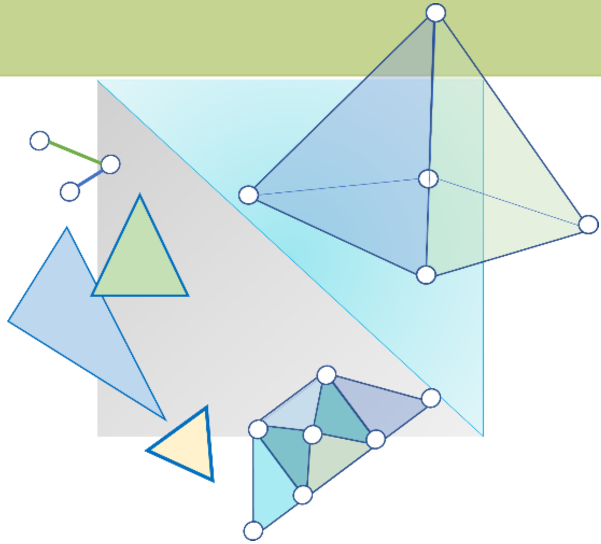
CITS3003 Graphics & Animation

Lecture 21:
3D Modelling



Objectives

- Modelling complex 3D objects
  - How to get 3D model
- Techniques used by 3D modelling software
  - Subdivision surfaces
    - Catmull Clark algorithm

Why do we need 3D modelling?

- So far we've mainly been concerned with drawing complex 3D models while mostly ignoring where the models are coming from

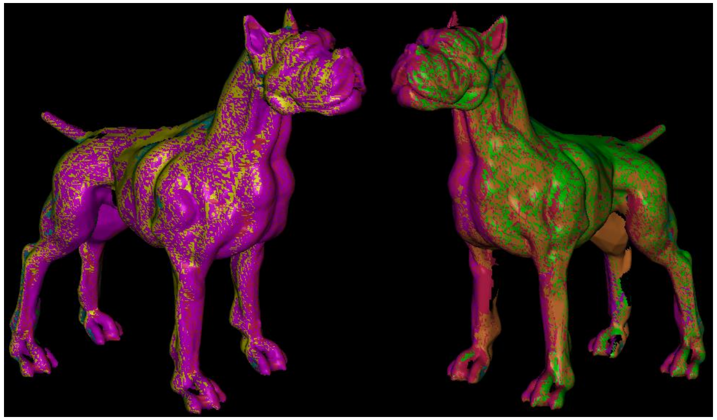
- Models need to come from somewhere

- 3D Models can come from three main sources
    1. Scanning real objects
    2. Non-rigid deformation of real object scans
    3. Making synthetic models in 3D modelling softwares (CAD)

Scanning Real Objects

- Many 3D scanners are available in the market. Price depends on the resolution of the scan. Examples are
  - Minolta Vivid laser scanner & 3dMD face scanner
  - Microsoft Kinnect ($200), PrimeSense, Realsense etc

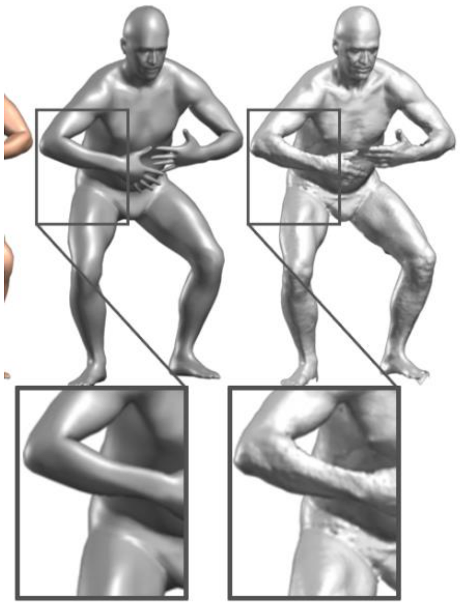
- To cover complete 360^o, we must scan the object from multiple directions and then stich them together



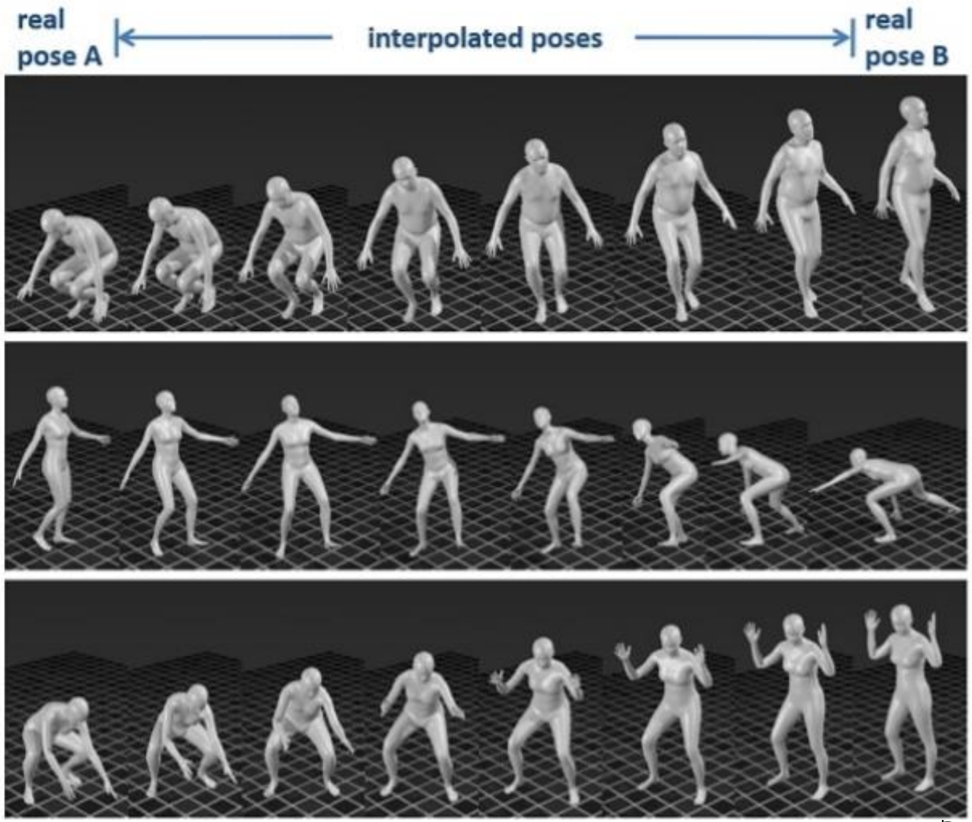
- Real scans need a lot of post-processing to remove noise, spikes and cover holes

Non-Rigid Deformation of Models

- Deformable models are formed by curves or surfaces defined that can be deformed by changing parameter.
- By changing the parameters of the deformable model, we get different 3D shapes



registered mesh 3D scan



SMPL: a skinned multi-person linear model: [link]()

Source: J. Liu et al. "*Temporally Coherent Full 3D Mesh Human Pose Recovery from Monocular Video*", arXiv, 2019

Computer Generated Models

- Scanning real objects or making deformable models are beyond the scope of this unit

- We will look into how to generate 3D models using computer software

Computer Generated Models

- Blender
  - Blender includes many different tools useful for different kinds of modelling.
  - You can import real animations (motion capture) into Blender to animate a model

We’ll focus only on a couple of fundamental techniques: subdivision surfaces and animation via skinning.

How can we easily model in 3D?

3D modelling can be tedious and time consuming.

– Even positioning a single point in 3D is tricky – Mice and displays are 2D devices

– OpenGL (and DirectX) is based mostly on drawing many triangles.

– So objects must be constructed from many vertices, edges and faces,

– Placing each vertex/edge/face individually is not usually feasible!

– How can we do this quickly and easily?

Can we easily model natural shapes?

We can quickly model “blocky” objects – with only a few faces.

– But most natural shapes aren’t blocky.

We can use prebuilt common shapes like spheres, cylinders, ellipsoids, ...

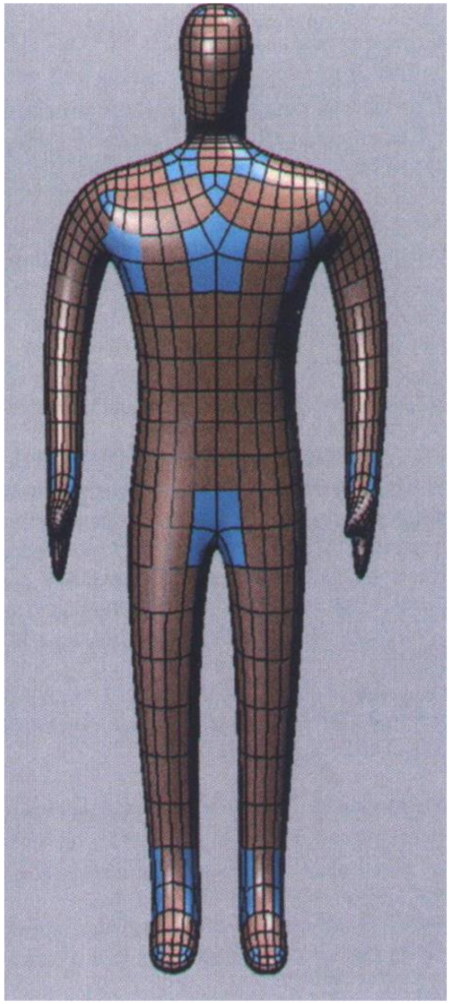
– But these still don’t allow us to create “natural” shapes – most shapes in the real world aren’t perfect spheres, etc.

