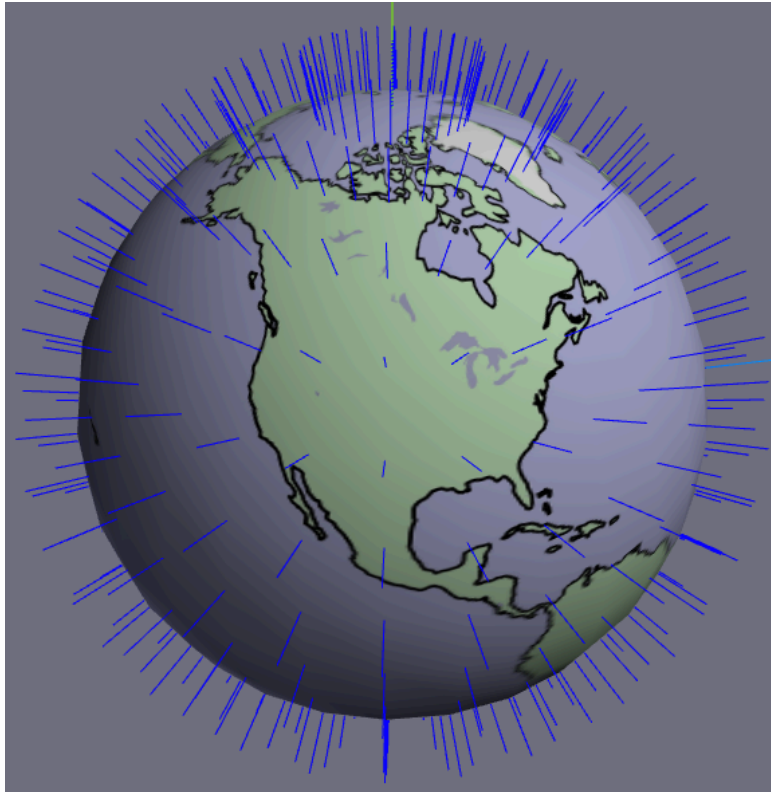
Computer Graphics



Lecture 3

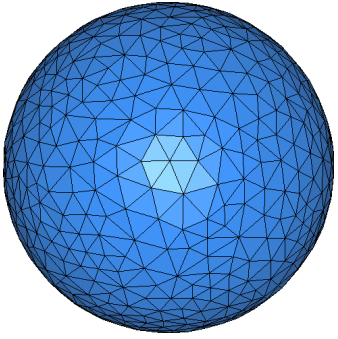
Triangle Meshes:
Surface Normals

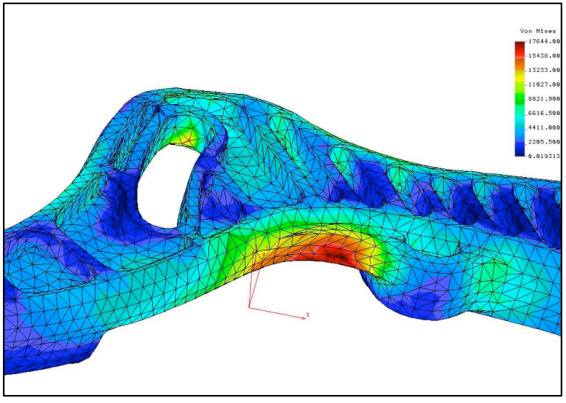
Where are we?

Pseudocode for 3D graphics:

```
Create a model of a scene
Render an image of the model
```

Triangle meshes - one way to approximate arbitrary surfaces



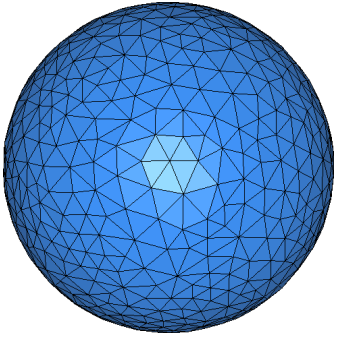


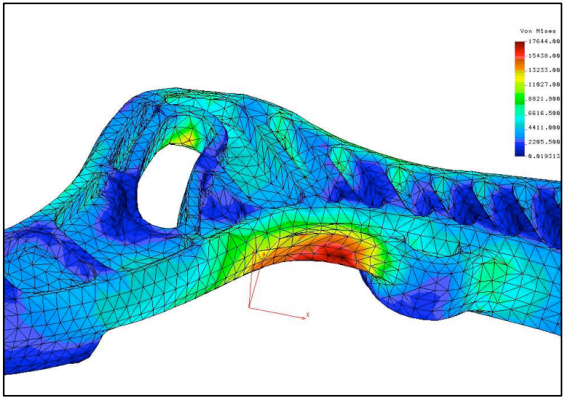
Where are we?

