Classical Subdivision Surfaces

Lecture #14: Thursday, 13 Nov 1997
Lecturer: Denis Zorin
Scribe: Mark Wang
Reviewer: Hoofar Razavi

In this lecture we discuss the basic ideas of the analysis of subdivision, using subdivision curves as an example, and introduce the a simple subdivision scheme for surfaces – Loop scheme.

1 Subdivision of Curves near Extraordinary Points

Last time we have shown that the uniform B-spline curves can be thought of as a special case of subdivision curves. So far, we have seen only examples for which we use a fixed set of coefficients to compute the control points everywhere. The coefficients define the appearance of the curve, for example, whether it is smooth or has sharp corners. It is possible to control the appearance of the curve by modifying the subdivision coefficients locally.

Consider the four-point curve interpolation algorithm from the previous lecture. Mark a single control point of the curve. Suppose that we change the coefficients that are used to compute new control points near a marked control point on the coarsest level. For example, we can replace the coefficients $-1/16$, $9/16$, $9/16$, $-1/16$ of the four-point scheme by the coefficients 0, $1/2$, $1/2$, 0, whenever we compute a new control point adjacent to the marked point. As a result, we introduce a corner into the limit curve at the marked point.

Changing the coefficients of subdivision for computing a fixed number of neighbors of a marked point on each level does not affect the smoothness of the curve outside of a arbitrarily small neighborhood of the control point.

To determine *local* properties of a subdivision curve, we do not need the whole infinite vector of control points or the infinite matrix describing subdivision of the entire curve. Smoothness is a local property of a curve, which means that we can consider an arbitrarily small piece of the curve around a given point, and still be able to determine if the curve is smooth at that point or not.

It turns out that for the four-point algorithm, we need to consider only 7 control points; these 7 points completely define the piece of the curve around a control point. We can consider a set of 7 control points *on any subdivision level*, as we do not care how small our piece of the curve is. Note that we can compute the positions of the seven control points on level $j+1$ from the positions of similar seven control points on level j, using a 7×7 submatrix $\mathbf{S}$ of the infinite subdivision matrix.

The local subdivision matrix for the four-point scheme is:

$$\begin{pmatrix} c_{-3}^{j+1} \\ c_{-2}^{j+1} \\ c_{-1}^{j+1} \\ c_{0}^{j+1} \\ c_{1}^{j+1} \\ c_{2}^{j+1} \\ c_{3}^{j+1} \end{pmatrix} = \begin{pmatrix} -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} \end{pmatrix} \begin{pmatrix} c_{-3}^{j} \\ c_{-2}^{j} \\ c_{-1}^{j} \\ c_{0}^{j} \\ c_{1}^{j} \\ c_{2}^{j} \\ c_{3}^{j} \end{pmatrix}$$

As we iteratively apply the subdivision $\mathbf{S}$ to the initial vector for control points, we see that each of the 7 control points becomes closer to the central control point c_0.

Some properties of the curves generated by subdivision can be inferred from the properties of the subdivision matrix. In particular, smoothness properties of the curve are related to the eigenstructure of the subdivision matrix.

2 Properties of the Subdivision Matrix

2.1 Preliminaries

Recall from linear algebra that an *eigenvector* x of the matrix $\mathbf{M}$ is a non-zero vector such that $\mathbf{Mx} = \lambda \mathbf{x}$, where λ is a scalar; λ is the *eigenvalue* corresponding to $\mathbf{x}$.

Assume $\mathbf{S}$ has real eigenvectors $x_0, x_1, \ldots, x_6$ which form a basis, with corresponding real eigenvalues $\lambda_0 \geq \lambda_1 \geq \ldots \geq \lambda_6$. We can then write any vector x of length 7 as a linear combination of eigenvectors:

$$x = \sum_{i=0}^{n} a_i x_i$$

Similarly, we can write a decomposition of this type for a vector $\mathbf{c}$ of 7 *2-D points* rather than single numbers. In this case each "coefficient" $\mathbf{a}_i$ is a single 2-D point. The eigenvectors $\mathbf{x_0}, \ldots, \mathbf{x_6}$ are simply vectors of 7 real numbers. The expression $\mathbf{c} = \sum_{i=0}^{6} \mathbf{a}_i \mathbf{x}_i$, where $\mathbf{c}$ is a vector of 2D-points can be written more explicitly as

