

Subdivision curves

Brian Curless
CSE 557
Fall 2009

1

Reading

Recommended:

- ♦ Stollnitz, DeRose, and Salesin. *Wavelets for Computer Graphics: Theory and Applications*, 1996, section 6.1-6.3, 10.2, A.5.

Note: there is an error in Stollnitz, et al., section A.5. Equation A.3 should read:

$$MV = V\Lambda$$

This is already fixed in the hand out.

2

Subdivision curves

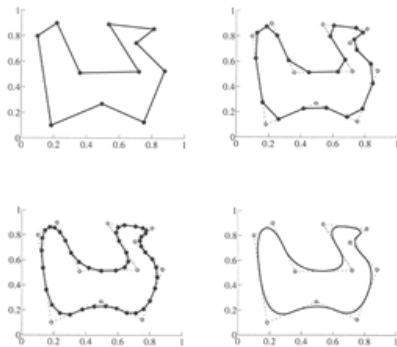
Idea:

- ♦ repeatedly refine the control polygon

$$P^1 \rightarrow P^2 \rightarrow P^3 \rightarrow \dots$$

- ♦ curve is the limit of an infinite process

$$Q = \lim_{j \rightarrow \infty} P^j$$

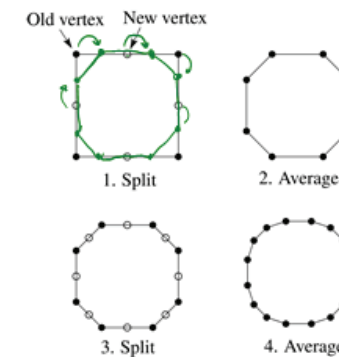


3

Chaikin's algorithm

Chaikin introduced the following "corner-cutting" scheme in 1974:

- ♦ Start with a piecewise linear curve
- ♦ Insert new vertices at the midpoints (the **splitting step**)
- ♦ Average each vertex with the "next" (clockwise) neighbor (the **averaging step**)
- ♦ Go to the splitting step



4


Averaging masks

The limit curve is a quadratic B-spline!

Instead of averaging with the nearest neighbor, we can generalize by applying an **averaging mask** during the averaging step:

$$r = (\dots, r_{-1}, r_0, r_1, \dots)$$

In the case of Chaikin's algorithm:

$$r = (0, \frac{1}{2}, \frac{1}{2})$$


5

Lane-Riesenfeld algorithm (1980)

Use averaging masks from Pascal's triangle: $n=0 \rightarrow$

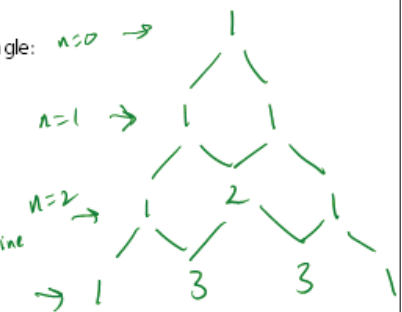
$$r = \frac{1}{2^n} \binom{n}{0}, \binom{n}{1}, \dots, \binom{n}{n}$$

Gives B-splines of degree $n+1$.

$n=0$: (1) \rightarrow linear B-spline

$n=1$: $\frac{1}{2}(1 \ 1)$ \rightarrow quadratic B-spline

$n=2$: $\frac{1}{4}(1 \ 2 \ 1)$ \rightarrow cubic B-spline



6

Subdivide ad nauseum?

After each split-average step, we are closer to the **limit curve**.

How many steps until we reach the final (limit) position? ∞

Can we push a vertex to its limit position without infinite subdivision? Yes!

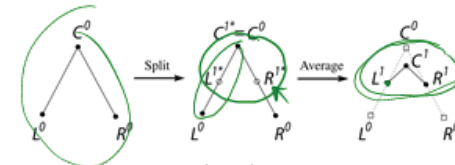
7

Local subdivision matrix

Consider the cubic B-spline subdivision mask:

$$\begin{pmatrix} 1 & 1 & 1 \\ 4 & 2 & 4 \end{pmatrix}$$

Now consider what happens during splitting and averaging in a small neighborhood:



We can write equations that relate points at one subdivision level to points at the previous:

$$L^{1*} = \frac{1}{2}(L^0 + C^0) \quad R^{1*} = \frac{1}{2}(C^0 + R^0)$$

$$L^1 = \frac{1}{4}L^0 + \frac{1}{2} \cdot \frac{1}{2}(L^0 + C^0) + \frac{1}{4}C^0 = \frac{1}{2}L^0 + \frac{1}{4}C^0$$

$$R^1 = \frac{1}{2}(C^0 + R^0)$$

$$C^1 = \frac{1}{4} \cdot \frac{1}{2}(L^0 + C^0) + \frac{1}{2}C^0 + \frac{1}{4} \cdot \frac{1}{2}(C^0 + R^0) = \frac{1}{8}L^0 + \frac{3}{4}C^0 + \frac{1}{8}R^0$$

8

Local subdivision matrix

We can write this as a recurrence relation in matrix form:

$$\begin{pmatrix} L^j \\ C^j \\ R^j \end{pmatrix} = \begin{pmatrix} 1/2 & 1/2 & 0 \\ 1/8 & 3/4 & 1/8 \\ 0 & 1/2 & 1/2 \end{pmatrix} \begin{pmatrix} L^{j-1} \\ C^{j-1} \\ R^{j-1} \end{pmatrix}$$

$$Q^j = S Q^{j-1}$$

Where the L, R, C 's are (for convenience) *row* vectors and S is the **local subdivision matrix**.

We can expand these row vectors:

$$\begin{pmatrix} L_x^j & L_y^j \\ C_x^j & C_y^j \\ R_x^j & R_y^j \end{pmatrix} = \begin{pmatrix} 1/2 & 1/2 & 0 \\ 1/8 & 3/4 & 1/8 \\ 0 & 1/2 & 1/2 \end{pmatrix} \begin{pmatrix} L_x^{j-1} & L_y^{j-1} \\ C_x^{j-1} & C_y^{j-1} \\ R_x^{j-1} & R_y^{j-1} \end{pmatrix}$$

and now think in terms of the behaviors of the x and y components, treating them as column vectors:

$$\begin{bmatrix} X^j & Y^j \end{bmatrix} = S \begin{bmatrix} X^{j-1} & Y^{j-1} \end{bmatrix}$$

9

Local subdivision matrix, cont'd

Let's focus on just the behavior of the x components:

$$\begin{pmatrix} L_x^j \\ C_x^j \\ R_x^j \end{pmatrix} = \begin{pmatrix} 1/2 & 1/2 & 0 \\ 1/8 & 3/4 & 1/8 \\ 0 & 1/2 & 1/2 \end{pmatrix} \begin{pmatrix} L_x^{j-1} \\ C_x^{j-1} \\ R_x^{j-1} \end{pmatrix}$$

$$X^j = S X^{j-1}$$

(The analysis of the y component will be the same.)

Tracking the x components through subdivision:

$$X^j = S X^{j-1} = S \cdot S X^{j-2} = S \cdot S \cdot S X^{j-3} = \dots = S^j X^0$$

The limit position of the x 's is then:

$$X^\infty = \lim_{j \rightarrow \infty} S^j X^0$$

OK, so how do we apply a matrix an infinite number of times??

10

Eigenvectors and eigenvalues

To solve this problem, we need to look at the eigenvectors and eigenvalues of S . First, a review...

Let v be a vector such that:

$$Sv = \lambda v$$

We say that v is an eigenvector with eigenvalue λ

An $n \times n$ matrix can have n eigenvalues and eigenvectors:

$$\begin{aligned} S v_1 &= \lambda_1 v_1 \\ &\vdots \\ S v_n &= \lambda_n v_n \end{aligned}$$

If the eigenvectors are linearly independent (which means that S is *non-defective*), then they form a basis, and we can re-write X in terms of the eigenvectors:

$$X = \sum_i^n a_i v_i$$

11

To infinity, but not beyond...

Now let's apply the matrix to the vector X :

$$X^1 = S X^0 = S \sum_i^n a_i v_i = \sum_i^n a_i S v_i = \sum_i^n a_i \lambda_i v_i$$

Applying it j times:

$$X^j = S^j X = S^j \sum_i^n a_i v_i = \sum_i^n a_i S^j v_i = \sum_i^n a_i \lambda_i^j v_i$$

Let's assume the eigenvalues are non-negative and sorted so that:

$$\lambda_1 > \lambda_2 > \lambda_3 \geq \dots \geq \lambda_n \geq 0$$

(The form of the inequalities is important.) Now let j go to infinity:

$$X^\infty = \lim_{j \rightarrow \infty} S^j X^0 = \lim_{j \rightarrow \infty} \sum_i^n a_i \lambda_i^j v_i = \lim_{j \rightarrow \infty} a_1 \lambda_1^j v_1 + a_2 \lambda_2^j v_2 + \dots$$

