

On-Line Geometric Modeling Notes

DIRECT CALCULATION OF POINTS ON CUBIC SUBDIVISION CURVES

Kenneth I. Joy
Visualization and Graphics Research Group
Department of Computer Science
University of California, Davis

Given an initial control polygon we can define a refinement process that generates a sequence of control polygons from the original. In the limit, this sequence converges to the uniform B-spline curve specified by the original control polygon. By examining this refinement process from a different angle, we can specify a procedure that allows us to directly calculate points on the curve.

Direct Calculation of Points on the Curve

In the cubic case, we define the general refinement procedure by classifying the points of a refinement as either “vertex” or “edge” points. Using this classification we can cleverly write this refinement process in a matrix form. This form takes a control point and its two adjacent control points of a refinement and applies a refinement matrix as follows

$$\begin{bmatrix} \mathbf{V}^1 \\ \mathbf{E}_0^1 \\ \mathbf{E}_1^1 \end{bmatrix} = A \begin{bmatrix} \mathbf{V} \\ \mathbf{E}_0 \\ \mathbf{E}_1 \end{bmatrix}$$

where A is called the refinement matrix and is given by

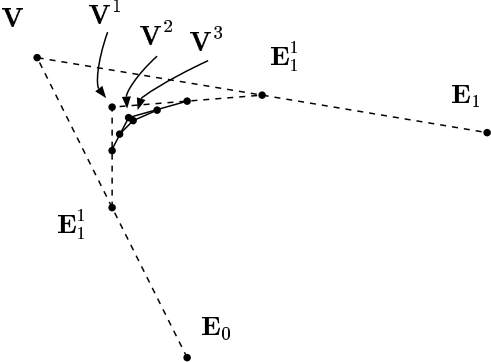
$$A = \frac{1}{8} \begin{bmatrix} 6 & 1 & 1 \\ 4 & 4 & 0 \\ 4 & 0 & 4 \end{bmatrix}$$

Here we have denoted the control points as vertex and edge points. This procedure creates two new edge points and a vertex point which are part of a new control polygon that represents the curve. We can apply A

again and obtain

$$\begin{bmatrix} \mathbf{V}^2 \\ \mathbf{E}_0^2 \\ \mathbf{E}_1^2 \end{bmatrix} = A \begin{bmatrix} \mathbf{V}^1 \\ \mathbf{E}_0^1 \\ \mathbf{E}_1^1 \end{bmatrix} = A^2 \begin{bmatrix} \mathbf{V} \\ \mathbf{E}_0 \\ \mathbf{E}_1 \end{bmatrix}$$

where $A^2 = AA$. In general, we can generate points on the k th refinement by applying A k -times. In the limit, as k approaches infinity, we obtain a point on the curve.



We call this point \mathbf{V}^∞ and note that it is equal to $\lim_{k \rightarrow \infty} \mathbf{V}^k$.

Calculating the Limit Point

The surprising thing is that we can actually calculate this limit. We utilize the fact that the refinement matrix can be diagonalized, which defines the matrix A by

$$A = R\Lambda L$$

where

$$R = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 1 & 2 \\ 1 & -1 & 2 \end{bmatrix}$$

is the matrix whose columns are the right eigenvectors of A ,

$$L = \begin{bmatrix} \frac{2}{3} & \frac{1}{6} & \frac{1}{6} \\ 0 & \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{bmatrix}$$

is the matrix whose rows are the left eigenvectors of A , and Λ is the matrix of eigenvalues

$$\Lambda = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{4} \end{bmatrix}$$

Since $R = L^{-1}$, this allows us to write

$$A^2 = (R\Lambda L)^2 = R\Lambda L R\Lambda L = R\Lambda\Lambda L = R\Lambda^2 L$$

(Noting that $R = L^{-1}$) or, in general

$$A^k = R\Lambda^k L$$

To calculate the limit, first consider applying A k -times

$$\begin{aligned} \begin{bmatrix} \mathbf{V}^k \\ \mathbf{E}_0^k \\ \mathbf{E}_1^k \end{bmatrix} &= A^k \begin{bmatrix} \mathbf{V} \\ \mathbf{E}_0 \\ \mathbf{E}_1 \end{bmatrix} \\ &= (R\Lambda L)^k \begin{bmatrix} \mathbf{V} \\ \mathbf{E}_0 \\ \mathbf{E}_1 \end{bmatrix} \\ &= R\Lambda^k L \begin{bmatrix} \mathbf{V} \\ \mathbf{E}_0 \\ \mathbf{E}_1 \end{bmatrix} \\ &= R \begin{bmatrix} (1)^k & 0 & 0 \\ 0 & (\frac{1}{2})^k & 0 \\ 0 & 0 & (\frac{1}{4})^k \end{bmatrix} L \begin{bmatrix} \mathbf{V} \\ \mathbf{E}_0 \\ \mathbf{E}_1 \end{bmatrix} \end{aligned}$$

