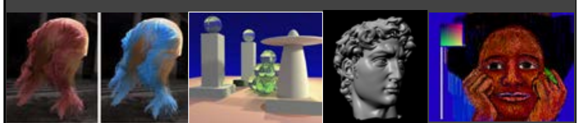
Advanced Computer Graphics

CSE 163 [Spring 2018], Lecture 10

Ravi Ramamoorthi

http://www.cs.ucsd.edu/~ravir



To Do

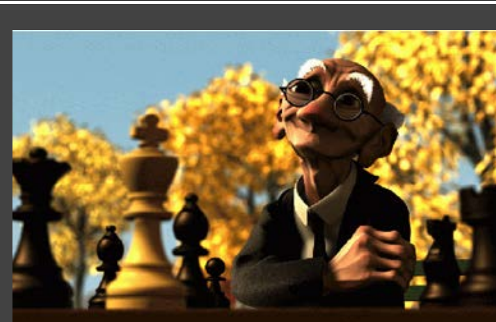
- Assignment 2 due May 18
  - Should already be well on way.
  - Contact us for difficulties etc.
  - This lecture is a "bonus": more advanced topic that is closer to the research frontier

Subdivision

- Was a very hot topic in computer graphics
- Brief survey lecture, quickly discuss ideas
- Detailed study quite sophisticated
  - See some of materials from readings

Advantages
  - Simple (only need subdivision rule)
  - Local (only look at nearby vertices)
  - Arbitrary topology (since only local)
  - No seams (unlike joining spline patches)

Video: Geri's Game (outside link)

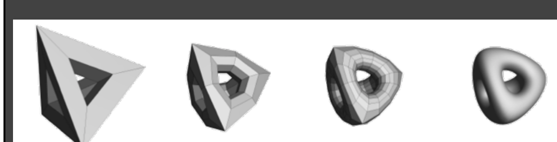


Outline

- *Basic Subdivision Schemes*
- Analysis of Continuity
- Exact and Efficient Evaluation (Stam 98)

Subdivision Surfaces

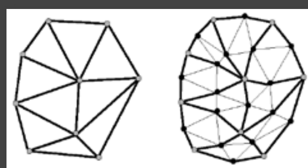
- Coarse mesh & subdivision rule
  - Smooth surface = limit of sequence of refinements



[Zorin & Schröder]

Key Questions

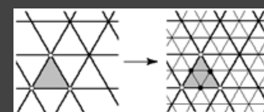
- How to refine mesh?
- Where to place new vertices?
  - Provable properties about limit surface



[Zorin & Schröder]

Loop Subdivision Scheme

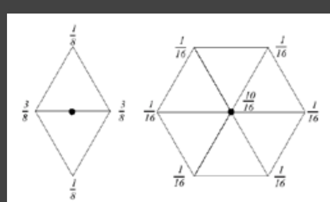
- How refine mesh?
  - Refine each triangle into 4 triangles by splitting each edge and connecting new vertices



[Zorin & Schröder]

Loop Subdivision Scheme

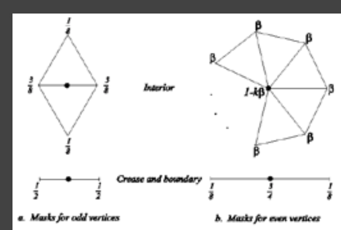
- Where to place new vertices?
  - Choose locations for new vertices as weighted average of original vertices in local neighborhood



[Zorin & Schröder]

Loop Subdivision Scheme

- Where to place new vertices?
  - Rules for *extraordinary vertices* and *boundaries*:



[Zorin & Schröder]

Loop Subdivision Scheme

Choose β by analyzing continuity of limit surface

- Original Loop

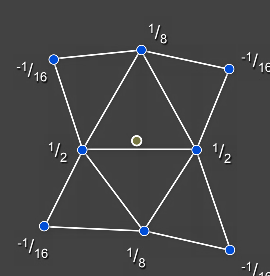
$$\beta = \frac{1}{n}\left(\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{n}\right)^2\right)$$

- Warren

$$\beta = \begin{cases} \frac{3}{8n} & n > 3 \\ \frac{3}{16} & n = 3 \end{cases}$$

Butterfly Subdivision

- Interpolating subdivision: larger neighborhood

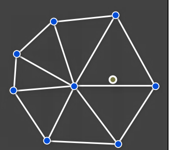


Modified Butterfly Subdivision

Need special weights near extraordinary vertices

- For n = 3, weights are $^5/_{12}$, $^{-1}/_{12}$, $^{-1}/_{12}$
- For n = 4, weights are 3/8, 0, -1/8, 0
- For n ≥ 5, weights are

