15
Subdivision Methods

15.1 Introduction

The Bézier curve can be constructed either as a weighted sum of control points or by the process of scaffolding. These are two very different approaches that lead to the same result. Extending these curves to surfaces is a natural process and it results in an elegant theory that is also easy to implement. B-spline curves and surfaces are more powerful than Bézier methods and have long been implemented and employed by many researchers, designers, and digital artists. However, these approaches to surface design are not completely general. They can represent only certain types of surfaces, namely those that are homeomorphic (i.e., similar in the topological sense) to a plane, a cylinder, or a torus.

This limitation has prompted researchers to look for other approaches to curve and surface design, and in 1974, George Chaikin made a breakthrough by showing how to describe curves as the limit of a sequence of subdivision steps (Plate E.1). This approach, which became known as subdivision curves and surfaces, was extended a few years later by Edwin Catmull and Jim Clark [Catmull and Clark 78], by Donald Doo and Malcolm Sabin, and by Charles Loop.

The subdivision approach lacks the beauty and power of Bézier and B-spline methods, which is why it was initially considered exotic and it languished for most of the 1980s and 1990s. Today, however, this situation is changing rapidly. Researchers in fields such as approximation theory and numerical analysis, not just computer graphics, are at work improving and extending subdivision techniques, with the result that new, high-quality three-dimensional commercial software often employs subdivision surfaces as its representation of choice.

Subdivision surfaces may be considered the third approach (after weighted sums and scaffolding) to curve and surface design. They employ the process of refinement (or

D. Salomon, *The Computer Graphics Manual*, Texts in Computer Science,
DOI 10.1007/978-0-85729-886-7_15, © Springer-Verlag London Limited 2011

corner cutting), which is the topic of this chapter. Refinement is a general approach that can produce Bézier curves, B-spline curves, and other types of curves. Its main advantage is that it can easily be extended from curves to surfaces. References [Peters and Reif 08] and [Warren and Weimer 02] are two advanced texts that deal with this topic. Reference [Holmes 10] has a detailed description, examples, and figures of subdivision surfaces.

> The surface subdivision method illustrated here is based on the approach employed in Section 8.8.2 to subdivide a curve. Hence, the reader is advised to read and understand Section 8.8.2 before tackling the material presented here.
>
> —Quoted from Section 8.15.

15.2 Chaikin's Refinement Method

In 1974, George Chaikin came up with the idea of constructing a smooth curve from a small number of control points in several refinement steps. The principle of Chaikin's method is to start with a given set of control points $\mathbf{P}_i$, perform a computation that results in a new set of points $\mathbf{P}_i^1$, and repeat the process, producing more and more sets of points $\mathbf{P}_i^k$. Thus, the original control polygon is successively refined. Table 15.1 shows the notation used.

$$\begin{array}{llll}
\mathbf{P}_0, & \mathbf{P}_1, & \ldots, & \mathbf{P}_n \\
\mathbf{P}_0^1, & \mathbf{P}_1^1, & \ldots, & \mathbf{P}_{n_1}^1 \\
\mathbf{P}_0^2, & \mathbf{P}_1^2, & \ldots, & \mathbf{P}_{n_2}^2 \\
 & & \vdots & \\
\mathbf{P}_0^k, & \mathbf{P}_1^k, & \ldots, & \mathbf{P}_{n_k}^k
\end{array}$$

Table 15.1: Refining Control Points.

Each point $\mathbf{P}_j^k$ is computed as a weighted sum of the points $\mathbf{P}_i^{k-1}$ of the previous iteration. Thus,

$$\mathbf{P}_j^k = \sum_{i=0}^{n_{k-1}} a_{ijk}\mathbf{P}_i^{k-1} = (a_{0jk}, a_{1jk}, \ldots, a_{n_{k-1},jk}) \begin{pmatrix} \mathbf{P}_0^{k-1} \\ \mathbf{P}_1^{k-1} \\ \vdots \\ \mathbf{P}_{n_{k-1}}^{k-1} \end{pmatrix},$$

where a_{ijk} are real coefficients. Notice that each iteration produces a different number $n_k + 1$ of points. If n_k gets smaller with k, then the number of points gets smaller and smaller until a single point is left. An example is the de Casteljau scaffolding construction, an iterative process that produces one point of the Bézier curve. At the other extreme, n_k may get larger with k, producing more points in each iteration. We then stop after a few iterations and draw the curve by drawing short straight segments between the points produced by the last iteration. An example of this case is the Chaikin algorithm, described in example (2) below.

Each iteration can be completely described by its coefficient matrix

$$\begin{aligned}\begin{pmatrix}\mathbf{P}_0^k\\ \mathbf{P}_1^k\\ \vdots\\ \mathbf{P}_{n_k}^k\end{pmatrix} &= \begin{pmatrix} a_{00k} & a_{10k} & \cdots & a_{n_{k-1},0k}\\ a_{01k} & a_{11k} & \cdots & a_{n_{k-1},1k}\\ \vdots & \vdots & & \vdots\\ a_{0,n_k,k} & a_{1,n_k,k} & \cdots & a_{n_{k-1},n_k,k}\end{pmatrix}\begin{pmatrix}\mathbf{P}_0^{k-1}\\ \mathbf{P}_1^{k-1}\\ \vdots\\ \mathbf{P}_{n_{k-1}}^{k-1}\end{pmatrix}\\ &= \mathbf{M}_k\begin{pmatrix}\mathbf{P}_0^{k-1}\\ \mathbf{P}_1^{k-1}\\ \vdots\\ \mathbf{P}_{n_{k-1}}^{k-1}\end{pmatrix},\end{aligned} \tag{15.1}$$

where $\mathbf{M}_k$ has $n_k + 1$ rows and $n_{k-1} + 1$ columns. Since the number of iterations may be large, the number of coefficients a_{ijk} may be huge. In practice, this number is significantly reduced in three ways by: (1) Employing a rule of calculation where most of these coefficients are zero. (2) Selecting coefficients a_{ij} that are independent of k. (3) Using coefficients a_{ik} that are independent of j. Case 2 is called *uniform refinement* and case 3 is termed *stationary refinement*.

Example: (1) This is the de Casteljau scaffolding construction expressed as a refinement process. The rule of refinement is

$$\mathbf{P}_j^{k+1} = 0.5(\mathbf{P}_j^k + \mathbf{P}_{j+1}^k), \tag{15.2}$$

which implies that the a_i coefficients are independent of j and k (this is a stationary uniform refinement method) and are zero except for the two coefficients a_j and a_{j+1}. The a_{ijk}'s therefore depend on i only and are given by

$$a_i = \begin{cases} 0.5, & i = j, j+1,\\ 0, & \text{otherwise.}\end{cases}$$

Since $\mathbf{P}_j^k$ depends on $\mathbf{P}_j^{k-1}$ and $\mathbf{P}_{j+1}^{k-1}$, the largest value for j is $n_k - 1$. This means that each iteration reduces the number of points by one (Figure 15.2a). We start with the $n+1$ points $\mathbf{P}_0$, $\mathbf{P}_1$,..., $\mathbf{P}_n$. The first iteration produces n points, the second iteration produces $n-1$ points, and so on, until iteration n produces one point. That point is located on the Bézier curve $\mathbf{P}(t)$ defined by the $n+1$ original control points. In fact, that point is $\mathbf{P}(0.5)$. If we generalize Equation (15.2) to $\mathbf{P}_j^{k+1} = (1-\alpha)\mathbf{P}_j^k + \alpha\mathbf{P}_{j+1}^k$, then the final point is $\mathbf{P}(\alpha)$. Matrix $\mathbf{M}_k$ of Equation (15.1) is

$$\mathbf{M}_k = \begin{pmatrix} 0.5 & 0.5 & 0 & 0 & \cdots & 0\\ 0 & 0.5 & 0.5 & 0 & \cdots & 0\\ 0 & 0 & 0.5 & 0.5 & \cdots & 0\\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots\\ 0 & 0 & 0 & \cdots & 0.5 & 0.5\end{pmatrix}.$$

It is independent of k and is of order $k \times (k+1)$.

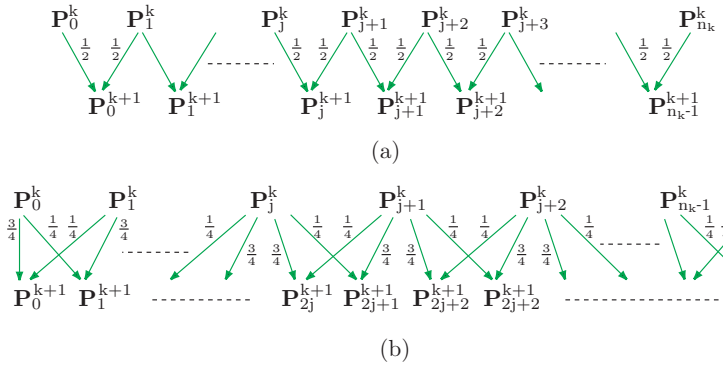


Figure 15.2: (a) de Casteljau Refinement. (b) Chaikin's Method.

Example: (2) We start with the $n+1$ control points $\mathbf{P}_0$, $\mathbf{P}_1$,..., $\mathbf{P}_n$ and apply the rule of refinement

$$\mathbf{P}_{2j}^{k+1} = \frac{3}{4}\mathbf{P}_j^k + \frac{1}{4}\mathbf{P}_{j+1}^k, \qquad \mathbf{P}_{2j+1}^{k+1} = \frac{1}{4}\mathbf{P}_j^k + \frac{3}{4}\mathbf{P}_{j+1}^k. \tag{15.3}$$

(This is illustrated in Figure 15.2b.) The first iteration starts with the original $n+1$ points and produces the $2n$ points $\mathbf{P}_i^1$ shown in Table 15.3. Each subsequent iteration doubles the number of points and brings the points closer to the curve. After k iterations (where k depends on the required precision), the curve is displayed by drawing straight segments between the points produced in the last iteration.

$$\begin{aligned}
\mathbf{P}_0^1 &= \tfrac{3}{4}\mathbf{P}_0 + \tfrac{1}{4}\mathbf{P}_1, & \mathbf{P}_1^1 &= \tfrac{1}{4}\mathbf{P}_0 + \tfrac{3}{4}\mathbf{P}_1,\\
\mathbf{P}_2^1 &= \tfrac{3}{4}\mathbf{P}_1 + \tfrac{1}{4}\mathbf{P}_2, & \mathbf{P}_3^1 &= \tfrac{1}{4}\mathbf{P}_1 + \tfrac{3}{4}\mathbf{P}_2,\\
\mathbf{P}_4^1 &= \tfrac{3}{4}\mathbf{P}_2 + \tfrac{1}{4}\mathbf{P}_3, & \mathbf{P}_5^1 &= \tfrac{1}{4}\mathbf{P}_2 + \tfrac{3}{4}\mathbf{P}_3,\\
&\vdots & &\vdots\\
\mathbf{P}_{2n-2}^1 &= \tfrac{3}{4}\mathbf{P}_{n-1} + \tfrac{1}{4}\mathbf{P}_n, & \mathbf{P}_{2n-1}^1 &= \tfrac{1}{4}\mathbf{P}_{n-1} + \tfrac{3}{4}\mathbf{P}_n.
\end{aligned}$$

Table 15.3: First Iteration of Chaikin's Algorithm.

This method is due to George Chaikin ([Chaikin 74] and [Riesenfeld 75]) and has a simple geometric interpretation, which is illustrated in Figure 15.5. Part (a) of the figure shows a control polygon made of five points. The rule of refinement is: Take a segment $\mathbf{P}_i\mathbf{P}_{i+1}$ of the control polygon and place two new points $\mathbf{Q}_i$ and $\mathbf{R}_i$ at distances from $\mathbf{P}_i$ of 1/4 and 3/4 the segment's size, respectively (Figure 15.5b, which justifies the term "corner cutting"). The new points are therefore given by

$$\mathbf{Q}_i = \frac{3}{4}\mathbf{P}_i + \frac{1}{4}\mathbf{P}_{i+1}, \quad \mathbf{R}_i = \frac{1}{4}\mathbf{P}_i + \frac{3}{4}\mathbf{P}_{i+1}.$$

This is repeated for all the polygon segments. If we start with $n+1$ control points defining a control polygon with n sides, we end up with $2n$ new points $\mathbf{Q}_i$ and $\mathbf{R}_i$. They should now be connected to form a new control polygon with $2n-1$ sides. As this process is repeated (Figure 15.5c), the control polygons get closer to the smooth curve shown in Figure 15.5d. This figure also shows that the midpoint of any segment of the control polygon is a point on the Chaikin curve. In fact, the midpoint of any segment generated at any stage of the refinement is a point on the Chaikin curve.

◇ **Exercise 15.1:** Is this curve a Bézier curve?

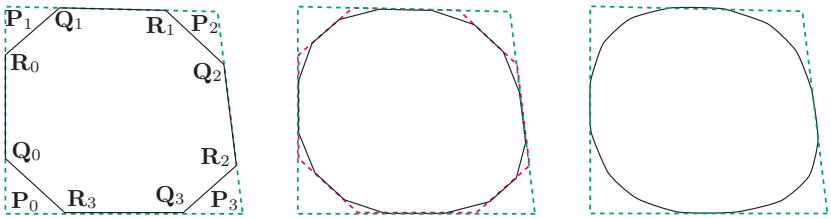


Figure 15.4: Chaikin's Algorithm for a Closed Curve.