– **Can we generate shapes with many vertices by controlling just a few?**

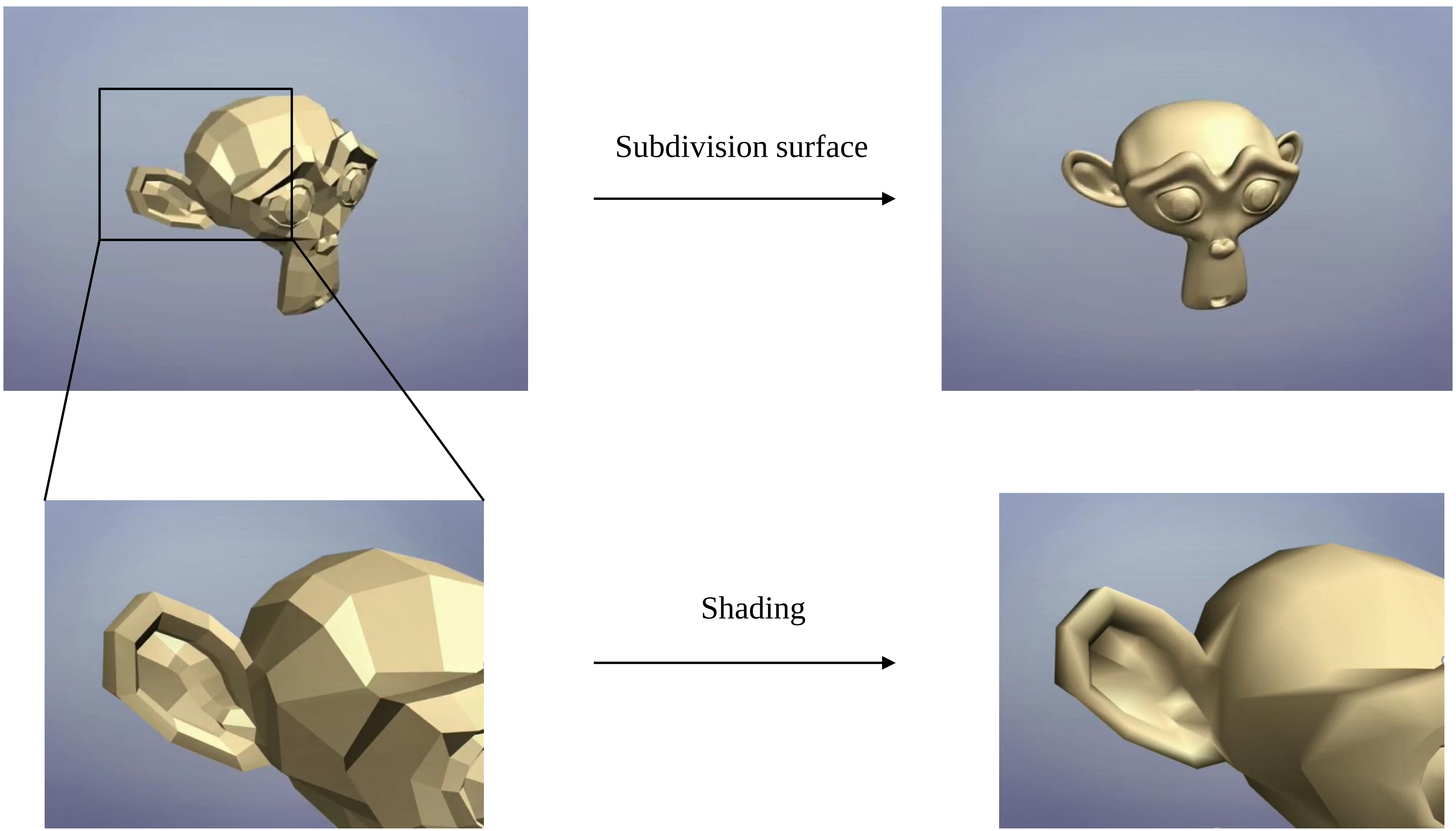
Subdivision Surface Method

Subdivision surface method is a method for producing **smooth surfaces** that can be adjusted easily.

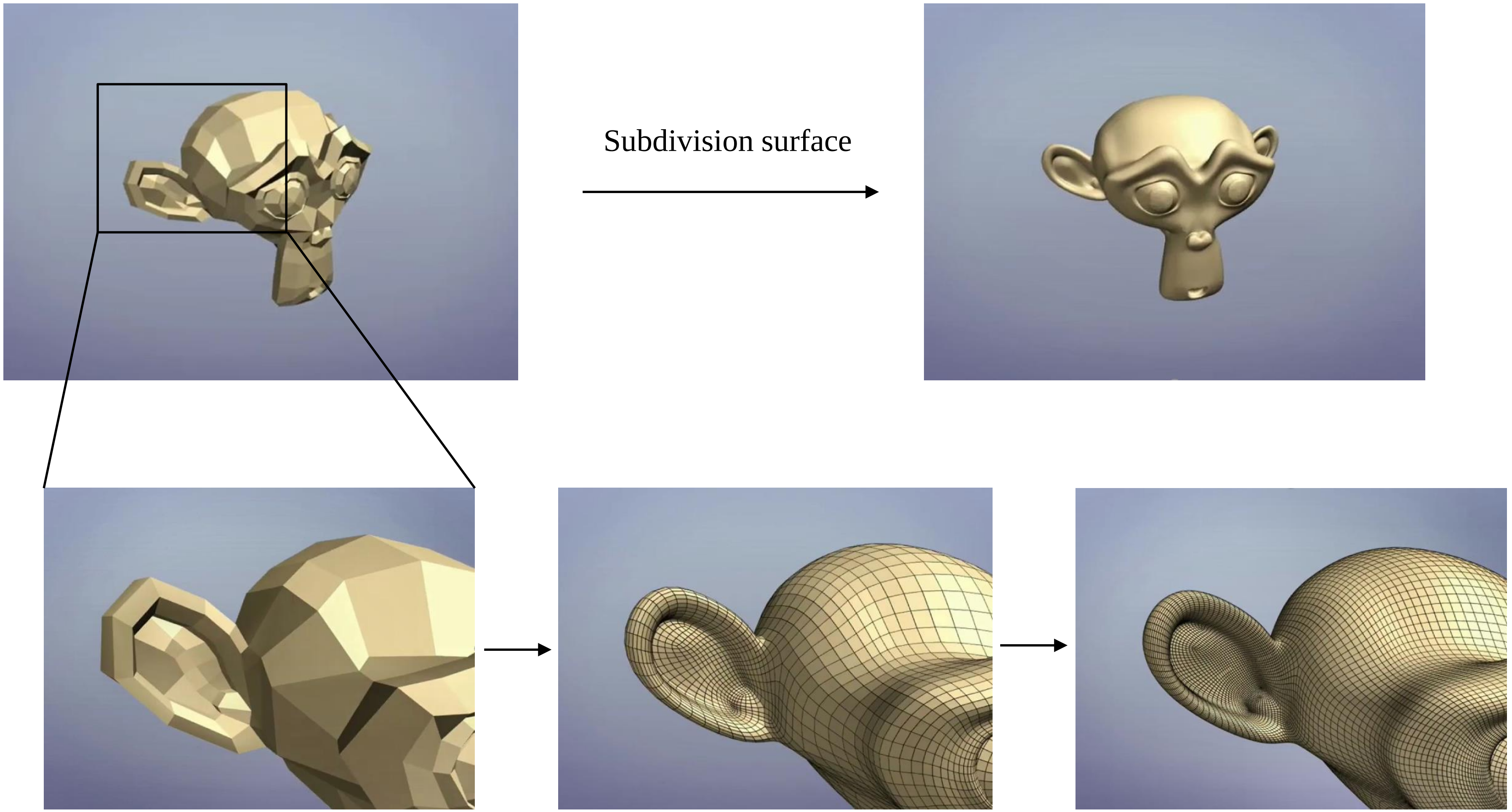
- The idea is to specify a blocky surface, with a manageable number of faces and to calculate a smooth surface that roughly follows it.
- The smoothing process needs to be predictable.



