Chapter 8

Subdivision for Graphics and Visualization
Introduction

- Tensor product B-spline surfaces restrict the control mesh to a rectangular topology → limit to the complexity of shapes
- **Subdivision surfaces** are a more general solution because they can handle arbitrary topology and yet produce regular B-spline surfaces in the normal way.
  - Were initiated by two papers in 1978

- Subdivision surfaces:
  - Provide simple techniques to generate a smooth surface from a given polygonal mesh (polyhedron)
  - Have the ability to handle meshes of arbitrary topology
  - Complex shapes can be obtained / retained / edited at various levels of the refinement process.
Notation

• Mask:
  - a set of scalars \( (m_i)_{1 \leq i \leq n} \)
  - can be 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 vertices:
  - For a closed polyhedron all vertices are interior vertices
  - Correspond to a point on the limit surface with an epsilon neighborhood homeomorphic to a closed disk

• Boundary vertices:
  - For an open polyhedron, a set of vertices fall on the boundary
  - Are the vertices that make up the skirt of the polyhedron
  - An edge linking 2 boundary vertices is always shared by one face of the polyhedron
Notation (2)

• **Valence of a vertex:**
  - The number of edges incident on it
  - An interior vertex is at least 3-valent
  - A boundary vertex could be 2-valent

• **Ordinary vertex / face:**
  - Depends on the subdivision scheme
  - For surfaces based on tensor products, ordinary vertex is:
    - 4-valent if it is an interior vertex
    - 3-valent if it is a boundary vertex
  - Ordinary face is a face with 4 ordinary vertices
  - For triangular meshes ordinary vertex is usually 6-valent

• **Extraordinary vertex:** is a vertex that is not ordinary

• **Extraordinary face:** is a face with \( n \ (n \neq 4) \) vertices
Notation (3)

- **1-ring:**
  - A 1-ring for an interior vertex $v_i$ is the set if vertices $(v_j)$, where $v_iv_j$ is an edge incident to $v_i$

- **Regular setting:**
  - When all vertices are ordinary vertices

- **Irregular setting:**
  - If the configuration contains at least one extraordinary vertex or one extraordinary face

- **Tensor product:**
  - Given two masks $(m_i)_{1 \leq i \leq r}$ and $(n_j)_{1 \leq j \leq p}$, the tensor product of these two masks is another mask of $r \times p$ elements $(m_i \times n_j)_{1 \leq i \leq r, 1 \leq j \leq p}
Subdivision Curves

- Popular subdivision curves:
  - Quadratic subdivision (Chaikin)
  - Cubic subdivision
  - Four-point subdivision

- Assume that:
  - The initial control-polygon vertices are \( \mathbf{v}_i \)
  - The vertices of its refined polygon are \( \mathbf{v}_i^j \)
  - The level of refinement is \( j \)
  - So \( \mathbf{v}_i = \mathbf{v}_i^0 \)
**Quadradic Curve Subdivision**

**Steps:**

1. For each edge $e_i^j$ connecting two vertices $v_i^{j-1}$ and $v_i^j$, compute two new vertices using the masks (1,3), and (3,1) as follows:

   $v_{2i-1}^{j+1} = \frac{3}{4}v_i^j + \frac{1}{4}v_i^{j-1}$

   $v_{2i}^{j+1} = \frac{3}{4}v_i^j + \frac{1}{4}v_i^{j+1}$

2. Construct a new polygon as follows:
   
   (a) For each vertex $v_i^j$, connect its two new vertices $v_{2i-2}^{j+1}$ and $v_{2i-1}^{j+1}$ forming a **V-edge** (corresponding to a vertex) of the new control polygon.