$$\mathbf{c} = \sum_{i=0}^{6} \begin{pmatrix} x_i^0 \mathbf{a}_i \\ x_i^1 \mathbf{a}_i \\ x_i^2 \mathbf{a}_i \\ x_i^3 \mathbf{a}_i \\ x_i^4 \mathbf{a}_i \\ x_i^5 \mathbf{a}_i \\ x_i^6 \mathbf{a}_i \end{pmatrix}$$

where $x_i^j \in \mathbf{R}$ represents the j-th component of eigenvector $\mathbf{x}_i$.

In the basis of eigenvectors we can easily compute the result of application of the subdivision matrix to a vector of control points, that is, the control points on the next level:

$$\begin{aligned}
\mathbf{S}c_0 &= \mathbf{S}\sum_{i=0}^{6}\mathbf{a}_i x_i \\
&= \sum_{i=0}^{6}\mathbf{a}_i \mathbf{S} x_i \quad \text{by linearity of } \mathbf{S} \\
&= \sum_{i=0}\mathbf{a}_i \lambda_i x_i
\end{aligned}$$

Applying $\mathbf{S}$ n times, we obtain

$$\mathbf{S}^n c_0 = \sum_{i=0}^{6}\mathbf{a}_i \lambda_i^n x_i$$

2.2 Convergence of Subdivision

If $\lambda_i > 1$ for some i, $\mathbf{S}^n c_0$ clearly would diverge as n got larger and larger.

Hence, we can see that in order for the sequence $\mathbf{S}^n c_0$ to converge at all, it is necessary that all eigenvalues are at most 1. It is also possible to show that only a single eigenvalue may have magnitude 1.

2.3 Invariance under Affine Transformations

If we moved all the control points simultaneously by the same amount, we would expect the curve defined by these control points to move in the same way as a rigid object. In other words, the curve should be *invariant under distance-preserving transformations, such as translation and rotation.* It follows from linearity of subdivision that if subdivision is invariant with respect to distance-preserving transformations, it also should be invariant under any affine transformations. The family of affine transformations in addition to distance-preserving transformations, contains shears.

Let $\vec{1} = [1111111]^T$ Then $\vec{1} \cdot \mathbf{a}$ represents a displacement of our seven points by a vector $\mathbf{a}$.

Applying subdivision to the transformed points, we get

$$\begin{aligned}
\mathbf{S}(\mathbf{c}^j + \vec{1}\cdot\mathbf{a}) &= \mathbf{S}(\mathbf{c}^j) + \mathbf{S}(\vec{1}\cdot\mathbf{a}) \quad \text{by linearity of } \mathbf{S} \\
&= \mathbf{c}^{j+1} + \mathbf{S}(\vec{1}\cdot\mathbf{a})
\end{aligned}$$

