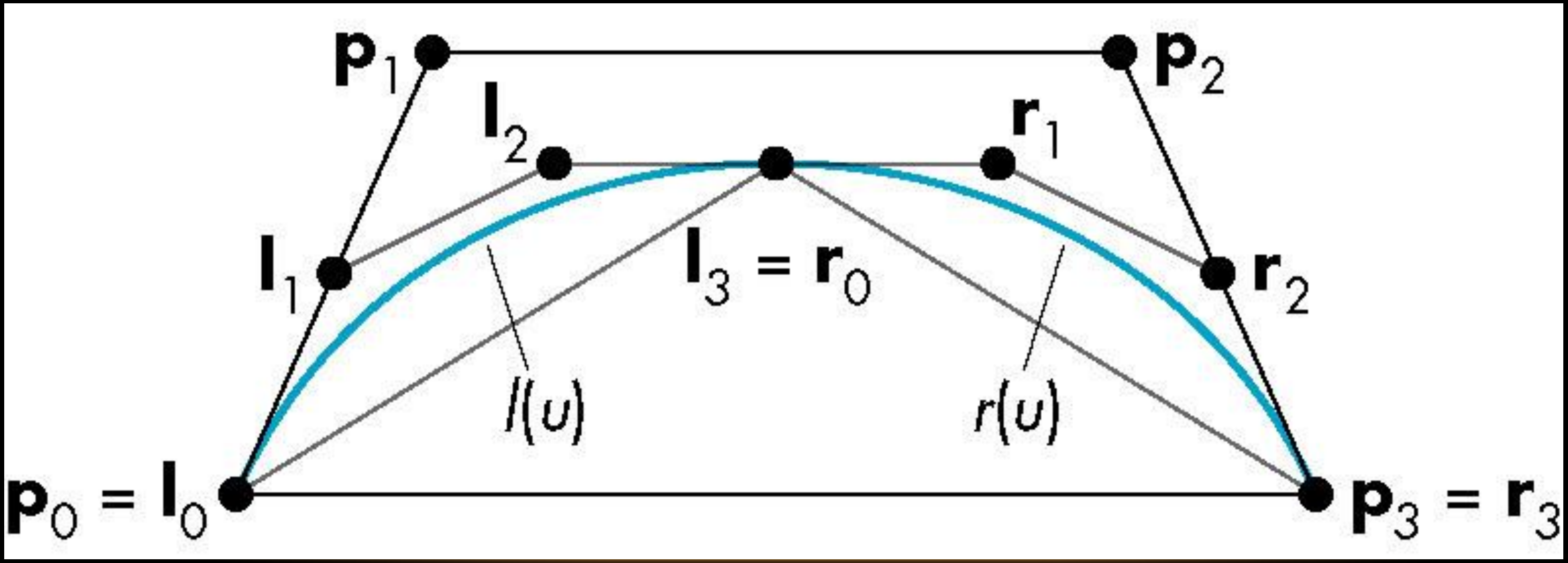
BEZIER SURFACES

OUTLINE

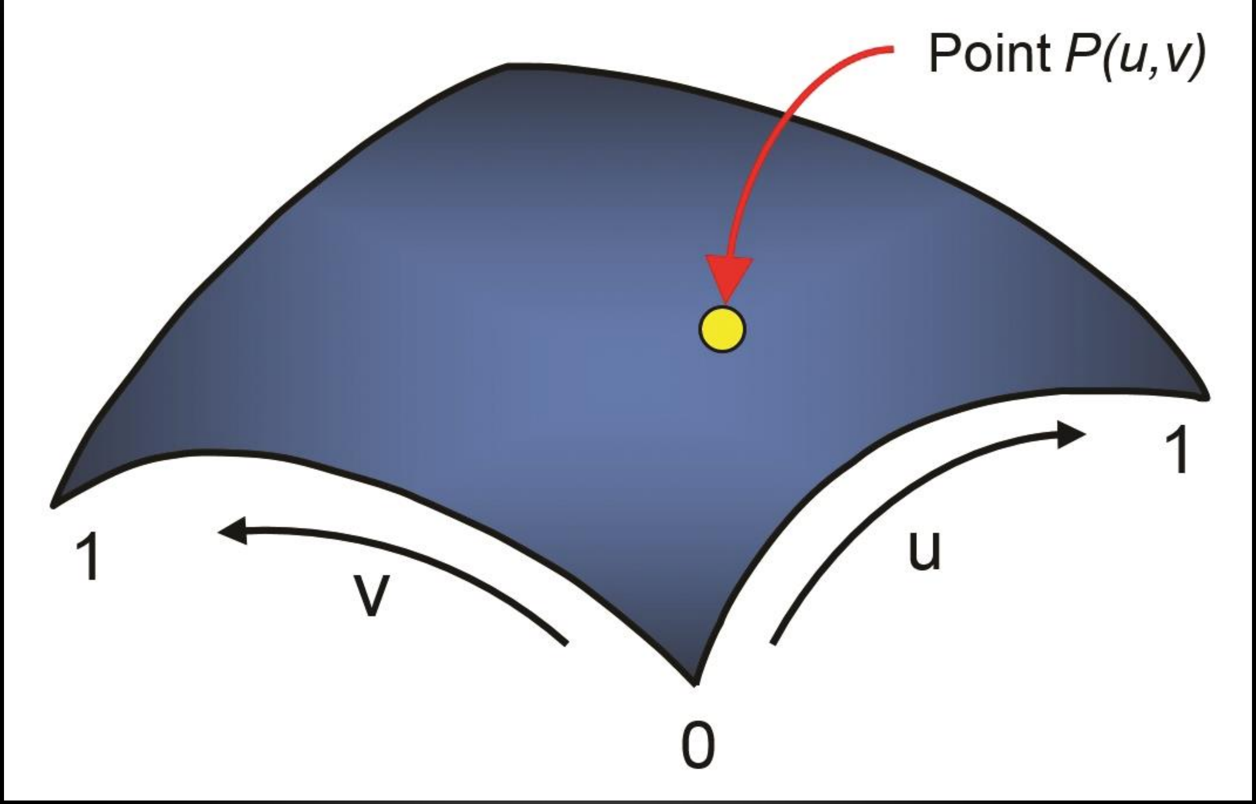
- Quadratic Bezier Surfaces
- Cubic Bezier Surfaces

DE CASTELJAU RECURSION REVISITED

