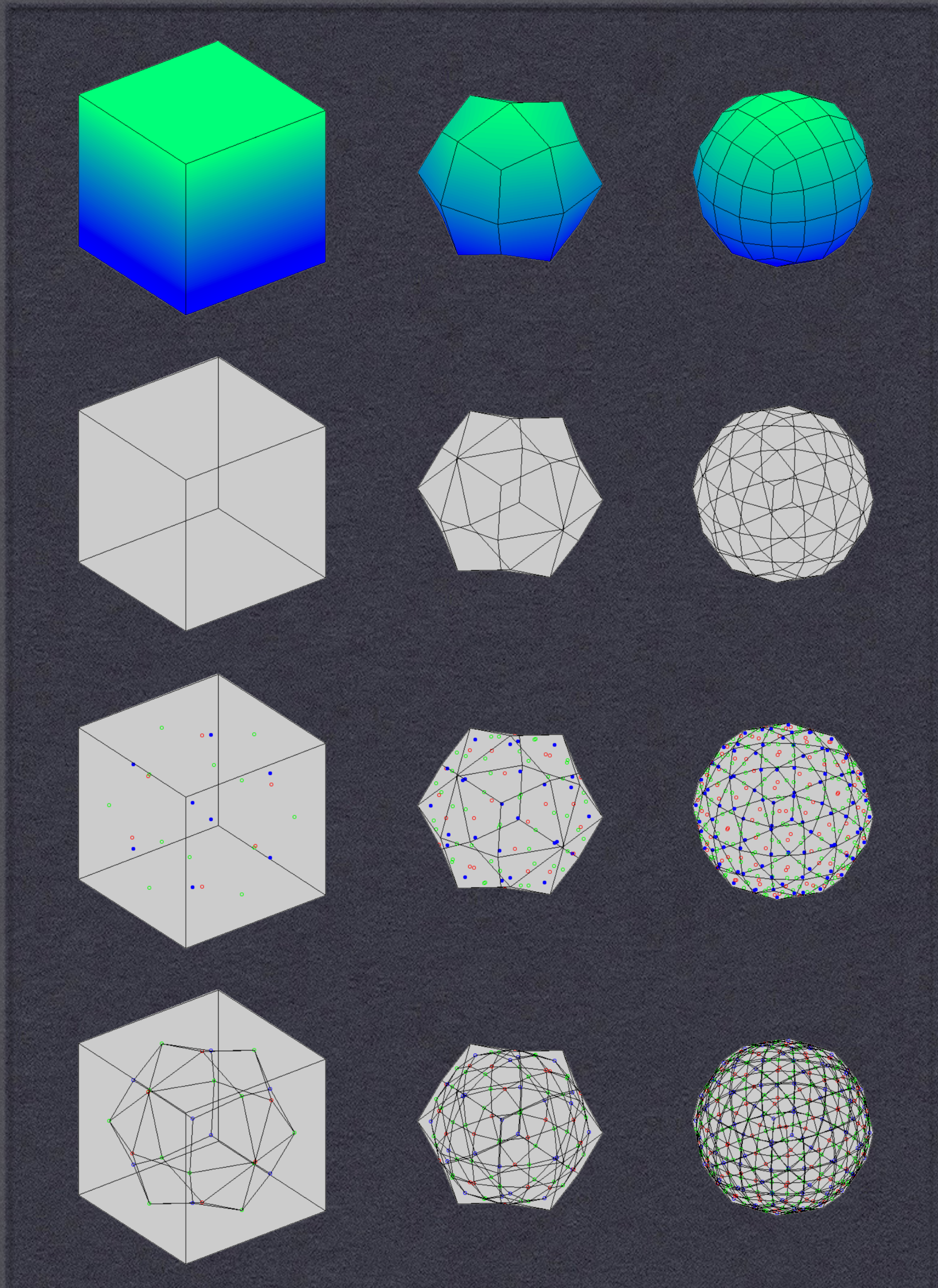
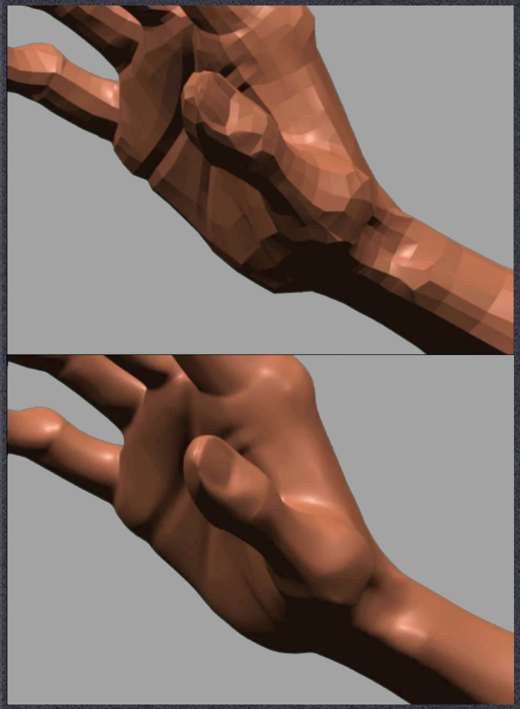
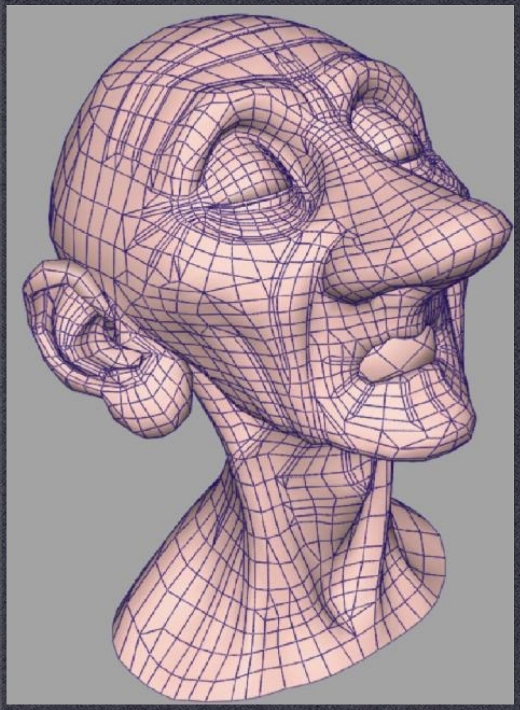
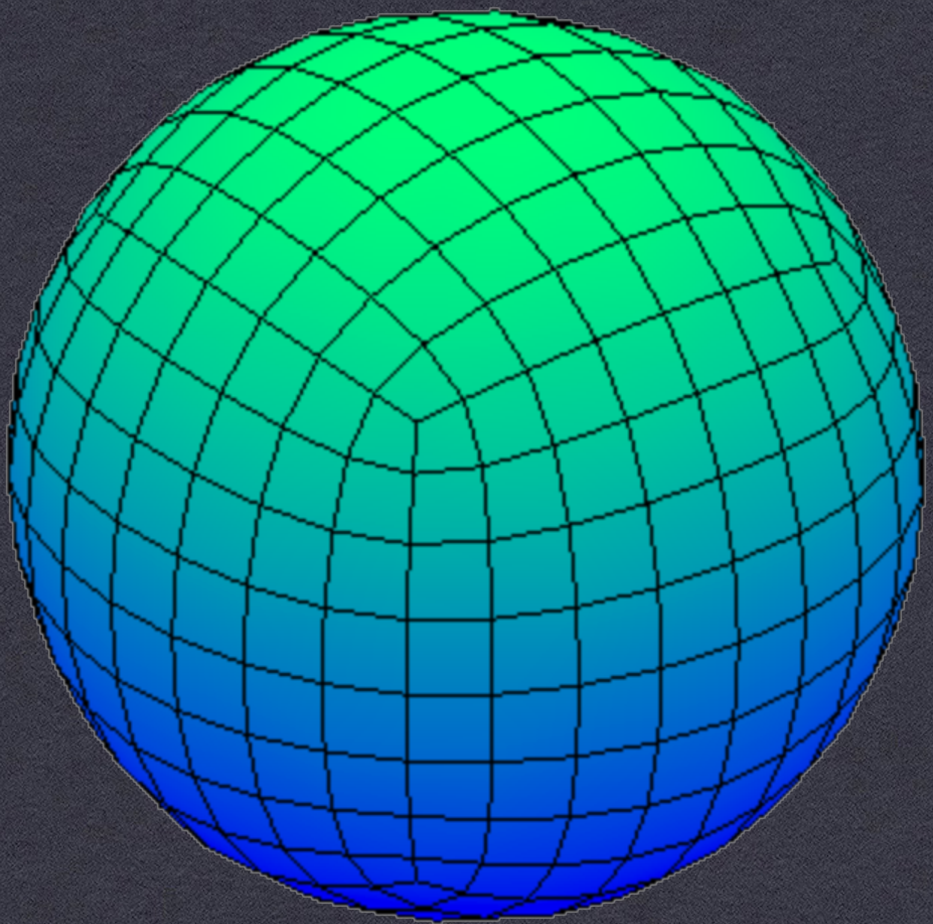








PROJECT

RECURSIVE B-SPLINE SURFACES

KAYLA BOLLINGER, PHIL WARTHER, LUKE WUKMER

DATE **12/15/15**

CLIENT **PROF. JEN-MEI CHANG**

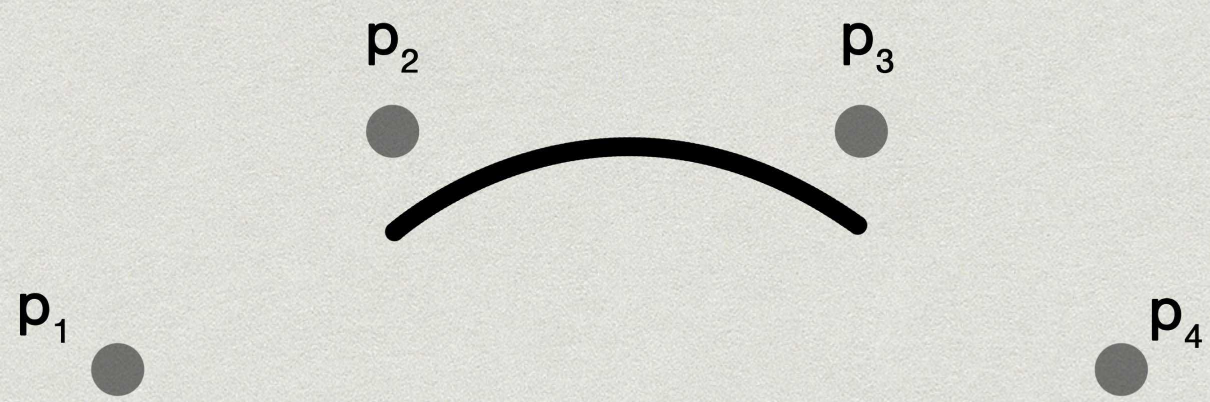
B-Spline Interpolation: 2D

For the points, $\mathbf{p} = [p_1\ p_2\ p_3\ p_4]$, the interpolating B-spline is defined

$$S(u) = \sum_{i=1}^{4} b_i(u)\mathbf{p}_i = \mathbf{u}^T\mathbf{M}\mathbf{p}$$

where $\mathbf{b}(u) = \mathbf{M}^T\mathbf{u} = \frac{1}{6}\begin{bmatrix} u^3 \\ 1+3u+3u^2-3u^3 \\ 4-6u^2+3u^3 \\ (1-u)^3 \end{bmatrix}$

$$\mathbf{u} = \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix} \qquad \mathbf{M} = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & 6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}$$



B-Spline Interpolation: 3D

For where $0 \leq u, v \leq 1$,

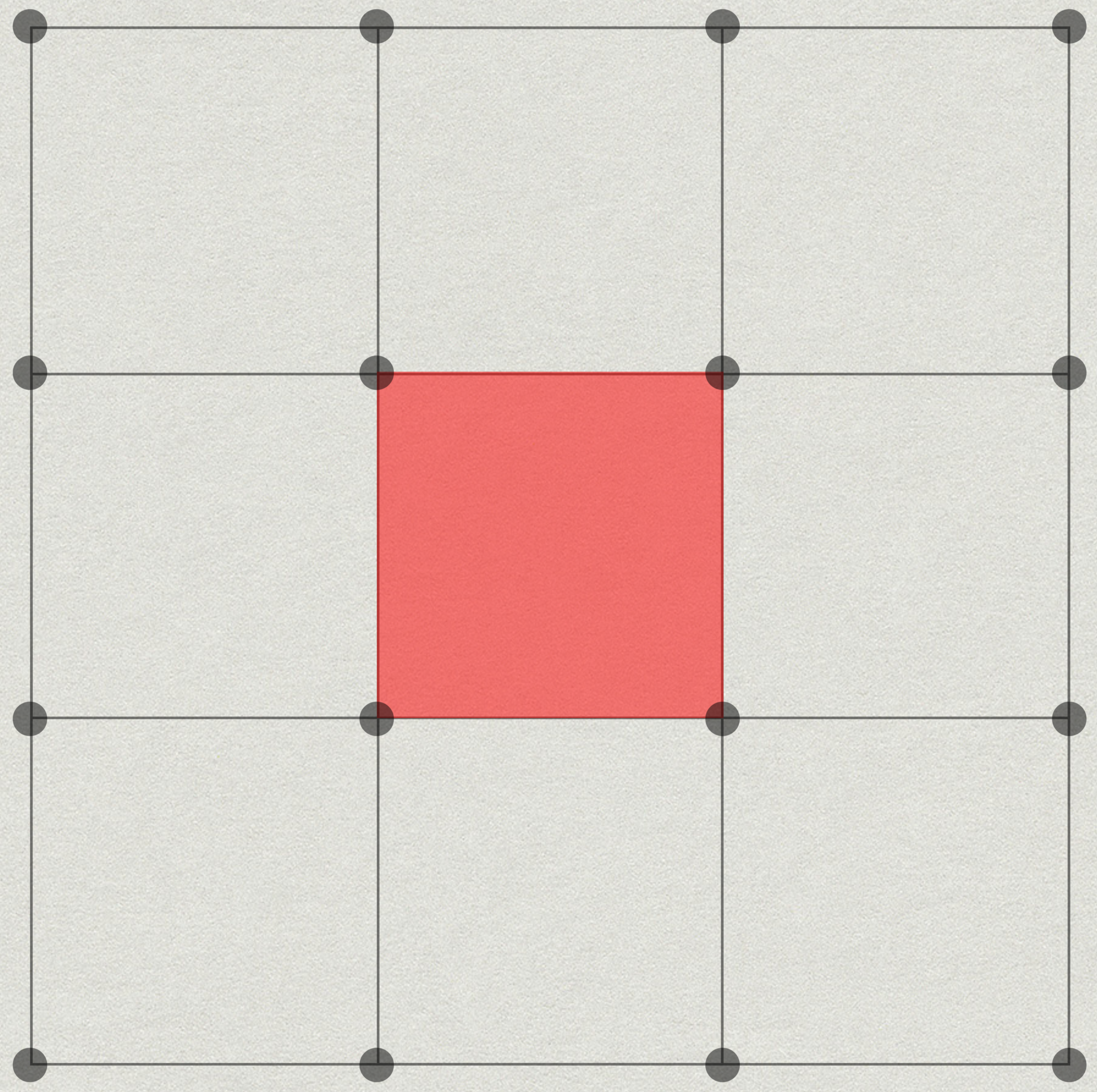
define $\mathbf{u} = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix}$ and

$\mathbf{v} = \begin{bmatrix} v^3 & v^2 & v & 1 \end{bmatrix}$.

$$S(u,v) = \sum_{i=1}^{4} \sum_{j=4}^{4} b_i(u) b_u(v) p_{ij} = \mathbf{uMPM}^T \mathbf{v}^T$$

Where a 4×4 mesh of points is defined as:

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix}$$



Recursive Method

Consider the subpatch of $\mathbf{P}$ where $0 \leq u, v \leq \frac{1}{2}$.

Let $\tilde{u} = \frac{u}{2}$ and $\tilde{v} = \frac{v}{2}$.