This algorithm works for closed curves too. The only modification needed is to connect the last point $\mathbf{P}_n$ to the first one $\mathbf{P}_0$ and compute the two auxiliary points $\mathbf{Q}_n$ and $\mathbf{R}_n$. This can be done in a natural way if we copy $\mathbf{P}_0$ and name the duplicate $\mathbf{P}_{n+1}$. Figure 15.4 shows three instances in the construction of such a curve. Again, we see that the midpoint of any segment of the control polygon is a point on the closed Chaikin curve.

To identify the kind of curve that Chaikin's algorithm produces, let's consider the control polygon defined by the three points $\mathbf{P}_0$, $\mathbf{P}_1 = \mathbf{B}$, and $\mathbf{P}_2$ (Figure 15.6). Let $\mathbf{A}$ and $\mathbf{C}$ be the midpoints of segments $\mathbf{P}_0\mathbf{P}_1$ and $\mathbf{P}_1\mathbf{P}_2$, respectively, and let point $\mathbf{P}$ be the midpoint of points $\mathbf{M}_{ab} = (\mathbf{A}+\mathbf{B})/2$ and $\mathbf{M}_{bc} = (\mathbf{B}+\mathbf{C})/2$. This point has the following properties:

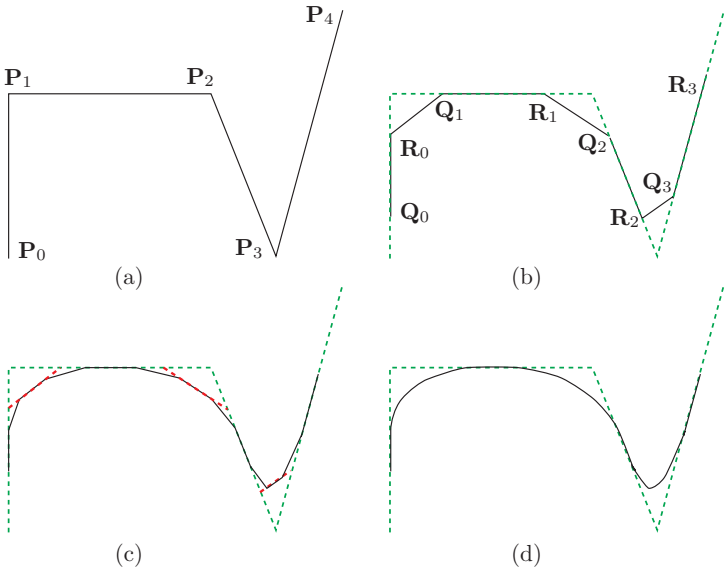
1. It is located on the Bézier curve defined by points $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ because it's been constructed using the de Casteljau scaffolding process.

2. It is located on the Chaikin curve defined by points $\mathbf{P}_0$, $\mathbf{P}_1$, and $\mathbf{P}_2$. This is because points $\mathbf{M}_{ab}$ and $\mathbf{M}_{bc}$ are the points constructed by the first step of Chaikin's algorithm and we already know that the midpoint of any Chaikin segment is a point on the Chaikin curve.

◇ **Exercise 15.2:** Show that points $\mathbf{M}_{ab}$ and $\mathbf{M}_{bc}$ are the points constructed by the first step of Chaikin's algorithm.

The second refinement step produces the two midpoints, $\mathbf{P}_{01}$ and $\mathbf{P}_{11}$ (Figure 15.6) using the recursive procedures

$$\begin{aligned} &\mathbf{B} \leftarrow (\mathbf{A}+\mathbf{B})/2, \quad \mathbf{C} \leftarrow \mathbf{P}, \quad \mathbf{P}_{01} \leftarrow (\mathbf{A}+2\mathbf{B}+\mathbf{C})/4, \\ &\mathbf{A} \leftarrow \mathbf{P}, \quad \mathbf{B} \leftarrow (\mathbf{B}+\mathbf{C})/2, \quad \mathbf{P}_{11} \leftarrow (\mathbf{A}+2\mathbf{B}+\mathbf{C})/4. \end{aligned}$$




```
(* Chaikin algorithm for a control polygon *)
n=4;
(*p={p0,p1,p2,p3,p4,p5};*)
p={{0,0},{0,4},{3,4},{4,0},{6,6}};
Show[Graphics[Line[p]]]
q=Table[If[OddQ[i],
(*then*){(3p[[i]]+p[[i+1]])/4,(p[[i]]+3p[[i+1]])/4},
(*else*){(3p[[i]]+p[[i+1]])/4,(p[[i]]+3p[[i+1]])/4}],{i,1,n}];
q=Flatten[q,1]
Show[Graphics[{Green,AbsoluteDashing[{5,2}],
 Line[p]}],Graphics[Line[q]],PlotRange->All]
r=Table[If[OddQ[i],
 (*then*){(3q[[i]]+q[[i+1]])/4,(q[[i]]+3q[[i+1]])/4},
 (*else*){(3q[[i]]+q[[i+1]])/4,(q[[i]]+3q[[i+1]])/4}],{i,1,2n-1}];
r=Flatten[r,1]
Show[Graphics[{Green,AbsoluteDashing[{2,2}],
 Line[p]}],Graphics[Line[r]],PlotRange->All]
```

Figure 15.5: Chaikin's Algorithm for a Control Polygon.

An argument similar to the previous one shows that these two points are also located on the quadratic Bézier curve defined by $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ as well as on the Chaikin curve defined by $\mathbf{P}_0$, $\mathbf{P}_1$, and $\mathbf{P}_2$. Applying this argument to all the points generated by the refinement steps shows that they are located on both curves, which proves that the Chaikin curve defined by $\mathbf{P}_0$, $\mathbf{P}_1$, and $\mathbf{P}_2$ is identical to the quadratic Bézier curve defined by $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$. This Bézier curve is

$$\mathbf{P}(t) = (1-t)^2\mathbf{A} + 2t(1-t)\mathbf{B} + t^2\mathbf{C},$$

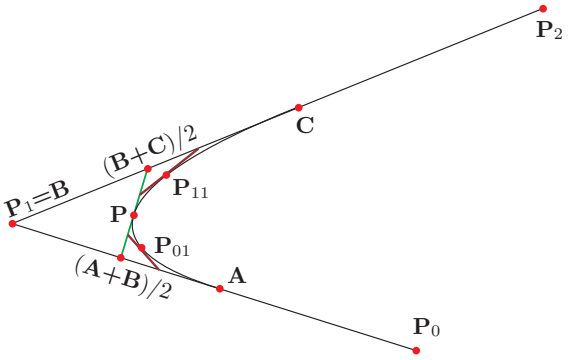


Figure 15.6: Points on the Chaikin Curve.

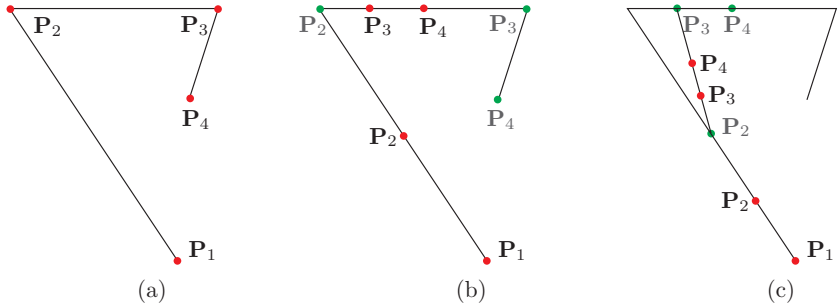
and it is easy to express in terms of the original control points $\mathbf{P}_i$

$$\begin{aligned}
\mathbf{P}(t) &= (t^2,t,1)\begin{pmatrix}1&-2&1\\-2&2&0\\1&0&0\end{pmatrix}\begin{pmatrix}\mathbf{A}\\\mathbf{B}\\\mathbf{C}\end{pmatrix}\\
&= (t^2,t,1)\begin{pmatrix}1&-2&1\\-2&2&0\\1&0&0\end{pmatrix}\begin{pmatrix}(\mathbf{P}_0+\mathbf{P}_1)/2\\\mathbf{P}_1\\(\mathbf{P}_1+\mathbf{P}_2)/2\end{pmatrix}\\
&= (t^2,t,1)\begin{pmatrix}1&-2&1\\-2&2&0\\1&0&0\end{pmatrix}\begin{pmatrix}1/2&1/2&0\\0&1&0\\0&1/2&1/2\end{pmatrix}\begin{pmatrix}\mathbf{P}_0\\\mathbf{P}_1\\\mathbf{P}_2\end{pmatrix}\\
&= \frac{1}{2}(t^2,t,1)\begin{pmatrix}1&-2&1\\-2&2&0\\1&1&0\end{pmatrix}\begin{pmatrix}\mathbf{P}_0\\\mathbf{P}_1\\\mathbf{P}_2\end{pmatrix}. \qquad (15.4)
\end{aligned}$$

The result is the quadratic B-spline curve segment, Equation (14.6).

We therefore conclude that the curve produced by Chaikin's algorithm is not a new type of curve but the quadratic B-spline for points $\mathbf{P}_0$, $\mathbf{P}_1$, and $\mathbf{P}_2$. This fact lets us see the B-spline in a new light and it also shows a relation between the quadratic Bézier and B-spline curves.

◇ **Exercise 15.3:** (Easy). State this relation.



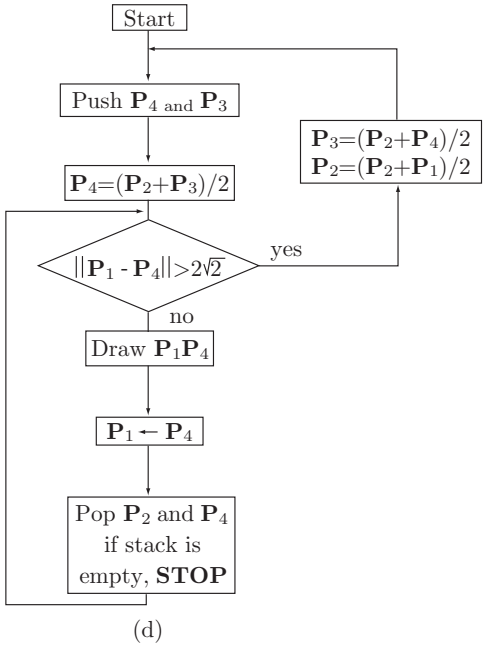


Figure 15.7: The Original Chaikin Algorithm.

The Original Chaikin Algorithm

The description of Chaikin's algorithm in this section differs from that originally proposed by George Chaikin. Here is the original description of the method, as it appears in [Chaikin 74]. Start with four points $\mathbf{P}_1$ through $\mathbf{P}_4$ (Figure 15.7a). Points $\mathbf{P}_4$ and $\mathbf{P}_3$ are pushed into a stack and a new $\mathbf{P}_4 = (\mathbf{P}_2 + \mathbf{P}_3)/2$ is constructed. Points $\mathbf{P}_1$ and $\mathbf{P}_4$ are now compared. If their distance is greater than or equal to three pixels, then points $\mathbf{P}_2$ and $\mathbf{P}_3$ are recomputed according to

$$\mathbf{P}_3 = (\mathbf{P}_2 + \mathbf{P}_4)/2, \quad \mathbf{P}_2 = (\mathbf{P}_2 + \mathbf{P}_1)/2$$

(Figure 15.7b,c), points $\mathbf{P}_4$ and $\mathbf{P}_3$ are pushed into the stack, point $\mathbf{P}_4$ is recalculated, and the distance between $\mathbf{P}_1$ and $\mathbf{P}_4$ is checked. This is repeated until the distance becomes smaller than three pixels, in which case the short segment $\mathbf{P}_1\mathbf{P}_4$ is drawn, point $\mathbf{P}_4$ is renamed $\mathbf{P}_1$, the stack is popped twice and the resulting points are named $\mathbf{P}_2$ and $\mathbf{P}_4$, and the distance $\mathbf{P}_1\mathbf{P}_4$ is checked again. The process terminates when the stack is empty. Figure 15.7d is a flowchart of this algorithm.

◇ **Exercise 15.4:** Why compare the distance to three pixels and not to two?

15.3 Quadratic Uniform B-Spline by Subdivision

The uniform B-spline for a group of $n+1$ control points can be constructed as a set of short segments, each a quadratic polynomial based on three control points. This section shows how Chaikin's algorithm (Section 15.2) can be applied to construct such a curve. We divide the original $n+1$ control points into $n-1$ overlapping groups of three points each, and use each group to calculate four new points. The groups are

$$\mathbf{P}_0\mathbf{P}_1\mathbf{P}_2, \quad \mathbf{P}_1\mathbf{P}_2\mathbf{P}_3, \ldots, \quad \mathbf{P}_{n-2}\mathbf{P}_{n-1}\mathbf{P}_n.$$

Subdividing the first group is done by

$$\begin{pmatrix} \mathbf{P}_0^1 \\ \mathbf{P}_1^1 \\ \mathbf{P}_2^1 \\ \mathbf{P}_3^1 \end{pmatrix} = \frac{1}{4}\begin{pmatrix} 3 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 3 & 1 \\ 0 & 1 & 3 \end{pmatrix}\begin{pmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \end{pmatrix} = \begin{pmatrix} \frac{3}{4}\mathbf{P}_0 + \frac{1}{4}\mathbf{P}_1 \\ \frac{1}{4}\mathbf{P}_0 + \frac{3}{4}\mathbf{P}_1 \\ \frac{3}{4}\mathbf{P}_1 + \frac{1}{4}\mathbf{P}_2 \\ \frac{1}{4}\mathbf{P}_1 + \frac{3}{4}\mathbf{P}_2 \end{pmatrix},$$

and it yields the four new points $\mathbf{P}_0^1$, $\mathbf{P}_1^1$, $\mathbf{P}_2^1$, and $\mathbf{P}_3^1$. Subdividing the second group is done similarly and yields the four points $\mathbf{P}_2^1$, $\mathbf{P}_3^1$, $\mathbf{P}_4^1$, and $\mathbf{P}_5^1$, of which only the last two are new. Each subsequent group also yields two new points when subdivided. The process is then repeated on the $2n$ segments defined by the $2n+2$ new points $\mathbf{P}_i^1$, yielding

$4n + 4$ points $\mathbf{P}_i^2$. These points, in turn, define $4n + 2$ segments. When the number of points is large enough, the curve can be drawn by connecting each pair of adjacent points with a straight segment.