   (b) For each edge $e_i^j$, connect its two new vertices $v_{2i}^{j+1}$, $v_{2i-1}^{j+1}$ forming an **E-edge** (corresponding to an edge) of the new control polygon.
Quadratic Curve Subdivision (2)

- Repeat the algorithm’s steps until the refined control polygon converges to a smooth limit curve.
- The resulting curve is simply a uniform quadratic B-spline if a knot is inserted at the middle of every interval of the initial control polygon.
- One example of Chaikin’s subdivision:
  - Black disks: the original vertices
  - Hollow disks: the refined vertices
Cubic Curve Subdivision

Steps:

1. For each vertex \( \mathbf{v}_i^j \), compute a new vertex \( \mathbf{v}_{2i}^{j+1} \), called V-vertex, using the mask \((1,6,1)\) as follows:

\[
\mathbf{v}_{2i}^{j+1} = \frac{\mathbf{v}_{i-1}^j + 6\mathbf{v}_i^j + \mathbf{v}_{i+1}^j}{8}
\]

2. For each edge \( \mathbf{v}_{i-1}^j \mathbf{v}_i^j \), compute a new vertex, called E-vertex, using the mask \((1,1)\) as follows:

\[
\mathbf{v}_{2i-1}^{j+1} = \frac{\mathbf{v}_{i-1}^j + \mathbf{v}_i^j}{2}
\]

3. Construct a new refined polygon by connecting the E- and V-vertices generated as above.
Cubic Curve Subdivision (2)

- The refined control polygon converges to the uniform cubic B-spline defined by the original control polygon.

- One refinement step:
  - Black disks: original vertices
  - Hollow disks: refined vertices
Four-Point Subdivision

Steps:
1. For each vertex $v_i$, denote the new corresponding vertex $v_{2i}^{j+1}$, called V-vertex:
   $$v_{2i}^{j+1} = v_i^j$$

2. For each edge $v_{i-1}^j v_i^j$, compute a new E-vertex using the mask $(-1,9,9,-1)$ as follows:
   $$v_{2i-1}^{j+1} = \frac{-v_{i-2}^j + 9v_{i-1}^j + 9v_i^j - v_{i+1}^j}{16}$$

3. Construct a refined control polygon by connecting each V-vertex to its neighboring E-vertices.
Four-Point Subdivision (2)

- It is an interpolating scheme: the limit curve interpolates the vertices of the original control polygon.
- One refinement step:
  - Black disks: original vertices
  - Hollow disks: refined vertices
Subdivision Surfaces

- The extension of the quadratic and cubic curve subdivision algorithms to tensor product surfaces is straightforward
  - These surfaces have rectangular topology
- Challenge: how to extend such surface definitions to control meshes with arbitrary topology
- Subdivision surfaces:
  - Quadratic Tensor Product Subdivision
  - Cubic Tensor Product Subdivision
  - Arbitrary topology subdivision surfaces
Quadratic Tensor Product Subdivision

• Assume a rectangular mesh:
  - The surface is parameterized by 2 parameters $s$ & $t$
  - Along s-direction, generate 2 vertices on each edge using the mask (1,3) & (3,1)
  - Along t-direction, generate 2 vertices on each edge joining 2 square vertices using the same masks
• The new vertices can be computed by the tensor product of the (3,1) mask
  - $A'$ is obtained by $(3,1) \times (1,3)$ applied to $A, B, C, D$ in that order
  - The other vertices are obtained by a rotation of this mask.
Quadratic Tensor Product Subdivision (2)

Observations:

1. For each face $f$ of the initial mesh, a new face (called F-face) is generated from its refined vertices.

2. For each edge $e$ of the initial mesh, a new face (called E-face) is generated from the refined vertices of that edge on the faces common to it.

3. For each vertex $v$ of the initial mesh, a new face (called V-face) is generated from the refined vertices of that vertex on the faces sharing it.

4. The refined polyhedron is actually obtained by connecting the refined vertices to form all of these faces.
The cubic subdivision rules are obtained by the tensor product of the masks used in cubic curve subdivision:

- Apply these masks in the s-direction
  - To generate a V-vertex corresponding to each old vertex
  - To generate a E-vertex corresponding to each edge vertex
- Apply the same masks in the t-direction to get the final refined vertices
- The new vertices can be computed by the tensor product of the masks used in cubic curve subdivision.
Cubic Tensor Product Subdivision (2)

Observations:

1. For each face $f$ of the initial mesh, a new vertex (called F-vertex) is generated as the centroid of that face.
   - using the mask $(1,1) \times (1,1)$