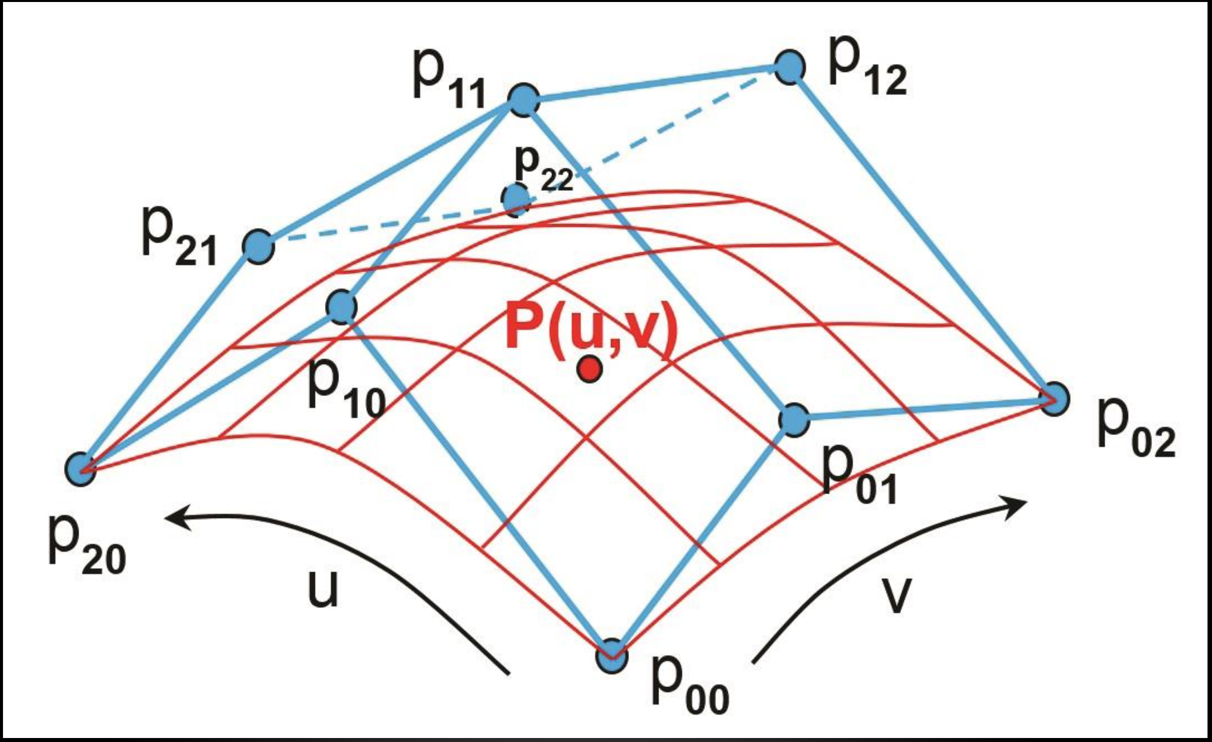
- $l_0 = p_0$, $r_3 = p_3$
- $l_1 = 1/2(p_0 + p_1)$, $r_2 = 1/2(p_2 + p_3)$
- $l_2 = 1/2(l_1 + 1/2(p_1 + p_2))$, $r_1 = 1/2(r_2 + 1/2(p_1 + p_2))$