Subdivision Surface Method



Subdivision Surface Method

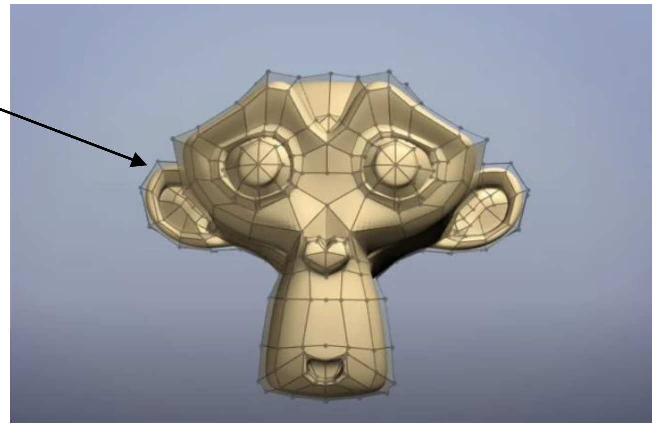


Subdivision Surface Method

Subdivision Surface Method

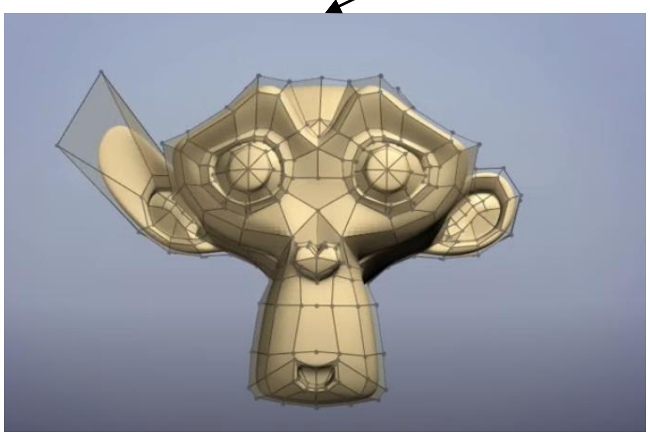
Control cage

Control points



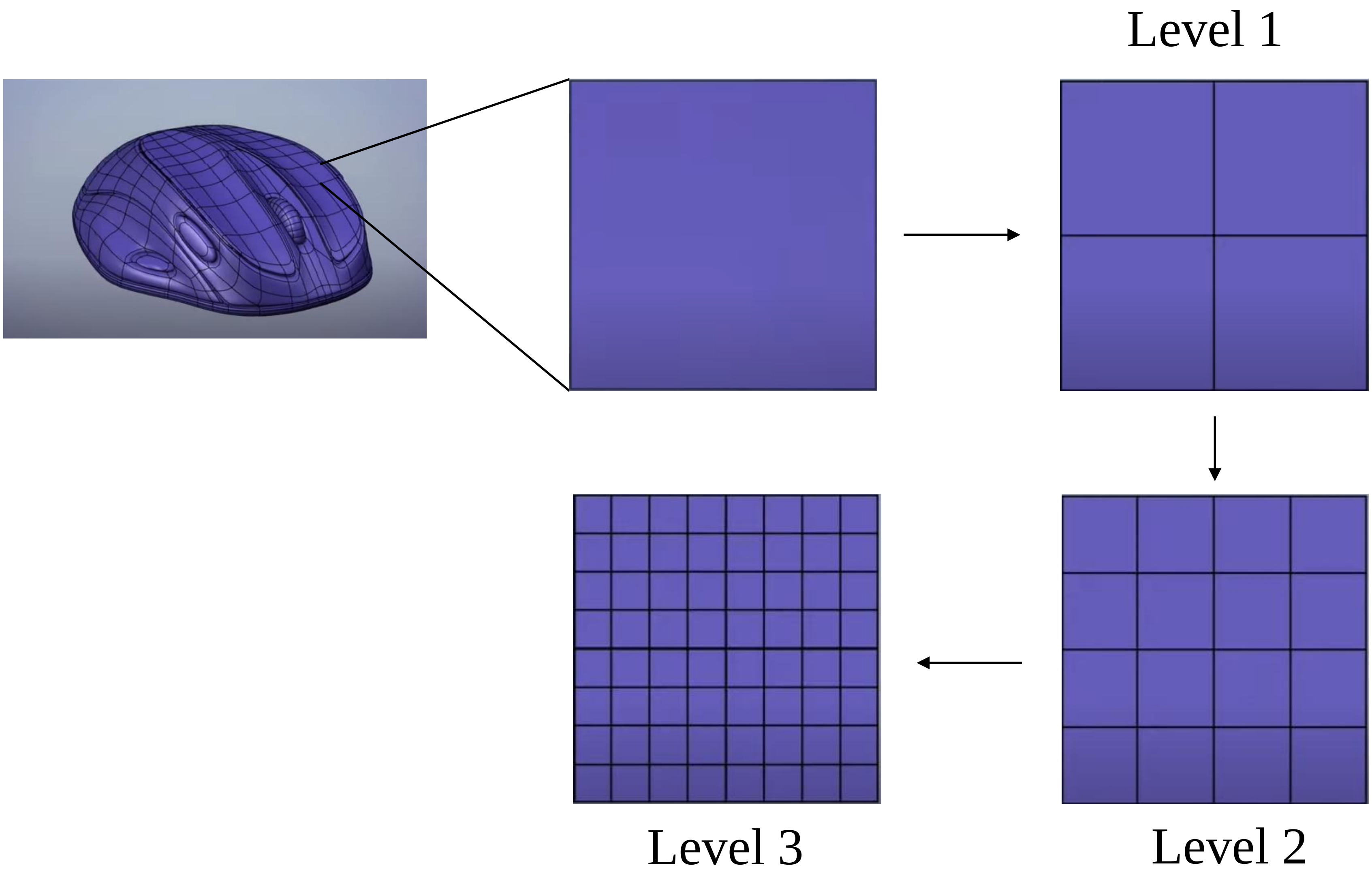
- The basic idea is to start with a single control mesh.
- This mesh can represent a closed surface
- We then use a set of rules to refine the mesh, resulting in a higher resolution control mesh.

Foundations of 3D Computer Graphics
S.J. Gortler

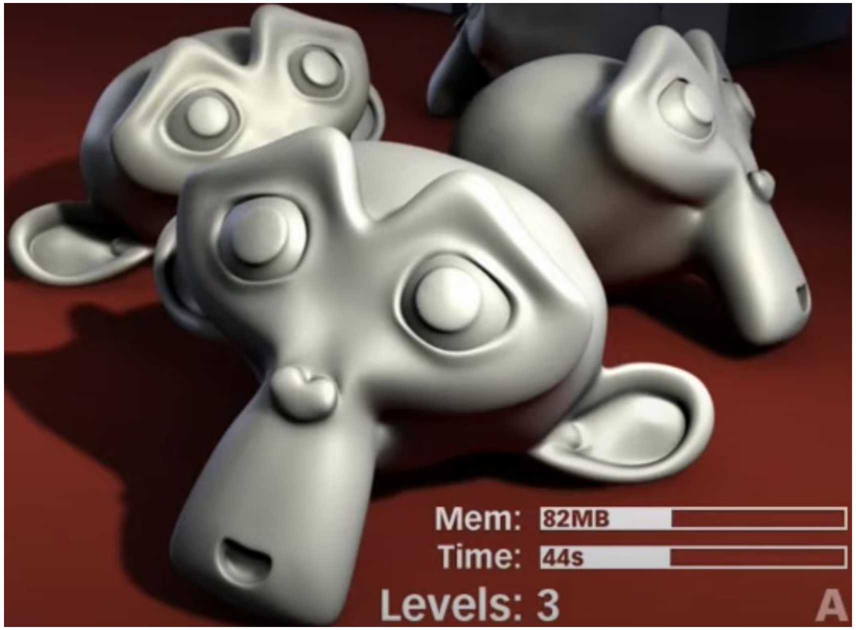


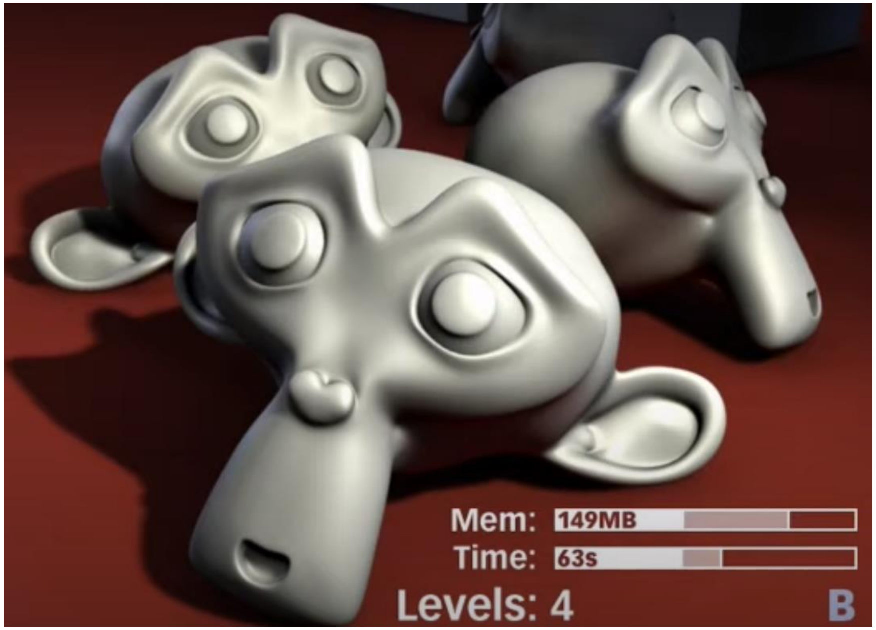
Subdivision Surface Method

Subdivision Surface Method



Subdivision Surface Method





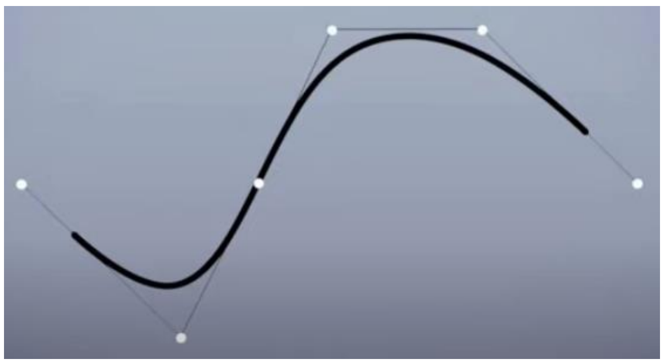
Subdivision Surface Method

- Using smooth shading with subdivision surface can help in smoother surface (low memory and time consumption)