2. For each edge $e$ of the initial mesh, a new vertex (called E-vertex) is generated from the vertices of that edge and the two F-vertices of its shared faces.
   - using the mask $(1,1) \times (1,6,1)$

3. For each vertex $v$ of the initial mesh, a new vertex (called V-vertex) is generated as a linear combination of that vertex, the E-vertices of the edges incident to it, and the F-vertices of the faces sharing it.
   - using the mask $(1,6,1) \times (1,6,1)$

4. The refined mesh is similarly obtained by connecting the refined vertices to form all these faces.
Subdivision Schemes

Generalization of tensor-product subdivision to arbitrary topology.

- **Subdivision Surface** is defined by a tuple \((P_0, R)\) where
  - \(P_0\): initial mesh of arbitrary topology, called a polyhedron
  - \(R\): set of rules, called a refinement procedure

- **Polyhedron**:
  - A set of vertices, edges and faces in 3D space
  - The faces don’t have to be planar

- **The refinement procedure** (scheme)
  - is applied to the polyhedron \(P_0\) to generate another polyhedron \(P_1\)
  - \(P_1\) in turn is taken as an input to the refinement procedure to generate another polyhedron \(P_2\)
  - If \(R\) satisfies some conditions then the sequence of polyhedra \(P_0, P_1, ..., P_i, ...\) will converge to a smooth surface.
Subdivision Schemes

- The Doo – Sabin scheme
- The Catmull – Clark scheme
- The Loop scheme
- The Modified Butterfly scheme
- The midpoint subdivision scheme
- The $\sqrt{3}$ subdivision scheme

Notation:
- $P^j$ is the refined polyhedron at level $j$.
- $v_i^j$, $e_i^j$, $f_i^j$ are the vertices, edges and faces at level $j$. 
The Doo-Sabin scheme

- Extends the quadratic tensor product to an arbitrary topology
- Generates biquadratic B-spline surfaces for regular meshes (where all faces are quads)
- Challenge: compute the refined vertices of an $n$-sided face $f_i^j$ with $n \neq 4$
The Doo-Sabin scheme (2)

Steps:

1. For each \( n \)-sided face \( f_i^j \), compute \( n \) refined vertices \( v_i^{j+1} \) as a linear combination of the vertices of that face:

\[
v_i^{j+1} = \sum_{k=1}^{n} \alpha_{ik} v_k^j
\]

with:

\[
\alpha_{ii} = \frac{n+5}{4}
\]

\[
\alpha_{ik} = \frac{3 + 2\cos(2\pi(i-k))/n}{4n} \quad \text{for } k \neq i
\]

2. Construct a refined polygon \( P_i^{j+1} \) as follows:

(a) For each \( n \)-sided face \( f_i^j \), generate a new face (called F-face) from its refined vertices.

(b) For each edge \( e_i^j \), generate a new face (called an E-face) from the refined vertices of that edge on the faces common to it.

(c) For each vertex \( v_i^j \), generate a new face (called V-face) from the refined vertices of that vertex on the faces sharing it.
The Doo-Sabin scheme (3)

Observations:
1. An $n$-sided face always generates an F-face with the same number of sides.
   - Extraordinary face if $n \neq 4$
   - Ordinary face if it is 4-sided
2. An $n$-valent vertex generates an $n$-sided face
   - All 4-valent vertices generate ordinary faces
3. After the first refinement, all vertices are 4-valent
4. All E-faces are ordinary faces
5. After a few steps of refinement, almost all the faces are 4-sided except of $ef + ev$ extraordinary faces, where
   1. $ef = \text{number of extraordinary faces of the initial mesh}$
   2. $ev = \text{number of extraordinary vertices of the initial mesh}$
The Doo-Sabin scheme (4)

Observations (cont):

6. -Every vertex shared by 4 quads corresponds to a quadratic B-spline patch.
   -Doo – Sabin surfaces are biquadratic B-spline surfaces, except at small areas that correspond to extraordinary faces & vertices.