CURVED SURFACE



CONTROL POINTS AND RESULTING SURFACE



QUADRATIC BLENDING FUNCTIONS

- These are the same as the quadratic Bezier curve blending functions
    - Except that now we use them in two dimensions

$$B_0(u) = (1 - u)^2$$

$$B_1(u) = -2u^2 + 2u$$

$$B_2(u) = u^2$$

$$B_0(v) = (1 - v)^2$$

$$B_1(v) = -2v^2 + 2v$$

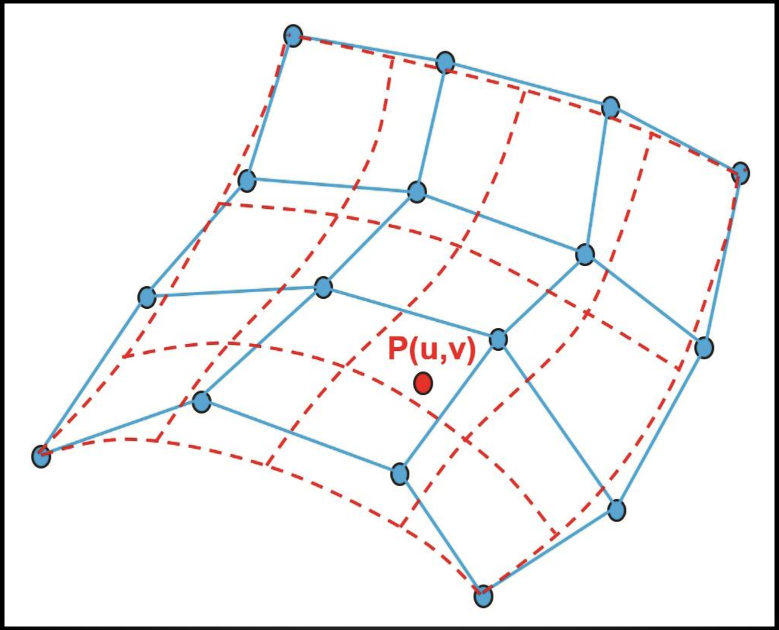
$$B_2(v) = v^2$$

BEZIER PATCHES

- Double summation used for surfaces as opposed to curves
- The set of points generated for the Bezier surface is called a Bezier patch

