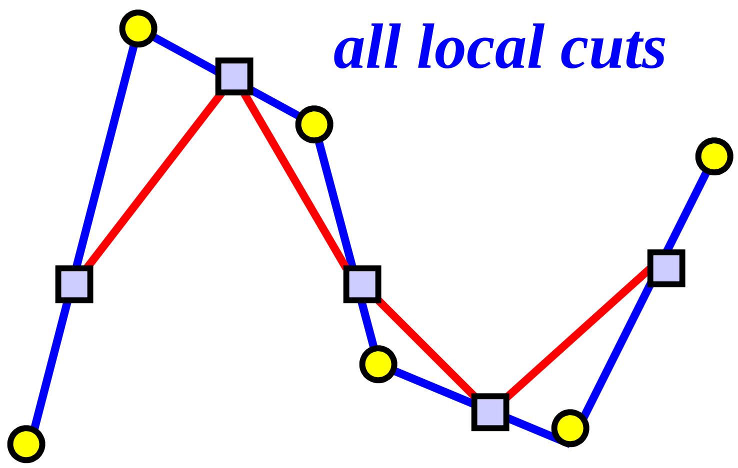
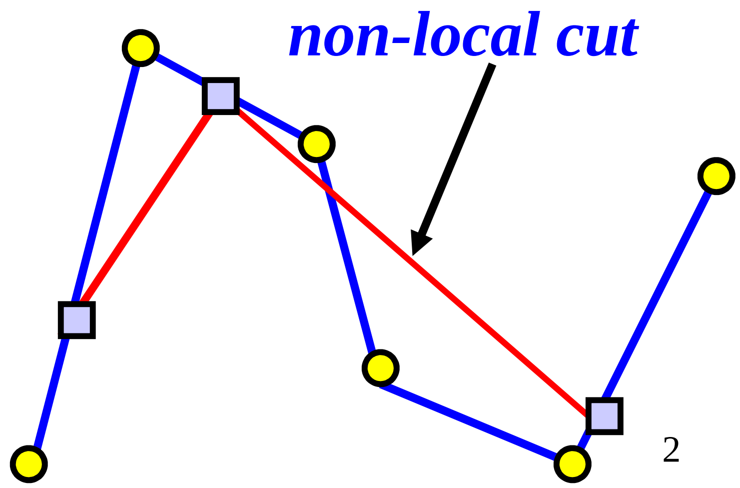
Subdivision Techniques

Curve Corner Cutting

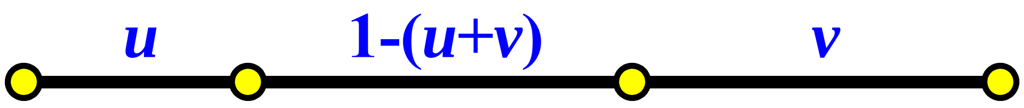
- **Take two points on different edges of a polygon and join them with a line segment. Then, use this line segment to replace all vertices and edges in between. This is corner cutting!**
- **Corner cutting can be local or non-local.**
- **A cut is *local* if it removes exactly one vertex and adds two new ones. Otherwise, it is *non-local*.**



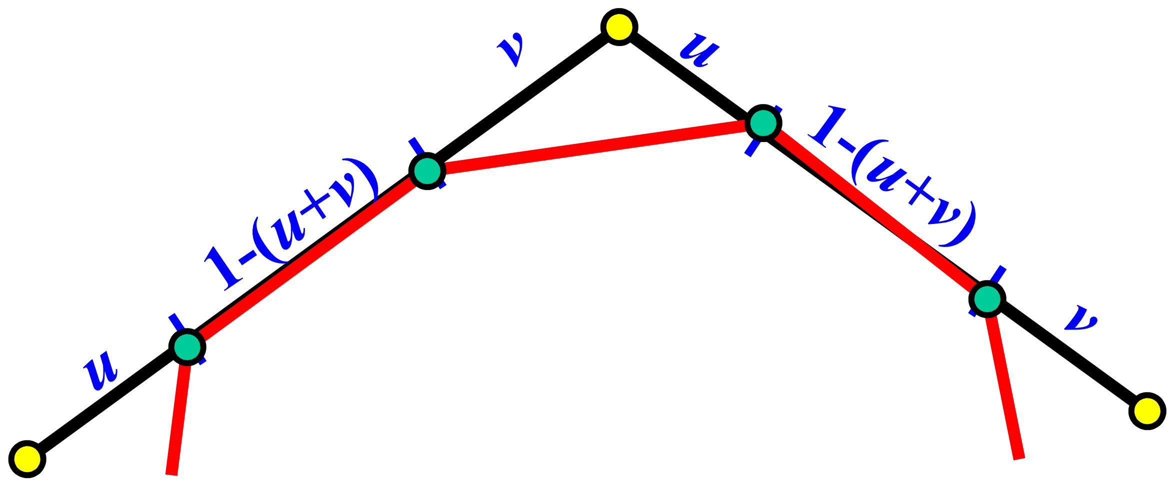


Simple Corner Cutting: 1/5

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



- **Here is how to cut a corner.**



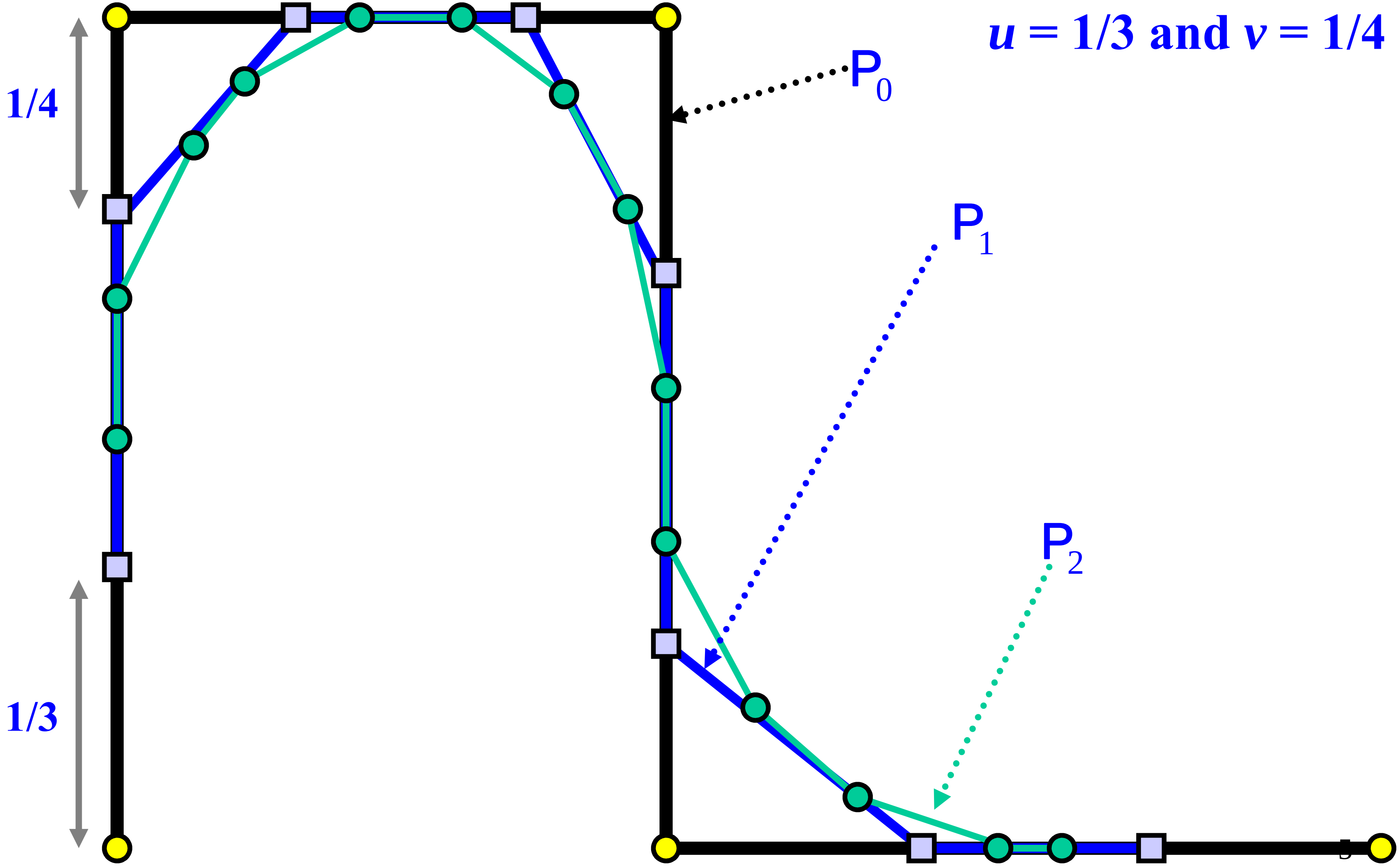
Simple Corner Cutting: 2/5

- **Suppose we have a polyline P_0. Divide its edges with the above scheme, yielding a new polyline P_1. Dividing P_1 yields P_2, …., and so on. What is**

$$P_\infty = \lim_{i \to \infty} P_i$$

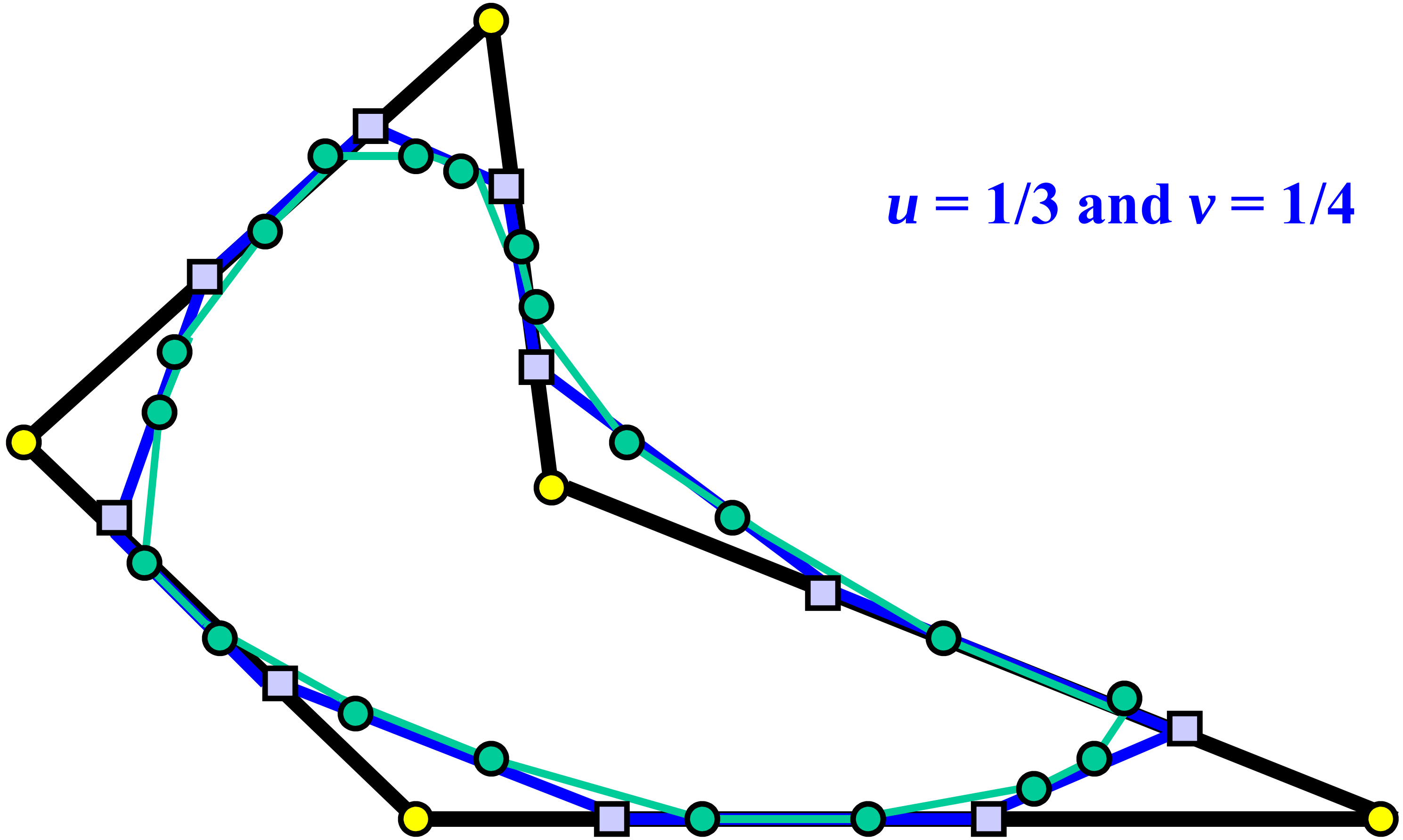
- **The u's and v's do not have to be the same for every edge. Moreover, the u's and v's used to divide P_i do not have to be equal to those u's and v's used to divide P_{i+1}.**

Simple Corner Cutting: 3/5

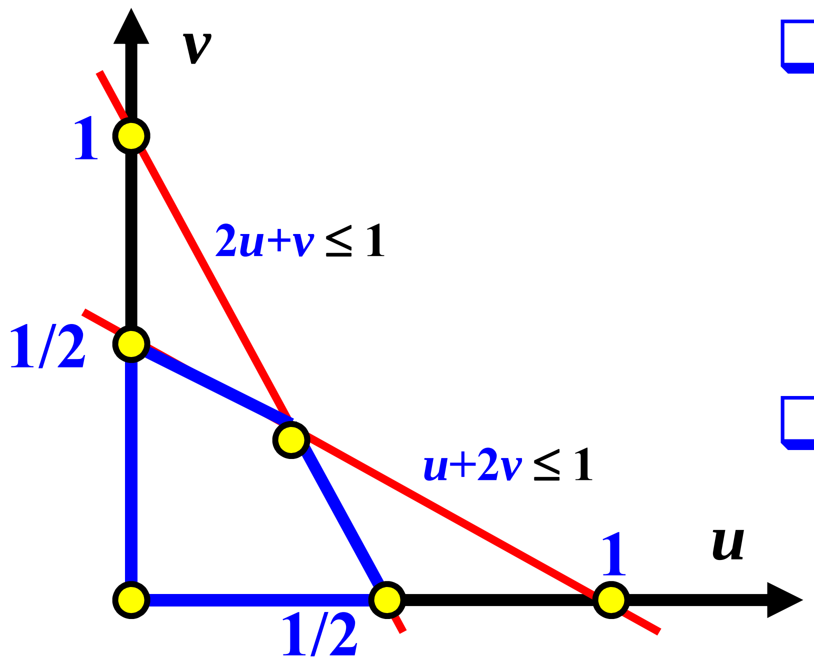


Simple Corner Cutting: 4/5

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



Simple Corner Cutting: 5/5



Chaikin used $u = v = 1/4$

- 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_∞ is a C^1 curve.
- This procedure was studied by Chaikin in 1974, and was later proved that the limit curve is a B-spline curve of degree 2.

