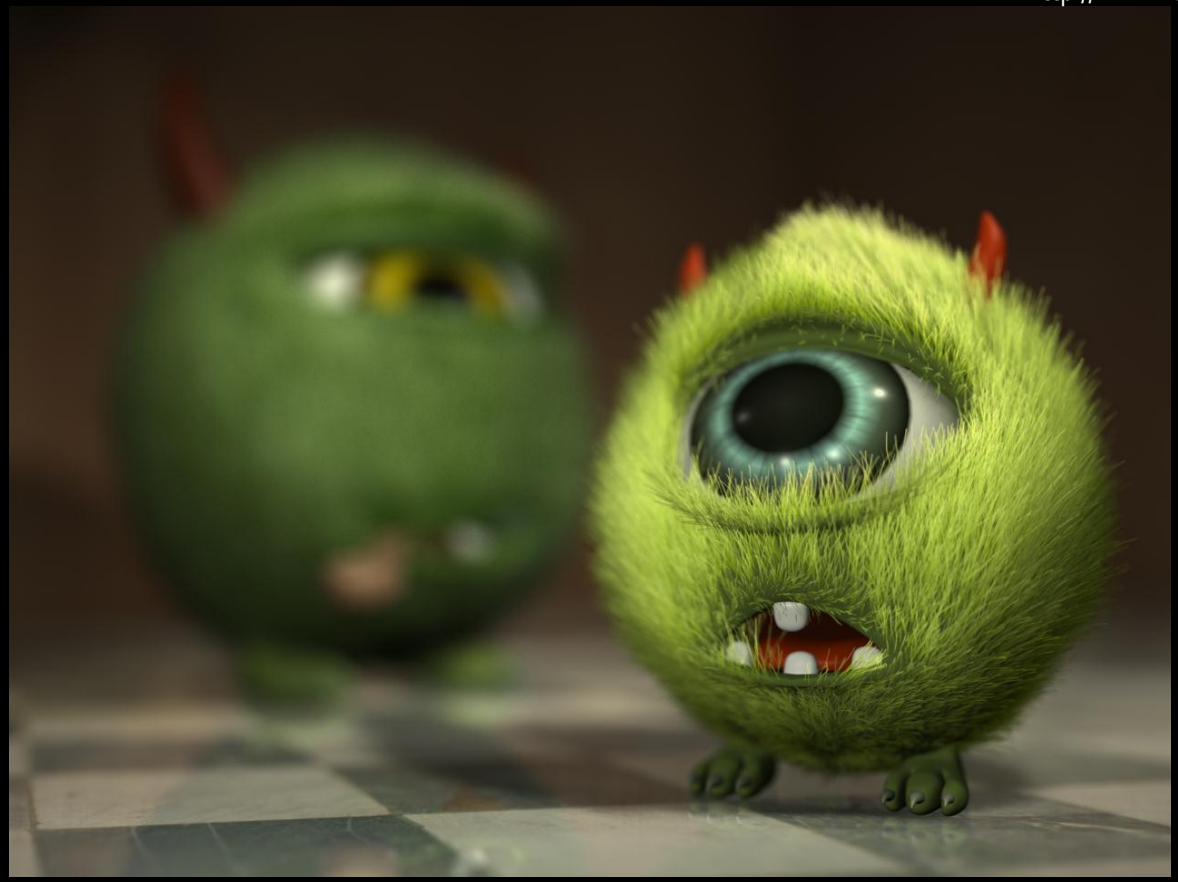
http://www.kunzhou.net/2010/mptracing.pdf

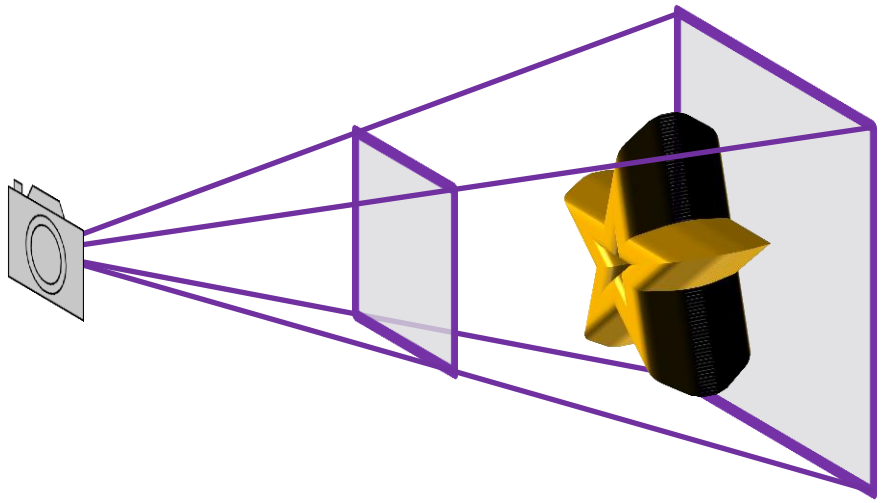


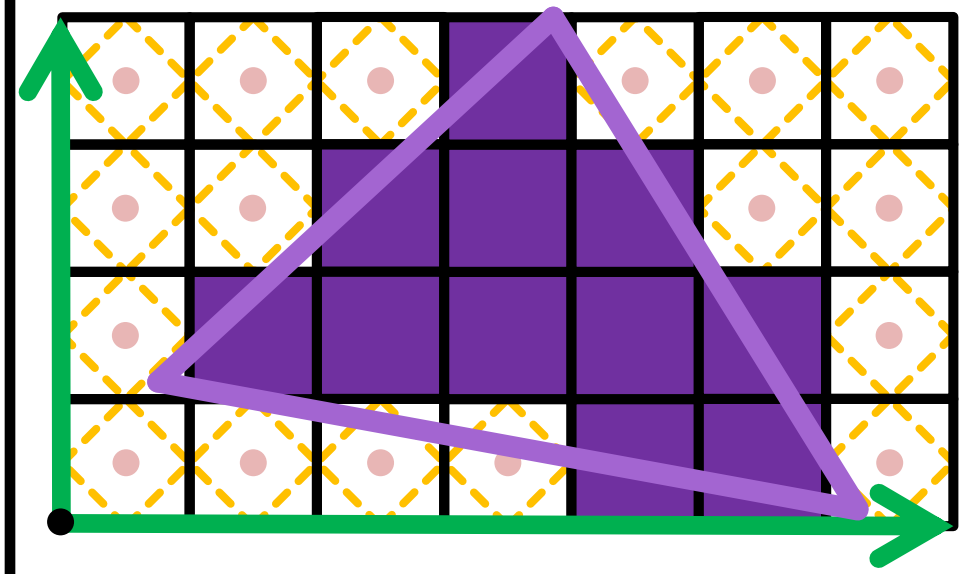
Basics of 3D Rendering

CS 148, Summer 2012
Introduction to Computer Graphics and Imaging

Justin Solomon

What We Have So Far





3D geometry

2D pipeline

Handy Fact

$$\begin{pmatrix} \ddots & & \\ & \text{matrix} & \\ & & \ddots \end{pmatrix} \times$$

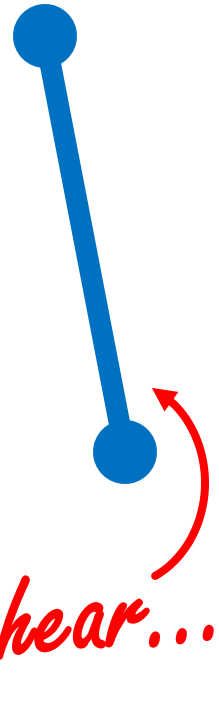
$=$

Matrices preserve flat geometry

Handy Fact

$$\begin{pmatrix} \ddots & & \\ & \text{matrix} & \\ & & \ddots \end{pmatrix} \times$$

$=$

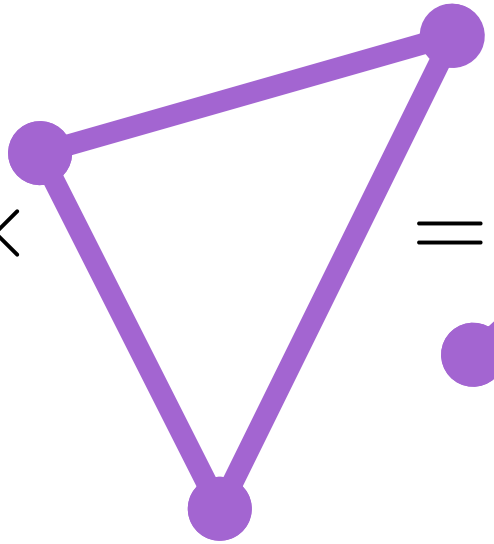


can rotate/scale/shear...