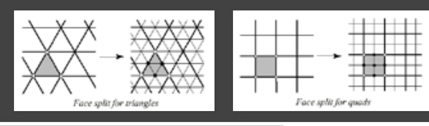
$$\frac{1}{n}\left(\frac{1}{4}+\cos\frac{2\pi j}{n}+\frac{1}{2}\cos\frac{4\pi j}{n}\right),\ j=0..n-1$$



- Weight of extraordinary vertex = 1 - Σ other weights

A Variety of Subdivision Schemes

- Triangles vs. Quads
- Interpolating vs. approximating



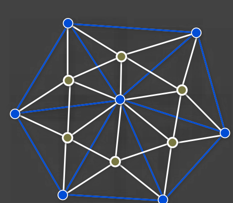
Face split		
	Triangular meshes	*Quad. meshes*
Approximating	Loop (C^2)	Catmull-Clark (C^2)
Interpolating	Mod. Butterfly (C^1)	Kobbelt (C^1)

Vertex split
Doo-Sabin, Midedge (C^1)
Biquartic (C^2)

[Zorin & Schröder]

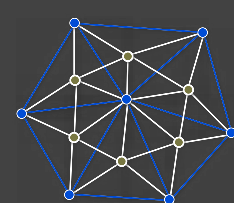
More Exotic Methods

- Kobbelt's subdivision:



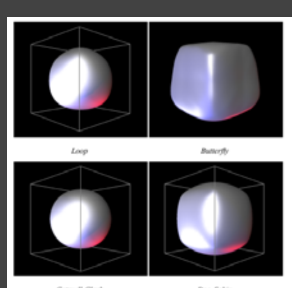
More Exotic Methods

- Kobbelt's subdivision:



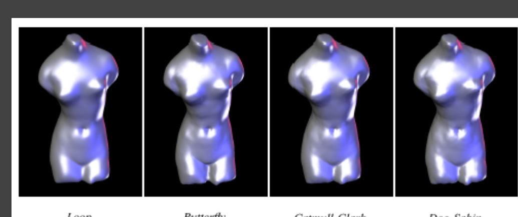
- Number of faces *triples* per iteration: gives finer control over polygon count

Subdivision Schemes



[Zorin & Schröder]

Subdivision Schemes



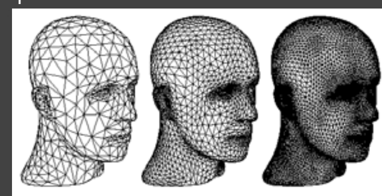
[Zorin & Schröder]

Outline

- Basic Subdivision Schemes
- *Analysis of Continuity*
- Exact and Efficient Evaluation (Stam 98)

Analyzing Subdivision Schemes

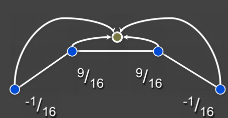
- Limit surface has provable smoothness properties



[Zorin & Schröder]

Analyzing Subdivision Schemes

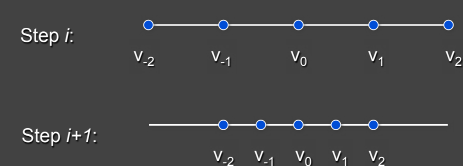
- Start with curves: 4-point interpolating scheme



(old points left where they are)

4-Point Scheme

- What is the support?



So, 5 new points depend on 5 old points

Subdivision Matrix

- How are vertices in neighborhood refined? (with vertex renumbering like in last slide)

$$\begin{pmatrix} v_{-2}^{(i+1)} \\ v_{-1}^{(i+1)} \\ v_{0}^{(i+1)} \\ v_{1}^{(i+1)} \\ v_{2}^{(i+1)} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} v_{-2}^{(i)} \\ v_{-1}^{(i)} \\ v_{0}^{(i)} \\ v_{1}^{(i)} \\ v_{2}^{(i)} \end{pmatrix}$$

Subdivision Matrix

- How are vertices in neighborhood refined? (with vertex renumbering like in last slide)

$$\vec{\mathbf{V}}^{(i+1)} = \mathbf{S}\vec{\mathbf{V}}^{(i)}$$

After *n* rounds: $\vec{\mathbf{V}}^{(n)} = \mathbf{S}^{n}\vec{\mathbf{V}}^{(0)}$

Convergence Criterion

$$\vec{V}^{(n)} = S^n \vec{V}^{(0)}$$

Expand in eigenvectors of **S**:

$$S = \sum_{i=0}^{4} \lambda_i \mathbf{e}_i \mathbf{e}_i^T$$

$$\vec{V}^{(0)} = \sum_{i=0}^{4} a_i \mathbf{e}_i$$

$$\vec{V}^{(n)} = \sum_{i=0}^{4} a_i \lambda_i^n \mathbf{e}_i$$

Criterion I: $|\lambda_i| \le 1$

Convergence Criterion

- What if all eigenvalues of **S** are < 1?
  - All points converge to 0 with repeated subdivision