FYI

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

- http://www.pixar.com/shorts/gg/

Facts about Subdivision Surfaces

- **Subdivision surfaces are *limit surfaces*:**
  - **It starts with a mesh**
  - **It is then refined by repeated subdivision**
- **Since the subdivision process can be carried out infinite number of times, the intermediate meshes are *approximations* of the actual subdivision surface.**
- **Subdivision surfaces is a simple technique for describing complex surfaces of arbitrary topology with guaranteed continuity.**
- **Also supports Multiresolution.**

What Can You Expect from …?

- **It is easy to model a large number of surfaces of various types.**
- **Usually, it generates smooth surfaces.**
- **It has simple and intuitive interaction with models.**
- **It can model sharp and semi-sharp features of surfaces.**
- **Its representation is simple and compact (*e.g.*, winged-edge and half-edge data structures, etc).**
- **We only discuss 2-manifolds without boundary.**

Regular Quad Mesh Subdivision: 1/3

- Assume 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):

$$\mathbf{c}_i = \frac{3}{16}\mathbf{C}_{i-1} + \frac{9}{16}\mathbf{C}_i + \frac{3}{16}\mathbf{C}_{i+1} + \frac{1}{16}\mathbf{C}_{i+2}$$

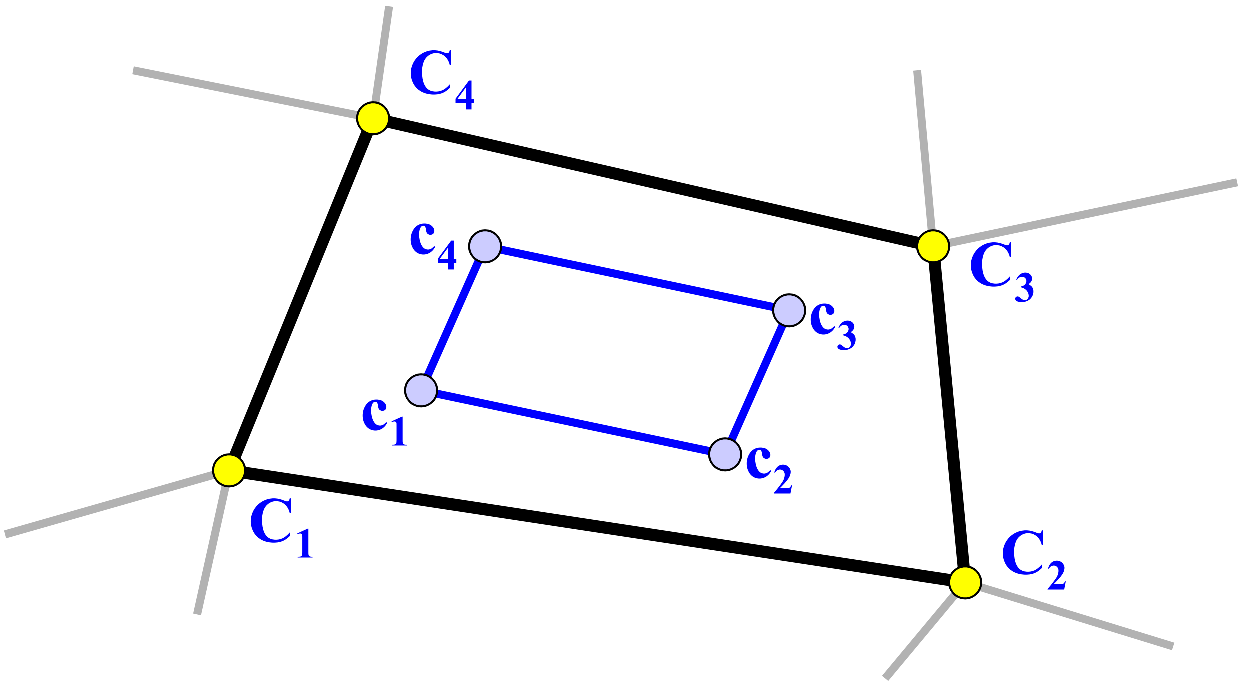
- If we define matrix $\mathbf{Q}$ as follows:

$$\mathbf{Q} = \begin{bmatrix} 9/16 & 3/16 & 1/16 & 3/16 \\ 3/16 & 9/16 & 3/16 & 1/16 \\ 1/16 & 3/16 & 9/16 & 3/16 \\ 3/16 & 1/16 & 3/16 & 9/16 \end{bmatrix}$$

Regular Quad Mesh Subdivision: 2/3

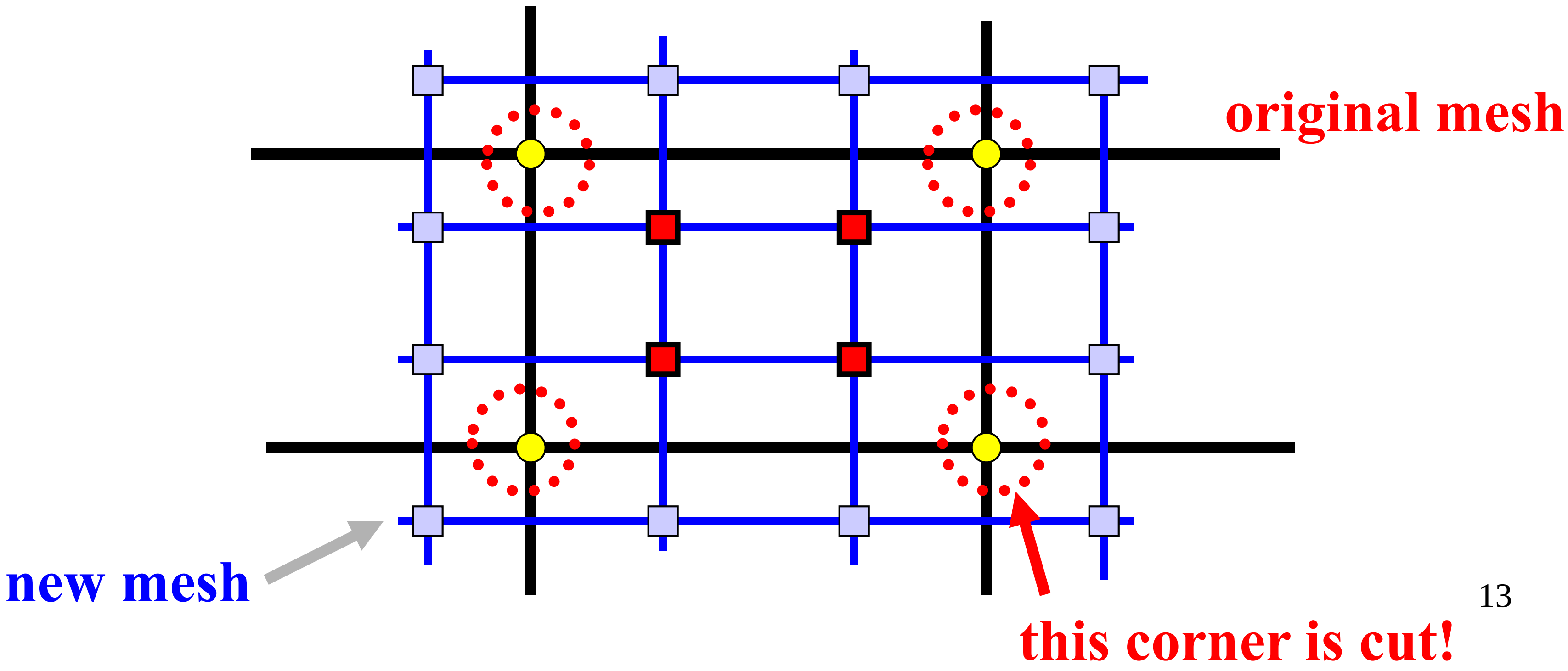
- **Then, we have the following relation:**

$$\begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \mathbf{c}_3 \\ \mathbf{c}_4 \end{bmatrix} = \mathbf{Q} \cdot \begin{bmatrix} \mathrm{C}_1 \\ \mathrm{C}_2 \\ \mathrm{C}_3 \\ \mathrm{C}_4 \end{bmatrix}$$



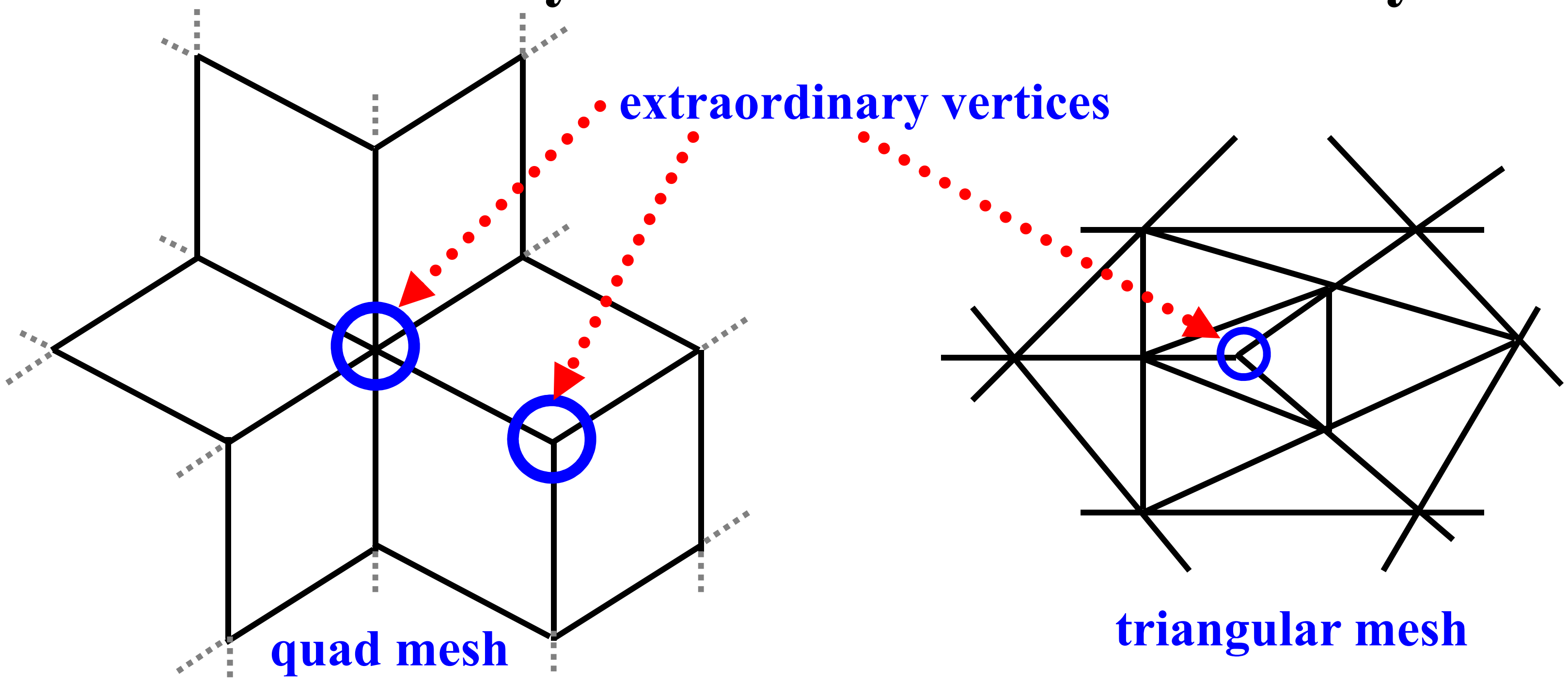
Regular Quad Mesh Subdivision: 3/3

- **New vertices c_1, c_2, c_3 and c_4 of the current face are connected to the c_i's of the neighboring faces to form new, smaller faces.**
- **The new mesh is still a quadrilateral mesh.**



Arbitrary Grid Mesh

- If a vertex in a quadrilateral (*resp*., triangular) mesh is not adjacent to four (*resp*., six) neighbors, it is an ***extraordinary vertex***.
- A non-regular quad or triangular mesh has extraordinary vertices and extraordinary faces.



