Subdivision Curves and Surfaces: An Introduction
De Casteljau’s and de Boor’s algorithms all use corner-cutting procedures.

Corner cutting can be local or non-local.

A cut is *local* if it removes exactly one vertex and adds two new ones. Otherwise, it is *non-local*.
Simple Corner Cutting: 1/4

- On each leg, choose two numbers $u \geq 0$ and $v \geq 0$ and $u+v \leq 1$, and divide the leg in the ratio of $u:1-(u+v):v$.

- Here is how to cut a corner.
Simple Corner Cutting: \(2/4\)

\[ u = 1/3 \text{ and } v = 1/4 \]
**Simple Corner Cutting: 3/4**

- For a polygon, one more leg from the last point to the first must also be divided accordingly.

\[ u = \frac{1}{3} \text{ and } v = \frac{1}{4} \]
The following result was proved by Gregory and Qu, de Boor, and Paluszny, Prautzsch and Schäfer.

If all $u$'s and $v$'s lies in the interior of the area bounded by $u \geq 0$, $v \geq 0$, $u+2v \leq 1$ and $2u+v \leq 1$, then $P_\infty$ is a $C^1$ curve.

This procedure was studied by Chaikin in 1974, and was later proved that the limiting curve is a B-spline curve of degree 2.
Chaikin’s Algorithm

- In Chaikin’s original study, \( u = 1/4 \) and \( v = 3/4 \). This process converges to a B-spline curve of degree 2.
- In fact, Chaikin’s algorithm is simply a knot insertion process (i.e., repeatedly inserting knots to a B-spline curve of degree 2).
- Consider inserting a knot at the midpoint of \( u_k \) and \( u_{k+1} \). Then, \( a_k \) and \( a_{k-1} \) divide their legs in the ratios of \( 1/4:3/4 \) and \( 3/4:1/4 \).
What is a Mesh?

- A two-dimensional *mesh* consists of a list of faces. If every edge is shared by exactly two faces, this mesh is closed. Otherwise, this mesh has boundaries. The boundaries are formed by those edges that have only one adjacent face.

- In what follows, we assume meshes are closed 2D-manifolds with arbitrary topology.

- The number of incident edges of a vertex is the *valency* of that vertex.
FYI

- Subdivision and refinement has its first significant use in Pixar’s *Geri’s Game*.
- *Geri’s Game* received Academy Award for Best Animated Short Film in 1997.

Regular Quad Mesh Subdivision: 1/3

- Suppose all faces in a mesh are quadrilaterals and each vertex has four adjacent faces.
- From the vertices $C_1, C_2, C_3$ and $C_4$ of a quadrilateral, four new vertices $c_1, c_2, c_3$ and $c_4$ can be computed in the following way (mod 4):

$$c_i = \frac{3}{16} C_{i-1} + \frac{9}{16} C_i + \frac{3}{16} C_{i+1} + \frac{1}{16} C_{i+2}$$

- If we define matrix $Q$ as follows:

$$Q = \begin{bmatrix}
\frac{9}{16} & \frac{3}{16} & \frac{1}{16} & \frac{3}{16} \\
\frac{3}{16} & \frac{9}{16} & \frac{3}{16} & \frac{1}{16} \\
\frac{1}{16} & \frac{3}{16} & \frac{9}{16} & \frac{3}{16} \\
\frac{3}{16} & \frac{1}{16} & \frac{3}{16} & \frac{9}{16}
\end{bmatrix}$$
Then, we have the following relation:

\[
\begin{bmatrix}
c_1 \\
c_2 \\
c_3 \\
c_4
\end{bmatrix} = Q \cdot \begin{bmatrix}
C_1 \\
C_2 \\
C_3 \\
C_4
\end{bmatrix}
\]
Regular Quad Mesh Subdivision: 3/3

- $c_1$, $c_2$, $c_3$ and $c_4$ of the current face are joined with the $c_i$’s of the neighboring face to form new, smaller faces.
Arbitrary Grid Mesh

- In the above scheme, since all faces are quadrilaterals and each vertex has four adjacent faces, the new mesh has all quadrilateral faces.
- If a mesh does not satisfy this “regular” assumption, the above is not true.
- If a vertex is not adjacent with four faces (or does not have four incident edges), it is an extraordinary vertex, and a face that is defined by other than four vertices is an extraordinary face.
- If a mesh is not a regular quad one, extraordinary vertices and extraordinary faces will occur.
Doo-Sabin Subdivision: 1/6

Doo and Sabin, in 1978, suggested the following for computing $c_i$’s from $C_i$’s:

$$c_i = \sum_{j=1}^{n} \alpha_{ij}C_j$$

where $\alpha_{ij}$’s are defined as follows:

$$\alpha_{ij} = \begin{cases} 
\frac{n+5}{4n} & \text{if } i = j \\
\frac{1}{4n} \left[ 3 + 2 \cos \left( \frac{2\pi (i-j)}{n} \right) \right] & \text{otherwise}
\end{cases}$$
There are three types of faces in the new mesh.

