SUBDIVISION OR REFINEMENT B-SPLINE CURVES AND SURFACES

by

KANG YUN PYNG

Dissertation submitted in partial fulfillment

of the requirements for the degree

of Master of Science in Mathematics

May 2005

ACKNOWLEDGEMENTS

I am grateful and thankful to my supervisor, Associate Professor Ahmad Abdul Majid of Universiti Sains Malaysia, for his invaluable guidance and assistance in the completion of this dissertation.

Further, I would like to thank my friend, Mr Chew Peng Chun for many useful discussions and suggestions.

Last but not least, I would like to thank my parents for their support, advice and encouragement.

Thank you.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

CAGD is the abbreviation for Computer Aided Geometric Design. In CAGD, the subdivision scheme is gaining popularity among the computer graphics and geometric modeling community and has achieved considerable success.

The objectives of this dissertation are to study and to understand the subdivision schemes of Chaikin's algorithm and Catmull & Clark's algorithm. Then, these subdivision schemes is applied to obtain refined curve and surface by using the software, Mathematica.

The Chaikin's algorithm is used to refine a curve from an arbitrary control polygon, formed by a set of control points.

By repeating the application of Catmull & Clark's algorithm, the limit of a sequence successfully refined the polyhedral meshes created. Therefore, a visually pleasing refined surface can be generated.

LENGKUNG DAN PERMUKAAN SPLIN-B SUBBAHAGIAN ATAU KEHALUSAN

ABSTRAK

CAGD ialah singkatan bagi " Computer Aided Geometric Design " atau dalam bahasa Malaysia sebagai " Rekabentuk Geometri Berbantukan Komputer " . Dalam CAGD, kaedah subbahagian semakin terkenal di kalangan komuniti komputer grafik dan pemodelan geometri.

Objektif disertasi ini ialah membaca dan memahami skema subbahagian algoritma Chaikin dan algoritma Catmull & Clark. Kemudian, kita mengaplikasikan skema subbahagian ini untuk memperoleh satu lengkung dan permukaan yang halus dengan menggunakan perisian " Mathematica ".

Bagi lengkung, kita menggunakan algoritma Chaikin untuk membina satu lengkung yang halus daripada pelbagai poligon kawalan yang dibentukkan daripada satu set titik kawalan.

Dengan mengulangi aplikasi algoritma Catmull & Clark, batas jujukan dapat menghaluskan lagi polyhedral yang dihasilkan. Maka, suatu permukaan yang lebih halus, dari segi visual, dapat dihasilkan.

CHAPTER 1

REVIEW ON B-SPLINE CURVES AND SURFACES

1.0 Introduction

Bèzier representations of curves and surfaces used in computer graphics, were indepently discovered by Pierre Bèzier about 1962, an engineer for Renault, and Paul de Casteljau about 1959, an engineer for Citroen. Both worked for automobile companies in year 1970, in France, these engineers initially developed a curve representation scheme that is geometrical in construction, and the mathematical theory is based on the concept of Bernstein polynomials. They extended it to a surface patch that has become the de-facto standard for surface generation in computer graphics (Mortenson, 1997).

B-splines are piecewise polynomial functions with minimal support. The proper B-spline blending functions are used to construct a basis of B-spline vector space functions. B-spline functions are important tools in computer aided geometric design (CAGD) and computer graphics. B-splines were introduced by I. J. Schoenberg in 1946. Schoenberg introduced these B-spline functions in the study of approximation of equidistant data by analytic functions, while parametric spline curves were used in CAGD by Ferguson, de Boor, and Gordon to design freeform curves and surfaces (Mortenson, 1997).

B-splines of degree n are piecewise polynomial functions with minimal support and C^{n-1} continuous. There are many methods to generate curves and surfaces using spline functions and the most efficient is to make use of B-splines. A spline function is simply a piecewise polynomial having a certain level of smoothness. A spline curve is an affine blending of points using piecewise polynomial blending functions (Marsh, 1999)

The purpose of this chapter is to review methods on generating B-spline curves and surfaces. B-spline curves and surfaces are briefly introduced in section 1.1. The uniform B-spline in terms of truncated power function is defined in section 1.1.1.

1.1 B-Spline Curves and Surfaces

1.1.1 Uniform B-splines curves

The truncated power function or one sided power function is used to show the B-spline basis function j.

$$M_n(t) = \frac{1}{n!}\sum_{i=0}^{n+1}(-1)^n\binom{n+1}{1}(i-t)_+^n \qquad \text{,for n = 0, 1, 2, ..., and } t \in \Re$$

where

$$t_+^n = \begin{cases} t^n & ,\forall t \geq 0 \\ 0 & ,\forall t < 0 \end{cases}$$

We should know that $t_+^n \in C^{n-1}(\Re)$ and the nth order derivative of t_+^n is continuous at all t except t = 0. Let us study a few examples with different n to clarify the equation above. (Marsh, 1999)