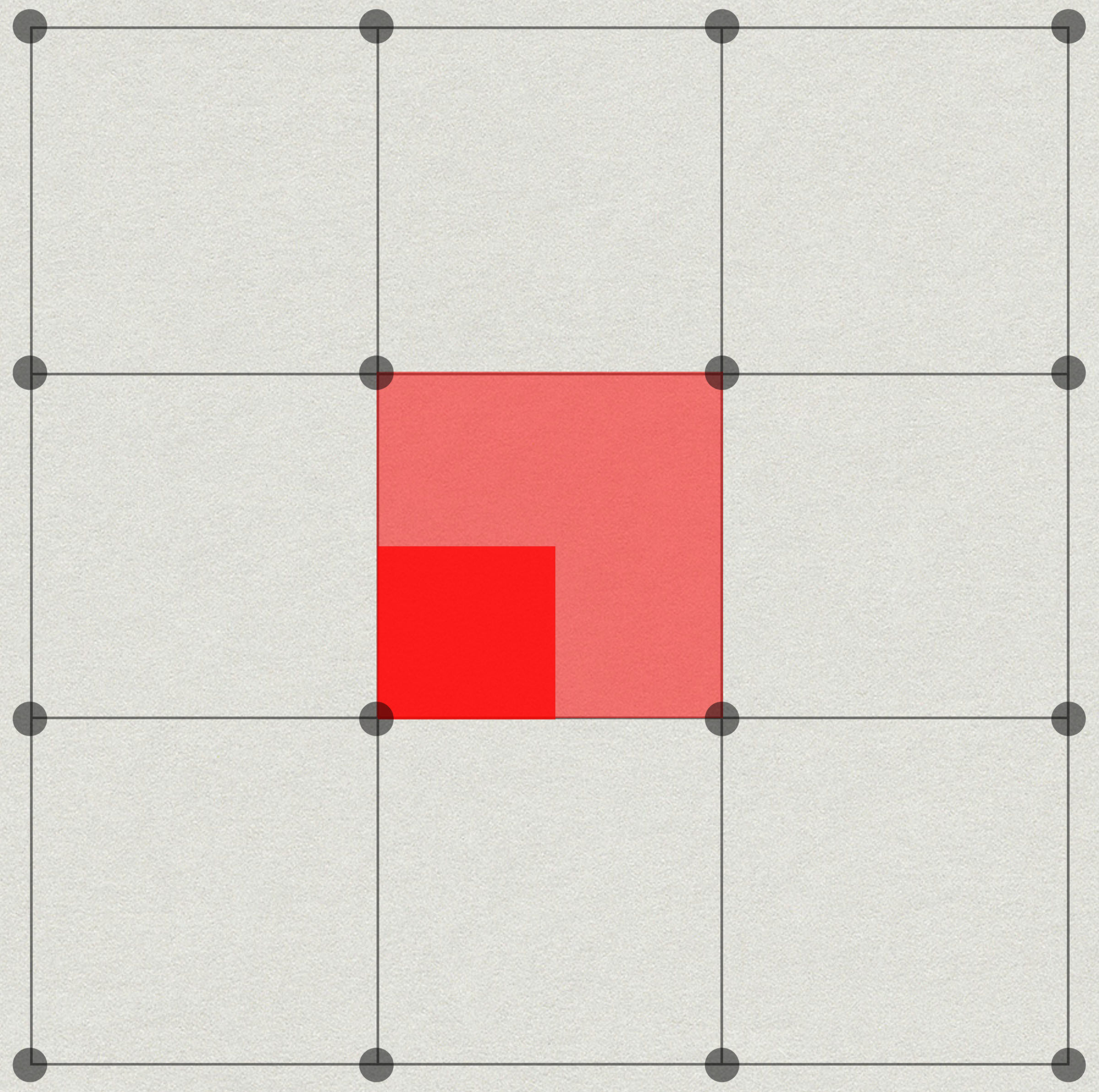
Let a matrix ψ be defined

$$\psi = \begin{bmatrix} \frac{1}{8} & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

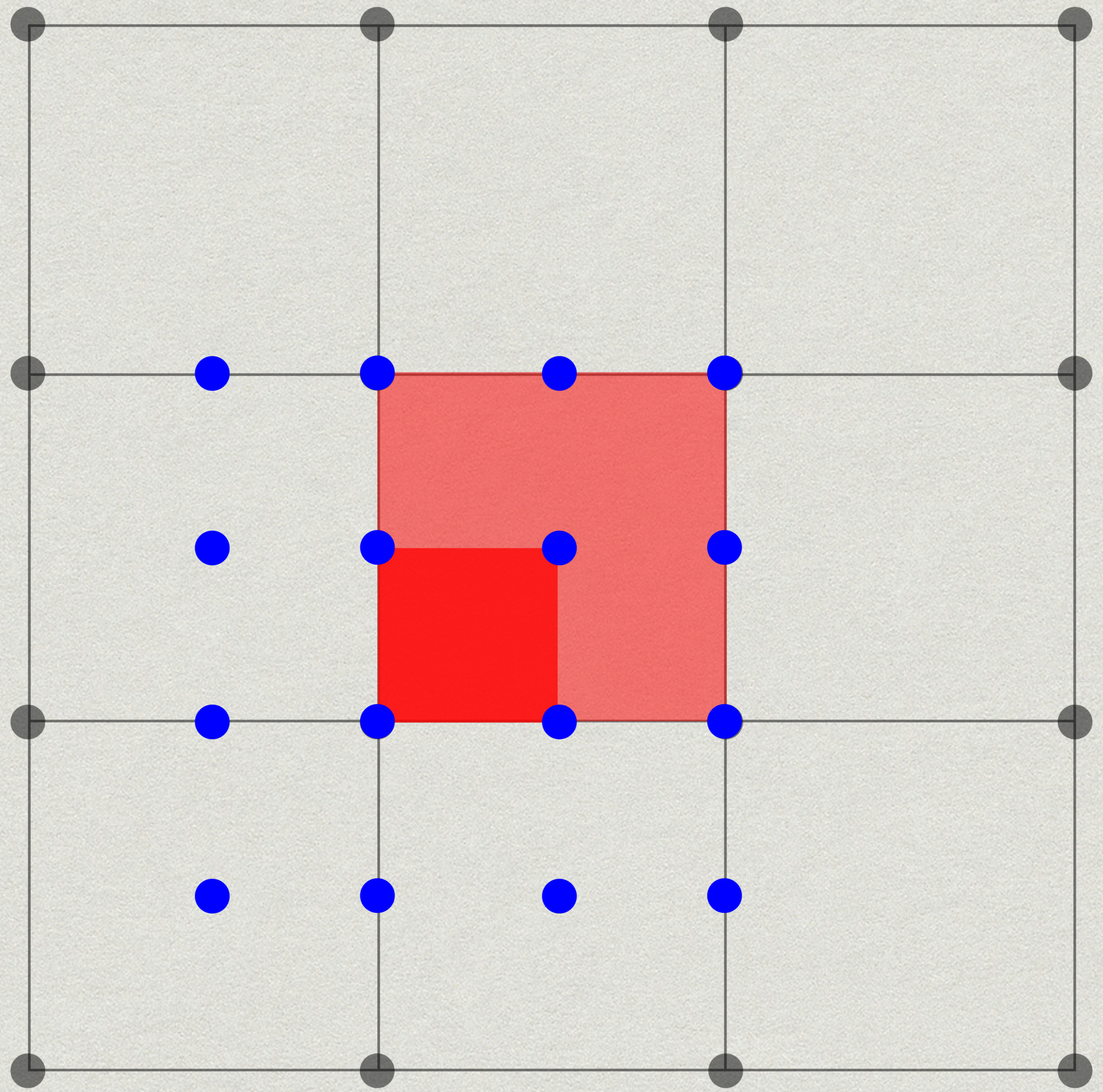
So $\mathbf{u}\Psi = \begin{bmatrix} \frac{u^3}{8} & \frac{u^2}{4} & \frac{u}{2} & 1 \end{bmatrix} = \tilde{\mathbf{u}}$ and

$$\mathbf{v}\Psi = \begin{bmatrix} \frac{v^3}{8} & \frac{v^2}{4} & \frac{v}{2} & 1 \end{bmatrix} = \tilde{\mathbf{v}} \ .$$

$$S(\tilde{u}, \tilde{v}) = \mathbf{u}\Psi\mathbf{M}\mathbf{P}\mathbf{M}^T\Psi^T\mathbf{v}^T$$



Recursive Method



There must exist a 4×4 mesh of points $\mathbf{P}_1$ that interpolates the subpatch.

$$S_1(u,v) = \mathbf{u}\mathbf{M}\mathbf{P}_1\mathbf{M}^T\mathbf{v}^T$$

Recursive Method

Since we require $S_1(u,v) = S(\tilde{u},\tilde{v})$:

$$\mathbf{M}\mathbf{P}_1\mathbf{M}^T = \Psi\mathbf{M}\mathbf{P}\mathbf{M}^T\Psi$$

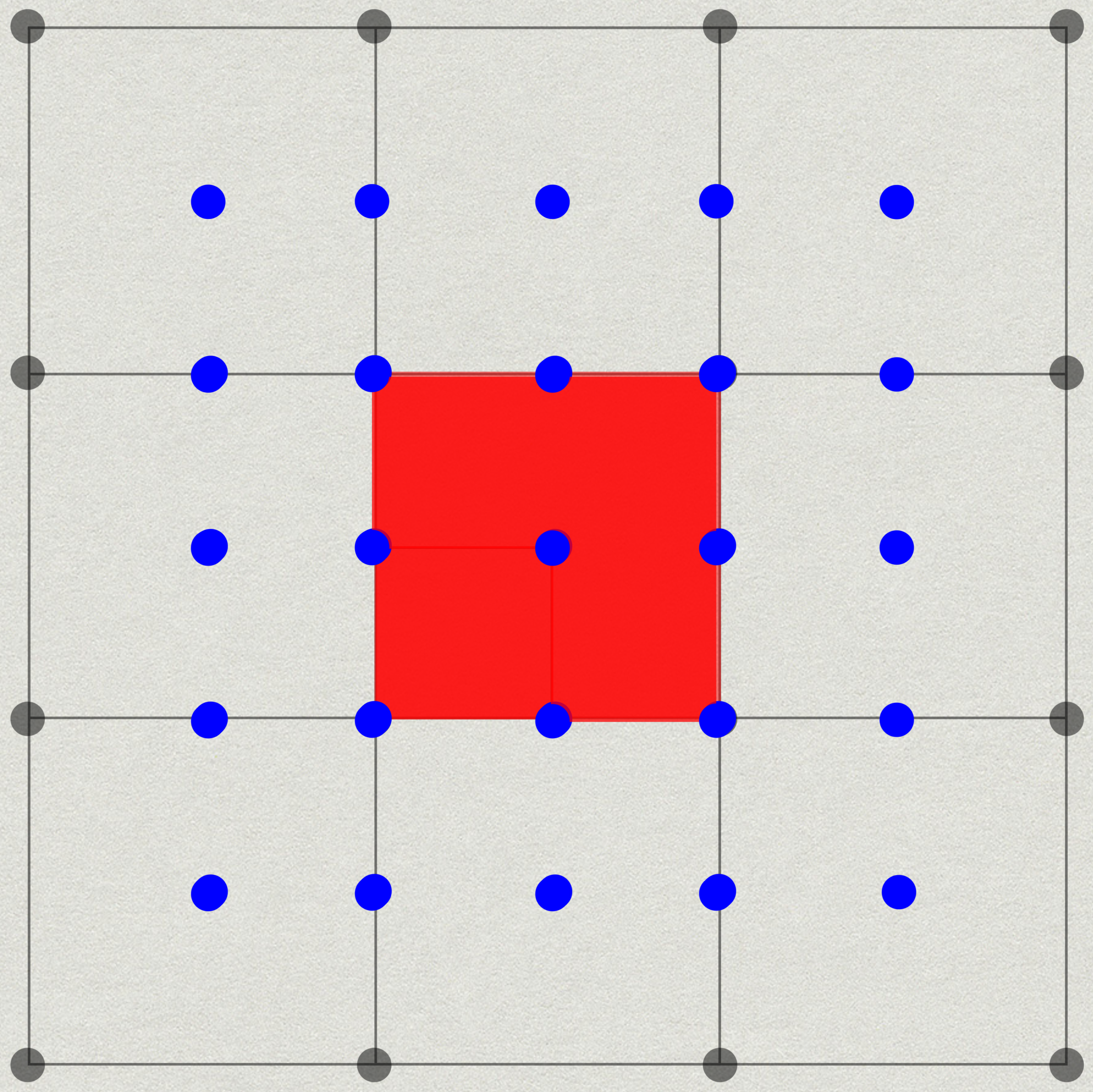
$$\mathbf{P}_1 = \mathbf{M}^{-1}\Psi\mathbf{M}\mathbf{P}\mathbf{M}^T\Psi\mathbf{M}^{-T}$$

$$\text{Let } \mathbf{H} = \mathbf{M}^{-1}\Psi\mathbf{M}\,.$$

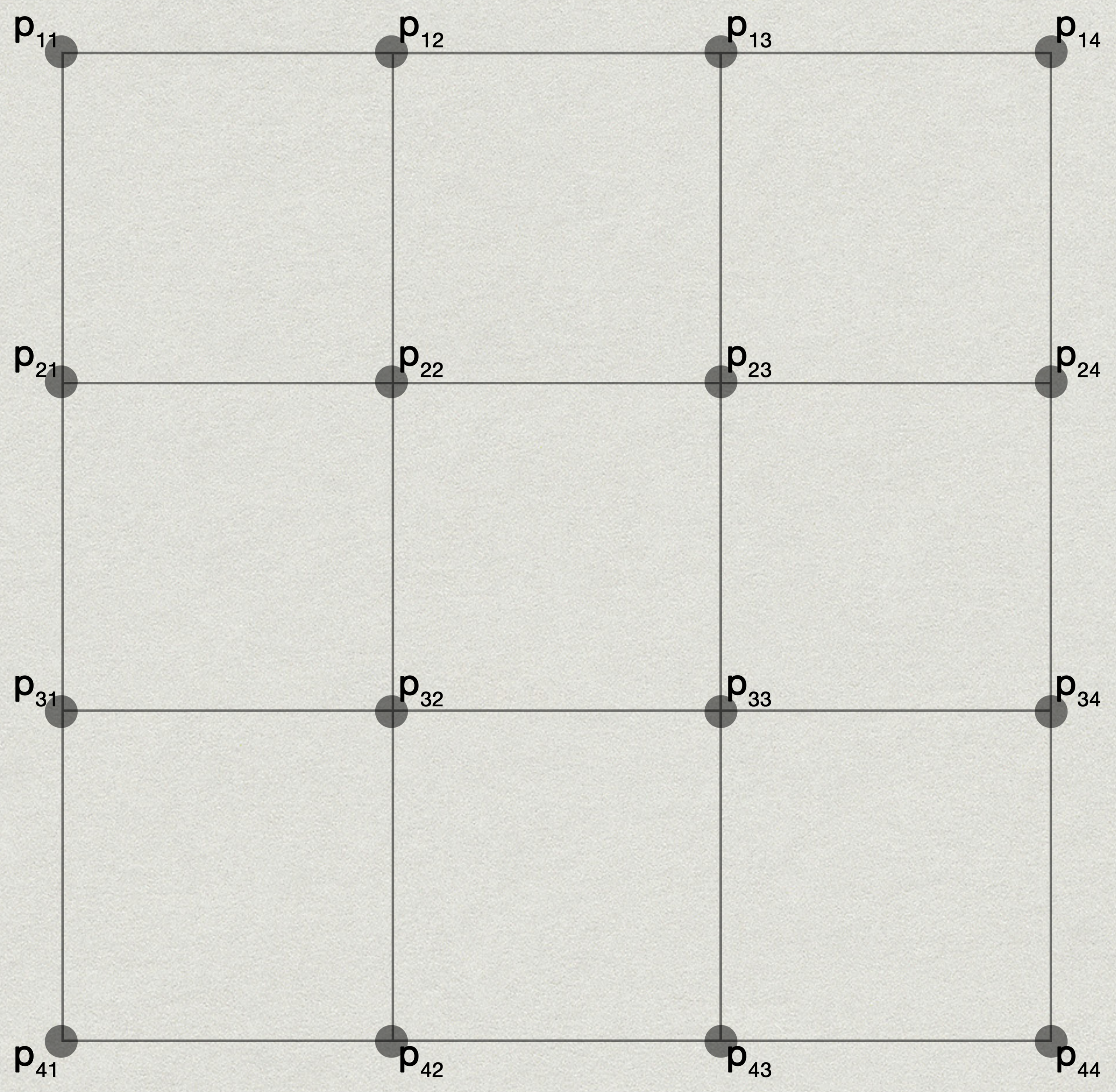
This gives us $\mathbf{P}_1 = \mathbf{H}^{-1}\mathbf{P}\mathbf{H}^T$ where:

$$\mathbf{H} = \frac{1}{8}\begin{bmatrix} 4 & 4 & 0 & 0 \\ 1 & 6 & 1 & 0 \\ 0 & 4 & 4 & 0 \\ 0 & 1 & 6 & 1 \end{bmatrix}$$