It can be shown (see Page 809) that the curve obtained this way is the quadratic uniform B-spline, Equation (14.6).

15.4 Cubic Uniform B-Spline by Subdivision

The approach to constructing cubic B-splines by subdivision is similar to that of Section 15.3. We show how Chaikin's methods (Section 15.2) can be applied to the construction of a cubic uniform B-spline for a set of $n+1$ control points $\mathbf{P}_i$. The points are divided into overlapping groups of four points each, and each group is used to compute, by refinement, a PC that becomes a segment in the entire curve. These cubic segments have C^2 continuity. Since refinement is an iterative process, we denote the control points obtained in the kth subdivision step by $\mathbf{P}_i^k$. Thus, it makes sense to denote the original control points by $\mathbf{P}_i^0$. They are divided into the overlapping groups

$$\mathbf{P}_0^0\mathbf{P}_1^0\mathbf{P}_2^0\mathbf{P}_3^0, \quad \mathbf{P}_1^0\mathbf{P}_2^0\mathbf{P}_3^0\mathbf{P}_4^0, \ldots, \quad \mathbf{P}_{n-3}^0\mathbf{P}_{n-2}^0\mathbf{P}_{n-1}^0\mathbf{P}_n^0.$$

Figure 15.8a illustrates the refinement process that leads from the group of four control points $\mathbf{P}_0^0\mathbf{P}_1^0\mathbf{P}_2^0\mathbf{P}_3^0$ to a segment of a cubic uniform B-spline. The treatment for the other groups is similar. The figure shows the positions of the five iteration-1 points $\mathbf{P}_i^1$ and the seven points $\mathbf{P}_i^2$ resulting from iteration 2. The first refinement step computes the five points $\mathbf{P}_0^1\mathbf{P}_1^1\mathbf{P}_2^1\mathbf{P}_3^1\mathbf{P}_4^1$ as follows:

1. Each of the three points with even subscripts $\mathbf{P}_0^1\mathbf{P}_2^1\mathbf{P}_4^1$ (termed the *edge* points) is located at the center of a segment delimited by two of the original control points. Thus, $\mathbf{P}_0^1$ is located midway between $\mathbf{P}_0^0$ and $\mathbf{P}_1^0$.
2. Each of the two points with odd subscripts $\mathbf{P}_1^1$ and $\mathbf{P}_3^1$ (termed the *vertex* points) is located at the center of a segment whose endpoints are located at the centers of two segments delimited by two new edge points and one original control point. Thus, $\mathbf{P}_1^1$ is located at the center of the segment whose endpoints are located at the centers of the two segments delimited by the three points $\mathbf{P}_0^1$, $\mathbf{P}_1^0$ and $\mathbf{P}_2^1$.

The five points produced by the first refinement step can be expressed in terms of the four original control points by

$$\begin{pmatrix} \mathbf{P}_0^1 \\ \mathbf{P}_1^1 \\ \mathbf{P}_2^1 \\ \mathbf{P}_3^1 \\ \mathbf{P}_4^1 \end{pmatrix} = \begin{pmatrix} \frac{1}{2}(\mathbf{P}_0^0 + \mathbf{P}_1^0) \\ \frac{1}{8}(\mathbf{P}_0^0 + 6\mathbf{P}_1^0 + \mathbf{P}_2^0) \\ \frac{1}{2}(\mathbf{P}_1^0 + \mathbf{P}_2^0) \\ \frac{1}{8}(\mathbf{P}_1^0 + 6\mathbf{P}_2^0 + \mathbf{P}_3^0) \\ \frac{1}{2}(\mathbf{P}_2^0 + \mathbf{P}_3^0) \end{pmatrix} = \frac{1}{8}\begin{pmatrix} 4 & 4 & 0 & 0 \\ 1 & 6 & 1 & 0 \\ 0 & 4 & 4 & 0 \\ 0 & 1 & 6 & 1 \\ 0 & 0 & 4 & 4 \end{pmatrix} \begin{pmatrix} \mathbf{P}_0^0 \\ \mathbf{P}_1^0 \\ \mathbf{P}_2^0 \\ \mathbf{P}_3^0 \end{pmatrix}.$$

Each of the new points $\mathbf{P}_i^1$ is computed from either two or three of the points $\mathbf{P}_j^0$. The five new points are then divided into two overlapping groups $\mathbf{P}_0^1\mathbf{P}_1^1\mathbf{P}_2^1\mathbf{P}_3^1$ and $\mathbf{P}_1^1\mathbf{P}_2^1\mathbf{P}_3^1\mathbf{P}_4^1$

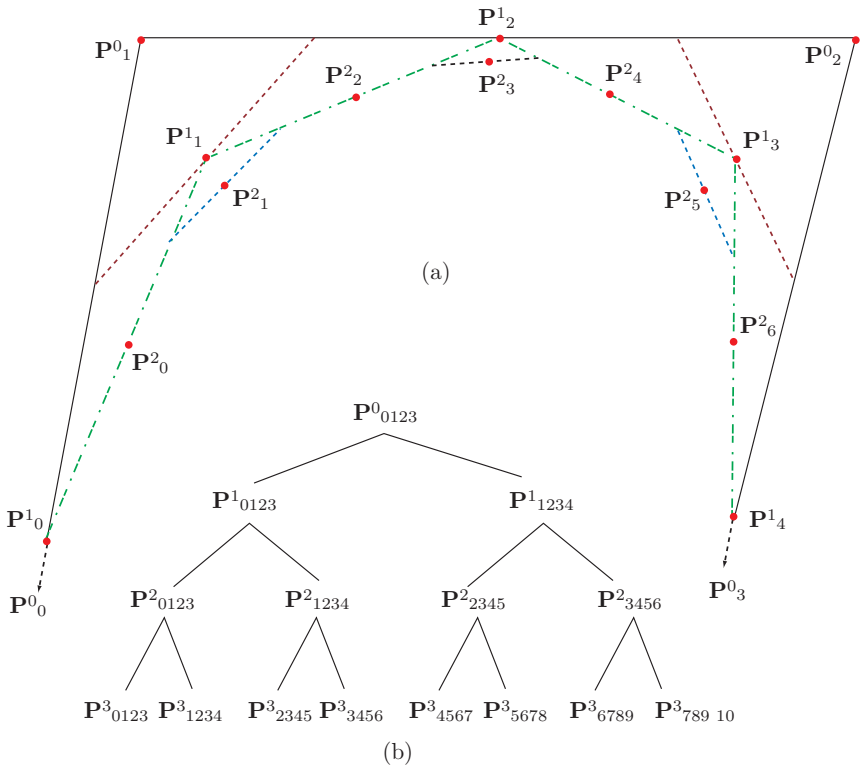


Figure 15.8: (a) The First Two Refinement Steps. (b) Groups After Three Steps.

of four points each, and the second subdivision step is applied to each group to produce five new points denoted by $\mathbf{P}_i^2$. Some of the $\mathbf{P}_i^2$ points, however, are identical, so this second step produces a total of seven distinct points. Figure 15.8b shows the points produced by the first three iterations of the refinement process and how each group of four points $\mathbf{P}_i^k$ produces two overlapping groups of four new points $\mathbf{P}_i^{k+1}$ each. The compact notation $\mathbf{P}_{0123}^3$ stands for a group of four points. It is easy to see that iteration k produces 2^k overlapping groups of four points each, for a total of $4+(2^k-1)=3+2^k$ distinct points. Thus, iteration 0 (the original control points) consists of $3+2^0=4$ points, and iterations 1, 2, 3, and 4 produce 5, 7, 11, and 19 points, respectively.

Since each point produced in step k is computed from either two or three points of step $k-1$, it is convenient to express a new triplet of points $\mathbf{P}_i^k\mathbf{P}_{i+1}^k\mathbf{P}_{i+2}^k$ as a function

of a triplet $\mathbf{P}_j^{k-1}\mathbf{P}_{j+1}^{k-1}\mathbf{P}_{j+2}^{k-1}$. We illustrate this relation for $k=1$

$$\begin{pmatrix}\mathbf{P}_0^1\\ \mathbf{P}_1^1\\ \mathbf{P}_2^1\end{pmatrix}=\mathbf{A}\begin{pmatrix}\mathbf{P}_0^0\\ \mathbf{P}_1^0\\ \mathbf{P}_2^0\end{pmatrix},\quad \begin{pmatrix}\mathbf{P}_2^1\\ \mathbf{P}_3^1\\ \mathbf{P}_4^1\end{pmatrix}=\mathbf{A}\begin{pmatrix}\mathbf{P}_1^0\\ \mathbf{P}_2^0\\ \mathbf{P}_3^0\end{pmatrix},\quad \text{where}\quad \mathbf{A}=\frac{1}{8}\begin{pmatrix}4&4&0\\ 1&6&1\\ 0&4&4\end{pmatrix},$$

or, using compact notation $\mathbf{P}_{012}^1=\mathbf{A}\mathbf{P}_{012}^0$ and $\mathbf{P}_{234}^1=\mathbf{A}\mathbf{P}_{123}^0$. In general $\mathbf{P}_{i\,i+1\,i+2}^1=\mathbf{A}\mathbf{P}_{j\,j+1\,j+2}^0$ for even values of i and for $j=i,i-1$.

For $k=2$, the computation of the seven points $\mathbf{P}_i^2$ can be summarized by the three overlapping triplets $\mathbf{P}_{012}^2=\mathbf{A}\mathbf{P}_{012}^1$, $\mathbf{P}_{234}^2=\mathbf{A}\mathbf{P}_{123}^1$, and $\mathbf{P}_{456}^2=\mathbf{A}\mathbf{P}_{234}^1$, or in general $\mathbf{P}_{i\,i+1\,i+2}^2=\mathbf{A}\mathbf{P}_{j\,j+1\,j+2}^1$, for even values of i and for $j=i$, $i-1$, and $i-2$. For $k=3$, the calculation of the 11 points $\mathbf{P}_i^3$ is summarized by the five triplets $\mathbf{P}_{i\,i+1\,i+2}^3=\mathbf{A}\mathbf{P}_{j\,j+1\,j+2}^2$, where i is even and $j=i$, $i-1$, $i-2$, and $i-3$. In general, the computation of the $3+2^k$ points of step k can be summarized by the $2^{k-1}+1$ triplets $\mathbf{P}_{i\,i+1\,i+2}^k=\mathbf{A}\mathbf{P}_{j\,j+1\,j+2}^{k-1}$ where i is even and j goes through the values i, $i-1$ and so on, down to $i-(2^{k-1}-1)$.

◇ **Exercise 15.5:** Write each of the nine triplets $\mathbf{P}_{i\,i+1\,i+2}^4$ (for even values of i) in terms of a triplet $\mathbf{P}_{j\,j+1\,j+2}^3$.

Because of the repeated use of matrix $\mathbf{A}$, most triplets produced in step k can be expressed in terms of triplets produced in earlier steps. For example, the trio of points $\mathbf{P}_{012}^3$ can be written as $\mathbf{A}\mathbf{P}_{012}^2=\mathbf{A}^2\mathbf{P}_{012}^1=\mathbf{A}^3\mathbf{P}_{012}^0$, the triplet $\mathbf{P}_{234}^3$ equals $\mathbf{A}\mathbf{P}_{123}^2$, and $\mathbf{P}_{456}^3$ can be written as $\mathbf{A}\mathbf{P}_{234}^2=\mathbf{A}^2\mathbf{P}_{123}^1$. (Note that for the triplet on the left-hand side, the first subscript is always even, but the first subscript of the triplet on the right can be even or odd.) These relations point the way to moving forward from an earlier triplet to a later one. If we start, say, with the triplet $\mathbf{P}_{123}^1$, we can easily compute the triplets $\mathbf{P}_{234}^2$, $\mathbf{P}_{456}^3$, $\mathbf{P}_{89\,10}^4$, $\mathbf{P}_{16\,17\,18}^5$, and so on by multiplying the three points $\mathbf{P}_{123}^1$ by powers of $\mathbf{A}$. We can use this method to leapfrog across many recursion steps and proceed, in one step, from any triplet $\mathbf{P}_{i\,i+1\,i+2}^k$ to a triplet many subdivision steps later! In the limit, this can be written $\lim_{k\to\infty}\mathbf{P}_{i\,i+1\,i+2}^k=\mathbf{A}^\infty\mathbf{P}_{i\,i+1\,i+2}^k$, where $\mathbf{A}^\infty$ denotes $\lim_{k\to\infty}\mathbf{A}^k$. Any triplet $\mathbf{P}_{i\,i+1\,i+2}^k$ is an approximation to the ideal B-spline curve, but the limit $\lim_{k\to\infty}\mathbf{P}_{i\,i+1\,i+2}^k$ converges to a point on the actual curve.

The problem is therefore to find the limit of $\mathbf{A}^k$ as k approaches infinity, and this can easily be done with the help of the following theorem (see any text on matrices or linear algebra for the proof and for more information on eigenvalues and eigenvectors):

Theorem: If $\mathbf{A}$ is an $n\times n$ matrix for which there exist n linearly-independent eigenvectors, then $\mathbf{A}$ has the form $\mathbf{Q}\Lambda\mathbf{Q}^{-1}$, where $\mathbf{Q}$ is the matrix whose columns are the n eigenvectors and Λ is the diagonal matrix whose diagonal elements are the eigenvalues of $\mathbf{A}$.