For n = 0,

$$M_0(t) = t_+^n - (t-1)_+^0$$

$$= \begin{cases} 0 & ,t < 0 \\ 1 & ,0 \le t < 1 \\ 0 & ,t \ge 1 \end{cases}$$

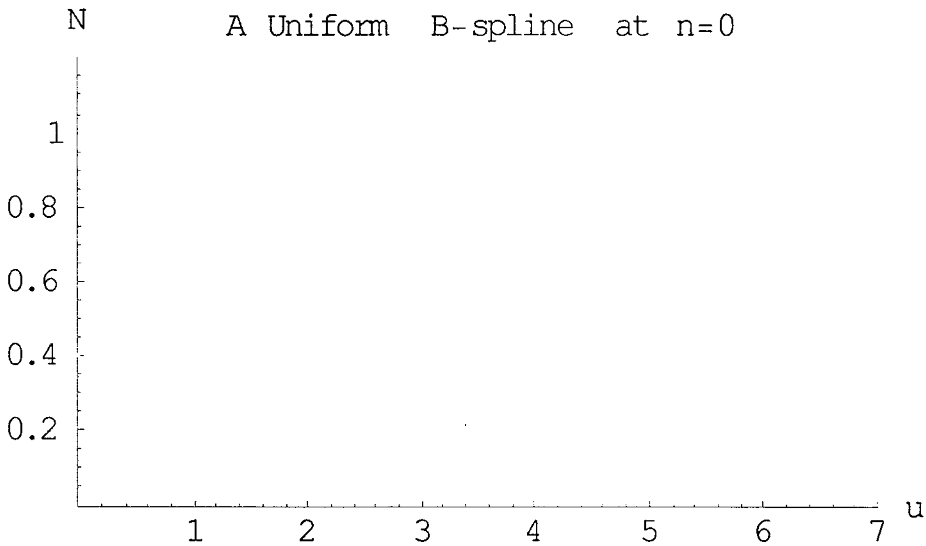

Figure 1.1 A uniform B-spline of degree 0, M_0 (t)

For n = 1,

$$M_1(t) = t_+ - 2(t-1)_+ + (t-2)_+$$

$$= \begin{cases} 0 & ,t < 0 \\ t & ,0 \le t < 1 \\ 2-t & ,1 \le t < 2 \\ 0 & ,2 \le t \end{cases}$$

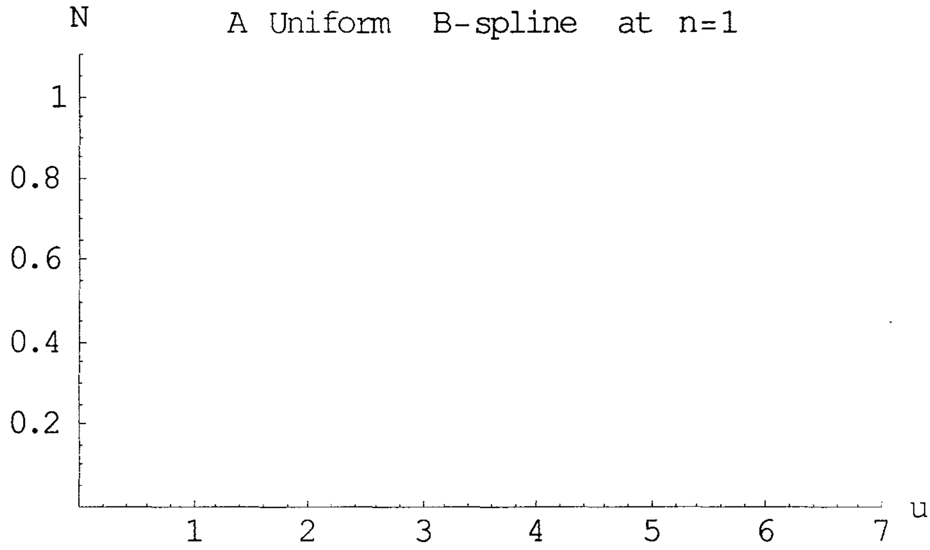

Figure 1.2 A uniform B-spline of degree 1, M_1 (t)

For n = 2,

$$M_2(t) = \frac{1}{2}\left(t_+^2 - 3\,(t-1)_+^2 + 3\,(t-2)_+^2 - (t-3)_+^2\right)$$

$$= \begin{cases} 0 & ,t < 0 \\ \frac{1}{2}t^2 & ,0 \le t < 1 \\ \frac{1}{2}(-2t^2 + 6t - 3) & ,1 \le t < 2 \\ \frac{1}{2}(t^2 - 6t + 9) & ,2 \le t < 3 \\ 0 & ,t \ge 3 \end{cases}$$

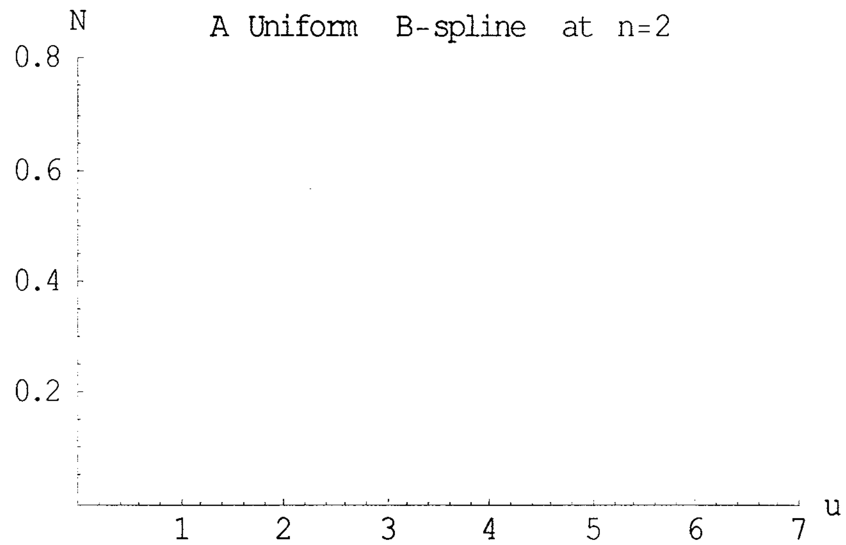

Figure 1.3 A uniform B-spline of degree 2, M_2 (t)