Recursive Method



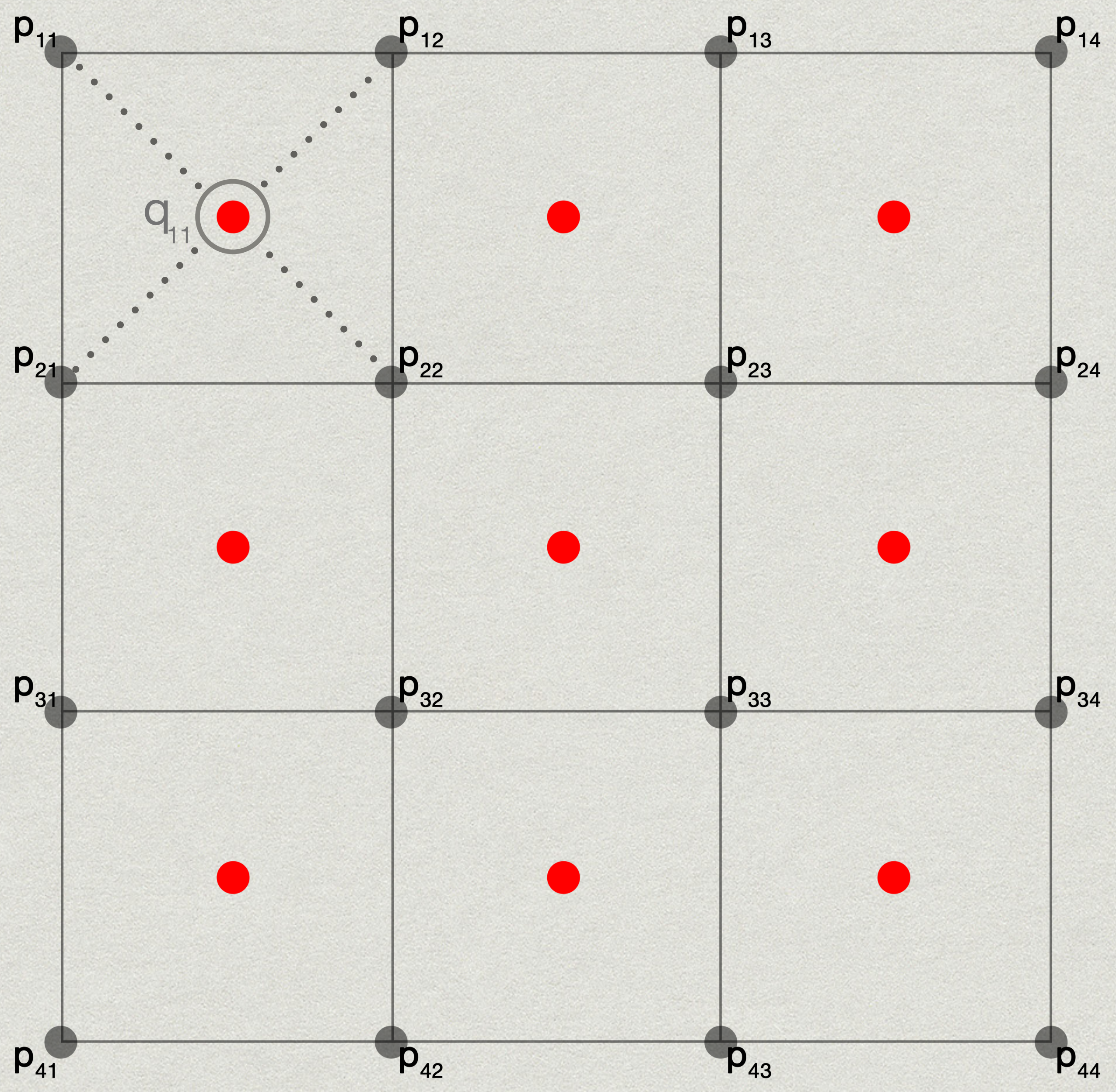
Steps of Subdivision



Face Points

- face point = average of vertices that define the face

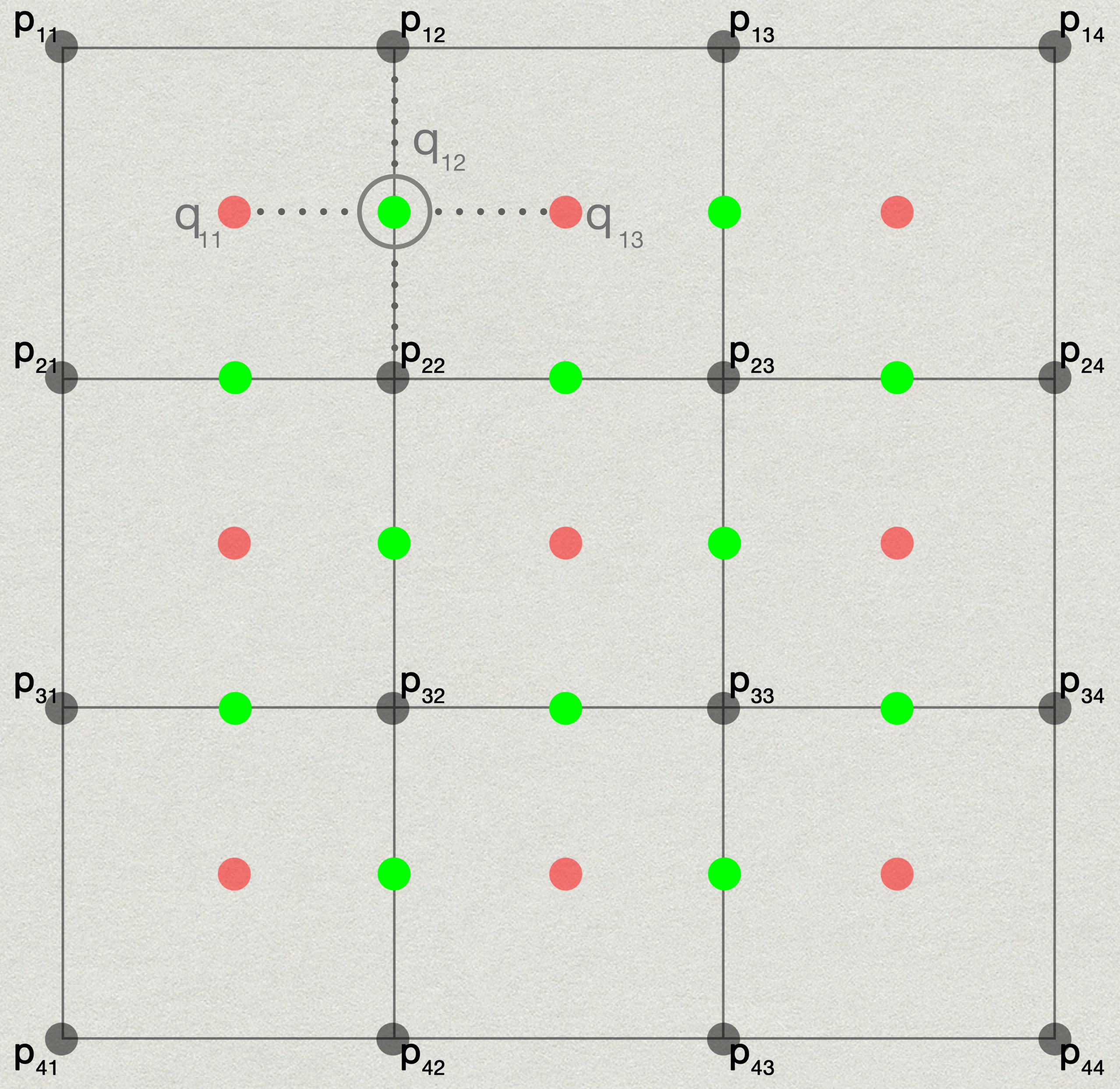
$$q_{11} = \frac{p_{11} + p_{12} + p_{21} + p_{22}}{4}$$



Edge Points

- edge points = average of midpoint of the edge with average of the two new face points of the faces sharing the edge

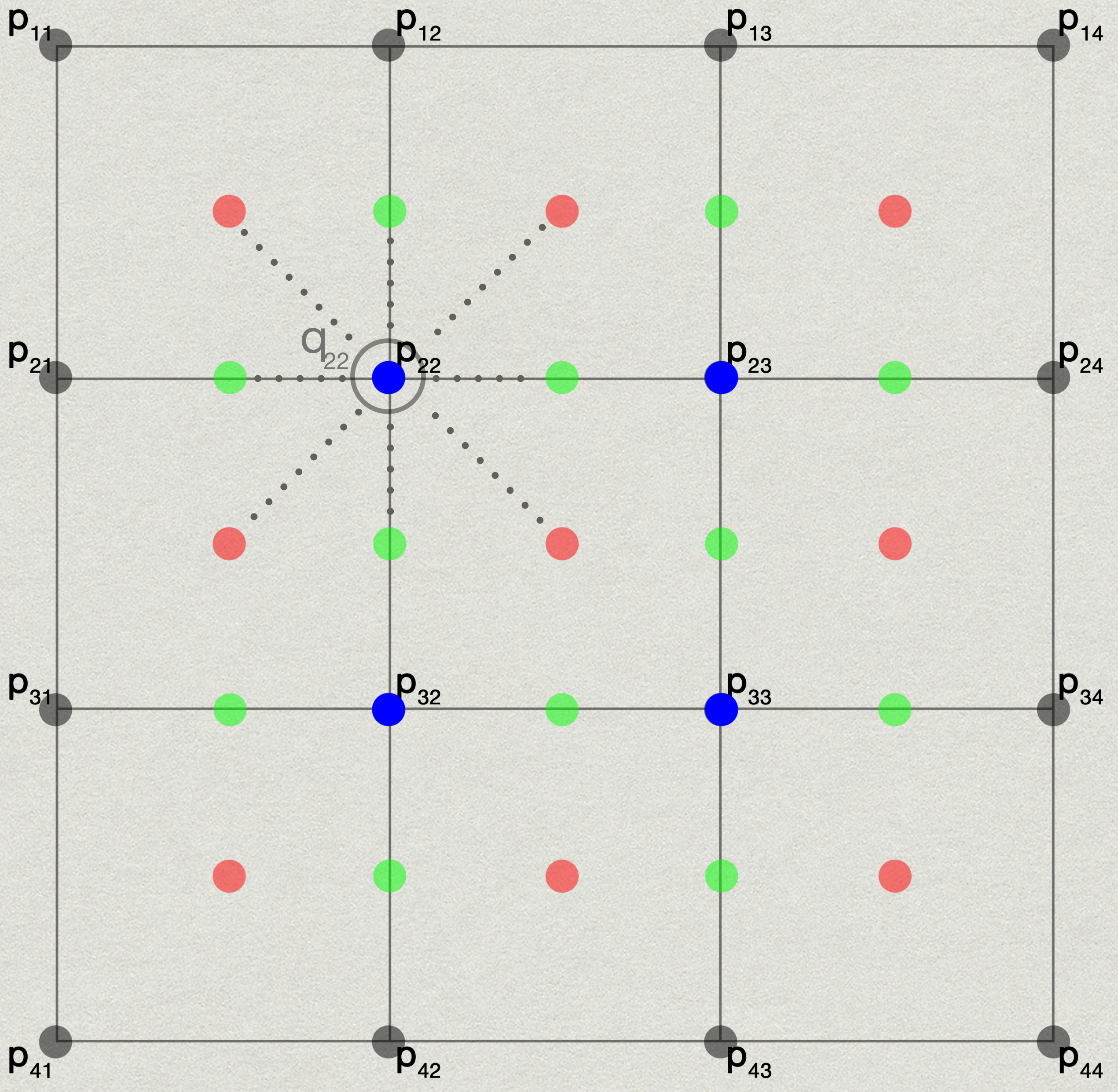
$$q_{12} = \frac{\frac{q_{11}+q_{13}}{2} + \frac{p_{12}+p_{22}}{2}}{2}$$



Vertex Points

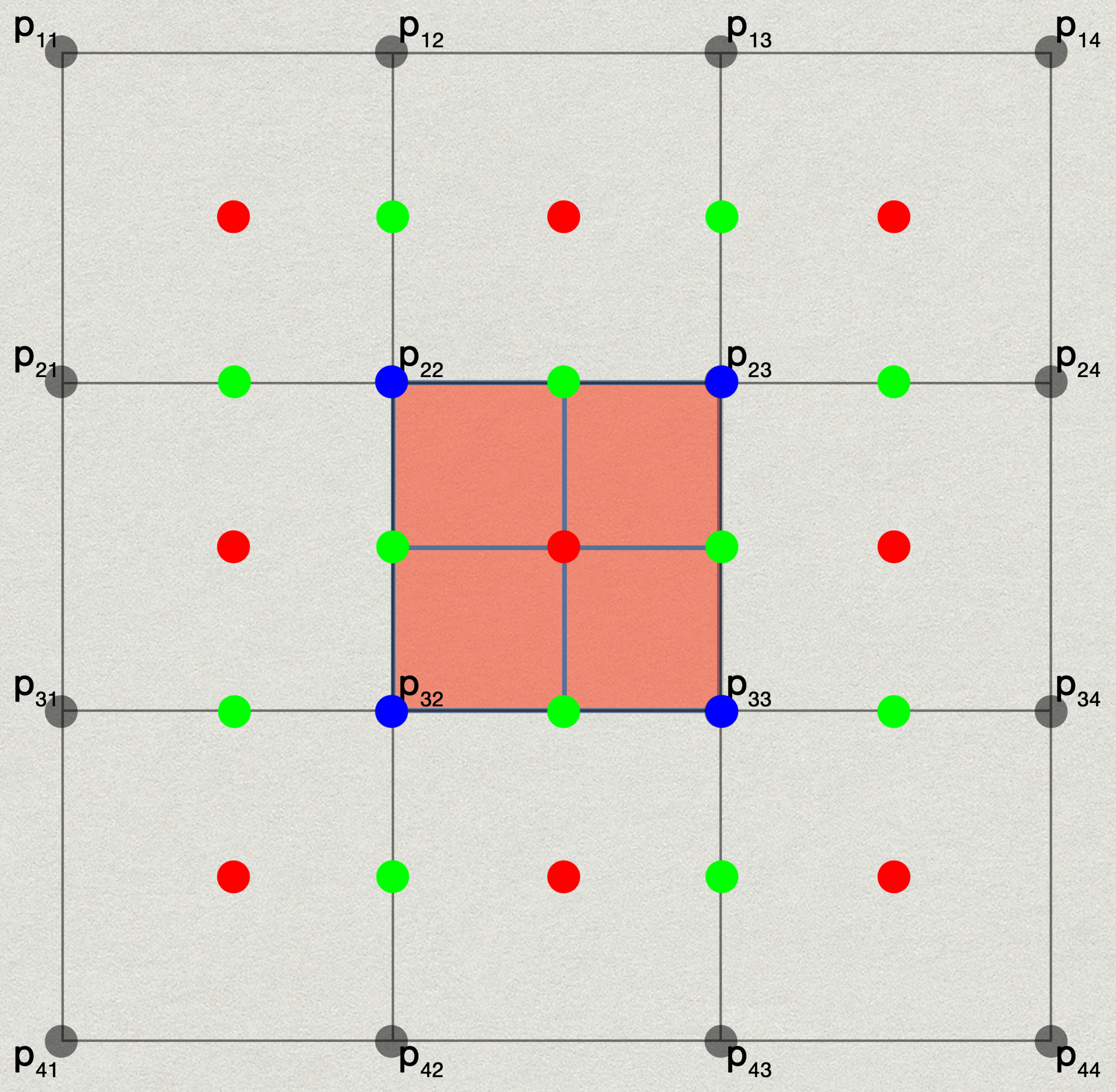
- Q = average of new face points of all faces adjacent to original vertex
- R = average of midpoints of all original edges incident to original vertex point
- new vertex point = average of Q, R, and original vertex point

$$q_{22} = \frac{Q}{4} + \frac{R}{2} + \frac{p_{22}}{4}$$



Result of Subdivision

- Requires all 16 points of p to interpolate center patch



Subdivision for Arbitrary Topology

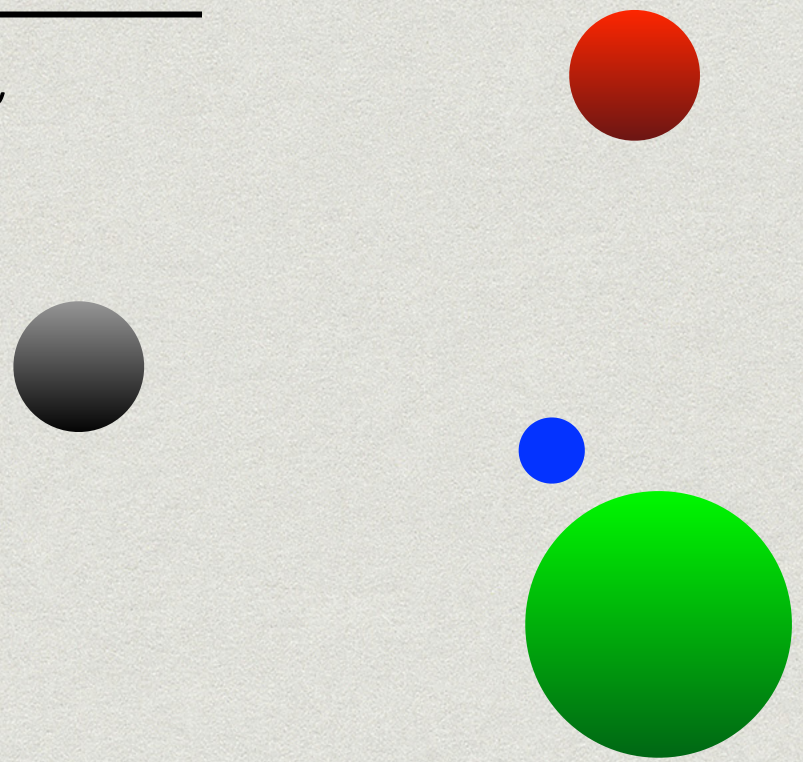
- S = original vertex point

$$vp = \frac{Q}{n} + \frac{2R}{n} + \frac{(n-3)S}{n}$$

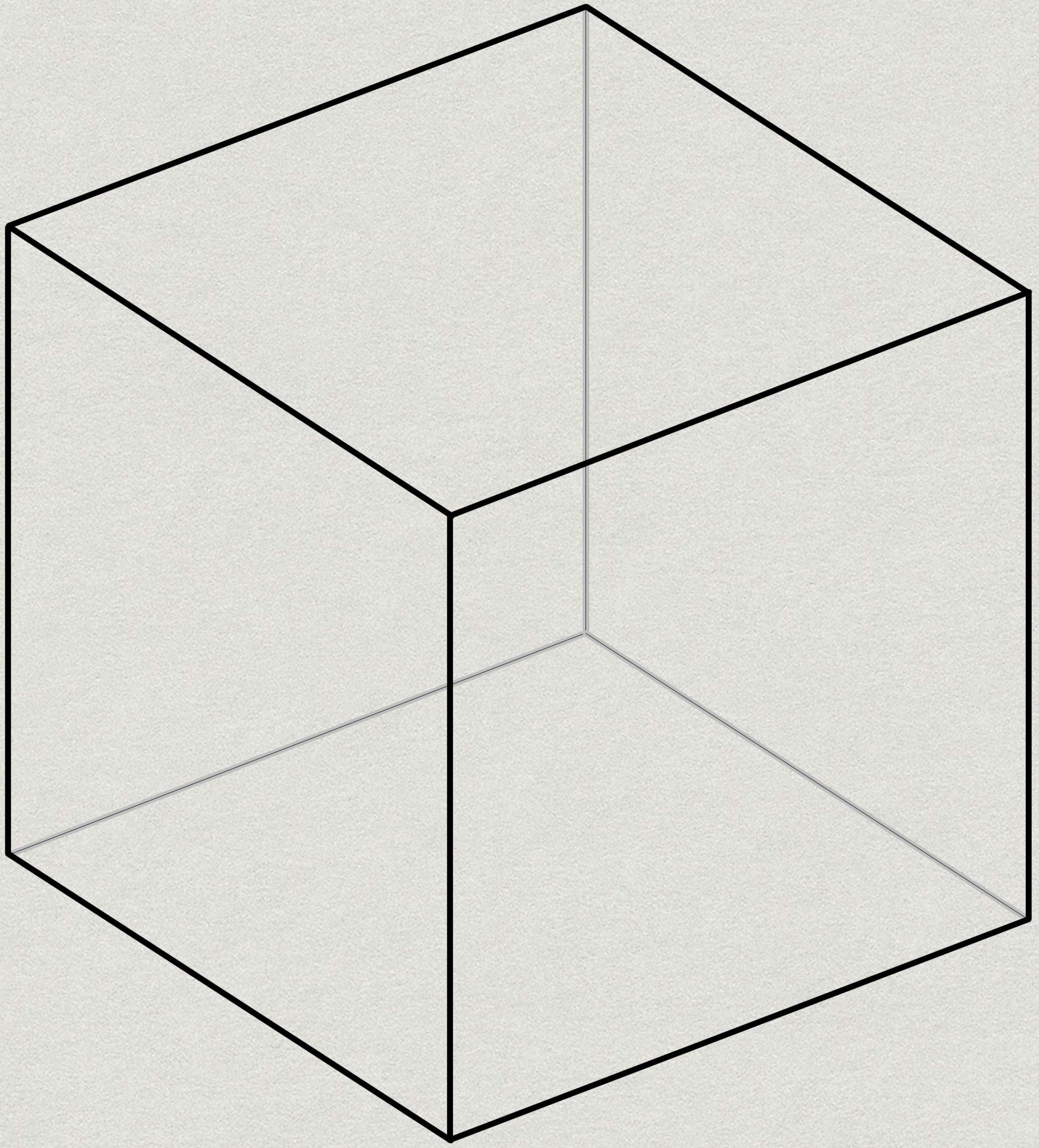
$$vp = \frac{4\beta Q}{n} + \frac{2(\alpha - 2\beta)R}{n} + \frac{(n-2\alpha)S}{n}$$

