Subdivision Surfaces
Generate smooth surfaces from a given polygonal mesh (polyhedron) with guaranteed continuity
• can handle meshes of arbitrary topology
• implementation and application is straightforward and intuitive
• analysis of continuity is mathematically involved

Originally extensions of B-spline surfaces
• Doo-Sabin scheme produces quadratic B-spline surfaces
• Catmull-Clark scheme produces cubic B-spline surfaces
• Loop subdivision generalizes quartic box-spline

Subdivision Concepts
Start with initial, discrete representation
• control points, line segments (for curves), polygons (for surfaces)

Repeated application of subdivision rules to make smoother surface
• topological splitting/refinement: how to add vertices
• smoothing/averaging: where to place vertices
• special treatment of extraordinary vertices and surface boundaries

Limit surface (or curve)
• the “mathematical” result after infinite refinements
Smoothness

A set of scalars \( m_i, 1 \leq i \leq n \), applied to a set of \( n \) vertices \( v_i \), to generate a new vertex \( w \):

\[
\mathbf{w} = \frac{\sum_{i=1}^{n} m_i \mathbf{v}_i}{\sum_{i=1}^{n} m_i}
\]

Interior and Boundary Vertices

**Interior vertices:**
- for a closed polyhedron all vertices are interior vertices
- an epsilon neighborhood is homeomorphic to a closed disk

**Boundary vertices:**
- vertices that make up the “skirt” of a polyhedron
- an edge linking 2 boundary vertices is always shared by one face of the polyhedron

Ordinary and Extraordinary Vertices

**Valence/degree** of a vertex: number of edges incident to vertex
- most schemes have an “ideal” valence for which the limit surface converges to a spline surface, except at extraordinary vertices

**Extraordinary vertices** have different valence than ordinary vertices
- subdividing a mesh does not add nor remove extraordinary vertices
- make up rules for extraordinary vertices to keep the surface “smooth”, though at lower degree of continuity

Subdivision Curves

Start with a piecewise linear curve

Chaikin’s algorithm (1974):
- **refinement**: insert new edge vertex midpoint on each edge
- **smoothing**: average each vertex with the clockwise neighbor
- repeat
Chaikin’s Curve Subdivision

Let initial vertices on control polygon be \( \mathbf{v}_i \). Vertices at refinement level \( j \) are \( \mathbf{v}_{ij} \), thus \( \mathbf{v}_i = \mathbf{v}_{i0}^0 \).

1. **Refinement**: insert new midpoint \( (\mathbf{m}_{ij}^{*}) \) vertex on each edge
   \[
   \mathbf{v}_{2ij}^{*+1} = \frac{1}{3} \mathbf{v}_i + \frac{1}{3} \mathbf{m}_{ij}^{*+1} + \frac{1}{3} \mathbf{v}_{i+1}
   \]
   superscript, not power

2. **Smoothing**: average each vertex with the clockwise neighbor to create new vertex point \( (\mathbf{v}_{ij}^{*+1}) \):
   \[
   \mathbf{v}_{2ij}^{*+1} = \frac{1}{3} \mathbf{v}_i + \frac{1}{3} \mathbf{v}_{i+1} + \frac{1}{3} \mathbf{m}_{ij}^{*+1}
   \]

3. **Averaging Mask**: (3 1) (also written as \((0 \ 1/2 \ 1/2)\) for \( \mathbf{m}_{ij}^{*+1} + \frac{1}{2} \mathbf{v}_i + \frac{1}{2} \mathbf{v}_{i+1} \))

4. **Continue on next slide** . . .

