

On-Line Geometric Modeling Notes

CHAIKIN'S ALGORITHMS FOR CURVES

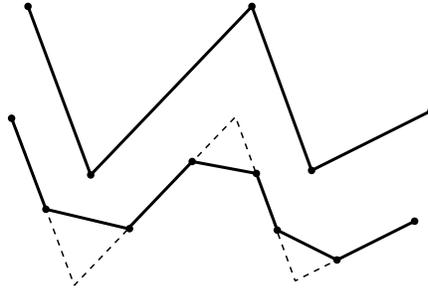
Kenneth I. Joy
Visualization and Graphics Research Group
Department of Computer Science
University of California, Davis

Overview

In 1974, George Chaikin [1] gave a lecture at the University of Utah in which he specified a novel procedure for generating curves from a limited number of points. This algorithm is interesting as it was one of the first corner cutting or refinement algorithms specified to generate a curve from a set of control points, or control polygon. The algorithm had been largely forgotten by the graphics/geometric-modeling community in the flurry of activity studying B-spline curves and surfaces, as it has been shown to generate a quadratic uniform B-spline curve. However, researchers have now drifted away from the rigidity of curve and surface models based upon an underlying analytical form, and the basic paradigm of Chaikin – his curves were generated by successive refinement of a control polygon – is now utilized to generate a wide variety of curve and surface types.

The Corner-Cutting Paradigm

Chaikin had a different idea. Researchers since Bézier had been working with curves generated by control polygons but had focused their analysis on the underlying analytical representation, frequently based upon Bernstein polynomials. Chaikin decided to develop algorithms that worked with the control polygon directly – so-called geometric algorithms. His curve generation scheme is based upon “corner cutting” where the algorithm generates a new control polygon by cutting the corners off the original one. The figure below illustrates this idea, where an initial control polygon has been refined into a second polygon (slightly offset) by cutting off the corners of the first sequence.



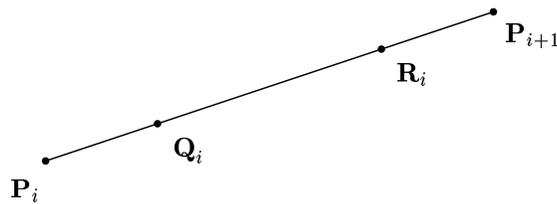
Clearly we could then take this second control polygon and cut the corners off it, producing a third sequence, etc. In the limit, hopefully we would have a curve. This was Chaikin's idea.

Chaikin's Method

Chaikin utilized fixed ratios on cutting off his corners, so that they were all cut the same. When written down mathematically, Chaikin's method proceeds as follows: Given a control polygon $\{\mathbf{P}_0, \mathbf{P}_1, \dots, \mathbf{P}_n\}$, we refine this control polygon by generating a new sequence of control points

$$\{\mathbf{Q}_0, \mathbf{R}_0, \mathbf{Q}_1, \mathbf{R}_1, \dots, \mathbf{Q}_{n-1}, \mathbf{R}_{n-1}\}$$

where each new pair of points $\mathbf{Q}_i, \mathbf{R}_i$ are to be taken taken to be at a ratio of $\frac{1}{4}$ and $\frac{3}{4}$ between the endpoints of the line segment $\overline{\mathbf{P}_i\mathbf{P}_{i+1}}$.