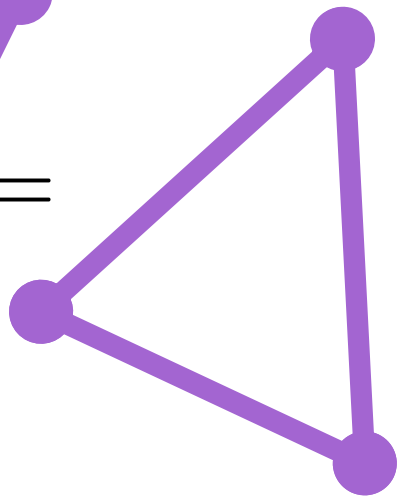
Matrices preserve flat geometry

Handy Fact

$$\begin{pmatrix} \ddots & & \\ & \text{matrix} & \\ & & \ddots \end{pmatrix} \times$$



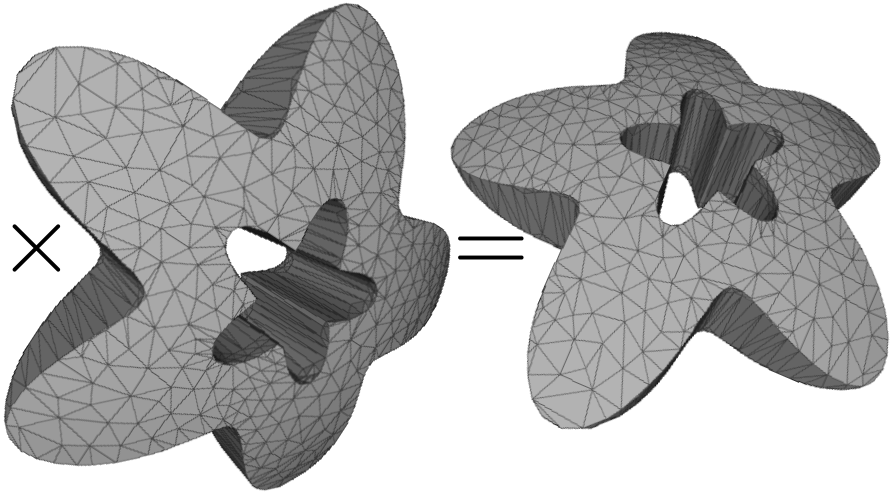
$=$



Matrices preserve flat geometry

Handy Fact

$$\begin{pmatrix} \ddots & & \\ & \text{matrix} & \\ & & \ddots \end{pmatrix}$$



Matrices preserve flat geometry

So What?

3D triangles look like 2D triangles under camera transformations.

So What?

3D triangles look like 2D triangles under camera transformations.

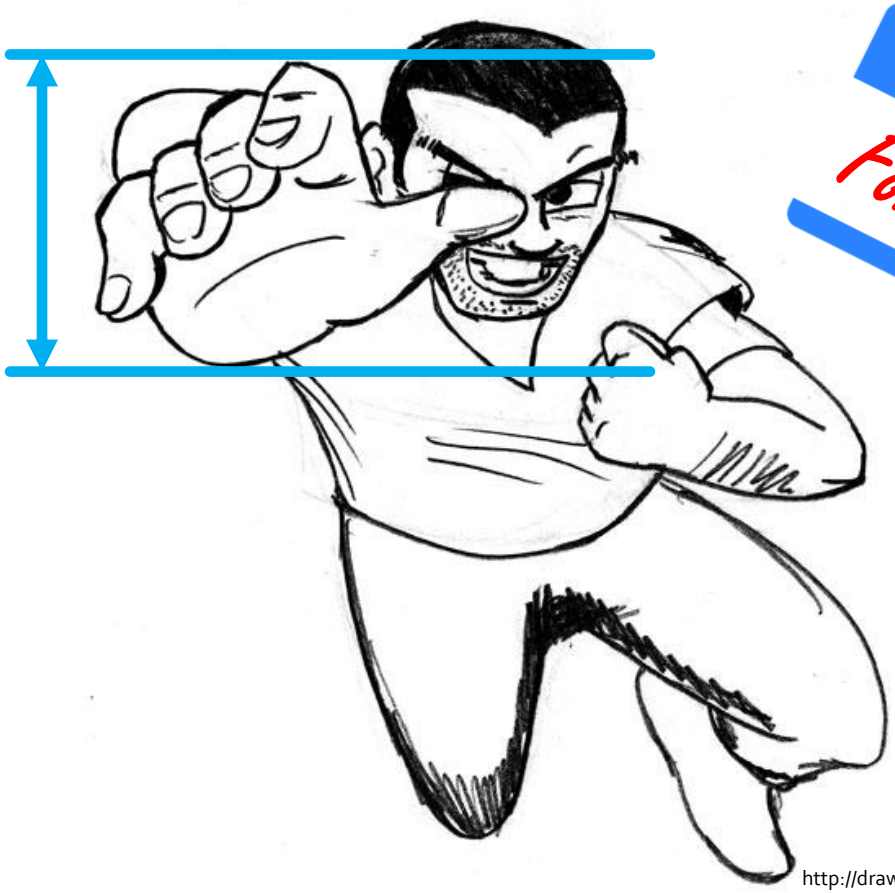
Use 2D pipeline for 3D rendering!

Side Note

http://drawntothis.com/wp-content/uploads/2010/09/Random_Guy.jpg

Only true for flat shapes!

Side Note



http://drawntothis.com/wp-content/uploads/2010/09/Random_Guy.jpg

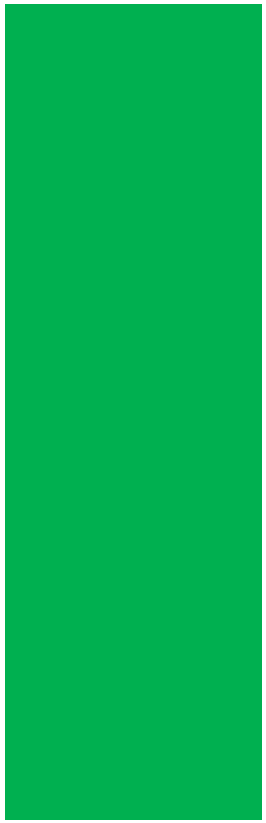
Only true for flat shapes!

Potential Pipeline

1. Apply camera transformation to all triangles in the scene.

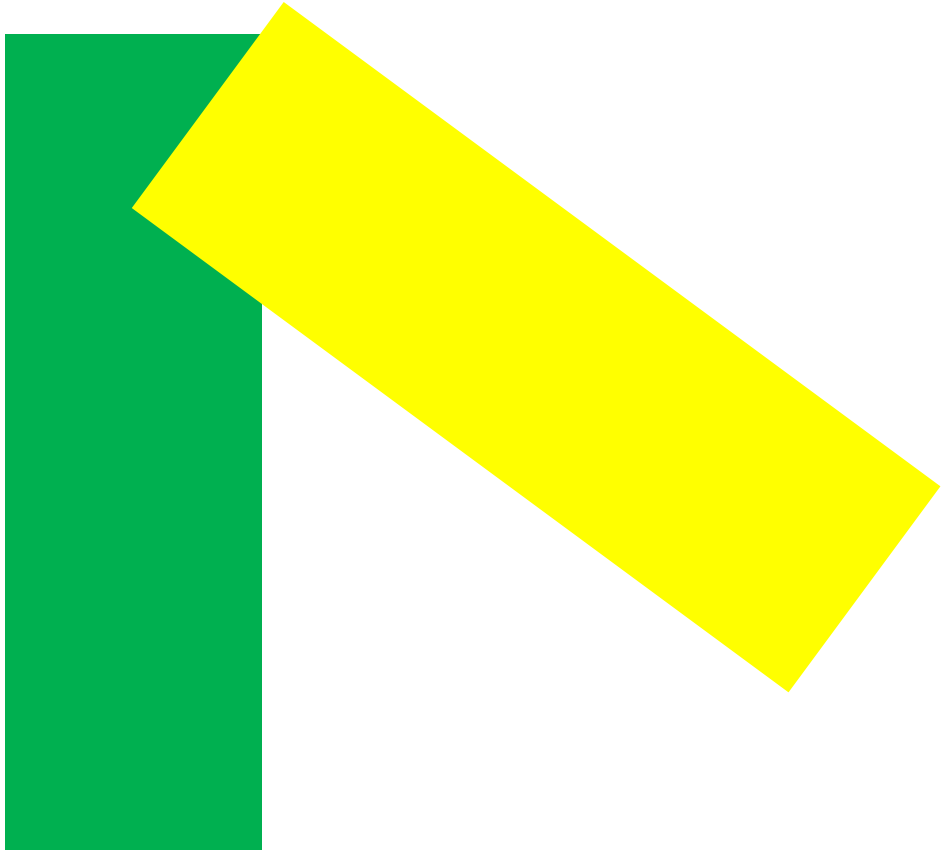
2. Render them one at a time.