5. **Connect all new vertex points to form refined curve**

Chaikin’s Curve Subdivision

Resulting curve is a uniform quadratic B-spline

Doo-Sabin Subdivision

Introduces a new vertex for each face at the midpoint between an old vertex and face centroid

Subdivision rules create a dual of the control net: a new face replaces each face, edge, and vertex of the control net
Doo-Sabin Subdivision

A generalization of quadratic curve subdivision (Chaikin’s algorithm) to surfaces with arbitrary topology

For regular quad meshes, resulting surface is a biquadratic B-spline surface

Cubic B-spline Curve Subdivision

1. For each edge compute a new edge point using the averaging mask $$(1 \ 1)$$: $$e_{2i}^{j+1} = \frac{1}{2} v_i^j + \frac{1}{2} v_{i+1}^j$$
2. Compute new vertex points using the mask $$(1 \ 6 \ 1)$$:
   $$v_{2i+1}^{j+1} = \frac{1}{8} v_i^j + \frac{6}{8} v_{i+1}^j + \frac{1}{8} v_{i+2}^j$$
3. Connect the new edge- and vertex points; resulting curve is a uniform cubic B-spline

Vertex points are midpoint between midpoints between old vertices and new edge points:

$$v_{2i+1}^{j+1} = \frac{1}{8} e_{2i}^{j+1} + \frac{3}{8} v_i^j + \frac{1}{8} v_i^{j+1}$$
$$+ \frac{1}{8} \left( \frac{1}{2} v_i^j + \frac{1}{2} e_{2i+1}^{j+1} \right)$$
$$= \frac{1}{8} e_{2i}^{j+1} + \frac{3}{8} v_i^j + \frac{1}{8} v_i^{j+1}$$
$$+ \frac{1}{8} \left( \frac{1}{2} v_i^j + \frac{1}{2} v_{i+1}^j + \frac{1}{2} v_{i+2}^j \right)$$
$$= \frac{1}{8} v_i^j + \frac{5}{8} v_{i+1}^j + \frac{1}{8} v_{i+2}^j$$

averaging mask $$(1 \ 6 \ 1)$$ of only old vertices, or also given as $$(14 \ 15 \ 14)$$
Subdivision → New Control Points

Let:
\[
P = \begin{bmatrix} v_0^0 & v_1^0 & e_1^1 & v_1^1 & e_2^1 & v_2^1 & e_3^1 & v_3^1 & e_4^1 \end{bmatrix}, \quad P' = \begin{bmatrix} v_0^0 & e_0^1 & v_1^1 & e_1^1 & v_2^1 & e_2^1 & v_3^1 & e_3^1 \end{bmatrix}
\]

\[
H_s = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{4} & \frac{3}{4} & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{4} & \frac{3}{4} & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}
\]

splitting matrix

New control points: \( P' = H_s P \)

Uniform B-spline Patch Subdivision

\[
s(u,v) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} B_s P B_s^T \begin{bmatrix} 1 & v & v^2 & v^3 \end{bmatrix}^T
\]

orthographic top-down view

3D perspective view

Subdivision Reparameterized

\[
s_{11}(u,v) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} B_{s1} P_{11} B_{s1}^T \begin{bmatrix} 1 & v & v^2 & v^3 \end{bmatrix}^T
\]

new control points!

Control mesh approaches surface

Limit of Subdivision

Control mesh approaches surface

Limit of subdivision is the surface
New Control Points

\[
P_{11} = H_{s1}P_{H_{s1}}^T
\]

\[
P_{12} = H_{s1}P_{H_{s2}}^T
\]

\[
P_{21} = H_{s2}P_{H_{s1}}^T
\]

\[
P_{22} = H_{s2}P_{H_{s2}}^T
\]

in this parametric view these knot points are collocated; the 3D control points are not

\[
H_{s1} = \begin{bmatrix}
\frac{1}{2} & \frac{1}{2} & 0 & 0 \\
\frac{1}{2} & \frac{1}{2} & 0 & 0 \\
0 & \frac{1}{2} & \frac{1}{2} & 0 \\
0 & \frac{1}{2} & \frac{1}{2} & 0
\end{bmatrix}
\]

\[
H_{s2} = \begin{bmatrix}
\frac{1}{2} & \frac{1}{2} & 0 & 0 \\
\frac{1}{2} & \frac{1}{2} & 0 & 0 \\
0 & \frac{1}{2} & \frac{1}{2} & 0 \\
0 & \frac{1}{2} & \frac{1}{2} & 0
\end{bmatrix}
\]

New Control Points

Instead, compute the new control points iteratively:

Face points

Edge points

Moved vertex points

Subdivision → New Control Points

Subdivision as a matrix \( H_S \) of weights \( w \):