For n = 3,

$$M_3(t) = \frac{1}{6}((t_+^3 - 4\,(t-1)_+^3 + 6\,(t-2)_+^3 - 4\,(t-3)_+^3 + (t-4)_+^3$$

$$= \begin{cases} 0 & ,t<0 \\ \frac{1}{6}t^3 & ,0 \le t < 1 \\ \frac{1}{6}(-3t^3 + 12t^2 - 12t + 4) & ,1 \le t < 2 \\ \frac{1}{6}(3t^3 - 24t^2 + 60t - 44) & ,2 \le t < 3 \\ \frac{1}{6}(-t^3 + 12t^2 - 48t + 64) & ,3 \le t < 4 \\ 0 & ,t \ge 4 \end{cases}$$

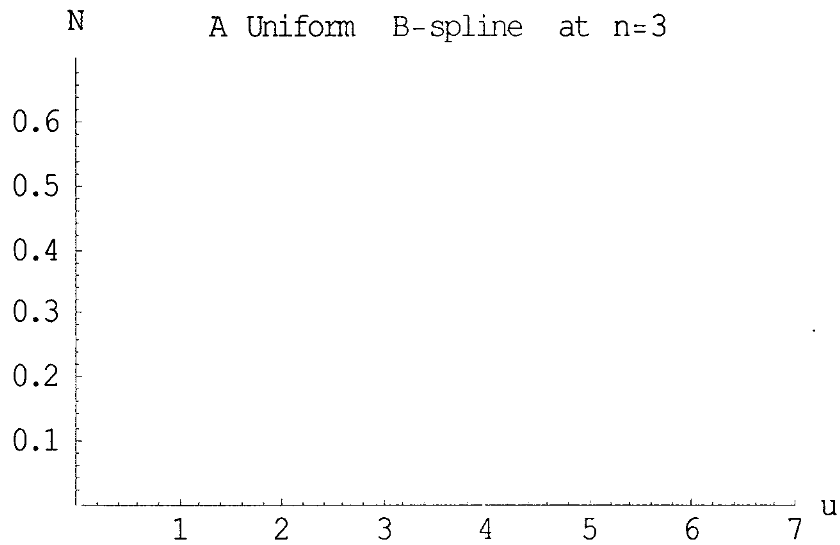

Figure 1.4 A uniform B-spline of degree 3, M_3 (t)

The properties of $M_3(t)$ are given below:

(i) $M_3(t) \in C^2(\Re)$ as $(t\text{-}1)^3_+ \in C^2(\Re)$, $\forall\ i = 0,1,2,3,4$.

(ii) $M_3(t)$ vanishes outside (0,4).

(iii) $M_3(t) > 0$, $\forall\ 0 < t < 4$.

(iv) $M_3(t)$ consists of 4 polynomial segments. (Marsh, 1999)

On the other hands, $M_3(t)$ in terms of 4 polynomial segments with $t \in [0,1]$ can also be written as follow,

$$b_0(t) = \frac{1}{6}t^3$$

$$b_1(t) = \frac{1}{6}(-3t^3 + 3t^2 - 3t + 4)$$

$$b_2(t) = \frac{1}{6}(3t^3 - 6t^2 + 4)$$

$$b_3(t) = \frac{1}{6}(-t^3 + 3t^2 - 3t + 1)$$

verify that

$$b_1(t) + b_2(t) + b_3(t) + b_4(t) = 1 \qquad , \forall\ t \in [0,1].$$

Generally,

(i) $M_n(t) \in C^{n-1}(\Re)$ as $(t\text{-}1)_+^n \in C^{n-1}(\Re)$, $\forall\ i = 0,1,2,\ldots,n.$

(ii) $M_n(t) = 0$, $\forall\ t \notin (0,n+1)$

(iii) $M_n(t) > 0$, $\forall\ t \in (0,n+1)$.

(iv) $M_n(t)$ consists of (n+1) polynomial segments. (Catmull & Clark, 1978)

$M_n(t)$ is known as a uniform B-spline of degree n. If n=3, $M_n(t)$ is called as a uniform cubic B-spline. Furthermore, we can translate B-spline $M_3(t)$ from $M_3(t\text{-}\upsilon)$, $\upsilon \in Z$.