Painter's Algorithm



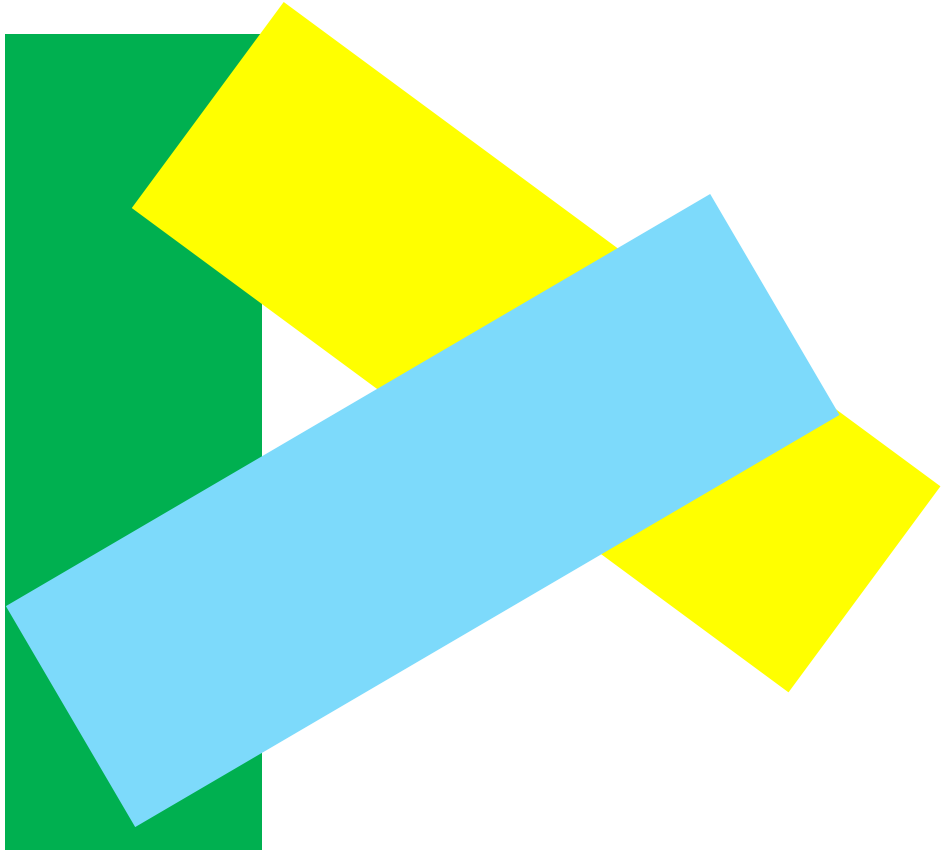
Draw items one at a time

Painter's Algorithm



Draw items one at a time

Painter's Algorithm



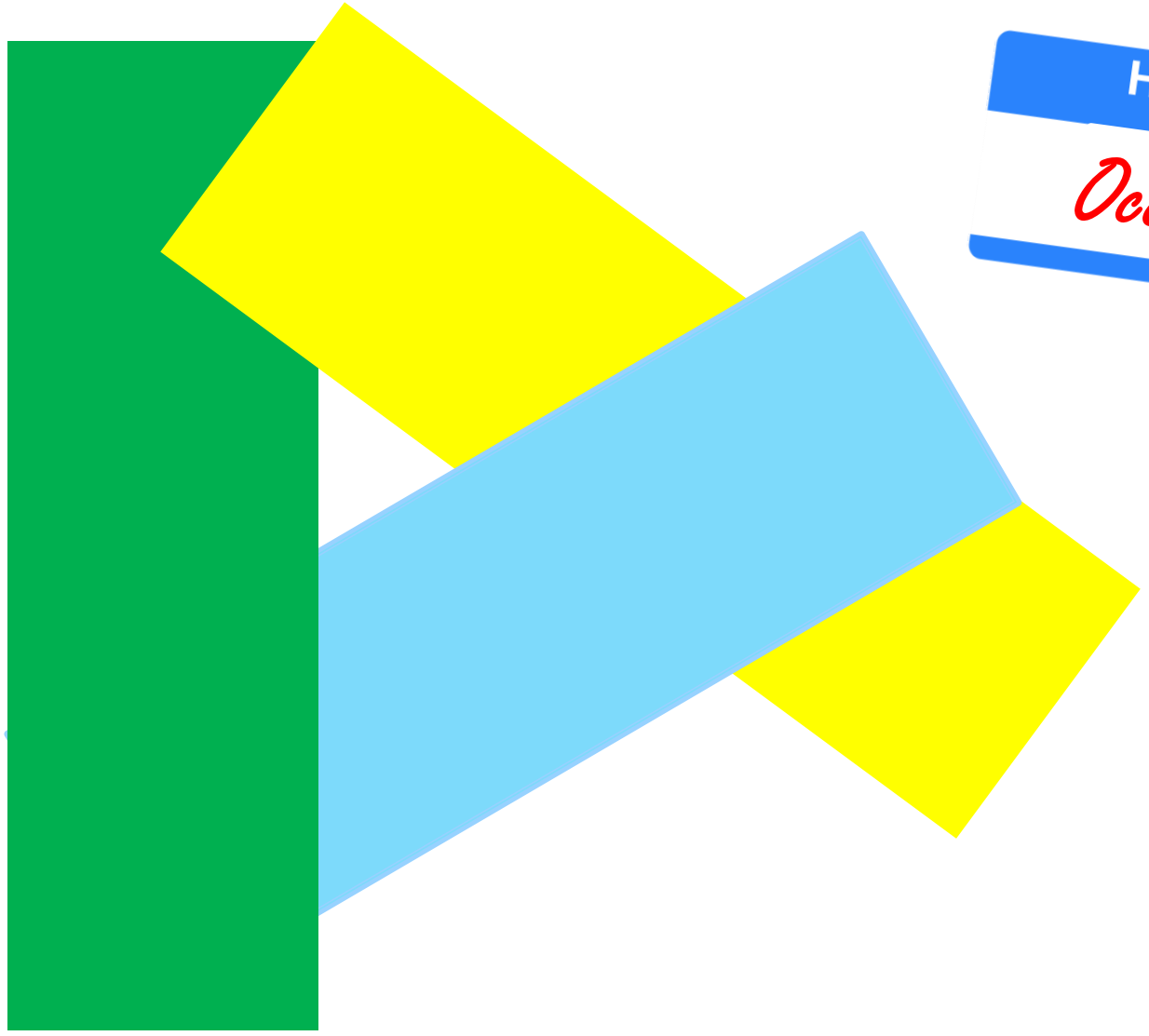
Draw items one at a time

Painter's Algorithm

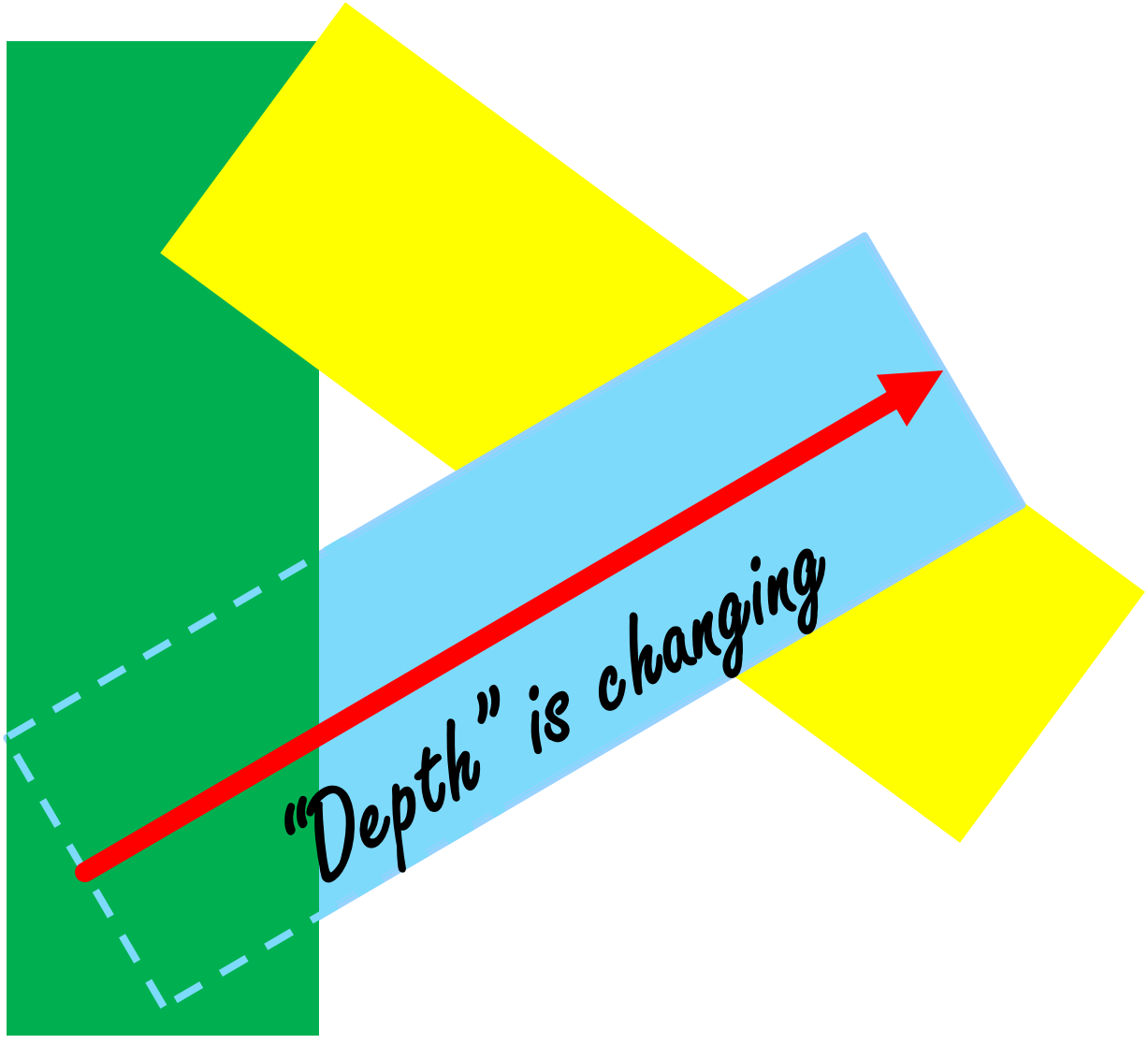
Draw items one at a time

What Order?