- \( H_S \) is very sparse
- limit surface: \( P^n = \lim_{j \to \infty} (H_S)^j P^0 \) allows for analysis: curvature, limit surface
- not for implementation!

\[
P^{j+1} = H_S P^j
\]

\[
\begin{bmatrix}
p_0^{j+1} \\
p_1^{j+1} \\
p_2^{j+1} \\
p_a^{j+1}
\end{bmatrix} =
\begin{bmatrix}
w_{00} & w_{01} & \cdots & 0 \\
w_{10} & w_{11} & \cdots & 0 \\
\vdots & \vdots & \ddots & 0 \\
0 & 0 & \cdots & w_{nk}
\end{bmatrix}
\begin{bmatrix}
p_0^j \\
p_1^j \\
p_2^j \\
p_a^j
\end{bmatrix}
\]

\( H_S \) 25x16 subdivision matrix

\( P^j \): vector of coarse control points, length 16

\( P^{j+1} \): vector of coarse control points, length 25

New Control Points

Face point:

\[
f^{j+1} = \frac{m^{j+1} + m_{j+1}}{2} = \frac{v_1 + v_4 + v_2 + v_3}{4}
\]

Edge point:

\[
e^{j+1} = \frac{1}{2} \left( f_1^{j+1} + m_{j+1} \right) + \frac{1}{2} \left( m_{j+1} + f_2^{j+1} \right)
\]

\[
= \frac{1}{2} \left( f_1^{j+1} + f_2^{j+1} \right) + m_{j+1}
\]

\[
= \frac{v_1 + v_2 + f_1^{j+1} + f_2^{j+1}}{4}
\]

Note \( f^0, m^0, e^0 \) not defined/used
**Vertex Points**

\[
w_0 = \frac{1}{2} \left( \frac{m_{i+1} + v_i + m_{i+2}}{2} \right) = \frac{1}{2} m_i + \frac{1}{2} v_i + \frac{1}{4} m_4
\]

\[
v'_0 = \frac{1}{2} \left( (e_1 + w_0) + \frac{1}{2} (w_0 + e_2) \right) = \frac{1}{4} \left( \frac{m_{i-1} + v_i + m_{i+1}}{2} \right) + \frac{1}{2} w_i = \frac{1}{4} \left( \frac{m_{i-1} + v_i + m_{i+1}}{2} \right) + \frac{1}{2} \left( 2 v_i + m_0 + m_4 \right)
\]

\[
v^j = \frac{1}{n} \sum_{i=1}^{n} \left( f_i^j + e_i^j + f_i^{j+1} + e_i^{j+1} \right) + \frac{1}{2} \left( m_{j+1} + m_j + m_{j-1} + m_{j-2} \right)
\]

\[
v^j = \frac{2}{n} \left( f_i^j + e_i^j + f_i^{j+1} + e_i^{j+1} \right) + \frac{1}{n} \left( v_i^j + v_i^{j+1} + v_i^{j+2} + v_i^{j+3} \right)
\]

**Catmull-Clark Subdivision (1978)**

Subdivision level \( j+1 \):

- **face point**: for each face, add a new vertex at its centroid that is the average of the surrounding \( m \) vertices:
  \[
f_i^{j+1} = \frac{1}{m} \sum_{i=1}^{m} v_i
\]

- **edge point**: for each edge, add a new edge point which is the average of the two vertices and the two face points adjacent to the edge:
  \[
e_i^{j+1} = \frac{v_i^j + v_i^{j+1} + f_i^{j+1} + f_i^{j+2}}{4}
\]

- **moved vertex point**: vertex moved to the weighted average between the original position, the \( n \) midpoints (not edge) points and the \( n \) face points surrounding the vertex (\( n \): vertex valence, \( =4 \)):
  \[
v_i^{j+1} = \frac{n-2}{n} v_i^j + \frac{1}{n} \sum_{i=1}^{n} v_i^j + \frac{1}{n} \sum_{i=1}^{n} f_i^{j+1}
\]