This theorem implies that $\mathbf{A}^2=\mathbf{Q}\Lambda\mathbf{Q}^{-1}\mathbf{Q}\Lambda\mathbf{Q}^{-1}=\mathbf{Q}\Lambda^2\mathbf{Q}^{-1}$, and in general $\mathbf{A}^k=\mathbf{Q}\Lambda^k\mathbf{Q}^{-1}$. Following this theorem, we can write our matrix $\mathbf{A}$ (after its eigenvalues and a set of linearly independent eigenvectors have been computed with appropriate software) as

$$\mathbf{A}=\begin{pmatrix}1&-1&1\\ -1/2&0&1\\ 1&1&1\end{pmatrix}\begin{pmatrix}1/4&0&0\\ 0&1/2&0\\ 0&0&1\end{pmatrix}\begin{pmatrix}1/3&-2/3&1/3\\ -1/2&0&1/2\\ 1/6&2/3&1/6\end{pmatrix}.$$

Since matrix Λ is diagonal, we have

$$\lim_{k\to\infty} \Lambda^k = \lim_{k\to\infty} \begin{pmatrix} (1/4)^k & 0 & 0 \\ 0 & (1/2)^k & 0 \\ 0 & 0 & 1^k \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The limit $\mathbf{A}^\infty$ is therefore

$$\begin{pmatrix} 1 & -1 & 1 \\ -1/2 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1/3 & -2/3 & 1/3 \\ -1/2 & 0 & 1/2 \\ 1/6 & 2/3 & 1/6 \end{pmatrix} = \frac{1}{6} \begin{pmatrix} 1 & 4 & 1 \\ 1 & 4 & 1 \\ 1 & 4 & 1 \end{pmatrix},$$

so we end up with the limits

$$\begin{aligned} \lim_{k\to\infty} \mathbf{P}^k_{i\,i+1\,i+2} &= \frac{1}{6} \begin{pmatrix} 1 & 4 & 1 \\ 1 & 4 & 1 \\ 1 & 4 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{P}^k_i \\ \mathbf{P}^k_{i+1} \\ \mathbf{P}^k_{i+2} \end{pmatrix} \stackrel{\text{def}}{=} \frac{1}{6}(1,4,1) \begin{pmatrix} \mathbf{P}^k_i \\ \mathbf{P}^k_{i+1} \\ \mathbf{P}^k_{i+2} \end{pmatrix} \\ &= \frac{1}{6}(\mathbf{P}^k_i + 4\mathbf{P}^k_{i+1} + \mathbf{P}^k_{i+2}), \end{aligned}$$

where k is any nonnegative integer. Notice that the three points of the triplet converge to the same point on the B-spline curve.

To summarize, we can (1) select four control points $\mathbf{P}^0_{0123}$, (2) select a value k and perform k refinement steps, (3) select a value i and a triplet $\mathbf{P}^k_{i\,i+1\,i+2}$, and (4) compute $(\mathbf{P}^k_i + 4\mathbf{P}^k_{i+1} + \mathbf{P}^k_{i+2})/6$. This will be a point on the cubic B-spline curve segment defined by the four original control points. To show that this is so, we can express each of the three points $\mathbf{P}^k_{i\,i+1\,i+2}$ in terms of the original control points $\mathbf{P}^0_{0123}$, and compare the result with the general cubic B-spline segment, Equation (14.11). Here are some examples.

Example: (1) We start with $k = 0$ and $i = 0$. The initial triplet is therefore $\mathbf{P}^0_{012}$.

$$\lim_{k\to\infty} \mathbf{P}^0_{012} = \frac{1}{6}(1,4,1) \begin{pmatrix} \mathbf{P}^0_0 \\ \mathbf{P}^0_1 \\ \mathbf{P}^0_2 \end{pmatrix} = \frac{1}{6}(\mathbf{P}^0_0 + 4\mathbf{P}^0_1 + \mathbf{P}^0_2),$$

which is the initial point $\mathbf{P}(0)$ of the B-spline segment, as can be seen from Equation (14.11).

Example: (2) The values $k = 0$ and $i = 1$ specify the triplet $\mathbf{P}^0_{123}$ (notice that i does not have to be even).

$$\lim_{k\to\infty} \mathbf{P}^0_{123} = \frac{1}{6}(1,4,1) \begin{pmatrix} \mathbf{P}^0_1 \\ \mathbf{P}^0_2 \\ \mathbf{P}^0_3 \end{pmatrix} = \frac{1}{6}(\mathbf{P}^0_1 + 4\mathbf{P}^0_2 + \mathbf{P}^0_3),$$

which is the final point $\mathbf{P}(1)$ of the B-spline segment, as can be seen from the same equation.

Example: (3) We perform one refinement step and select the triplet $\mathbf{P}^1_{123}$ specified by $k = 1$ and $i = 1$. When this triplet is expressed in terms of the control points $\mathbf{P}^0_i$, the result is

$$\begin{aligned}\lim_{k\to\infty} \mathbf{P}^1_{123} &= \frac{1}{6}(\mathbf{P}^1_1 + 4\mathbf{P}^1_2 + \mathbf{P}^1_3) \\ &= \frac{1}{6}\left(\frac{1}{8}(\mathbf{P}^0_0 + 6\mathbf{P}^0_1 + \mathbf{P}^0_2) + \frac{4}{2}(\mathbf{P}^0_1 + \mathbf{P}^0_2) + \frac{1}{8}(\mathbf{P}^0_1 + 6\mathbf{P}^0_2 + \mathbf{P}^0_3)\right) \\ &= \frac{1}{48}(\mathbf{P}^0_0 + 23\mathbf{P}^0_1 + 23\mathbf{P}^0_2 + \mathbf{P}^0_3).\end{aligned}$$

Equation (14.11) tells us that this is the midpoint $\mathbf{P}(1/2)$ of the curve segment.

◇ **Exercise 15.6:** Select $k = 3$ and $i = 6$ and compute the point on the cubic B-spline curve segment obtained from these values at the limit of subdivision.

15.5 Biquadratic B-Spline Surface by Subdivision

The method of subdivision has been introduced in Section 15.2, where Chaikin's algorithm for curves is discussed. Generating the quadratic B-spline curve by subdivision is described in Section 15.3. This material should be reviewed before reading ahead. The technique of subdivision can be extended to surfaces that are defined by a mesh of control points. We use the biquadratic B-spline surface patch as an example. Such a patch is constructed by Equation (14.49) from a grid of 3×3 control points $\mathbf{P}_{ij}$. We denote this patch by BSP and the original points by $\mathbf{P}^0_{ij}$. The principle of constructing a BSP by subdivision is to find a way to subdivide the mesh of original points into a finer mesh with more points $\mathbf{P}^1_{ij}$ and, as a result, with more subpatches. If this is done right, the new control points $\mathbf{P}^1_{ij}$ will be closer to the ideal BSP surface than the original ones. When this process is repeated, it results in more and more control points $\mathbf{P}^k_{ij}$ that get closer and closer to the BSP. At the limit, we end up with infinitely many points that lie on the surface. In practice, we stop the subdivision process after a finite number k of steps and display the surface as a wireframe by connecting points $\mathbf{P}^k_{ij}$ with straight segments.

The refinement rule for a BSP $\mathbf{P}(u, w)$ employs reparametrization to calculate four new patches $\mathbf{Q}(u, w)$. The technique of reparametrization was introduced in Section 13.10 for curves and has been extended for Bézier surface patches in Section 13.27. It can easily be modified for the biquadratic B-spline surface by rewriting Equation (13.62) in the form

$$\begin{aligned}\mathbf{Q}(u, w) &= \mathbf{P}([b-a]u + a, [d-c]w + c) \\ &= \big(([b-a]u + a)^2, ([b-a]u + a), 1\big)\mathbf{M}\cdot\mathbf{P}\cdot\mathbf{M}^{-1}\begin{pmatrix}([d-c]w + c)^2\\ [d-c]w + c\\ 1\end{pmatrix} \\ &= (u^2, u, 1)\mathbf{A}_{ab}\mathbf{M}\cdot\mathbf{P}\cdot\mathbf{M}^T\cdot\mathbf{A}^T_{cd}(w^2, w, 1)^T\end{aligned}$$

$$
\begin{aligned}
&= (u^2, u, 1)\mathbf{M}(\mathbf{M}^{-1} \cdot \mathbf{A}_{ab} \cdot \mathbf{M})\mathbf{P}(\mathbf{M}^T \cdot \mathbf{A}_{cd}^T \cdot (\mathbf{M}^T)^{-1})\mathbf{M}^T(w^2, w, 1)^T \\
&= (u^2, u, 1)\mathbf{M} \cdot \mathbf{B}_{ab} \cdot \mathbf{P} \cdot \mathbf{B}_{cd}^T \cdot \mathbf{M}^T(w^2, w, 1)^T \\
&= (u^2, u, 1)\mathbf{M} \cdot \mathbf{Q} \cdot \mathbf{M}^T(w^2, w, 1)^T,
\end{aligned}
$$

where

$$
\mathbf{M} = \frac{1}{2}\begin{pmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{pmatrix}, \quad \mathbf{A}_{ab} = \begin{pmatrix} (b-a)^2 & 0 & 0 \\ 2a(b-a) & b-a & 0 \\ a^2 & a & 1 \end{pmatrix}
$$

($\mathbf{M}$ is the basis matrix for the biquadratic B-spline surface),

$$
\mathbf{P} = \begin{pmatrix} \mathbf{P}_{00} & \mathbf{P}_{01} & \mathbf{P}_{02} \\ \mathbf{P}_{10} & \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{20} & \mathbf{P}_{21} & \mathbf{P}_{22} \end{pmatrix},
$$

$$
\begin{aligned}
\mathbf{B}_{ab} &= \mathbf{M}^{-1} \cdot \mathbf{A}_{ab} \cdot \mathbf{M} \\
&= \begin{pmatrix} ((1-a)(1-2a+b))/2 & (1+3a-4a^2-b+2ab)/2 & a^2-(ab)/2 \\ 1/2-a/2-b/2+(ab)/2 & (1+a+b-2ab)/2 & (ab)/2 \\ ((1+a-2b)(1-b))/2 & (1-a+3b+2ab-4b^2)/2 & -(ab)/2+b^2 \end{pmatrix},
\end{aligned}
$$

$$
\begin{aligned}
&\mathbf{B}_{cd}^T = \mathbf{M}^T \cdot \mathbf{A}_{cd}^T \cdot (\mathbf{M}^T)^{-1} \\
&= \begin{pmatrix} ((1-c)(1-2c+d))/2 & 1/2-c/2-d/2+(cd)/2 & ((1+c-2d)(1-d))/2 \\ (1+3c-4c^2-d+2cd)/2 & (1+c+d-2cd)/2 & (1-c+3d+2cd-4d^2)/2 \\ c^2-(cd)/2 & (cd)/2 & -(cd)/2+d^2 \end{pmatrix},
\end{aligned}
$$

and

$$
\mathbf{Q} = \mathbf{B}_{ab} \cdot \mathbf{P} \cdot \mathbf{B}_{cd}^T. \tag{15.5}
$$

The elements of $\mathbf{Q}$ depend on the four parameters a, b, c, and d, and on the $\mathbf{P}_{ij}$'s. Once the four parameters are known, matrix $\mathbf{Q}$ is easy to calculate symbolically with appropriate mathematical software.

The rule for subdividing a biquadratic B-spline surface patch $\mathbf{P}(u, w)$ is as follows: Call the original surface patch the subdivision step-0 surface. Use reparametrization to calculate the four step-1 surface patches defined by the following sets of parameters:

$$
\begin{aligned}
&a = 0,\ b = 0.5,\ c = 0,\ d = 0.5, \quad a = 0.5,\ b = 1,\ c = 0,\ d = 0.5, \\
&a = 0,\ b = 0.5,\ c = 0.5,\ d = 1, \quad a = 0.5,\ b = 1,\ c = 0.5,\ d = 1.
\end{aligned}
$$