Now as $k \rightarrow \infty$, this approaches

$$R \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} L \begin{bmatrix} \mathbf{V} \\ \mathbf{E}_0 \\ \mathbf{E}_1 \end{bmatrix}$$

and by substituting for R and L , this becomes

$$\begin{aligned}
R \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} L \begin{bmatrix} \mathbf{V} \\ \mathbf{E}_0 \\ \mathbf{E}_1 \end{bmatrix} &= \begin{bmatrix} 1 & 0 & -1 \\ 1 & 1 & 2 \\ 1 & -1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{2}{3} & \frac{1}{6} & \frac{1}{6} \\ 0 & \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{bmatrix} \begin{bmatrix} \mathbf{V} \\ \mathbf{E}_0 \\ \mathbf{E}_1 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{2}{3} & \frac{1}{6} & \frac{1}{6} \\ 0 & \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{bmatrix} \begin{bmatrix} \mathbf{V} \\ \mathbf{E}_0 \\ \mathbf{E}_1 \end{bmatrix} \\
&= \begin{bmatrix} \frac{2}{3} & \frac{1}{6} & \frac{1}{6} \\ \frac{2}{3} & \frac{1}{6} & \frac{1}{6} \\ \frac{2}{3} & \frac{1}{6} & \frac{1}{6} \end{bmatrix} \begin{bmatrix} \mathbf{V} \\ \mathbf{E}_0 \\ \mathbf{E}_1 \end{bmatrix} \\
&= \frac{1}{6} \begin{bmatrix} 4 & 1 & 1 \\ 4 & 1 & 1 \\ 4 & 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{V} \\ \mathbf{E}_0 \\ \mathbf{E}_1 \end{bmatrix}
\end{aligned}$$

That is, the vertex and edge points all converge to the same value, which is just the dot product of the left eigenvector of A that corresponds to the eigenvalue one, and the original vertex-edge vector of the refinement – obtaining the point $\frac{1}{6}(4\mathbf{V} + \mathbf{E}_0 + \mathbf{E}_1)$.

In short, given any vertex \mathbf{V}_i with corresponding edge points \mathbf{E}_i and \mathbf{E}_{i+1} , we can directly calculate points on the curve by dotting the left eigenvector that corresponds to the eigenvalue 1 by the vertex-edge vector

$$\frac{1}{6} \begin{bmatrix} 4 & 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{V}_i \\ \mathbf{E}_i \\ \mathbf{E}_{i+1} \end{bmatrix}$$

This says that any vertex point on any refinement directly corresponds to a point on the curve.

Examples

Consider a cubic uniform B-spline curve with control polygon $\{\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3\}$. This curve can be

written as

$$\mathbf{P}(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \frac{1}{6} \begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix}$$

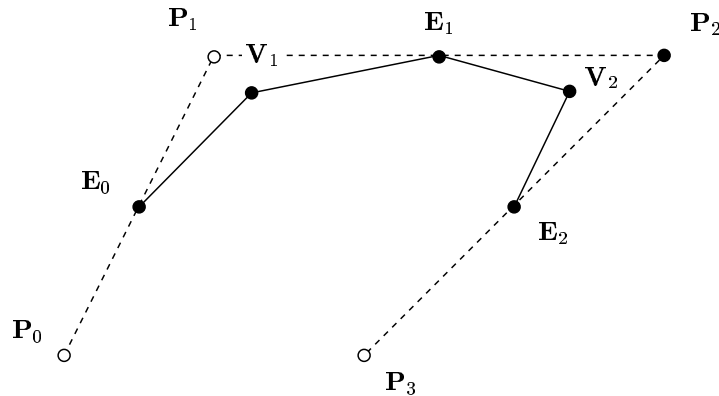
Consider the first step of the refinement using \mathbf{P}_1 as the vertex with \mathbf{P}_0 and \mathbf{P}_2 as the two respective edge points. The corresponding point on the curve can be calculated directly as

$$\frac{1}{6} \begin{bmatrix} 4 & 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_0 \\ \mathbf{P}_2 \end{bmatrix} = \frac{1}{6} (\mathbf{P}_0 + 4\mathbf{P}_1 + \mathbf{P}_2)$$

which is exactly the point $\mathbf{P}(0)$ on the curve.

Similarly we can calculate a point on the curve using the vertex \mathbf{P}_2 with \mathbf{P}_1 and \mathbf{P}_3 as the two respective edge points. In this case, we find that this is exactly the point $\mathbf{P}(1)$ on the curve.

Carrying this one step further, suppose we generate one full refinement of the curve, generating new edge and vertex points.



These new points become the input to the refinement process, and if we consider \mathbf{E}_1 as the vertex point,

with \mathbf{V}_1 and \mathbf{V}_2 as the respective edge points, we obtain the point on the curve

$$\begin{aligned} \begin{bmatrix} \frac{2}{3} & \frac{1}{6} & \frac{1}{6} \end{bmatrix} \begin{bmatrix} \mathbf{E}_1 \\ \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix} &= \begin{bmatrix} \frac{2}{3} & \frac{1}{6} & \frac{1}{6} \end{bmatrix} \begin{bmatrix} \frac{1}{2} (\mathbf{P}_1 + \mathbf{P}_2) \\ \frac{1}{8} (\mathbf{P}_0 + 6\mathbf{P}_1 + \mathbf{P}_2) \\ \frac{1}{8} (\mathbf{P}_1 + 6\mathbf{P}_2 + \mathbf{P}_3) \end{bmatrix} \\ &= \frac{1}{48} (\mathbf{P}_0 + 23\mathbf{P}_1 + 23\mathbf{P}_2 + \mathbf{P}_3) \end{aligned}$$

which is exactly $\mathbf{P}(\frac{1}{2})$.

Summary

It is possible, using eigenanalysis, to formulate simple procedures that calculate directly a point on the limit curve. This, in many cases, can be used to replace the overall refinement process.

The tangent vector on the curve at this limit point can also be directly calculated, by much the same procedure.

References

- [1] CATMULL, E., AND CLARK, J. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design 10* (Sept. 1978), 350–355.
- [2] HALSTEAD, M., KASS, M., AND DEROSE, T. Efficient, fair interpolation using Catmull-Clark surfaces. In *Computer Graphics (SIGGRAPH '93 Proceedings)* (Aug. 1993), J. T. Kajiya, Ed., vol. 27, pp. 35–44.

All contents copyright (c) 1996, 1997, 1998, 1999, 2000
 Computer Science Department, University of California, Davis
 All rights reserved.