7. -The centroid of every face is a point on the limit surface.
   -A planar face is tangent to the limit surface at that point.
The Catmull-Clark scheme

- Is an extension of the tensor cubic subdivision.
- The E-vertices and F-vertices are computed using the same formulas adopted in the rectangular topology case.
- Challenge: how to compute the V-vertices of the $n$-valent ($n \neq 4$) vertices.
The Catmull-Clark scheme (2)

Steps:

1. For each face $f_i^j$ of the input polyhedron, generate a new vertex $v_{f_i^{j+1}}$ (called F-vertex) as the centroid of that face.

2. For each edge $e_i^j = v_i^j v_{i+1}^j$ of the input polyhedron, generate a new vertex $v_{e_i^{j+1}}$ (called E-vertex) as a linear combination of $v_i^j$, $v_{i+1}^j$, and their four adjacent vertices on the two faces shared by that edge.
The Catmull-Clark scheme (3)

Steps:

3. For each vertex $v_j^i$ of the input polyhedron, generate a new vertex (called V-vertex) $v_i^{j+1}$ as a linear combination of $v_j^i$, the E-vertices of the edges sharing it, and the F-vertices of the faces sharing it.

$$v_i^{j+1} = \alpha_n \sum_{k=1}^{n} vf_k^j + \beta_n \sum_{k=1}^{n} ve_k^j + \gamma_n v_i^j$$

where $\alpha_n = \beta_n = \frac{1}{n^2}$

$$\gamma_n = \frac{n-2}{n}$$
The Catmull-Clark scheme (4)

Observations:

1. An \( n \)-valent vertex always generates a V-vertex of the same valency.
   - If \( n \neq 4 \) the vertex is extraordinary.
   - If \( n = 4 \) the vertex is ordinary.

2. After the 1\(^{st}\) refinement, every initial \( n \)-sided face generates an F-vertex with valence \( n \).
   - All faces of the refined polyhedra will become 4-sided.
   - The Catmull–Clark algorithm is described for quad meshes, as one refinement step will get rid of the \( n \)-sided faces.

3. All E-vertices are 4-valent.

4. After one step of refinement, all vertices become 4-valent except for \( ev+ef \) extraordinary vertices.
The Catmull-Clark scheme (5)

Observations (cont):

5. -Every quad surrounded by 8 other quads corresponds to a bicubic B-spline patch.
- Catmull-Clark subdivision surfaces are cubic B-spline surfaces except around a small number of extraordinary vertices.

6. -Every n-valent vertex $v_i^1$ of the polyhedron $P^l$ converges to a point on the limit surface given by:

$$v_i^\infty = \frac{n^2 v_i^1 + 4 \sum_{j=1}^{n} v e_j^1 + \sum_{j=1}^{n} v f_j^1}{n(n+5)}$$

where $v e_j^1$: the E-vertices of the edges incident to the vertex $v_i^1$
$v f_j^1$: the F-vertices of the faces sharing it
The Loop scheme

- Is devoted to triangular meshes

- The algorithm:
  - Takes as an input a polyhedron $P^j$ at level $j$
  - Generates another polyhedron $P^{j+1}$

- The smooth limit surface is an extension of the three-direction quartic box-spline
The Loop scheme (2)

Steps:

1. For each edge $e_i^j$, do the following:
   (a) Let $t_1$ and $t_2$ be the two triangles sharing that edge.
   (b) Let $c_1$ and $c_2$ be the vertices of that edge and $c_3$ and $c_4$ be the other two vertices of $t_1$ and $t_2$.
   (c) Generate an E-vertex $ve_i^{j+1}$ using the mask as follows:

$$ve_i^{j+1} = \frac{3c_1 + c_3 + 3c_2 + c_4}{8}$$
The Loop scheme (3)

Steps:
2. For each $n$-valent vertex $v_i^j$, do the following:
   (a) Let $c_1, c_2, \cdots, c_n$ be the vertices of the 1-ring around $v_i^j$.
   (b) Generate a V-vertex $v_i^{j+1}$ using a linear combination of the vertex $v_i^j$ and its 1-ring vertices as follows:

\[ v_i^{j+1} = (1 - n\alpha_n) v_i^j + \alpha_n \sum_{k=1}^{n} c_k \]

where $\alpha_3 = \frac{3}{16}$

\[ \alpha_n = \frac{1}{n} \left( \frac{5}{8} - \frac{3}{8} + \frac{1}{4} \cos \left( \frac{2\pi}{n} \right) \right) \] for $n > 3$
The Loop scheme (4)