What Order?



What Order?



Early Hidden Surface Approaches

- Pre-compute rendering order
- Cut geometry as needed
- …

A Characterization of Ten Hidden-Surface Algorithms

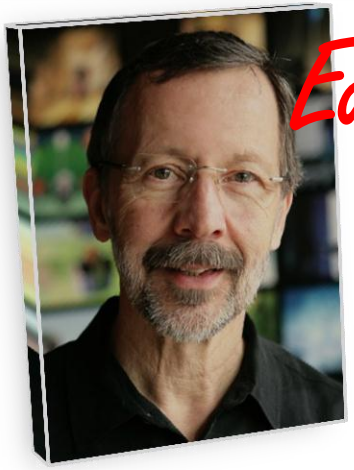
IVAN E. SUTHERLAND*, ROBERT F. SPROULL**, AND ROBERT A. SCHUMACKER*

This paper discusses the hidden-surface problem from the point of view of sorting. The various surfaces of an object to be shown in hidden-surface

Observation

Each pixel can decide what is on top independently.

Observation

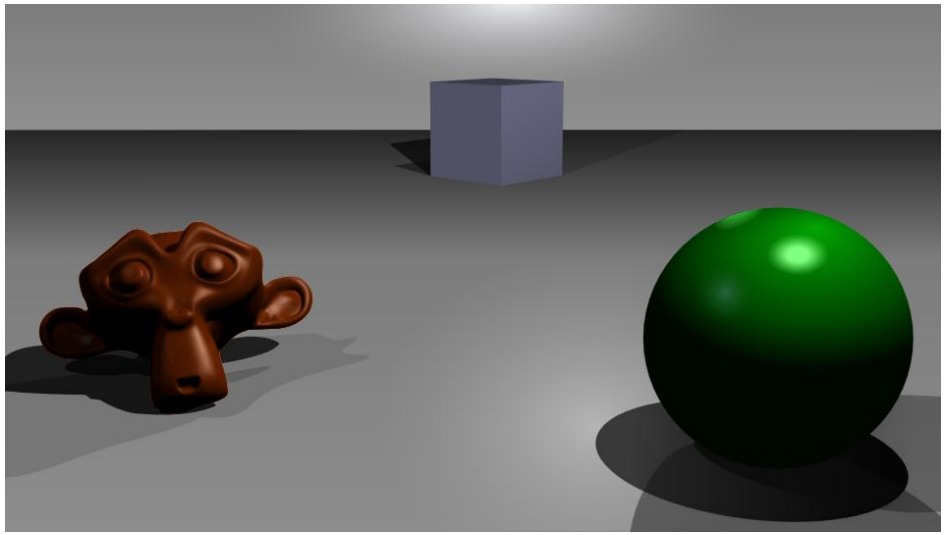


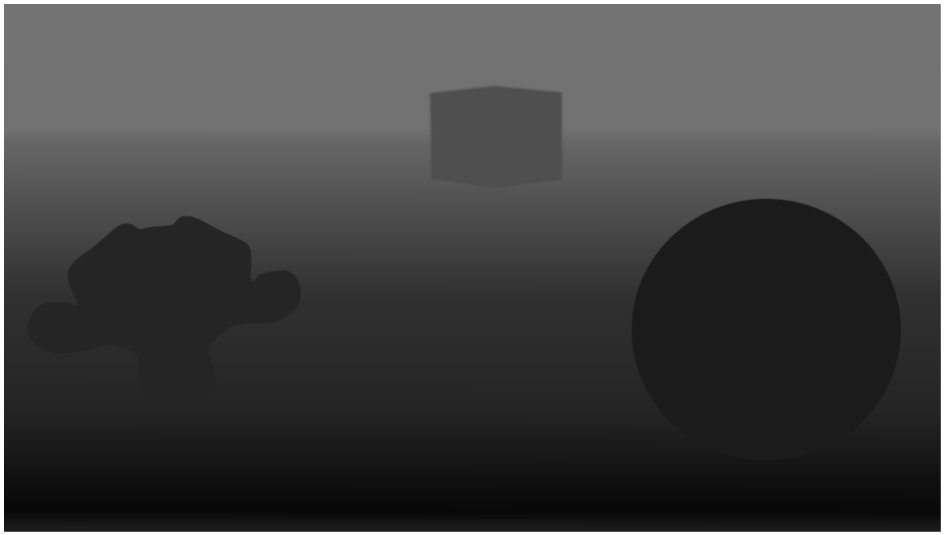
Ed Catmull

Each pixel can decide what is on top independently.

Wolfgang Straßer

The Z-Buffer





http://upload.wikimedia.org/wikipedia/commons/4/4e/Z_buffer.svg

Render depth alongside color

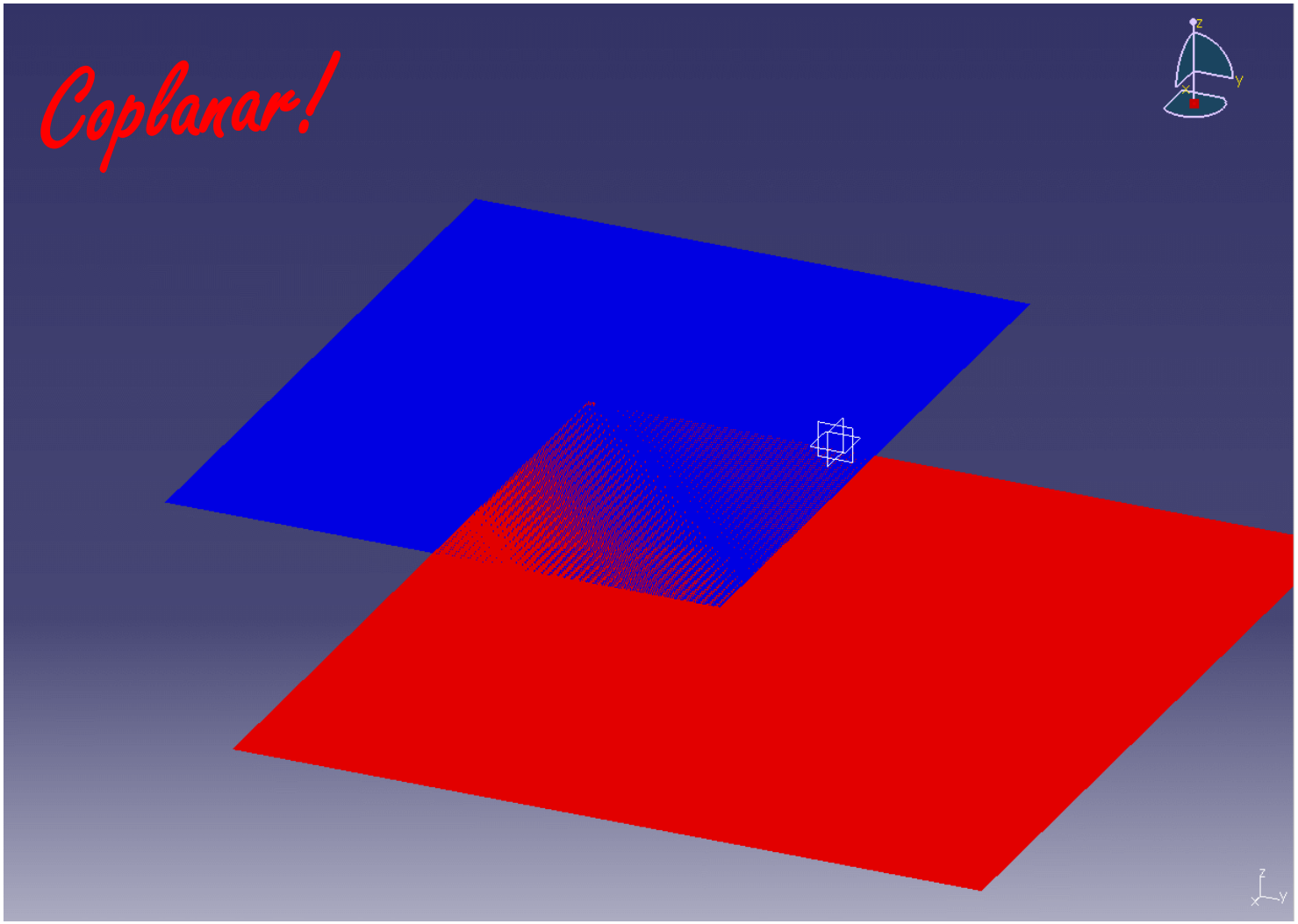
Implementing Z Buffers

Generate fragments containing color *and* depth

Select the closest one!