If $\lambda_1 > 1$, then: *blows up*

If $\lambda_1 < 1$, then: *shrinks to 0*

If $\lambda_1 = 1$, then: $X^\infty = a_1 v_1$

12

Evaluation masks

What are the eigenvalues and eigenvectors of our cubic B-spline subdivision matrix?

$$\lambda_1 = 1 \quad \lambda_2 = \frac{1}{2} \quad \lambda_3 = \frac{1}{4}$$

$$\mathbf{v}_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \mathbf{v}_2 = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} \quad \mathbf{v}_3 = \begin{pmatrix} 2 \\ -1 \\ 2 \end{pmatrix}$$

We're OK! In fact, for proper subdivision matrices, the first eigenvector will always be $[1 \dots 1]^T$. Why?

So, the first thing we do is expand the x and y components of a vertex in terms of the eigenbasis:

$$x = \sum_1^n a_i \mathbf{v}_i \quad y = \sum_1^n b_i \mathbf{v}_i$$

Then, after infinite subdivision, the x components will end up at...

$$X = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

What about the y-coordinates?

$$Y = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$\alpha^j = \begin{bmatrix} a_1 \\ 0 \\ 0 \end{bmatrix}$$

13

$$X^j = a_1 \lambda_1^j \mathbf{v}_1 + a_2 \lambda_2^j \mathbf{v}_2 + a_3 \lambda_3^j \mathbf{v}_3$$

$$= a_1 \lambda_1^j \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + a_2 \lambda_2^j \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} + a_3 \lambda_3^j \begin{bmatrix} 2 \\ -1 \\ 2 \end{bmatrix}$$

$$\Delta x^j = R^j - C^j = a_1 \lambda_1^j (1-1) + a_2 \lambda_2^j (1-0) + a_3 \lambda_3^j (2-1)$$

$$= a_2 \lambda_2^j + 3 a_3 \lambda_3^j$$

$$\Delta y^j = b_2 \lambda_2^j + 3 b_3 \lambda_3^j$$

$$\vec{f} = \lim_{j \rightarrow \infty} \frac{\Delta \hat{Q}^j}{\|\Delta \hat{Q}^j\|} = \frac{(a_2 \lambda_2^j + 3 a_3 \lambda_3^j, b_2 \lambda_2^j + 3 b_3 \lambda_3^j)}{\|(a_2 \lambda_2^j + 3 a_3 \lambda_3^j, b_2 \lambda_2^j + 3 b_3 \lambda_3^j)\|}$$

$$= \lim_{j \rightarrow \infty} \frac{(a_2 \lambda_2^j, b_2 \lambda_2^j)}{\|(a_2 \lambda_2^j, b_2 \lambda_2^j)\|} = \frac{(a_2, b_2)}{\|(a_2, b_2)\|}$$

Evaluation masks, cont'd

To finish up, we need to compute a_1 . First, we can reorganize the expansion of X into the eigenbasis:

$$X^0 = a_1 \mathbf{v}_1 + a_2 \mathbf{v}_2 + \dots + a_n \mathbf{v}_n = \begin{bmatrix} \vdots \\ \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_n \\ \vdots \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \mathbf{V} \mathbf{A}$$

$n \times n$

We can then solve for the coefficients in this new basis:

$$\mathbf{A} = \mathbf{V}^{-1} X^0 \quad \mathbf{U} \equiv \mathbf{V}^{-1}$$

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \dots & \mathbf{u}_1^T & \dots \\ \dots & \mathbf{u}_2^T & \dots \\ \dots & \vdots & \dots \\ \dots & \mathbf{u}_n^T & \dots \end{bmatrix} X^0$$

Now we can compute the limit position of the x-coordinate:

$$x_c^\infty = a_1 = \mathbf{u}_1^T X^0$$

We call \mathbf{u}_i the **evaluation mask**.

Evaluation masks, cont'd

Note that we need not start with the 0th level control points and push them to the limit.