Splines

One can typically design surfaces using splines in a geometric modelling package

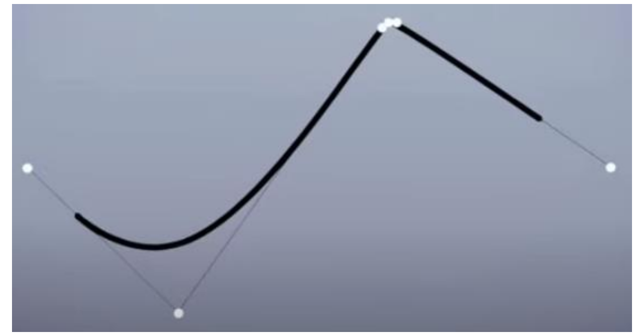
Spline

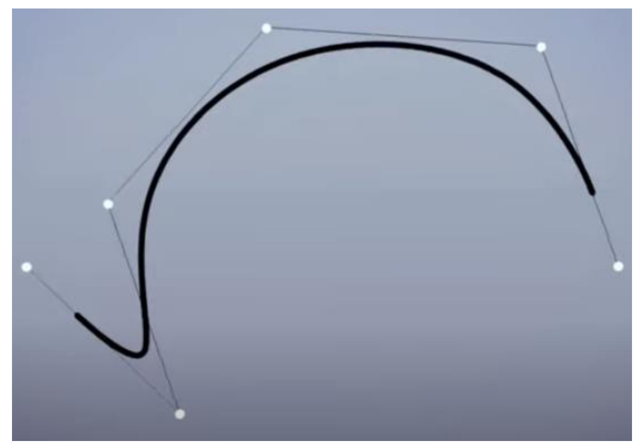


"Spline is a flexible strip used to produce a smooth curve or a curve that is defined by two or more points"

"We can specify a spline curve by giving a set of control points, which indicate general shape of the curve"

Sharp corner





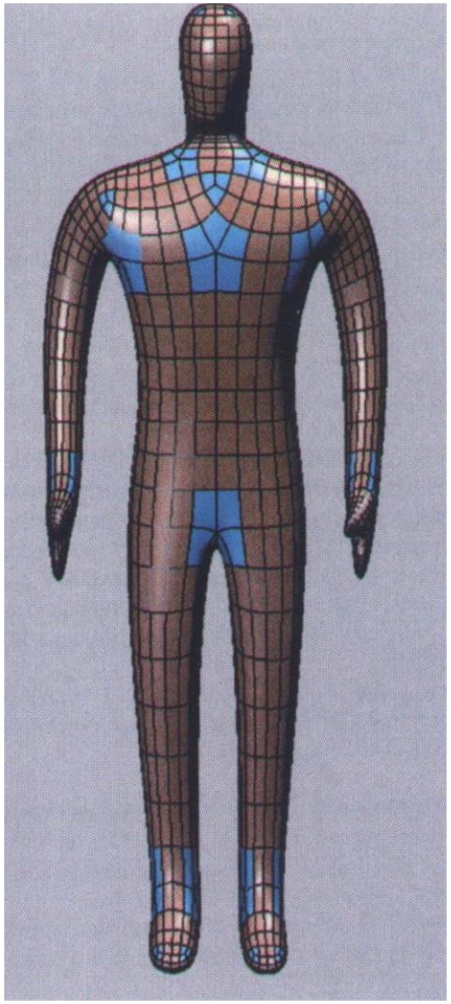
Smooth corner

Book on Splines: RAMSHAW, L. *Blossoming: A connect-the-dots approach to splines*. Digital Systems Research Center, 1987.

Subdivision Surface Method

- ***Subdivision surface*** is related to earlier techniques, like **NURBS** (**Non-Uniform Rational B-Splines**) which also use a small number of control points.
  - Subdivision surfaces is better for 3D modelling because it doesn't have as strict requirements, such as the points forming a grid of quadrilaterals.
  - It is also useful to be able to edit the mesh at the different levels of subdivision, which isn't possible with NURBS and similar techniques.

- Subdivision surfaces have proven to be simple, useful, and general

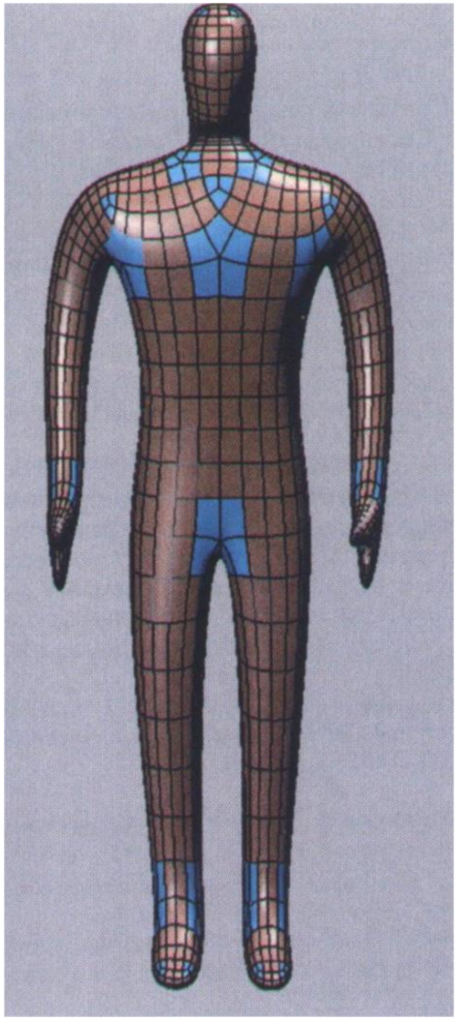


You can read more: https://strategy.asee.org/concept-modeling-with-nurbs-polygon-and-subdivision-surfaces.pdf

Catmull-Clark Subdivision Surface Technique

Catmull-Clark subdivision surface technique is often the preferred technique for generating smooth surfaces from a "control mesh" with a relatively small number of points, because **it is simple, predictable** and has **desirable properties** such as**:**

- **Each original point affects only a small part of the surface** – roughly up to each neighbour.
- **The 1-st derivative is always continuous** – i.e., the normals never change suddenly.
- **The 2-nd derivative is nearly always continuous**, i.e., the curvature (rate of change of the normals) doesn't change suddenly. The exception is at extraordinary vertices – where the mesh is "irregular", i.e., not a grid of quadrilaterals



Subdivision Surfaces: technical details

[From Catmull & Clark]

o = old vertices (p_{ij})

x = new vertices (q_{ij})

After one subdivision step, there is a new vertex for:

- Each old face
- Each old edge
- Each old vertex

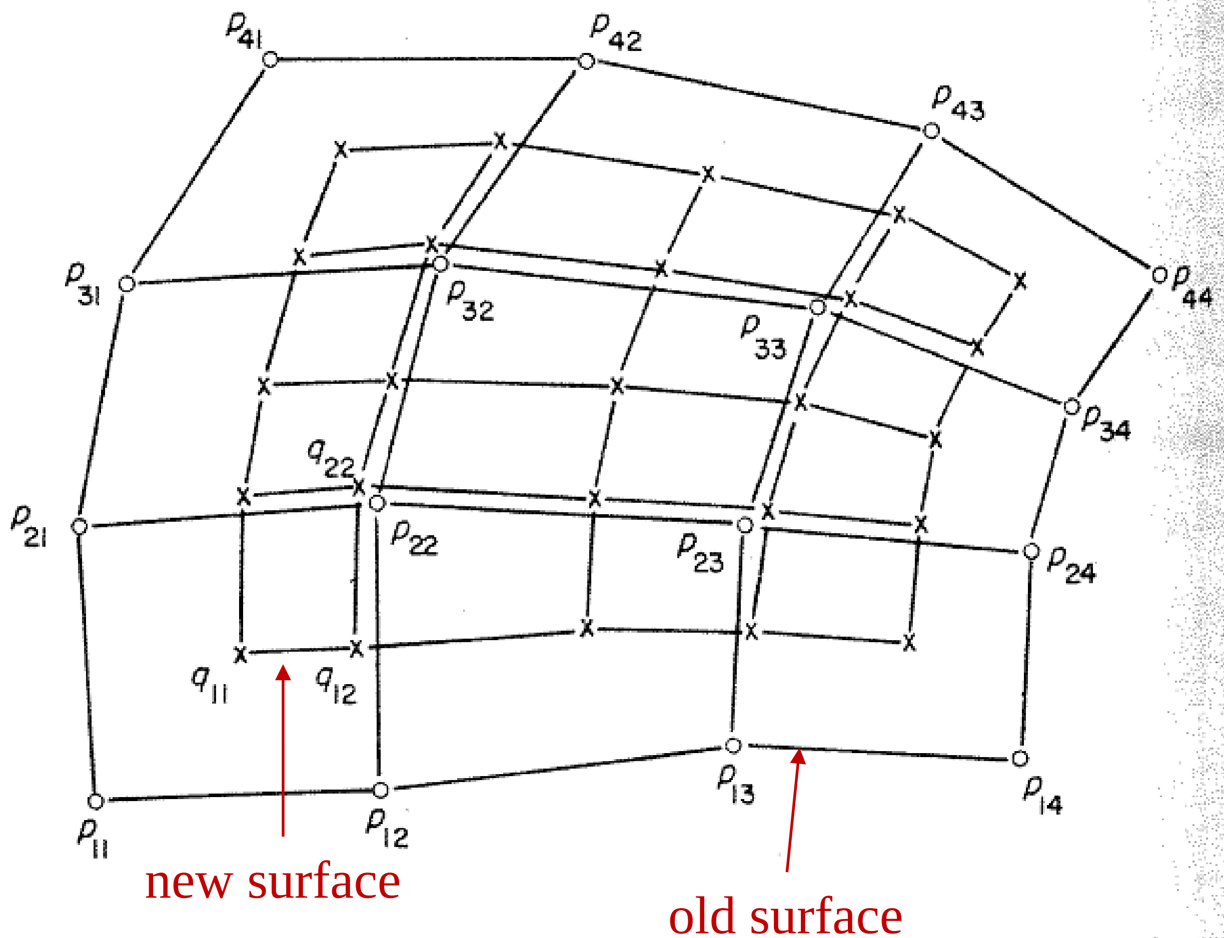


Figure 1. Standard bicubic B-spline patch on a rectangular control-point mesh

Subdivision Surfaces: technical details

[From Catmull & Clark]

o = old vertices (p_{ij})

x = new vertices (q_{ij})

After one subdivision step, there is a new vertex for:

- Each old face
- Each old edge
- Each old vertex

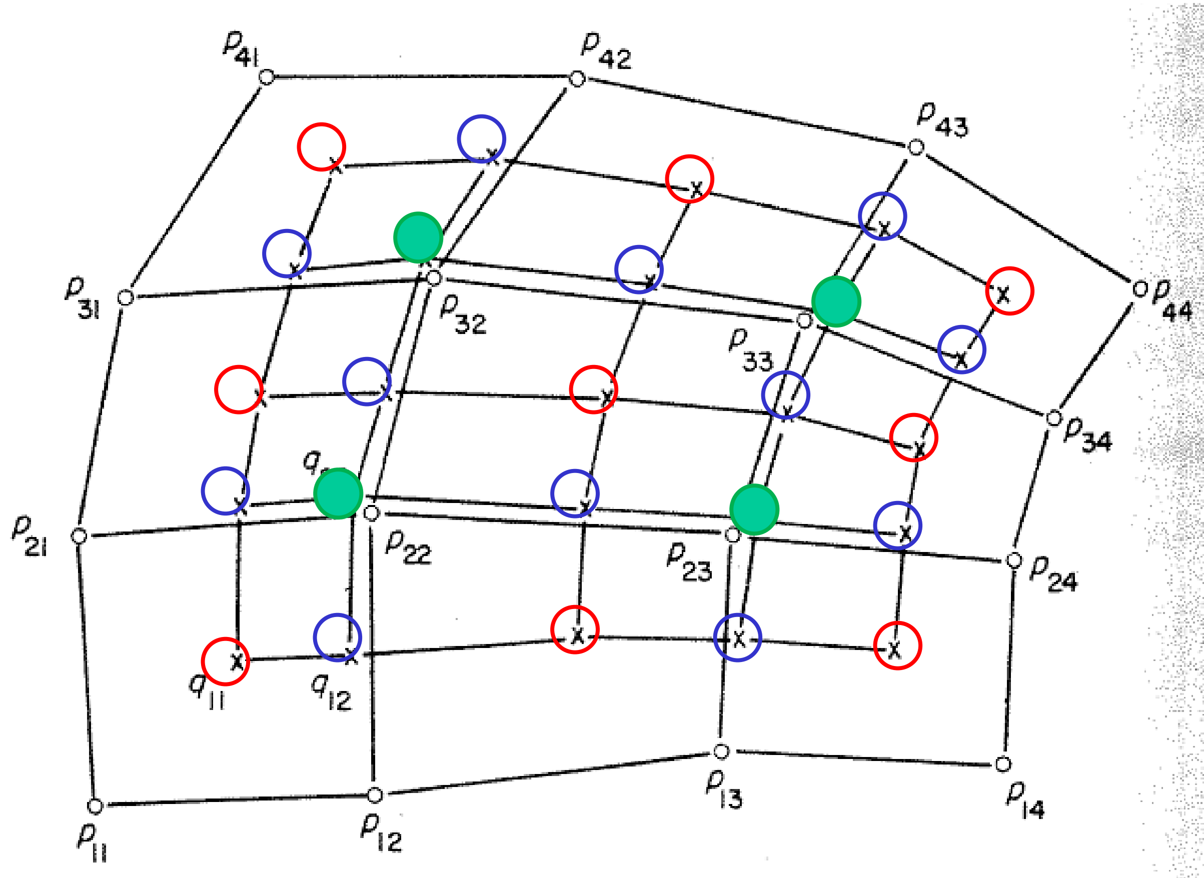


Figure 1. Standard bicubic B-spline patch on a rectangular control-point mesh

Subdivision Surfaces: technical details

[From Catmull & Clark]

There is a new vertex for:

- Each old face

On the old surface, there are 9 faces. So there are 9 new vertices marked as ○

(e.g. $p_{11}p_{12}p_{22}p_{21}$ is a face on the old surface)

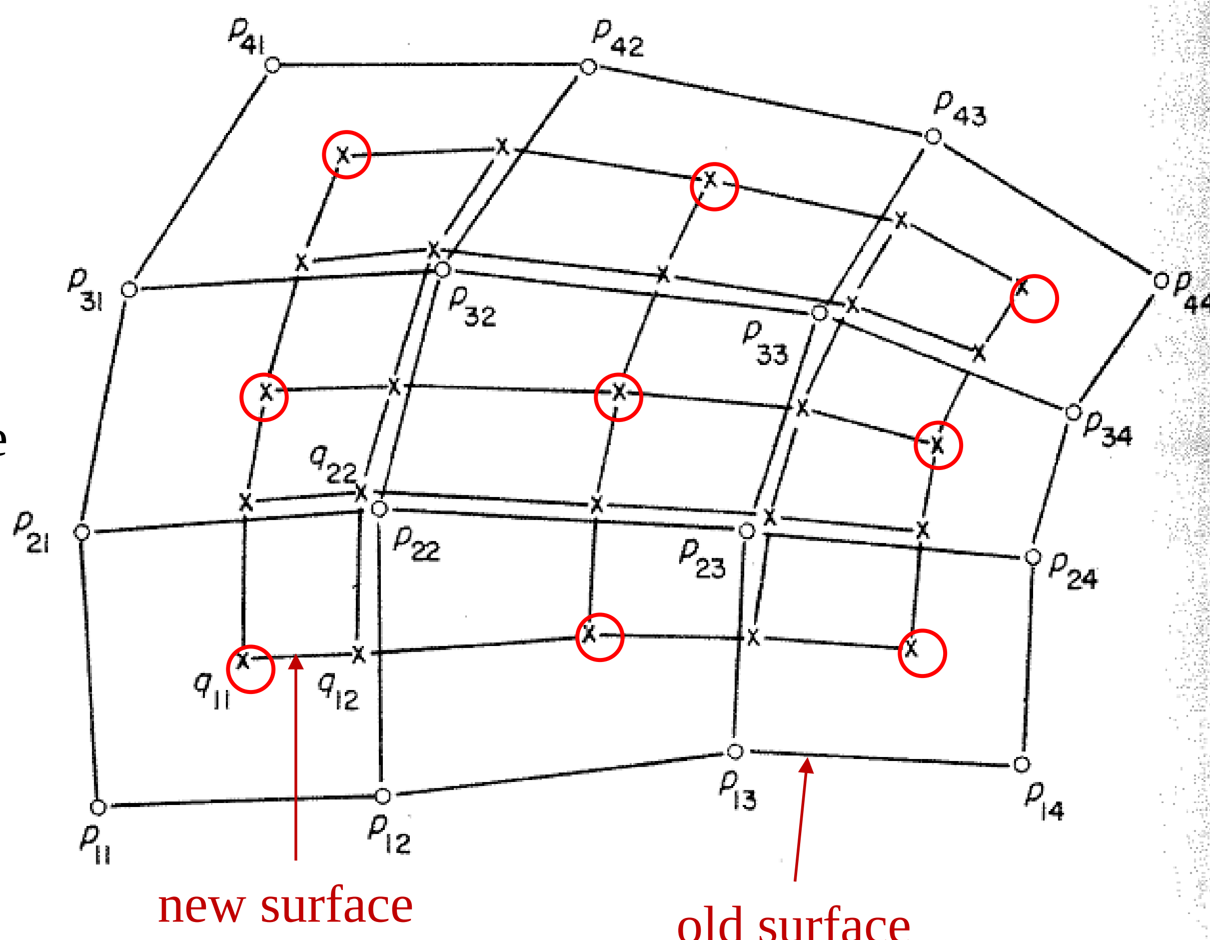


Figure 1. Standard bicubic B-spline patch on a rectangular control-point mesh

Subdivision Surfaces: technical details

[From Catmull & Clark]

