# Mathematical tools 1

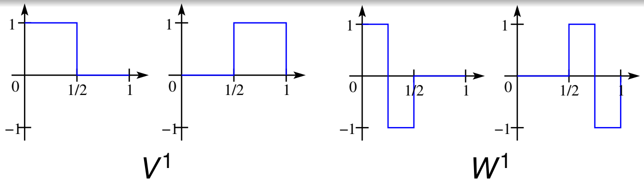
## Session 4

Franck HÉTROY

M2R IVR, October 26th 2006

# Interpolation and approximation

# Context




Haar: multiresolution analysis of piecewise-constant functions

- Any piecewise-constant function on $[0, 1]$ can be decomposed into a weighted sum of a scale factor and wavelets
- Opposite idea: a recursive subdivision of a Haar scale factor, using appropriate weights and wavelets, can lead to any piecewise-constant function on $[0, 1]$. This is called multiresolution synthesis.
- This is not sufficient for many applications (discontinuities) $\Rightarrow$ need for generalization

# Uniform subdivision

We focus on multiresolution synthesis.

Idea:

- Start from a piecewise-linear function $f^0$
- Repeatedly refine it, to produce a sequence of increasingly detailed functions $f^1, f^2, \ldots$
- These functions converge to a limit function $f = \lim_{j \to +\infty} f^j$

$\rightsquigarrow$ properties of $f$ (continuity, ...) ?

# "Corner-cutting" procedure

- $f^j$ piecewise-linear with vertices at $\frac{i}{2^j}$ (dyadic points)
  $\rightsquigarrow$ twice more control points at each stage
- Let us note $\forall i, \forall j, c_i^j = f^j(\frac{i}{2^j})$
- Splitting step:
  $\overline{c}_{2i}^j := c_i^{j-1}, \overline{c}_{2i+1}^j := \frac{1}{2}(c_i^{j-1} + c_{i+1}^{j-1})$
- Averaging step:
  $$c_i^j := \sum_k r_k \overline{c}_{i+k}^j$$
  $\rightsquigarrow$ matricial notation: $C^j = R\overline{C}^j$
- $r = (\ldots, r_{-1}, r_0, r_1, \ldots)$ = averaging mask
- Uniform subdivision scheme if same mask everywhere along the curve (i.e., $r$ independent from $i$)
- Stationary subdivision scheme if same mask used on each iteration (i.e., $r$ independent from $j$)

# Example: Chaikin's algorithm (1974)

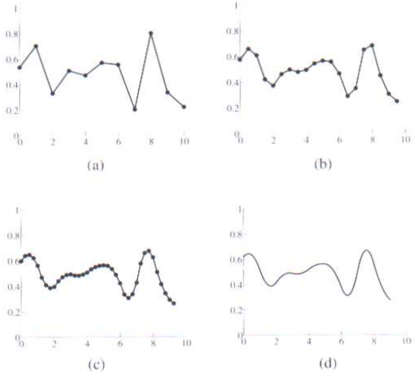
## Example

$r = (r_0, r_1) = \frac{1}{2}(1, 1)$. Start from some $f^0$ function with $\sim 10$ points and draw $f^1$ and $f^2$. How does the matrix $R$ look like ?

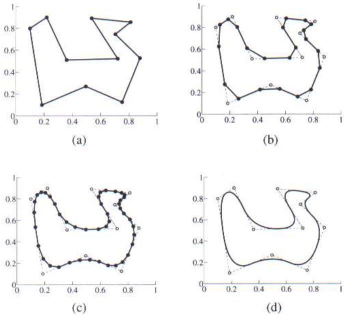
# Example: Chaikin's algorithm (1974)

## Example

$r = (r_0, r_1) = \frac{1}{2}(1, 1)$. Start from some $f^0$ function with $\sim 10$ points and draw $f^1$ and $f^2$. How does the matrix $R$ look like ?