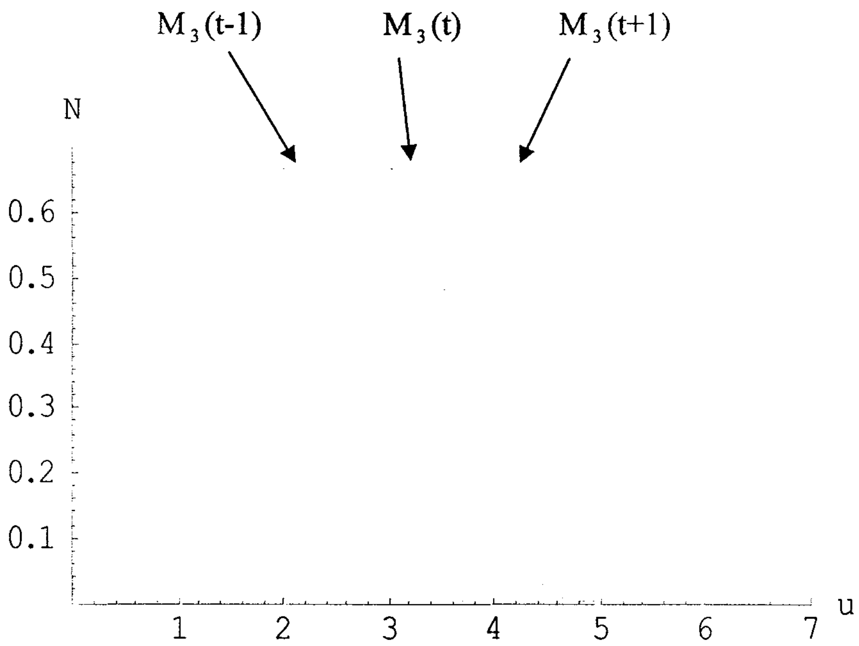

Figure 1.5 Translation of uniform B-spline of degree 3, $M_3(t)$

In addition, we can use the equation below to generate a curve with a set of control points, $V_i \in \Re^2$ or $\Re^3$, where i = 1, 2, 3, … , m

$$P(t) = \sum_{i=0}^{m} V_i M_n(t-i)$$

Let n = 3,

$$P(t) = \sum_{i=0}^{3} V_i M_3(t-i)$$

$$= V_0 M_3(t) + V_1 M_3(t-1) + V_2 M_3(t-2)$$

The curve which be generated is known as a uniform B-spline curve. The control polygon of the curve is formed by joining consecutive points, V_i, V_{i+1}, V_{i+2}, … , and so on.

Figure 1.8 shows that the closed curve is formed by 4 pieces B-spline curves.

Observe that

$$M_0(0) = M_3(1)$$

With

$$V_0 = V_3$$

$$V_1 = V_0$$

$$V_2 = V_1$$

$$V_3 = V_2$$

Finally, the curve which be formed is shown in the matrix form below,

$$M_3(t) = \frac{1}{6}\begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}\begin{bmatrix} V_j \\ V_{j+1} \\ V_{j+2} \\ V_{j+3} \end{bmatrix} \qquad \text{(Marsh, 1999)}$$

Generally, in order to form the closed curve the matrix form is as follows,

$$M_j(t) = \frac{1}{6}\begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}\begin{bmatrix} V_{j \bmod (n+1)} \\ V_{j+1 \bmod (n+1)} \\ V_{j+2 \bmod (n+1)} \\ V_{j+3 \bmod (n+1)} \end{bmatrix} \qquad \text{(Marsh, 1999)}$$

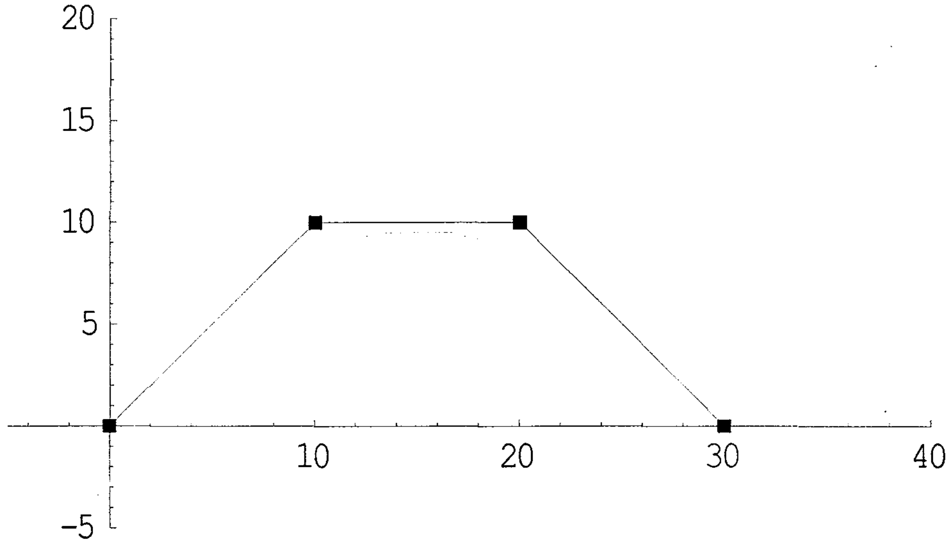

Figure 1.6 A cubic B-spline curve which is formed by 4 control points.

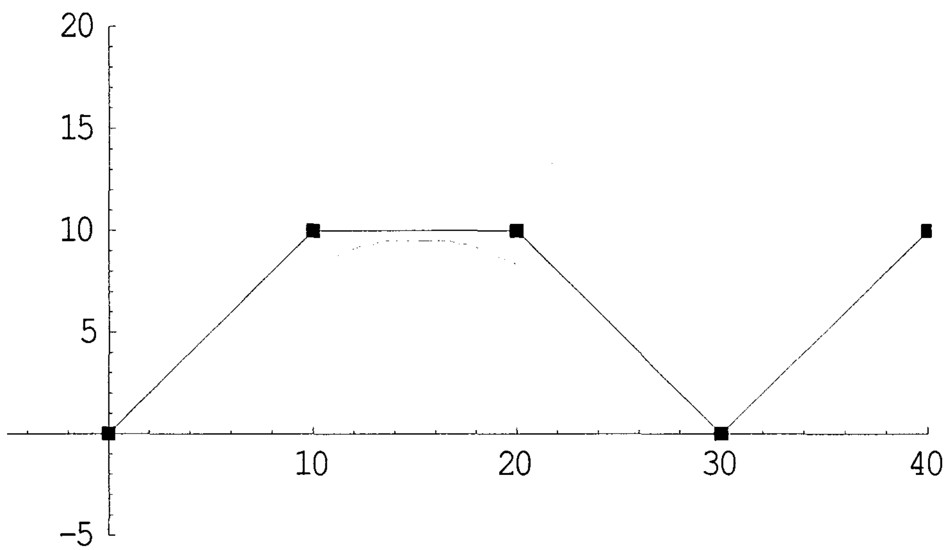

Figure 1.7 A quartic B-spline curve which is formed by 5 control points.