If we subdivide and average the control polygon j times, we can push the vertices of the refined polygon to the limit as well:

$$X^\infty = \mathbf{S}^\infty X^j = \mathbf{u}_1^T X^j$$

The same result obtains for the y-coordinate:

$$Y^\infty = \mathbf{S}^\infty Y^j = \mathbf{u}_1^T Y^j$$

Left eigenvectors

What are these \mathbf{u} -vectors? Consider the eigenvector relation:

$$\mathbf{S} \mathbf{v}_i = \lambda_i \mathbf{v}_i$$

We can re-write this as a matrix (we'll use the 3x3 case for illustration here):

$$\mathbf{S} [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \mathbf{v}_3] = [\lambda_1 \mathbf{v}_1 \quad \lambda_2 \mathbf{v}_2 \quad \lambda_3 \mathbf{v}_3]$$

$$\mathbf{S} [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \mathbf{v}_3] = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \mathbf{v}_3] \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

$$\mathbf{S} \mathbf{V} = \mathbf{V} \mathbf{\Lambda}$$

where \mathbf{V} is the concatenation of the eigenvectors into a matrix and $\mathbf{\Lambda}$ is a diagonal matrix filled with the eigenvalues of \mathbf{S} .

Left eigenvectors (cont'd)

Now lets multiply both sides by V^{-1} from the left and right and then simplify:

$$\begin{aligned} V^{-1}(SV)V^{-1} &= V^{-1}(VA)V^{-1} \\ V^{-1}S &= AV^{-1} \\ U &= AV^{-1} \end{aligned}$$

where $U \equiv V^{-1}$. If we "de-construct" this relation, we get:

$$\begin{aligned} U &= AV^{-1} \\ \begin{bmatrix} u_1^T \\ u_2^T \\ u_3^T \end{bmatrix} S &= \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} u_1^T \\ u_2^T \\ u_3^T \end{bmatrix} \\ \begin{bmatrix} u_1^T \\ u_2^T \\ u_3^T \end{bmatrix} S &= \begin{bmatrix} \lambda_1 u_1^T \\ \lambda_2 u_2^T \\ \lambda_3 u_3^T \end{bmatrix} \end{aligned}$$

Thus, we find that the u -vectors obey the relation:

$$u_i^T S = \lambda_i u_i^T$$

These are the "left eigenvectors" of S . (Alternatively, they are the eigenvectors of S^T .)

17

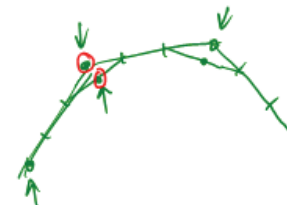
Recipe for subdivision curves

The evaluation mask for the cubic B-spline is:

$$\begin{pmatrix} 1 & 2 & 1 \\ 6 & 3 & 6 \end{pmatrix}$$

Now we can cook up a simple procedure for creating subdivision curves:

- Subdivide (split+average) the control polygon a few times. Use the averaging mask.
- Push the resulting points to the limit positions. Use the evaluation mask.



18

Tangent analysis

What is the tangent to the cubic B-spline curve?

First, let's consider how we represent the x and y coordinate neighborhoods:

$$\begin{aligned} X^0 &= a_1 v_1 + a_2 v_2 + a_3 v_3 \\ Y^0 &= b_1 v_1 + b_2 v_2 + b_3 v_3 \end{aligned}$$

We can view the point neighborhoods then as:

$$\begin{aligned} Q^0 &= \begin{bmatrix} X^0 & Y^0 \end{bmatrix} = [a_1 v_1 \quad b_1 v_1] + [a_2 v_2 \quad b_2 v_2] + [a_3 v_3 \quad b_3 v_3] \\ &= v_1 [a_1 \quad b_1] + v_2 [a_2 \quad b_2] + v_3 [a_3 \quad b_3] \end{aligned}$$

After j subdivisions, we would get:

$$\begin{aligned} Q^j &= S^j \{ v_1 [a_1 \quad b_1] + v_2 [a_2 \quad b_2] + v_3 [a_3 \quad b_3] \} \\ &= \lambda_1^j v_1 [a_1 \quad b_1] + \lambda_2^j v_2 [a_2 \quad b_2] + \lambda_3^j v_3 [a_3 \quad b_3] \end{aligned}$$

We can write this more explicitly as:

$$\begin{bmatrix} L^j \\ C^j \\ R^j \end{bmatrix} = \lambda_1^j \begin{bmatrix} v_{1L} \\ v_{1C} \\ v_{1R} \end{bmatrix} [a_1 \quad b_1] + \lambda_2^j \begin{bmatrix} v_{2L} \\ v_{2C} \\ v_{2R} \end{bmatrix} [a_2 \quad b_2] + \lambda_3^j \begin{bmatrix} v_{3L} \\ v_{3C} \\ v_{3R} \end{bmatrix} [a_3 \quad b_3]$$

19

Tangent analysis (cont'd)

The tangent to the curve is along the direction:

$$t = \lim_{j \rightarrow \infty} (R^j - C^j)$$

What's wrong with this definition?