The basic idea is shown in idealized form in Figure 15.11. Each of the four new step-1 patches is defined by nine points, but some of the new points are identical, so the four patches are fully defined by 16 points $\mathbf{P}_{ij}^1$ for i and j from 00 to 33. The first of the four patches (Figure 15.11a) is constructed by setting $a = 0$, $b = 0.5$, $c = 0$, and $d = 0.5$ (it is a reparametrization of the "upper left" quadrant of the original, step-0 surface patch) and then applying Equation (15.5). The resulting nine control points $\mathbf{P}_{ij}^1$ are (from the

code of Figure 15.9)

$$\begin{aligned}
\mathbf{P}^1_{00} &= \frac{1}{16}(9\mathbf{P}^0_{00} + 3\mathbf{P}^0_{10} + 3\mathbf{P}^0_{01} + \mathbf{P}^0_{11}), \quad \mathbf{P}^1_{01} = \frac{1}{16}(3\mathbf{P}^0_{00} + \mathbf{P}^0_{10} + 9\mathbf{P}^0_{01} + 3\mathbf{P}^0_{11}),\\
\mathbf{P}^1_{02} &= \frac{1}{16}(9\mathbf{P}^0_{01} + 3\mathbf{P}^0_{11} + 3\mathbf{P}^0_{02} + \mathbf{P}^0_{12}), \quad \mathbf{P}^1_{10} = \frac{1}{16}(3\mathbf{P}^0_{00} + 9\mathbf{P}^0_{10} + \mathbf{P}^0_{01} + 3\mathbf{P}^0_{11}),\\
\mathbf{P}^1_{11} &= \frac{1}{16}(\mathbf{P}^0_{00} + 3\mathbf{P}^0_{10} + 3\mathbf{P}^0_{01} + 9\mathbf{P}^0_{11}), \quad \mathbf{P}^1_{12} = \frac{1}{16}(3\mathbf{P}^0_{01} + 9\mathbf{P}^0_{11} + \mathbf{P}^0_{02} + 3\mathbf{P}^0_{12}),\\
\mathbf{P}^1_{20} &= \frac{1}{16}(9\mathbf{P}^0_{10} + 3\mathbf{P}^0_{20} + 3\mathbf{P}^0_{11} + \mathbf{P}^0_{21}), \quad \mathbf{P}^1_{21} = \frac{1}{16}(3\mathbf{P}^0_{10} + \mathbf{P}^0_{20} + 9\mathbf{P}^0_{11} + 3\mathbf{P}^0_{21}),\\
\mathbf{P}^1_{22} &= \frac{1}{16}(9\mathbf{P}^0_{11} + 3\mathbf{P}^0_{21} + 3\mathbf{P}^0_{12} + \mathbf{P}^0_{22}).
\end{aligned} \tag{15.6}$$

```
(* reparametrize biquadratic B-spline surface *)
Clear[a,b,c,d,A,B,TB,H,M,P,Q];
M={{1,-2,1},{-2,2,0},{1,1,0}}/2;
A={{(b-a)^2,0,0},{2a(b-a),b-a,0},{a^2,a,1}};
(* B=MatrixForm[Simplify[Inverse[M].A.M]] *)
B={{((1 - a)*(1 - 2*a + b))/2, (1 + 3*a - 4*a^2 - b + 2*a*b)/2,
  a^2 - (a*b)/2}, {1/2 - a/2 - b/2 + (a*b)/2, (1 + a + b - 2*a*b)/2,
  (a*b)/2}, {((1 + a - 2*b)*(1 - b))/2, (1 - a + 3*b + 2*a*b - 4*b^2)/2,
  -(a*b)/2 + b^2}};
TB={{((1 - c)*(1 - 2*c + d))/2, 1/2 - c/2 - d/2 + (c*d)/2,
   ((1 + c - 2*d)*(1 - d))/2},
 {(1 + 3*c - 4*c^2 - d + 2*c*d)/2, (1 + c + d - 2*c*d)/2,
  (1 - c + 3*d + 2*c*d - 4*d^2)/2},
 {c^2 - (c*d)/2, (c*d)/2, -(c*d)/2 + d^2}};
P={{P00,P01,P02},{P10,P11,P12},{P20,P21,P22}};
Q=Simplify[B.P.TB]
a=0; b=.5; c=0; d=.5; Q
```

Figure 15.9: Code for the Nine Control Points of the "Upper-Left" Patch.

These points can be interpreted geometrically in two ways as follows:

1. The original surface patch has four faces and each new control point is located on one of these faces. Such a point is a weighted sum (with weights 9/16, 3/16, 3/16, and 1/16) of the four points on its face. Figure 15.10 shows the four possible weight patterns.

2. Take the two points $\mathbf{P}^0_{00}$ and $\mathbf{P}^0_{10}$ and add them with weights 3/4 and 1/4. Do the same with $\mathbf{P}^0_{01}$ and $\mathbf{P}^0_{11}$. Add the two results also with weights 3/4 and 1/4, and call the resulting point $\mathbf{P}^1_{00}$. Each new point is therefore the sum of two quantities, each a sum of two points on the same edge, where all the sums use weights of 3/4 and 1/4. Recall that these are the weights used by the original Chaikin's algorithm.

Using each of the other three sets of parameters to reparametrize the surface results in three more sets of nine more points each, only some of which are new. Fig-

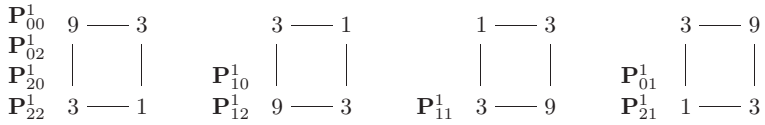


Figure 15.10: The Four Weight Patterns.

ure 15.11b,c,d shows the points (as small triangles) for the sets

$$\begin{aligned} &a = 0.5,\ b = 1,\ c = 0,\ d = 0.5, \quad \text{part (b)},\\ &a = 0,\ b = 0.5,\ c = 0.5,\ d = 1, \quad \text{part (c)},\\ &a = 0.5,\ b = 1,\ c = 0.5,\ d = 1, \quad \text{part (d)}. \end{aligned}$$

The total number of points $\mathbf{P}^1_{ij}$ is $9 + 3 + 3 + 1 = 16$, enough for four new (step 1) biquadratic patches based on nine points each. We can either display the four surface patches or proceed to step 2.

In step 2 of the subdivision, the new mesh of 16 points is used to calculate $4 \times 9 = 36$ points $\mathbf{P}^2_{ij}$. Figure 15.11e shows nine of them, marked as $\times$ (green), and Figure 15.11f shows all 36, enough points for $4 \times 4 = 16$ step-2 biquadratic patches based on nine points each.

This subdivision process is repeated several times, resulting in more and more points. When enough points have been obtained, the surface can be generated by connecting the points with short straight segments. It becomes a polygonal surface made of four-sided polygons (quadrilaterals).

An examination of all the parts of Figure 15.11 seems to suggest that the subdivision process produces smaller and smaller meshes of control points, thereby generating smaller and smaller surface patches. It is easy to show that this is not so. Let $\mathbf{Q}(u, w)$ denote the reparametrization of the "upper left" quadrant of the original surface patch $\mathbf{P}(u, w)$. These two patches are based on different meshes of control points, but we show that they have the same "upper left" corner point, i.e., $\mathbf{Q}(0, 0) = \mathbf{P}(0, 0)$. The corner point $\mathbf{P}(0, 0)$ of a general biquadratic B-spline surface patch $\mathbf{P}(u, w)$ is shown by Equation (14.50) to be

$$\mathbf{P}(0, 0) = \frac{1}{4}(\mathbf{P}_{00} + \mathbf{P}_{01} + \mathbf{P}_{10} + \mathbf{P}_{11}).$$

The corner point $\mathbf{Q}(0, 0)$ is therefore

$$\begin{aligned} \mathbf{Q}(0, 0) &= \frac{1}{4}(\mathbf{P}^1_{00} + \mathbf{P}^1_{01} + \mathbf{P}^1_{10} + \mathbf{P}^1_{11})\\ &= \frac{1}{4 \cdot 16}\big[(9\mathbf{P}^0_{00} + 3\mathbf{P}^0_{10} + 3\mathbf{P}^0_{01} + \mathbf{P}^0_{11}) + (3\mathbf{P}^0_{00} + \mathbf{P}^0_{10} + 9\mathbf{P}^0_{01} + 3\mathbf{P}^0_{11})\\ &\quad + (3\mathbf{P}^0_{00} + 9\mathbf{P}^0_{10} + \mathbf{P}^0_{01} + 3\mathbf{P}^0_{11}) + (\mathbf{P}^0_{00} + 3\mathbf{P}^0_{10} + 3\mathbf{P}^0_{01} + 9\mathbf{P}^0_{11})\\ &= \frac{1}{4}(\mathbf{P}^0_{00} + \mathbf{P}^0_{01} + \mathbf{P}^0_{10} + \mathbf{P}^0_{11})\big]\\ &= \mathbf{P}(0, 0). \end{aligned}$$

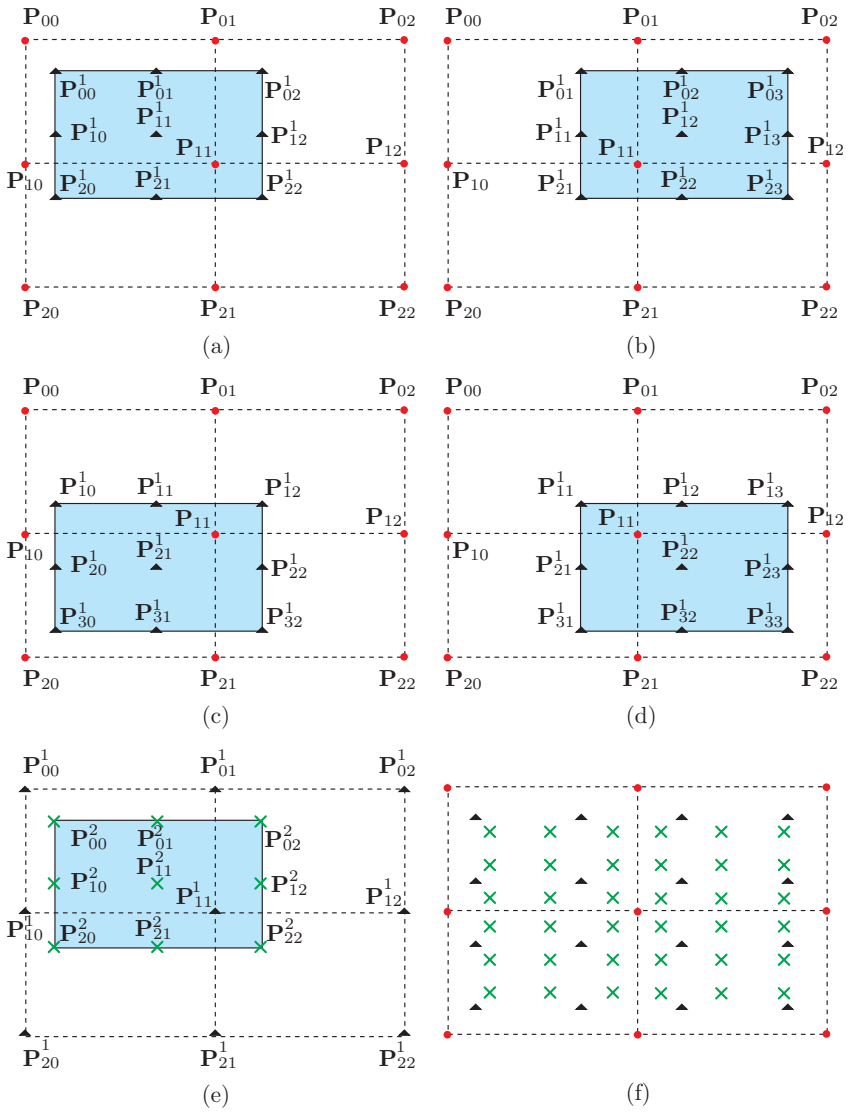


Figure 15.11: The First Two Subdivision Steps.

It turns out that even though consecutive steps of the subdivision process result in smaller meshes, those meshes converge to a limit and do not shrink indefinitely.

15.6 Bicubic B-Spline Surface by Subdivision

The technique employed in this section to subdivide a bicubic B-spline surface patch is similar to the one used in Section 15.5 to subdivide the biquadratic B-spline surface.

The rule for subdividing a bicubic B-spline surface patch $\mathbf{P}(u,w)$ uses reparametrization to compute four new patches $\mathbf{Q}(u,w)$. This is done by rewriting Equation (13.62) in the form

$$
\begin{aligned}
&\mathbf{Q}(u,w)\\
&= \mathbf{P}([b-a]u+a, [d-c]w+c)\\
&= \big(([b-a]u+a)^3, ([b-a]u+a)^2, ([b-a]u+a), 1\big)\mathbf{M}\cdot\mathbf{P}\cdot\mathbf{M}^{-1}\begin{pmatrix}([d-c]w+c)^3\\ ([d-c]w+c)^2\\ [d-c]w+c\\ 1\end{pmatrix}\\
&= (u^3,u^2,u,1)\mathbf{A}_{ab}\mathbf{M}\cdot\mathbf{P}\cdot\mathbf{M}^T\cdot\mathbf{A}_{cd}^T(w^3,w^2,w,1)^T\\
&= (u^3,u^2,u,1)\mathbf{M}(\mathbf{M}^{-1}\cdot\mathbf{A}_{ab}\cdot\mathbf{M})\mathbf{P}(\mathbf{M}^T\cdot\mathbf{A}_{cd}^T\cdot(\mathbf{M}^T)^{-1})\mathbf{M}^T(w^3,w^2,w,1)^T\\
&= (u^3,u^2,u,1)\mathbf{M}\cdot\mathbf{B}_{ab}\cdot\mathbf{P}\cdot\mathbf{B}_{cd}^T\cdot\mathbf{M}^T(w^3,w^2,w,1)^T\\
&= (u^3,u^2,u,1)\mathbf{M}\cdot\mathbf{Q}\cdot\mathbf{M}^T(w^3,w^2,w,1)^T, \qquad (15.7)
\end{aligned}
$$

where

$$
\mathbf{M} = \frac{1}{6}\begin{pmatrix}-1 & 3 & -3 & 1\\ 3 & -6 & 3 & 0\\ -3 & 0 & 3 & 0\\ 1 & 4 & 1 & 0\end{pmatrix}, \quad \mathbf{A}_{ab} = \begin{pmatrix}(b-a)^2 & 0 & 0\\ 2a(b-a) & b-a & 0\\ a^2 & a & 1\end{pmatrix}
$$