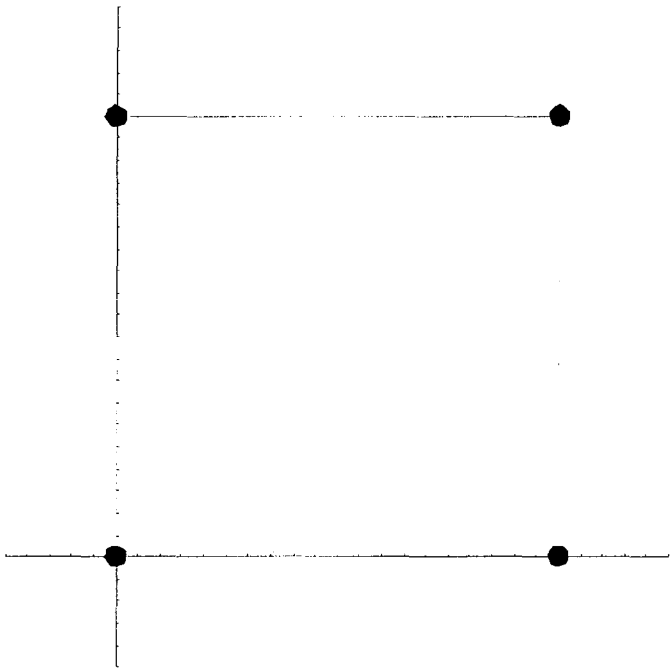

Figure 1.8 A closed B-spline curve formed by 4 pieces B-spline curve with 4 control points.

1.1.2 B-spline surfaces

Let $N_{i,d}(u)$ be the B-spline basis functions of degree d with knot vector $u_0, u_1, \ldots, u_m$, and let $N_{j,e}(v)$ be the B-spline basis functions of degree e with knot vector $v_0, v_1, \ldots, v_q$. A B-spline surface with control points $\mathbf{P}_{i,j}$ $(0 \le i \le n = m - d - 1,\ \ 0 \le j \le p = q - e - 1)$ is defined by

$$s(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{p} \boldsymbol{P}_{i,j} N_{i,d}(u) N_{j,e}(v) \qquad \text{, for } (u,v) \in [u_d, u_{m-d}] \times [v_e, v_{q-e}]$$

1.1.1.1 Properties of B-spline Surfaces

A B-spline surface satisfies the following properties.

- **Local Control**

  Each segment is determined by a $(d+1)\times(e+1)$ mesh of control points. If $u \in [u_\sigma, u_{\sigma+1})$ and $v \in [v_r, v_{r+1})$ $(d \le \sigma \le m-d-1,\ e \le r \le n-e-1)$ then

  $s(u,v) = \sum_{i=\sigma-d}^{\sigma} \sum_{j=r-e}^{r} \boldsymbol{P}_{i,j} N_{i,d}(u) N_{j,e}(v)$, for $(u,v) \in [u_d, u_{m-d}] \times [v_e, v_{q-e}]$. (Marsh, 1999)

- **Convex Hull**

  If $u \in [u_\sigma, u_{\sigma+1})$ and $v \in [v_r, v_{r+1})$ $(d \le \sigma \le m-d-1,\ e \le r \le n-e-1)$ then s(u,v) lies in the convex hull defined by its control net. Since s(u,v) is the linear combination of all its control points with positive coefficients whose sum is 1 (partition of unity), the surface lies in the convex hull of its control points. (Marsh, 1999)

CHAPTER 2

SUBDIVISION CURVES

2.0 Introduction

First of all, let us study about subdivision schemes, Catmull and Clark described a simple generalization of the subdivision rules for bicubic B-splines to arbitrary quadrilateral surface meshes. This subdivision scheme has become a main scheme of surface modeling systems. Unfortunately, little is known about the smoothness and regularity of this scheme due to the complexity of the subdivision rules. The scheme automatically produces reasonable rules for topology and can easily be extended to incorporate boundaries and embedded creases expressed as Catmull-Clark surfaces and B-spline curves. (Catmull & Clark, 1978)

This curve generation methods refines the control polygon into a sequence of new control polygons that converge to the curve. While the freeform from a closed form mathematical expression is achieved, a wide variety of curve types can be expressed. The curves are known as subdivision curves as the methods are based upon the binary subdision of the uniform B-spline curves. (Chaikin, 1974)

What's the subdivision algorithm for uniform B-spline curves? Recall that the control points which is formed the control polygon with V_i, i = 0, 1, 2, … ,m , to generate the curve by P (t) = $\sum_{i=0}^{m} V_i M_n(t)$. The importance of this subdivision algorithm is to obtain at each step, a finer control polygon to approximate the curve. The final control polygon which is fine enough it would be the time the curve is formed.

This chapter purposely to discuss on methods of subdivision curves. After introduction of subdivision curves, we talk on constructing curve segments. In details, section 2.1.1 is discussed about linear blend, quadratic blend in section 2.1.2. and cubic blend in section 2.1.3. The Chaikin's algorithm, well-known as an algorithm for high speed curve generation, which be introduced in section 2.2. A few examples on this curve generation method on the subsection of 2.2. The corner-cutting algorithm which generates a new control polygon by cutting the corners off the original one is introduced in section 2.2.1.

