Subdivision curves

Idea:
- repeatedly refine the control polygon
  \[ P_0 \rightarrow P_1 \rightarrow P_2 \rightarrow \cdots \]
  \[ C = \lim_{n \to \infty} P_n \]
- curve is the limit of an infinite process

Chaikin’s algorithm

Chakin 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” neighbor (the averaging step)
- Go to the splitting step

Reading
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:

\[ r = (K, r_{n-1}, r_n, r_{n+1}, K) \]

In the case of Chaikin’s algorithm:

Lane-Riesenfeld algorithm (1980)

Use averaging masks from Pascal’s triangle:

\[ r = \frac{1}{2^n} \binom{n}{n} \binom{n}{1} \binom{n}{n-1} \]

Gives B-splines of degree \( n+1 \).

Subdivide ad nauseum?

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

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

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

Local subdivision matrix

Consider the cubic B-spline subdivision mask: \( \frac{1}{4} (1 \ 2 \ 1) \)

Now consider what happens during splitting and averaging:

Relating points at one subdivision level to points at the previous:

\[ Q_i^1 = \frac{1}{4} (Q_i^0 + Q_{i+1}^0) = \frac{1}{4} (4Q_i^0 + 4Q_{i+1}^0) \]

\[ Q_i^1 = \frac{1}{4} (Q_i^0 + 6Q_{i+1}^0 + Q_{i+2}^0) \]

\[ Q_i^1 = \frac{1}{4} (Q_{i-1}^0 + Q_i^0 + Q_{i+1}^0) = \frac{1}{4} (4Q_i^0 + 4Q_{i+1}^0) \]
**Local subdivision matrix**

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

\[
\begin{pmatrix}
Q_l^j \\
Q_m^j \\
Q_n^j
\end{pmatrix} = \frac{1}{8} \begin{pmatrix} 4 & 4 & 0 \\ 1 & 6 & 1 \\ 0 & 4 & 4 \end{pmatrix} \begin{pmatrix}
Q_l^{j-1} \\
Q_m^{j-1} \\
Q_n^{j-1}
\end{pmatrix}
\]

\[Q^j = SQ^{j-1}\]

Q’s are row vectors and S is the local subdivision matrix.

Looking at the x-coordinate independently:

\[
\begin{pmatrix}
\frac{x^j_1}{x^1_r} \\
\frac{x^j_2}{x^2_r} \\
\frac{x^j_3}{x^3_r}
\end{pmatrix} = \frac{1}{8} \begin{pmatrix} 4 & 4 & 0 \\ 1 & 6 & 1 \\ 0 & 4 & 4 \end{pmatrix} \begin{pmatrix}
\frac{x^{j-1}_1}{x^{j-1}_r} \\
\frac{x^{j-1}_2}{x^{j-1}_r} \\
\frac{x^{j-1}_3}{x^{j-1}_r}
\end{pmatrix}
\]

\[X^j = SX^{j-1}\]

---

**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 \(\lambda\).

An \(n \times n\) matrix can have \(n\) eigenvalues and eigenvectors:

\[Sv_1 = \lambda_1 v_1, \quad \ldots, \quad Sv_n = \lambda_n v_n\]

\[M = \sum^n v_i, \quad X = \sum^n a_i v_i\]

\[Sv_n = \lambda_n v_n\]

For non-defective matrices, the eigenvectors form a basis, which means we can re-write \(X\) in terms of the eigenvectors.

---

**Local subdivision matrix, cont’d**

Tracking just the x components through subdivision:

\[X^j = SX^{j-1} = S \cdot S \cdot S \cdot \ldots \cdot S \cdot X^0 = S^j X^0\]

The limit position of the x’s is then:

\[X^\infty = \lim_{j \to \infty} S^j X^0\]

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

---

**To infinity, but not beyond…**

Now let’s apply the matrix to the vector \(X\):

\[SX = \sum^n a_i v_i = \sum^n a_i S v_i = \sum^n a_i \lambda_i v_i\]

Applying it \(j\) times:

\[S^j X = \sum^n a_i S^j v_i = \sum^n a_i \lambda_i^j v_i\]

Let’s assume the eigenvalues are sorted so that:

\[\lambda_1 > \lambda_2 > \lambda_3 \geq \lambda_4 \geq \lambda_5 \geq \ldots \geq \lambda_n\]

Now let \(j\) go to infinity.

If \(\lambda_i > 1\), then:

\[S^\infty X = \sum^n a_i \lambda_i^\infty v_i = a_i v_i\]

If \(\lambda_i < 1\), then:

If \(\lambda_i = 1\), then:
**Evaluation masks**

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

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

\[ v_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad v_2 = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} \quad v_3 = \begin{pmatrix} 2 \\ -1 \\ 2 \end{pmatrix} \]

We’re OK!

But where did the x-coordinates end up?

**Recipe for subdivision curves**

The evaluation mask for the cubic B-spline is:

\[ \frac{1}{6} \begin{pmatrix} 1 & 4 & 1 \end{pmatrix} \]

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.

Question: what is the tangent to the curve?

Answer: apply the second left eigenvector, \( u_2 \), as a tangent mask.

**DLG interpolating scheme (1987)**

Slight modification to algorithm:

- splitting step introduces midpoints
- averaging step only changes midpoints

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

\[ r = \frac{1}{16}(-2,6,10,6,-2) \]

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

What to take home:

- How to perform the splitting and averaging steps
- What an evaluation mask is and how to use it
- An appreciation for the mathematics behind subdivision curves