We see that for translational invariance we need

$$\mathbf{S}(\vec{1}\cdot\mathbf{a}) = \vec{1}\cdot\mathbf{a}$$

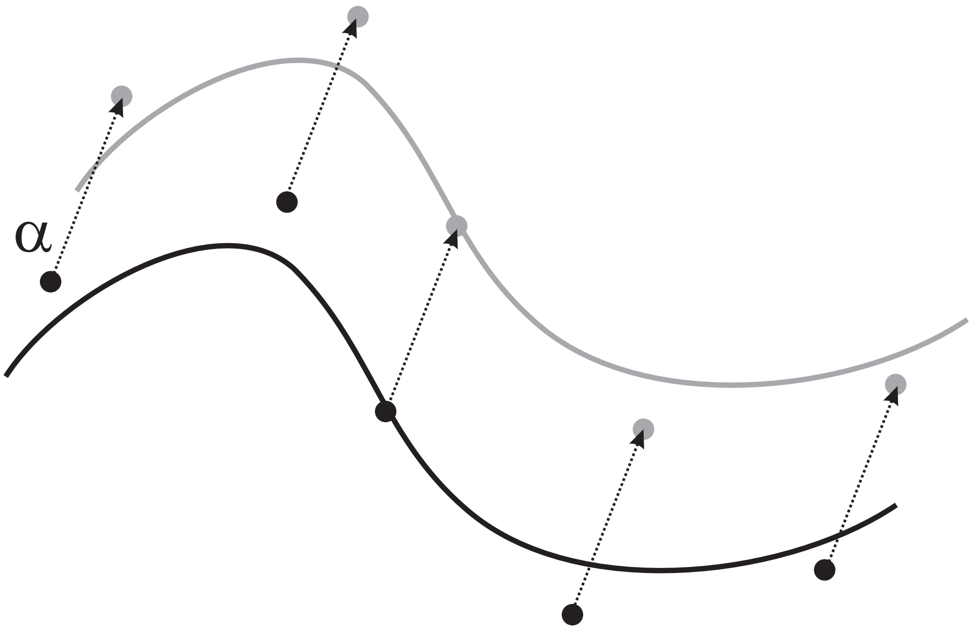

Figure 1: Invariance under translation

Therefore, $\vec{1}$ should be the eigenvector of $\mathbf{S}$ with eigenvalue $\lambda_0 = 1$.

2.4 Geometric Behavior of Repeated Subdivision

If we assume that λ_0 is 1, and all other eigenvalues are less than 1, we can choose our coordinate system in such a way that a_0 is a zero vector; then we have:

$$\mathbf{c}^n = \sum_{i=1}^{6} \mathbf{a}_i \lambda_i^n x_i$$

Dividing both sides by λ_1^n, we obtain:

$$\frac{\mathbf{c}^n}{\lambda_1^n} = \mathbf{a}_1 x_1 + \sum_{i=2}^{6} a_i \left(\frac{\lambda_i}{\lambda_1}\right)^n x_i$$

Assume that $\lambda_2, \ldots, \lambda_6 < \lambda_1$. In this case we see that the sum on the right approaches zero as $n \to \infty$. In other words the term corresponding to λ_1 will "dominate" the behavior of the vector of control points (decrease much slower than the other ones).

In the limit, we get a set of 7 points arranged along the vector $\mathbf{a}_1$. Geometrically, this is the tangent vector of our curve at our center point.

If there were two equal eigenvalues, say $\lambda_1 = \lambda_2$, as n increases, the points in the limit configuration will be linear combinations of two vectors $\mathbf{a}_1$ and $\mathbf{a}_2$, and in general would not be on the same line. This indicates that there will be no tangent vector at

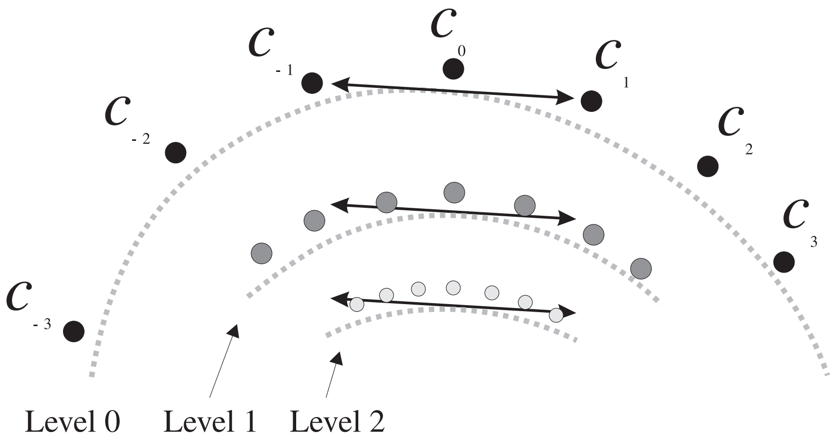

Figure 2: Repeatedly applying the subdivision matrix to our set of seven control points results in the control points converging to a configuration aligned with the tangent vector. The various subdivision levels have been offset vertically for clarity.

the central point. This leads us to the following condition, that, under some additional assumptions, is necessary for existence of the tangent:

All eigenvalues of S except $\lambda_0 = 1$ should be less than λ_1.