$$\alpha = \frac{3}{2}, \quad \beta = \frac{1}{2}, \quad n = 4$$

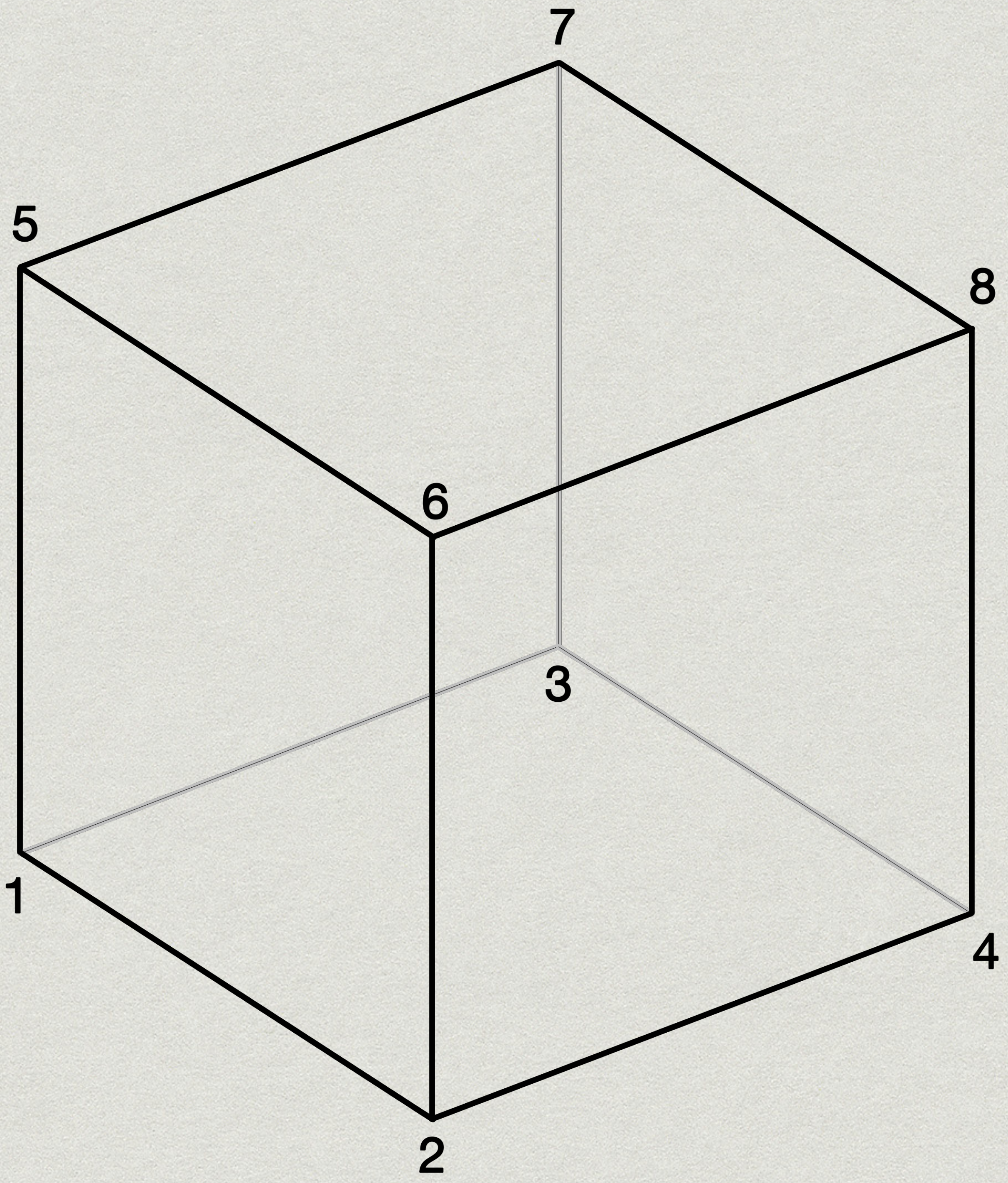
$$vp = \frac{Q}{4} + \frac{2R}{4} + \frac{S}{4}$$



Defining the 3D Object



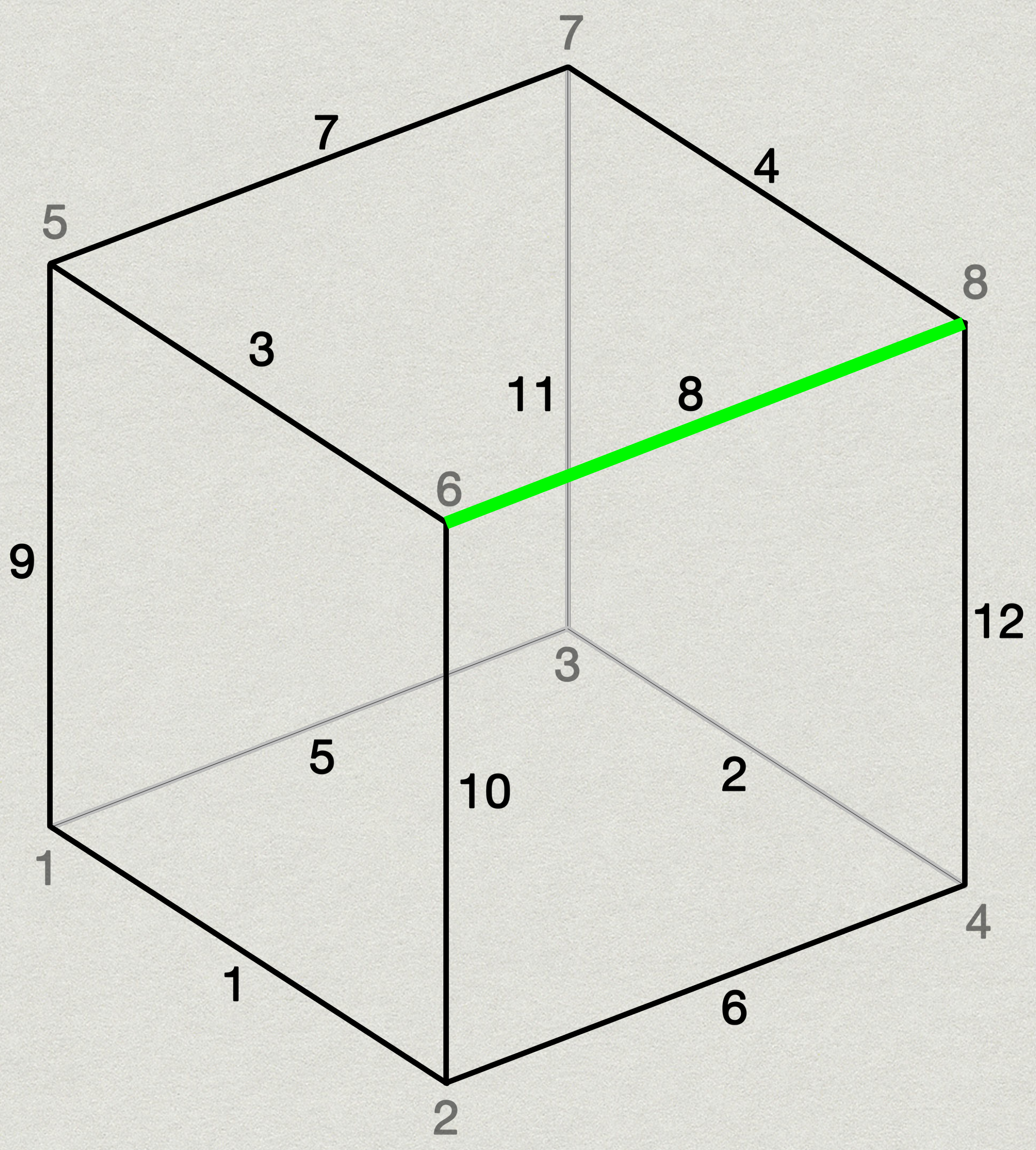
Vertex Coordinates



$$v = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

8 x 3

Define Edges

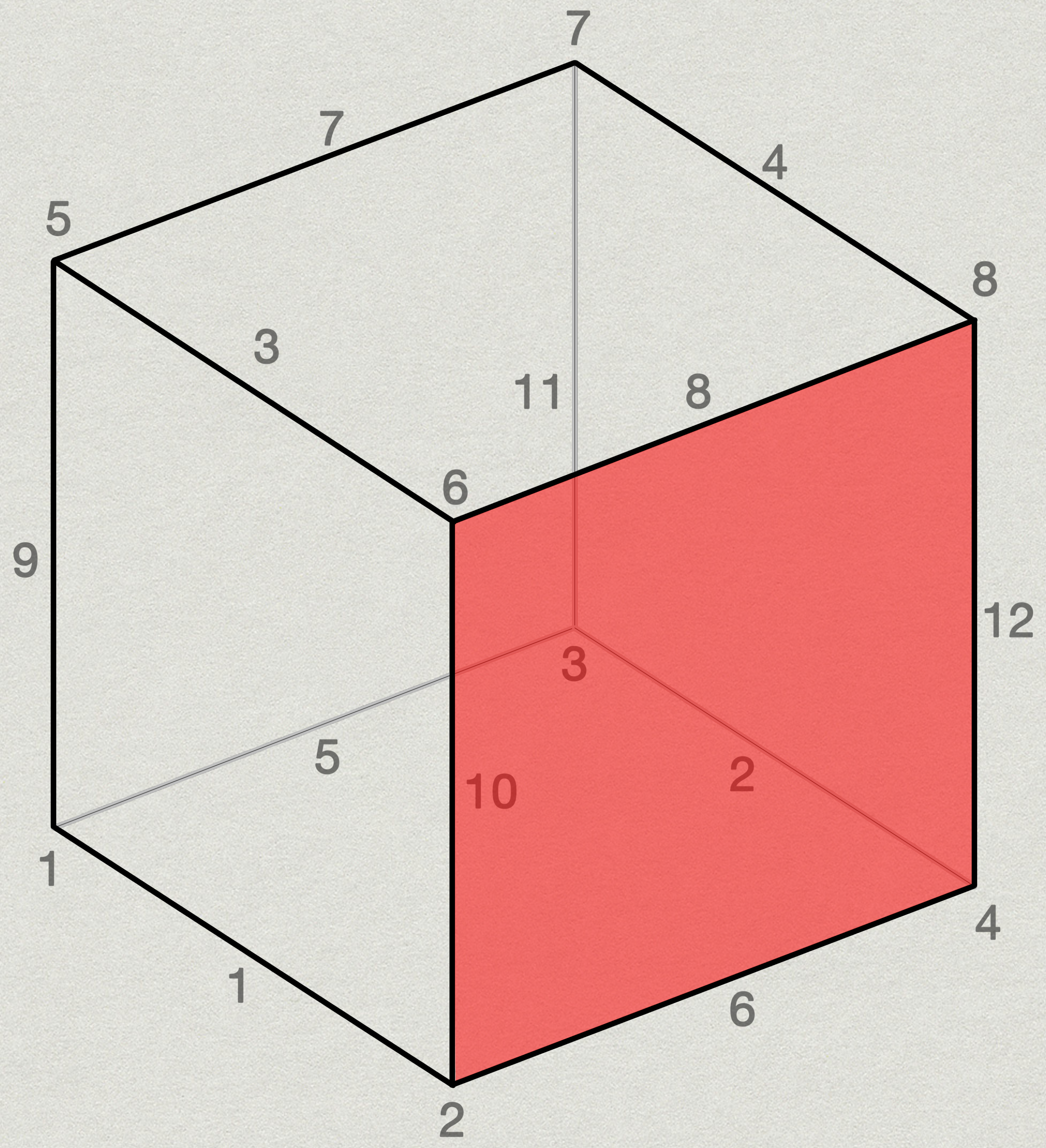


$$e = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \\ 1 & 3 \\ 2 & 4 \\ 5 & 7 \\ 6 & 8 \\ 1 & 5 \\ 2 & 6 \\ 3 & 7 \\ 4 & 8 \end{bmatrix}$$

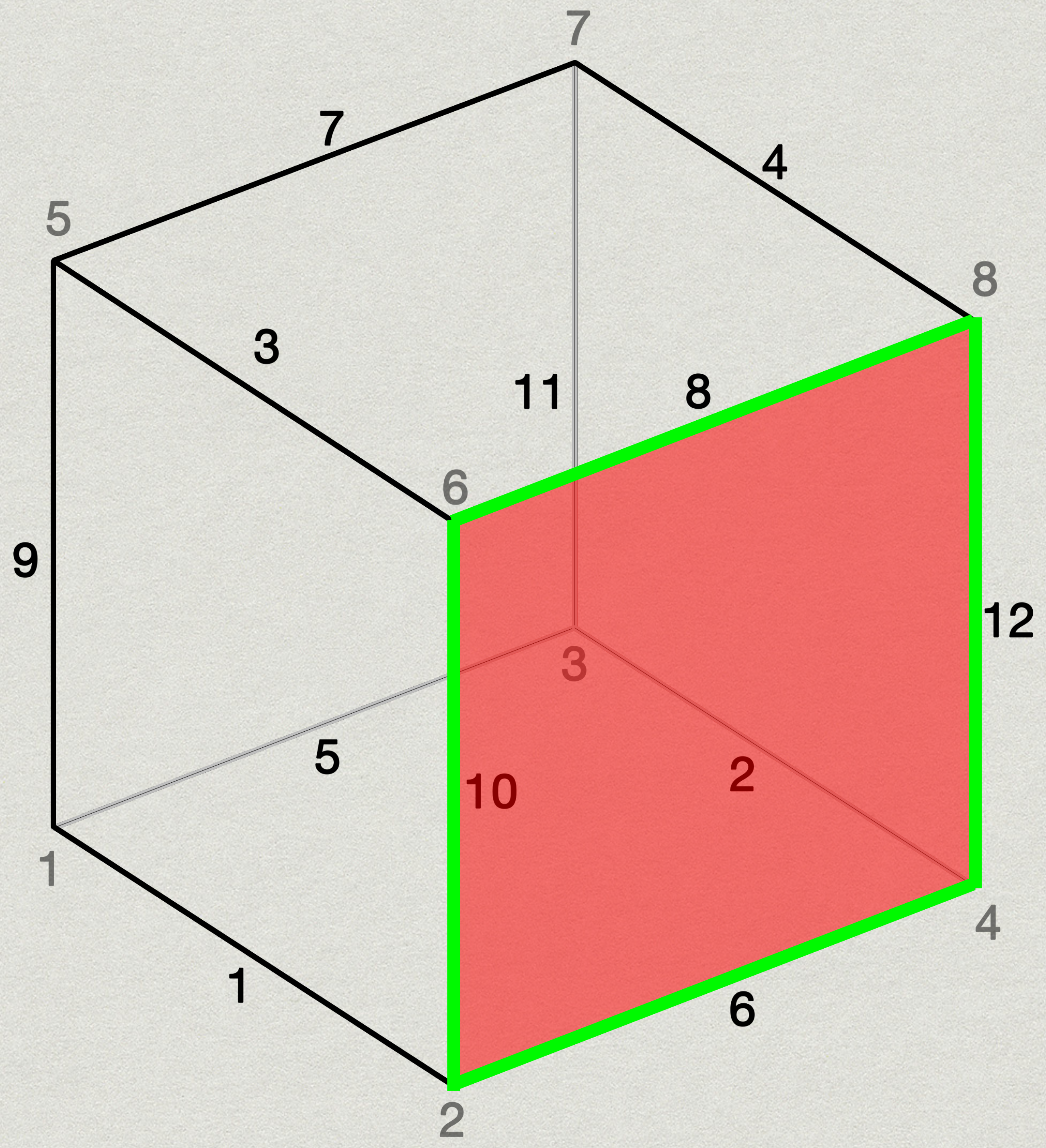
8

12 x 2

Define Faces:



Define Faces: With Edges

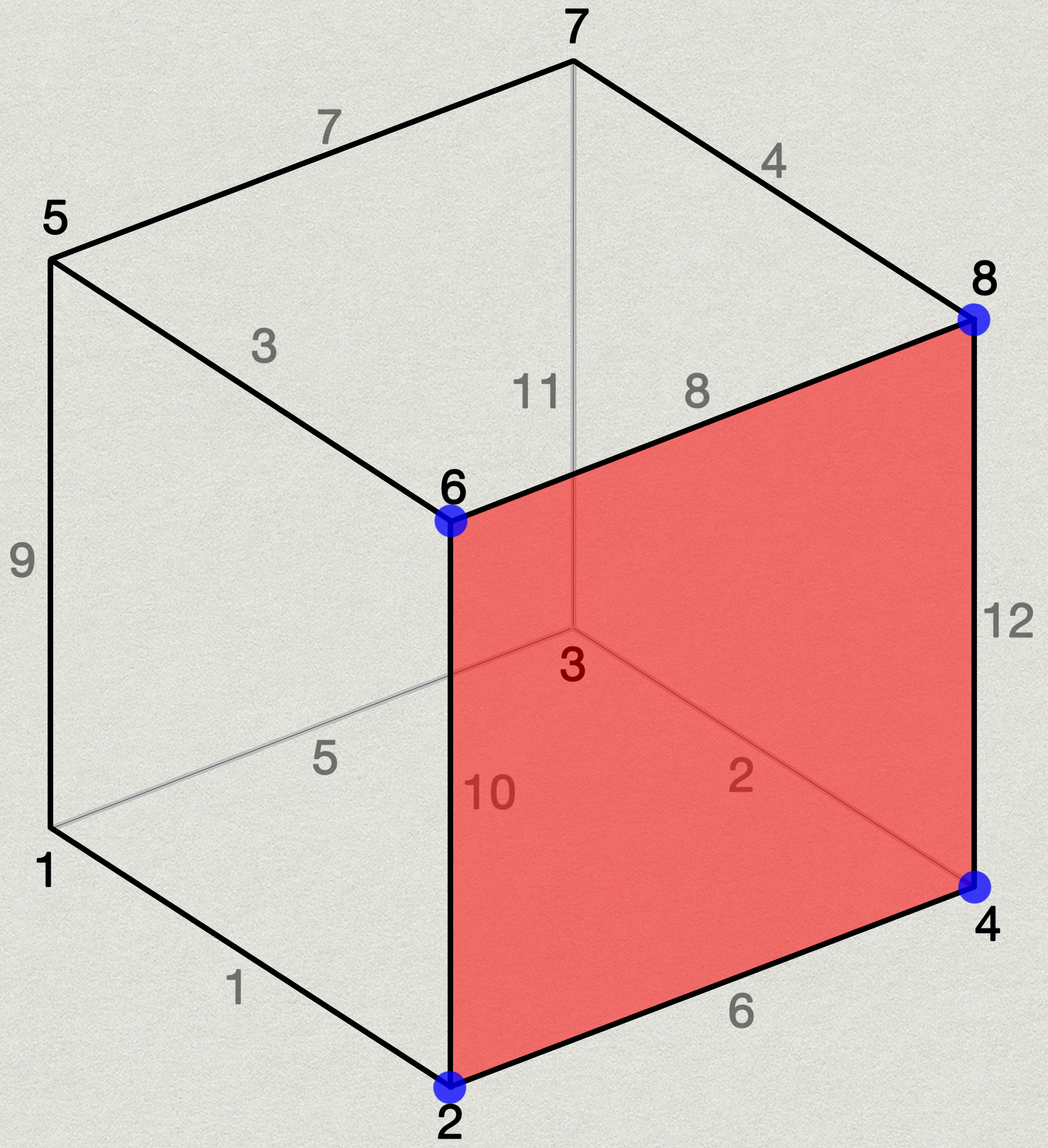


$$fe = \begin{bmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 7 & 8 \\ 1 & 3 & 9 & 10 \\ 2 & 4 & 11 & 12 \\ 5 & 7 & 9 & 11 \\ 6 & 8 & 10 & 12 \end{bmatrix}$$