Doo-Sabin Subdivision: 1/6

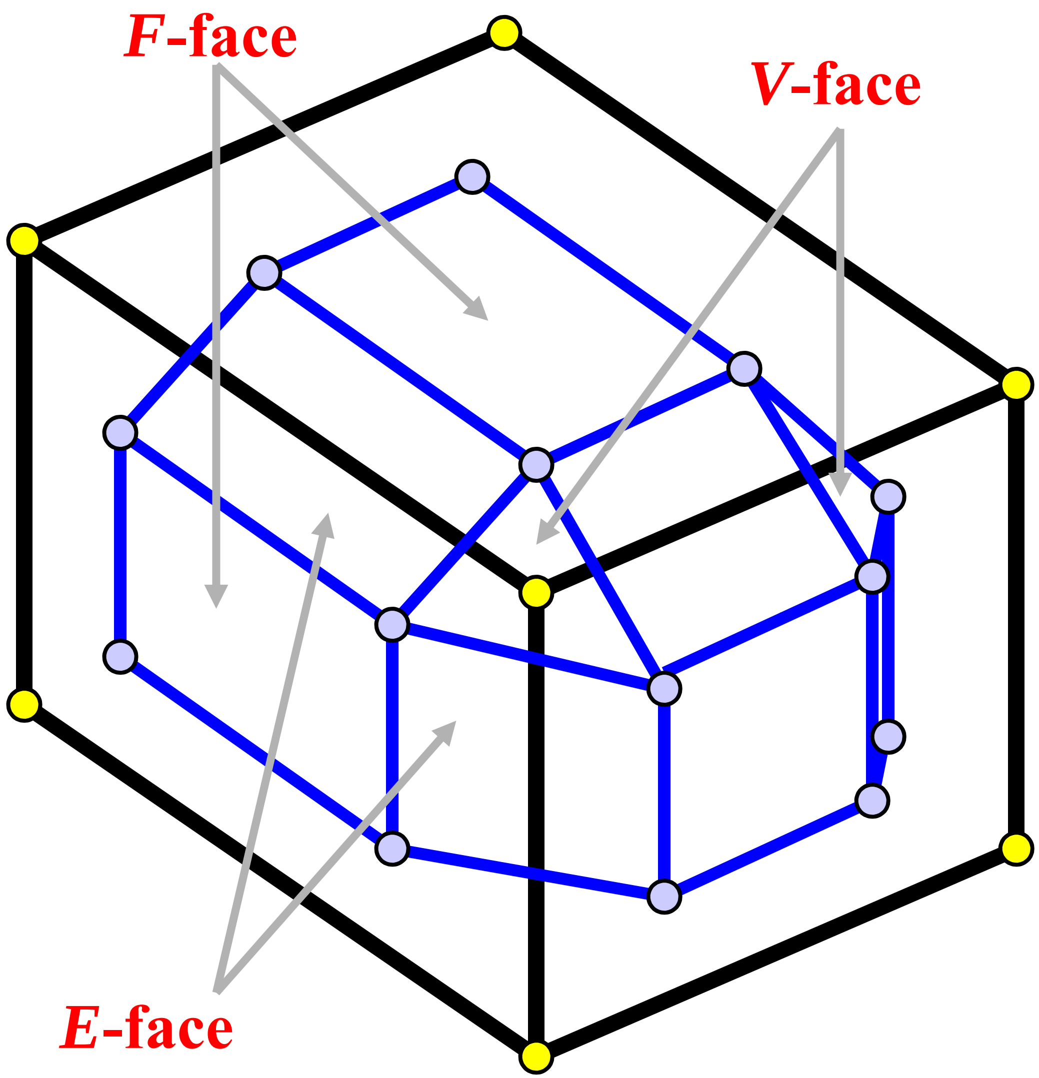
- **Doo and Sabin, in 1978, suggested the following for computing $\mathbf{c}_i$'s from $\mathbf{C}_i$'s:**

$$\mathbf{c}_i = \sum_{j=1}^{n} \alpha_{ij} \mathbf{C}_j$$

where α_{ij} 's are defined as follows:

$$\alpha_{ij} = \begin{cases} \dfrac{n+5}{4n} & i = j \\ \dfrac{1}{4n}\left[3 + 2\cos\left(\dfrac{2\pi(i-j)}{n}\right)\right] & \text{otherwise} \end{cases}$$

Doo-Sabin Subdivision: 2/6

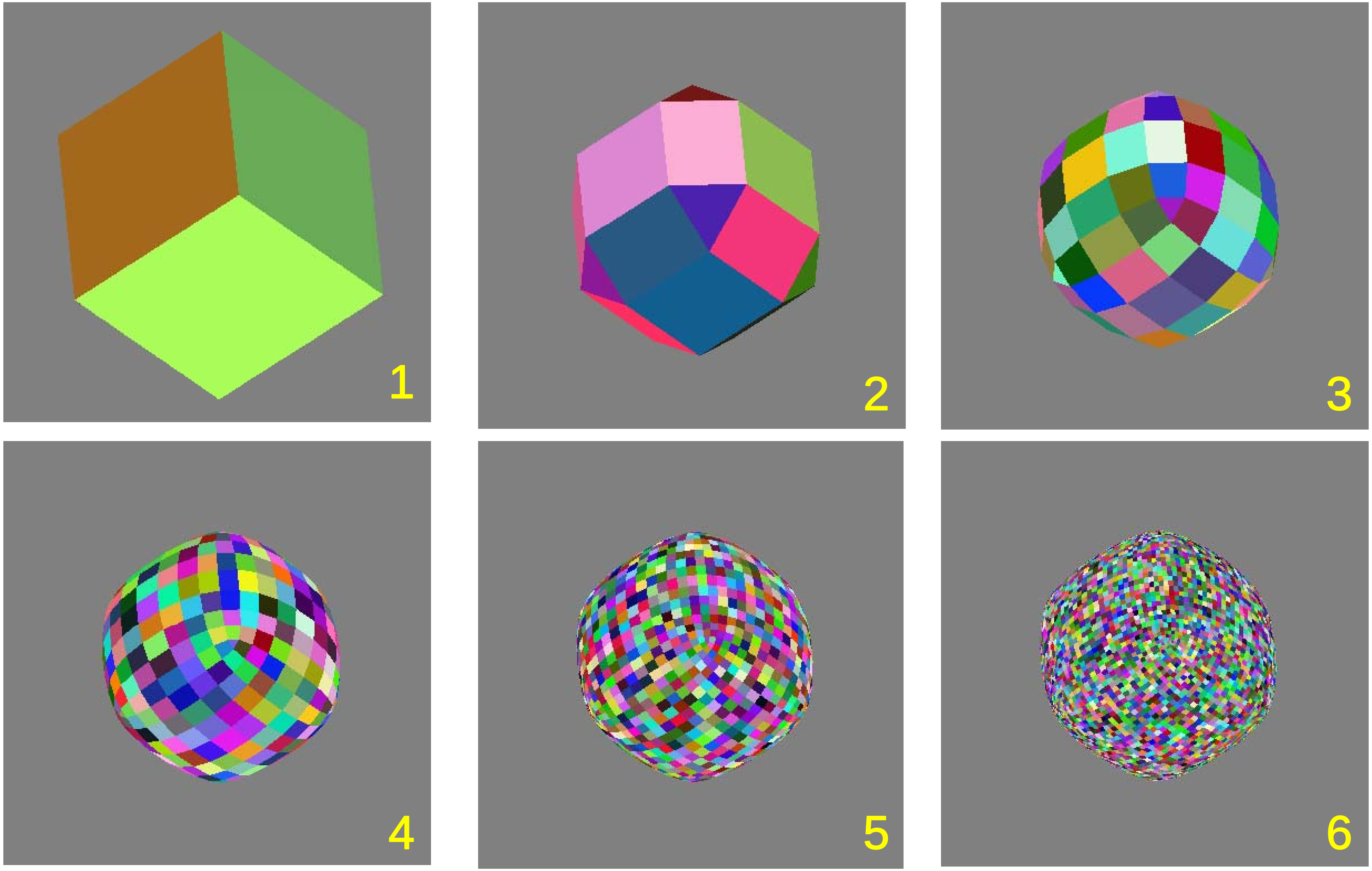


- There are ***three*** types of faces in the new mesh.
- A ***F*-face** is obtained by connecting the c_i's of a face.
- An ***E*-face** is obtained by connecting the c_i's of the faces that share an edge.
- A ***V*-face** is obtained by connecting the c_i's that surround a vertex.

