Subdivision Curves and Surfaces

Lecture #4: 7 October 2002
Lecturer: Denis Zorin
Scribe: Christopher DeCoro

1 Subdivision Curves

Although curves can be generally described by using concepts such as NURBS, we will introduce subdivision curves as a lead-in to the more general concept of subdivision surfaces.

Suppose we wanted to perform knot-insertion in the middle of a B-spline curve defined by four control points, p_0 through p_3. Using the blossoming technique, we would determine the control point to be as in Figure 1. This results in a new point that is in the middle of the curve.

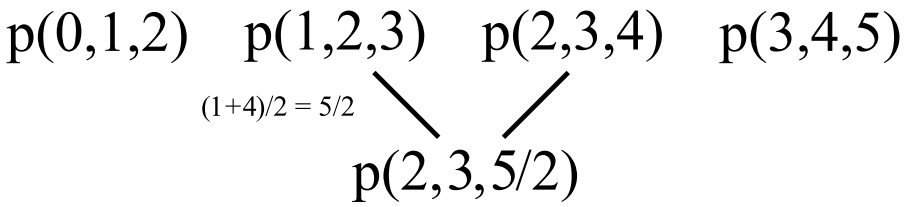


Figure 1: Adding a single knot in the middle of a B-spline curve

Suppose then we wanted to similarly perform knot insertion everywhere; that is, we want to double the number of control points. We can handle this in a general manner by considering that each point on the curve is affected by only 4 control points: the two points adjacent on both the left and right sides.

From following the blossoming procedure as in Figure 2, we can determine that the equations for an arbitrary inserted knot are as follows:

$$\begin{aligned} p &= \frac{1}{2}\cdot\frac{1}{4}p_0 + \frac{1}{2}\cdot\frac{1}{4}p_1 + \frac{5}{6}\cdot\frac{3}{4}p_1 + \frac{1}{6}\cdot\frac{3}{4}p_2 \\ &= \frac{1}{8}p_0 + \frac{3}{4}p_1 + \frac{1}{8}p_2 \end{aligned}$$

We can see in Figure 3 that the new control points are just an average of points in the previous level, and that the new control line (the line connecting the control points) is now closer to the curve. To compute a control point on the refinement level $j+1$, we have the

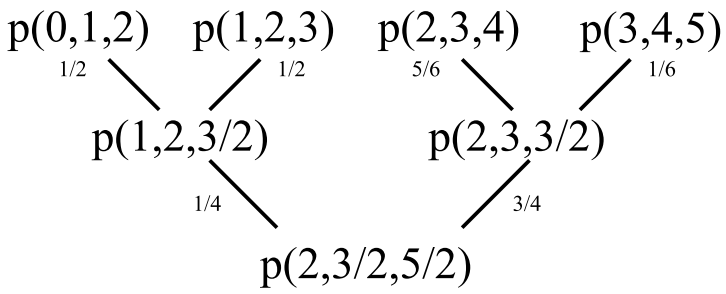


Figure 2: Adding arbitrary knots in B-spline curves.

following rules for subdivision, one for dding the extra knots/points and the other moving the existing ones:

$$\begin{aligned} p_{2i+1}^{j+1} &= \frac{1}{2}(p_i^j + p_{i+1}^j) \\ p_{2i}^{j+1} &= \frac{1}{8}p_{i-1}^j + \frac{3}{4}p_i^j + \frac{1}{8}p_{i+1}^j \end{aligned}$$

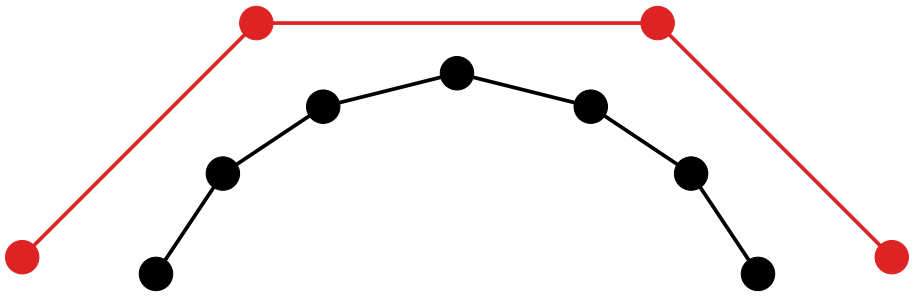


Figure 3: Subdividing an initial set of control points (upper, red) results in additional control points (lower, black), that more closely approximate a curve.

Repeated iteration of this process will result in a smooth limit curve (assuming a theoretical infinite number of interations). Of course, an infinite number of iterations is unnecessary, as we will surpass the resolution of our output device in a finite number of repetitions. This process is useful, because most output devices are designed to draw a series of lines, as would be output by this process.