Fragment blending phase

Depth Fighting *or "Z Fighting," "stitching," "bleeding," …*



http://ps-2.kevo09.com/CATIA-B18/basug_C2/basugbt1510.htm

How to Resolve Fighting

Offset or scale one of the objects

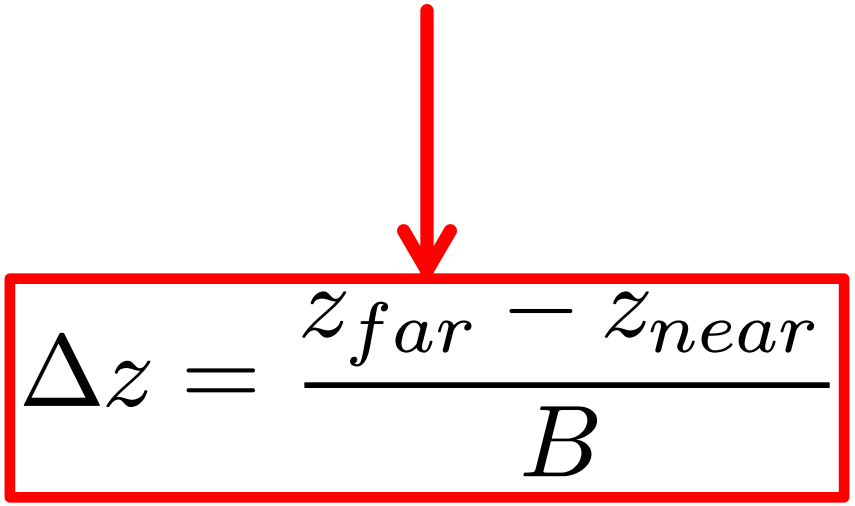
How to Resolve Fighting

Offset or scale one of the objects

Hack?!

Z Buffer Details

$$[z_{near}, z_{far}] \mapsto \{0, 1, \ldots, B-1\} \cup \{\infty\}$$



$$\Delta z = \frac{z_{far} - z_{near}}{B}$$

Integer Arithmetic!

Z Buffer Details

$$[z_{near}, z_{far}] \mapsto \{0, 1, \ldots, B-1\} \cup \{\infty\}$$

$$\Delta z = \frac{z_{far} - z_{near}}{B}$$

Must resolve depths!

Integer Arithmetic!

Z Buffer Details

$$[z_{near}, z_{far}] \mapsto \{0, 1, \ldots, B-1\} \cup \{\infty\}$$

Can change any variable

$$\Delta z = \frac{z_{far} - z_{near}}{B}$$

Integer Arithmetic!

Nonlinearity in Rasterized Depths

$$z_{world} \mapsto z = z_{near} + z_{far} - \frac{z_{near} z_{far}}{z_{world}}$$

Nonlinearity in Rasterized Depths

$$z_{world} \mapsto z = z_{near} + z_{far} - \frac{z_{near} z_{far}}{z_{world}}$$

$$\frac{dz}{dz_{world}} = \frac{z_{near} z_{far}}{z_{world}^2}$$

Nonlinearity in Rasterized Depths

$$z_{world} \mapsto z = z_{near} + z_{far} - \frac{z_{near} z_{far}}{z_{world}}$$

$$\frac{dz}{dz_{world}} = \frac{z_{near} z_{far}}{z_{world}^2}$$

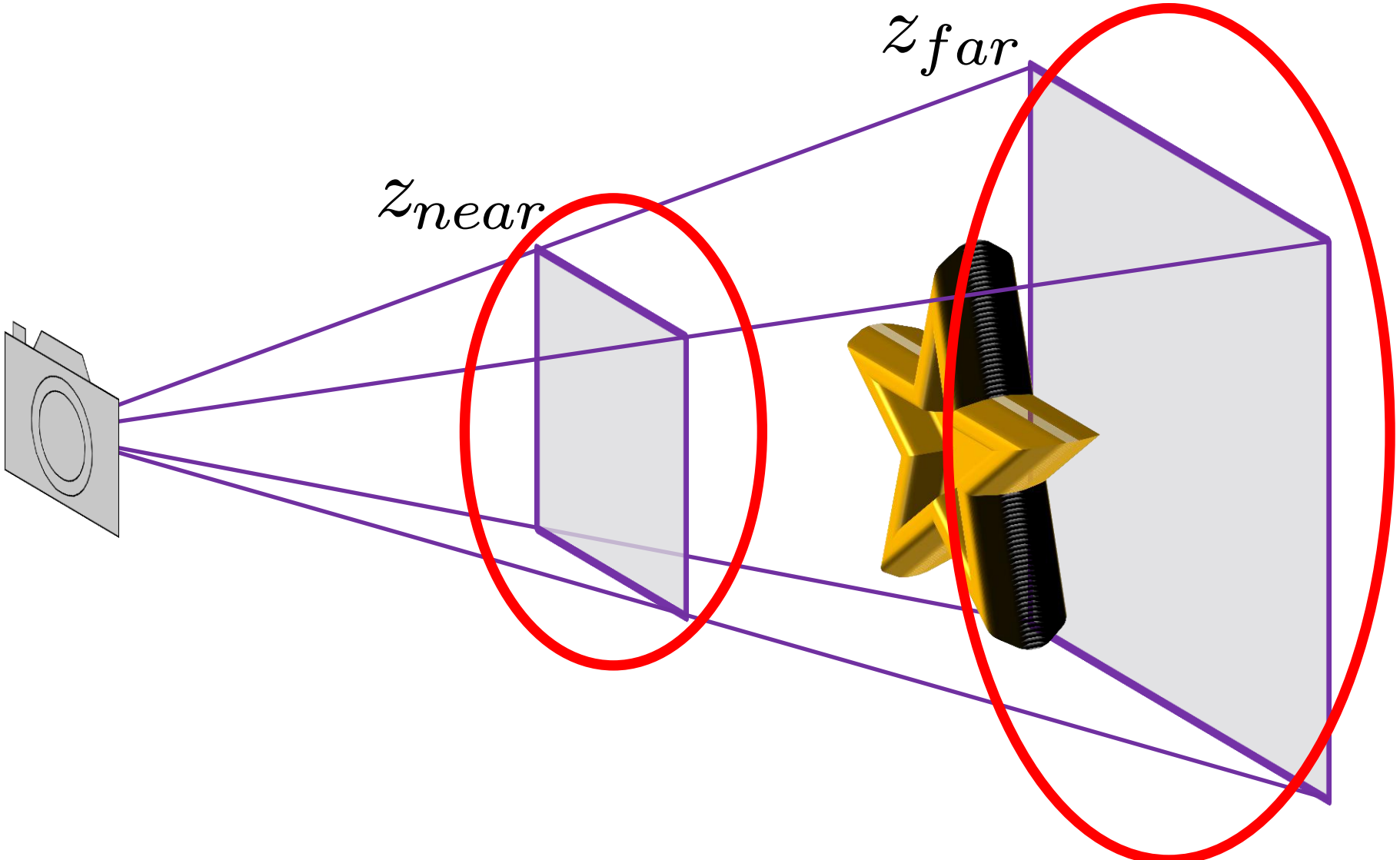
$$\Delta z \approx \frac{z_{near} z_{far} \Delta z_{world}}{z_{world}^2}$$

Nonlinearity in Rasterized Depths

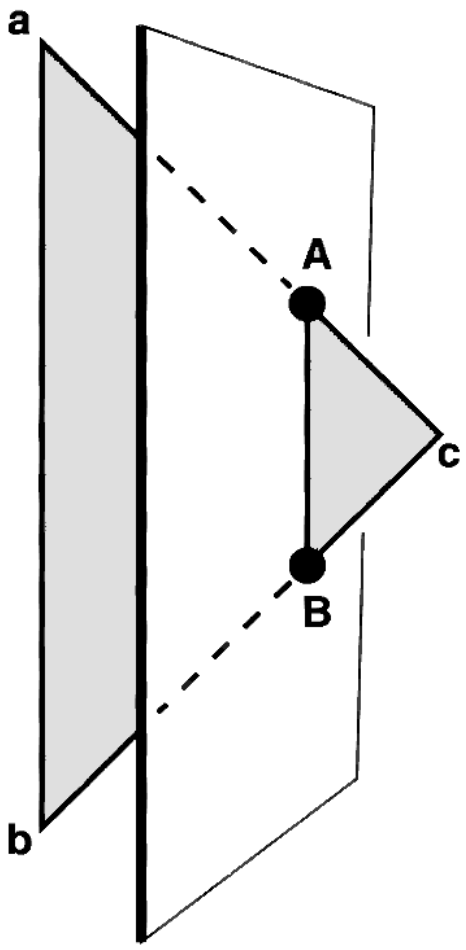
Better resolution when closer to the screen.