**Catmull-Clark Subdivision Rules**

- **face point**: for each face, add a new vertex at its centroid that is the average of the surrounding \( m \) vertices:
  \[
f_i^{j+1} = \frac{1}{m} \sum_{i=1}^{m} v_i
\]

- **edge point**: for each edge, add a new edge point which is the average of the two vertices and the two face points adjacent to the edge:
  \[
e_i^{j+1} = \frac{v_i^j + v_i^{j+1} + f_i^{j+1} + f_i^{j+2}}{4}
\]

- **moved vertex point**: vertex moved to the weighted average between the original position, the \( n \) midpoints (not edge) points and the \( n \) face points surrounding the vertex (\( n \): vertex valence, \( =4 \)):
  \[
v_i^{j+1} = \frac{n-2}{n} v_i^j + \frac{1}{n} \sum_{i=1}^{n} v_i^j + \frac{1}{n} \sum_{i=1}^{n} f_i^{j+1}
\]

**Catmull-Clark Subdivision**

Works with arbitrary polygonal mesh:

- **face point**: for each face, add a new vertex at its centroid that is the average of the surrounding \( m \) vertices:
  \[
f_i^{j+1} = \frac{1}{m} \sum_{i=1}^{m} v_i
\]

- **edge point**: for each edge, add a new edge point which is the average of the two vertices and the two face points adjacent to the edge:
  \[
e_i^{j+1} = \frac{v_i^j + v_i^{j+1} + f_i^{j+1} + f_i^{j+2}}{4}
\]

- **moved vertex point**: vertex moved to the weighted average between the original position, the \( n \) midpoints (not edge) points and the \( n \) face points surrounding the vertex (\( n \): vertex valence, \( =4 \)):
  \[
v_i^{j+1} = \frac{n-2}{n} v_i^j + \frac{1}{n} \sum_{i=1}^{n} v_i^j + \frac{1}{n} \sum_{i=1}^{n} f_i^{j+1}
\]
Catmull-Clark Subdivision

Subdivision rules are **chosen** to improve continuity

Smoothness of limit surface:
- $C^2$ almost everywhere
- $C^1$ at extraordinary vertices
- strictly generalize uniform tensor-product bicubic B-splines:
  - works with existing tools for tensor-product B-splines
- generalization of cubic B-splines subdivision to irregular patch:

![Diagram of non-quad face](image1)

extraordinary vertex

---

Any mesh can be subdivided
- cut holes, create unusual topology, stitch pieces together
- no matter how complicated the mesh, it will lead to a smooth surface!

Extensions: localized subdivision rules
- **creases**: NURBS requires use of trim curves; for subdivision, just modify the subdivision mask
- **edge preservation**: hard edges
- **adaptive subdivision**

---

Edge Preservation

To get sharpness and creases, define new subdivision rules for "creased" edges and vertices
- **crease**: a smooth curve with continuity $G^0$ on the surface (2 sharp edges)
- **corner**: a vertex where $\geq 3$ sharp edges meet
- **dart**: a vertex where a crease ends and smoothly blends into the surface (1 sharp edge)

![Diagram of edge preservation](image2)

corner
crease
dart

compared to:

Hoppe94
LoopRT
Funkhouser, Zhang
Sharp Edges

Idea: edges with a sharpness \( j \) are subdivided using sharp rules for the first \( j \) iterations, and then smoothly, as usual, to the limit surface
- tag edges as sharp or not sharp:
  - newly created edges are assigned a sharpness of \( j-1 \)
  - edges with \( j = 0 \) are not sharp
  - edges with \( j > 0 \) are sharp

During subdivision, if an edge is not-sharp use normal smooth subdivision rules; if an edge is sharp, use sharp subdivision rules

Approximating subdivision algorithm can be made interpolating

Sharp Rules

Subdivision level \( j+1 \):
- face point unchanged
- edge point
  \[
  e^{j+1} = \frac{v^j + v^j}{2}
  \]
- moved vertex
  - corner: \( >2 \) \( v^{j+1} = v^j \)
  - crease: \( \frac{2}{8} \) \( v^{j+1} = \frac{v^j + 6v^j + v^j}{8} \)
  - dart: 1 unchanged