2.5 Summary

For our subdivision matrix **S** we desire the following characteristics:

- The eigenvectors should form a basis.
- The first eigenvalue λ_0 should be 1.
- The second eigenvalue λ_1 should be less than 1.
- All other eigenvalues should be less than λ_1.

3 Schemes for Subdivision Surfaces

We now focus our attention on the 2-D extension of curves, subdivision surfaces. There are a variety of subdivision schemes for surfaces, including

- Loop
- Catmull-Clark
- Doo-Sabin

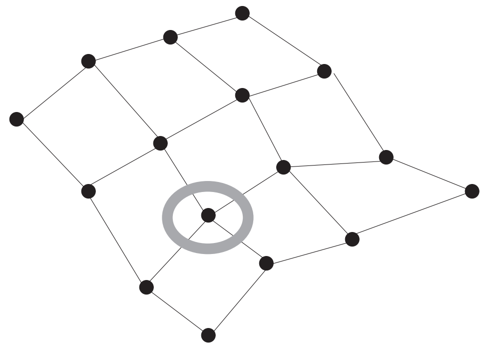

Figure 3: A tensor product B-spline.

- Butterfly

In this lecture, we focus on the Loop scheme.

3.1 Representation of Surfaces with Splines

In the 2-D case, we use a polyhedral surface as a *control mesh* analogous to the control points in the curve case. Additional restrictions may be imposed on the mesh. As an example, consider a *uniform tensor product B-spline* where all points not on the boundary of the control mesh are restricted to having exactly 4 neighbors.

While B-splines may seem like a good general solution at first as a control mesh for surfaces, what if we wanted to represent, say, a sphere? Unlike our B-spline, it is a *closed surface* with no boundary, and it is topologically impossible to represent it using such a mesh. For a mesh in which each vertex has 4 neighbors, the quantity $f - e + v$ (known as the *Euler characteristic*), where f is the number of faces, e is the number of edges, and v is the number of vertices, is 0. On the other hand, for a polyhedral mesh topologically equivalent to a sphere the Euler characteristic has to be 2.

3.2 3-Directional Quartic Splines

The Loop scheme is based on three-directional quartic Box splines, which are defined over regular triangular meshes (each face is a triangle, each vertex has 6 neighbors.) Other schemes are based on other types of splines: tensor product of quadratic B-splines for the Doo-Sabin scheme, and tensor product of cubic B-splines for Catmull-Clark.

The generating function for 3-directional Box spline is

$$f(z_1, z_2) = \frac{1}{16}(1 + z_1)^2(1 + z_2)^2(1 + z_1 z_2)^2$$

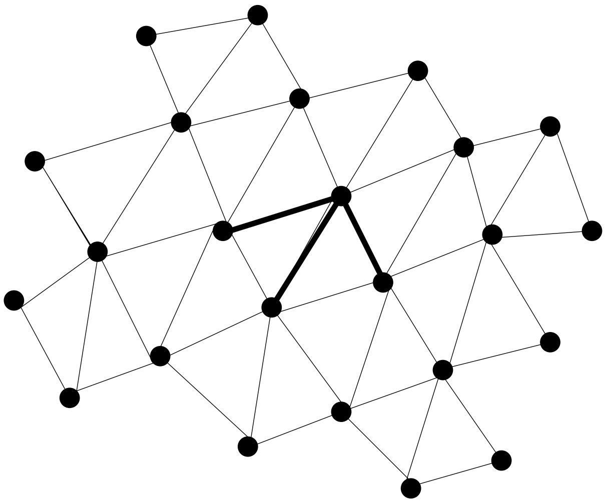

Figure 4: Part of a 3-directional quartic spline, with the three directions of symmetry highlighted.