($\mathbf{M}$ is the basis matrix for the bicubic B-spline surface),

$$
\mathbf{P} = \begin{pmatrix}\mathbf{P}_{00} & \mathbf{P}_{01} & \mathbf{P}_{02} & \mathbf{P}_{03}\\ \mathbf{P}_{10} & \mathbf{P}_{11} & \mathbf{P}_{12} & \mathbf{P}_{13}\\ \mathbf{P}_{20} & \mathbf{P}_{21} & \mathbf{P}_{22} & \mathbf{P}_{23}\\ \mathbf{P}_{30} & \mathbf{P}_{31} & \mathbf{P}_{32} & \mathbf{P}_{33}\end{pmatrix},
$$

$$
\begin{aligned}
&\mathbf{B}_{ab} = \mathbf{M}^{-1}\cdot\mathbf{A}_{ab}\cdot\mathbf{M}\\
&= \begin{pmatrix}((1-a)(1-5a+6a^2+3b-7ab+2b^2))/6 & (4-22a^2+18a^3+20ab-21a^2b-4b^2+6ab^2)/6\\ ((a-1)(-1+2a-2ab+b^2))/6 & (4-4a^2-4ab+6a^2b+2b^2-3ab^2)/6\\ ((a-1)(1+a-2b)(b-1))/6 & (4+2a^2-4ab-3a^2b-4b^2+6ab^2)/6\\ ((1-b)(1+3a+2a^2-5b-7ab+6b^2))/6 & (4-4a^2+20ab+6a^2b-22b^2-21ab^2+18b^3)/6\end{pmatrix}\\
&\qquad\begin{pmatrix}1/6+a+(11a^2)/6-3a^3-b/2-(5ab)/3+(7a^2b)/2+b^2/3-ab^2 & a^3-(7a^2b)/6+(ab^2)/3\\ 1/6+a/2+a^2/3+(ab)/3-a^2b-b^2/6+(ab^2)/2 & (a(2a-b)b)/6\\ 1/6-a^2/6+b/2+(ab)/3+(a^2b)/2+b^2/3-ab^2 & (ab(-a+2b))/6\\ 1/6-a/2+a^2/3+b-(5ab)/3-a^2b+(11b^2)/6+(7ab^2)/2-3b^3 & (a^2b)/3-(7ab^2)/6+b^3\end{pmatrix},
\end{aligned}
$$

$$\mathbf{B}_{cd}^T = \mathbf{M}^T \cdot \mathbf{A}_{cd}^T \cdot (\mathbf{M}^T)^{-1}$$

$$= \begin{pmatrix} ((1-c)(1-5c+6c^2+3d-7cd+2d^2))/6 & ((-1+c)(-1+2c-2cd+d^2))/6 \\ (4-22c^2+18c^3+20cd-21c^2d-4d^2+6cd^2)/6 & (4-4c^2-4cd+6c^2d+2d^2-3cd^2)/6 \\ 1/6+c+(11c^2)/6-3c^3-d/2-(5cd)/3+(7c^2d)/2+d^2/3-cd^2 & 1/6+c/2+c^2/3+(cd)/3-c^2d-d^2/6+(cd^2)/2 \\ c^3-(7c^2d)/6+(cd^2)/3 & (c(2c-d)d)/6 \\ ((-1+c)(1+c-2d)(-1+d))/6 & ((1-d)(1+3c+2c^2-5d-7cd+6d^2))/6 \\ (4+2c^2-4cd-3c^2d-4d^2+6cd^2)/6 & (4-4c^2+20cd+6c^2d-22d^2-21cd^2+18d^3)/6 \\ 1/6-c^2/6+d/2+(cd)/3+(c^2d)/2+d^2/3-cd^2 & 1/6-c/2+c^2/3+d-(5cd)/3-c^2d+(11d^2)/6+(7cd^2)/2-3d^3 \\ (cd(-c+2d))/6 & (c^2d)/3-(7cd^2)/6+d^3 \end{pmatrix},$$

and

$$\mathbf{Q} = \mathbf{B}_{ab} \cdot \mathbf{P} \cdot \mathbf{B}_{cd}^T. \tag{15.8}$$

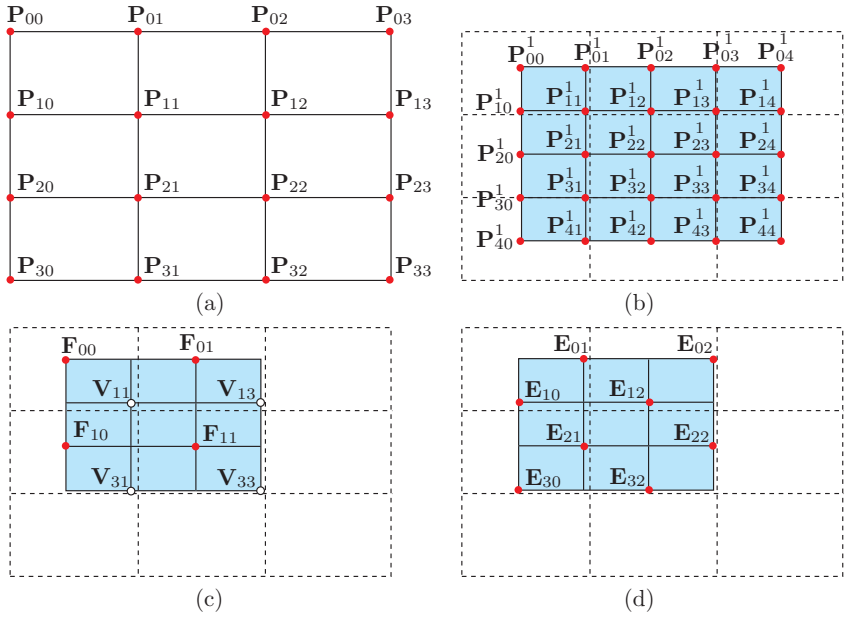


Figure 15.12: The First Subdivision Step.

The refinement rule for a bicubic B-spline patch $\mathbf{P}(u, w)$ is to use reparametrization to calculate the four surface patches defined by the following sets of parameters:

$$a = 0,\ b = 0.5,\ c = 0, d = 0.5, \quad a = 0.5,\ b = 1,\ c = 0,\ d = 0.5,$$
$$a = 0,\ b = 0.5,\ c = 0.5,\ d = 1, \quad a = 0.5,\ b = 1,\ c = 0.5,\ d = 1.$$

The basic idea is shown in idealized form in Figure 15.12a,b. Each of the new patches is defined by 16 points, but some of the new points are identical, so the four patches are fully defined by 25 points. The first of the four patches is constructed by setting $a = 0$, $b = 0.5$, $c = 0$, and $d = 0.5$ (this is a reparametrization of the "upper left" quadrant of the original surface patch) and applying Equation (15.8). The resulting 16 control points $\mathbf{P}^1_{ij}$ are (see Figure 15.13 for the computations)

$$
\begin{aligned}
\mathbf{P}^1_{00} &= \frac{1}{4}(\mathbf{P}^0_{00} + \mathbf{P}^0_{10} + \mathbf{P}^0_{01} + \mathbf{P}^0_{11}),\\
\mathbf{P}^1_{01} &= \frac{1}{16}(\mathbf{P}^0_{00} + \mathbf{P}^0_{10} + 6(\mathbf{P}^0_{01} + \mathbf{P}^0_{11}) + \mathbf{P}^0_{02} + \mathbf{P}^0_{12}),\\
\mathbf{P}^1_{02} &= \frac{1}{4}(\mathbf{P}^0_{01} + \mathbf{P}^0_{11} + \mathbf{P}^0_{02} + \mathbf{P}^0_{12}),\\
\mathbf{P}^1_{03} &= \frac{1}{16}(\mathbf{P}^0_{01} + \mathbf{P}^0_{11} + 6(\mathbf{P}^0_{02} + \mathbf{P}^0_{12}) + \mathbf{P}^0_{03} + \mathbf{P}^0_{13}),\\
\mathbf{P}^1_{01} &= \frac{1}{16}(\mathbf{P}^0_{00} + \mathbf{P}^0_{01} + 6(\mathbf{P}^0_{10} + \mathbf{P}^0_{11}) + \mathbf{P}^0_{20} + \mathbf{P}^0_{21}),\\
\mathbf{P}^1_{11} &= \frac{1}{64}(\mathbf{P}^0_{00} + 6\mathbf{P}^0_{10} + \mathbf{P}^0_{20} + 6(\mathbf{P}^0_{01} + 6\mathbf{P}^0_{11} + \mathbf{P}^0_{21}) + \mathbf{P}^0_{02} + 6\mathbf{P}^0_{12} + \mathbf{P}^0_{22}),\\
\mathbf{P}^1_{12} &= \frac{1}{16}(\mathbf{P}^0_{01} + \mathbf{P}^0_{12} + 6(\mathbf{P}^0_{11} + \mathbf{P}^0_{12}) + \mathbf{P}^0_{21} + \mathbf{P}^0_{22}),\\
\mathbf{P}^1_{13} &= \frac{1}{64}(\mathbf{P}^0_{01} + 6\mathbf{P}^0_{11} + \mathbf{P}^0_{21} + 6(\mathbf{P}^0_{02} + 6\mathbf{P}^0_{12} + \mathbf{P}^0_{22}) + \mathbf{P}^0_{03} + 6\mathbf{P}^0_{13} + \mathbf{P}^0_{23}),\\
\mathbf{P}^1_{20} &= \frac{1}{4}(\mathbf{P}^0_{10} + \mathbf{P}^0_{20} + \mathbf{P}^0_{11} + \mathbf{P}^0_{21}),\\
\mathbf{P}^1_{21} &= \frac{1}{16}(\mathbf{P}^0_{10} + \mathbf{P}^0_{20} + 6(\mathbf{P}^0_{11} + \mathbf{P}^0_{21}) + \mathbf{P}^0_{12} + \mathbf{P}^0_{22}),\\
\mathbf{P}^1_{22} &= \frac{1}{4}(\mathbf{P}^0_{11} + \mathbf{P}^0_{21} + \mathbf{P}^0_{12} + \mathbf{P}^0_{22}),\\
\mathbf{P}^1_{23} &= \frac{1}{16}(\mathbf{P}^0_{11} + \mathbf{P}^0_{21} + 6(\mathbf{P}^0_{12} + \mathbf{P}^0_{22}) + \mathbf{P}^0_{13} + \mathbf{P}^0_{23}),\\
\mathbf{P}^1_{30} &= \frac{1}{16}(\mathbf{P}^0_{10} + \mathbf{P}^0_{11} + 6(\mathbf{P}^0_{20} + \mathbf{P}^0_{21}) + \mathbf{P}^0_{30} + \mathbf{P}^0_{31}),\\
\mathbf{P}^1_{31} &= \frac{1}{64}(\mathbf{P}^0_{10} + 6\mathbf{P}^0_{20} + \mathbf{P}^0_{30} + 6(\mathbf{P}^0_{11} + 6\mathbf{P}^0_{21} + \mathbf{P}^0_{31}) + \mathbf{P}^0_{12} + 6\mathbf{P}^0_{22} + \mathbf{P}^0_{32}),\\
\mathbf{P}^1_{32} &= \frac{1}{16}(\mathbf{P}^0_{11} + \mathbf{P}^0_{12} + 6(\mathbf{P}^0_{21} + \mathbf{P}^0_{22}) + \mathbf{P}^0_{31} + \mathbf{P}^0_{32}),\\
\mathbf{P}^1_{33} &= \frac{1}{64}(\mathbf{P}^0_{11} + 6\mathbf{P}^0_{21} + \mathbf{P}^0_{31} + 6(\mathbf{P}^0_{12} + 6\mathbf{P}^0_{22} + \mathbf{P}^0_{32}) + \mathbf{P}^0_{13} + 6\mathbf{P}^0_{23} + \mathbf{P}^0_{33}).
\end{aligned}
\tag{15.9}
$$

These points can be classified into face points, edge points, and vertex points. The four face points are (Figure 15.12c) $\mathbf{F}_{00} = \mathbf{P}^1_{00}$, $\mathbf{F}_{01} = \mathbf{P}^1_{02}$, $\mathbf{F}_{10} = \mathbf{P}^1_{20}$, and $\mathbf{F}_{11} = \mathbf{P}^1_{22}$. Each is the average of four corner points of one face of the original patch. The eight edge points are (Figure 15.12d)