2.1 Constructing Curve Segments

2.1.1 Linear blend

We have to study the method of constructing curve segments. Basically, linear blend of line segment is formed from an affine combination of points.

$$P_0^1(t) = (1-t)\,P_0 + t\,P_1 \qquad\qquad = 1$$

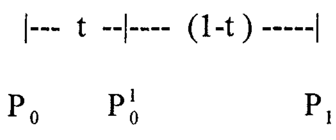

Figure 2.1 Linear blend

2.1.2 Quadratic blend

Let us discuss on quadratic blend which 3 control points are needed to form a control polygon. Quadratic blend is a quadratic segment which is formed from an affine combination of line segments.

$$P_0^1(t) = (1\text{-}t)\,P_0 + t\,P_1 \qquad = 1$$

$$P_1^1(t) = (1\text{-}t)\,P_1 + t\,P_2$$

$$P_0^2(t) = (1\text{-}t)\,P_0^1(t) + t\,P_1^1(t) \qquad = 2$$

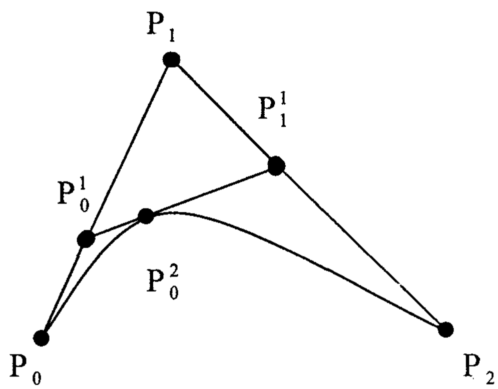

Figure 2.2 Quadratic Blend

2.1.3 Cubic Blend

We see how is the cubic blend done. The cubic segment can be formed from an affine combination of quadratic segments. 4 control points are needed to form the control polygon of this cubic blend.

$$P_0^1(t) = (1\text{-}t)\,P_0 + t\,P_1 \qquad = 1$$

$$P_1^1(t) = (1\text{-}t)\,P_1 + t\,P_2$$

$$P_0^2(t) = (1\text{-}t)\,P_0^1(t) + t\,P_1^1(t) \qquad = 2$$

$$P_1^1(t) = (1-t)\,P_1 + t\,P_2$$

$$P_2^1(t) = (1-t)\,P_2 + t\,P_3$$

$$P_1^2(t) = (1-t)\,P_1^1(t) + t\,P_2^1(t)$$

$$P_0^3(t) = (1-t)\,P_0^2(t) + t\,P_1^2(t)$$

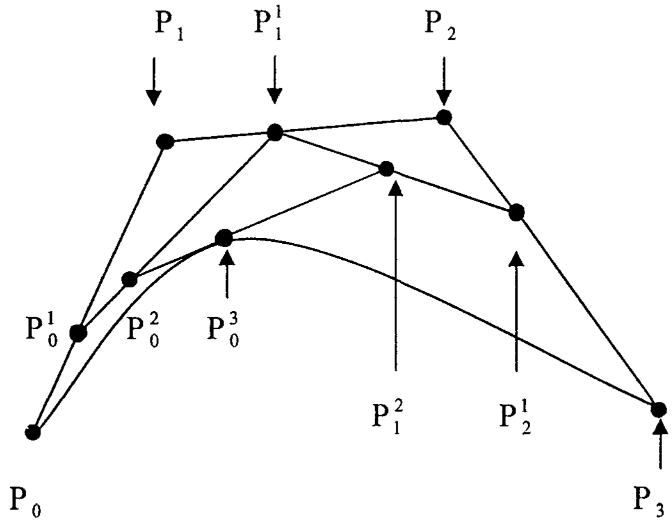

Figure 2.3 Cubic blend.

2.2 Chaikin's Algorithm

Chaikin's algorithm which is known as an algorithm for high speed curve generation is verified by George Merrill Chaikin ,year 1974. Chaikin's algorithm was the first corner cutting (refinement algorithms) used to generate a curve from a set of control points. (Chaikin, 1974)

The algorithm is recursive, using only integer addition, one-bit right shifts, complementation and comparisons, and produces a sequential list of raster points which constitute the curve. The curve consists of concatenated segments, where each segment is smooth and open. When the segment is smooth enough, it would be the time the curve is formed. (Chaikin, 1974)

2.2.1 The Corner-cutting Algorithm

Researchers since Bèzier had been working with curves generated by control polygons but had focused their analysis on the underlying analytical representation, frequently based upon Bernstein polynomials. Chaikin had different idea and had decided to develop algorithm that directly worked with the control polygon, called as geometric algorithm.

Chaikin's curve generation scheme is mostly based upon corner-cutting where the algorithm generates a new control polygon by cutting the corners off the original one. Figure 2.4 shows the idea where an initial control polygon has been refined into a second polygon which is slightly offset by cutting off the corners of the first sequence. Clearly, we could then take the second control polygon to cut the corners off it in order to produce a third sequence. Finally, a curve would be formed.