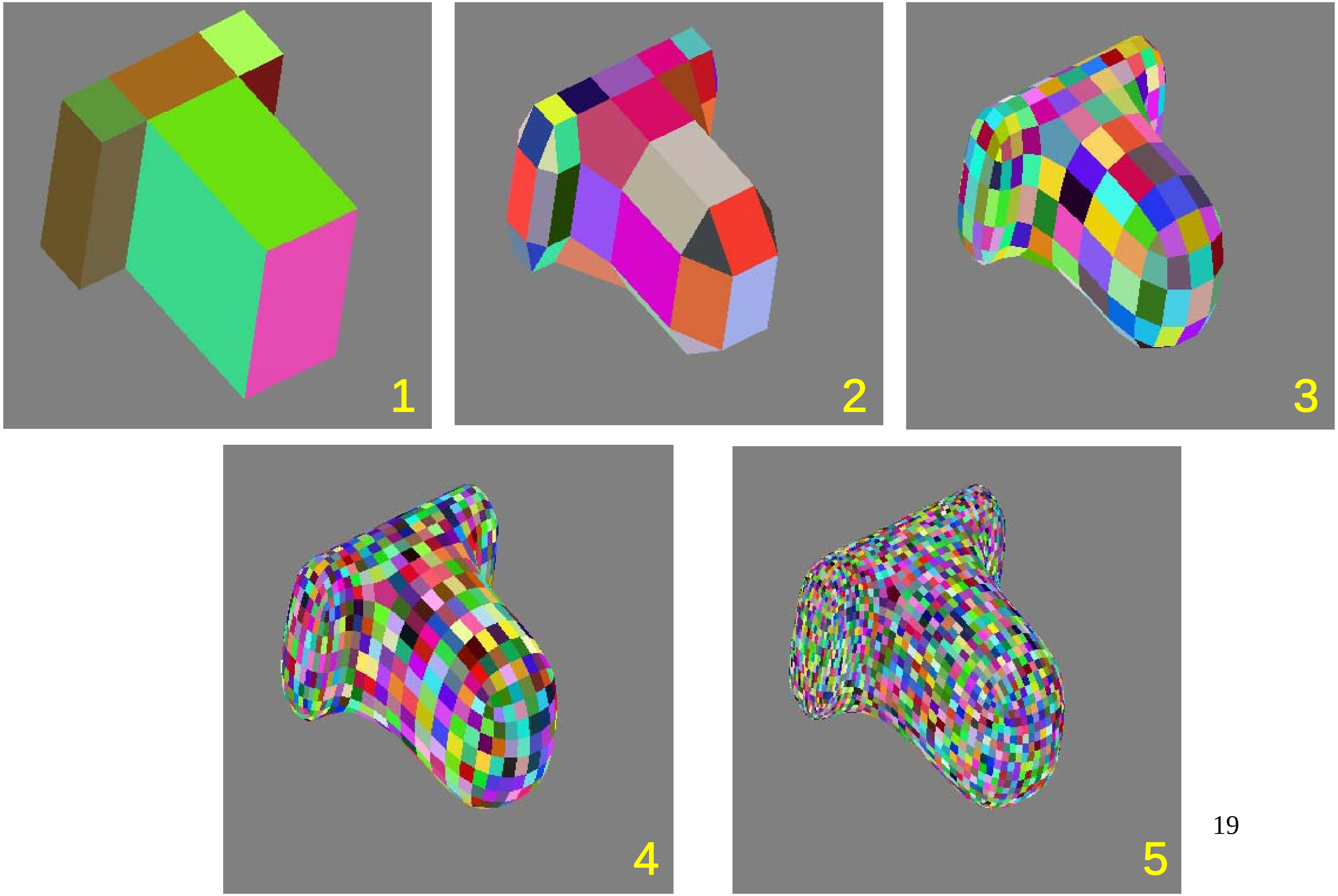
Doo-Sabin Subdivision: 3/6

- **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 only G^1 everywhere.**

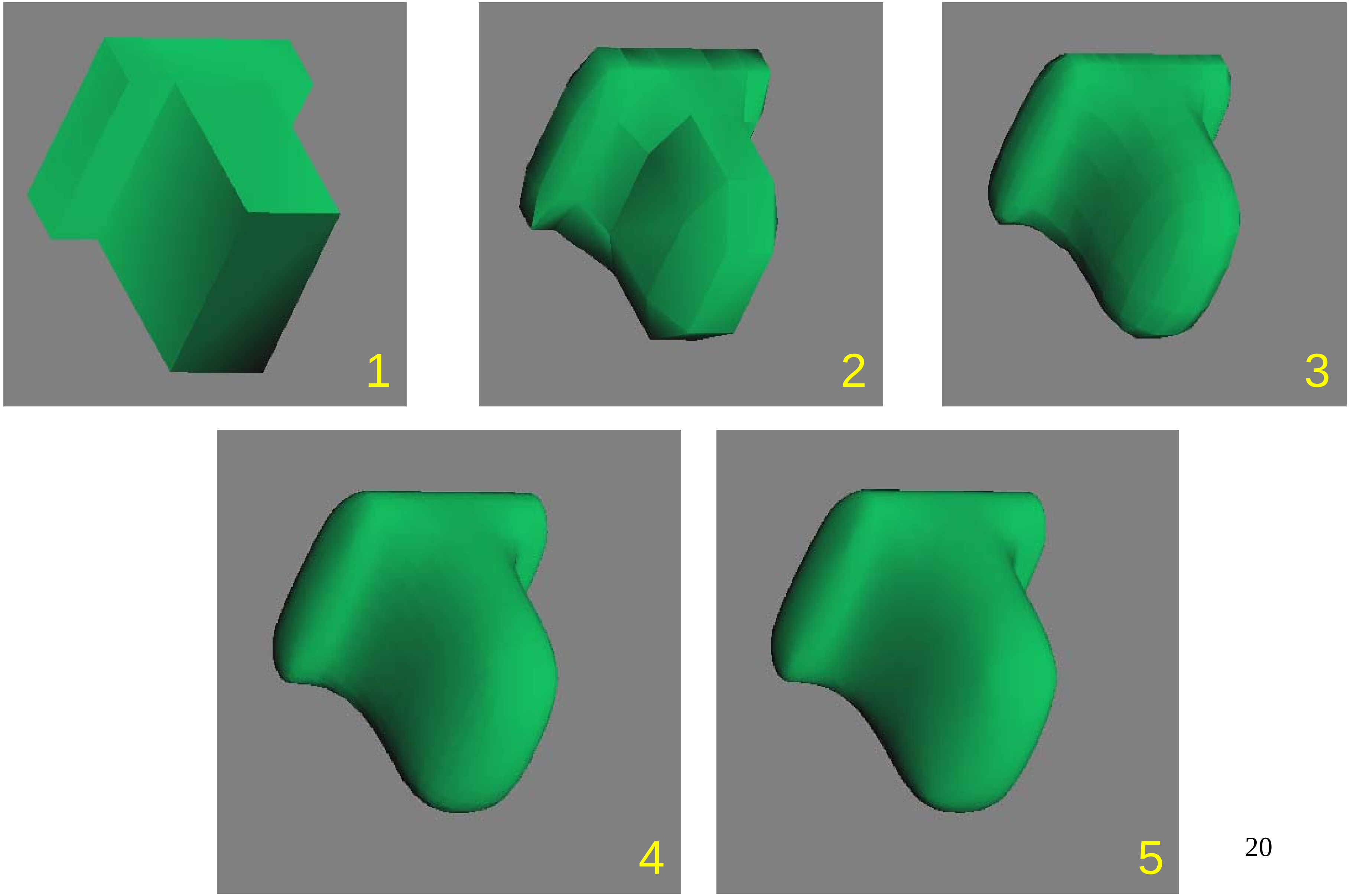
Doo-Sabin Subdivision: 4/6



Doo-Sabin Subdivision: 5/6



Doo-Sabin Subdivision: 6/6

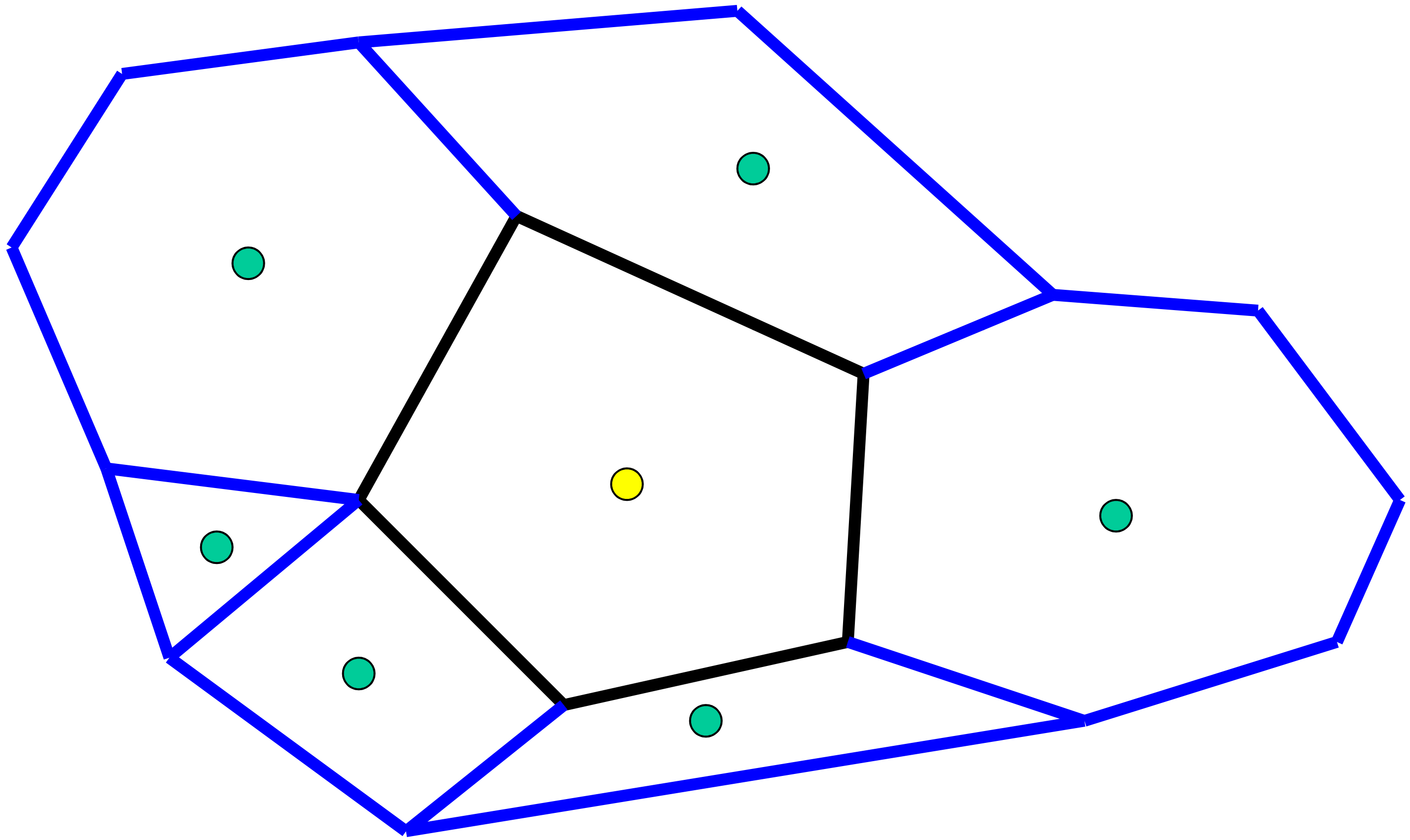


Catmull-Clark Algorithm: 1/10

- **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's algorithm 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 are available, a new mesh is constructed.**

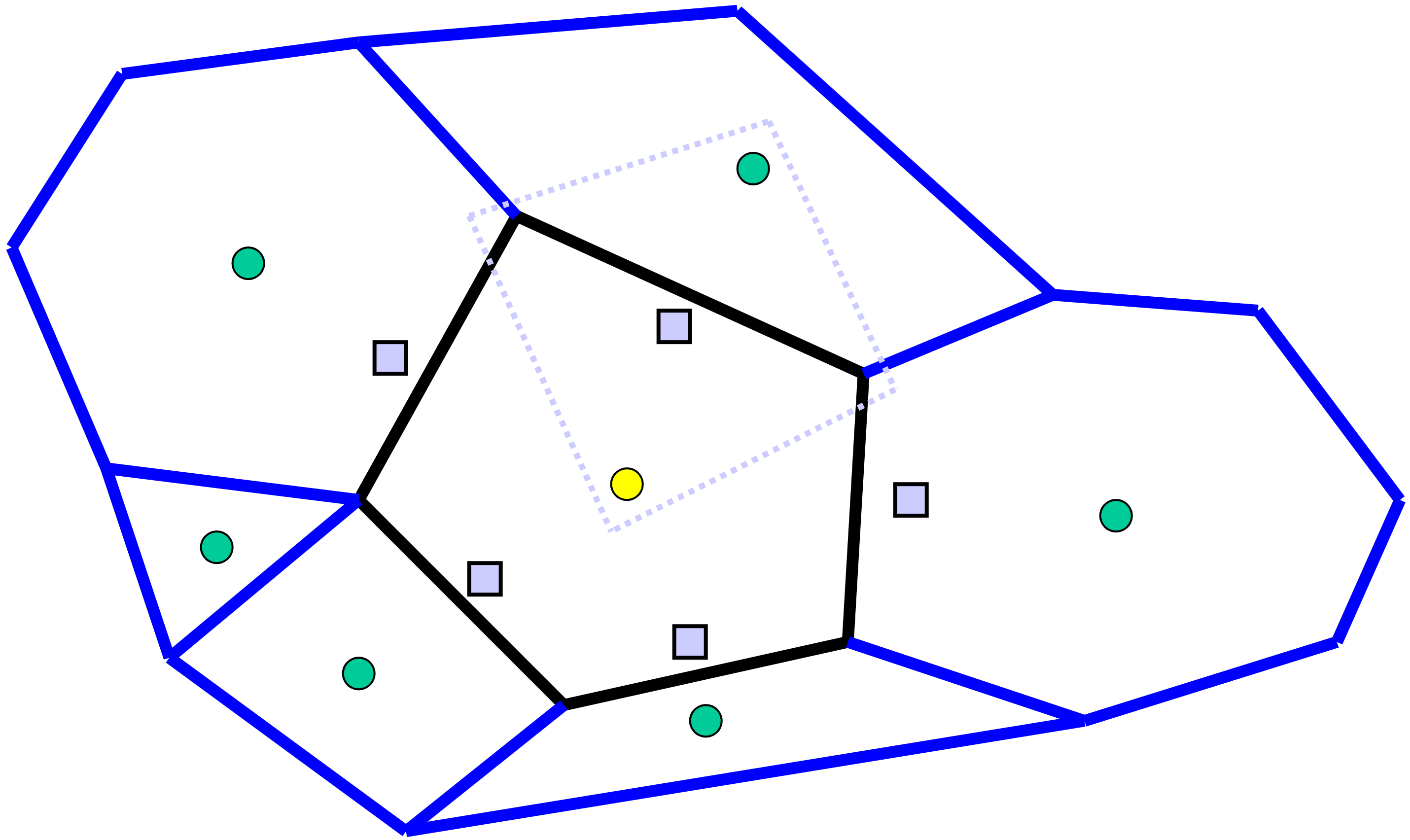
Catmull-Clark Algorithm: 2/10

- 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:



Catmull-Clark Algorithm: 3/10

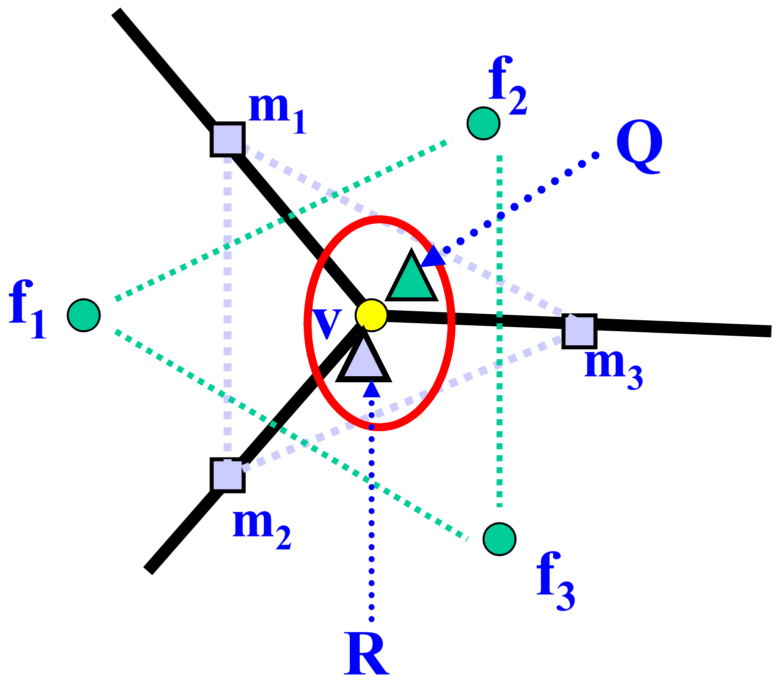
- Compute an edge point for each edge. An edge point is the average of the two endpoints of that edge and the two face points of that edge's adjacent faces.



Catmull-Clark Algorithm: 4/10

- Compute a **vertex point** for each vertex **v** as follows:

$$\mathbf{v}' = \frac{1}{n}\mathbf{Q} + \frac{2}{n}\mathbf{R} + \frac{n-3}{n}\mathbf{v}$$



Q – the average of all new face points of **v**

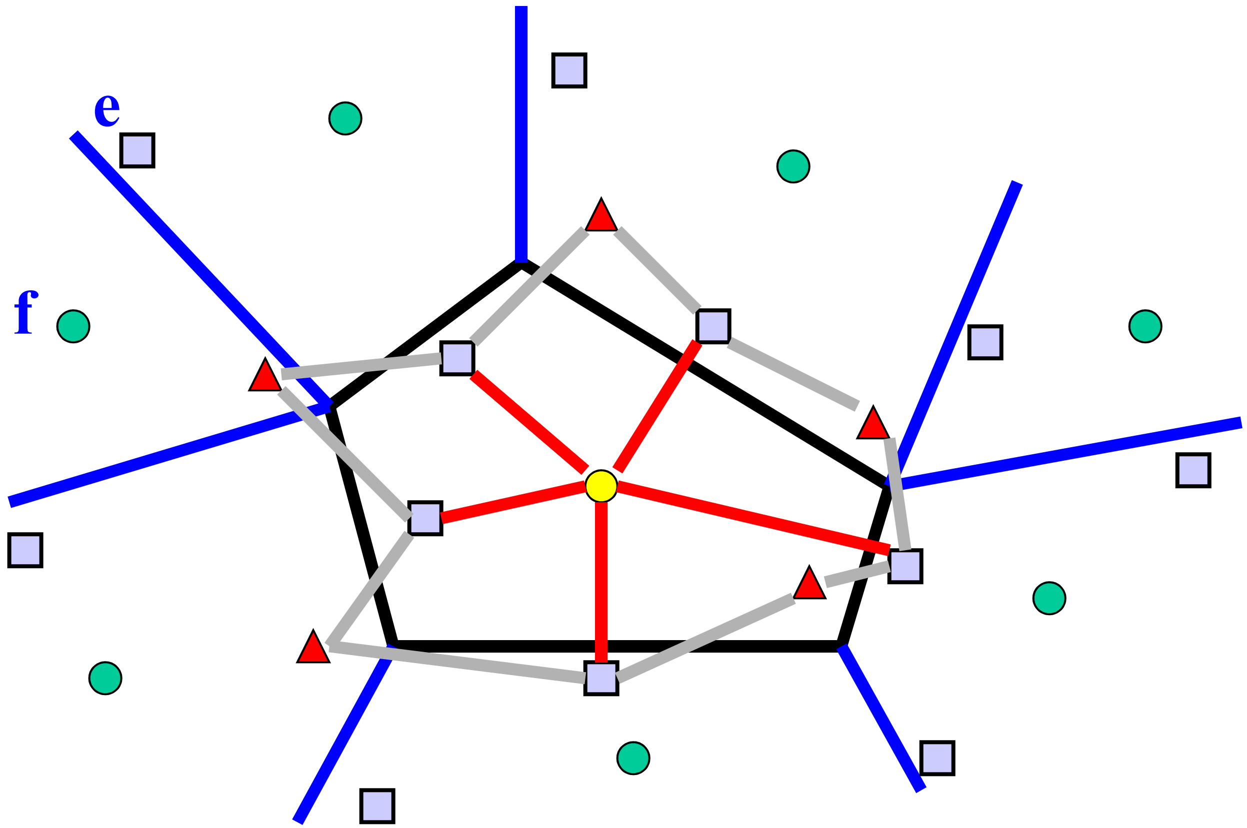
R – the average of all mid-points (*i.e.*, $\mathbf{m}_i$'s) of vertex **v**