Instead, we'll find the *normalized* tangent direction:

$$t = \lim_{j \rightarrow \infty} \frac{R^j - C^j}{\|R^j - C^j\|}$$

Now, let's look at the "right" and "center" points in isolation:

$$\begin{aligned} R^j &= \lambda_1^j v_{1,R} [a_1 \quad b_1] + \lambda_2^j v_{2,R} [a_2 \quad b_2] + \lambda_3^j v_{3,R} [a_3 \quad b_3] \\ C^j &= \lambda_1^j v_{1,C} [a_1 \quad b_1] + \lambda_2^j v_{2,C} [a_2 \quad b_2] + \lambda_3^j v_{3,C} [a_3 \quad b_3] \end{aligned}$$

The difference between these is:

$$\begin{aligned} R^j - C^j &= \lambda_1^j (v_{1,R} - v_{1,C}) [a_1 \quad b_1] + \\ &\quad \lambda_2^j (v_{2,R} - v_{2,C}) [a_2 \quad b_2] + \lambda_3^j (v_{3,R} - v_{3,C}) [a_3 \quad b_3] \\ &= \lambda_2^j (v_{2,R} - v_{2,C}) [a_2 \quad b_2] + \lambda_3^j (v_{3,R} - v_{3,C}) [a_3 \quad b_3] \end{aligned}$$

20

The tangent mask

And now computing the tangent:

$$\begin{aligned}
 \lim_{j \rightarrow \infty} \frac{R^j - C^j}{\|R^j - C^j\|} &= \lim_{j \rightarrow \infty} \frac{\lambda_2^j (v_{2,R} - v_{2,C}) [a_2 \ b_2] + \lambda_1^j (v_{1,R} - v_{1,C}) [a_1 \ b_1]}{\|\lambda_2^j (v_{2,R} - v_{2,C}) [a_2 \ b_2] + \lambda_1^j (v_{1,R} - v_{1,C}) [a_1 \ b_1]\|} \\
 &= \lim_{j \rightarrow \infty} \frac{(v_{2,R} - v_{2,C}) [a_2 \ b_2] + \left(\frac{\lambda_2}{\lambda_1}\right)^j (v_{1,R} - v_{1,C}) [a_1 \ b_1]}{\|(v_{2,R} - v_{2,C}) [a_2 \ b_2] + \left(\frac{\lambda_2}{\lambda_1}\right)^j (v_{1,R} - v_{1,C}) [a_1 \ b_1]\|} \\
 &= \frac{(v_{2,R} - v_{2,C}) [a_2 \ b_2]}{\|(v_{2,R} - v_{2,C}) [a_2 \ b_2]\|} \\
 &= \frac{[a_2 \ b_2]}{\|[a_2 \ b_2]\|} \\
 &= \frac{[\mathbf{u}_2^T \mathbf{x}^0 \quad \mathbf{u}_2^T \mathbf{y}^0]}{\|[\mathbf{u}_2^T \mathbf{x}^0 \quad \mathbf{u}_2^T \mathbf{y}^0]\|} \\
 &= \frac{\mathbf{u}_2^T \mathbf{Q}^0}{\|\mathbf{u}_2^T \mathbf{Q}^0\|}
 \end{aligned}$$

Thus, we can compute the tangent using the *second* left eigenvector! This analysis holds for general subdivision curves and gives us the **tangent mask**.

21

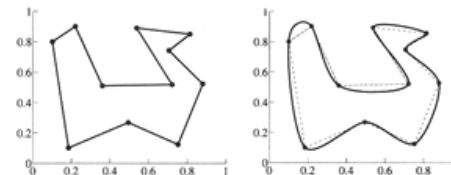
DLG interpolating scheme (1987)

Slight modification to subdivision algorithm:

- ♦ splitting step introduces midpoints
- ♦ averaging step *only changes midpoints*

For DLG (Dyn-Levin-Gregory), use:

$$r = \frac{1}{16}(-2, 5, 10, 5, -2)$$



Since we are only changing the midpoints, the points after the averaging step do not move.

22