Pseudocode for 3D graphics:

```
Create a model of a scene
Render an image of the model
```

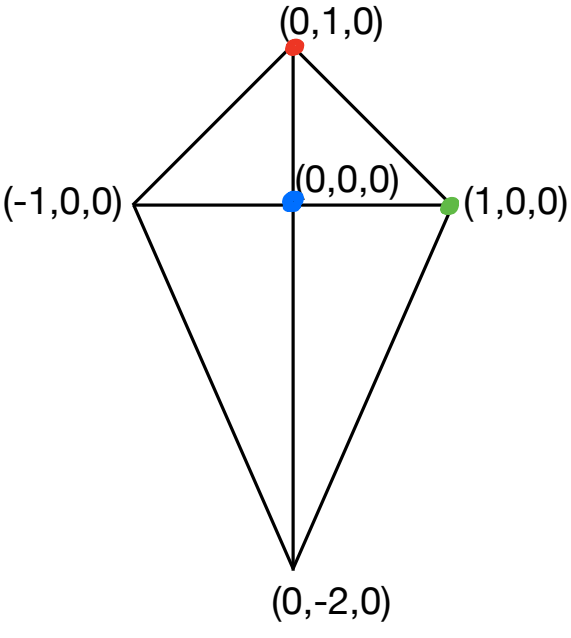
Triangle meshes - one way to approximate arbitrary surfaces





OBJ Format: Positions

A file format for indexed triangle meshes



positions

```
v 0.0 0.0 0.0
v 1.0 0.0 0.0
v 0.0 1.0 0.0
v -1.0 0.0 0.0
v 0.0 -2.0 0.0
```

faces
triangles

```
f 1 2 3
f 1 3 4
f 1 4 5
f 1 5 2
```

Demo: kite.obj, cube.obj

Data on Meshes

To render realistic images, we'll need more than just the geometry.