v - the original vertex

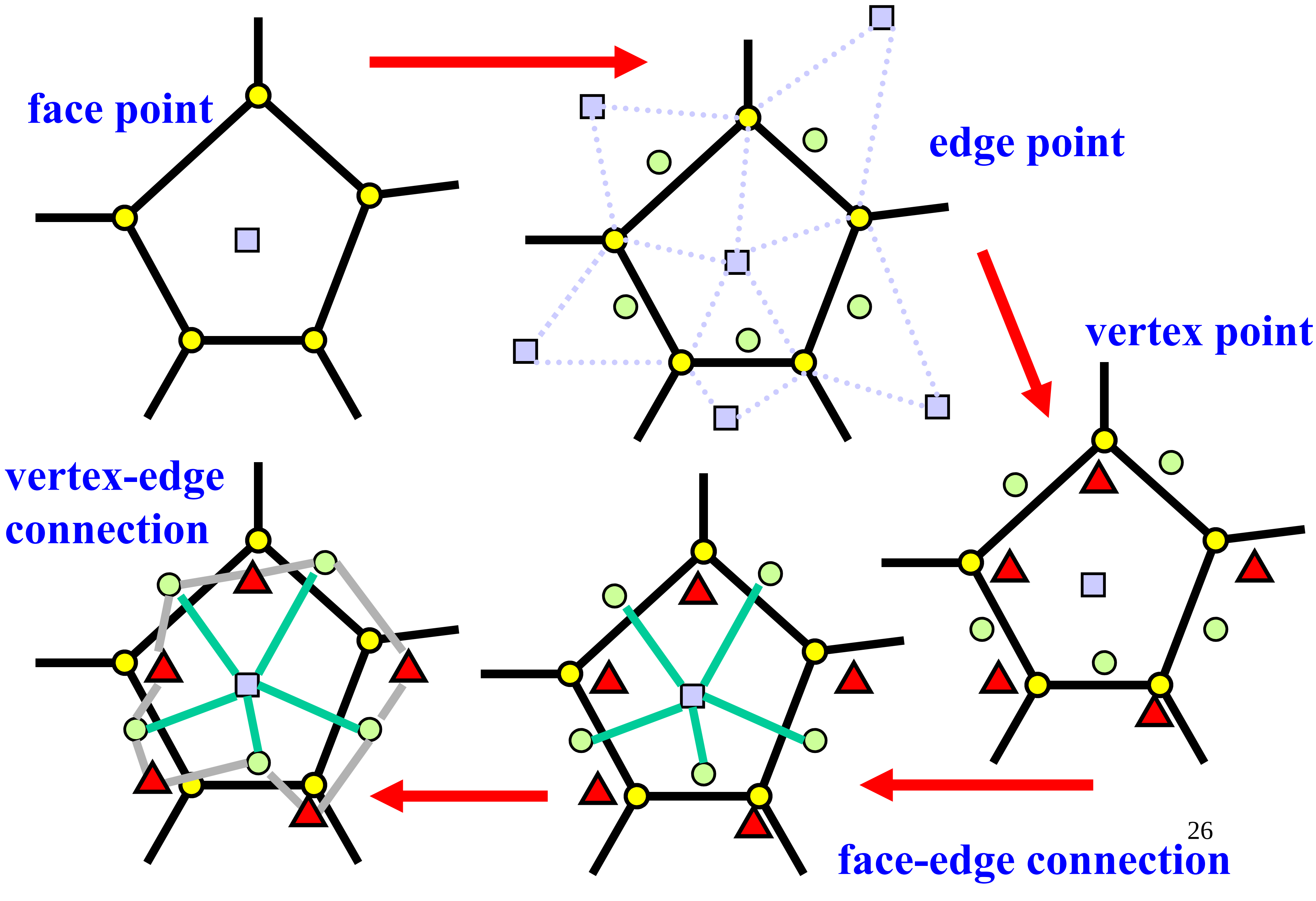
n - # of incident edges of **v**

Catmull-Clark Algorithm: 5/10

- For each face, connect its face point **f** to each edge point, and connect each new vertex **v'** to the two edge points of the edges incident to **v**.



Catmull-Clark Algorithm: 6/10



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 $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$ and $\mathbf{e}_4$, four vertices $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$ and $\mathbf{v}_4$, and one new vertex $\mathbf{v}$. Their relation can be represented as follows:

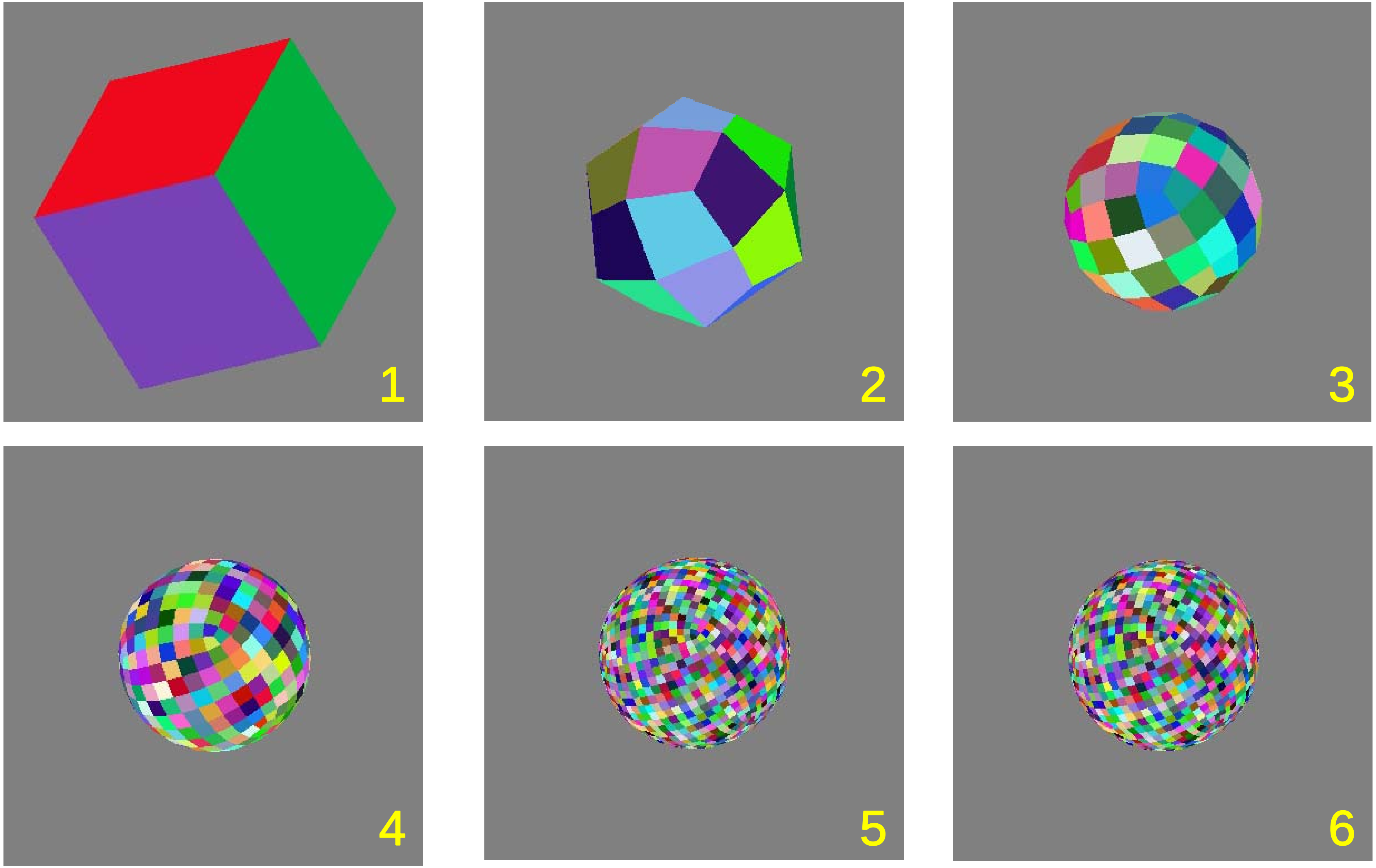
$$\begin{bmatrix} \mathbf{v}' \\ \mathbf{e}_1' \\ \mathbf{e}_2' \\ \mathbf{e}_3' \\ \mathbf{e}_4' \\ \mathbf{v}_1' \\ \mathbf{v}_2' \\ \mathbf{v}_3' \\ \mathbf{v}_4' \end{bmatrix} = \frac{1}{16} \begin{bmatrix} 9 & \frac{3}{2} & \frac{3}{2} & \frac{3}{2} & \frac{3}{2} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 6 & 6 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 6 & 1 & 6 & 1 & 0 & 1 & 1 & 0 & 0 \\ 6 & 0 & 1 & 6 & 1 & 0 & 1 & 1 & 0 \\ 6 & 1 & 0 & 1 & 6 & 0 & 0 & 1 & 1 \\ 4 & 4 & 4 & 0 & 0 & 4 & 0 & 0 & 0 \\ 4 & 0 & 4 & 4 & 0 & 0 & 4 & 0 & 0 \\ 4 & 0 & 0 & 4 & 4 & 0 & 0 & 4 & 0 \\ 4 & 4 & 0 & 0 & 4 & 0 & 0 & 0 & 4 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{v} \\ \mathbf{e}_1 \\ \mathbf{e}_2 \\ \mathbf{e}_3 \\ \mathbf{e}_4 \\ \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \\ \mathbf{v}_4 \end{bmatrix}$$

- A vertex at any level converges to the following:

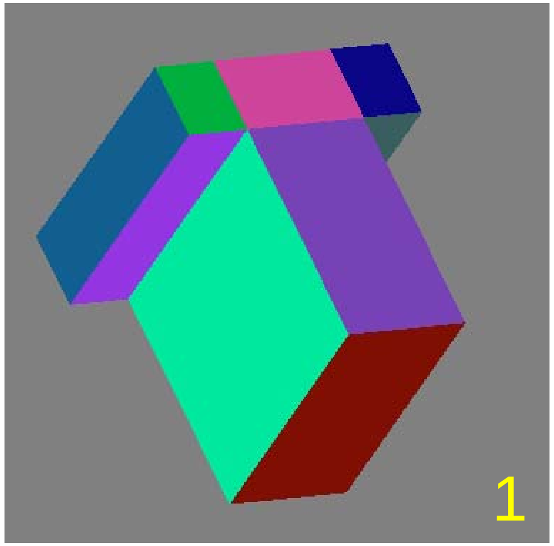
$$\mathbf{v}_\infty = \frac{n^2\mathbf{v} + 4\sum_{j=1}^{4}\mathbf{e}_j + \sum_{j=1}^{4}\mathbf{f}_j}{n(n+5)}$$

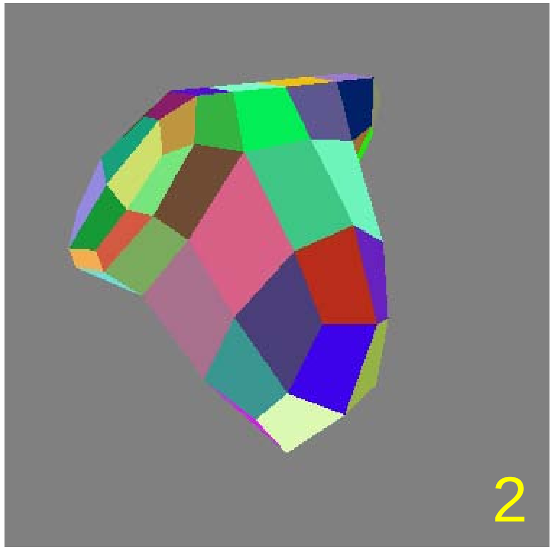
- The limit surface is a B-spline surface of degree (3,3).