$$
\begin{aligned}
&\mathbf{E}_{01} = \mathbf{P}^1_{01}, \quad \mathbf{E}_{02} = \mathbf{P}^1_{03}, \quad \mathbf{E}_{10} = \mathbf{P}^1_{10}, \quad \mathbf{E}_{12} = \mathbf{P}^1_{12},\\
&\mathbf{E}_{21} = \mathbf{P}^1_{21}, \quad \mathbf{E}_{22} = \mathbf{P}^1_{23}, \quad \mathbf{E}_{30} = \mathbf{P}^1_{30}, \quad \mathbf{E}_{32} = \mathbf{P}^1_{32}.
\end{aligned}
$$

```
(* reparametrize bicubic B-spline surface *)
Clear[a,b,c,d,A,B,TB,H,M,P,Q];
M={{-1,3,-3,1},{3,-6,3,0},{-3,0,3,0},{1,4,1,0}}/6;
A={{(b-a)^3,0,0,0},{3a(b-a)^2,(b-a)^2,0,0},{3a^2(b-a),2a(b-a),b-a,0},{a^3,a^2,a,1}};
(*B=Simplify[Inverse[M].A.M] *)
B={{((1 - a)*(1 - 5*a + 6*a^2 + 3*b - 7*a*b + 2*b^2))/6,
  (4 - 22*a^2 + 18*a^3 + 20*a*b - 21*a^2*b - 4*b^2 + 6*a*b^2)/6,
  1/6 + a + (11*a^2)/6 - 3*a^3 - b/2 - (5*a*b)/3 + (7*a^2*b)/2 + b^2/3 -
   a*b^2, a^3 - (7*a^2*b)/6 + (a*b^2)/3},
 {((-1 + a)*(-1 + 2*a - 2*a*b + b^2))/6,
  (4 - 4*a^2 - 4*a*b + 6*a^2*b + 2*b^2 - 3*a*b^2)/6,
  1/6 + a/2 + a^2/3 + (a*b)/3 - a^2*b - b^2/6 + (a*b^2)/2,
  (a*(2*a - b)*b)/6}, {((-1 + a)*(1 + a - 2*b)*(-1 + b))/6,
  (4 + 2*a^2 - 4*a*b - 3*a^2*b - 4*b^2 + 6*a*b^2)/6,
  1/6 - a^2/6 + b/2 + (a*b)/3 + (a^2*b)/2 + b^2/3 - a*b^2,
  (a*b*(-a + 2*b))/6}, {((1 - b)*(1 + 3*a + 2*a^2 - 5*b - 7*a*b + 6*b^2))/
   6, (4 - 4*a^2 + 20*a*b + 6*a^2*b - 22*b^2 - 21*a*b^2 + 18*b^3)/6,
  1/6 - a/2 + a^2/3 + b - (5*a*b)/3 - a^2*b + (11*b^2)/6 + (7*a*b^2)/2 -
   3*b^3, (a^2*b)/3 - (7*a*b^2)/6 + b^3}};
TB={{((1 - a)*(1 - 5*a + 6*a^2 + 3*b - 7*a*b + 2*b^2))/6,
  ((-1 + a)*(-1 + 2*a - 2*a*b + b^2))/6,
  ((-1 + a)*(1 + a - 2*b)*(-1 + b))/6,
  ((1 - b)*(1 + 3*a + 2*a^2 - 5*b - 7*a*b + 6*b^2))/6},
 {(4 - 22*a^2 + 18*a^3 + 20*a*b - 21*a^2*b - 4*b^2 + 6*a*b^2)/6,
  (4 - 4*a^2 - 4*a*b + 6*a^2*b + 2*b^2 - 3*a*b^2)/6,
  (4 + 2*a^2 - 4*a*b - 3*a^2*b - 4*b^2 + 6*a*b^2)/6,
  (4 - 4*a^2 + 20*a*b + 6*a^2*b - 22*b^2 - 21*a*b^2 + 18*b^3)/6},
 {1/6 + a + (11*a^2)/6 - 3*a^3 - b/2 - (5*a*b)/3 + (7*a^2*b)/2 +
   b^2/3 - a*b^2, 1/6 + a/2 + a^2/3 + (a*b)/3 - a^2*b - b^2/6 +
   (a*b^2)/2, 1/6 - a^2/6 + b/2 + (a*b)/3 + (a^2*b)/2 + b^2/3 - a*b^2,
  1/6 - a/2 + a^2/3 + b - (5*a*b)/3 - a^2*b + (11*b^2)/6 + (7*a*b^2)/2 -
   3*b^3}, {a^3 - (7*a^2*b)/6 + (a*b^2)/3, (a*(2*a - b)*b)/6,
  (a*b*(-a + 2*b))/6, (a^2*b)/3 - (7*a*b^2)/6 + b^3}};
P={{P30,P31,P32,P33},{P20,P21,P22,P23},{P10,P11,P12,P13},{P00,P01,P02,P03}};
Q=Simplify[B.P.TB]
a=0; b=.5; c=0; d=.5; Q
```

Figure 15.13: Code for the 16 Control Points of the "Uper-Left" Patch.

Each is the average of two face points and the two points $\mathbf{P}_{ij}^0$ that are closest to it. The remaining four points are called vertex points. There is one vertex point for each interior vertex of the original mesh. The vertex points are shown in Figure 15.12c and they have the form $\mathbf{V} = (\mathbf{Q} + 2\mathbf{R} + \mathbf{S})/4$, where $\mathbf{S}$ is an interior vertex, $\mathbf{Q}$ is the average of the four face points located on the faces adjacent to $\mathbf{S}$, and $\mathbf{R}$ is the average of the midpoints of the four edges that meet at $\mathbf{S}$. As an example, consider the interior vertex $\mathbf{P}_{11}$ (Figure 15.14). If we denote $\mathbf{S} = \mathbf{P}_{11}$, then $\mathbf{Q}$ is the average of the four face points $\mathbf{F}_{00}$, $\mathbf{F}_{01}$, $\mathbf{F}_{10}$, and $\mathbf{F}_{11}$, and $\mathbf{R}$ is the average of the midpoints of the four edges $\mathbf{P}_{01}\mathbf{P}_{11}$, $\mathbf{P}_{10}\mathbf{P}_{11}$, $\mathbf{P}_{12}\mathbf{P}_{11}$, and $\mathbf{P}_{21}\mathbf{P}_{11}$ (the points labeled $\times$ in the figure). This interior vertex therefore corresponds to vertex point

$$\begin{aligned}&\frac{1}{4}(\mathbf{Q} + 2\mathbf{R} + \mathbf{S})\\ &= \frac{1}{4}(\mathbf{F}_{00} + \mathbf{F}_{01} + \mathbf{F}_{10} + \mathbf{F}_{11})\end{aligned}$$

$$
\begin{aligned}
&+\frac{2}{4}\left(\frac{\mathbf{P}_{01}+\mathbf{P}_{11}}{2}+\frac{\mathbf{P}_{10}+\mathbf{P}_{11}}{2}+\frac{\mathbf{P}_{12}+\mathbf{P}_{11}}{2}+\frac{\mathbf{P}_{21}+\mathbf{P}_{11}}{2}\right)+\mathbf{P}_{11}\\
&=\frac{1}{16}\left((\mathbf{P}_{00}+\mathbf{P}_{10}+\mathbf{P}_{01}+\mathbf{P}_{11})+(\mathbf{P}_{10}+\mathbf{P}_{20}+\mathbf{P}_{11}+\mathbf{P}_{21})\right.\\
&\quad\left.+(\mathbf{P}_{01}+\mathbf{P}_{11}+\mathbf{P}_{02}+\mathbf{P}_{12})+(\mathbf{P}_{11}+\mathbf{P}_{21}+\mathbf{P}_{12}+\mathbf{P}_{22})\right)\\
&\quad+\frac{1}{4}\left(\mathbf{P}_{01}+\mathbf{P}_{10}+\mathbf{P}_{21}+\mathbf{P}_{12}+4\mathbf{P}_{11}\right)+\mathbf{P}_{11}\\
&=\frac{1}{16}(\mathbf{P}_{00}+6\mathbf{P}_{10}+6\mathbf{P}_{01}+36\mathbf{P}_{11}+\mathbf{P}_{20}+6\mathbf{P}_{21}+\mathbf{P}_{02}+6\mathbf{P}_{12}+\mathbf{P}_{22})\\
&=\mathbf{P}_{11}^{1}.
\end{aligned}
$$

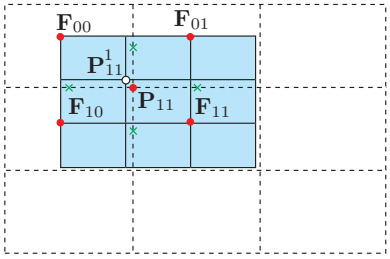


Figure 15.14: Constructing Vertex Point $\mathbf{P}_{11}^{1}$.

Here are the rules for calculating all 25 points $\mathbf{P}_{ij}^{1}$:

1. Construct one face point for each face of the original mesh. This point is the average of all the points defining the face.

2. Construct one edge point for each interior edge of the original mesh. This point is the average of the midpoint of the edge and the two face points of the faces adjacent to the edge.

3. Construct one vertex point for each interior vertex of the original mesh. This point is the average of (1) four face points, (2) four midpoints of edges, and (3) one interior vertex.

Since the original bicubic mesh consists of 9 faces, 12 interior edges, and 4 interior vertices, the first subdivision step results in 9 face points, 12 edge points, and 4 vertex points, a total of 25 points.

It should be noted that even though the mesh resulting from each subdivision is smaller than its predecessor, they do not shrink to a point but converge to a limit. All these meshes define the same bicubic B-spline surface.

◇ **Exercise 15.7:** Equation (14.52) shows that the "top left" corner of a bicubic B-spline patch is given by

$$\mathbf{P}(0,0)=\frac{1}{36}(\mathbf{P}_{00}+\mathbf{P}_{02}+4\mathbf{P}_{10}+4\mathbf{P}_{12}+\mathbf{P}_{20}+4\mathbf{P}_{01}+16\mathbf{P}_{11}+4\mathbf{P}_{21}+\mathbf{P}_{22}).$$

Show that this is still the same corner of the patch after one subdivision.

15.7 Polygonal Surfaces by Subdivision

Polygonal surfaces have been discussed in Section 9.2. Such a surface is normally obtained by measuring the coordinates of points on an object, either manually or with a three-dimensional digitizer. The designer then selects a set of points and the software connects those points with straight segments, resulting in a polygon. This is how the original mesh of points is converted to a set of polygons. The only condition is that the polygons be flat. The entire polygonal surface can then be shaded using Gouraud or Phong shading, Section 17.3. If the result is not smooth enough, it can be improved by subdividing the original mesh of points, which is why the subdivision of polygonal surfaces is important.

15.8 Doo Sabin Surfaces

The method described in this section is due to Donald Doo and Malcolm Sabin [Doo and Sabin 78]. They observed that the method used in Section 15.5 to subdivide a biquadratic B-spline surface patch generates each new point $\mathbf{P}^1_{ij}$ as a weighted sum of four points: a vertex point, two edge points, and a face point. For example, Equation (15.6) gives point $\mathbf{P}^1_{00}$ as

$$\mathbf{P}^1_{00} = \frac{1}{16}(9\mathbf{P}^0_{00} + 3\mathbf{P}^0_{10} + 3\mathbf{P}^0_{01} + \mathbf{P}^0_{11}),$$

so we write it in the form

$$\begin{aligned}
\mathbf{P}^1_{00} &= \frac{1}{16}(9\mathbf{P}^0_{00} + 3\mathbf{P}^0_{10} + 3\mathbf{P}^0_{01} + \mathbf{P}^0_{11}) \\
&= \frac{1}{16}\big(4\mathbf{P}^0_{00} + 2(\mathbf{P}^0_{00} + \mathbf{P}^0_{01}) + 2(\mathbf{P}^0_{00} + \mathbf{P}^0_{10}) + (\mathbf{P}^0_{00} + \mathbf{P}^0_{01} + \mathbf{P}^0_{10} + \mathbf{P}^0_{11})\big) \\
&= \frac{1}{4}\big(4\mathbf{P}^0_{00} + (\mathbf{P}^0_{00} + \mathbf{P}^0_{01})/2 + (\mathbf{P}^0_{00} + \mathbf{P}^0_{10})/2 + (\mathbf{P}^0_{00} + \mathbf{P}^0_{01} + \mathbf{P}^0_{10} + \mathbf{P}^0_{11})/4\big) \\
&= \frac{1}{4}\big(4\mathbf{V} + \mathbf{E}_1 + \mathbf{E}_2 + \mathbf{F}\big),
\end{aligned}$$

where $\mathbf{V}$ is the vertex point $\mathbf{P}^0_{00}$, $\mathbf{E}_1$ is the average of $\mathbf{P}^0_{00}$ and $\mathbf{P}^0_{01}$ (i.e., it is located midway between them), $\mathbf{E}_2$ is the average of $\mathbf{P}^0_{00}$ and $\mathbf{P}^0_{10}$, and $\mathbf{F}$ is the average of the four corners of the polygon being subdivided.

The idea of Doo and Sabin is to subdivide a mesh of points that consists of any polygons, not just quadrilaterals, by performing the following steps:

1. Consider a vertex $\mathbf{P}^0_i$ on the original mesh (Figure 15.15). It is located on a certain face F (and perhaps on other faces as well) and it lies at the intersection of two edges, $E1$ and $E2$ (two of the edges that form F). Create a new point $\mathbf{P}^1_i$ as a weighted average of $\mathbf{P}^0_i$, the two edge points adjacent to $\mathbf{P}^0_i$ (i.e., the center points of $E1$ and $E2$), and the face point that's the average of all the vertices forming F. Repeat this for every vertex $\mathbf{P}^0_i$. See Figure 15.16a for an example.

2. Consider face F again. It now contains some new points $\mathbf{P}^1_i$. Connect them so that they form a new polygon. This polygon will become a face in the new, refined surface. Repeat for all faces F (Figure 15.16b).

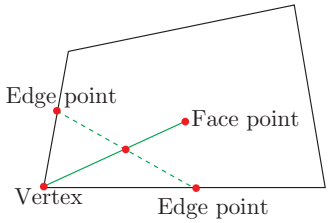


Figure 15.15: Edge and Face Points.

3. Consider again a vertex $\mathbf{P}_i^0$ on the original mesh. Such a vertex is normally common to several faces. For each of those faces, find the new point that's nearest $\mathbf{P}_i^0$. Connect those points to each other to form a new polygon. This polygon will also become a face in the new, subdivided surface. Repeat for all vertices $\mathbf{P}_i^0$ (Figure 15.16c).

4. Consider an edge of the original mesh of points. There will normally be two faces adjacent to this edge and they will have new points $\mathbf{P}_i^1$. Connect the new points around the edge to form a new polygon. This polygon will also become a face in the new, subdivided surface. Repeat this step for all edges (Figure 15.16d).

Notice that the new mesh may contain all kinds of polygons, not just triangles or quadrilaterals.