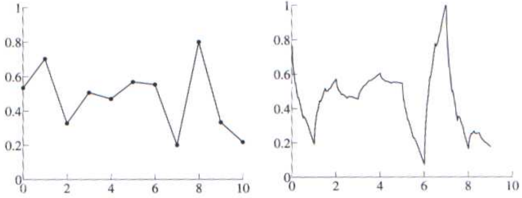
# Chaikin's algorithm for a closed parametric curve




- In 1975, Riesenfeld proved that curves generated by this algorithm are in fact uniform quadratic B-splines
- In 1980, Lane and Riesenfeld showed that it can be generalized to produce uniform B-splines of any degree using masks whose entries come from Pascal's triangle

# Other uniform subdivision schemes

- Linear B-splines: identity mask $r = (r_0) = (1)$
- Cubic B-splines: $r = (r_{-1}, r_0, r_1) = \frac{1}{4}(1, 2, 1)$
- Daubechies scheme: $r = (r_0, r_1) = \frac{1}{2}(1 + \sqrt{3}, 1 - \sqrt{3})$




Limit curve is nowhere differentiable (fractal)

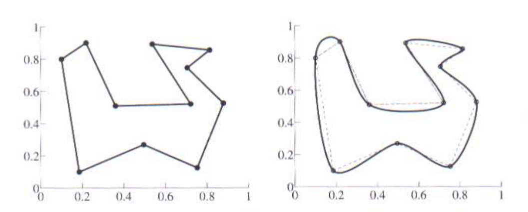
# Interpolation

- These are approximating schemes
- If we want interpolating schemes, we just need to change the averaging step:

$$c_i^j = \begin{cases} \overline{c}_i^j & \text{if } i \text{ is even} \\ \sum_k r_k \overline{c}_{i+k}^j & \text{if } i \text{ is odd} \end{cases}$$

## Example (Dyn/Levin/Gregory scheme, 1987)

$r = (r_{-2}, r_{-1}, r_0, r_1, r_2) = \frac{1}{16}(-2, 5, 10, 5, -2).$



# Subdivision and multiresolution synthesis

- Since each splitting and averaging step is linear w.r.t. the initial values $c_i^0$, each $f^j(x)$, thus $f(x)$, is a linear combination of the $c_i^0$:
$$f(x) = \sum_i c_i^0 \phi_i^0(x) = \cdots = \sum_i c_i^j \phi_i^j(x)$$
- Functions $\phi_i^j$ are to be found
- Let $V^j$ be the vector space generated by the $\phi_i^j$: we can easily show (next slide) that the $V^j$ are nested spaces: $V^0 \subset V^1 \subset \dots$
- As for Haar functions, these functions $\phi_i^j$ are called scale factors of $V^j$

# Nested spaces: a short proof

- Matricial notation: $\forall j$, let $\Phi^j(x) = (\phi^j_0(x)\phi^j_1(x)\dots)$
  $\rightsquigarrow$ we have $\forall j, f(x) = \Phi^j(x)C^j$
- Remember that $C^j = R\overline{C}^j$. Let us note $C^j = R'C^{j-1}$
  $\rightsquigarrow$ $R'$ is called a subdivision matrix
- We thus have $\Phi^{j-1}(x) = R'\Phi^j(x)$
- This refinement relation means each coarse scale factor $\phi^{j-1}_i$ can be rewritten as a linear combination of the fine scale factors $\phi^j_i$

# Subdivision and multiresolution synthesis

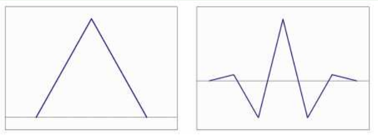
We can apply multiresolution theory to curves generated by a subdivision process:

- Let $V^j$ be the vector space generated by the $\phi_i^j$
- These are nested spaces: $V^0 \subset V^1 \subset \dots$
- Let $W^j$ be a (not necessarily orthogonal) supplement of $V^j$ in $V^{j+1}$
- Let $(\psi_i^j)_i$ be a basis of $W^j$; $\psi_i^j$ are called wavelets
- If subdivision is uniform and stationary, we can prove that $\phi_i^j(x) = \phi(2^j x - i)$ and $\psi_i^j(x) = \psi(2^j x - i)$

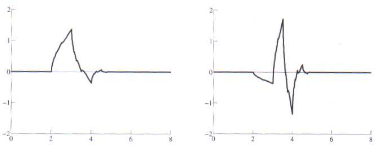
# Basis functions $\phi$ and $\psi$

$\phi$ and $\psi$ exist for each subdivision scheme, even if we don't know them beforehand.

