Introduction to Subdivision and Normal Meshes

Olof Runborg

Numerical Analysis,
School of Computer Science and Communication, KTH

RTG Summer School on Multiscale Modeling and Analysis
University of Texas at Austin
2008-07-21 – 2008-08-08

Subdivision Surfaces

- Way to represent and approximate surfaces.
- Widely used in animated movies and computer games.

(Pictures from PIXAR.)

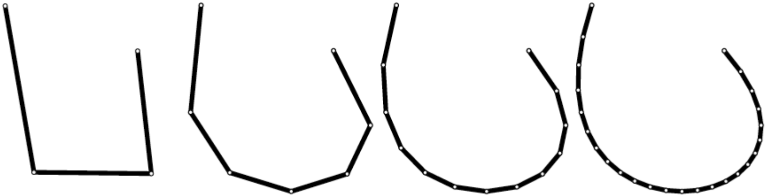
Subdivision Surfaces

- Way to represent and approximate surfaces.
- Widely used in animated movies and computer games.

(Pictures from PIXAR.)

Subdivision – Basic Idea

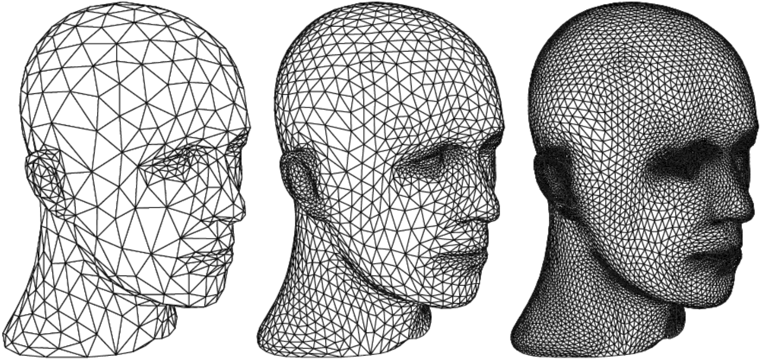
Use simple refinement rules to construct smooth surfaces/curves from a coarse initial mesh.



(Pictures from Schroder/Zorin.)

Subdivision – Basic Idea

Use simple refinement rules to construct smooth surfaces/curves from a coarse initial mesh.



(Pictures from Schroder/Zorin.)

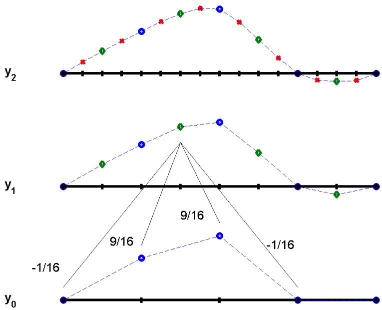
Subdivision

- Simple method for describing complex surfaces
- Arbitrary topology
- Scalability/multiresolution
- Local definition – efficiency, simplicity
- Smooth surfaces

Subdivision

Procedure to iteratively create smooth curves and surfaces.

Example in 1D: “4-point scheme:”



Curve $\sim \boldsymbol{y}_j = \{y_{j,k}\}$

Refinement: $\boldsymbol{y}_{j+1} = S\boldsymbol{y}_j$

S defined as

$$y_{j+1,2k} = y_{j,k}$$

(even points)

$$y_{j+1,2k+1} = \sum_{\ell=-1}^{2} s_\ell y_{j,k+\ell}$$

(odd points)

Subdivision, examples

More general, let S be a (linear, local, stationary) *subdivision operator* such that $\boldsymbol{y}_{j+1} = S\boldsymbol{y}_j$ and

$$y_{j+1,2k} = \sum_\ell s^e_\ell y_{j,k+\ell}, \qquad y_{j+1,2k+1} = \sum_\ell s^o_\ell y_{j,k+\ell}.$$

where $\boldsymbol{s}^o = \{s^o_\ell\}$ and $\boldsymbol{s}^e\{s^e_\ell\}$ are the odd/even masks.