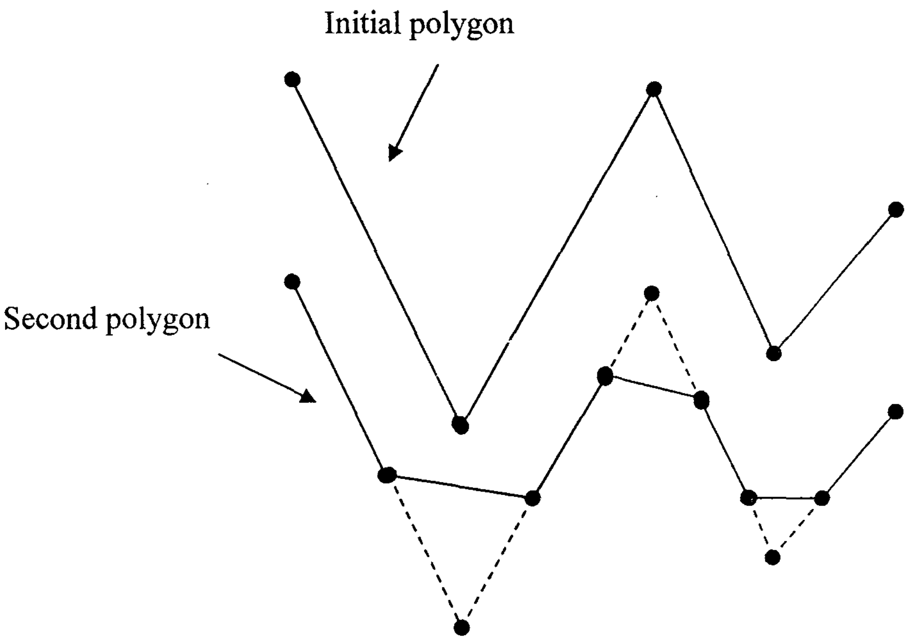

Figure 2.4 An initial polygon has beed refined into a second polygon which is offset by cutting off the corners of the first sequence.

2.2.2 Chaikin's Algorithm for B-spline

Given $V_0, V_1, V_2, \ldots, V_N$ as control points to form a control polygon. There are two steps of Chaikin's algorithm on generating a B-spline curve. This is a high speed algorithm of curve generation which is done starting from a given polygon, constructed a sequence of new control polygons converging to a new curve. (Chaikin, 1974)

Step I

$$B_{2j+\upsilon} = \frac{1}{2}(V_j + V_{j+\upsilon})$$

where, for $\upsilon = 0$, j = 0, 1, 2, … , N

for $\upsilon = 1$, j = 0, 1, 2, … , N-1

Step II

$$V_j^1 = \frac{1}{2}(B_j + B_{j+1})$$

where, for j = 0, 1, 2, … , 2N-1 (N ≥ 2)

An initial control polygon is formed by these initial control points,

$$V_0, V_1, V_2, \ldots, V_N,$$

after that, a new control polygon be evaluated by the control points,

$$V_0^1, V_1^1, V_2^1, \ldots, V_{2N-1}^1.$$

Note that B_0 as a control polygon which is denoted by the control points or called first iteration of Chaikin's algorithm. Step I and step II are repeated and B_i, a new control polygon be formed in second iteration.

Subsequently, a refined curve would be generated after a few iterations are done on the particular control polygons.

For instance,

we form a quadratic curve, N = 2, where N is called as number of control points, V_0, V_1 and V_2. Assign that, n = 2, where n is refered to degree.

Step I,

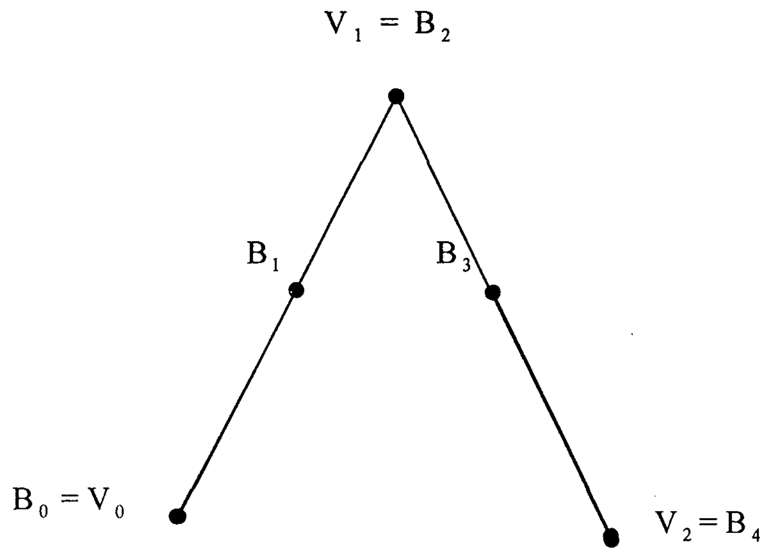

Figure 2.5 Step I on constructing the points of B_0, B_1, B_2, B_3 and B_4.

In Figure 2.5,

For $\upsilon = 0$, j = 0, 1, 2.

$$B_0 = \frac{V_0 + V_0}{2} = V_0$$

$$B_2 = \frac{V_1 + V_1}{2} = V_1$$

$$B_4 = \frac{V_2 + V_2}{2} = V_2$$

For $\upsilon = 1$, j = 0, 1

$$B_1 = \frac{V_0 + V_1}{2}$$

$$B_3 = \frac{V_1 + V_2}{2}$$

Step II,

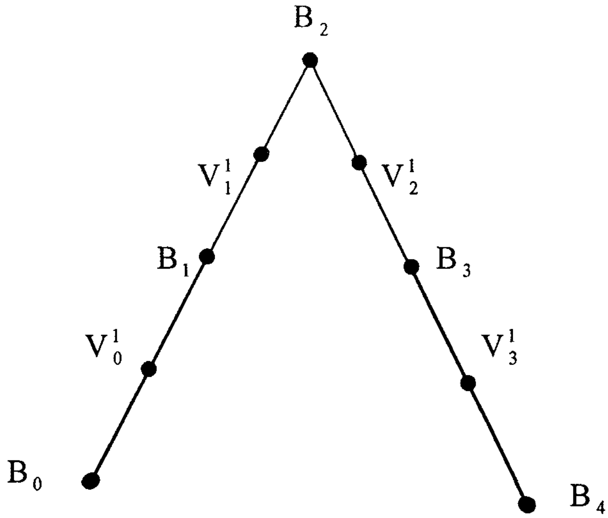

Figure 2.6 4 points are constructed

For j = 0, 1, 2, 3.