Steps:

3. Generate a refined polyhedron $P^{j+1}$ as follows:
   
   i. For each triangle of the polyhedron $P^j$, connect the E-vertices of its 3 edges to form a triangle of $P^{j+1}$.
   
   ii. For each $n$-valent vertex of the polyhedron $P^j$, connect its V-vertex to the E-vertices of all edges incident to it.

As such, $n$ triangles are added to the refined polyhedron $P^{j+1}$. 

Observations:

1. A 6-valent vertex is ordinary, otherwise it is extraordinary.

2. The limit surface is $C^2$, except at the extraordinary points where it is $C^1$.

3. A $n$-valent vertex $v_i^0$ on the initial mesh converges to a limit point:

$$v_i^\infty = \frac{3 + 8\alpha_n (n-1)}{3 + 8n\alpha_n} v_i^0 + \frac{8\alpha_n}{3 + 8n\alpha_n} \sum_{j=1}^{n} v_j^0$$

where $v_j^0$: the 1-ring vertices around $v_i^0$. 

---

The Loop scheme (5)
The Modified Butterfly scheme

- Was initially developed as an extension of the four-point scheme
- Later was modified by Zorin to improve its smoothness
- Is a triangle – based algorithm
The Modified Butterfly scheme (2)

Steps:
1. For each vertex \( v_i^j \), let its V-vertex be the same as the original vertex, 
   \[ v_i^{j+1} = v_i^j \]
2. For each edge \( e_i^j \):
   (a) If both vertices of that edge are 6-valent, then do the following:
      i. Let \( t_1 \) and \( t_2 \) be the two triangles sharing that edge.
      ii. Let \( t_3 \) and \( t_4 \) be the two other triangles sharing an edge with \( t_1 \), and let \( t_5 \) and \( t_6 \) be the triangles sharing an edge with \( t_2 \)
      iii. Compute an E-vertex of that edge as a linear combination of the vertices of the above triangles as follows:
         \[ v_{e_i}^{j+1} = \sum_{k=1}^{8} \alpha_k c_k \]
         where \( \alpha_k = -1 \) for \( k = 1, 3, 6, 7 \)
         \( \alpha_k = 2 \) for \( k = 2, 8 \)
         \( \alpha_k = 8 \) for \( k = 4, 5 \)
The Modified Butterfly scheme (3)

Steps:
(b) Else, if one of the vertices of the edge is \( n \)-valent, where \( n \neq 6 \), (an extraordinary vertex say \( v_i^j \)), then:

i. Let \((c_k)(0 \leq k \leq n-1)\) be the one-ring vertices around \( v_i^j \)

ii. Compute the E-vertex of the edge as a linear combination of the 1-ring vertices as follows:

\[
\mathbf{v}_{e_i}^{j+1} = \sum_{k=0}^{n-1} \alpha_k \mathbf{c}_k
\]

where the \( \alpha_k \) depend on the valence \( n \):

\[
\alpha_k = \frac{1}{n} \left( \frac{1}{4} + \cos \frac{2\pi k}{n} + \frac{1}{2} \cos \frac{4\pi k}{n} \right)
\]  \text{for} \ n > 5

\[
\alpha_0 = \frac{5}{12}, \quad \alpha_1 = \alpha_2 = -\frac{1}{12}
\]  \text{for} \ n = 3

\[
\alpha_0 = \frac{3}{8}, \quad \alpha_2 = -\frac{1}{8}, \quad \alpha_1 = \alpha_3 = 0
\]  \text{for} \ n = 4
The Modified Butterfly scheme (4)

Steps:
(c) Else:
• compute an average of the coefficients obtained by treating each vertex as extraordinary vertex
• use the resulting mask to compute the E-vertex of that edge
The Modified Butterfly scheme (5)