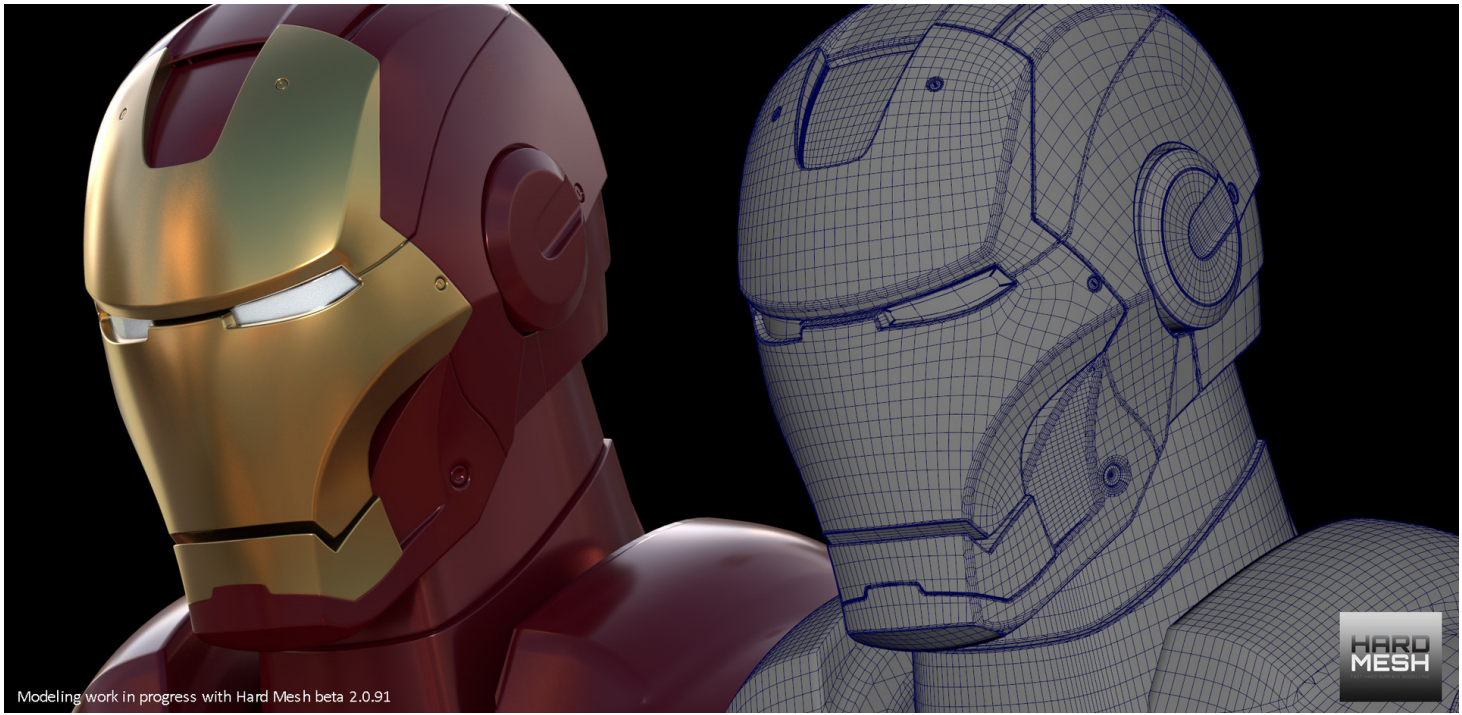


image: http://www.hard-mesh.com/

Data on Meshes

- We need more than just geometry.
- Where do we store it: **Vertices**? Edges? **Faces**?
- Examples:
  - colors stored on face, for faceted objects
  - information about sharp creases stored at edges
  - anything that varies continuously is stored at vertices

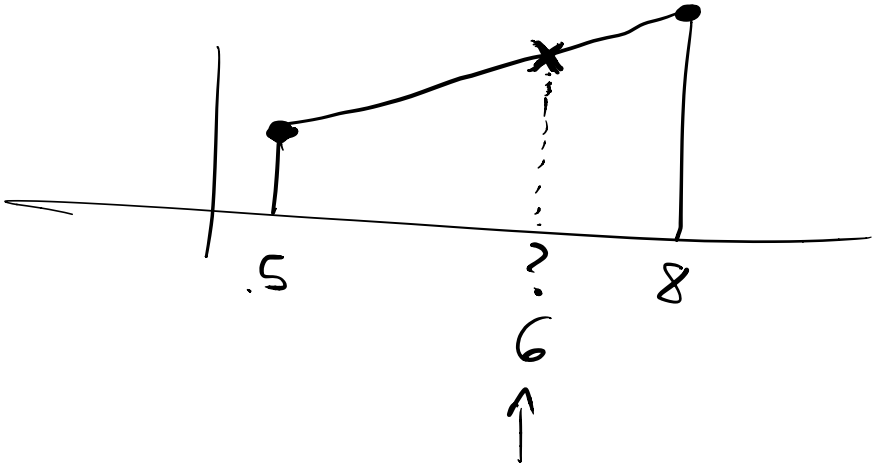
Data on Meshes

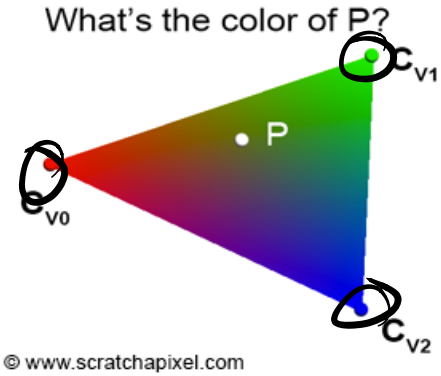
- We need more than just geometry.
- Where do we store it: **Vertices**? Edges? **Faces**?
- Examples:
  - colors stored on face, for faceted objects
  - information about sharp creases stored at edges
  - anything that varies continuously is stored at vertices