6

6 x 4

Define Faces: With Vertices



$$fv = \begin{bmatrix} 1 & 2 & 4 & 3 \\ 5 & 6 & 8 & 7 \\ 1 & 2 & 6 & 5 \\ 3 & 4 & 8 & 7 \\ 1 & 3 & 7 & 5 \\ 2 & 4 & 8 & 6 \end{bmatrix}$$

6

6 x 4

Catmull-Clark Subdivision Algorithm

INPUT <—— {v, e, Fe, Fv, (Nv, Ne, Nf)}

DO:

1. calculate face points ***fp*** *(for each face)*

2. calculate edge points ***ep*** *(for each edge)*

3. calculate vertex points ***vp*** *(for each vertex)*

4a. update vertices **v** *-> [* ***vp*** *;* ***ep****;* ***fp*** *]*

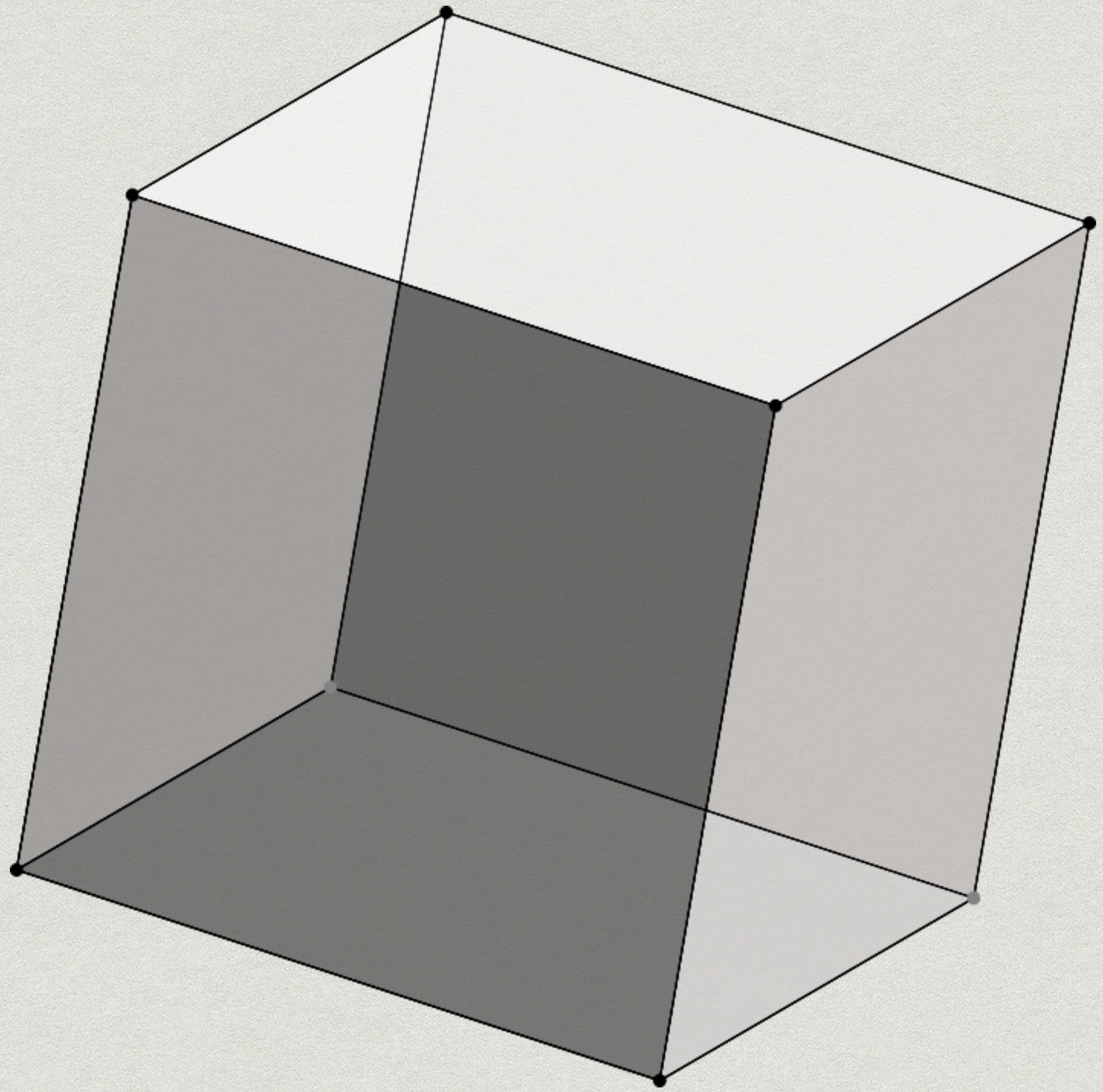
4b. update edges and faces

WHILE (not ***stopping condition***)

OUTPUT ——> {v, e, Fe, Fv}

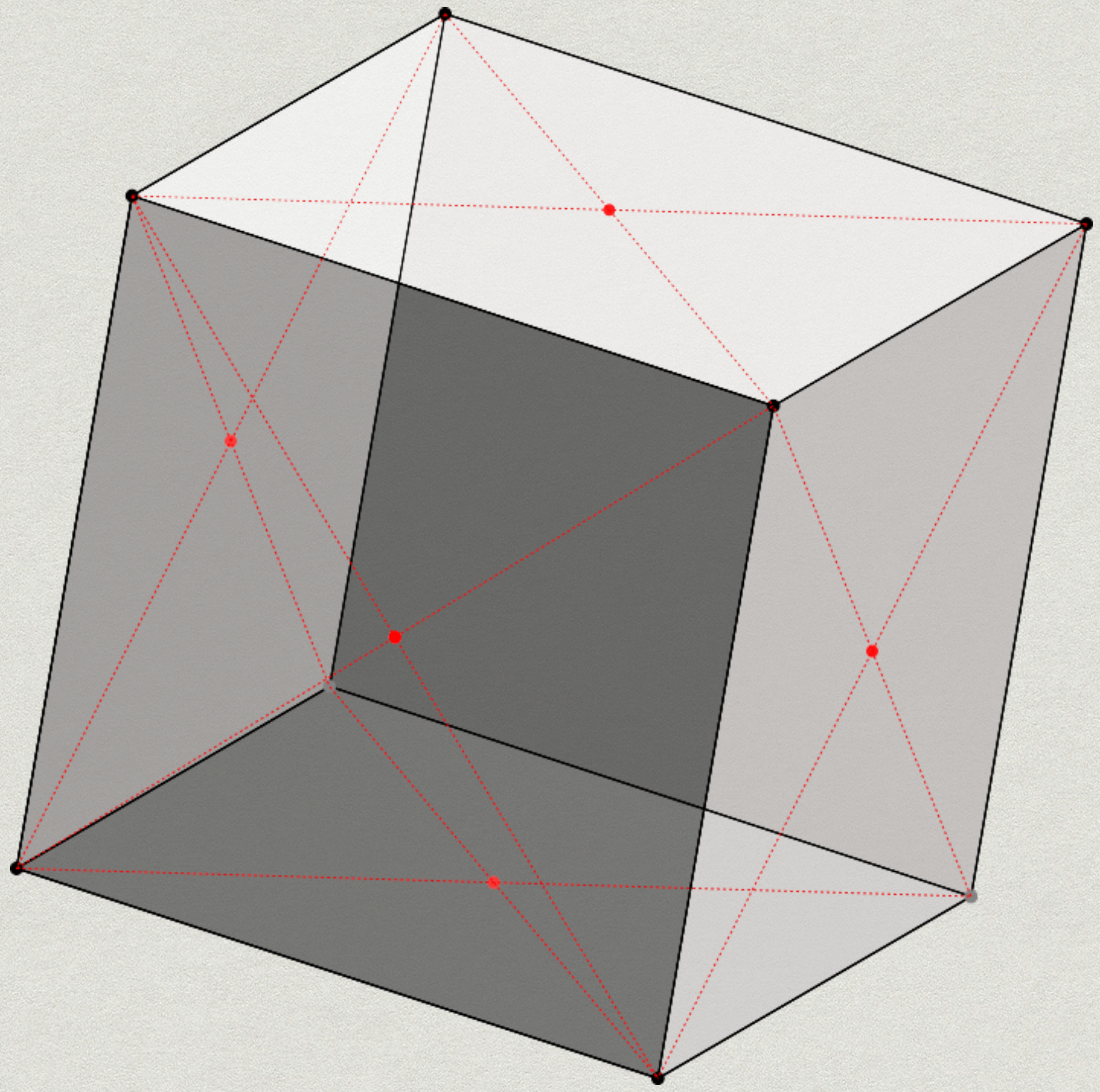
STEP 1: Face Points

* For each face, create a **face point** as the average of vertices on the face.



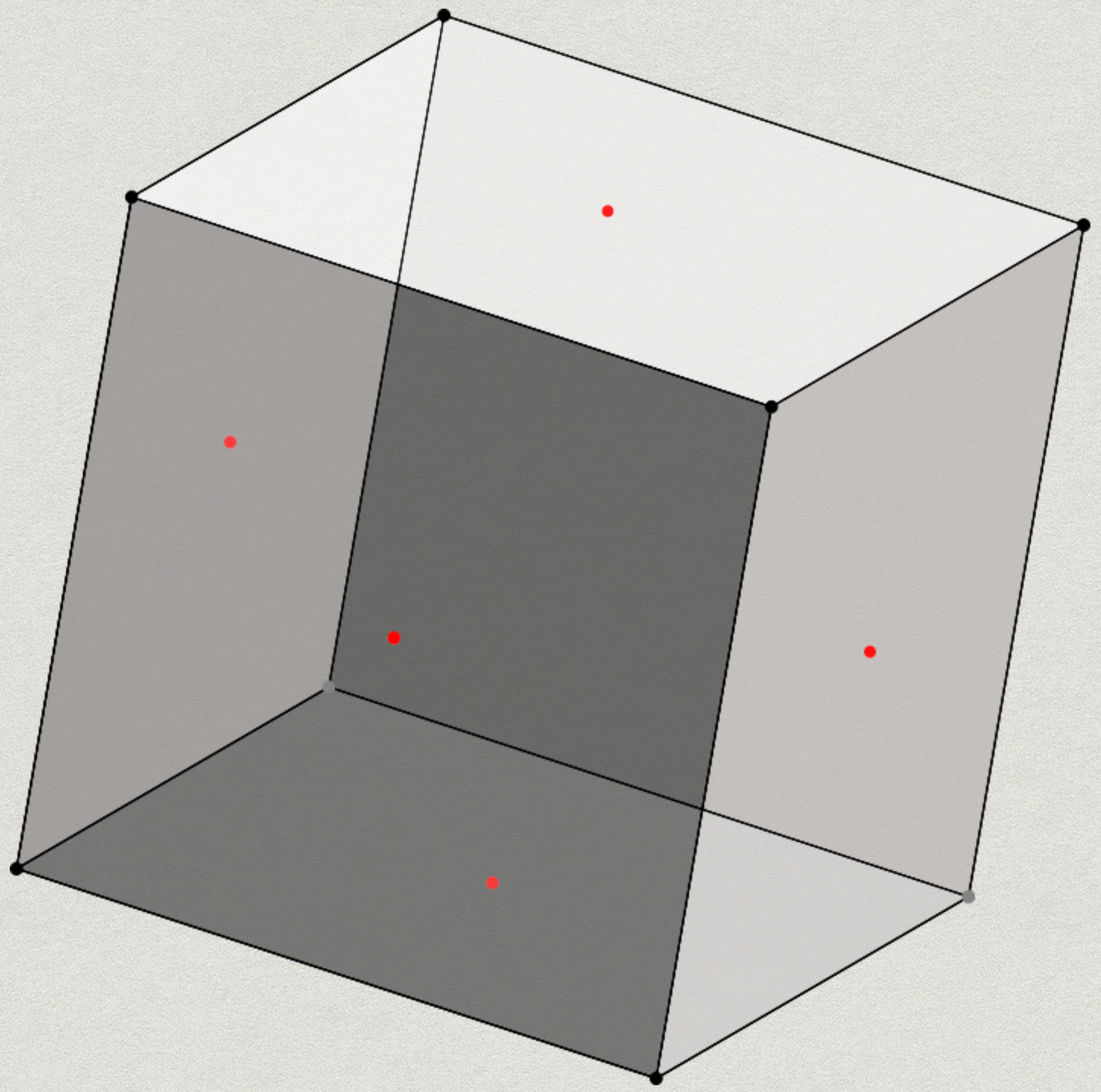
STEP 1: Face Points

- For each face, create a **face point** as the average of vertices on the face.



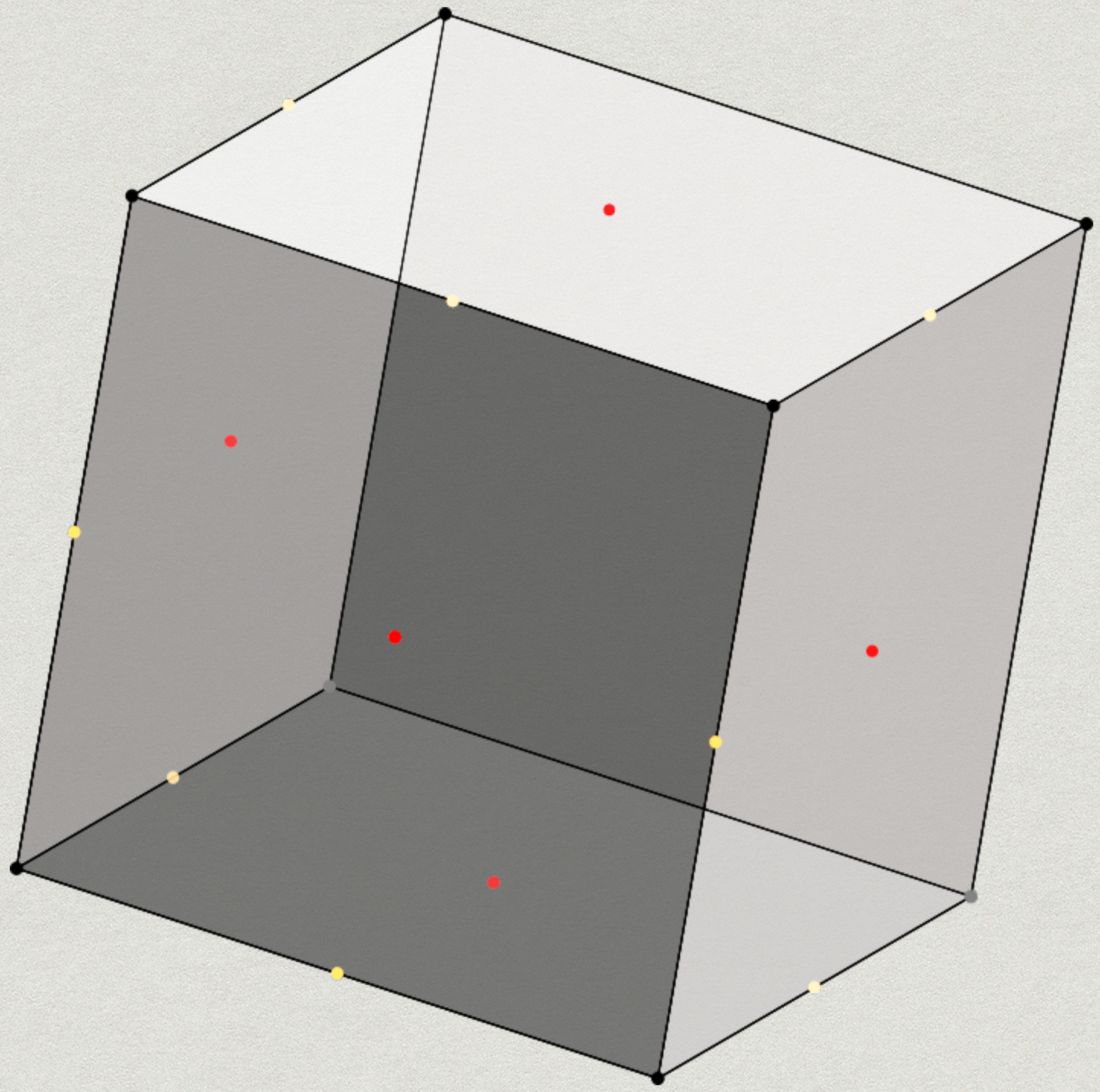
STEP 1: Face Points

- For each face, create a **face point** as the average of vertices on the face.