Observations:
1. The scheme is interpolating: all initial vertices are part of the refined polyhedra
2. The refinement can be done adaptively
3. For regular meshes the scheme is only $C^1$
4. For irregular topology produces smooth $C^1$ surfaces
The Midpoint Subdivision scheme

- The simplest subdivision scheme

**Steps:**

1. For each edge $e^j_i$, compute its E-vertex as the midpoint of that edge.
2. Construct a new polyhedron as follows:
   (a) For each face $f^j_i$, construct an F-face by connecting the E-vertices of its edges.
   (b) For each vertex $v^j_i$, construct a V-face by connecting the E-vertices of the edges incident to it.
The Midpoint Subdivision scheme (2)

Observations:

1. 2 steps resemble 1 step of Doo-Sabin with different coefficients:
   (a) On each \( n \)-sided face \( f_i^j \), generate \( n \) vertices \( v_{i}^{j+1} \) as linear combinations of the old vertices \( v_i^j \):
      \[
      v_{i}^{j+1} = \sum_{k=1}^{n} \alpha_r v_k^j
      \]
      where \( r = (k - j + n) \mod n \) and the coefficients \( \alpha_r \) are given by:
      \[
      \alpha_r = 2\sum_{j=0}^{\bar{n}} 2^{-j} \cos \frac{2\pi r j}{n} \quad \text{and} \quad \bar{n} = \left\lfloor \frac{n-1}{2} \right\rfloor
      \]

2. The limit surface is \( C^2 \)

3. The algorithm converges slowly for large \( n \)-sided faces
The $\sqrt{3}$ scheme

- Is a triangular based scheme
- Is not an interpolating scheme
- Has the ability to accommodate adaptive subdivision
- The limit surface is $C^2$ except at the extraordinary vertices where it is $C^1$
The $\sqrt{3}$ scheme (2)

Steps:
1. For each face $f_i^j$, generate an F-vertex as the centroid of that face.
2. For each $n$-valent vertex $v_i^j$, do the following:
   (a) Let $(b_k)$ be the 1-ring vertices around $v_i^j$.
   (b) Generate a V-vertex $v_i^{j+1}$:
       $$v_i^{j+1} = (1 - \alpha_n)v_i^j + \frac{\alpha_n}{n} \sum_{k=1}^{n} b_k$$

where \( \alpha_n = \frac{1}{9} \left( 4 - 2 \cos \left( \frac{2\pi}{n} \right) \right) \)

3. Construct a new polyhedron as follows:
   (a) For each old edge, connect the F-vertices (centroid) of the two faces common to that edge.
   (b) For each old face, connect its F-vertex to the V-vertices of its corresponding vertices.
Manipulation of Subdivision Surfaces

- Subdivision algorithms are considered as smoothing operators
- But a smooth surface is not needed at all times
- Often, it is needed to generate a surface with a crease or a sharp edge
  - Subdivision surfaces with sharp features are a necessity in modeling and animation
- Two types of subdivision algorithms:
  1. Interpolating
     - Interpolates some or all of its vertices
     - E.g. the butterfly, the $\sqrt{3}$ schemes
  2. Approximating
     - Approximates an initial given polygon
     - E.g. Catmull-Clark, Doo-Sabin, midpoint, Loop schemes
- Approximating schemes can be made interpolating
Sharp Features

• Reduce the smoothness of a subdivision surface to $C^0$ at a vertex by modifying appropriate masks

• **Sharp vertex:**
  - Is the vertex at which we want to reduce the smoothness
  - Is labeled according to the number of tagged sharp edges incident to it
  - If #tagged edges > 2 then is a dart vertex
  - If #tagged edges = 2 then is a crease vertex
  - Otherwise is a corner vertex

E.g. In the Catmull-Clark scheme:

• If we modify the V-vertex coefficients of a tagged vertex $v_i^0$ and its adjacent E-vertices as shown in the figure on the next slide,

• Then that vertex will generate a dart
Sharp Features (2)