An important question to ask is whether or not this method can be generalized. A more general form of subdivision places arbitrary coefficients in front of the terms. In our case, the generalized rules are:

$$p_{2i+1}^{j+1} = a_{11}p_i^j + a_{12}p_{i+1}^j$$

$$p_{2i}^{j+1} = a_{21}p_{i-1}^{j} + a_{22}p_i^j + a_{23}p_{i+1}^j$$

It is then important to determine for what values of a these functions will converge to a smooth limit curve. For example a rule of the form $p_{2i}^{j+1} = ap_i^j$, or, iterated k times,

$$p_{2^k i}^{j+k} = a^k p_i^j$$

will result in points moving to infinity if $a > 1$. Therefore, to converge, a should be no greater than 1.

To achieve affine the coefficients should sum up to one. FOr the type of rules we are considering, this amounts to the conditions

$$a_{11} + a_{12} = 1$$

$$a_{21} + a_{22} + a_{23} = 1$$

Such conditions can be expressed in a more general form. If we are only interested in a local behavior of the curve near a point only few control points on each level matter. Given 5 points $p_{0\ldots4}^j$ on level j, we can compute the corresponding five points $p_{0\ldots4}^{j+1}$ from level $j+1$ using the following equations. Note that for each successive level, the distance between the points decreases.

$$\begin{aligned}
p_0^{j+1} &= \frac{1}{8}p_0^j + \frac{3}{4}p_1^j + \frac{1}{8}p_2^j \\
p_1^{j+1} &= \frac{1}{2}p_0^j + \frac{1}{2}p_1^j \\
p_2^{j+1} &= \frac{1}{8}p_1^j + \frac{3}{4}p_2^j + \frac{1}{8}p_3^j \\
p_3^{j+1} &= \frac{1}{2}p_1^j + \frac{1}{2}p_2^j \\
p_4^{j+1} &= \frac{1}{8}p_2^j + \frac{3}{4}p_3^j + \frac{1}{8}p_4^j
\end{aligned}$$

We can write these equations in matrix form, producing the subdivision matrix S, the matrix such that $p^j = S^j p$. In other words, we may find the points p^j by applying the j-th power of the matrix S to the points p.

$$S = \begin{pmatrix}
\frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 & 0 \\
\frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\
0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 \\
0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\
0 & 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8}
\end{pmatrix}$$

Using the subdivision matrix, we can apply linear algebra to perform analysis of the algorithm. The matrix can be transformed into the form:

$$\begin{aligned} S &= A \cdot diag(\lambda_1, \lambda_2, ...)A^{-1} \\ S^j &= A \cdot diag(\lambda_1^j, \lambda_2^j, ...)A^{-1} \end{aligned}$$

Thus, we can establish that a necessary condition for convergence is that S must be diagonalizable, and tha for all i, $|\lambda_i| \leq 1$ (this is actually not quite necessary in the most general case, but in all practical cases it is).

Finally, to analyze the limit position of the vertices as they are adjusted through the subdivision rules, we can apply decomposition with respect to the eigenbasis of S. In the following equations, we represent the i-th eigenvectors of S with x_i. We then establish that a_0x_0 is the limit position of the vertex.

$$\begin{aligned} p^0 &= \sum_{i=1}^{n} a_i x_i \\ Sx_i &= \lambda_i x_i \\ S^j p^0 = p^j &= \sum_{i=1}^{n} a_i \lambda_i^j x_i = a_0 x_0 + \sum_{i=1}^{n-1} a_i \lambda_i^j x_i \end{aligned}$$

The second term of the last equation vanishes as i increases.

2 Subdivision Surfaces

Subdivision surfaces are a two-dimensional generalization of subdivision curves, in which we will insert knots in both dimensions. We discuss a particular algorithm, known as **Catmull-Clark subdivision**, which operates on manifold polygonal meshes. We will focus on the case when the mesh is quadrilateral, although the original version of the algorithm works for an arbitrary mesh, converting it to a quadrilateral mesh after one subdivision step.

This algorithm reads a base (coarse) mesh M^0, and through the process of subdivision, reaches a smooth limit surface. This is shown in Figure 4, where the top-left mesh is M^0, the top-right mesh is M^1, the bottom-left mesh is M^2, and the bottom-right is the limit surface.

The algorithm provides a set of rules to transform a mesh M^n into a smoother mesh M^{n+1}. Each quadrilateral of M^n will be subdivided into 4 quads. The positioning of new control points is computed using 3 rules, one for each of the different types of points: face, edge, and vertex points. Examples of each type of point is given in Figure 5. For each type of rule, we can create a **subdivision mask**, which graphically shows the relation of M^n points to M^{n+1} points.