## Example (Linear B-spline)



## Example (Daubechies)



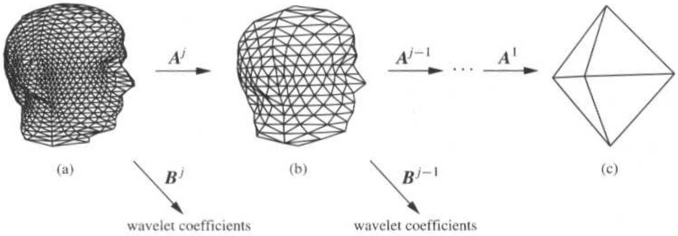
# Subdivision surfaces

- We are interested in two kinds of surface:
  1. Spline surfaces (tensor product of spline curves);
  2. Polyhedral meshes (faces are flat).
- We would like to construct hierarchical representations of both types of surfaces.
- Possible applications: compression, progressive transmission across a network, multiresolution editing, shape matching, . . .
- We restrict here the study to polyhedral meshes with triangular or quadrangular faces

# Multiresolution analysis for surfaces

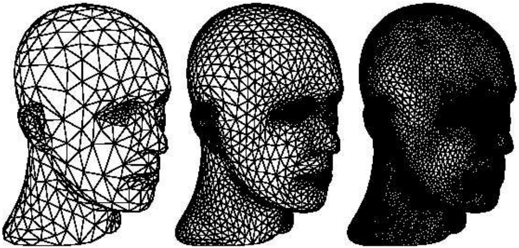
Idea remains the same:

- Decompose a high-resolution surface into a low-resolution part and a detail part, and iterate
- Geometry (e.g. vertex positions) for the coarse version computed as average of geometry for the fine version
- Coarse surface computation = multiplication by a matrix $A^j$, detail computation = multiplication by a matrix $B^j$



# Multiresolution synthesis

- As for curves, made by successive subdivisions
- Subdivision curve: iteratively refine a control polygon
- Subdivision surface: iteratively refine a control polyhedron/mesh $M^0$
  $\rightsquigarrow$ sequence of increasingly faceted meshes $M^1, M^2, \ldots,$ converging to a surface $M^\infty$



# Tricky points

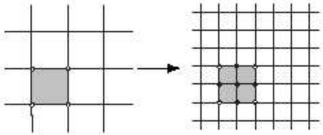
Remain the same:

- Low-resolution versions must be good approximations of high-resolution versions
- Analysis and synthesis must be done in linear time wrt to the number of surface's vertices
- The magnitude of a wavelet coefficient should provide some measure of the error introduced when this coefficient is set to zero

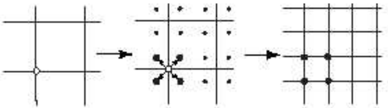
# Classification of subdivision schemes

- Two types of subdivision schemes: approximating ones and interpolating ones
- One refinement step = splitting and averaging
- Two types of splitting steps: split faces (primal schemes) or split vertices (dual schemes)
- If face split, two main types of faces: triangular and quadrangular ones

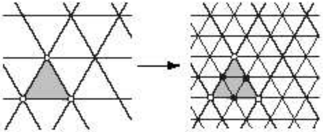
# How to split a face or a vertex ?



Face split for quads



Vertex split for quads



Face split for triangles

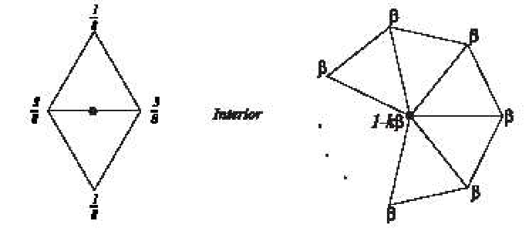
Masks are represented using a picture:

- new control point = black dot
- coefficient associated with each neighboring vertex = number next to the vertex

# Example of a subdivision scheme (1)

## Example (Loop, 1987)

- Face-split scheme for triangular meshes
- Approximating scheme
- Mask:

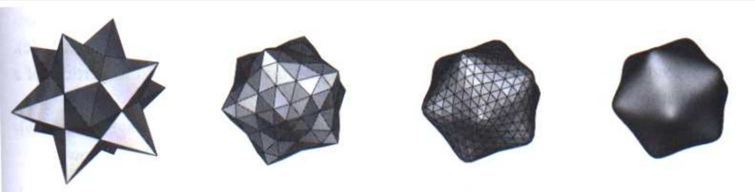



(left: mask for inserted vertices; right: mask for new position of existing vertices, $k$ = number of neighbors)