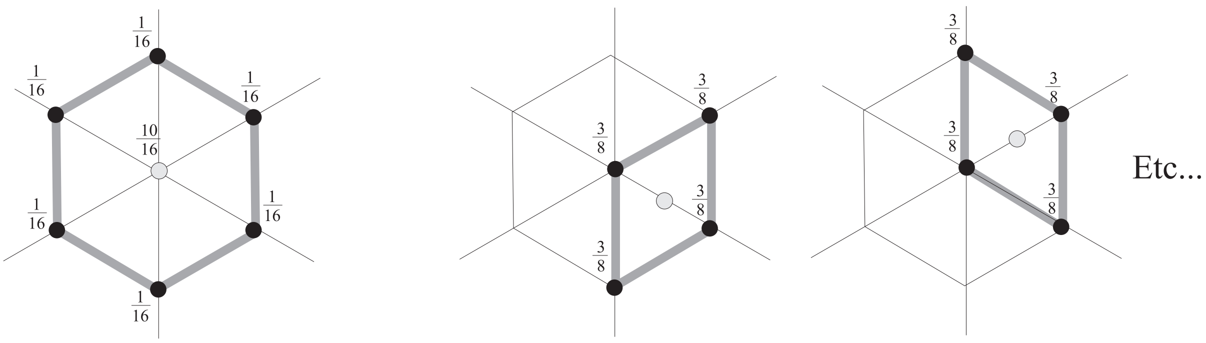

Figure 5: Loop subdivision rules for vertex (left) and edge (right) points at a regular vertex of degree 6.

3.3 Rules for Subdivision

Similar to the one-dimensional case, for the Loop scheme we need 2 rules, a "even" one (for adding new control points) and a "odd" one (for modifying existing ones.) Consider the case where each vertex has six neighbors (has degree 6).

New vertices are created by splitting the edges of the old mesh. Each old edge gives rise to two new edges and three more edges per triangle are created by connecting the new vertices. Each old triangle is subdivided in four.

Our challenge is to make the scheme work for arbitrary triangular meshes. We cannot simply use the spline rules, because they assume that each vertex has 6 neighbors. How do we update the position of vertices with degree other than 6 (known as *extraordinary*

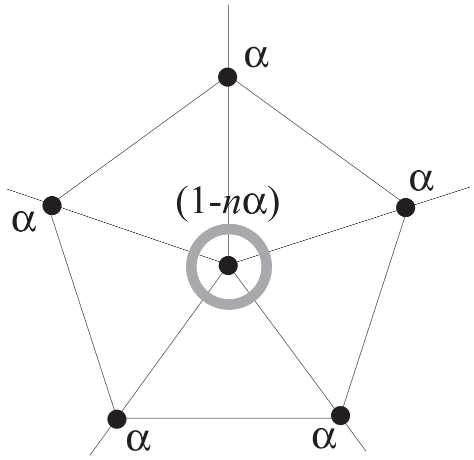

Figure 6: At an extraordinary point, we need to alter our vertex subdivision rule.

vertices)?

Observe that the rule for the spline is

$$c_0^{j+1} = \frac{5}{8}\mathbf{c}_0^j + \frac{1}{16}\sum_{i=1}^{6}\mathbf{c}_i^j$$

where $\mathbf{c}_0$ is the vertex that we recompute, and c_i, $i = 1\ldots 6$ are the immediate neighbors. This rule can be naturally generalized to the case of n neighbors as

$$c_0^{j+1} = (1-\alpha n)\mathbf{c}_0^j + \alpha\sum_{i=1}^{6}\mathbf{c}_i^j$$

We need to set the coefficient of our center point to be $(1 - n\alpha)$ in order for the sum of the coefficients to add up to 1 which is required for affine invariance of the scheme. Note that changing a rule at an extraordinary point will only change the characteristics of the generated surface locally, so the analysis in this case is similar to the analysis that we did for subdivision curves with special rules near a point.

The choice of the parameter α in the formula above is not unique, and different considerations can be used to pick an appropriate value. The original value chosen by Loop was

$$\alpha = \frac{1}{n}\left(\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{n}\right)^2\right)$$

We can easily verify that in the case $n = 6$, α is $\frac{1}{16}$ as we would expect. However, the exact value of α does not matter too much: the generated surfaces are smooth for a range of α. For example, for $n > 3$, we can choose α to be simply $\frac{3}{8n}$; the resulting surfaces will be quite similar to those obtained using the original Loop's rules.