Mask for a crease vertex

Mask for a dart vertex

Mask for E-vertex incident to any sharp vertex

An example of Catmull-Clark subdivision surface with a dart
Open Polyhedra

- Most subdivision schemes discussed so far:
  - Work nicely for closed polyhedra
  - Lack control of the boundary curves of the limit surfaces generated from open polyhedra
- Why?
  - A limit surface from an open polyhedron shrinks to its interior
  - It is hard to control its boundary curves
Open Polyhedra (2)

- **Method 1:**
  - Modify the boundary faces to have some specific structure
  - So that a limit surface has its boundary curves controlled by the boundary vertices of the initial configuration
  - These vertices form the boundary control polygon of the surface

- E.g. consider a Doo-Sabin surface:
  - All boundary surfaces are 3-valent
  - Modify the boundary faces: extend every edge $v_i v_j$ by reflecting its interior $v_j$ symmetrically about $v_i$
Open Polyhedra (3)

- **Method 2:**
  - N-reflected faces
  - Used for more complicated boundary situations
  - **Advantage:** maintains the same subdivision coefficients, so:
    1. Not specialized analysis of the limit surface
    2. Two subdivision surfaces can be joined with the smoothness across their boundary curves
Open Polyhedra (4)

• **Method 3:**
  - Modify the subdivision coefficients along the boundary
  - Refine the boundary control polygon by using a curve subdivision algorithm
  - E.g. In the Catmull-Clark scheme, follow the steps:
    1. **For each boundary edge**
       
       generate an E-vertex at its midpoint
    2. **For each boundary vertex**
       
       generate a V-vertex using cubic curve subdivision
  - Similar algorithm exists for the Doo-Sabin scheme
Interpolation in Approximating Schemes

- Aim: extend approximating schemes so as to interpolate some of the control vertices
- Given a polyhedron $P$ with a set of tagged vertices $v_k$ to be interpolated, find another polyhedron $Q$ with a set of vertices $w_k$ whose limit surface interpolates the tagged surfaces $v_k$.
- Assume that the polyhedron $Q$ has a similar topology to $P$
- Treat every tagged vertex $v_k$ as a limit vertex $w_k^\infty$ to which a face or vertex converges
- Express $v_k$ as a linear combination of a number of vertices ($w_k$)
- Set up a system of linear equations whose solution gives $w_k$
- For Doo-Sabin surfaces:
  - A centroid of a face is a limit point on the surface
  - Associate every tagged vertex $v_k$ with a centroid of a certain face
  - Compute the matrix $M$ of the linear system with the following algorithm:
Interpolation in Approximating Schemes (2)

Algorithm:

1. Initialize all elements of the \( l \times l \) matrix \( M \) to zero, where \( l \): the total number of vertices of the original polyhedron.

2. For each \( n \)-valent vertex \( w_k \) do the following:
   
   If \( w_k \) is to be interpolated then:
   
   a) Let \( VF_k \) be the V-face generated from that vertex.
   
   b) Let \( (w^1_i)_{1 \leq i \leq n} \) be the vertices of \( VF_k \).

   The superscript indicates that the vertices belong to the 1\(^{st}\) subdivision.

   c) Form the equation:

   \[
   v_k = \frac{1}{n} \left( \sum_{i=1}^{n} w^1_i \right)
   \]
Interpolation in Approximating Schemes (3)

*Algorithm (cont):*

d) Replace every vertex $w^1_i$ by the linear combination of the vertices $w_k$ of the face it belongs to.

Assuming that this face is $m$-sided then:

$$w^1_i = \sum_{r=1}^{m} \alpha_{ri} w_r$$

which gives:

$$v_k = \frac{1}{n} (\sum_{i=1}^{n} \sum_{r=1}^{m} \alpha_{ri} w_r)$$

e) Form the row $k$ of the matrix $M$ using the coefficients

$$\frac{1}{n} \alpha_{ri}$$

This defines the $m \times n$ elements of this row

The remaining $l - m \times n$ elements are set to zero

Else, set $w_k = v_k$ so the corresponding row of $M$ is 0 everywhere except at position $k$ where it is 1
Interpolation in Approximating Schemes (4)

Algorithm (cont):