*when rendering, **interpolate** values in between*

Interpolation - Intuition

Fill in missing values between known values

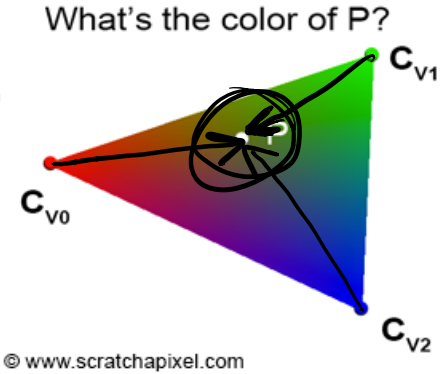




Interpolation - Intuition

Fill in missing values between known values

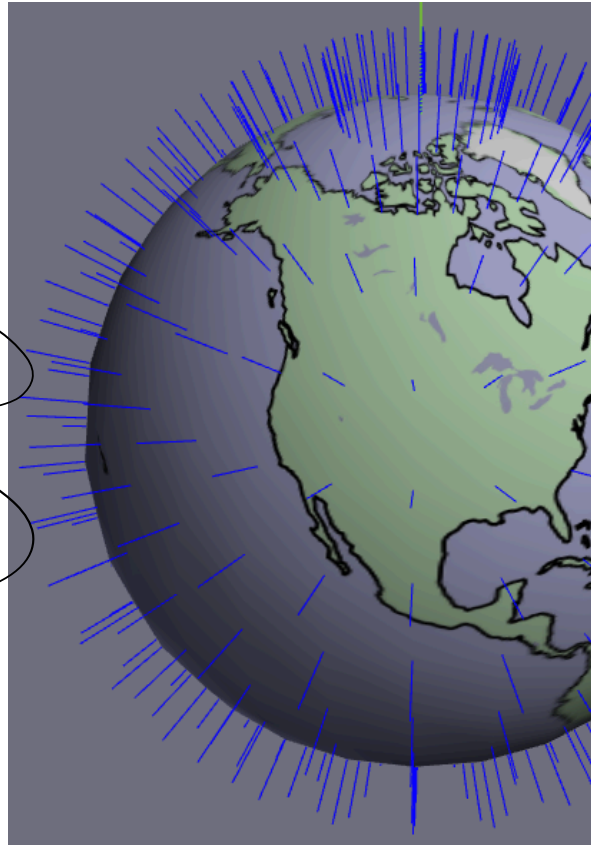
How? This is a question for rendering.
We'll talk about the specifics later.



Data on Meshes

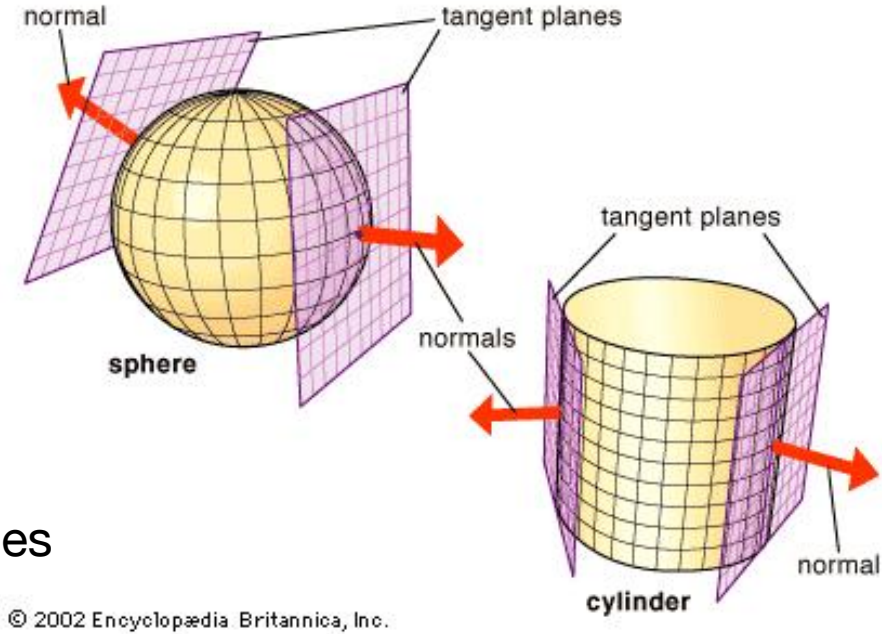
What do we need to store at vertices?

- **Surface Normals**
  to more accurately portray geometry
- **Texture Coordinates**
  to paste image data onto surfaces
- **Positions!?** (last lecture)
  just another piece of per-vertex data!



Surface Normals - Formally

- A point on a smooth surface has a **tangent plane**
- A **normal vector** is orthogonal to the surface (i.e., its tangent plane).