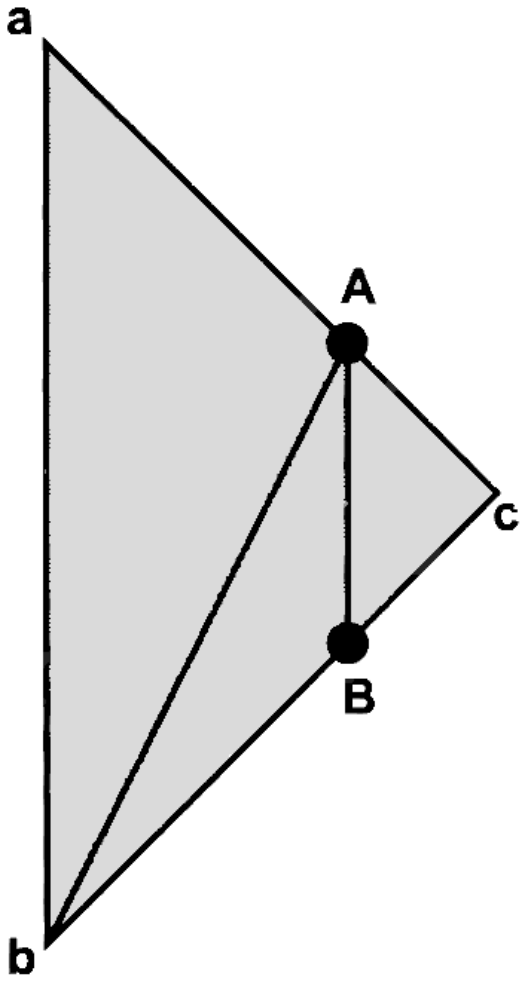
$$\Delta z \approx \frac{z_{near} z_{far} \Delta z_{world}}{z_{world}^2}$$

Clipping



Clipping

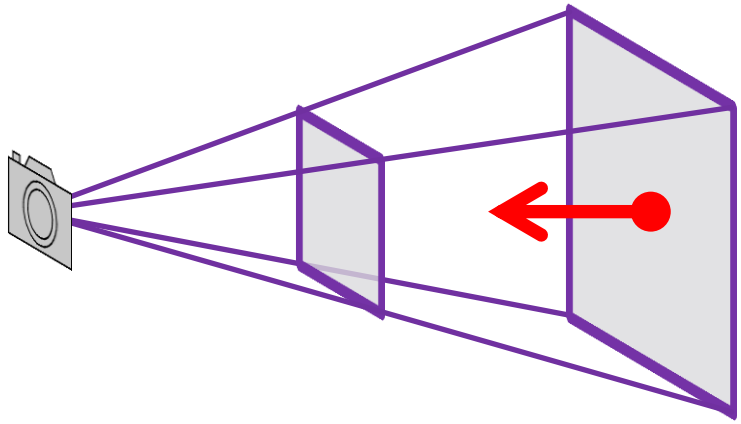




http://www.cs.unc.edu/Courses/comp575-f11/present/present11.html

Clipping

```
For each of six planes do
    If entirely outside of half space
        Break (not visible)
    Else if triangle intersects plane
        Clip triangle
        If quadrilateral is left
            Break into two triangles
```



Textbook, page 171

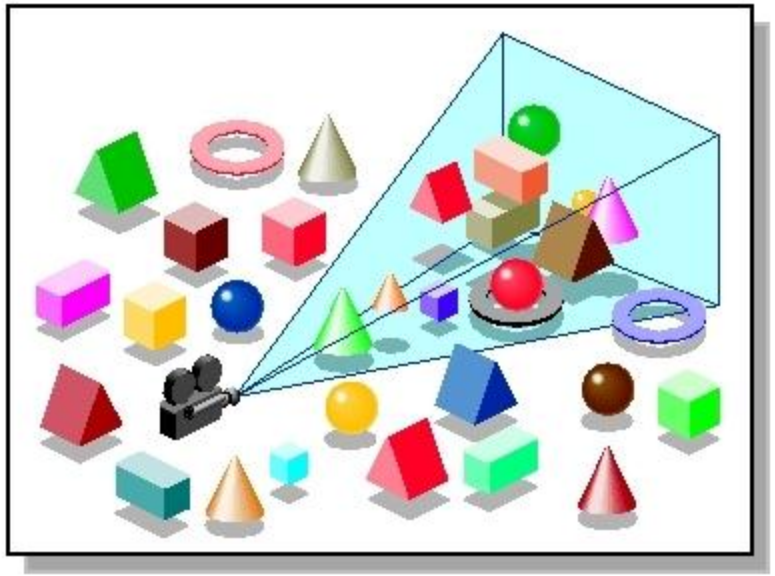
Cull [kuhl]:

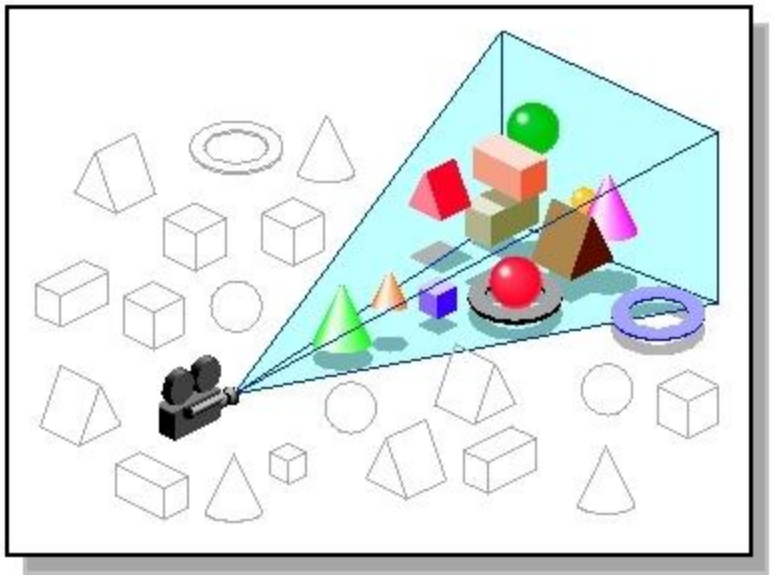
To identify and throw away invisible geometry to save processing time.

Basic Culling Strategies

- **View volume culling**: remove geometry outside frustum
- **Occlusion culling**: remove invisible geometry
- **Backface culling**: remove geometry facing away from the camera

View Volume Culling





Potential strategies:

- Approximate frustum with simpler shape
- Store scene hierarchically
- …

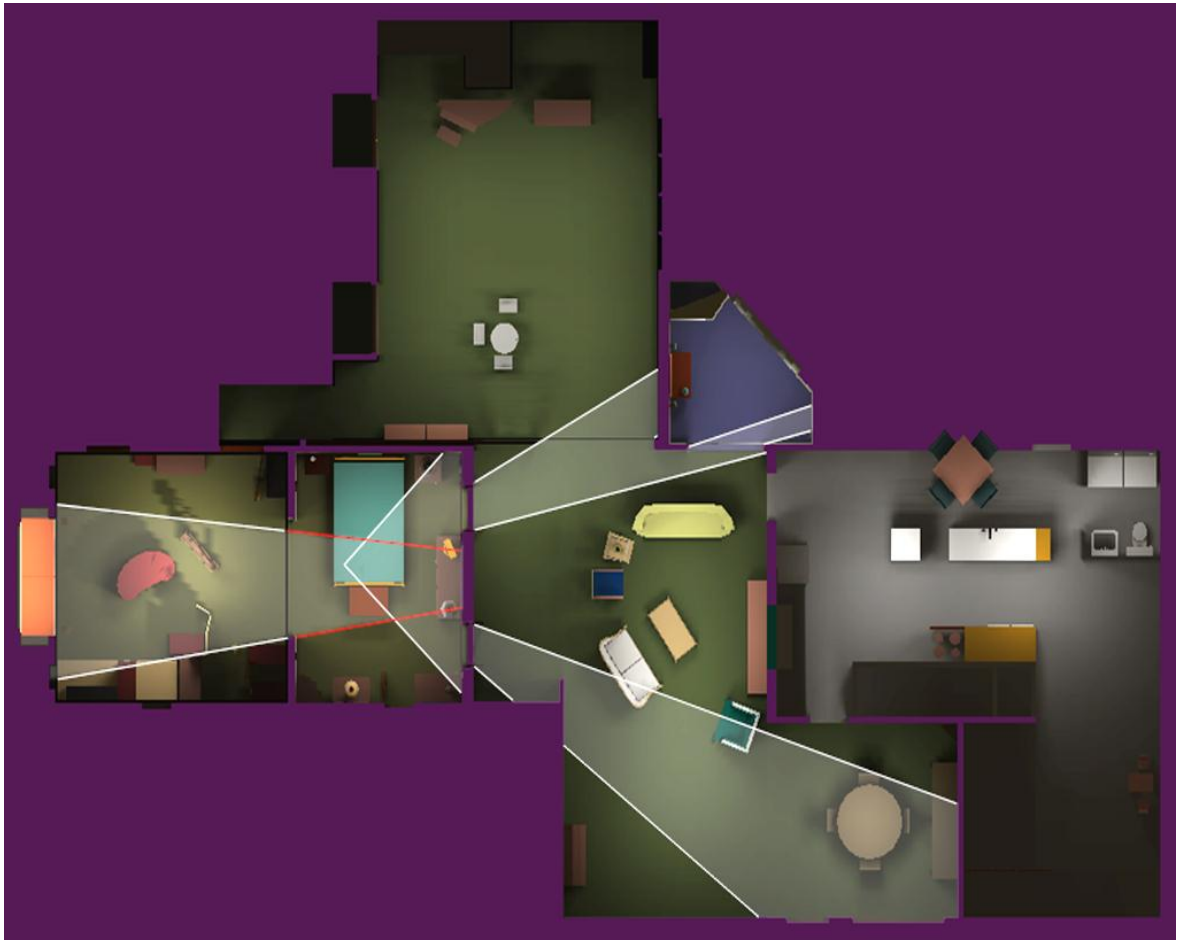
http://i.minus.com/i75qjiyFQzVCI.jpg

Occlusion Culling

Portal rendering

http://www.aaid.ca/flash/media/hkmh/images/floor1/oooa-geology-portal-cg-rendering.jpg

