e., for any u , $S^r(u)$ is a convex combination of a local subset of the de Boor points d_i^r . This local nature means that a change in a de Boor point will affect only a small portion of the curve. The local control property defines $[i, i+r+1]$ as the *support* of $N^r(u-i)$. The $N^r(u)$ form a basis for degree r polynomials, i.e., any univariate degree r polynomial is a linear combination of B-splines. The important recursion formula, independently discovered by Cox, de Boor, and Mansfield [de Boor 1978] shows that a B-spline of degree $r > 0$ is a linear blend of lower degree B-splines. It also provides a stable and efficient means of evaluating $S^r(u)$.

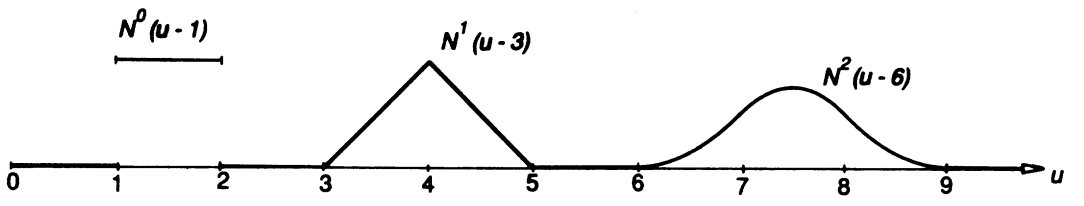

Figure 2.1: Examples of univariate B-splines

2.2 Subdivision of B-spline curves

Subdivision of the de Boor polygon results in a refined polygon that more closely approximates the shape of the underlying curve. It is well known [Riesenfeld 1975] that under repeated subdivision, the de Boor polygon will converge to the underlying curve.

The idea behind a subdivision scheme is to rewrite the curve (2.2), as a curve over a refined knot sequence. Binary refinement of the sequence $\mathcal{Z}$ results in the sequence

$$\mathcal{Z}/2 = \{\ldots, -1, -\tfrac{1}{2}, 0, \tfrac{1}{2}, 1, \ldots\}.$$

From this, (2.2) becomes

$$S^r(u) = \sum_{j \in \mathcal{Z}/2} \hat{d}_j^r N^r(2(u-j)) \tag{2.3}$$

where the $\hat{d}_j^r$ make up the refined de Boor polygon. Note that the support of $N^r(2u)$ is half that of $N^r(u)$.

It is possible to determine the $\hat{d}_j^r$ by considering the subdivision of a single B-spline. A single B-spline may be *decomposed* into similar B-splines of half the support, as in Figure 2.2. This is accomplished algebraically by rewriting $N^r(u)$ as a linear combination of B-splines of degree r over the refined knot vector. This results in

$$N^r(u) = \sum_{j \in \mathcal{Z}/2} c_j^r N^r(2(u-j)). \tag{2.4}$$

The c_j^r are the special case $\hat{d}_j^r$ for a single B-spline. Each c_j^r is a function of r and j, which may be written

$$c_j^r = 2^{-r}\binom{r+1}{2j}. \tag{2.5}$$

Proof that this formula is correct may be found by induction on r, using the B-spline recursion formula to relate B-splines of different degrees.