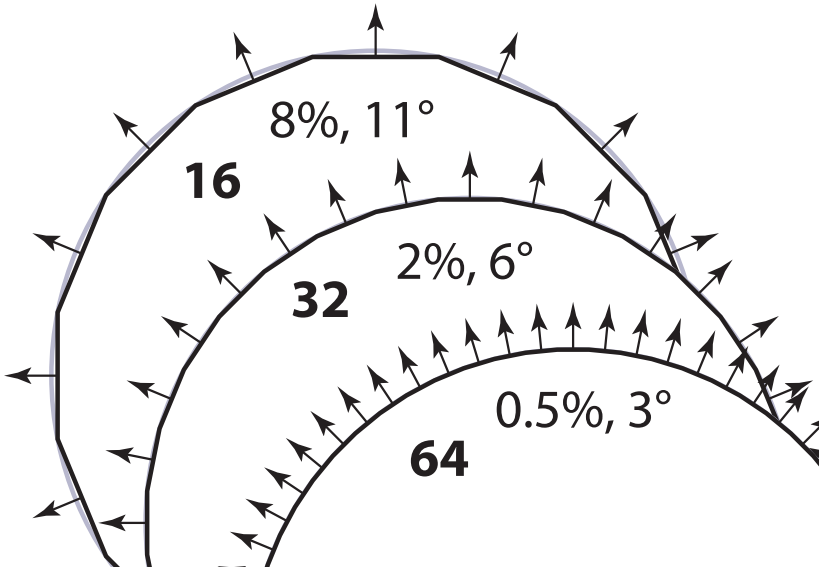


Only unique for smooth surfaces (i.e., not at corners or edges).

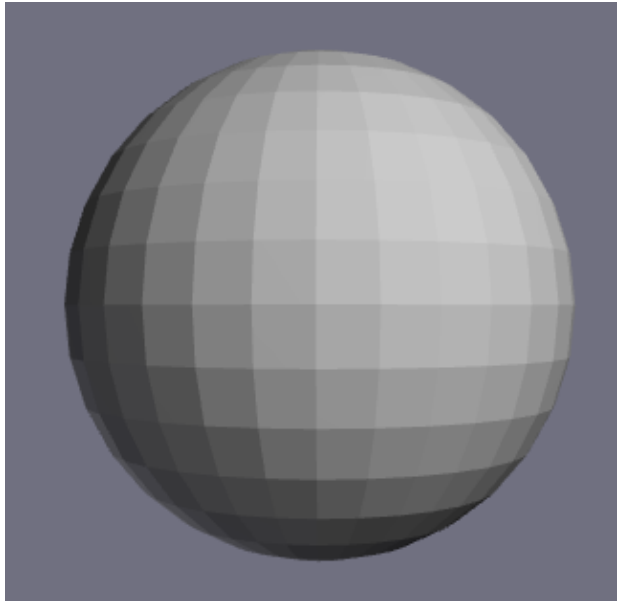
By convention, normal vectors are (usually) **unit length**.

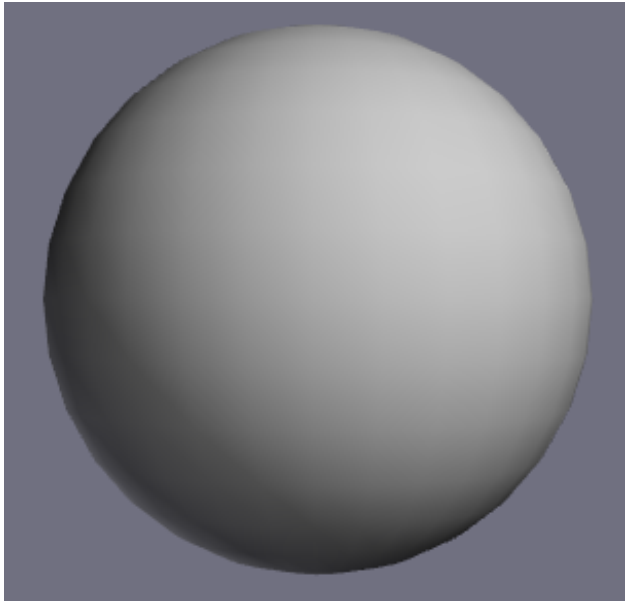
Why are normals important?

- Can't we just use more triangles?
- Error in surface normal shrinks slower than geometry
- Intuition - circle:



Surface Normals: Visual Intuition

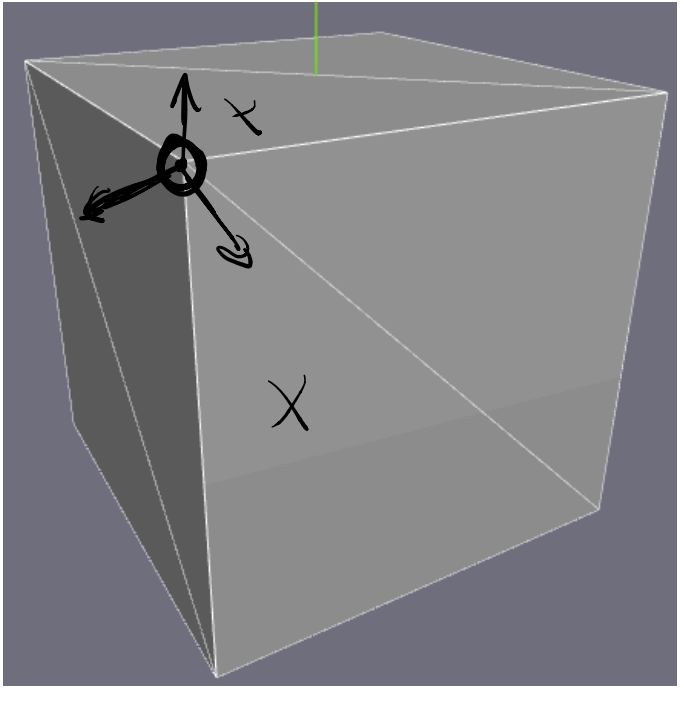




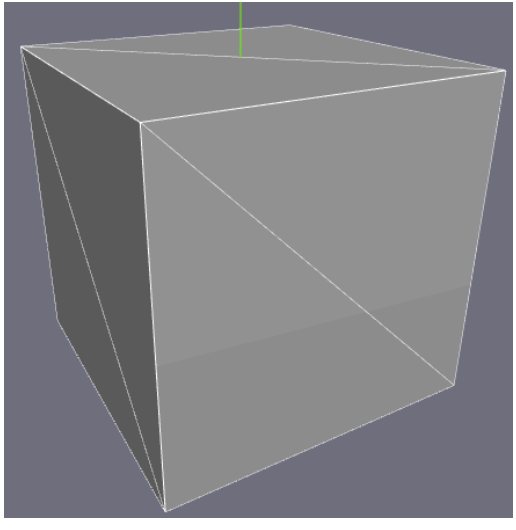
https://facultyweb.cs.wwu.edu/~wehrwes/courses/csci480_21w/meshviewer

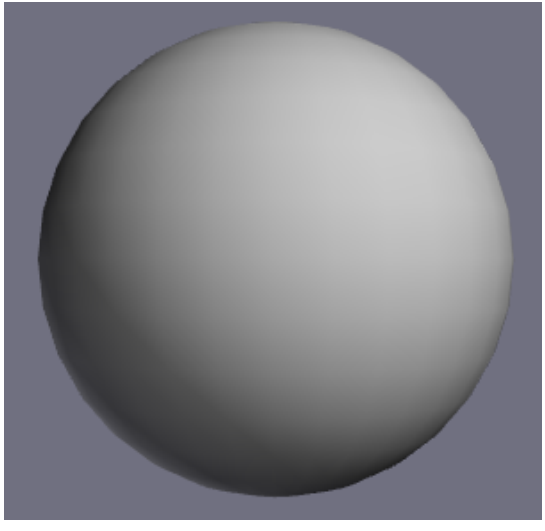
Normals at Discontinuities

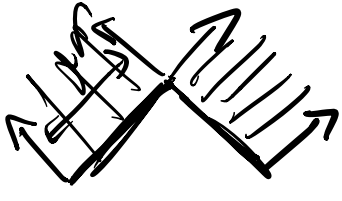
- What is the vertex normal at the corner of a cube?

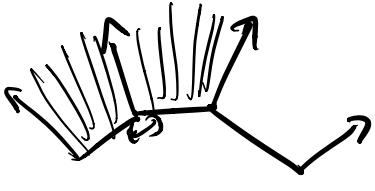


Surface Normals: Smooth vs Faceted Surfaces



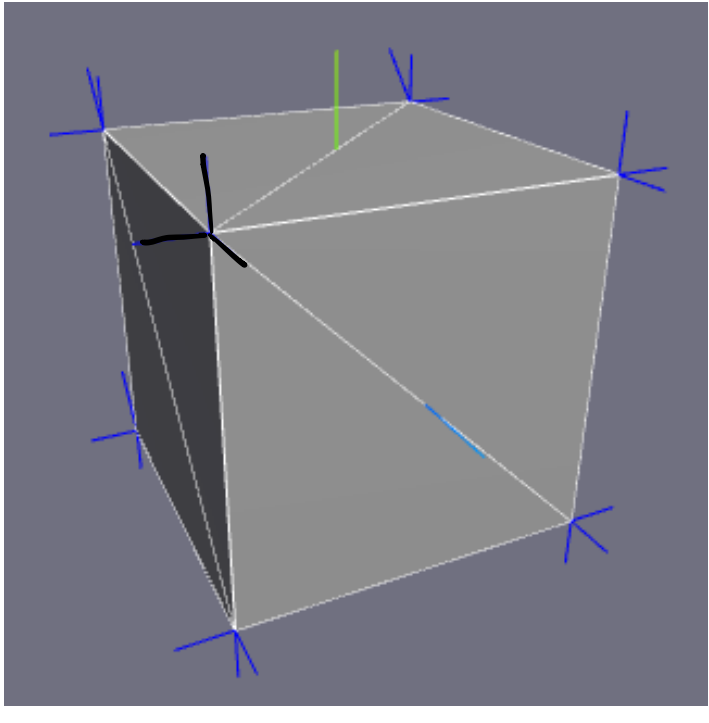






Normals at Discontinuities

- Vertex normal is not **unique**
- Depends which triangle!
- Idea: just like positions:
  - store normals in a list
  - each corner of a triangle has a position index and a normal index