3. Set up the system of equations:

\[
\begin{pmatrix}
v_1 \\
v_2 \\
v_3 \\
\vdots \\
v_n
\end{pmatrix}
= M \cdot
\begin{pmatrix}
w_1 \\
w_2 \\
w_3 \\
\vdots \\
w_n
\end{pmatrix}
\]

4. Solve the system for the unknown vertices \( w_k \)

5. Construct a new polyhedron \( Q \) from a copy of \( P \) but with new vertices given by the solution of the system above.
Interpolation in Approximating Schemes (5)

A Doo-Sabin surface
Both surfaces interpolate the four top vertices of a cube

A Catmull-Clark surface
Interpolation in Approximating Schemes (6)

- For Catmull-Clark surfaces:
  - Use a similar algorithm
  - Associate every vertex to be interpolated with the limit of its V-vertex
  - A limit vertex is given in terms of vertices of the 1st refinement
  - Replace each of these by their combinations of the vertices $w_k$
  - Each vertex will correspond to a row of the matrix needed to solve the underlying linear system

- For closed polyhedra:
  - The matrix $M$ is a square matrix
  - We have 1 equation for every unknown vertex
  - A solution exists as long as $M$ is not singular
Interpolation of Curves by Subdivision Surfaces

- Given a tagged control polygon \( cp \) on a polyhedron \( P^0 \), force the limit surface of \( P^0 \) to interpolate the B-spline curve defined by \( cp \).

- Two types of curves:
  a. A curve with \( C^0 \) continuity
  b. A curve with \( C^1 \) continuity

A. A curve with \( C^0 \) continuity (crease)

- Generate a crease in one of 2 ways:
  i. Treat the \( cp \) as a boundary of 2 subdivision surfaces
     Each surface will have to undergo the procedure of boundary modification
  ii. Modify the subdivision coefficients, so that:
     Each refinement of the polyhedron will refine the tagged control polygon to generate the desired curve
Interpolation of Curves by Subdivision Surfaces (2)

A. A curve with $C^0$ continuity

- E.g. In Catmull-Clark surfaces, assume that the control polygon $cp$ is given by vertices $c_i$.

**Algorithm:**

1. For each edge $v_{i-1}v_i$ of the control polygon $cp$ make its E-vertex the midpoint of that edge
2. For each vertex $c_i$ of the control polygon make its V-vertex: $v_{i-1} + 6v_i + v_{i+1} = \frac{8}{8}$
3. For all other edges and vertices generate the E- and V-vertices as indicated by the Catmull-Clark subdivision scheme
Interpolation of Curves by Subdivision Surfaces (3)

B. A curve with $C^1$ continuity

- *Polygonal complex*: is a polyhedron $C$ that converges to a curve under a given subdivision scheme $S$
- Embed such a complex in a polyhedron $P \rightarrow$ it will generate a limit surface that interpolates the curve defined by $C$
- For the Doo-Sabin scheme $S$:
  - the simplest of these complexes is a strip of quads
  - It converges to a quadratic B-spline curve.
  - The control vertices are the midpoints of the shared edges between the adjacent quads
  - If two such complexes share one quad $q$, then the resulting 2 curves will intersect at the centroid of $q$.
  - If more than 2 curves are to be interpolated through an extraordinary vertex, then an $n$-reflected face could be used as the shared face between the two corresponding complexes.
Interpolation of Curves by Subdivision Surfaces (5)

• For the Catmull-Clark scheme:
  ■ A polygonal complex can be defined by 2 adjacent rows of faces
  ■ It converges to its corresponding cubic B-spline curve
  ■ The control vertices of this curve are also computed from the vertices of the shared edges between the faces of the complex

• Based on curve interpolation, lofted subdivision surfaces can be generated.

• Given a set of cross section curves:
  ■ construct a polygonal complex for each of these curves
  ■ connect these complexes into one polyhedron whose limit surface interpolates these curves.
Interpolation of Curves by Subdivision Surfaces (4)

A Doo-Sabin surface interpolating a crease

A Catmull-Clark surface interpolating a $C^1$ continuous curve