- An **F-face** is obtained by connecting the $c_i$’s of a face.
- An **E-face** is obtained by connecting the $c_i$’s of the faces that share an edge.
- An **V-face** is obtained by connecting the $c_i$’s that surround a vertex.
Doo-Sabin Subdivision: 3/6

- Most faces are quadrilaterals. None four-sided faces are those $V$-faces and converge to points whose valency is not four (i.e., extraordinary vertices).

- Thus, a large portion of the limit surface are covered by quadrilaterals, and the surface is mostly a B-spline surfaces of degree (2,2). However, it is $G^1$ everywhere and mostly $C^1$.

- At any iteration, the $c_i$ vertices of any face will be on the limit surface because a face will produce a sequence of $F$-faces converging to its $c_i$’s.
Doo-Sabin Subdivision: 5/6
Doo-Sabin Subdivision: 6/6

1

2

3

4

5
Catmull-Clark Algorithm: 1/10

- Catmull and Clark proposed another algorithm in the same year as Doo and Sabin did (1978).
- In fact, both papers appeared in the journal *Computer-Aided Design* back to back!
- Catmull-Clark is rather complex. It computes a face point for each face, followed by an edge point for each edge, and then a vertex point for each vertex.
- Once these new points become available, a new mesh is constructed.
- We only consider 2D manifold without boundary.
Catmull-Clark Algorithm: 2/10

- Compute a face point for each face. This face point is the gravity center or centroid of the face, which is the average of all vertices of that face:
Compute an edge point for each edge. An edge point is the average of the two endpoints of that edge and the two face points of that edge’s adjacent faces.
Catmull-Clark Algorithm: 4/10

- Compute a **vertex point** for each vertex, where \(e_k\) (resp., \(f_k\)) are edge points (resp., face points) of the edge (resp., faces) incident to \(v\).

\[
v' = \frac{1}{n} Q + \frac{2}{n} R + \frac{n - 3}{n} v
\]

- **Q** – the average of all new face points of \(v\)
- **R** – the average of all edge-points of vertex \(v\)
- **v** – the original vertex
- **n** – # of incident edges of \(v\)

Use Q, R and v to compute \(v'\)
Catmull-Clark Algorithm: 5/10

- Connect each face point $f$ to each edge point of $f$’s edge, and connect each new vertex $v'$ to every edge point of the edges incident to $v$. 
Catmull-Clark Algorithm: 6/10

- Face point
- Edge point
- Vertex point
- Vertex-edge connection
- Face-edge connection
Catmull-Clark Algorithm: 7/10

- After the first run, all faces are four sided.
- If all faces are four-sided, each has four edge points $e_1$, $e_2$, $e_3$ and $e_4$, four vertices $v_1$, $v_2$, $v_3$ and $v_4$, and one new vertex $v$. Their relation can be represented as follows:

$$
\begin{bmatrix}
 v' \\
 e'_1 \\
 e'_2 \\
 e'_3 \\
 e'_4 \\
 v'_1 \\
 v'_2 \\
 v'_3 \\
 v'_4
\end{bmatrix} = \frac{1}{16}
\begin{bmatrix}
 9 & 3 & 3 & 3 & 3 & 1 & 1 & 1 & 1 \\
 2 & 2 & 2 & 2 & 4 & 4 & 4 & 4 & 4 \\
 6 & 6 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\
 6 & 1 & 6 & 1 & 0 & 1 & 1 & 0 & 0 \\
 6 & 0 & 1 & 6 & 1 & 0 & 1 & 1 & 0 \\
 6 & 1 & 0 & 1 & 6 & 0 & 0 & 1 & 1 \\
 4 & 4 & 4 & 0 & 0 & 4 & 0 & 0 & 0 \\
 4 & 0 & 4 & 4 & 0 & 0 & 4 & 0 & 0 \\
 4 & 0 & 0 & 4 & 4 & 0 & 0 & 4 & 0 \\
 4 & 4 & 0 & 0 & 4 & 0 & 0 & 0 & 4
\end{bmatrix}
\begin{bmatrix}
 v \\
 v_1 \\
 v_2 \\
 v_3 \\
 v_4 \\
 e_1 \\
 e_2 \\
 e_3 \\
 e_4
\end{bmatrix}.
$$

- A vertex at any level converges to the following:

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

- The limit surface is a B-spline surface of degree (3,3).
Catmull-Clark Algorithm: 8/10
Catmull-Clark Algorithm: 9/10
Catmull-Clark Algorithm: 10/10
Loop’s Algorithm: 1/5

Loop’s algorithm goes as follows:

- For each edge compute its edge point \( e \). Let the edge be \( v_1v_2 \) and the two other vertices of the incident triangles be \( v_{\text{left}} \) and \( v_{\text{right}} \).

\[
e = \frac{3}{8} (v_1 + v_2) + \frac{1}{8} (v_{\text{left}} + v_{\text{right}})
\]

- For each vertex \( v \), the new vertex point is computed below, where \( v_1, v_2, \ldots, v_n \) are adjacent vertices

\[
v' = (1 - n\alpha) v + \alpha \sum_{j=1}^{n} v_j
\]

where \( \alpha \) is

\[
\alpha = \begin{cases} 
\frac{3}{16} & \text{if } n = 3 \\
\frac{1}{n} \left[ \frac{5}{8} - \left( \frac{3}{8} + \frac{1}{4} \cos \frac{2\pi}{n} \right)^2 \right] & \text{if } n > 3
\end{cases}
\]
Loop’s Algorithm: 2/5

- Loop’s algorithm only works for triangular meshes.
- Let a triangle be defined by $X_1$, $X_2$ and $X_3$ and the corresponding new vertex points be $v_1$, $v_2$ and $v_3$.
- Let the edge points of edges $v_1v_2$, $v_2v_3$ and $v_3v_1$ be $e_3$, $e_1$ and $e_2$. The triangles are $v_1e_2e_3$, $v_2e_3e_1$, $v_3e_1e_2$ and $e_1e_2e_3$.
- Loop’s algorithm was developed by Charles Loop in 1987, and only works for triangular mesh.
Loop’s Algorithm: 3/5

- Pick a vertex in the original or an intermediate mesh. If this vertex has \( n \) adjacent vertices \( v_1, v_2, ..., v_n \), then it converges to \( v_\infty \):

\[
v_\infty = \frac{3 + 8\alpha(n - 1)}{3 + 8n\alpha} + \frac{8\alpha}{3 + 8n\alpha} \sum_{j=1}^{n} v_j
\]

- If all vertices have valency 6, the limit surface is a collection of \( C^2 \) Bézier triangles.

- However, a close surface cannot be formed with valency 6 vertices only. Vertices with different valencies converge to extraordinary vertices where the surface is only \( G^1 \).
Loop’s Algorithm: 4/5

Doo-Sabin

Catmull-Clark
Loop’s Algorithm: 5/5

Doo-Sabin

Catmull-Clark
This technique was proposed by Leif Kobbelt in 2000, and only works on triangular meshes.

This is a very simple algorithm and consists of two steps:

1) Dividing each triangle at the center into 3 more triangles
2) “Flip” the edges of the original triangle (see next slide).
\sqrt{3}\text{-Subdivision of Kobbelt: 2/6}

Step 1: Subdividing

- For each triangle, compute its center: 
  \[ C = \frac{(V_1 + V_2 + V_3)}{3} \]
- Connect the center to each vertex to create 3 triangles.
- This is a 1-to-3 scheme!
3-Subdivision of Kobbelt: 3/6
Step 2: Flipping Edges

Since each edge has two adjacent triangles, “flipping” an edge means removing the original and replaced by the new edge joining the centers.

Dotted: original    Solid: “flipped”
Remove the “flipped” edges and we have a triangular mesh!

But, the original vertices must also be “perturbed” to preserve “smoothness”.
\[ \sqrt{3}\text{-Subdivision of Kobbelt: 5/6} \]

**Actual Computation**

- For each triangle with vertices \( V_1, V_2 \) and \( V_3 \), compute its center \( C \):
  \[
  C = \frac{1}{3} [V_1 + V_2 + V_3]
  \]

- For each vertex \( V \) and its neighbors \( V_1, V_2, \ldots, V_n \), compute \( V' \) as follows:
  \[
  V' = (1 - \alpha_n) V + \frac{\alpha_n}{n} \sum_{i=1}^{n} V_i
  \]

  where \( \alpha_n \) is computed as follows:
  \[
  \alpha_n = \frac{1}{9} \left[ 4 - 2 \cos \left( \frac{2\pi}{n} \right) \right]
  \]
Important Results

- The $\sqrt{3}$-subdivision converges!
- The limit surface is $C^2$ everywhere except for extra-ordinary points.
- At extra-ordinary points, it is only $C^1$.
- The $\sqrt{3}$-subdivision can be extended to an adaptive scheme for finer subdivision control.