$$V_j^1 = \frac{1}{2}(B_j + B_{j+1})$$

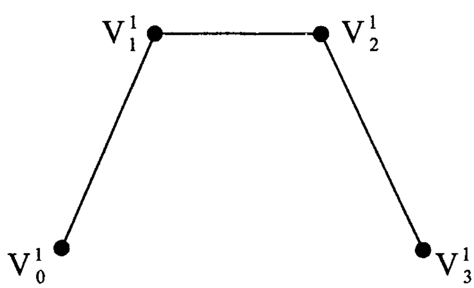

Figure 2.7 New control polygon

In step II, a procedure of constructing a new control polygon which is shown in Figure 2.7. Repeat step I and step II until a refined curve is generated.

2.2.3 Subdivision Algorithm for Uniform B-spline

Given any control points with $a_i \in \Re^2$ or $\Re^3$. A uniform B-spline curve can be generated as follows,

$$P(t) = \sum_{i=0}^{N} a_i M_n(x-i) \qquad , \text{where } n \le x \le N+1$$

In order to generate the curve more efficiently, the subdivision algorithm is suitable to apply on it. Let us go through step by step. (Chaikin, 1974)

Step I,

For k = 0, 1, 2, ... , where k is number of iteration.

$$b^1_{2j+\upsilon} = \frac{1}{2}(a^k_j + a^k_{j+\upsilon}) \qquad , \text{For } \upsilon = 0, j = 0, 1, \dots, m_k$$

$$, \text{For } \upsilon = 1, j = 0, 1, \dots, m_k - 1$$

$$m_k = 2^k N - (2^k - 1)n + (2^k - 1) \qquad , k = 0, 1, 2, \dots$$

Step II,

For p = 2, 3, ... , n , where p is degree.

$$b^p_j = \frac{1}{2}(b^{(p-1)}_j + b^{(p-1)}_{j+1}) \qquad , j = 0, 1, \dots, 2m_k - p + 1$$

next, set the points as

$$a^{(k+1)}_j = b^n_j \qquad , j = 0, 1, 2, \dots, 2m_k - n + 1$$

The following steps are keep on repeating both step I and step II until a refined curve is achieved.

For example,

we form a cubic curve, N = 3, where N is number of control points, with 4 control points, a_0, a_1 , a_2 and a_3. Assign that, n = 3, where n is degree, and k is number of iteration.

Step I,

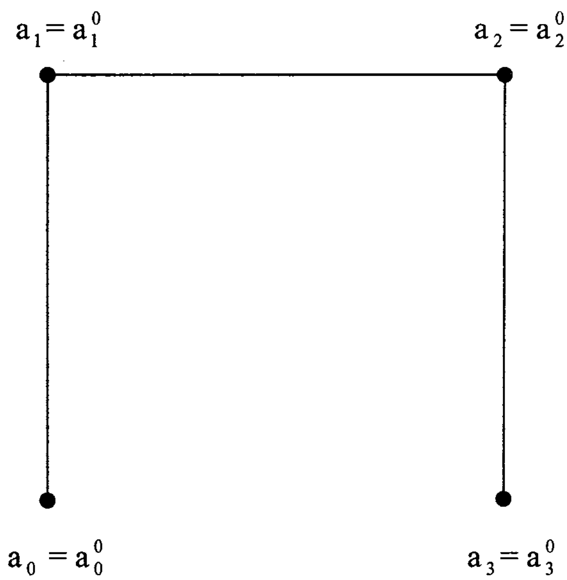

Figure 2.8 Control polygon.

For $k = 0$,

When $\upsilon = 0$,

$$m_k = 2^0 (3) - (2^0\text{-}1)\, 3 + (2^0\text{-}1) = 3$$

For $j = 0, 1, 2, 3$

$$b^1_{2j+\upsilon} = \frac{1}{2}(a^k_j + a^k_{j+\upsilon})$$

To obtain $b_0^1, b_2^1, b_4^1, b_6^1$,

$$b_0^1 = \frac{1}{2}(a_0^0 + a_0^0) = a_0^0$$

$$b_2^1 = \frac{1}{2}(a_1^0 + a_1^0) = a_1^0$$

$$b_4^1 = \frac{1}{2}(a_2^0 + a_2^0) = a_2^0$$

$$b_6^1 = \frac{1}{2}(a_3^0 + a_3^0) = a_3^0$$

When $\upsilon = 1$,

For $j = 0, 1, 2$

$$b_{2j+\upsilon}^1 = \frac{1}{2}(a_j^k + a_{j+\upsilon}^k)$$

To obtain b_1^1, b_3^1, b_5^1,

$$b_1^1 = \frac{1}{2}(a_0^0 + a_1^0)$$

$$b_3^1 = \frac{1}{2}(a_1^0 + a_2^0)$$

$$b_5^1 = \frac{1}{2}(a_2^0 + a_3^0)$$

A new control polygon is formed by the new control points, b_0^1, b_2^1, b_4^1, b_6^1, b_1^1, b_3^1, b_5^1 is shown in Figure 2.9.