$$P(u,v) = \sum_{i=0}^{2} \sum_{j=0}^{2} \boldsymbol{p}_{ij} * B_i(u) * B_j(v)$$

CUBIC BEZIER PATCHES



CUBIC BEZIER SURFACES

$$P(u,v) = \sum_{i=0}^{3}\sum_{j=0}^{3} \boldsymbol{p}_{ij} * B_j(u) * B_j(v)$$

CUBIC BEZIER SURFACE BLENDING FUNCTIONS

$$B_0(u) = (1-u)^3$$

$$B_1(u) = 3u^3 - 6u^2 + 3u$$

$$B_2(u) = -3u^3 + 3u^2$$

$$B_3(u) = u^3$$

$$B_0(v) = (1-v)^3$$

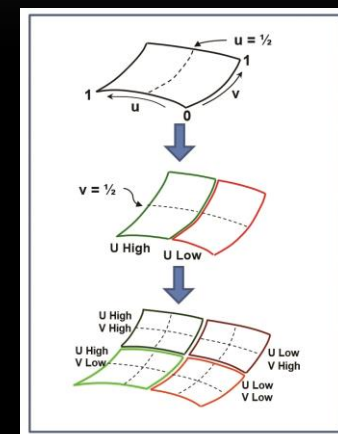
$$B_1(v) = 3v^3 - 6v^2 + 3v$$

$$B_2(v) = -3v^3 + 3v^2$$

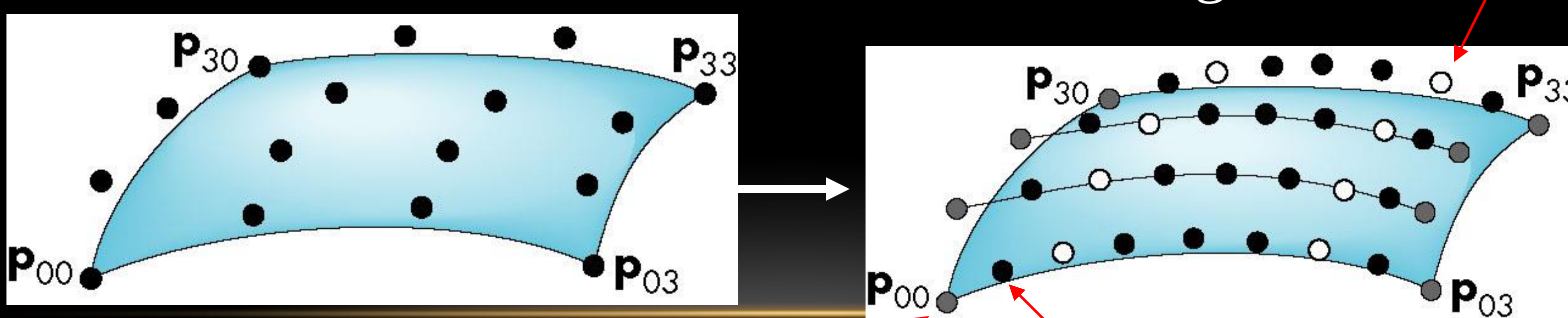
$$B_3(v) = v^3$$

SURFACES

- Can apply the recursive method to surfaces - a Bezier patch curves of constant u (or v) are Bezier curves in u (or v)
- First subdivide in u
  - Process creates new points
  - Some of the original points are discarded