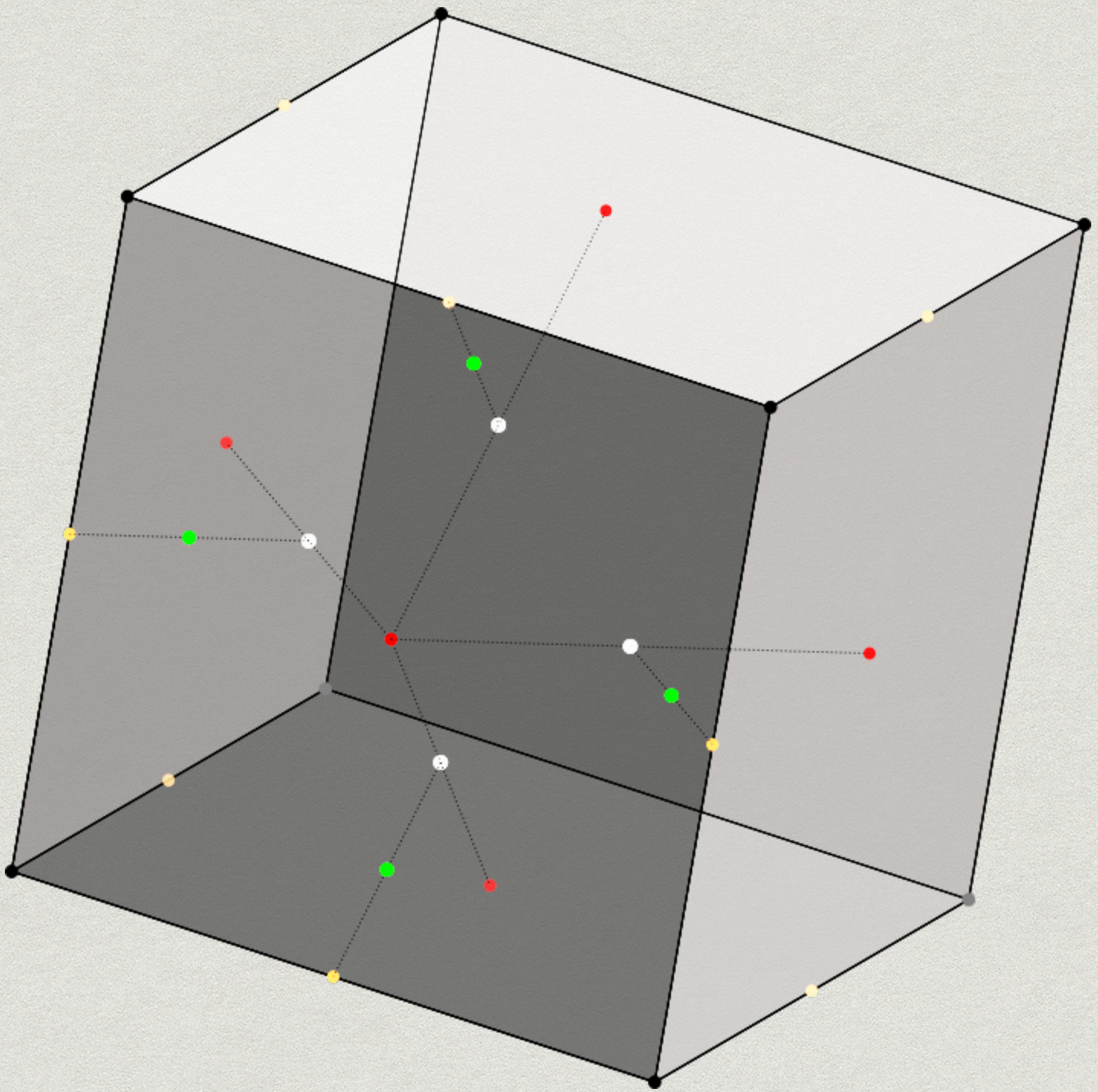
STEP 2: Edge Points

- Find the **midpoint** of each edge
- Find the **average** of the two face points whose faces share that edge
- Average these two points -> **ep**



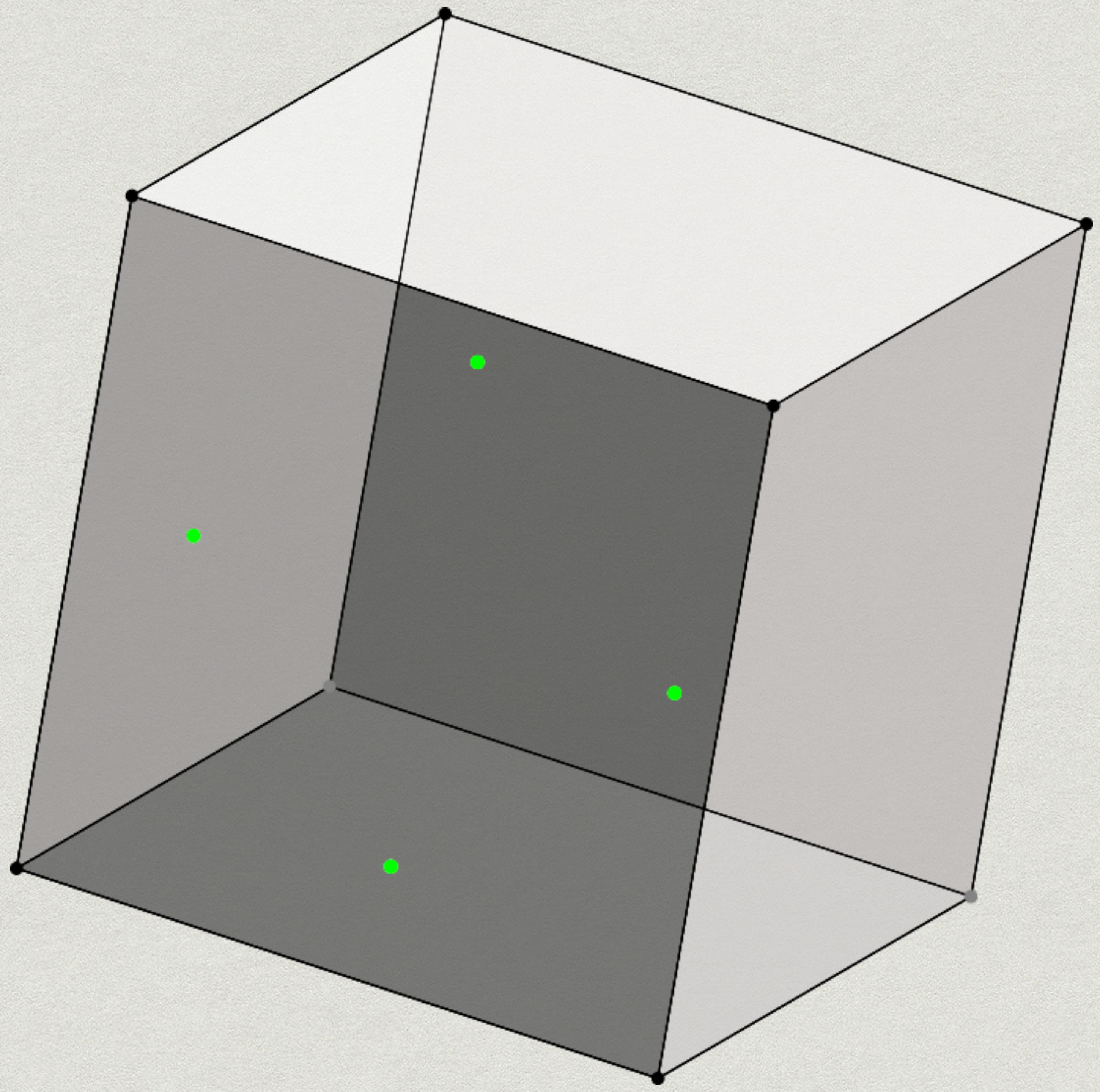
STEP 2: Edge Points

- Find the **midpoint** of each edge
- Find the **average** of the two face points whose faces share that edge
- Average these two points -> **ep**



STEP 2: Edge Points

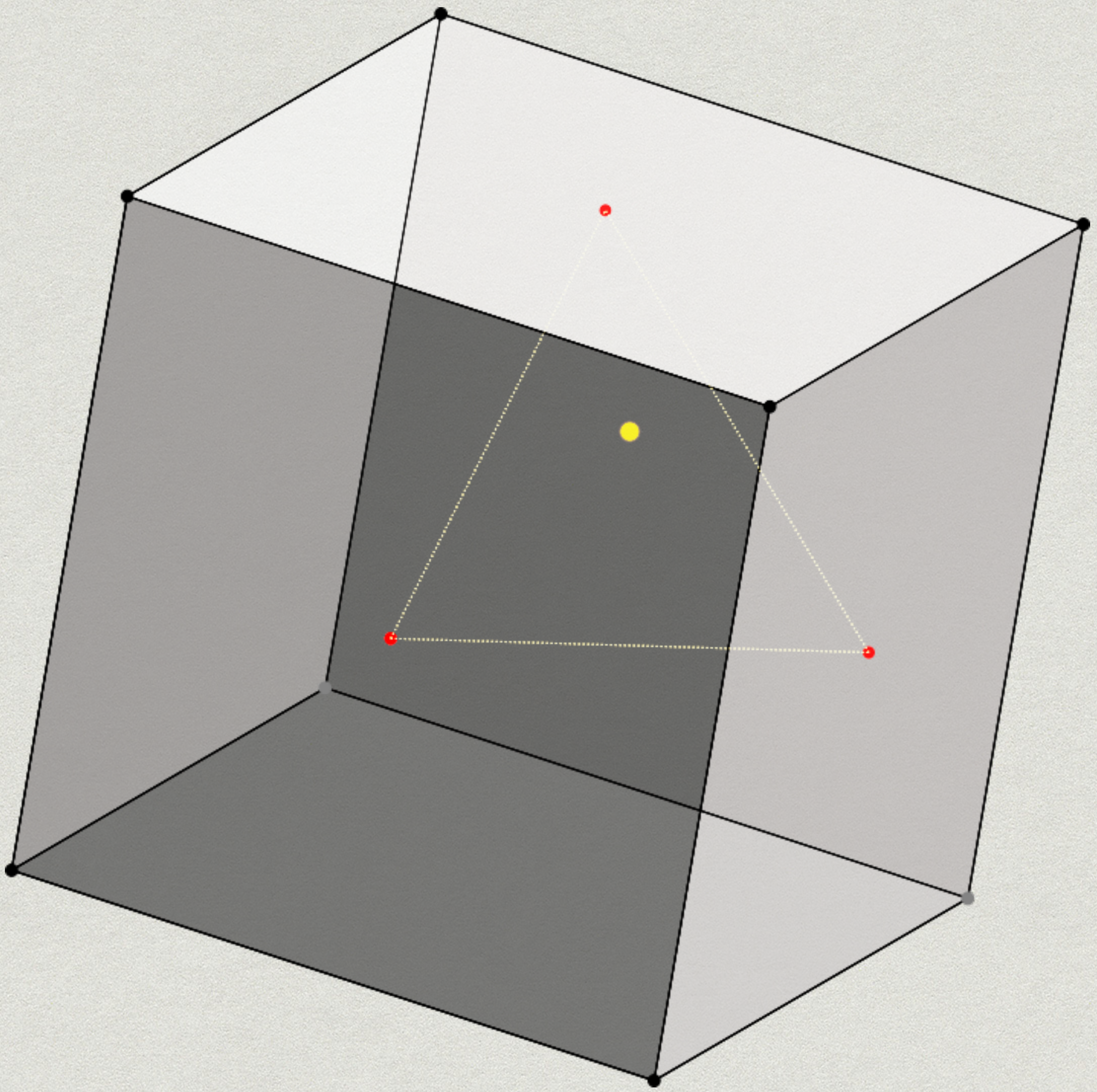
- Find the **midpoint** of each edge
- Find the **average** of the two face points whose faces share that edge
- Average these two points -> **ep**



STEP 3: New Vertices

For each vertex:

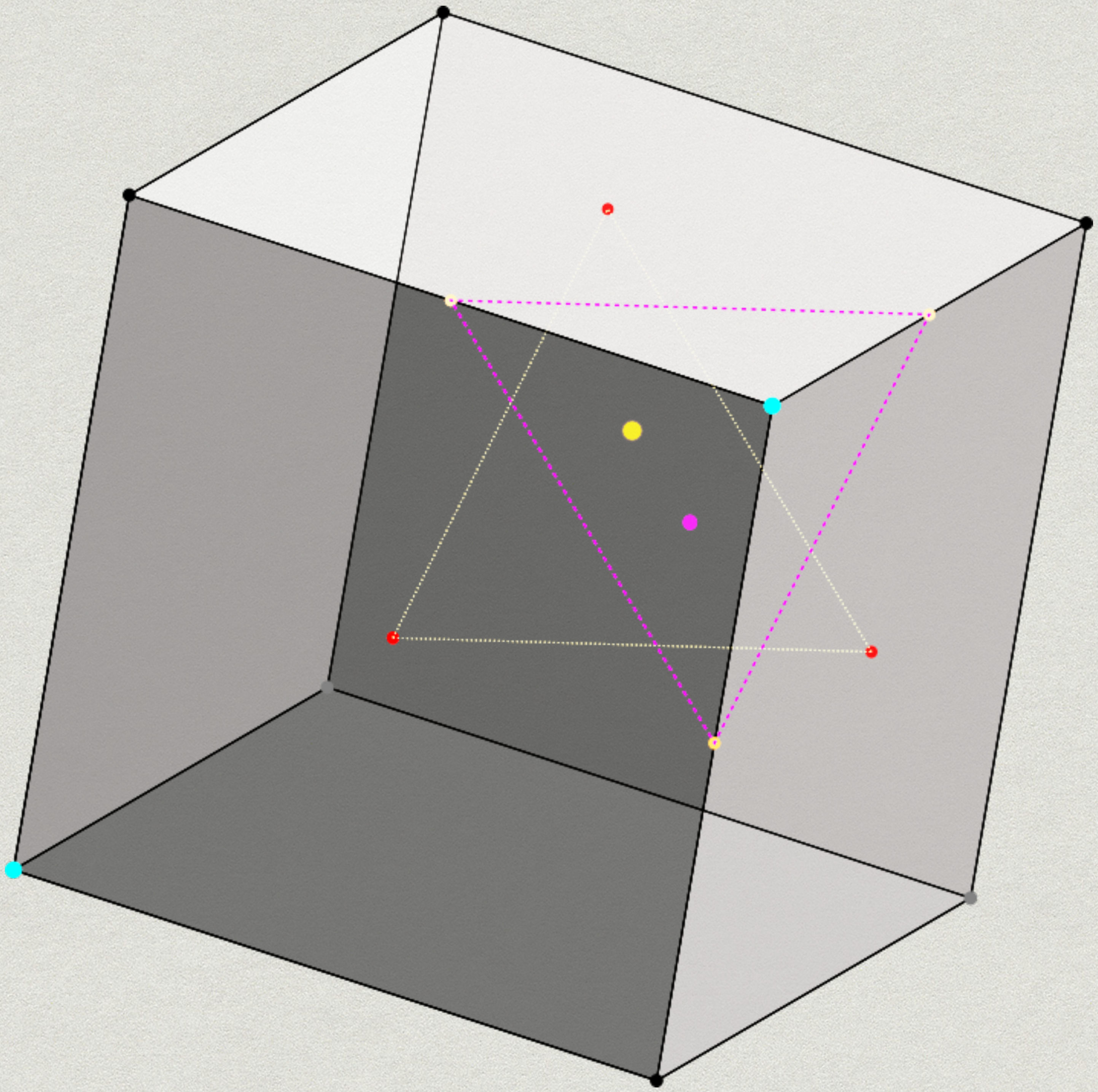
- find the **average** of all the face points whose faces have that matrix
- find the **average** of all the edge midpoints whose edges have that vertex
- take the original **vertex**
- take the weighted average of these to create the new vertex point -> **vp**



STEP 3: New Vertices

For each vertex:

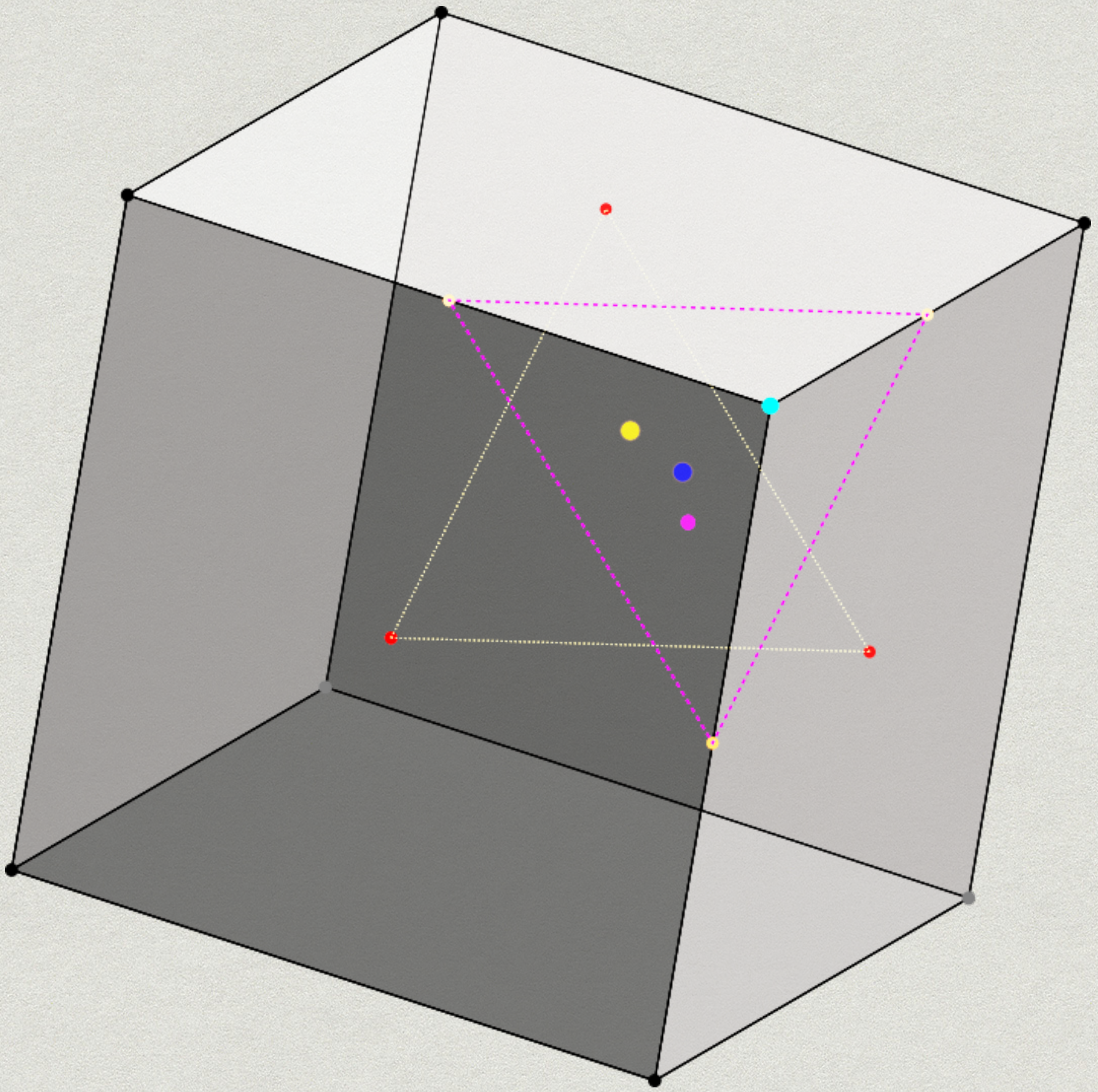
- find the **average** of all the face points whose faces have that matrix
- find the **average** of all the edge midpoints whose edges have that vertex
- take the original **vertex**
- take the weighted average of these to create the new vertex point -> **vp**