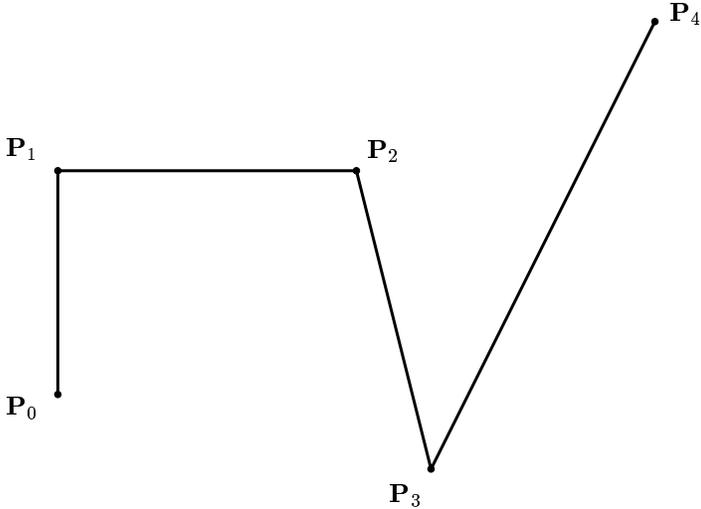
That is

$$\begin{aligned} \mathbf{Q}_i &= \frac{3}{4}\mathbf{P}_i + \frac{1}{4}\mathbf{P}_{i+1} \\ \mathbf{R}_i &= \frac{1}{4}\mathbf{P}_i + \frac{3}{4}\mathbf{P}_{i+1} \end{aligned}$$

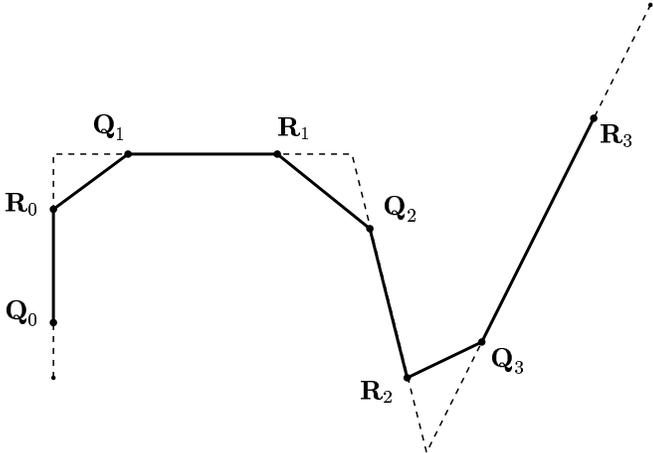
These $2n$ new points can be considered a new control polygon – a *refinement* of the original control polygon.

Example – How Chaikin’s Algorithm Works

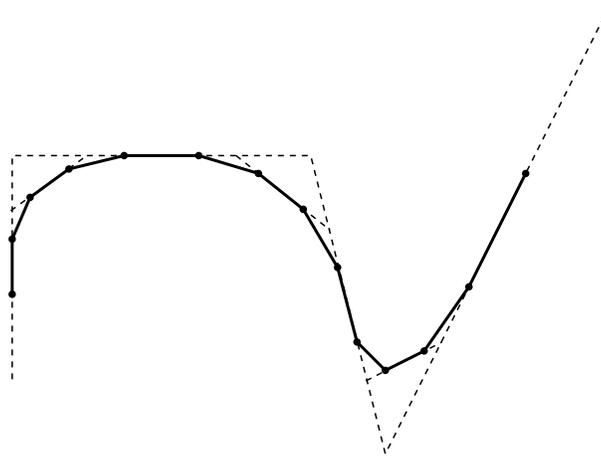
To give an example of Chaikin’s Algorithm, consider the following control polygon:



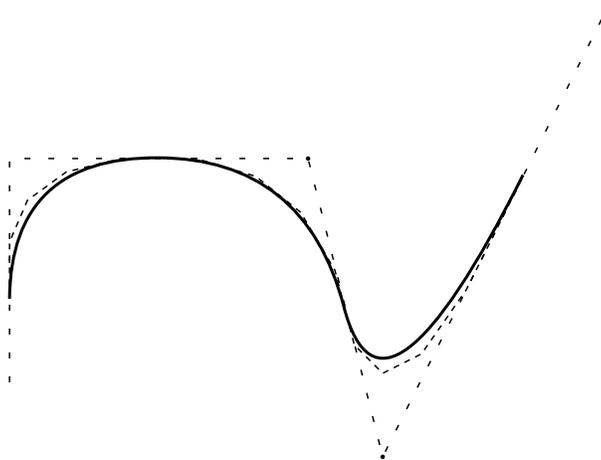
Chaikin’s algorithm generates the points Q_i and R_i and uses these points to refine the curve and obtain the control polygon shown in the figure below