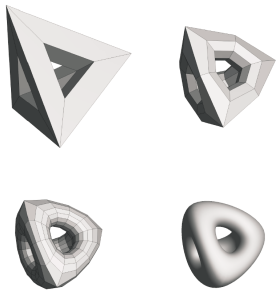


Figure 4: Successive applications of Catmull-Clark subdivision

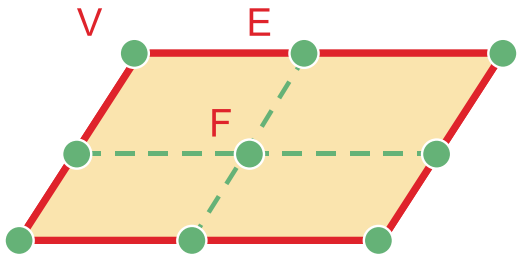


Figure 5: A quadrilateral being subdivided into four additional quads through Catmull-Clark subdivision. F indicates a face point, V a vertex point, and E an edge point

2.1 Face points

Face points are those newly added points in M^{n+1} which correspond to faces (quads) in M^n. New face points are placed at the centroid of each quadrilateral. This results in a new point that is an average of each of the four corner points; the corresponding mask assigns a weight of $\frac{1}{4}$ to each surrounding vertex of M^n, as follows in Figure 6

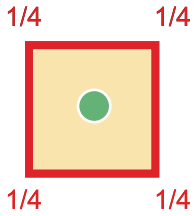


Figure 6: Subdivision mask for the face point rule. The green circle represents the added vertex

2.2 Edge points

Edge points are those that correspond the edges of M^n. A weight of $\frac{3}{8}$ is given to each of the endpoints of the edge; the other four corners of the quadrilaterals are assigned a weight of $\frac{1}{16}$, as shown in Figure 7.

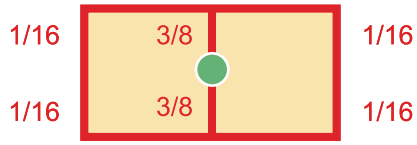


Figure 7: Subdivision mask for the edge point rule

2.3 Vertex points

The original vertices of the mesh M^n are repositioned in mesh M^{n+1} according to the subdivision rules. We will first present the rule for modifying a vertex point with four adjacent faces; after which we will present the general rule for n adjacent faces, and show that it coincides with the simple rule when $n = 4$.

In order to maintain consistency with B-spline patches, the vertex rule of Catmull-Clark subdivision, corresponding to the mask in Figure 8 assigns the endpoints of the edges adjacent to the vertex weights of $\frac{3}{32}$. Other vertices of the adjacent quadrilaterals are assigned weights of $\frac{1}{64}$, while the vertex itself is assigned a weight of $\frac{9}{16}$

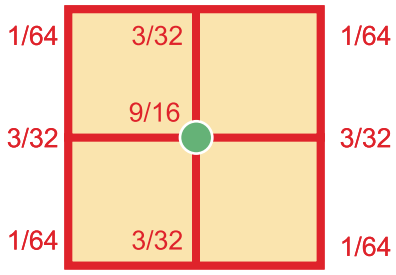


Figure 8: Subdivision mask for the vertex point rule when $n = 4$

To apply Catmull-Clark subdivision to general quadrilateral meshes (where the number of faces n adjacent to a vertex is not necessarily equal to 4), we construct a formula that assigns weights as a function of n. One valid solution is to assign the vertex itself a weight of $1 - \frac{7}{4n}$, adjacent vertices a weight of $\frac{3}{2n^2}$, and other vertices a weight of $\frac{1}{4n^2}$. Notice that when $n = 4$, this formulas yilds the B-spline weights.

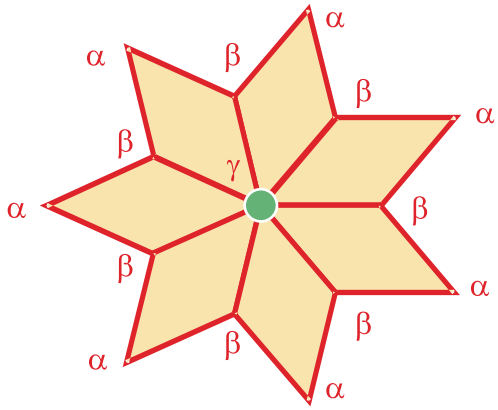


Figure 9: Subdivision mask for the general vertex point rule. The values for the weights are: $\alpha = \frac{1}{4k^2}, \beta = \frac{3}{2k^2}, \gamma = 1 - \frac{7}{4k}$