Criterion II: $\lambda_0 = 1$

Translation Invariance

- For any translation t, want:

$$\begin{pmatrix} v_{-2}^{(i+1)} + t \\ v_{-1}^{(i+1)} + t \\ v_{0}^{(i+1)} + t \\ v_{1}^{(i+1)} + t \\ v_{2}^{(i+1)} + t \end{pmatrix} = S \begin{pmatrix} v_{-2}^{(i)} + t \\ v_{-1}^{(i)} + t \\ v_{0}^{(i)} + t \\ v_{1}^{(i)} + t \\ v_{2}^{(i)} + t \end{pmatrix}$$

$$\vec{V}^{(i+1)} + t\vec{1} = S\left(\vec{V}^{(i)} + t\vec{1}\right)$$

$$S\vec{1} = \vec{1}$$

Criterion III: $\mathbf{e}_0 = 1$, all other $|\lambda_i| < 1$

Smoothness Criterion

- Plug back in: $\vec{V}^{(n)} = a_0 \mathbf{e}_0 + \sum_{i=1}^{4} a_i \lambda_i^n \mathbf{e}_i$
- Dominated by largest λ_i
- Case 1: $|\lambda_1| > |\lambda_2|$

$$\vec{V}^{(n)} = a_0 \mathbf{e}_0 + a_1 \lambda_1^n \mathbf{e}_1 + (small)$$

  - Group of 5 points gets shorter
  - All points approach multiples of $\mathbf{e}_1$ → on a straight line
  - Smooth!

Smoothness Criterion

- Case 2: $|\lambda_1| = |\lambda_2|$
  - Points can be anywhere in space spanned by $\mathbf{e}_1$, $\mathbf{e}_2$
  - No longer have smoothness guarantee

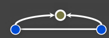
Criterion IV: Smooth iff $\lambda_0 = 1 > |\lambda_1| > |\lambda_i|$

Continuity and Smoothness

- So, what about 4-point scheme?
  - Eigenvalues = 1, $^1/_2$, $^1/_4$, $^1/_4$, $^1/_8$
  - $\mathbf{e}_0 = \mathbf{1}$
  - Stable ✓
  - Translation invariant ✓
  - Smooth ✓

2-Point Scheme

- In contrast, consider 2-point interpolating scheme



$^1/_2$ $^1/_2$

- Support = 3
- Subdivision matrix = $\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$

Continuity of 2-Point Scheme

- Eigenvalues = 1, $^1/_2$, $^1/_2$
- $\mathbf{e}_0$ **= 1**
- Stable ✓
- Translation invariant ✓
- Smooth ✗
  - *Not* smooth; in fact, this is piecewise linear

For Surfaces…

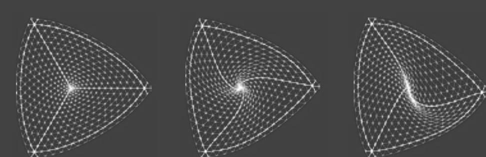
- Similar analysis: determine support, construct subdivision matrix, find eigenstuff
  - Caveat 1: separate analysis for each vertex valence
  - Caveat 2: consider more than 1 subdominant eigenvalue

Reif's smoothness condition: $\lambda_0 = 1 > |\lambda_1| \geq |\lambda_2| > |\lambda_i|$

- Points lie in subspace spanned by $\mathbf{e}_1$ and $\mathbf{e}_2$
  - If $|\lambda_1| \neq |\lambda_2|$, neighborhood stretched when subdivided, but remains 2-manifold

Fun with Subdivision Methods

Behavior of surfaces depends on eigenvalues



Real Complex Degenerate

(recall that symmetric matrices have real eigenvalues)

[Zorin]

Outline

- Basic Subdivision Schemes
- Analysis of Continuity
- *Exact and Efficient Evaluation (Stam 98)*

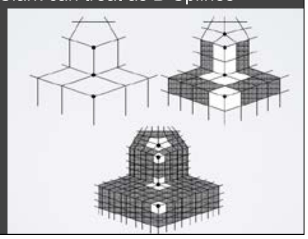
Slides courtesy James O'Brien

Practical Evaluation

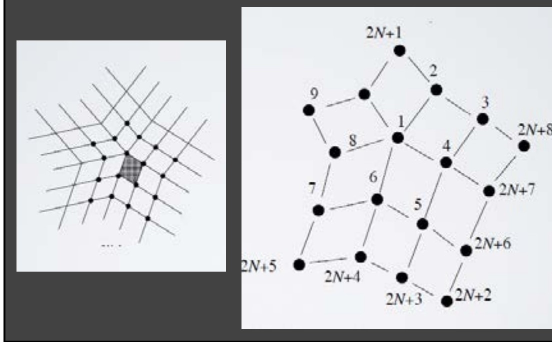
- Problems with Uniform Subdivision
  - Exponential growth of control mesh
  - Need several subdivisions before error is small
  - Ok if you are "drawing and forgetting", otherwise …
- (Exact) Evaluation at arbitrary points
- Tangent and other derivative evaluation needed
- Paper by Jos Stam SIGGRAPH 98 efficient method
  - Exact evaluation (essentially take out "subdivision")
  - Smoothness *analysis* methods used to *evaluate*