STEP 3: New Vertices

For each vertex:

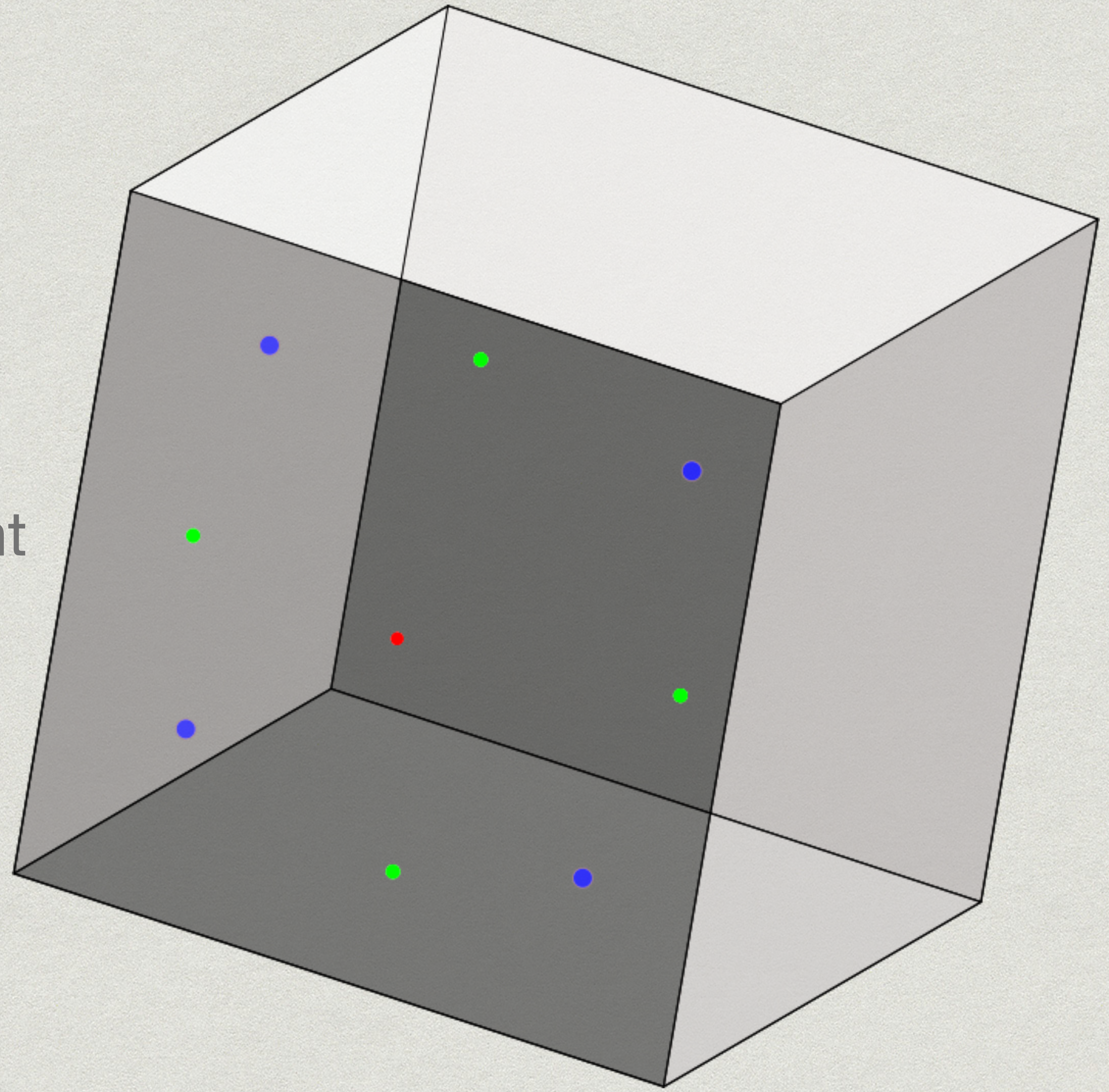
- find the **average** of all the face points whose faces have that matrix
- find the **average** of all the edge midpoints whose edges have that vertex
- take the original **vertex**
- take the weighted average of these to create the new vertex point -> **vp**



STEP 4: Update

For each vertex:

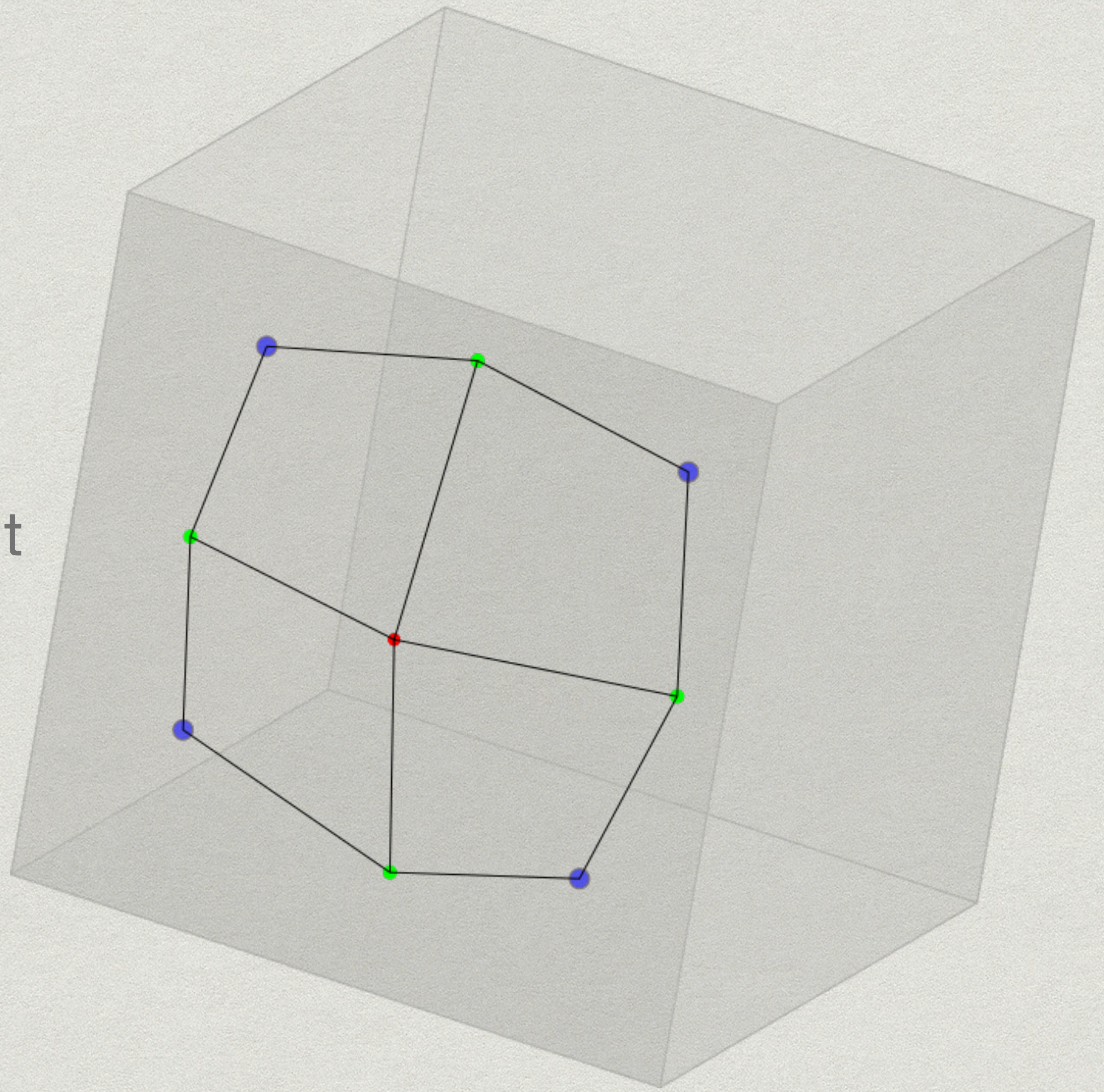
- **v -> [vp ; ep ; fp]**
- add new edges between the face point and edge points
- … and between edge points and vertex points

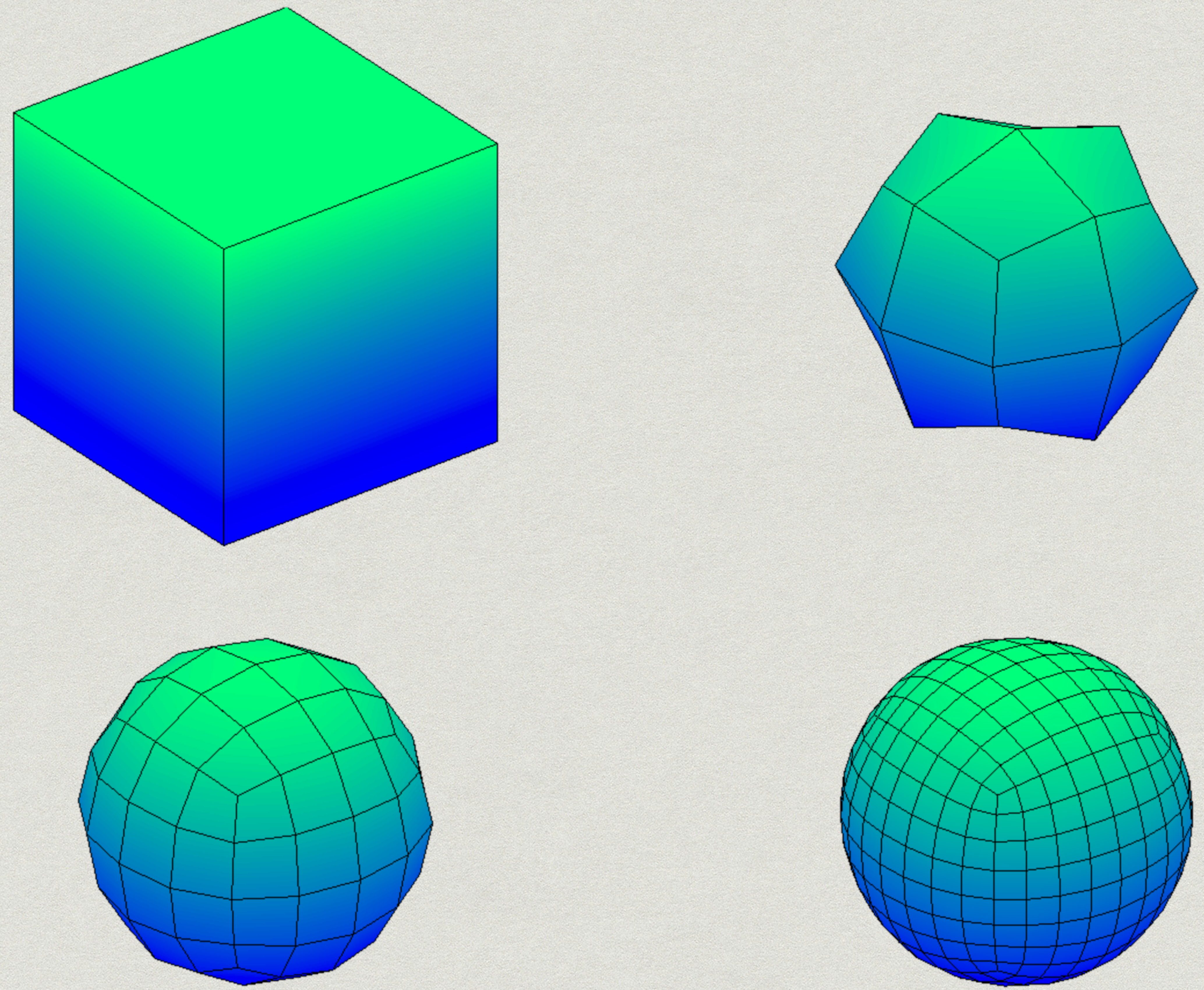


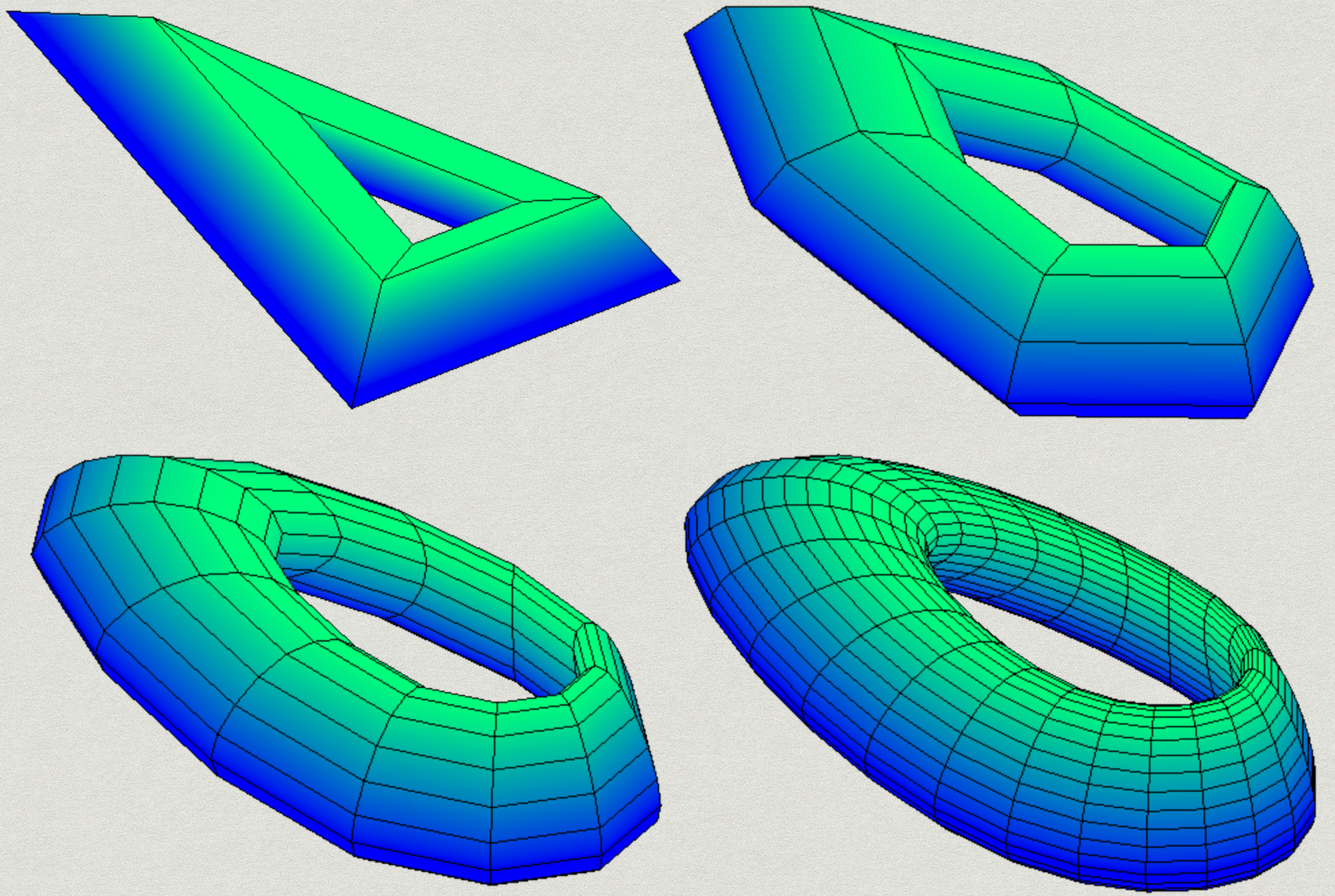
STEP 4: Update

For each vertex:

- **v** -> [**vp** ; **ep** ; **fp**]
- add new edges between the face point and edge points
- … and between edge points and vertex points