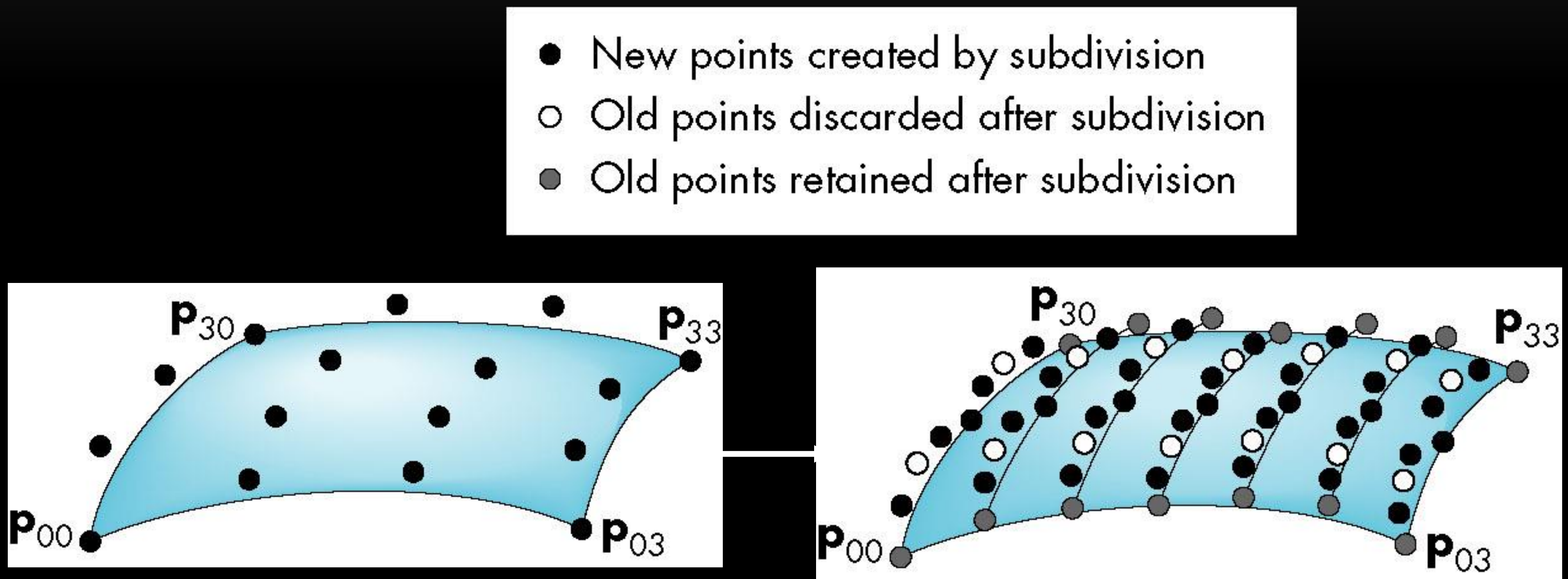
original and discarded



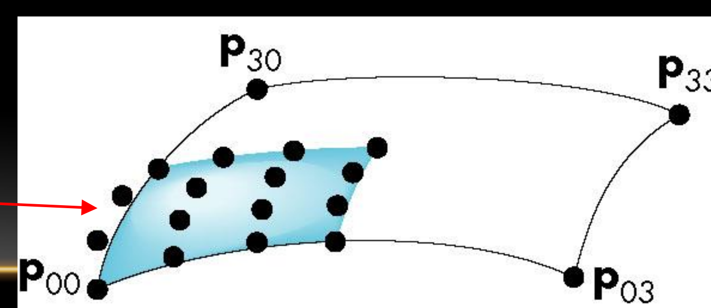
original and kept

new

SECOND SUBDIVISION



16 final points for
1 of 4 patches created



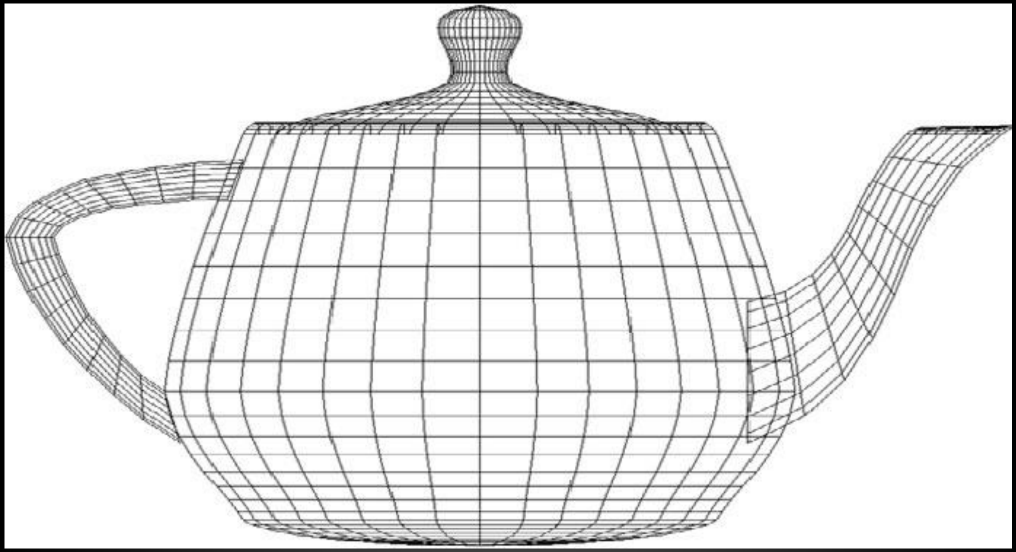
BASE CONDITION

- With Bezier curves, the base condition was whether the curve was “straight” enough
- With surfaces, the base condition is whether the surface is “flat” enough