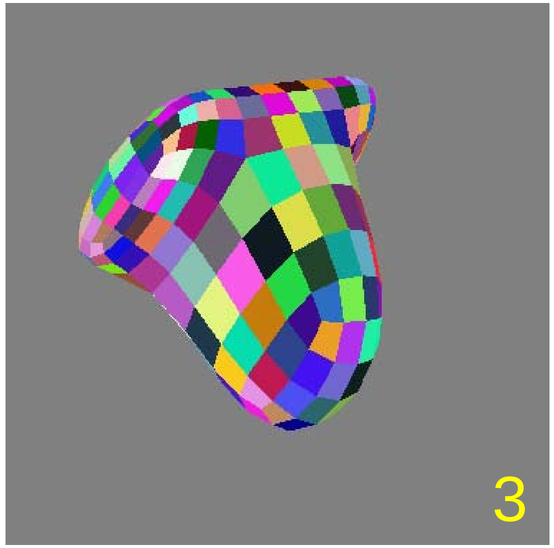
Catmull-Clark Algorithm: 8/10

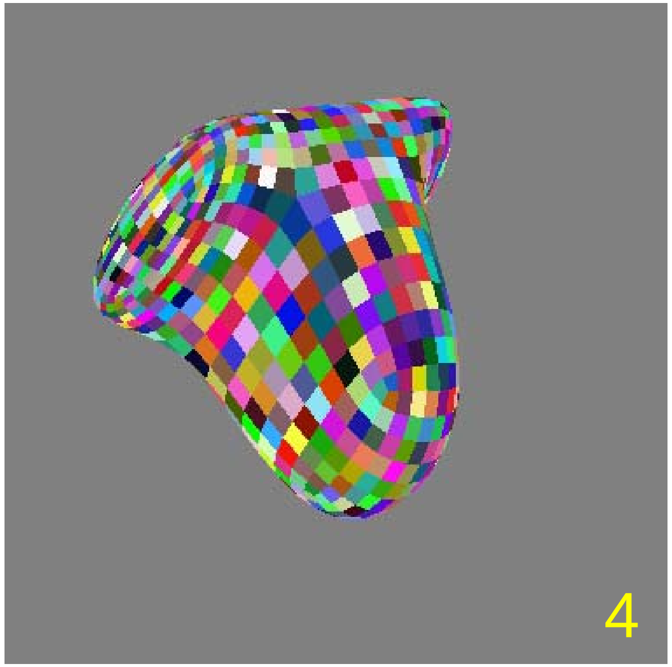


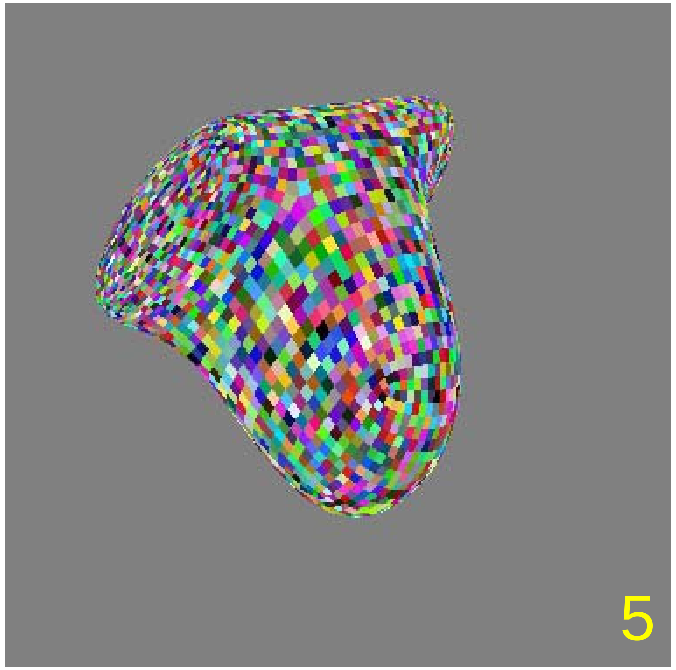
Catmull-Clark Algorithm: 9/10



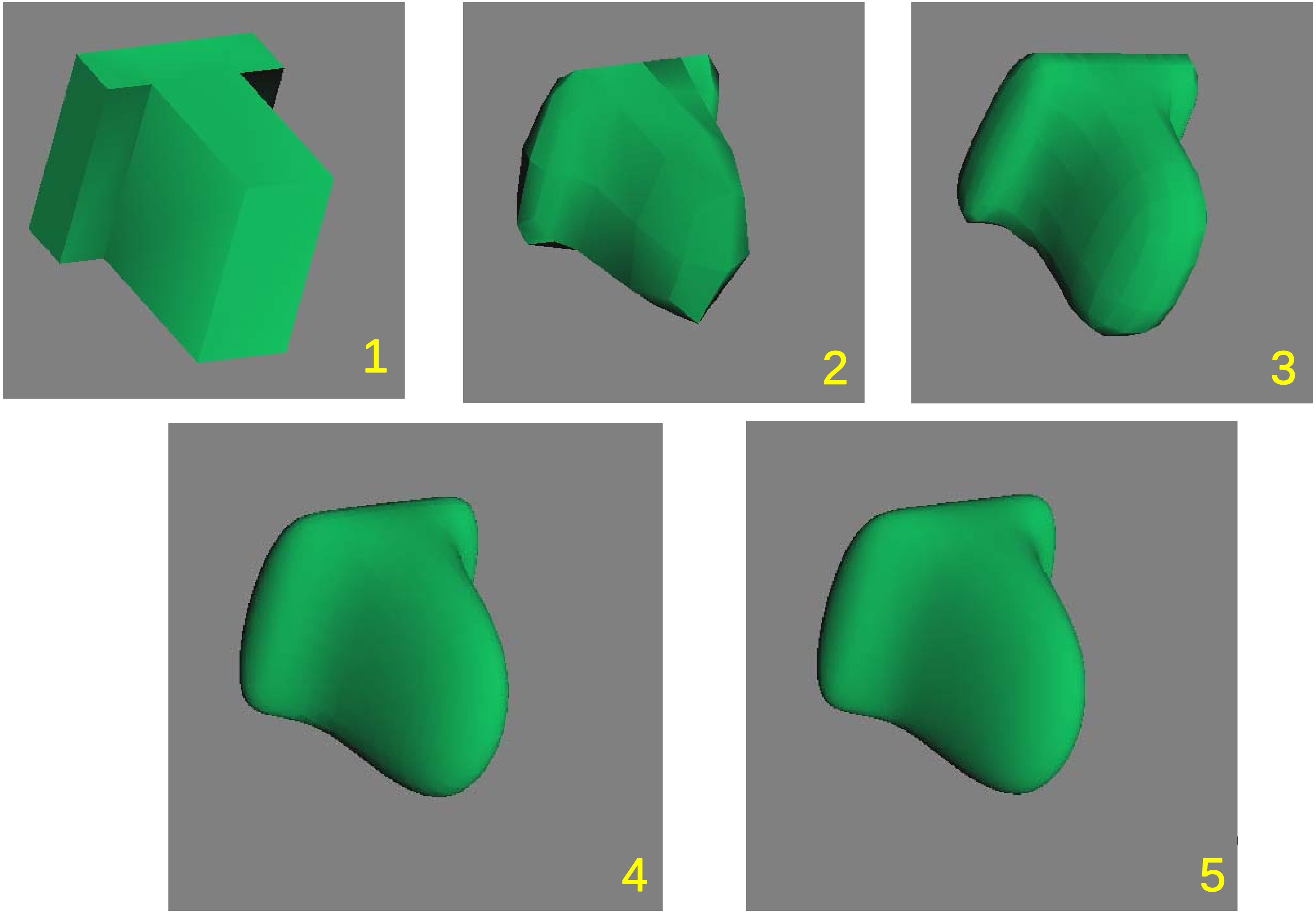








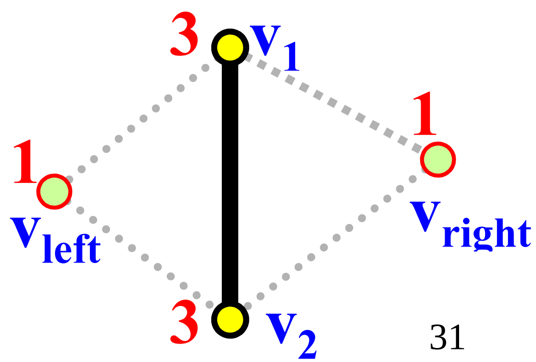
Catmull-Clark Algorithm: 10/10



Loop's Algorithm: 1/6

- Loop's (*i.e.*, Charles Loop's) algorithm only works for triangle meshes.
- Loop's algorithm computes a new edge point for each edge and a new vertex for each vertex.
- Let $\mathbf{v}_1\mathbf{v}_2$ be an edge and the other vertices of the incident triangles be $\mathbf{v}_{left}$ and $\mathbf{v}_{right}$. The new edge point $\mathbf{e}$ is computed as follows.

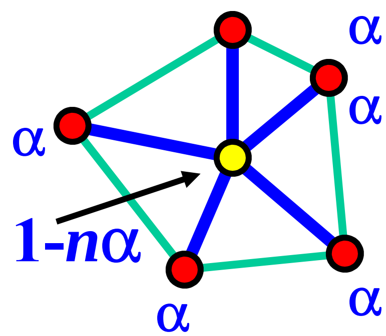
$$\mathbf{e} = \frac{3}{8}(\mathbf{v}_1 + \mathbf{v}_2) + \frac{1}{8}(\mathbf{v}_{\text{left}} + \mathbf{v}_{\text{right}})$$



Loop's Algorithm: 2/6

- **For each vertex v, its new vertex point v' is computed below, where $\mathbf{v}_1$, $\mathbf{v}_2$, …, $\mathbf{v}_n$ are adjacent vertices**

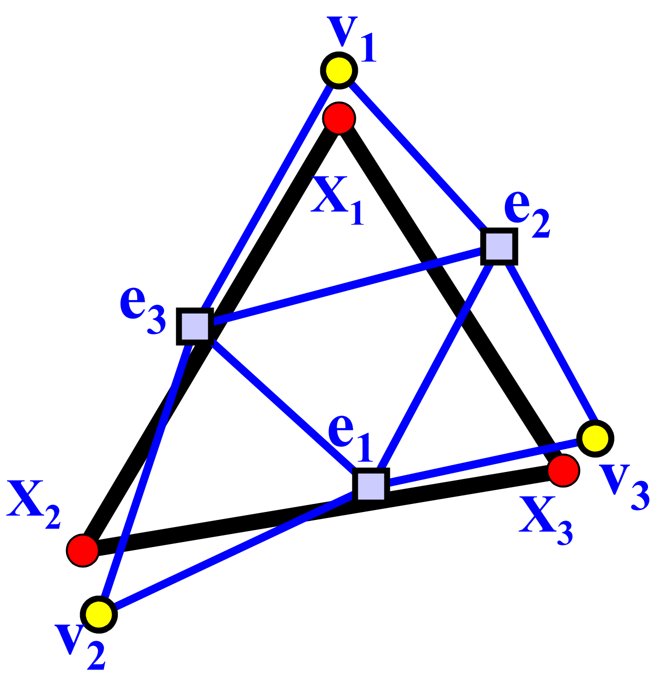
$$\mathbf{v}' = (1 - n\alpha)\mathbf{v} + \alpha \sum_{j=1}^{n} \mathbf{v}_j$$



where α is

$$\alpha = \begin{cases} \dfrac{3}{16} & n=3 \\ \dfrac{1}{n}\left[\dfrac{5}{8} - \left(\dfrac{3}{8} + \dfrac{1}{4}\cos\dfrac{2\pi}{n}\right)^2\right] & n>3 \end{cases}$$

Loop's Algorithm: 3/6



- 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 new triangles are $v_1e_2e_3$, $v_2e_3e_1$, $v_3e_1e_2$ and $e_1e_2e_3$. This is a 1-to-4 scheme.
- This algorithm was developed by Charles Loop in 1987.

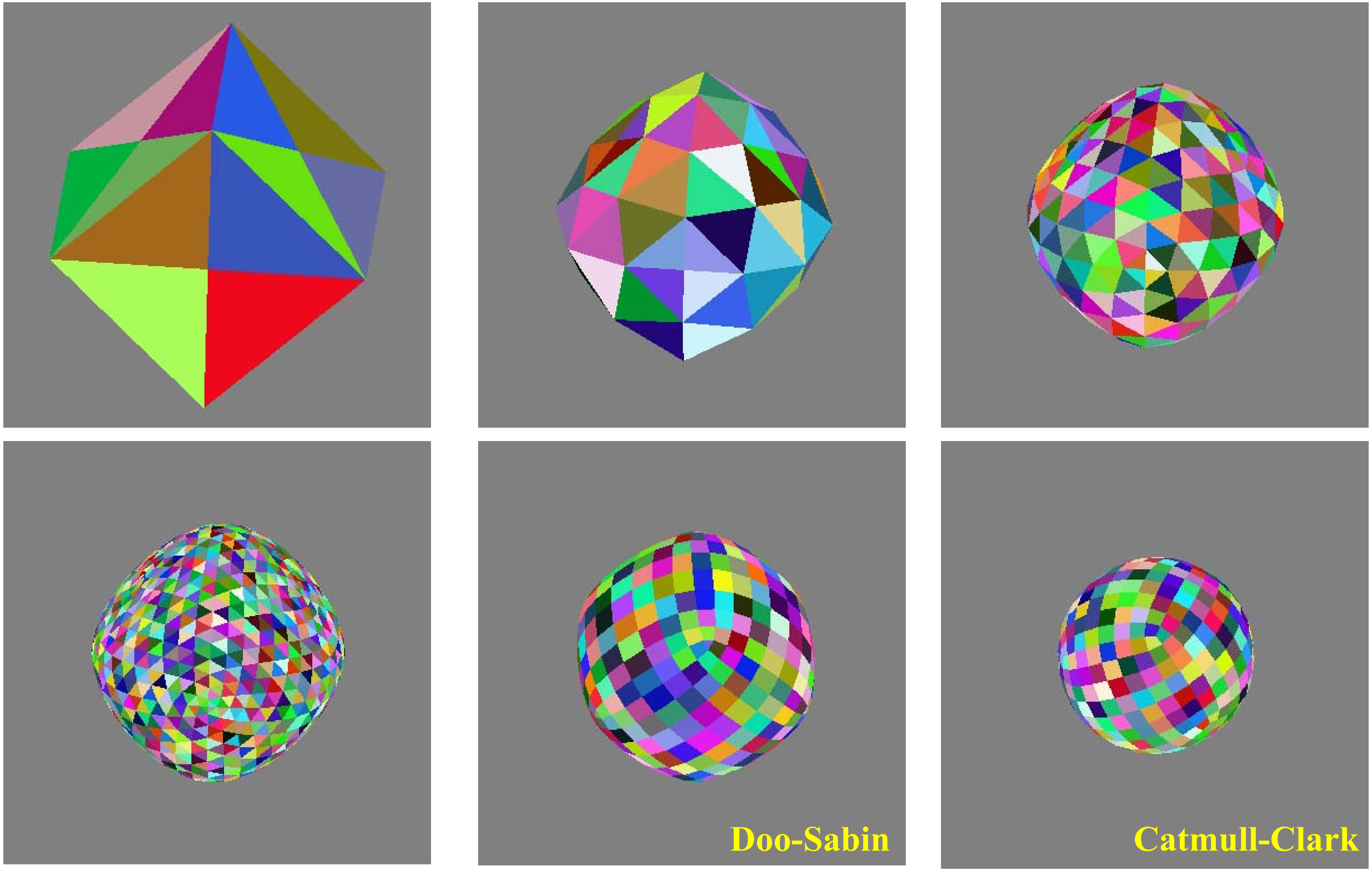
Loop's Algorithm: 4/6

- **Pick a vertex in the original or an intermediate mesh. If this vertex has n adjacent vertices $\mathbf{v}_1$, $\mathbf{v}_2$, …, $\mathbf{v}_n$, it converges to $\mathbf{v}_\infty$:**

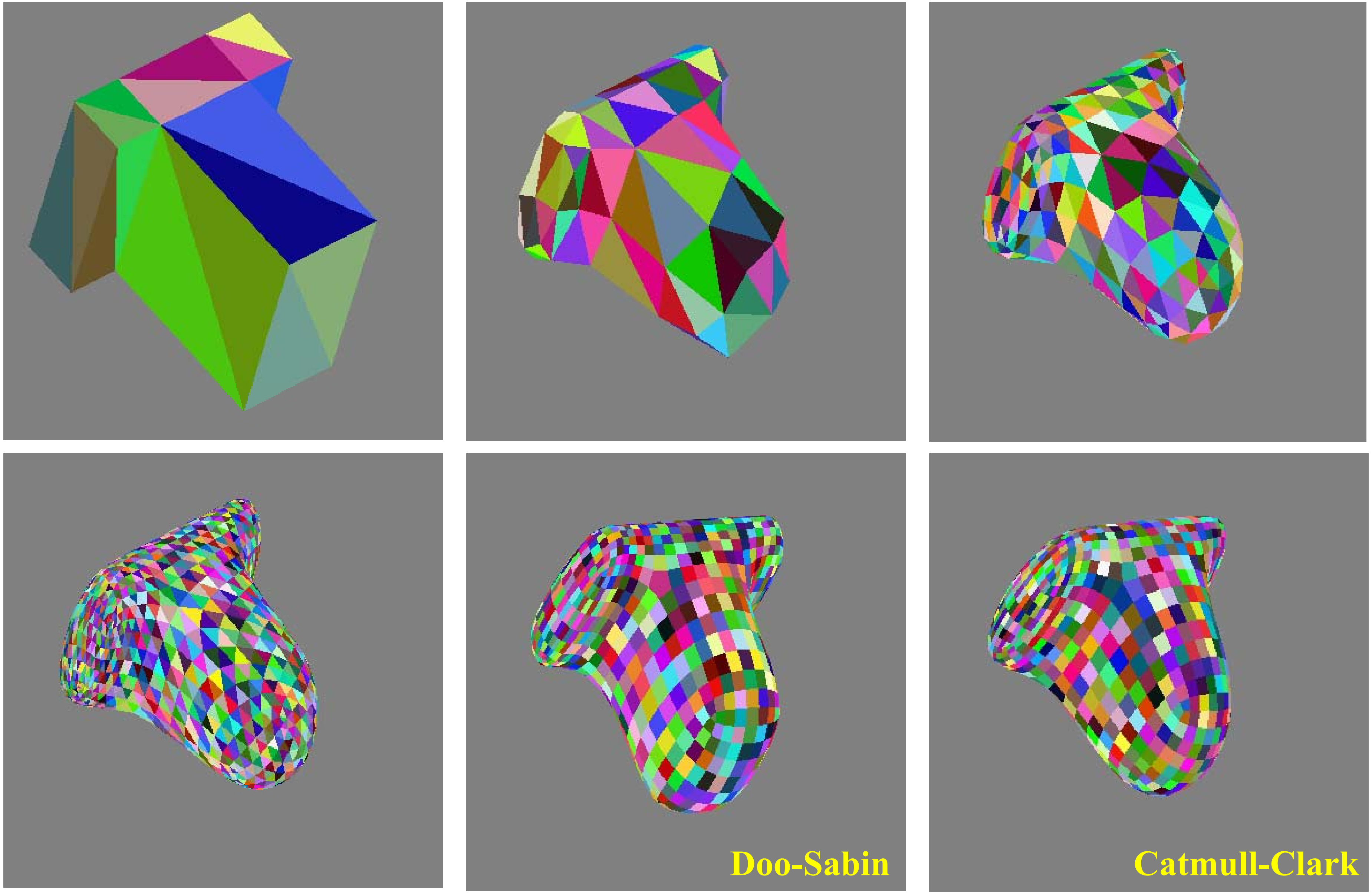
$$\mathbf{v}_\infty = \frac{3+8(n-1)\alpha}{3+8n\alpha} + \frac{8\alpha}{3+8n\alpha}\sum_{j=1}^{n}\mathbf{v}_j$$