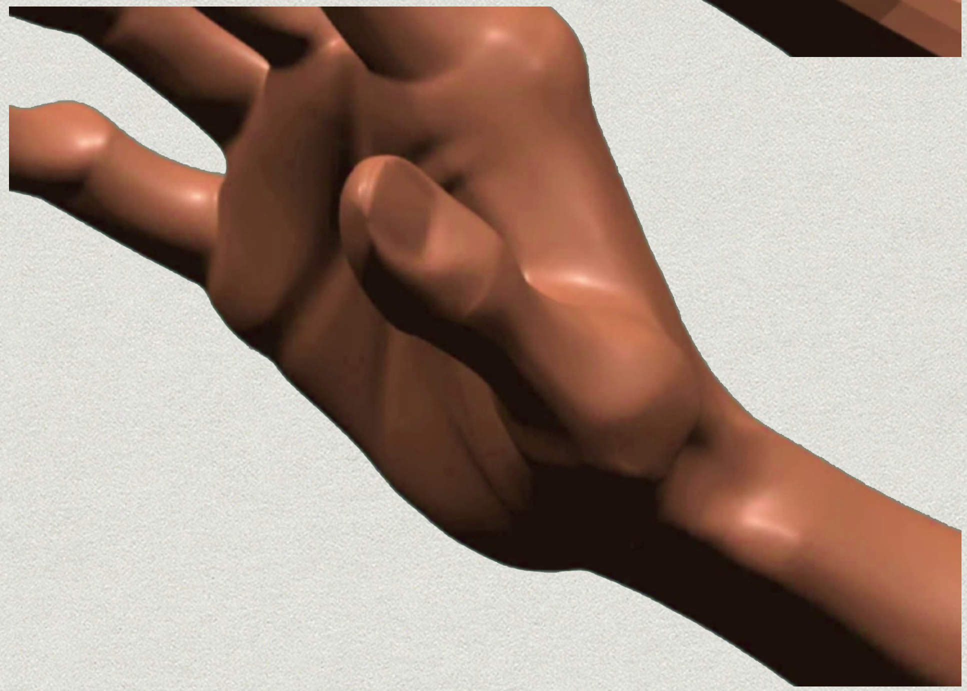


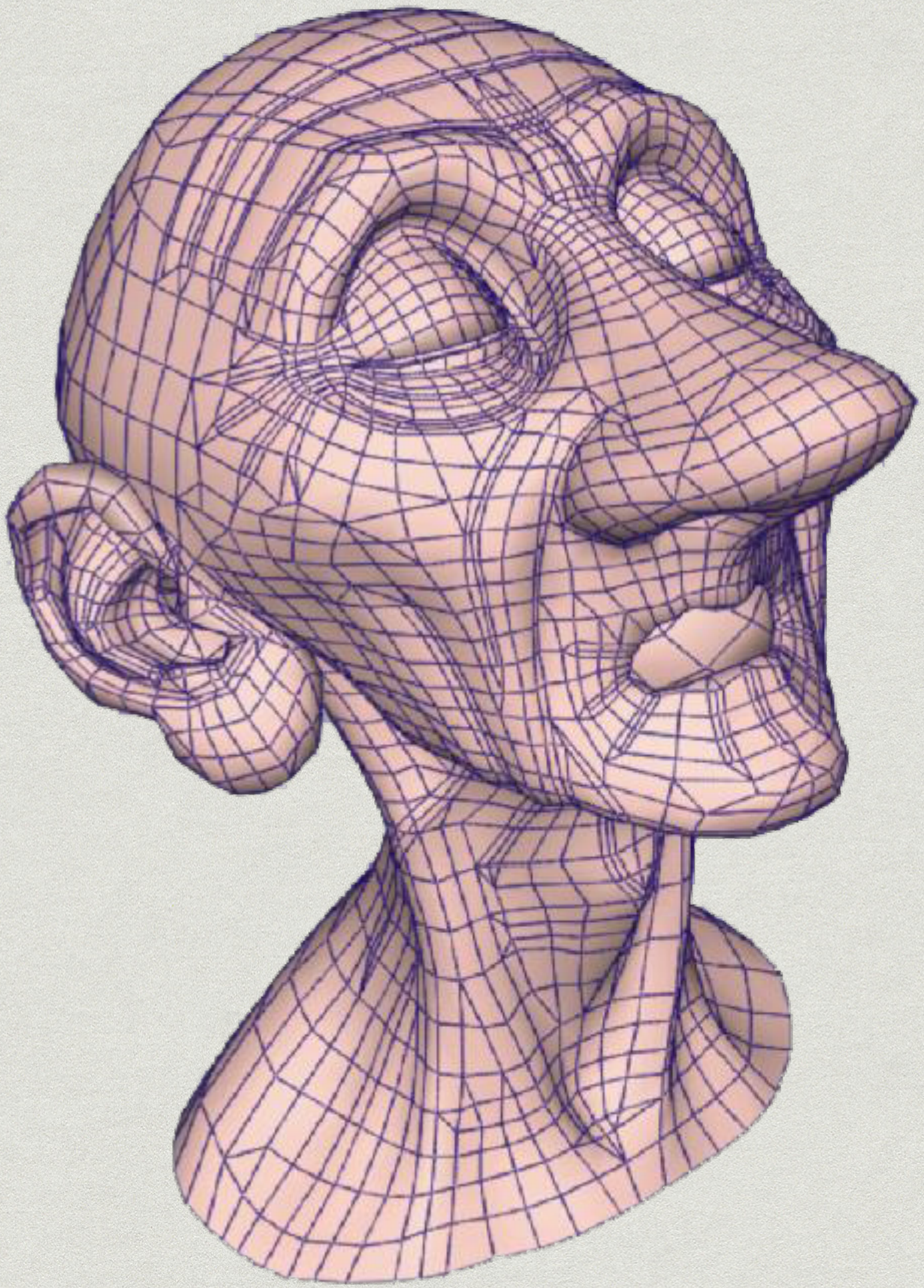


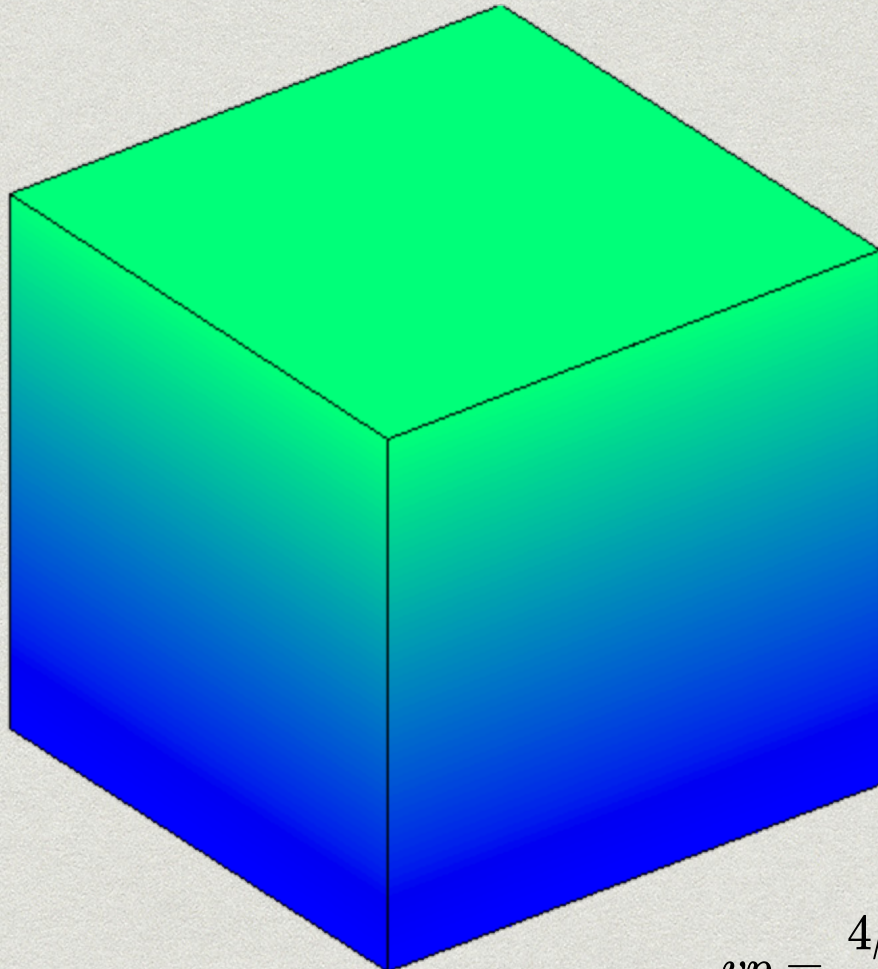
Possible Stopping Conditions

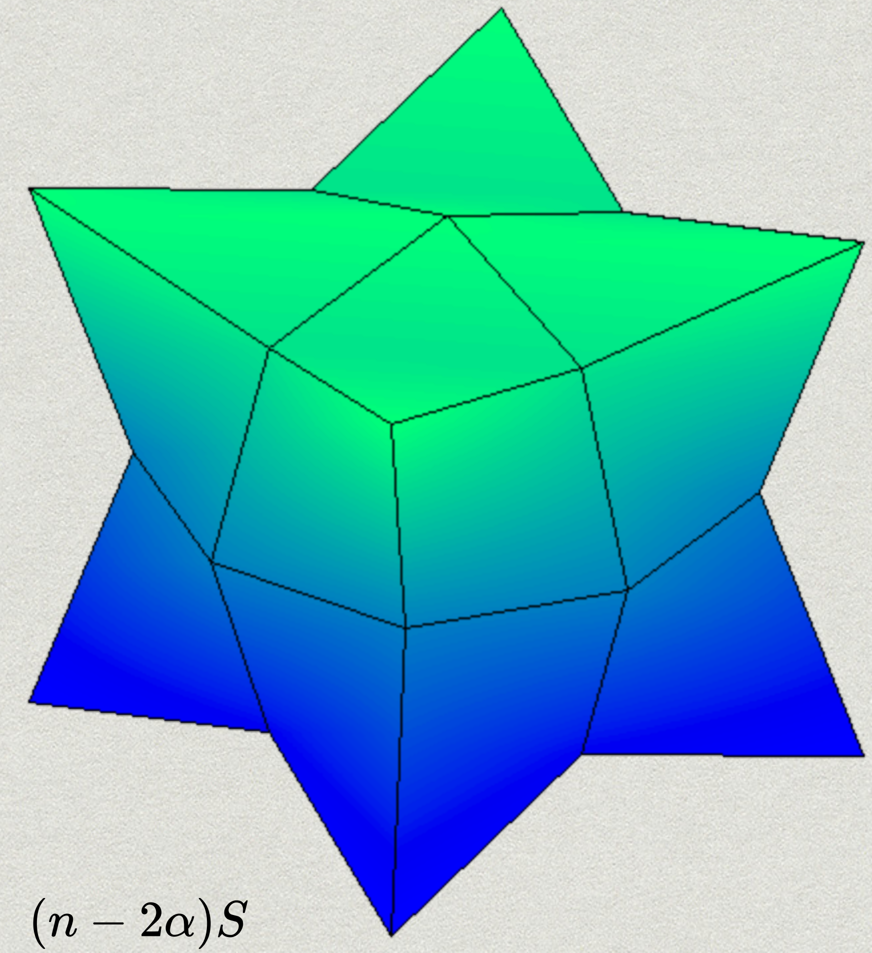
- MAX_ITERATIONS
- time-based (in real-time rendering)
- memory-based (number of vertices greater than some *N*)
- per resolution (vertices less than 1 pixel apart, etc.)

Applications

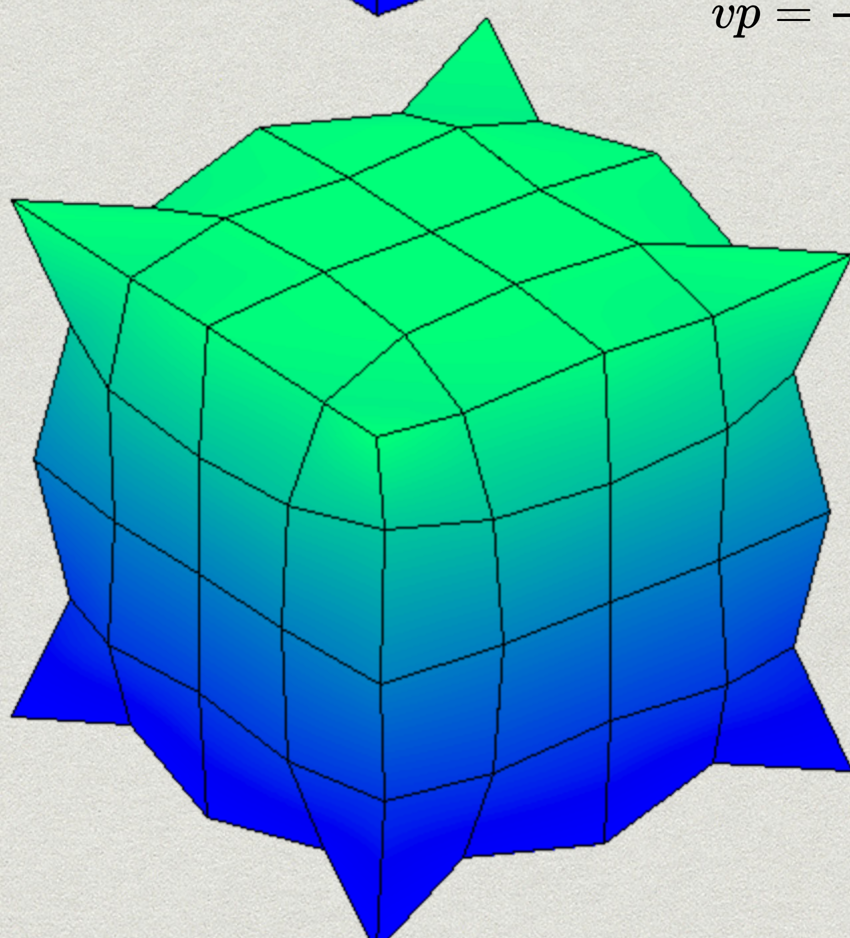




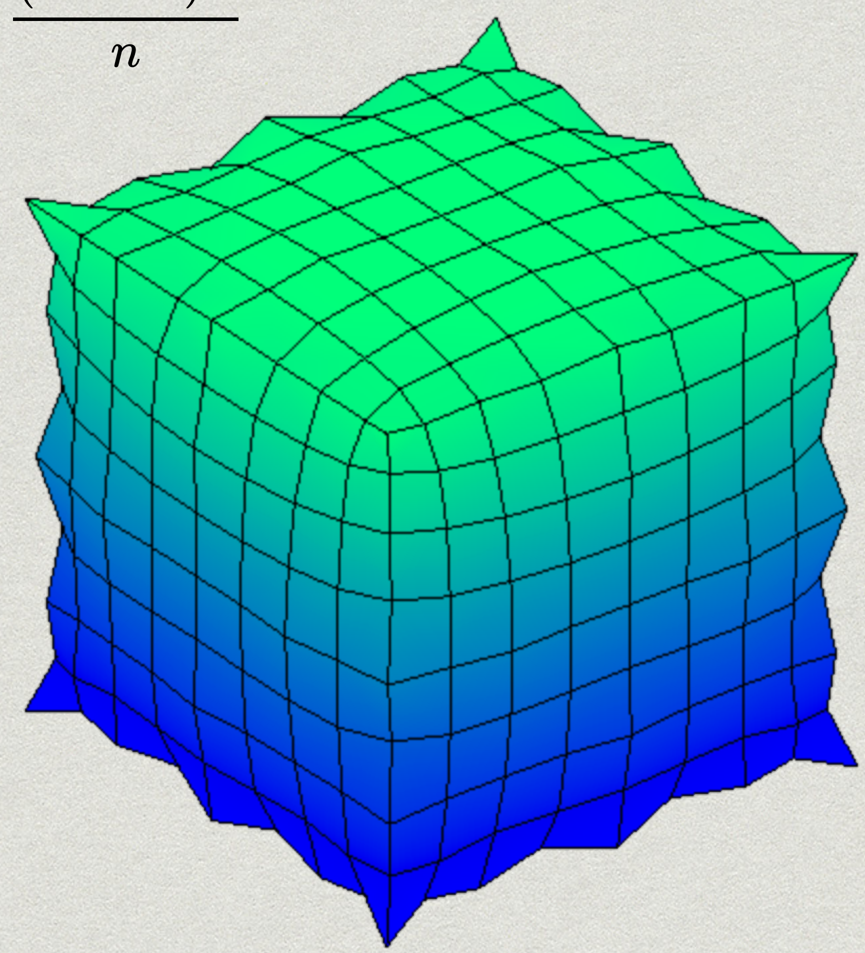


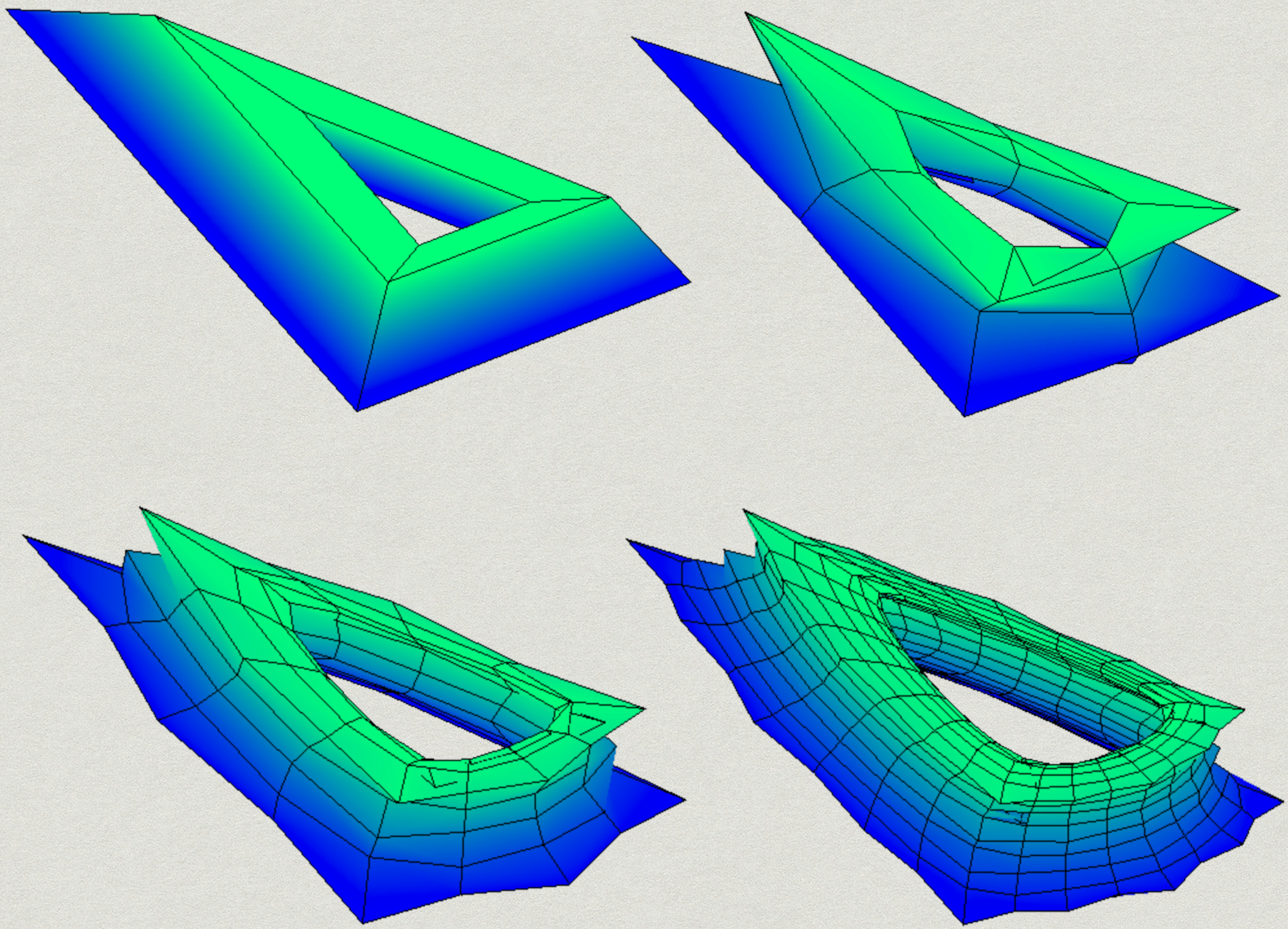


$$vp = \frac{4\beta Q}{n} + \frac{2(\alpha - 2\beta)R}{n} + \frac{(n - 2\alpha)S}{n}$$



$$\alpha = 2$$
$$\beta = -1$$





References

Recursively Generated B-Spline Surfaces on Arbitrary Topological Meshes

- E Catmull and J Clark
- http://www.cs.berkeley.edu/~sequin/CS284/PAPERS/CatmullClark_SDSurf.pdf