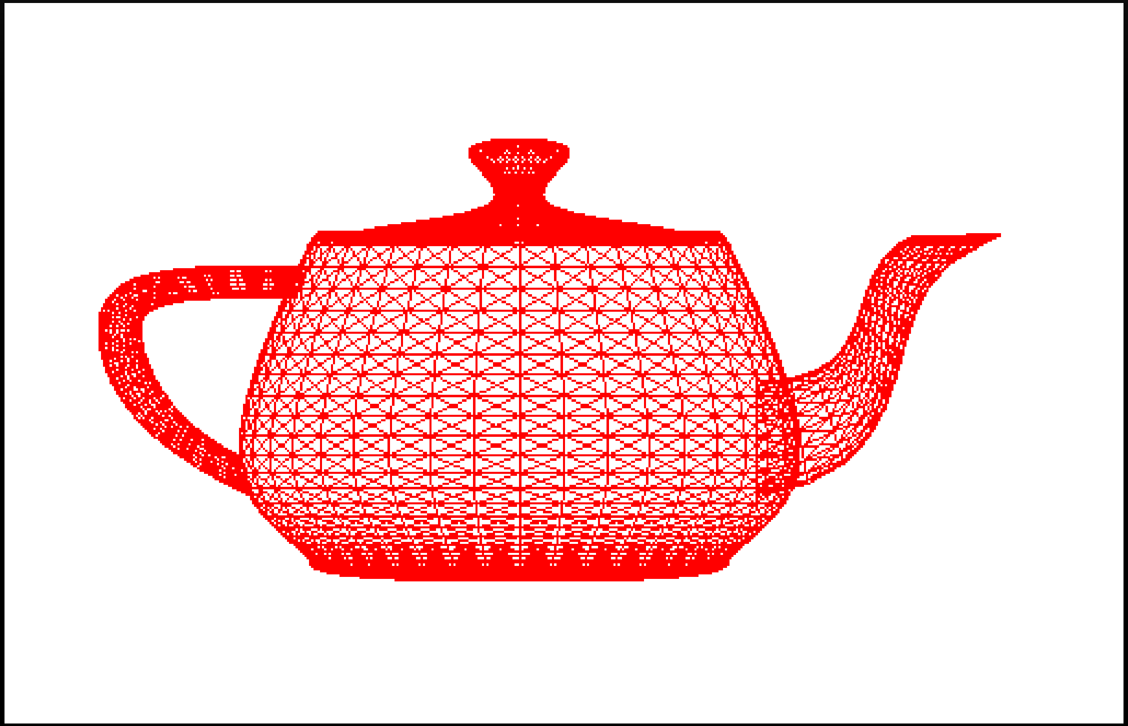
$$d = abs\left(\frac{Ax + By + Cz + D}{\sqrt{A^2 + B^2 + C^2}}\right)$$

UTAH TEAPOT

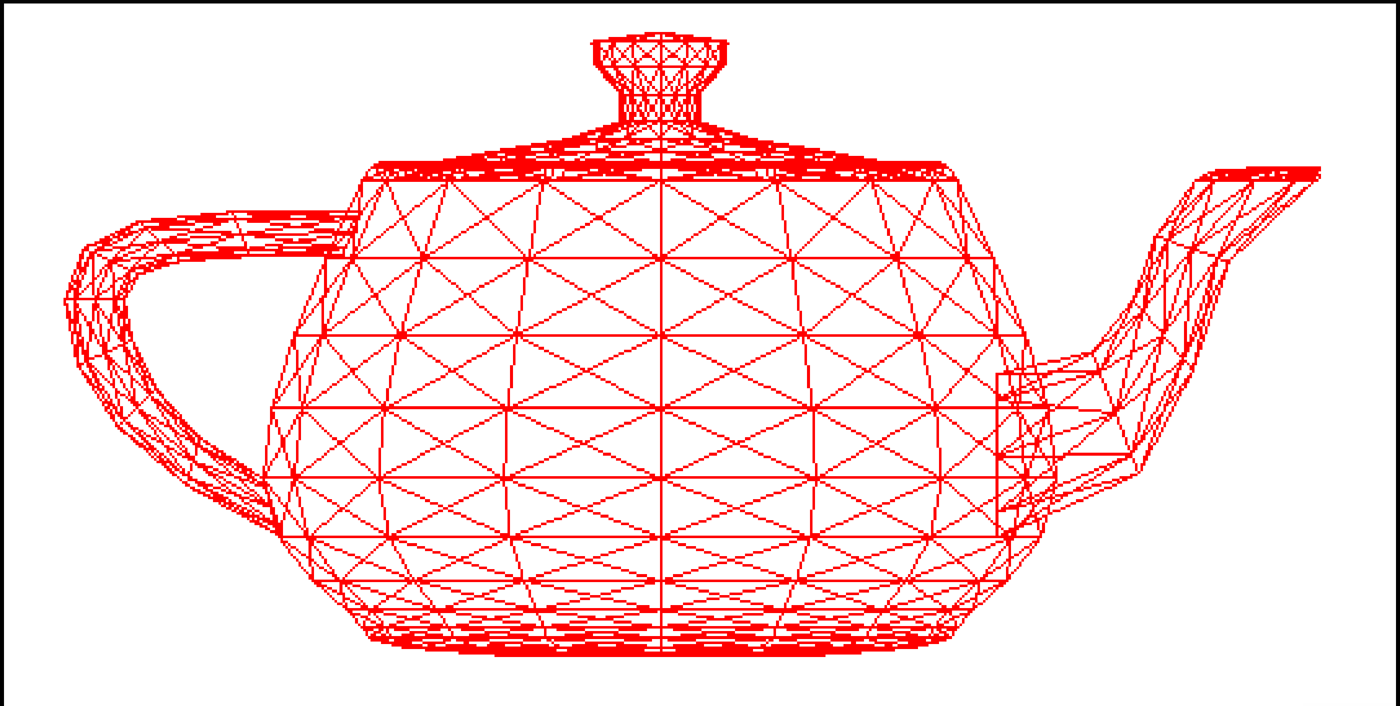
- Most famous data set in computer graphics
- Widely available as a list of 306 3D vertices and the indices that define 32 Bezier patches