There is a new vertex for:

- Each old edge

On the old surface, there are 12 edges. So there are 12 new vertices marked as

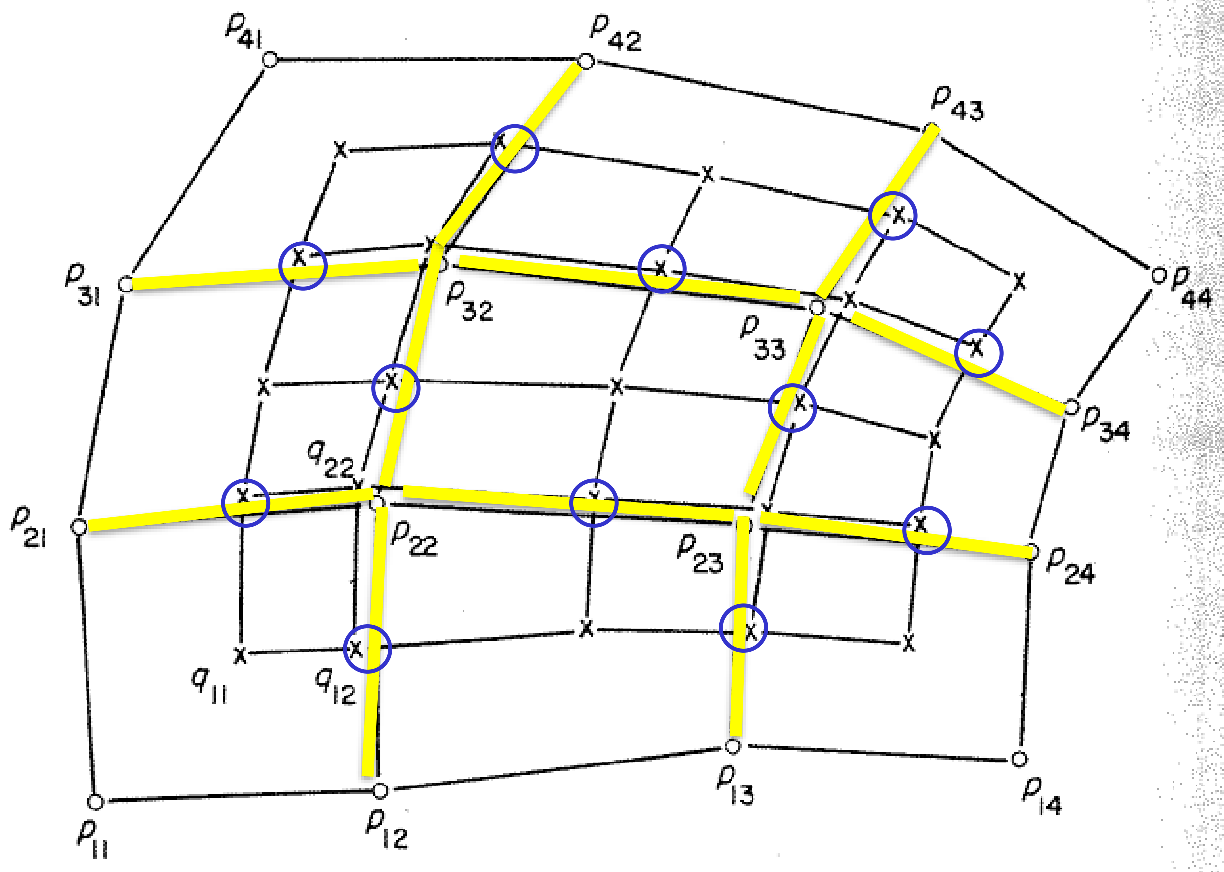


Figure 1. Standard bicubic B-spline patch on a rectangular control-point mesh

Subdivision Surfaces: technical details

[From Catmull & Clark]

There is a new vertex for:

- Each old vertex

On the old surface, there are 4 vertices: $p_{22}, p_{23}, \ p_{33}, p_{32}$. So there are 4 new vertices marked as

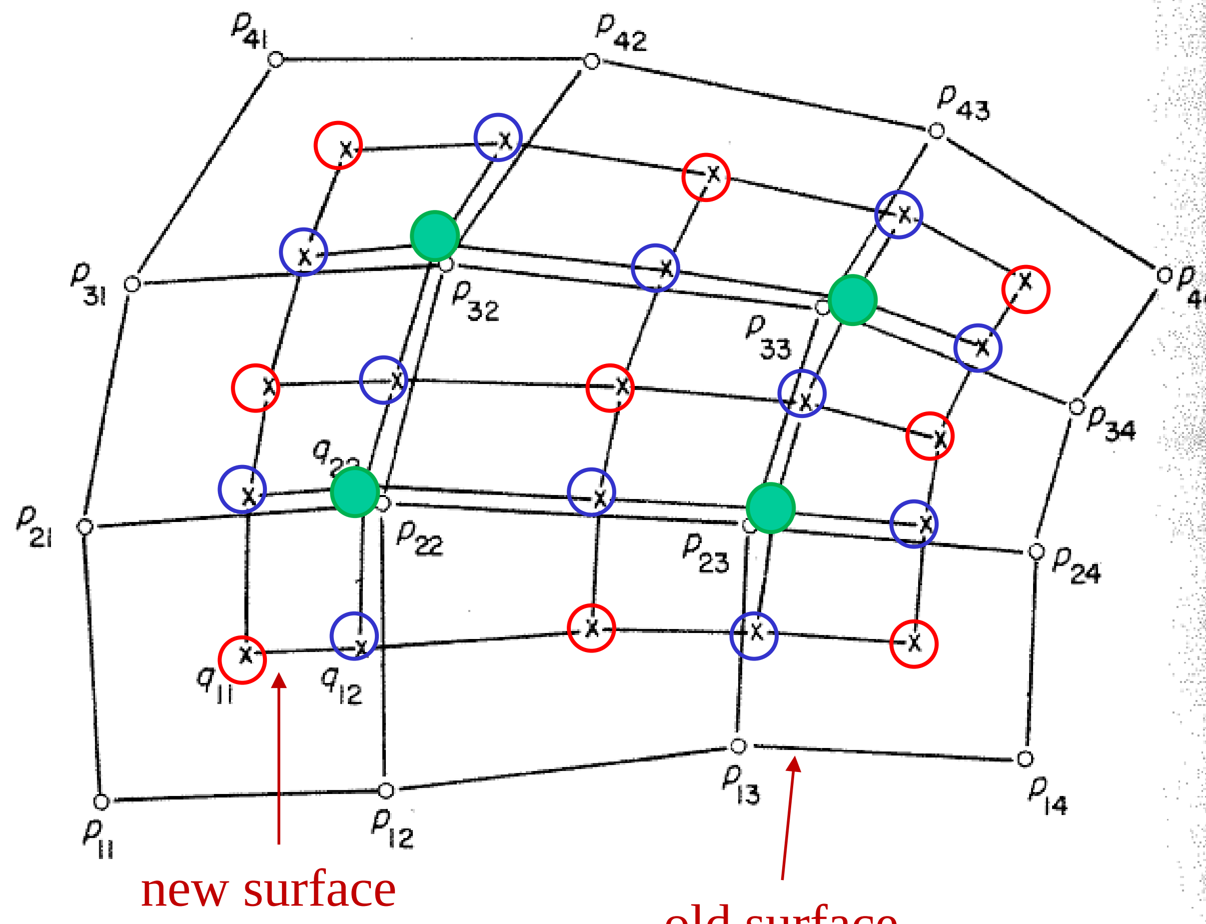


Figure 1. Standard bicubic B-spline patch on a rectangular control-point mesh

Subdivision Surfaces: technical details

[From Catmull & Clark]

o = old vertices (p_{ij})

x = new vertices (q_{ij})

After one subdivision, there is a new vertex for:

- Each old face
- Each old edge
- Each old vertex

So, in total, the new surface has 9+12+4=25 vertices

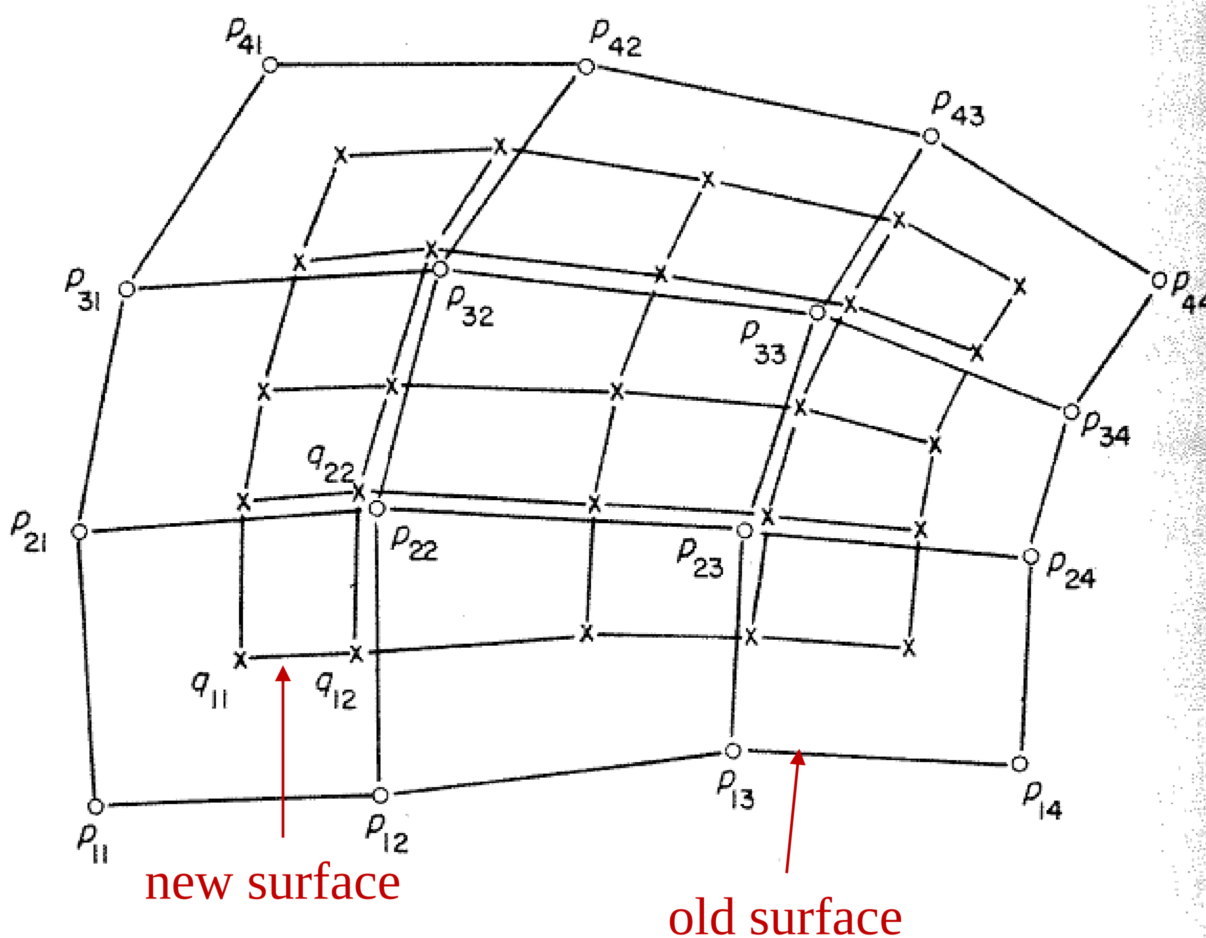


Figure 1. Standard bicubic B-spline patch on a rectangular control-point mesh

Subdivision Surfaces: technical details

Let's refer to the new vertices as points and the old vertices as vertices

[From Catmull & Clark]

- New "face" points are at the average of the vertices for the face
- New "edge" points are at the average of the two vertices on the edge and the two face points on either side of the edge
- The placement of new "vertex" points is a bit complicated (see next slide)

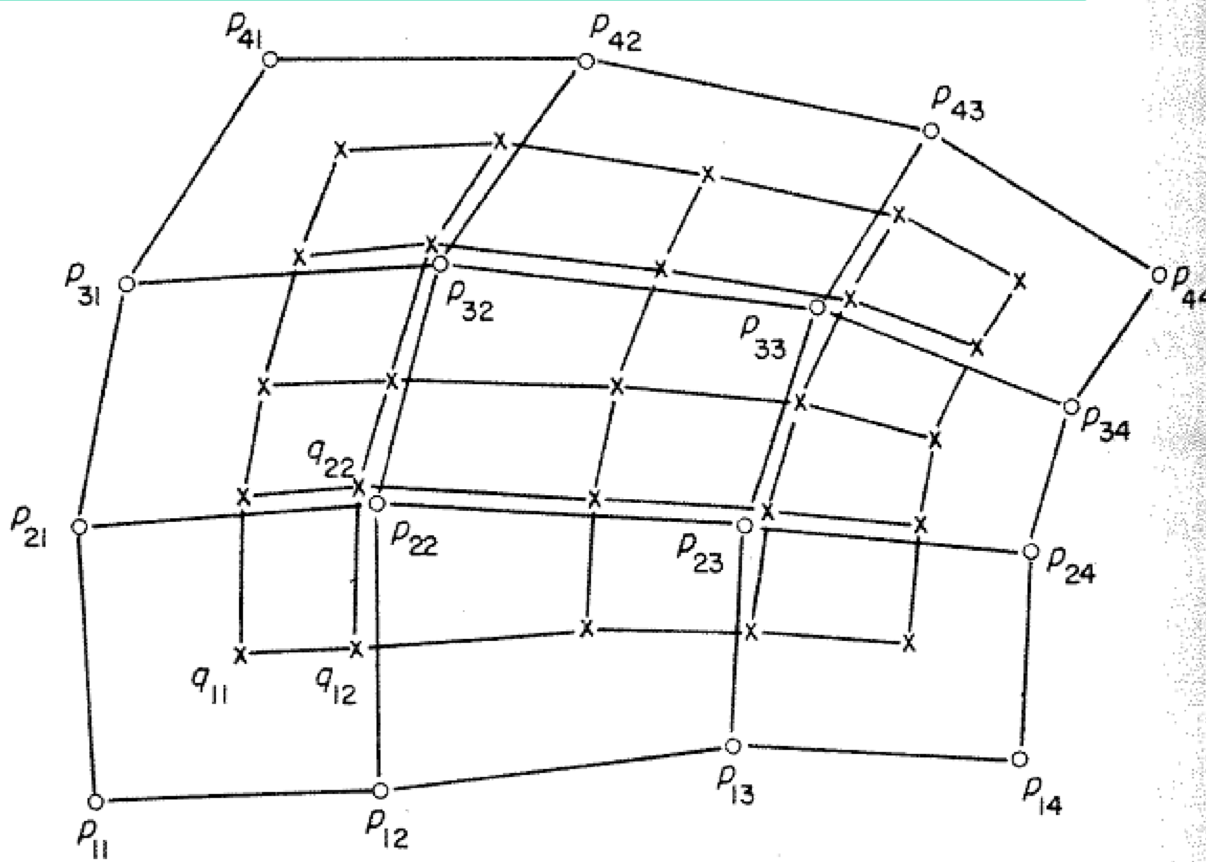


Figure 1. Standard bicubic B-spline patch on a rectangular control-point mesh

Subdivision Surfaces: technical details

Let's refer to the new vertices as points and the old vertices as vertices

[From Catmull & Clark]

For the vertex P, a new point is placed at

$$\frac{F + 2E + (n-3)P}{n}$$

Where F is the average of the face points, E is the average of the edge points and n is the number of edges

The faces and edges are the original ones that touch the original P

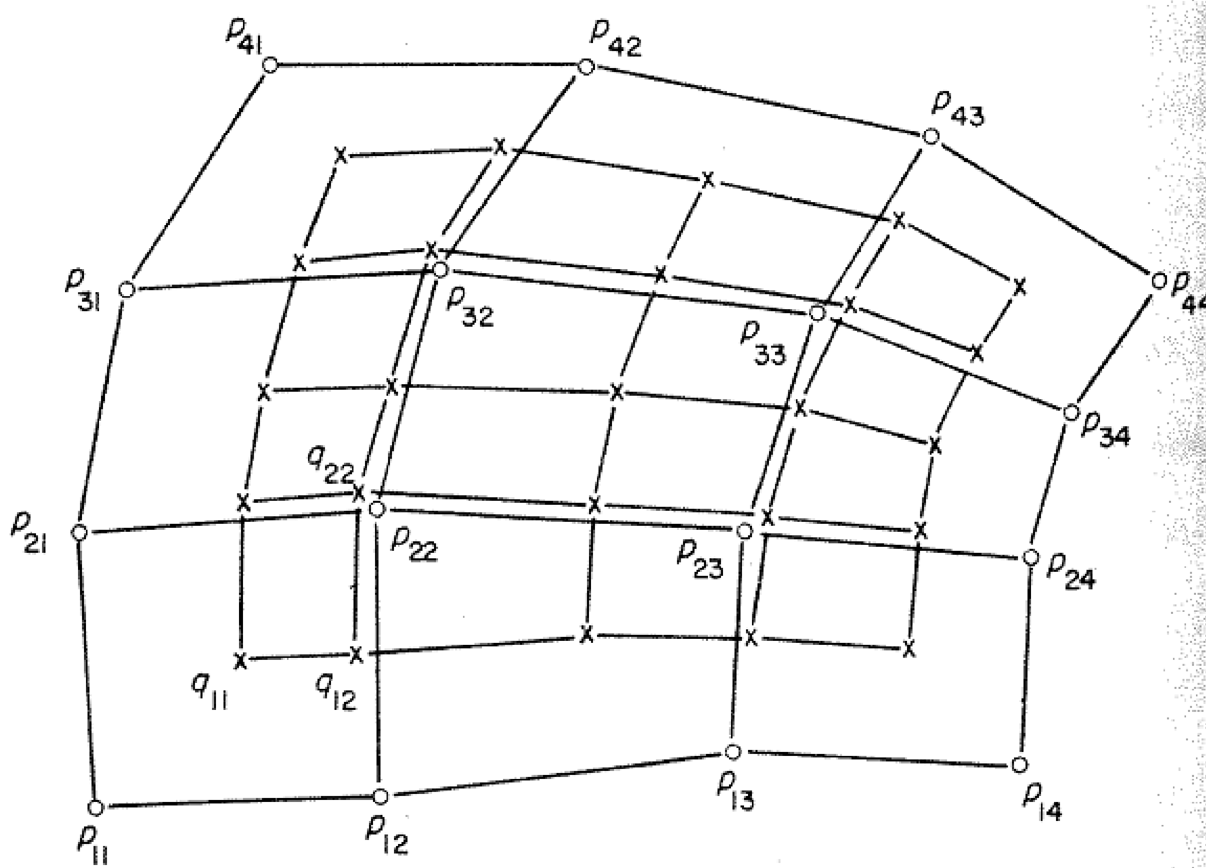


Figure 1. Standard bicubic B-spline patch on a rectangular control-point mesh

Subdivision Surfaces: technical details

Other important properties of *Catmull & Clark* subdivision:

- When the control points form a simple grid topology (as in Figure 1) the surface tends towards a bicubic B-Spline, a standard kind of surface used when smoothness is required.
- Unlike other techniques for generating such surfaces (like **NURBS**), the technique naturally extends to other topologies, giving 3D modellers much freedom.
- Properties like texture coordinates can be smoothly generated in a similar way to the vertex positions: by averaging them with the same weights during subdivision.

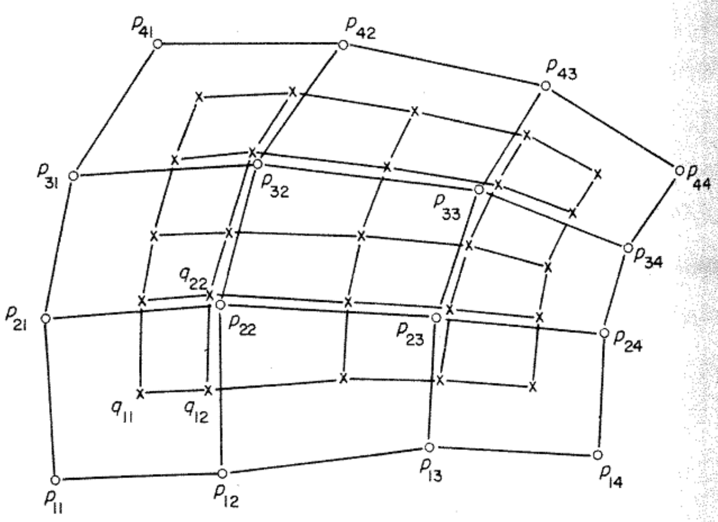


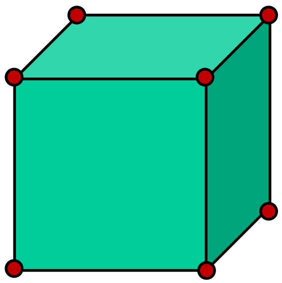
Figure 1. Standard bicubic B-spline patch on a rectangular control-point mesh

Catmull & Clark subdivision surface method on closed surfaces

- Counting the number of new vertices for open surfaces after one subdivision step can be a bit confusing. For closed surfaces, the counting is easier and more intuitive.

Catmull & Clark subdivision surface method on closed surfaces

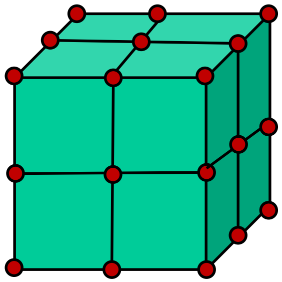
For a cube, initially



There are 8 vertices, 6 faces, and 12 edges.

$V_0 = 8$
$F_0 = 6$
$E_0 = 12$

- After one subdivision step, how many vertices are there?



There are 26 vertices, 24 faces, and 48 edges.

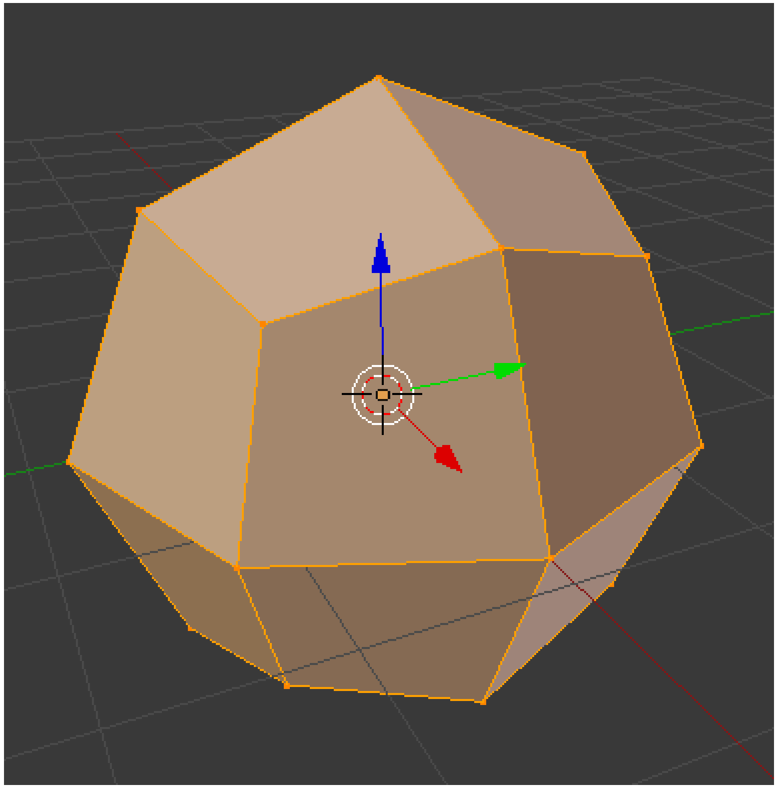
$V_1 = V_0 + F_0 + E_0 = 26$
$F_1 = 4F_0 = 24$
$E_1 = 2E_0 + 4F_0 = 24 + 24 = 48$

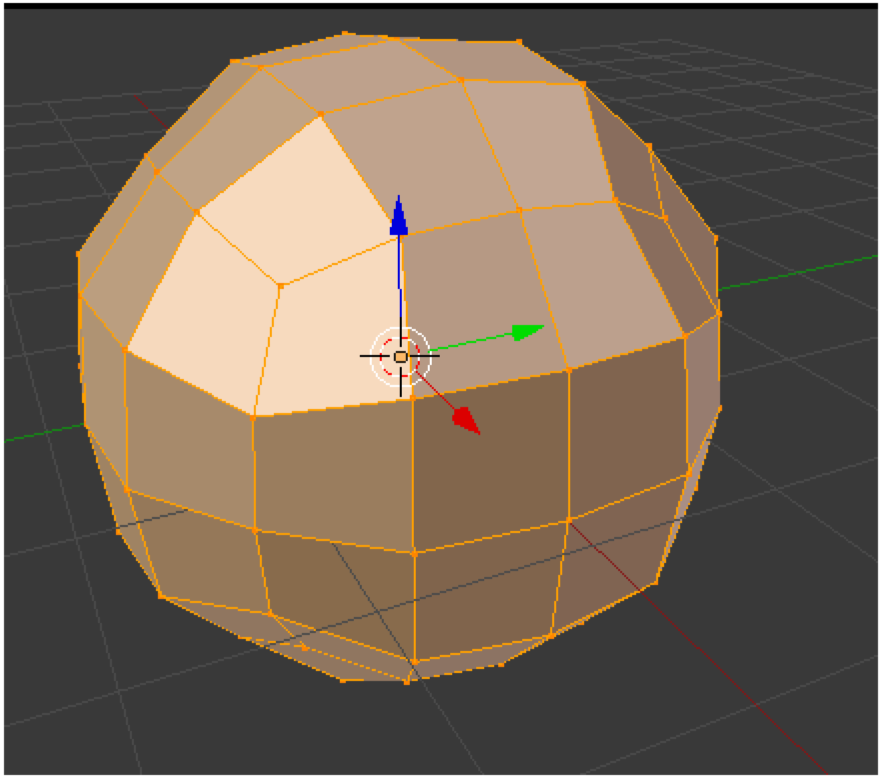
- In general, after n subdivision steps,

$$V_n = V_{n-1} + F_{n-1} + E_{n-1}$$
$$F_n = 4F_{n-1}$$
$$E_n = 2E_{n-1} + 4F_{n-1}$$

Catmull & Clark subdivision surface method on closed surfaces

As the Catmull & Clark subdivision surface method constrains the surface to be smooth, the cube would approach the shape of a **sphere** after a few subdivisions.





Further Reading

- E. Angel and D. Shreiner: Interactive Computer Graphics 6E © Addison-Wesley 2012
  - Catmull-Clark subdivision Ch-10 Section 10.12

  The slides are partially based on “The Guerrila CG Project of guerrilla.org”, topic Subdivision surfaces: Overview.

- Other Resources/ References

  Foundations of 3D Computer Graphics, S.J. Gortler
  - Chapter 22

  https://learnopengl.com/Model-Loading/Assimp