Subdivision Surfaces

Scheme classification by:
- interpolating or approximating
- mesh type: quads, triangles, hex, ..., combination
- subdivision by face split (primal) or vertex split (dual)
- B-spline order of limit surface
- smoothness

Algorithms:
- Doo-Sabin '78 approximate \( C^1 \) quad dual
- Catmull-Clark '78 approximate \( C^2 \) quad primal
- Loop '87 approximate \( C^2 \) triad primal
- DLG midpoint '87 approximate \( C^2 \) triad dual
- Butterfly (mod) '90, '96 interpolate \( C^1 \) quad dual
- Kobelt '96 interpolate \( C^1 \) quad primal
- \( \sqrt{3} \) '00 approximate \( C^0 \) triad dual

Loop Subdivision

Named after Charles Loop

Start with a triangular mesh

Resulting surface is a generalization of three-direction quartic box-spline

Subdivision rules:
- refinement: break edges at midpoint, for both faces
- smoothing: different averaging masks for new ("odd") and old ("even") vertices
Loop Subdivision Masks

New (“odd”) vertices are placed based on weighted average of old vertices on both faces

Old (“even”) vertices are moved based on surrounding neighbors

Odd mask: Even mask:

Loop Subdivision

How to choose $\beta$?
- must ensure tangent plane or normal continuity ($G^1$) of limit surface
- involves calculating eigenvalues of matrices

Original Loop:

$$\beta = \frac{1}{8n} \left[ 40 - \left( 3 + 2 \cos \left( \frac{2 \pi}{n} \right) \right)^2 \right]$$

Warren:

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

Loop Subdivision

Approximating subdivision
- does not interpolate the control mesh
- within convex hull
- in the limit a smooth surface
- $C^2$ almost everywhere
- $C^1$ at extraordinary vertices (valence $\neq 6$)
\(\sqrt{3}\) Subdivision Scheme

Starts with a triangle mesh

Number of faces triples per iteration
• slower growth rate

Gives finer control over polygon count
• better for adaptive subdivision

Adaptive Subdivision

Not all regions of a model must be subdivided to the same resolution
• may be due to limited triangle budget

Stop subdivision at different levels across the surface, depending on:
• local surface curvature
• projected screen size of triangles
• view dependence
• distance from viewer
• silhouettes
• in view frustum
• careful to avoid “cracks”!

Balanced Subdivision

Crack avoidance: replace incompatible coarse triangles with triangle fan

Balanced subdivision: neighboring subdivision levels must not differ by more than one

Subdivision Surfaces

Characteristics and advantages:
• one surface, not a patchwork (collection of patches)
• no seams, can deform/animate geometry without cracks
• guaranteed continuity (smooth at boundaries)
• arbitrary control mesh, not limited to quads
• can make surfaces with arbitrary topology or connectivity
• simple, only need subdivision rule
• adaptive subdivision: areas of surface with higher curvature can be more finely subdivided
• multiresolution: LoD, scalable
• local support: only look at nearby vertices
• numerical stability, well-behaved meshes
• affine invariance
• efficient rendering
Subdivision Surfaces

Disadvantages:
• non-intuitive specification: it’s a procedural definition
• non-parametric, not implicit: hard to parameterize
  • no global \((u, v)\) parameters
• hard to compute intersections
• tricky at special vertices (those with more or less than 6 neighbors in a triangular mesh)

Parametric vs. Subdivision Surfaces

Parametric B-splines
• smooth
• must be tessellated
  • sampling issues
  • triangle size issue
  • cracking concern
• have uniform resolution
  • detail must be global
• require regular grid
• complex topology hard
  • no corners, holes
  • trimming hard
  • stitching hard
  • creases and sharp edges hard
• \((u, v)\) parameterization
  • but not controllable

Subdivision
• limit surfaces are smooth
• gives meshes
• subdivide as needed
• always connected
• get as many poly as you need
• put details where needed
  • detail is multiresolution
• works with arbitrary mesh
• any topology can be handled
  • easy to make corners, holes
  • trimming easy
  • stitching easy
  • creases and sharp edges easy
• \((u, v)\) parameterization
  • by subdivision of points
  • controllable

Funkhouser, Durand

Gleicher