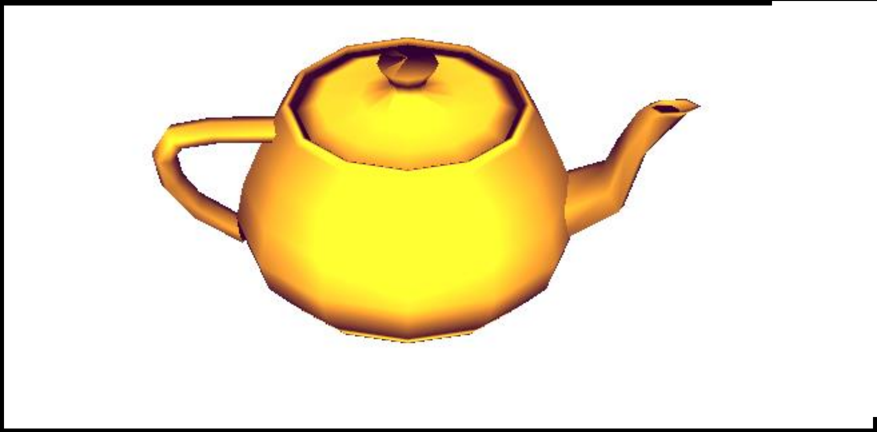
TESSELLATION

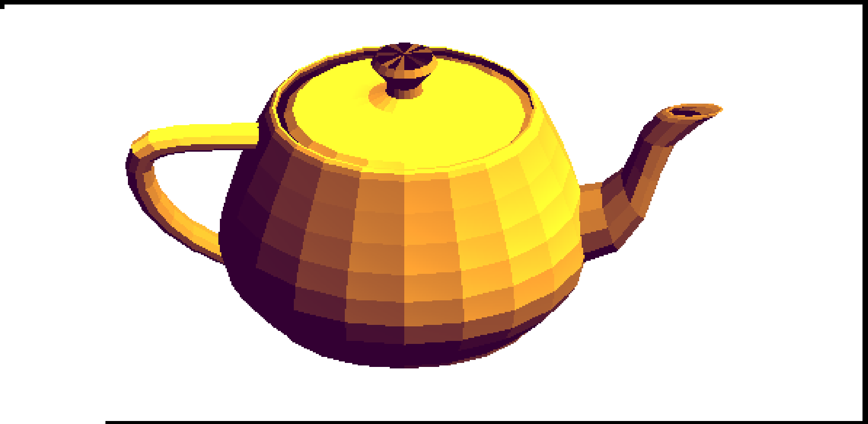


RECURSIVE SUBDIVISION



ADDING SHADING





GEOMETRY SHADER

- Basic limitation on rasterization is that each execution of a vertex shader is triggered by one vertex and can output only one vertex
- Geometry shaders allow a single vertex and other data to produce many vertices
- Example: send four control points to a geometry shader and it can produce as many points as needed for Bezier curve

TESSELLATION SHADERS

- Can take many data points and produce triangles
- More complex since tessellation has to deal with inside/outside issues and topological issues such as holes
- We'll be looking at geometry and tessellation shaders in upcoming topics

SUMMARY

- Quadratic Bezier Surfaces
- Cubic Bezier Surfaces