# Example of a subdivision scheme (1)

## Example (Loop, 1987)

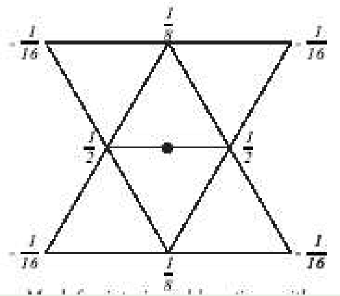
- Limit surface has been proved to be $C^2$-continuous everywhere, except at some extraordinary vertices, where it is $C^1$-continuous



# Example of a subdivision scheme (2)

## Example (Butterfly, 1990)

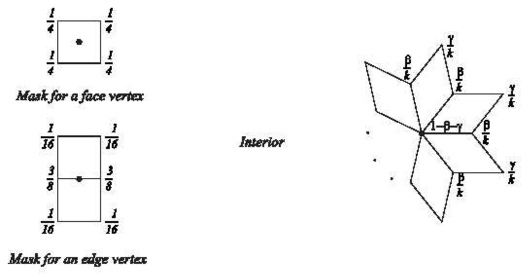
- Face-split scheme for triangular meshes
- Interpolating scheme
- Proposed by Dyn/Levin/Gregory, limit surface $C^1$-continuous except at some extraordinary vertices
- Mask:



# Example of a subdivision scheme (3)

## Example (Catmull-Clark, 1978)

- Face-split scheme for quadrangular meshes
- Approximating scheme
- Mask:

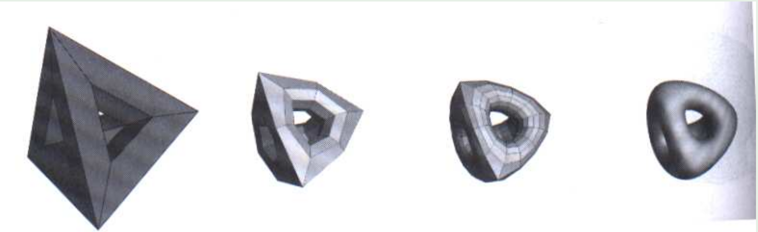



$$\beta = \frac{3}{2k}, \gamma = \frac{1}{4k}$$

# Example of a subdivision scheme (3)

## Example (Catmull-Clark, 1978)

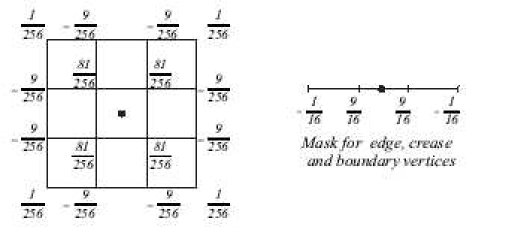
- Generalizes tensor-product cubic B-splines
- Limit surface $C^2$-continuous except at some extraordinary vertices, where it is $C^1$-continuous



# Example of a subdivision scheme (4)

## Example (Kobbelt, 1996)

- Face-split scheme for quadrangular meshes
- Interpolating scheme
- Limit surface $C^1$-continuous
- Mask:



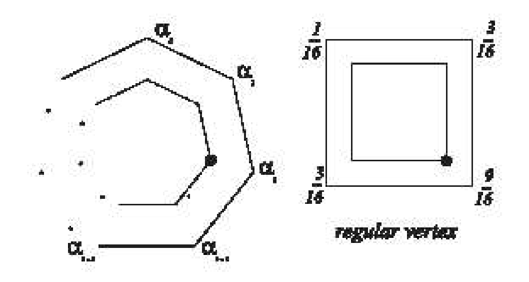

*Mask for a face vertex*

*Mask for edge, crease and boundary vertices*

# Example of a subdivision scheme (5)

## Example (Doo-Sabin, 1978)

- Vertex-split scheme for quadrangular meshes
- Approximating scheme
- Mask:

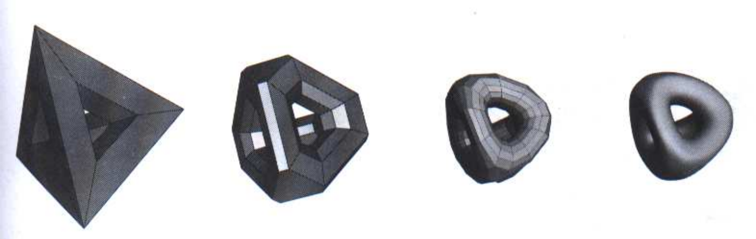



$$\alpha_0 = \frac{1}{4} + \frac{5}{4k}, \alpha_i = \frac{3+2\cos(2i\pi/k)}{4k}$$