Occlusion Culling



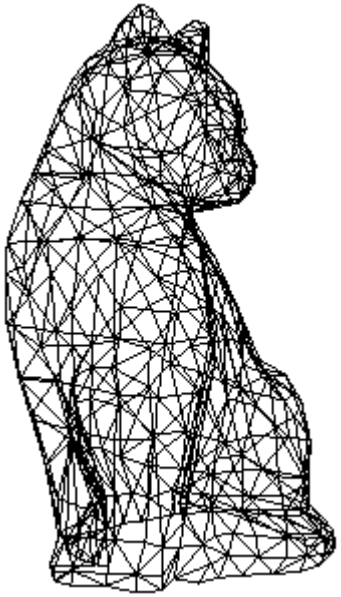
Potentially Visible Set (mirror reflection!)

http://www.cs.virginia.edu/~luebke/publications/portals.html

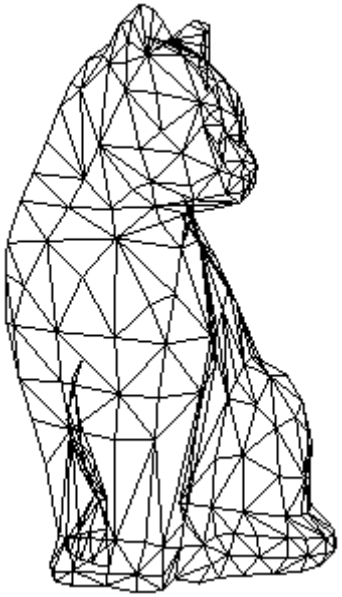
Normal [**nawr**-m*uh*l]:

A vector perpendicular to a surface or piece of geometry; constant for planes and triangles.