- **If all vertices have valency 6, the limit surface is a collection of C^2 Bézier triangles.**
- **However, only a torus can be formed with all valency 6 vertices. Vertices with different valencies converge to extraordinary vertices where the surface is only G^1.**

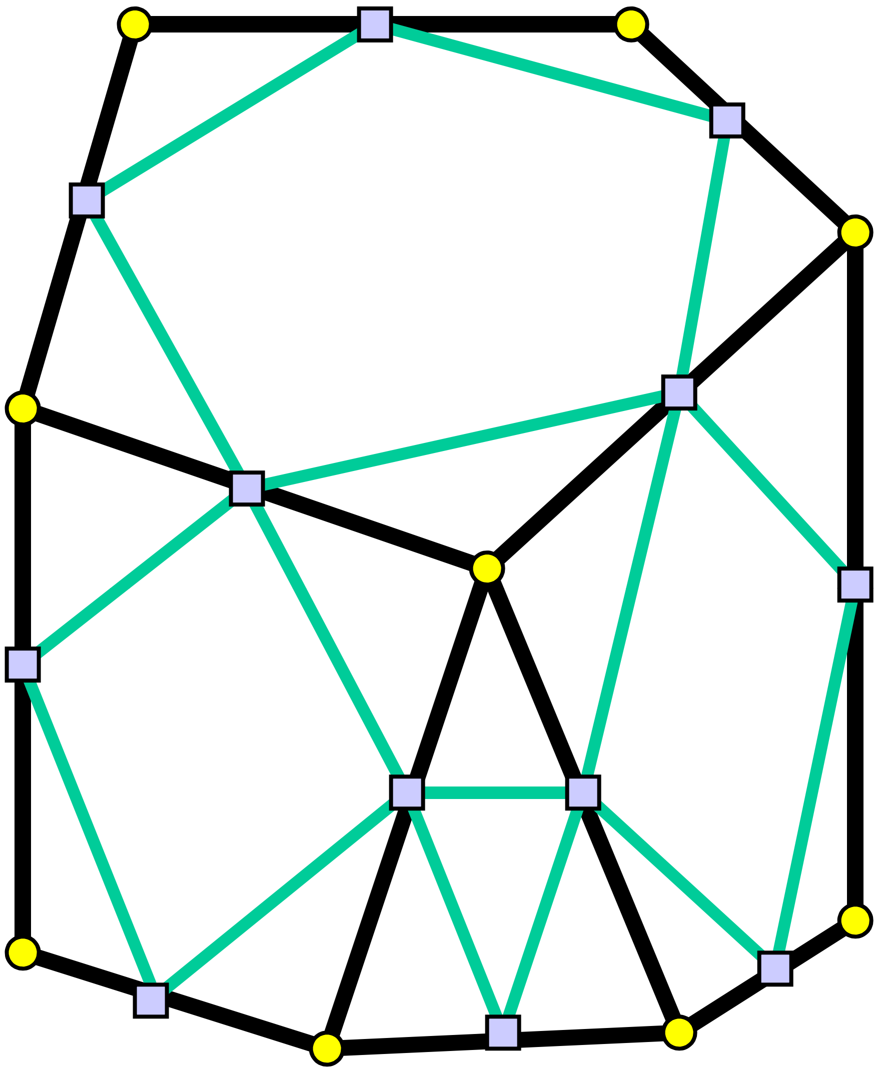
Loop’s Algorithm: 5/6



Loop's Algorithm: 6/6



Peters-Reif Algorithm: 1/4



- This is an extremely simple algorithm.
  - Compute the midpoint of each edge
  - For each face, create a face by connecting the midpoints of it edges
- There are *two* types of faces: faces inscribed to the existing ones and faces whose vertices are the midpoints of edges that are incident to a common vertex.

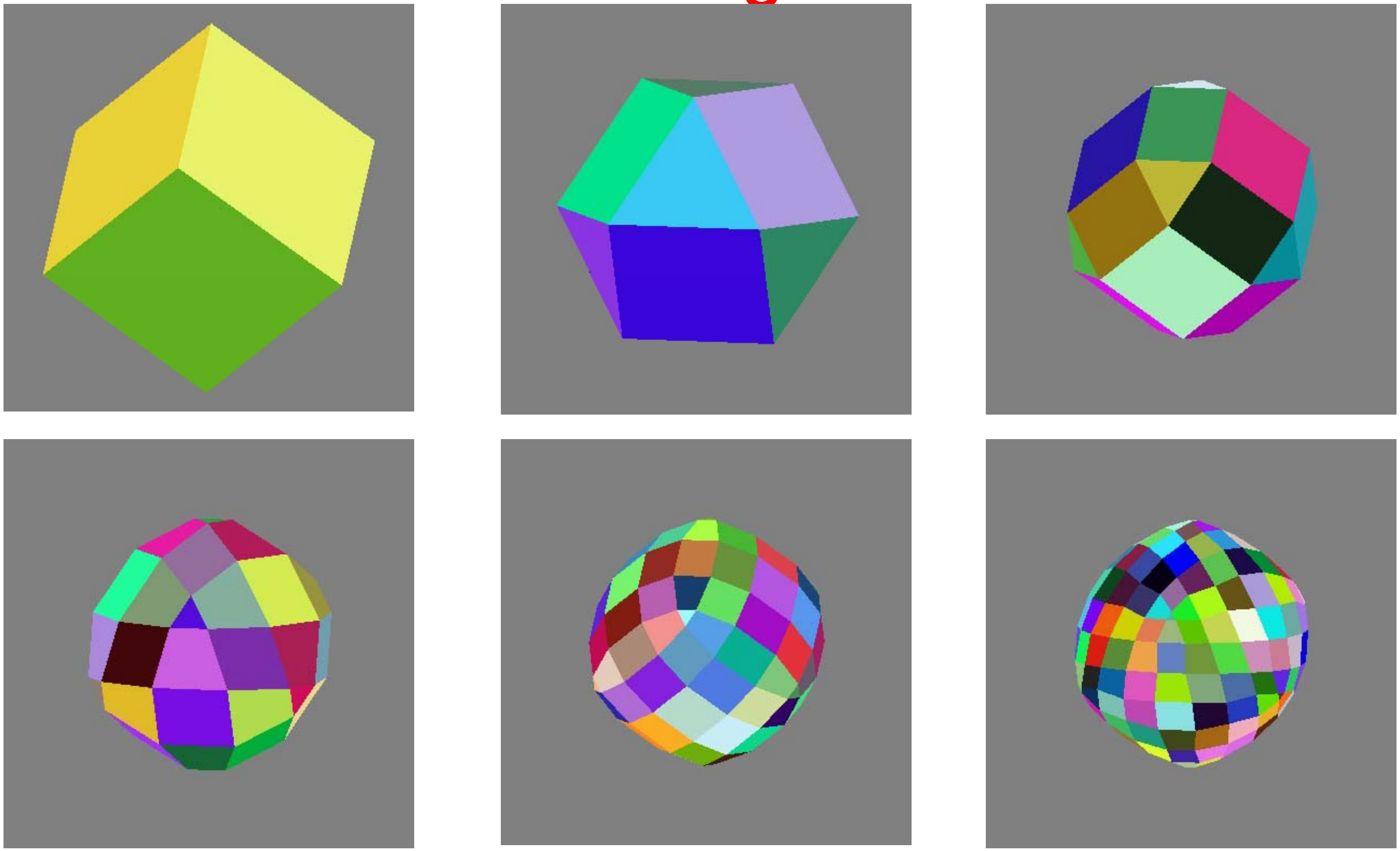
Peters-Reif Algorithm: 2/4

- **The original and new vertices has a relationship as follows:**

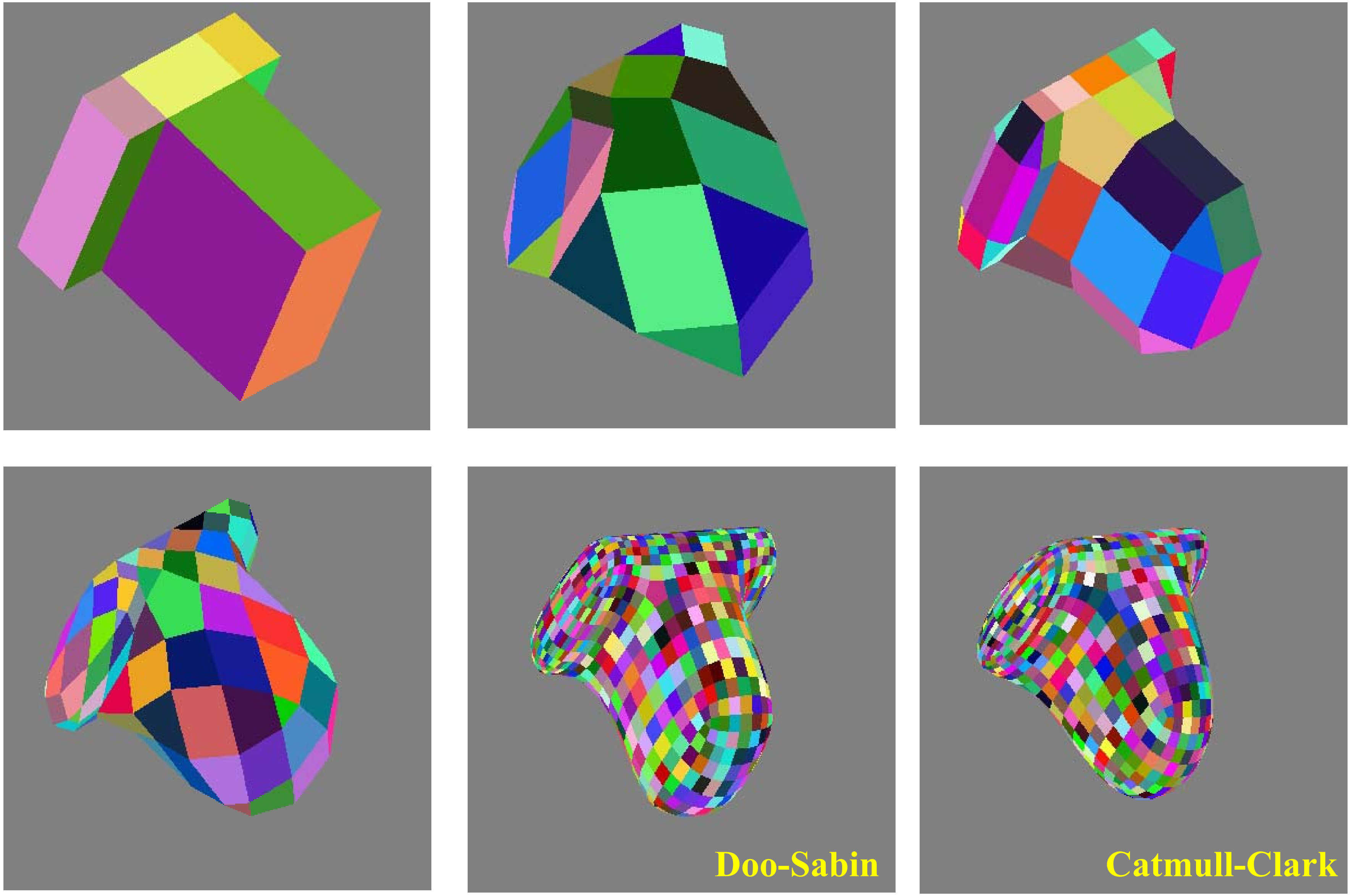
$$\begin{bmatrix} \mathbf{v}'_1 \\ \mathbf{v}'_2 \\ \vdots \\ \mathbf{v}'_{n-1} \\ \mathbf{v}'_n \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \cdots & \cdots & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & \cdots & 0 & \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_{n-1} \\ \mathbf{v}_n \end{bmatrix}$$

- **The limit of this process consists of a set of regular planar polygons that are the tangent planes of the limit surface, which is G^1.**
- **Peters-Reif algorithm was developed by J. Peters and U. Reif in 1998.**