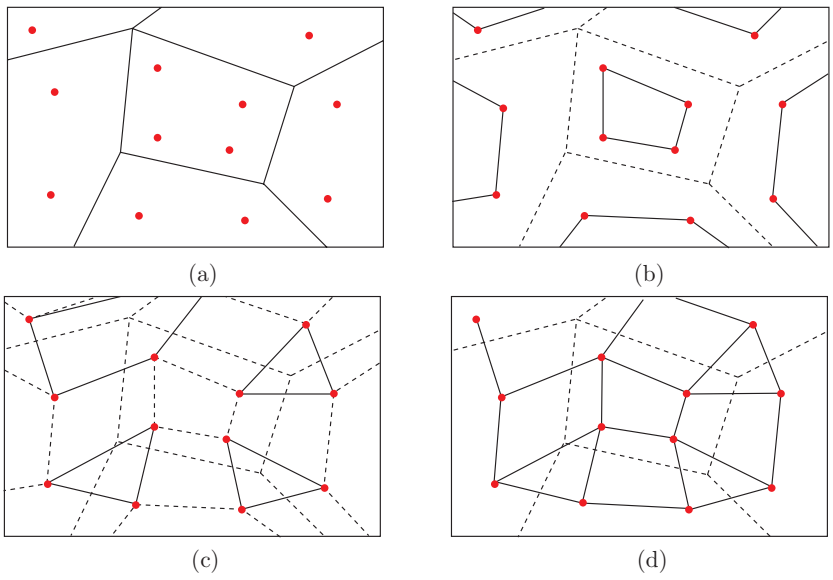


Figure 15.16: The First Doo–Sabin Subdivision Step.

15.9 Catmull–Clark Surfaces

The method described here is due to Edwin Catmull and Jim Clark [Catmull and Clark 78] and is an extension of the method of Section 15.6 to arbitrary polygonal surfaces. We have seen that subdividing a bicubic B-spline surface patch generates each new point $\mathbf{P}_{ij}^1$ as either a face point, an edge point, or a vertex point. A Catmull–Clark surface patch starts with an arbitrary polygonal surface and subdivides it by generating new face, edge, and vertex points and connecting them in a simple way. The rules for generating the points are the following:

1. A face point is calculated for each face of the original mesh. The point is simply the average of all the points that bound the face.

2. An edge point is created for each interior edge of the polygonal surface. The point is the average of the midpoint of the edge and of the two face points on both sides of the edge.

3. A vertex point is generated for each interior vertex $\mathbf{P}$ of the original mesh. The point is the average of $\mathbf{Q}$, $2\mathbf{R}$, and $\mathbf{S}(n-3)/4$, where $\mathbf{Q}$ is the average of the face points on all the faces adjacent to $\mathbf{P}$, $\mathbf{R}$ is the average of the midpoints of all the edges incident on $\mathbf{P}$, and $\mathbf{S}$ is simply $\mathbf{P}$ itself.

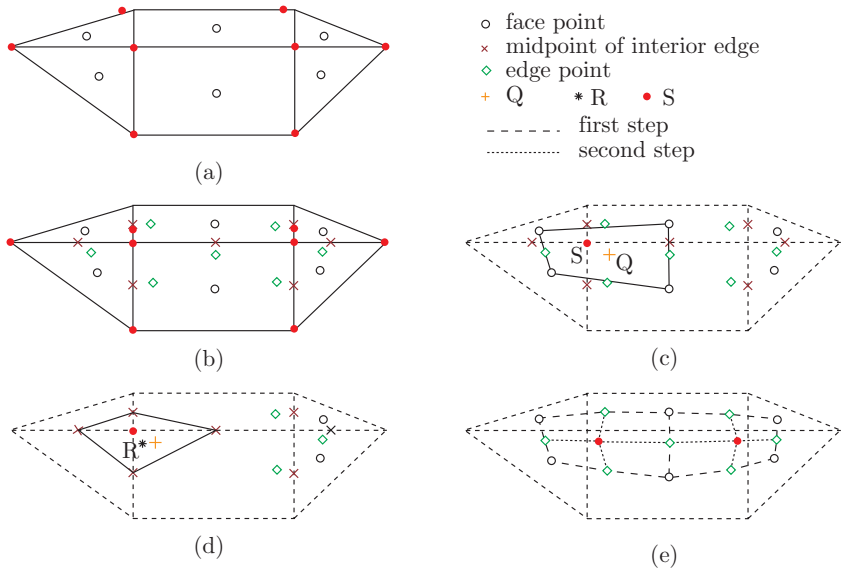


Figure 15.17: The First Catmull–Clark Subdivision Step.

Figure 15.17 shows an example. We start with a mesh of eight vertices defining six polygons, two rectangles and four triangles (notice that the polygons may have any number of sides, not just three or four). This surface has six faces, seven interior edges,

and two interior vertices. The six new face points are shown in Figure 15.17a as small circles. Each is the average of the points bounding its face. Figure 15.17b shows the midpoints of the edges as small ×'s and the seven new edge points as diamonds. In Figure 15.17c, we select one of the two interior vertices as **S**, temporarily connect the four face points surrounding it (just to identify them), and calculate **Q** (shown as a small "+") as their average. In Figure 15.17d, we show how **R** (the asterisk) is computed as the average of four midpoints of edges (temporarily connected).

After the new points have been generated, they are connected according to the following rules:

1. Each face point is connected to all the edge points of the interior edges bounding its face. These are shown as long dashes in Figure 15.17e.
2. Each new vertex point is connected to all the edge points that were used in calculating it. These lines are shown as short dashes in Figure 15.17e.

Notice that even though the original polygonal mesh may have polygons with any number of sides, the new, subdivided mesh will consist of quadrilaterals (four-sided polygons) only.

15.10 Loop Surfaces

The Loop subdivision scheme subdivides a triangular mesh surface by performing several iterations, yet the term Loop refers to its developer, Charles Teorell Loop. In his MS thesis [Loop 87] Loop developed an algorithm that subdivides each triangle into four smaller triangles (Figure 15.18a). This is now referred to as a binary Loop subdivision. In [Loop 02] he extended this algorithm to subdivide each triangle into nine smaller ones (Figure 15.18b). The extended algorithm is termed Loop ternary subdivision. Notice that the polygons that make up the surface must be triangles, they cannot be arbitrary flat polygons.

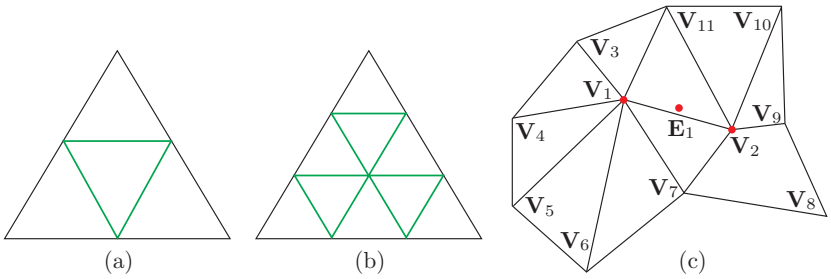


Figure 15.18: Binary and Ternary Loop Triangle Subdivisions.

The binary Loop algorithm starts with a set of points that are the vertices of triangles. Each iteration computes a new set of edge and vertex points that become the vertices of the new, smaller triangles. Specifically, a new edge point is computed for

each edge and a new vertex point is computed for each vertex of the triangular mesh. If the original mesh has E edges and V vertices, the new mesh will have $E + V$ vertex points (and a number of edges that depends on the complexity of the original mesh). The new points become the vertices of the new, finer mesh, and more iterations may be applied to refine the mesh as much as needed.

To understand the rule for generating an edge point, consider the edge between vertices $\mathbf{V}_1$ and $\mathbf{V}_2$ of Figure 15.18c. This edge, like any other edge, connects two vertices. Like most edges, it is shared by two triangles. The new edge point $\mathbf{E}_1$ for this edge is constructed as the weighted sum $\frac{3}{8}(\mathbf{V}_1 + \mathbf{V}_2) + \frac{1}{8}(\mathbf{V}_{11} + \mathbf{V}_7)$. The two vertices connected by the edge are given the large weights 3/8, whereas the two edges of the triangles sharing the edge are assigned the small weights 1/8. The weights are barycentric. If the edge is on the boundary of the surface and is part of only one triangle, the edge point is computed as the average of the two vertices connected by the edge. Thus, if the edge connected by $\mathbf{V}_3$ and $\mathbf{V}_{11}$ is on the boundary (i.e., there is no triangle "above" it), then the new edge point for this edge is the average $(\mathbf{V}_3 + \mathbf{V}_{11})/2$ and is located on the edge.

Notice that even though $\mathbf{E}_1$ is called an edge point, it does not have to be located on an edge, and it becomes a vertex, not an edge, in the new, finer triangular mesh constructed after the iteration.

Similarly, a new vertex point is also constructed as a weighted sum. The new vertex for $\mathbf{V}_1$, for example, is computed as the sum $\frac{5}{8}\mathbf{V}_1 + \frac{3}{8}\mathbf{Q}_1$ where $\mathbf{Q}_1$ is the average of the vertices of all the triangles sharing $\mathbf{V}_1$, i.e.,

$$\mathbf{Q}_1 = (\mathbf{V}_3 + \mathbf{V}_4 + \mathbf{V}_5 + \mathbf{V}_6 + \mathbf{V}_7 + \mathbf{V}_2 + \mathbf{V}_{11})/7.$$

If a vertex is located on the boundary of the surface, the weights are slightly different. For example, if the three vertices $\mathbf{V}_3$, $\mathbf{V}_{11}$, and $\mathbf{V}_{10}$ are on the boundary of the surface, then the new vertex point for $\mathbf{V}_{11}$ is computed as the sum $\frac{6}{8}\mathbf{V}_{11}+\frac{1}{8}(\mathbf{V}_3+\mathbf{V}_{10})$. Similarly, if $\mathbf{V}_8$ is a boundary point, then the new vertex for $\mathbf{V}_8$ is $\mathbf{V}_8$ itself.

A downside of this simple algorithm is that the fine mesh obtained after a few iterations may have several "extraordinary" points where the surface is not smooth. More precisely, the continuity of the tangent plane is lost at the extraordinary points. An improvement to the original algorithm computes each new vertex point as the weighted sum $\alpha_n \mathbf{V}_i + (1 - \alpha_n)\mathbf{Q}_i$, where $\mathbf{V}_i$ is a vertex shared by n triangles, $\mathbf{Q}_i$ is the average of the n vertices around $\mathbf{V}_i$, and

$$\alpha_n = \left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{n}\right)^2 + \frac{3}{8}.$$

Figure 15.19 illustrates the principle of ternary triangle subdivision. Dividing a triangle into nine smaller triangles requires (Figure 15.19a) the construction of one face point, six edge points (two on each edge) and three vertex points. Part (b) of the figure shows how one edge point (labeled "b") is computed as a weighted sum of seven vertices from six triangles. The weights shown should be normalized by dividing them by their sum, 81. The other edge points are computed similarly. Part (c) of the figure shows how the face point "c" is computed as a weighted sum of six vertices in four different

triangles. The weights should again be divided by their sum, 27. Computing a vertex point, such as "a" in part (a) of the figure, depends on the number n of the triangles sharing the vertex. Each vertex on an edge sharing "a" is assigned a weight of $(1-\alpha)/n$ and "a" itself is assigned weight α, where the value $\alpha = 5/9$ was found to work in most cases. Notice that the weights are barycentric.

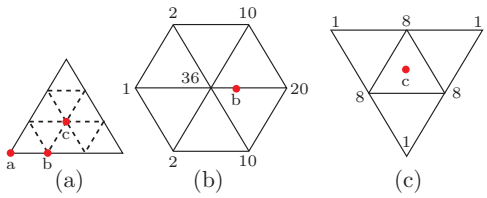


Figure 15.19: Edge and Face Points in Ternary Loop Subdivision.

As in the binary algorithm, the weights have to be modified for cases where a vertex or an edge is located on the boundary of the surface.

> Subdivision is a powerful paradigm for the generation of surfaces of arbitrary topology. Given an initial triangular mesh the goal is to produce a smooth and visually pleasing surface whose shape is controlled by the initial mesh. Of particular interest are interpolating schemes since they match the original data exactly, and play an important role in fast multiresolution and wavelet techniques.
>
> —D. Zorin, P. Schröder, and W. Sweldens

16
Sweep Surfaces

The surfaces described in this chapter are obtained by transforming a curve. They are not generated as interpolations or approximations of points or vectors and are consequently different from the surfaces described in previous chapters. A reader who wishes a full understanding of this chapter should first become familiar with the important three-dimensional transformations (rotation, translation, scaling, reflection, and shearing) and how they are described mathematically by a 4×4 transformation matrix. This material is discussed in Section 4.4, but the next paragraph provides a short summary, for those who only need a refresher.

A three-dimensional point $\mathbf{P} = (x, y, z)$ is transformed to a point $\mathbf{P}^* = (x^*, y^*, z^*)$ by appending a fourth coordinate of 1 to it and then multiplying it by the 4×4 transformation matrix

$$\mathbf{T} = \begin{pmatrix} a & b & c & p \\ d & e & f & q \\ h & i & j & r \\ l & m & n & s \end{pmatrix}. \tag{4.23}$$

The product $(x, y, z, 1)\mathbf{T}$ is a 4-tuple (X, Y, Z, H), where $H = xp + yq + zr + s$. The three coordinates (x^*, y^*, z^*) of $\mathbf{P}^*$ are obtained by dividing (X, Y, Z) by H. Hence, $(x^*, y^*, z^*) = (X/H, Y/H, Z/H)$. The top left 3×3 submatrix of $\mathbf{T}$ is responsible for scaling and reflection (parameters a, e, and j), shearing (b, c, f, and d, h, i), and rotation (all nine). The three quantities l, m, and n are responsible for translation, and s is a global scale factor. The three parameters p, q, and r are used for perspective projection.

D. Salomon, *The Computer Graphics Manual*, Texts in Computer Science,
DOI 10.1007/978-0-85729-886-7_16, © Springer-Verlag London Limited 2011