These points are in turn utilized to generate a new refinement,

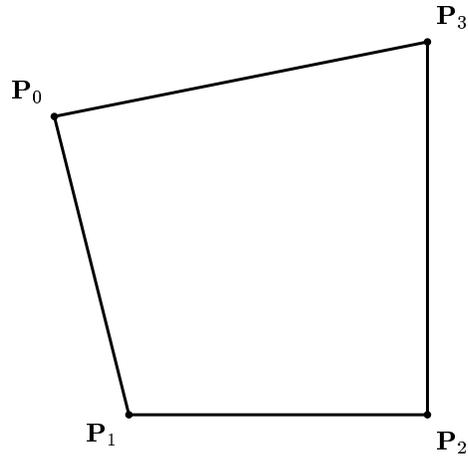


and again, these points are utilized to obtain another refinement, etc. The following illustration shows the initial control polygon, the third control polygon in the sequence, and the final curve.

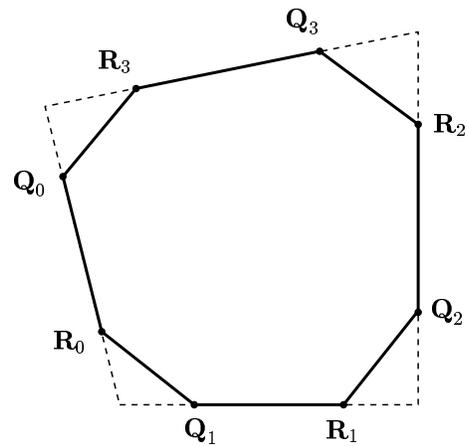


Example – A Closed Curve

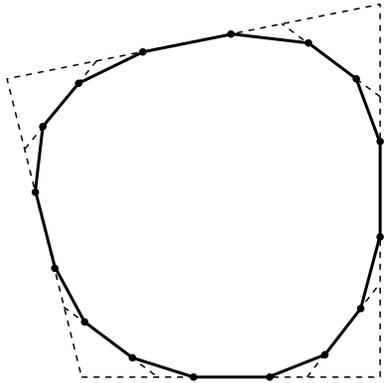
To illustrate Chaikin’s curve on a closed control polygon, consider the following figure.



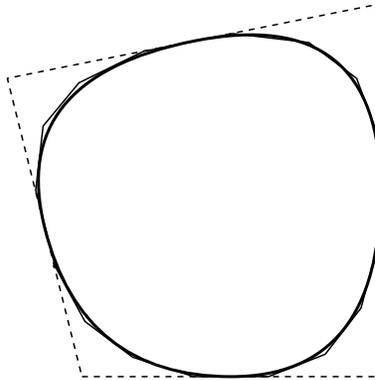
In this case, the control point indices are taken modulo $n + 1$ (or 4, in the case of the figure). We can still apply Chaikin's method to this figure, obtaining



This new control polygon can then be utilized to obtain a second refinement as in the following figure



and it is clear that we could continue this process indefinitely. For graphics purposes, we will stop after a number of refinements and approximate the curve by connecting the points of the resulting control polygon by straight lines. The initial control polygon and the second refinement are shown in the following figure along with the resulting curve.



Summary

Chaikin specified a simple scheme by which curves could be generated from a given control polygon. The idea is unique in that the underlying mathematical description is ignored in favor of a geometric algorithm that just selects new control points along the line segments of the original control polygon.

It should be noted that Chaikin's curve has been shown to be equivalent to a quadratic B-spline curve [2] (a piecewise quadratic Bézier curve). However, Chaikin's method avoids the analytical definition of B-splines and provides a simple, elegant curve drawing mechanism.

References

- [1] CHAIKIN, G. An algorithm for high speed curve generation. *Computer Graphics and Image Processing* 3 (1974), 346–349.
- [2] RIESENFELD, R. On Chaikin's algorithm. *IEEE Computer Graphics and Applications* 4, 3 (1975), 304–310.

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