Peters-Reif Algorithm: 3/4



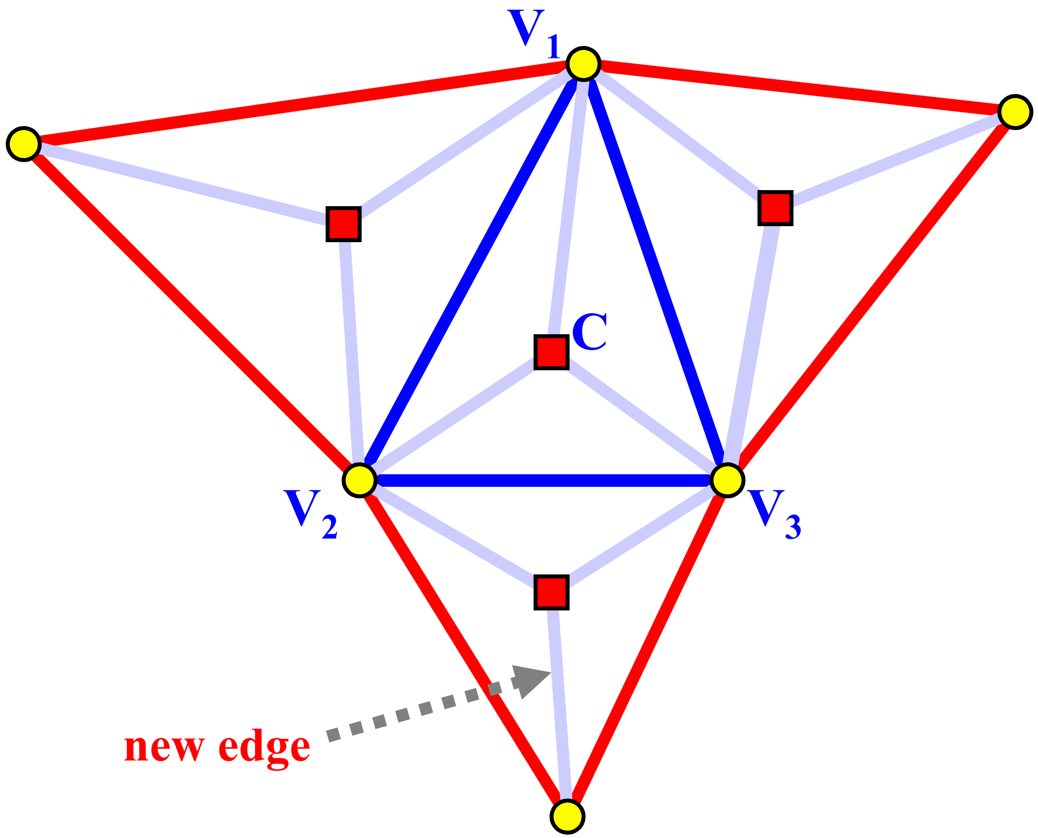
Peters-Reif Algorithm: 4/4



√3-Subdivision of Kobbelt: 1/8

- **This algorithm was developed by Leif Kobbelt in 2000, and only works for triangle meshes.**
- **This simple algorithm consists of three steps:**
  1) **Dividing each triangle at the center into 3 more triangles**
  2) **Perturb the vertices of each triangle**
  3) **"Flip" the edges of the perturbed triangle (see next slide).**

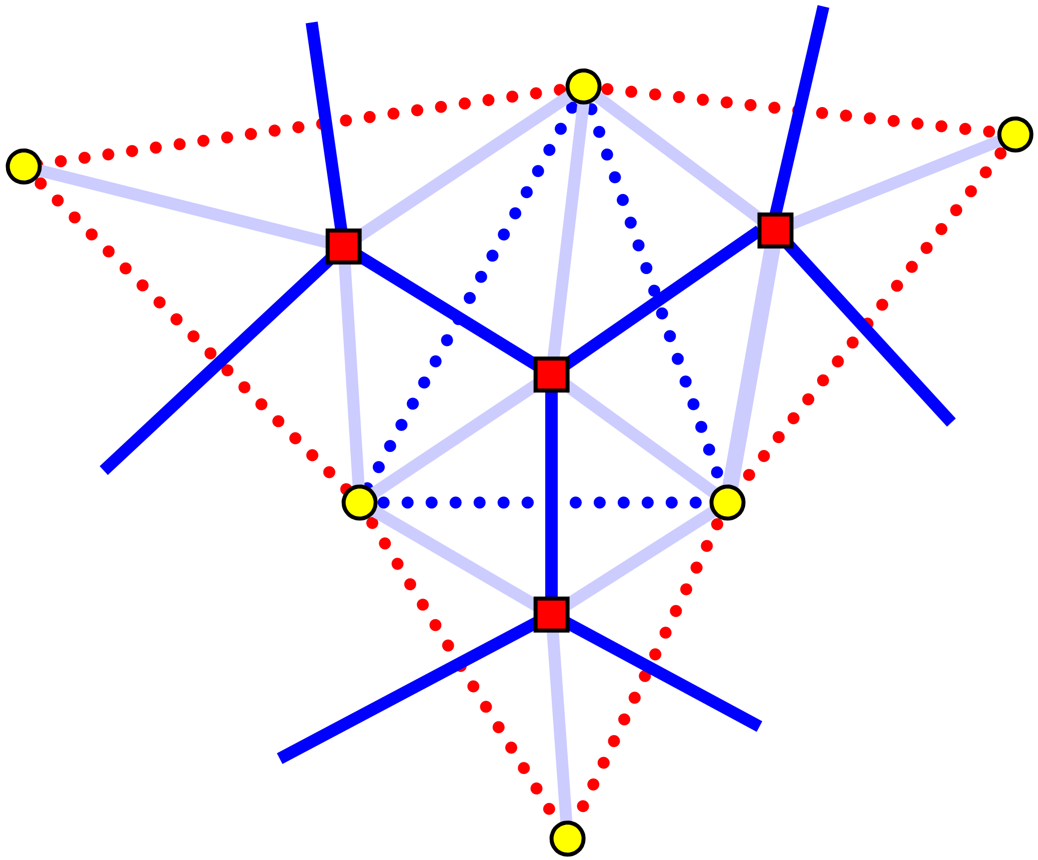
√3-Subdivision of Kobbelt: 2/8
Step 1: Subdividing



- For each triangle, compute its center: $C = (V_1+V_2+V_3)/3$
- Connect the center to each vertex to create 3 triangles.
- This is a 1-to-3 scheme!

√3-Subdivision of Kobbelt: 3/8

Step 2: Flipping Edges

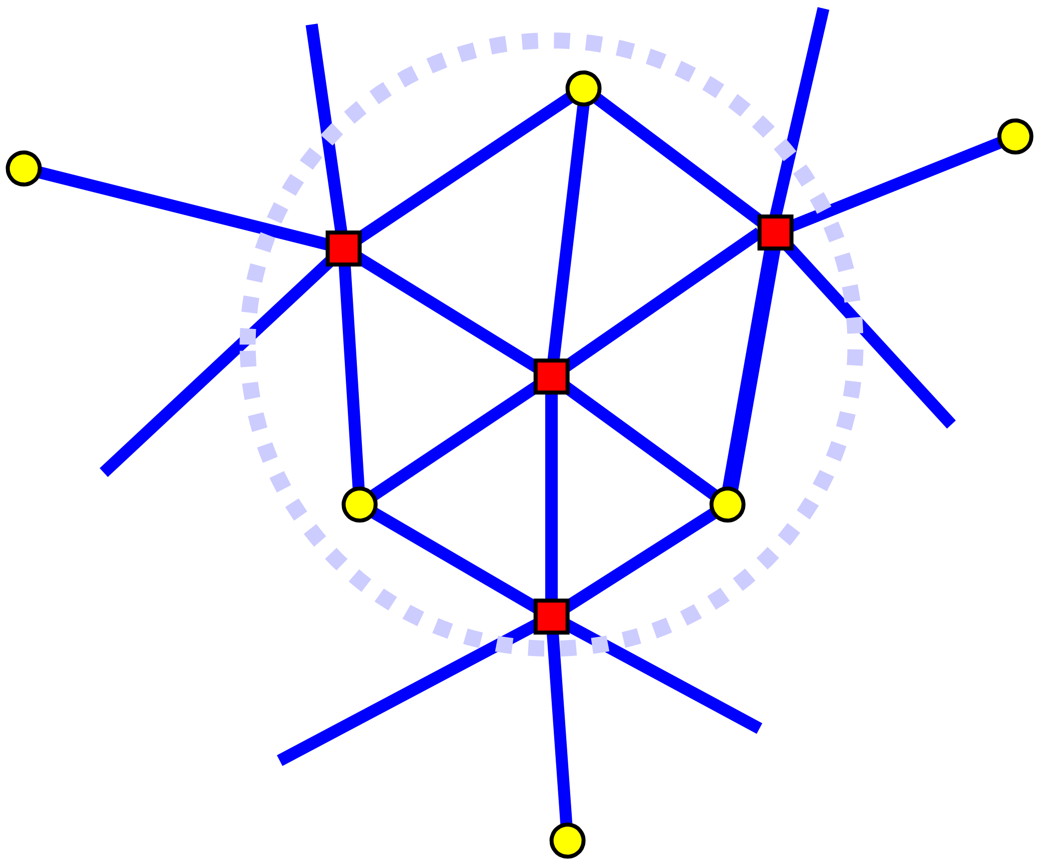


- Since each original edge is adjacent to two triangles, "flipping" an edge means removing the original edge and replacing it by the new edge joining the centers.

Dotted: original Solid: "flipped"

√3-Subdivision of Kobbelt: 4/8

Final Result



- Remove the original edges and we have a new triangle mesh!
- But, the original vertices must also be "perturbed" a little to preserve "smoothness".

√3-Subdivision of Kobbelt: 5/8
Actual Computation

- **For each triangle with vertices V_1, V_2 and V_3, compute its center C:**

$$C = \frac{1}{3}\left(V_1 + V_2 + V_3\right)$$

- **For each vertex V and its neighbors V_1, V_2, …, V_n, compute a perturbed V' as follows:**

$$V' = (1-\alpha_n)V + \frac{\alpha_n}{n}\sum_{i=1}^{n} V_i$$

where α_n is computed as follows:

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

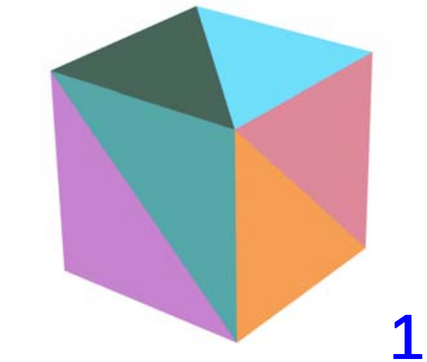
- **Replace V_i's with V'_i's and do edge flipping.**

√3-Subdivision of Kobbelt: 6/8

Important Results

- **The √3-subdivision converges!**
- **The limit surface is *C^2* everywhere except for extraordinary points.**
- **It is only *C^1* at extraordinary points (*i.e.*, vertices with valance ≠ 6).**
- **The √3-subdivision can be extended to an adaptive scheme for finer subdivision control.**

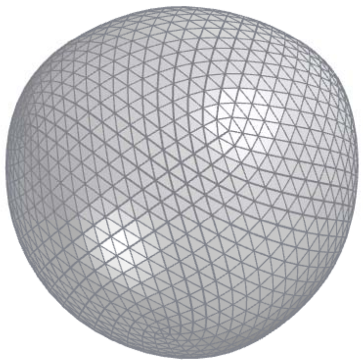
√3-Subdivision of Kobbelt: 7/8

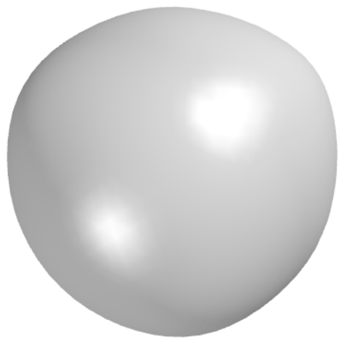

1


2

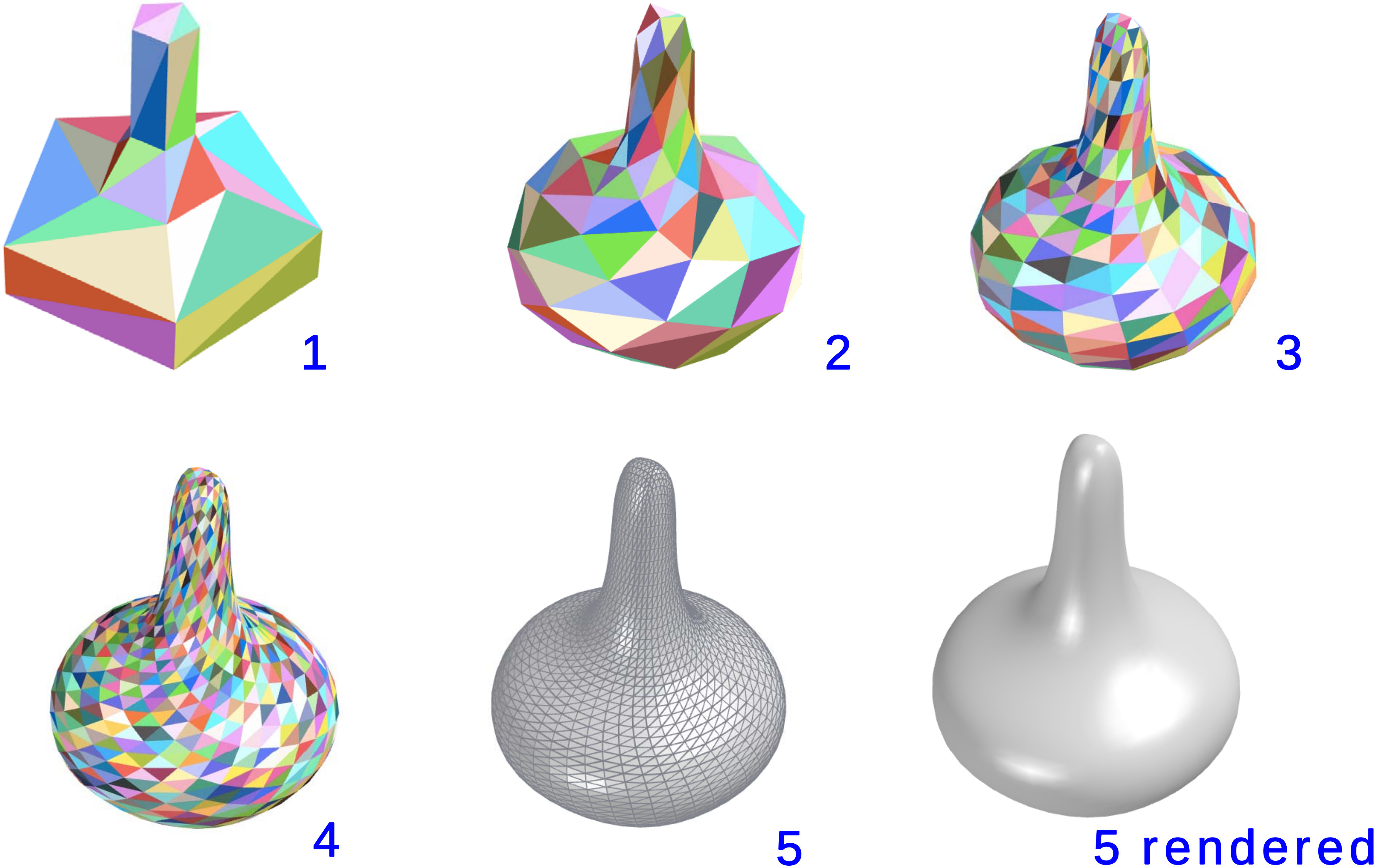

3


4


5


5 rendered

√3-Subdivision of Kobbelt: 8/8



The End