S is *interpolating* if $\boldsymbol{s}^e = \{\delta_\ell\}$, i.e. $y_{j+1,2k} = y_{j,k}$.

Examples:

- "2-point" (linear), $\boldsymbol{s}^o = \frac{1}{2}[1,\ 1]$,
- "4-point" (cubic), $\boldsymbol{s}^o = \frac{1}{16}[-1,\ 9,\ 9,\ -1]$,
- "6-point", $\boldsymbol{s}^o = \frac{1}{256}[3,\ -25,\ 150,\ 150,\ -25,\ 3]$.

More Properties of Subdivision Operators

Order of S is $\mathcal{P}$ if S preserves p-degree polynomials for $0 \le p < \mathcal{P}$.

Derived subdivision schemes, $S^{[p]}$, act on divided differences of $\boldsymbol{y}_j$,

$$S^{[p]} \boldsymbol{y}_j^{[p]} = \boldsymbol{y}_{j+1}^{[p]}, \qquad \boldsymbol{y}^{[p]} = D_j^p \boldsymbol{y}_j.$$

(Also, $S^{[p]} = D_{j+1}^p S D_j^{-p}$.) $S^{[p]}$ well-defined for $p \le$ order of S.
A derived scheme $S^{[p]}$ is in general not interpolating even if S is.
Example: If S is the 4-point scheme, then

- $S^{[1]}$: $\boldsymbol{s}^e = [\frac{1}{8},\ 1,\ -\frac{1}{8}]$, $\boldsymbol{s}^o = [-\frac{1}{8},\ 1,\ \frac{1}{8}]$,
- $S^{[2]}$: $\boldsymbol{s}^e = [\frac{1}{2},\ \frac{1}{2}]$, $\boldsymbol{s}^o = [-\frac{1}{4},\ \frac{3}{2},\ -\frac{1}{4}]$.

Subdivision limit functions

Let $\tilde{y}_j(t)$ be linear interpolant of $\boldsymbol{y}_j$. Does $\phi(t)$ exist such that

$$\lim_{j\to\infty} \tilde{y}_j(t) \to \phi(t)?$$

If so, what is the regularity of $\phi(t)$?

Theorem

Let S be a subdivision scheme of order $\mathcal{P} \geq 1$. Assume C, μ exist such that

$$||S^{[p]^j}||_\infty \leq C\, 2^{\mu j}, \qquad \forall j \geq 0.$$

If $\mathcal{P} \geq p > \mu$, then $\exists \phi \in C^{p-\mu-\varepsilon}$ such that $\tilde{y}_j \to \phi$ uniformly.

Ex.:

- "2-point:" $||S^{[1]}||_\infty = 1$ gives $p = 1$, $\mu = 0$, so $\phi \in C^{1-\varepsilon}$
- "4-point:" $||S^{[3]}||_\infty = 2$ gives $p = 3$, $\mu = 1$, so $\phi \in C^{2-\varepsilon}$.
- "6-point", $\phi \in C^{2.83}$.

Subdivision in 2D – Surfaces

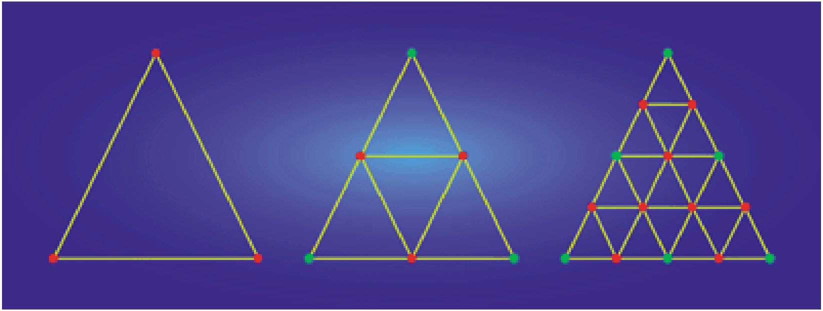
New challenges:

- Triangular/quadrilaterals instead of intervals
- Extraordinary vertices – varying number of neighbors
- Boundaries
- Piecewise smooth curves – creases, corners, cusps, etc.