# Example of a subdivision scheme (5)

## Example (Doo-Sabin, 1978)

- Generalizes tensor-product quadratic B-splines (Chaikin)
- Limit surface $C^1$-continuous



# Back to the classification of subdivision schemes

Face split:

| | Triang. | Quad. |
|---|---|---|
| Approx. | Loop ($C^2$) | Catmull-Clark ($C^2$) |
| Interp. | Butterfly ($C^1$) | Kobbelt ($C^1$) |

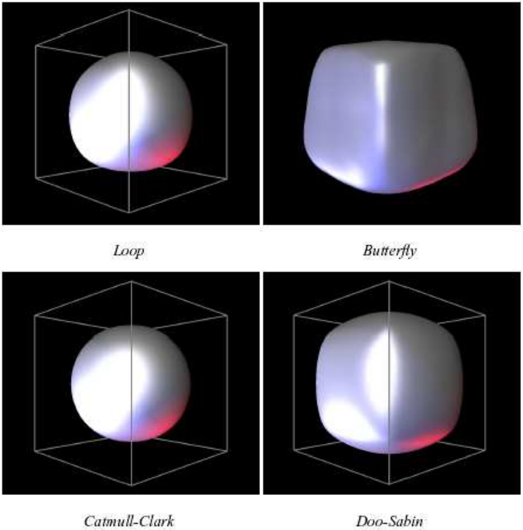
Vertex split:

| | Triang. | Quad. |
|---|---|---|
| Approx. | | Doo-Sabin ($C^1$) |
| Interp. | | |

Note that the Doo-Sabin scheme can be generalized to produce $C^n$-continous splines

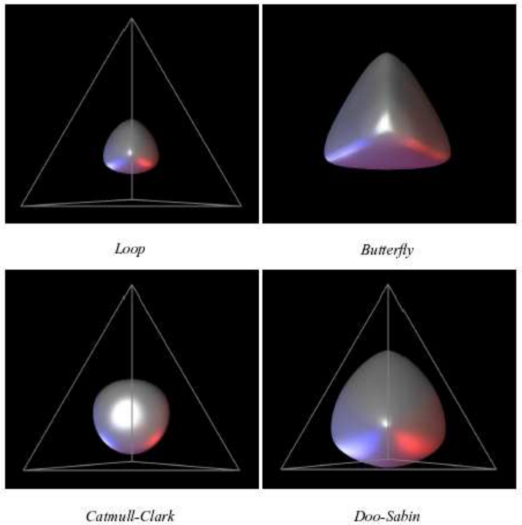
# Comparison of several subdivision schemes (1)

Starting from a cube:



# Comparison of several subdivision schemes (2)

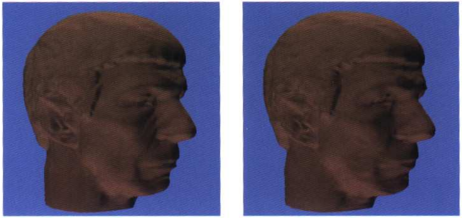
Starting from a tetrahedron:



# Applications of surface subdivision

Possible applications:

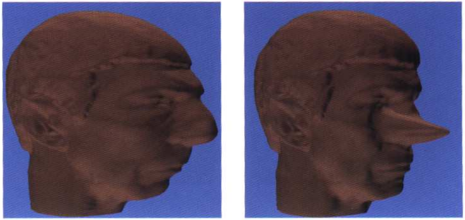
- Surface compression



# Applications of surface subdivision

Possible applications:

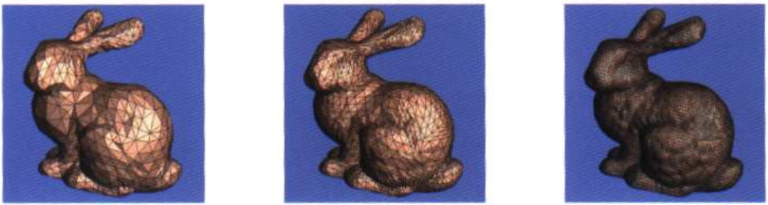
- Multiresolution editing



# Applications of surface subdivision

Possible applications:

- Progressive transmission



# Book

Main stuff in this lecture has been inspired by the following book:

E. Stollnitz, T. DeRose, D. Salesin
Wavelets for Computer Graphics
Morgan Kaufmann, 1996

# See you next week

The end !