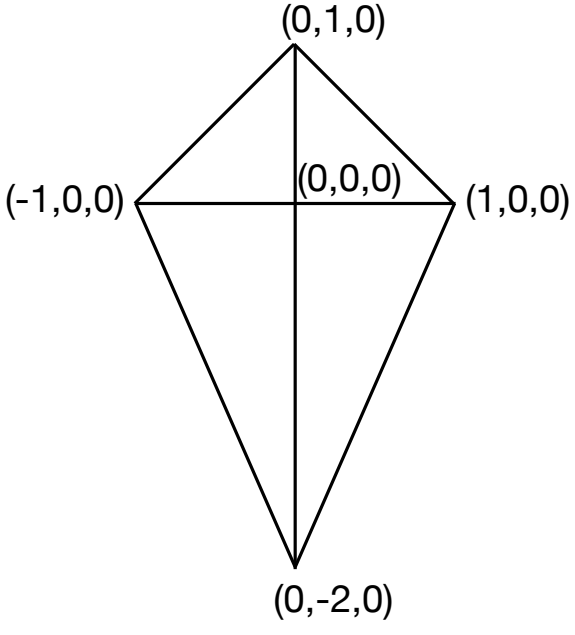
OBJ Format: Triangle Vertices

A face's vertex is specified by 1, 2, or 3 **indices**:

- position index (required)
- texture coordinate index (optional)
- normal index (optional)

position only:	`f 1`	`2`	`3`
all 3:	`f 1/1/1`	`2/2/1`	`3/3/1`
position and texture:	`f 1/1`	`2/2`	`3/3`
position and normal:	`f 1//1`	`2//1`	`3//1`

OBJ Format: Normals



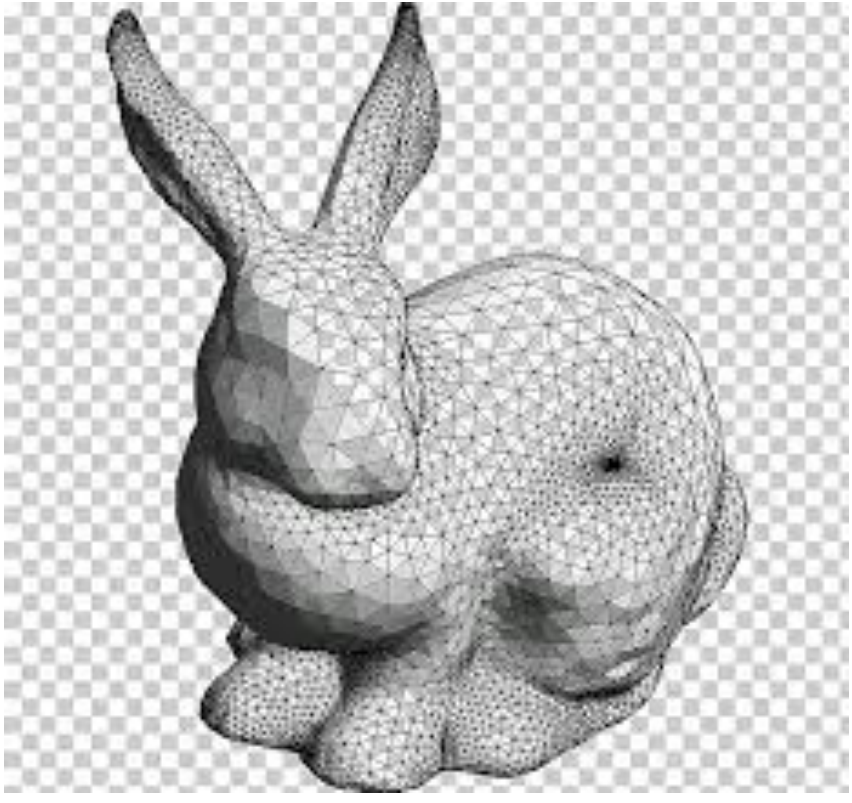

```
v 0.0 0.0 0.0
v 1.0 0.0 0.0
v 0.0 1.0 0.0
v -1.0 0.0 0.0
v 0.0 -2.0 0.0

vn 0.0 0.0 1.0

f 1//1 2//1 3//1
f 1//1 3//1 4//1
f 1//1 4//1 5//1
f 1//1 5//1 2//1
```

Estimating Surface Normals

- In shapes like a sphere and a cube, the normal is easy to calculate.
- What if the "true" surface isn't known?