Isolated Extraordinary Points

- After 2+ subdivisions, isolated "extraordinary" points where irregular valence
- Regular region is usually easy
  - For example, Catmull Clark can treat as B-Splines

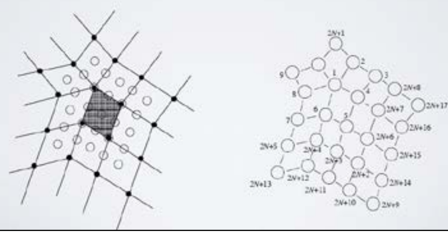


Isolated Extraordinary Points



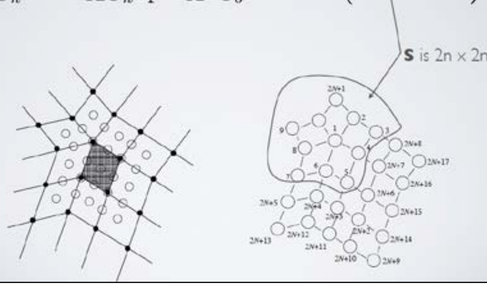
Subdivision Matrix

$$\mathbf{C}_1 = \mathbf{A}\mathbf{C}_0.$$
$$\mathbf{C}_n = \mathbf{A}\mathbf{C}_{n-1} = \mathbf{A}^n\mathbf{C}_0$$



Subdivision Matrix

$$\mathbf{C}_1 = \mathbf{A}\mathbf{C}_0.$$
$$\mathbf{C}_n = \mathbf{A}\mathbf{C}_{n-1} = \mathbf{A}^n\mathbf{C}_0 \qquad \mathbf{A} = \begin{pmatrix} \mathbf{S} & \mathbf{0} \\ \mathbf{S}_{11} & \mathbf{S}_{12} \end{pmatrix}$$



Eigen Space

$$\mathbf{C}_1 = \mathbf{A}\mathbf{C}_0.$$
$$\mathbf{C}_n = \mathbf{A}\mathbf{C}_{n-1} = \mathbf{A}^n\mathbf{C}_0 \qquad \mathbf{A} = \begin{pmatrix} \mathbf{S} & \mathbf{0} \\ \mathbf{S}_{11} & \mathbf{S}_{12} \end{pmatrix}$$

$$\mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{\Lambda} \longrightarrow \mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$$

$$\bar{\mathbf{C}}_n = \bar{\mathbf{A}}\mathbf{A}^{n-1}\mathbf{C}_0 = \bar{\mathbf{A}}\mathbf{V}\mathbf{\Lambda}^{n-1}\mathbf{V}^{-1}\mathbf{C}_0$$

$$\mathbf{s}_{k,n}(u,v) = \hat{\mathbf{C}}_0^T\mathbf{\Lambda}^{n-1}\left(\mathbf{P}_k\bar{\mathbf{A}}\mathbf{V}\right)^T\mathbf{b}(u,v)$$

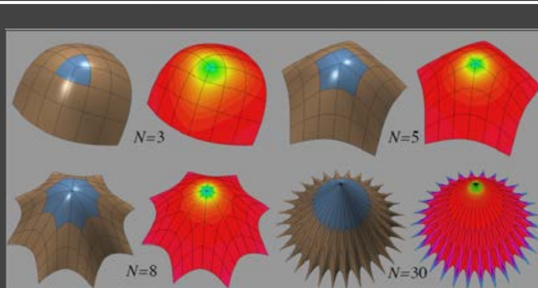
$$\hat{\mathbf{C}}_0 = \mathbf{V}^{-1}\mathbf{C}_0$$

Only depends on valence of extraordinary vertex.

Comments

- Computing Eigen-Vectors is tricky
  - See Jos' paper for details
  - He includes solutions for valence up to 500
- All eigenvalues are (abs) less than one
  - Except for lead value which is exactly one
  - Well defined limit behavior
- Exact evaluation allows "pushing to limit surface"

Curvature Plots



See Stam 98 for details

Summary

- Advantages:
  - Simple method for describing complex, smooth surfaces
  - Relatively easy to implement
  - Arbitrary topology
  - Local support
  - Guaranteed continuity
  - Multiresolution
- Difficulties:
  - Intuitive specification
  - Parameterization
  - Intersections

[Pixar]