Subdivision in 2D – Surfaces

Butterfly algorithm:

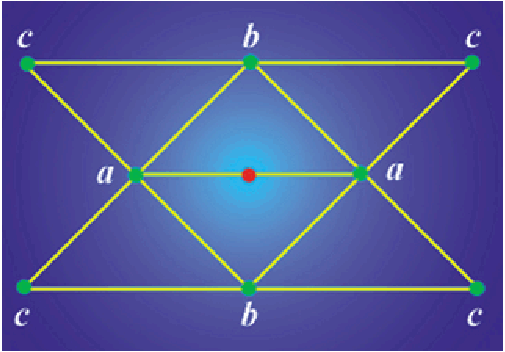
- Work with triangulated meshes.
- Split each face in four with each refinement.



Butterfly, cont.

- Interpolating: Old vertices stay the same.
- New vertex a weighted sum of surrounding vertices:

$$x_{\mathrm{new}} = \frac{1}{2}(a_1 + a_2) + \left(\frac{1}{8} + 2w\right)(b_1 + b_2) - \left(\frac{1}{16} + w\right)(c_1 + c_2).$$

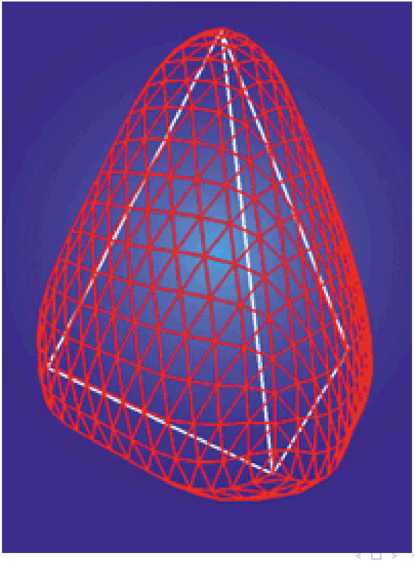


(w = tension parameter)

- Gives C^1 surfaces almost everywhere.

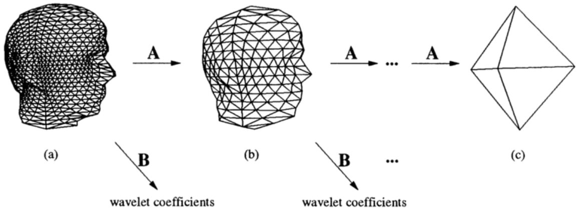
Butterfly, example

Initial mesh a pyramid:



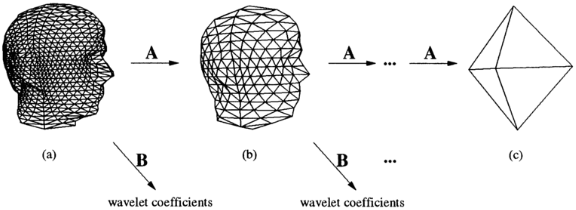
Multiresolution Meshes

Analysis:

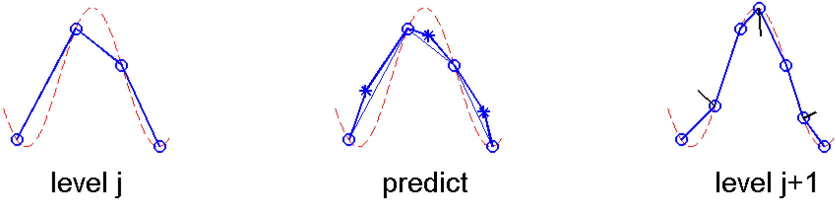


Multiresolution Meshes

Analysis:



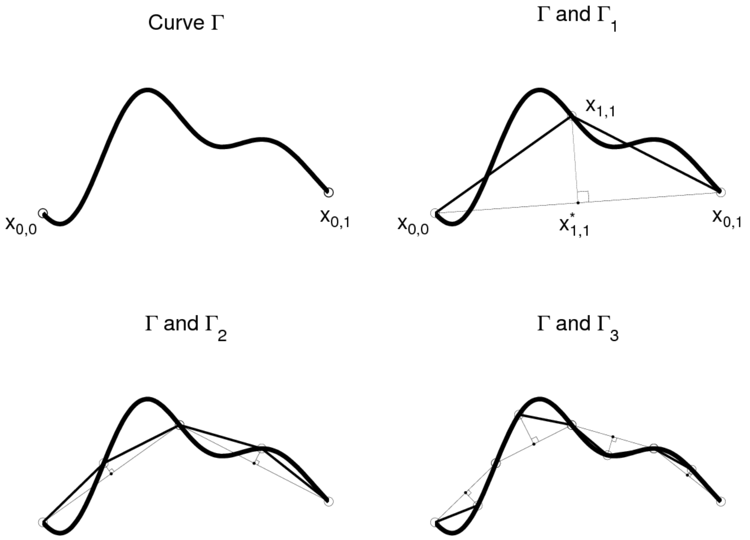
Synthesis:



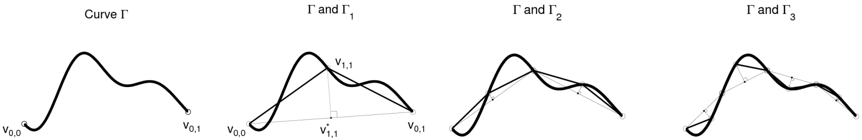
- Predict new vertices only based on previous level.
- Add wavelet coefficient (“detail vector”) to correct.

Normal meshes

Guskov, Khodakovsky, Schröder, Sweldens,. . .



Normal meshes



- Multiresolution meshes (predict + detail, at each level).
- Wavelet vectors align with *normal* direction defined by coarser levels.
- Purely scalar representation (1 float/vertex) possible. Good compression properties.
- No parameterization/connectivity info (improves compression rates)
- Scalar wavelet compression codes can be used.

Mathematical Issues

- Convergence,
- Decay of normal offsets,
- Regularity of normal parameterizaton,
- Stability.

Some results available in one dimension: Daubechies, Sweldens, O.R..

Suppose the curve is given by $\Gamma(s)$ with $s \in [0, 1]$.
Let the node points on level j be given by

$$x_{j,k} = \Gamma(k2^{-j}), \qquad 0 \le k \le 2^j.$$

(Note that $x_{j+1,2k} = x_{j,k}$.)
The wavelet details vectors are

$$w_{j,k} = x_{j+1,2k+1} - \frac{1}{2}(x_{j,k} + x_{j,k+1}), \qquad j \ge 0.$$

Normal meshes

Theory

Suppose the curve is given by $\Gamma(s)$ with $s \in [0,1]$.
Let the node points on level j be given by

$$x_{j,k} = \Gamma(k2^{-j}), \qquad 0 \leq k \leq 2^j.$$

(Note that $x_{j+1,2k} = x_{j,k}$.)
The wavelet details vectors are

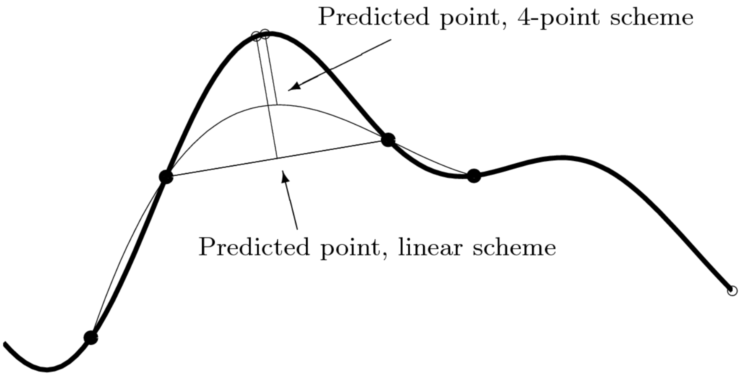
$$w_{j,k} = x_{j+1,2k+1} - \frac{1}{2}(x_{j,k} + x_{j,k+1}), \qquad j \geq 0.$$

Theorem (Daubechies,Sweldens,OR)

If $\Gamma \in C^2([0,1];\mathbb{R}^2)$ *then*

$$\Gamma_j \to \Gamma, \qquad |w_{j,k}| \leq C2^{-2j}, \qquad |x_{j,k+1} - x_{j,k}| \leq C'2^{-j}.$$

Higher order subdivision scheme as predictor



Will give faster decay of wavelet vectors and their time derivatives.