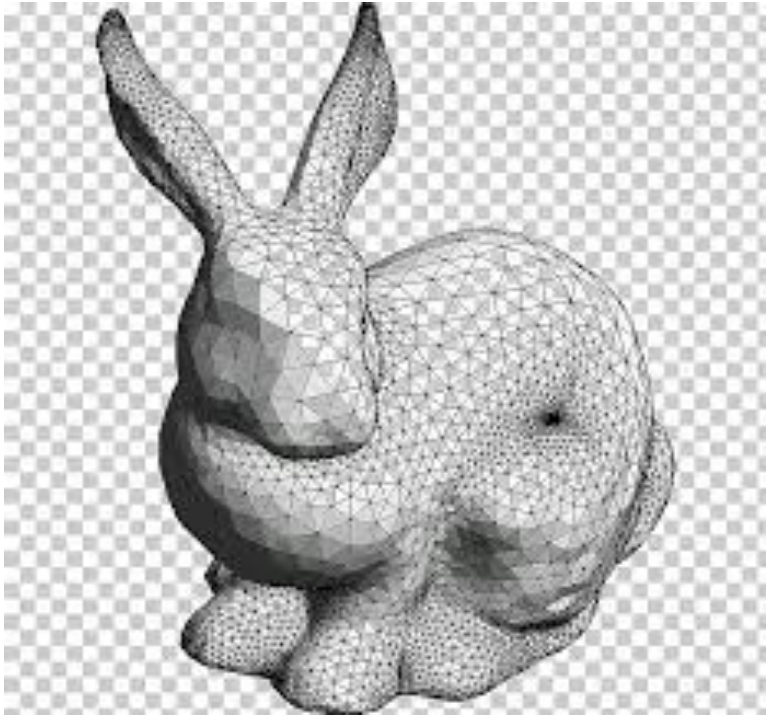


Estimating Surface Normals

Faceted Objects

Triangles represent geometry exactly.

Like the cube, all normals are normal to their triangles.



Estimating Surface Normals

Faceted Objects

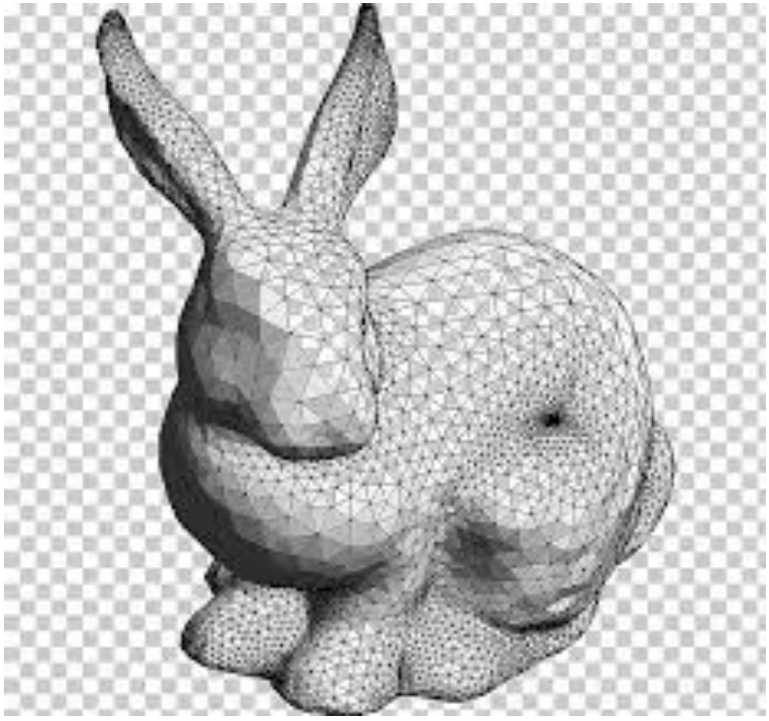
Triangles represent geometry exactly.

Like the cube, all normals are normal to their triangles.

Smooth Objects

Triangles approximate smooth geometry.

Vertex normal is the average of all surrounding triangle normals.



This is the last part of A1.