Higher order subdivision – Theory

For higher order subdivision normal wavelet vectors decay faster:

Theorem (Daubechies,Sweldens,OR)

If $\Gamma \in C^{Q+\varepsilon}([0,1];\mathbb{R}^2)$ *then*

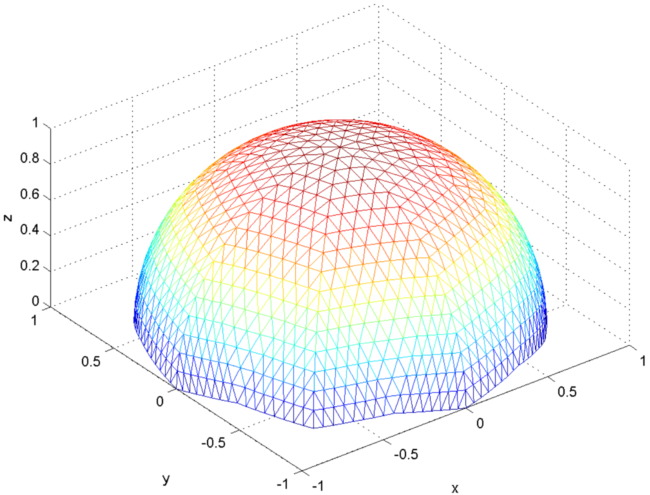
$$|w_{j,k}| \leq C2^{-(Q-\varepsilon)j},$$

where Q depends on the subdivision operator in a nontrivial way. Ex:

- *"4-point:"* $Q = 3$,
- *"6-point:"* $Q = 3.83$,
- *"8-point:"* $Q = 4.55$.

Two dimensions

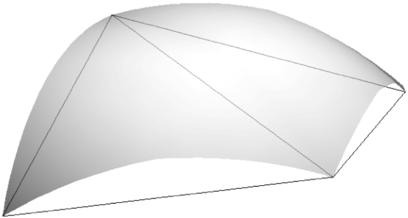
Triangulated wavefront:



Two dimensions

Simplified normal mesh construction

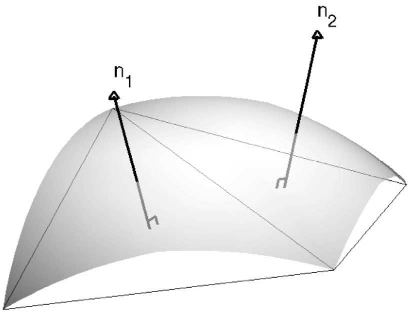
1. Start from two adjacent triangles.



Two dimensions

Simplified normal mesh construction

1. Start from two adjacent triangles.
2. Construct normals to the triangles, n_1 and n_2.

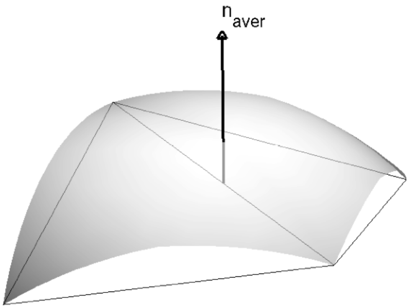


Two dimensions

Simplified normal mesh construction

1. Start from two adjacent triangles.
2. Construct normals to the triangles, n_1 and n_2.
3. Compute an average normal on the connecting edge:

$$n_{\text{aver}} = \frac{n_1 + n_2}{|n_1 + n_2|}$$



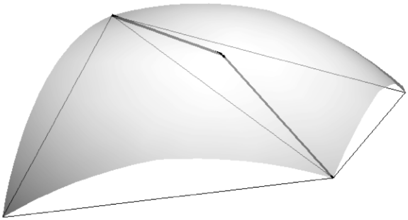
Two dimensions

Simplified normal mesh construction

1. Start from two adjacent triangles.
2. Construct normals to the triangles, n_1 and n_2.
3. Compute an average normal on the connecting edge:

$$n_{\text{aver}} = \frac{n_1 + n_2}{|n_1 + n_2|}$$

4. Find point where n_{aver} pierces surface.



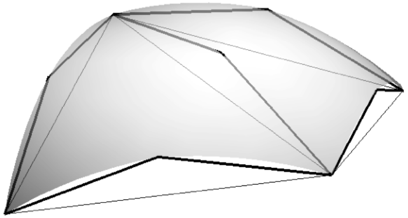
Two dimensions

Simplified normal mesh construction

1. Start from two adjacent triangles.
2. Construct normals to the triangles, n_1 and n_2.
3. Compute an average normal on the connecting edge:

$$n_{\text{aver}} = \frac{n_1 + n_2}{|n_1 + n_2|}$$

4. Find point where n_{aver} pierces surface.
5. Do same thing for all edges.



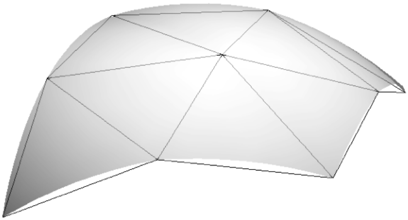
Two dimensions

Simplified normal mesh construction

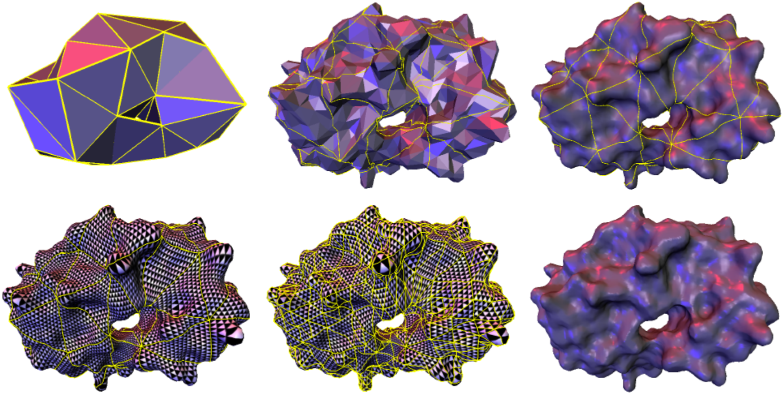
1. Start from two adjacent triangles.
2. Construct normals to the triangles, n_1 and n_2.
3. Compute an average normal on the connecting edge:

$$n_{\text{aver}} = \frac{n_1 + n_2}{|n_1 + n_2|}$$

4. Find point where n_{aver} pierces surface.
5. Do same thing for all edges.
6. Connect new points to a new finer triangulation.



Better results when using higher order subdivision schemes as predictors. Here the Butterfly scheme was used:



(Pictures from Guskov, Vidimce, Schröder, Sweldens.)

Two dimensions
Normal mesh example

Better results when using higher order subdivision schemes as predictors. Here the Butterfly scheme was used:

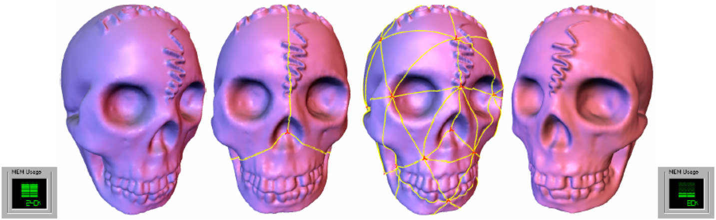


Figure 1: *Left: original mesh (3 floats/vertex). Middle: two stages of our algorithm. Right: normal mesh (1 float/vertex). (Skull dataset courtesy Headus, Inc.)*

(Pictures from Guskov, Vidimce, Schröder, Sweldens.)