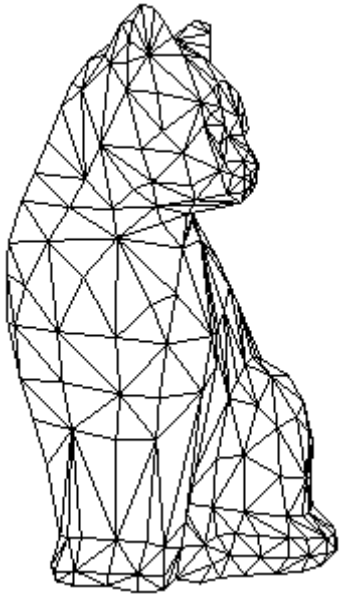
Backface Culling



None



Backface culling

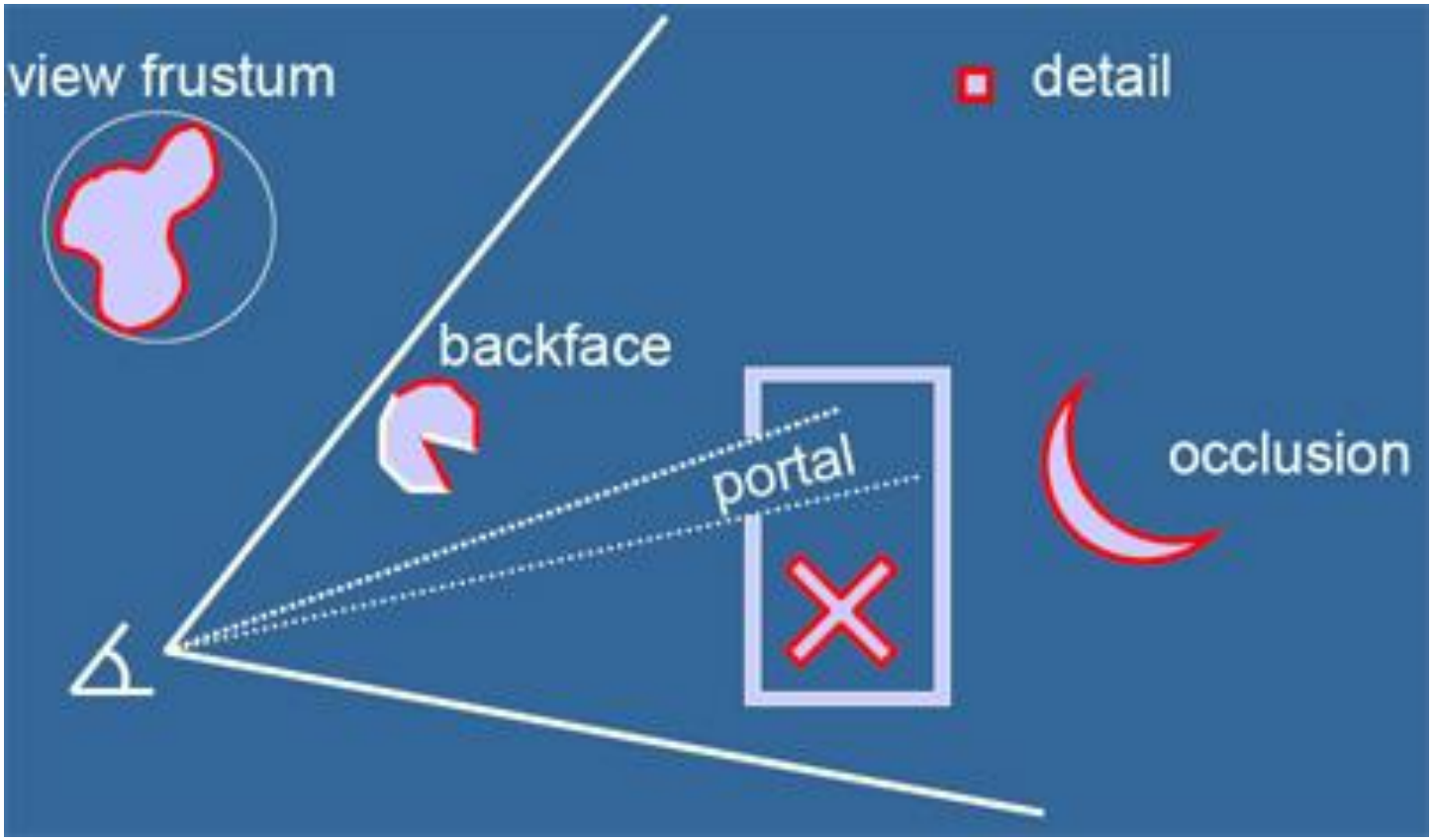


Hidden surface removal

http://medialab.di.unipi.it/web/IUM/Waterloo/node70.html

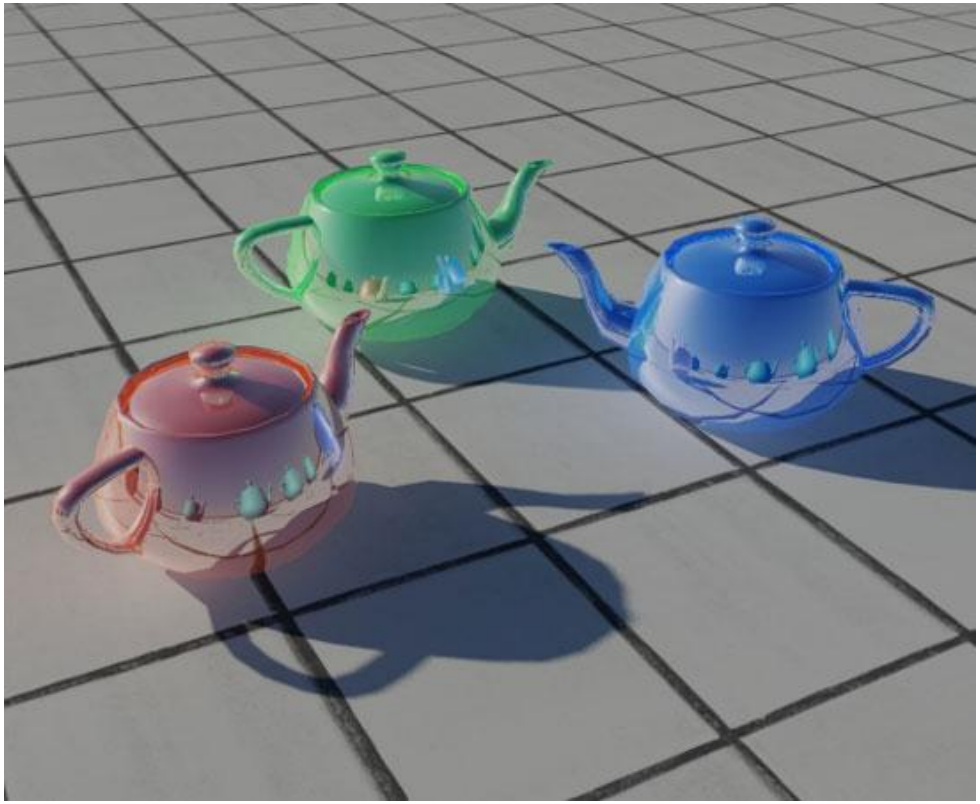
Cull triangles with back-facing normals

Culling Techniques: Summary



http://www.gamerendering.com/category/optimizations/culling-optimizations/

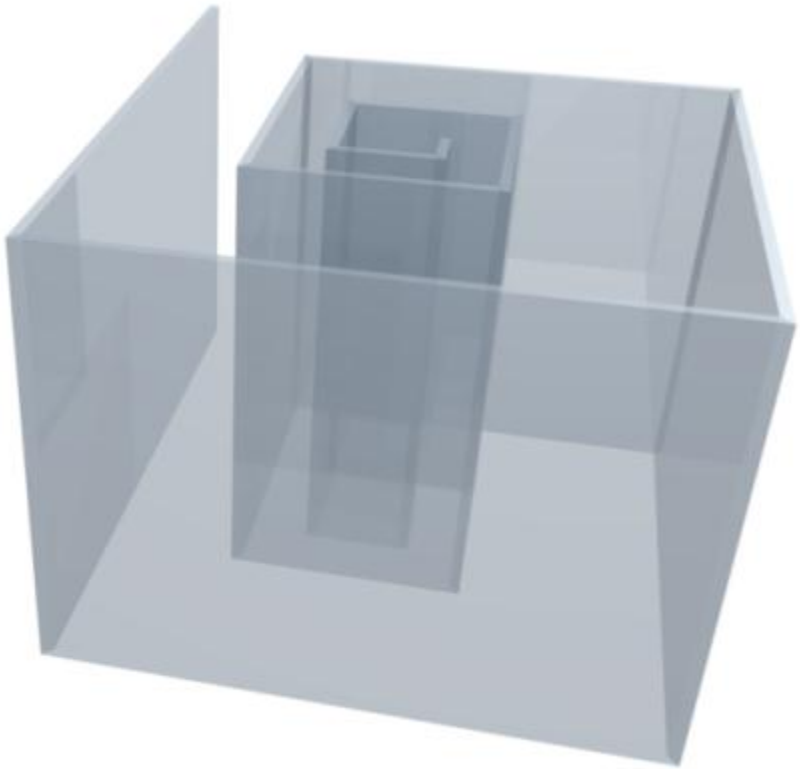
Challenges for Raster Methods



Reflection and Shadows

http://artist-3d.com/free_3d_models/uploads/mantalray.jpg

Challenges for Raster Methods



Transparency

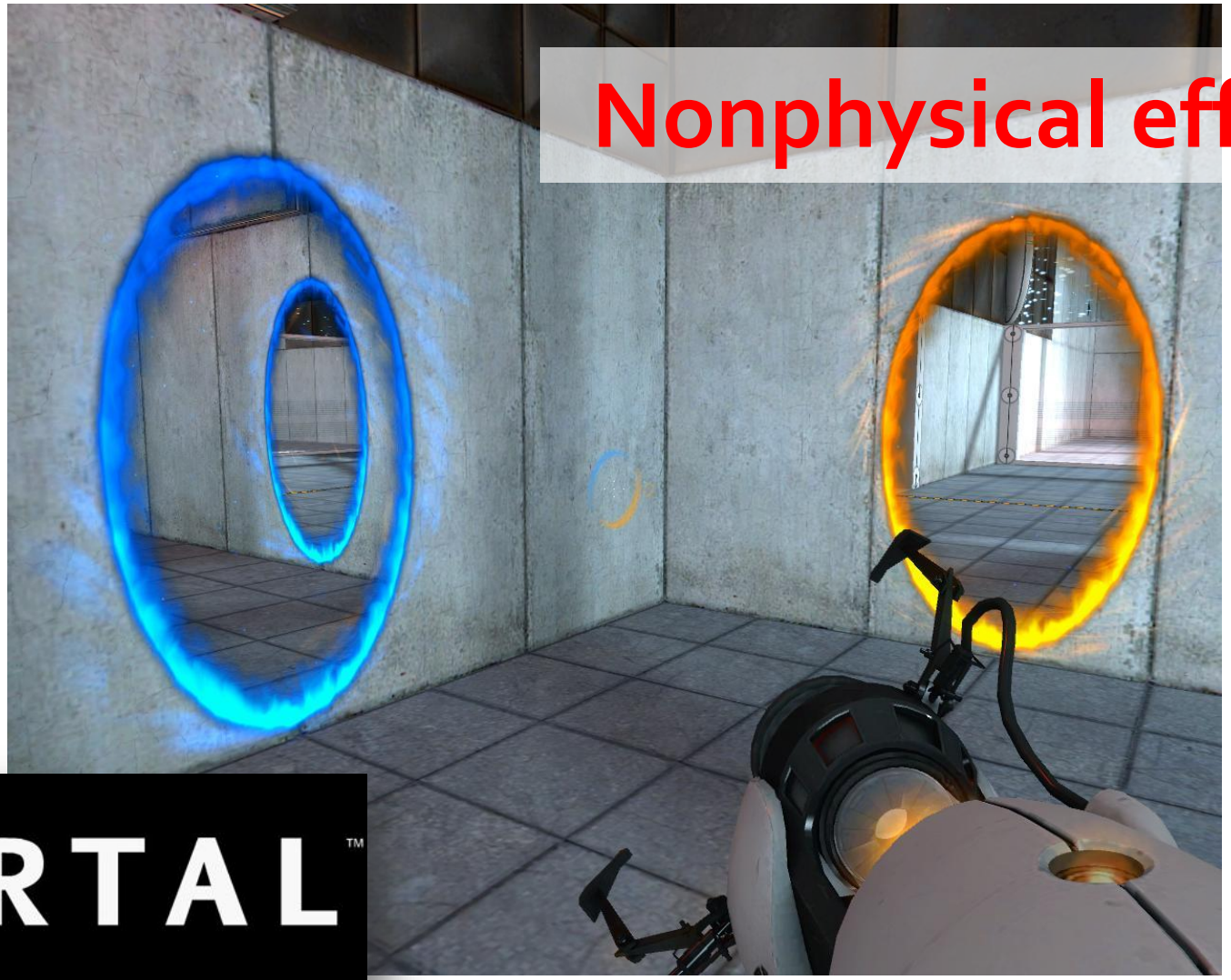
http://www.archicadwiki.com/Bugs/TransparencyIn3dWindow

Challenges for Raster Methods

Depth of Field

http://www.seemsartless.com/guides/camera-dof-cars-fast-360.jpg

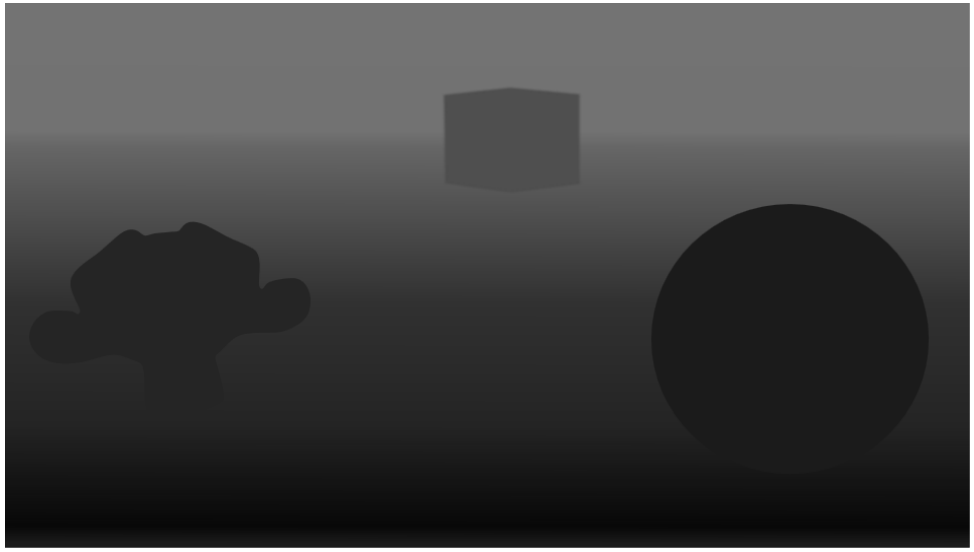
Challenges for Raster Methods



http://littlemissmetamorph.blogspot.com/2011/07/game-inspired-portal-doing-makeup-with.html

Two Related Issues

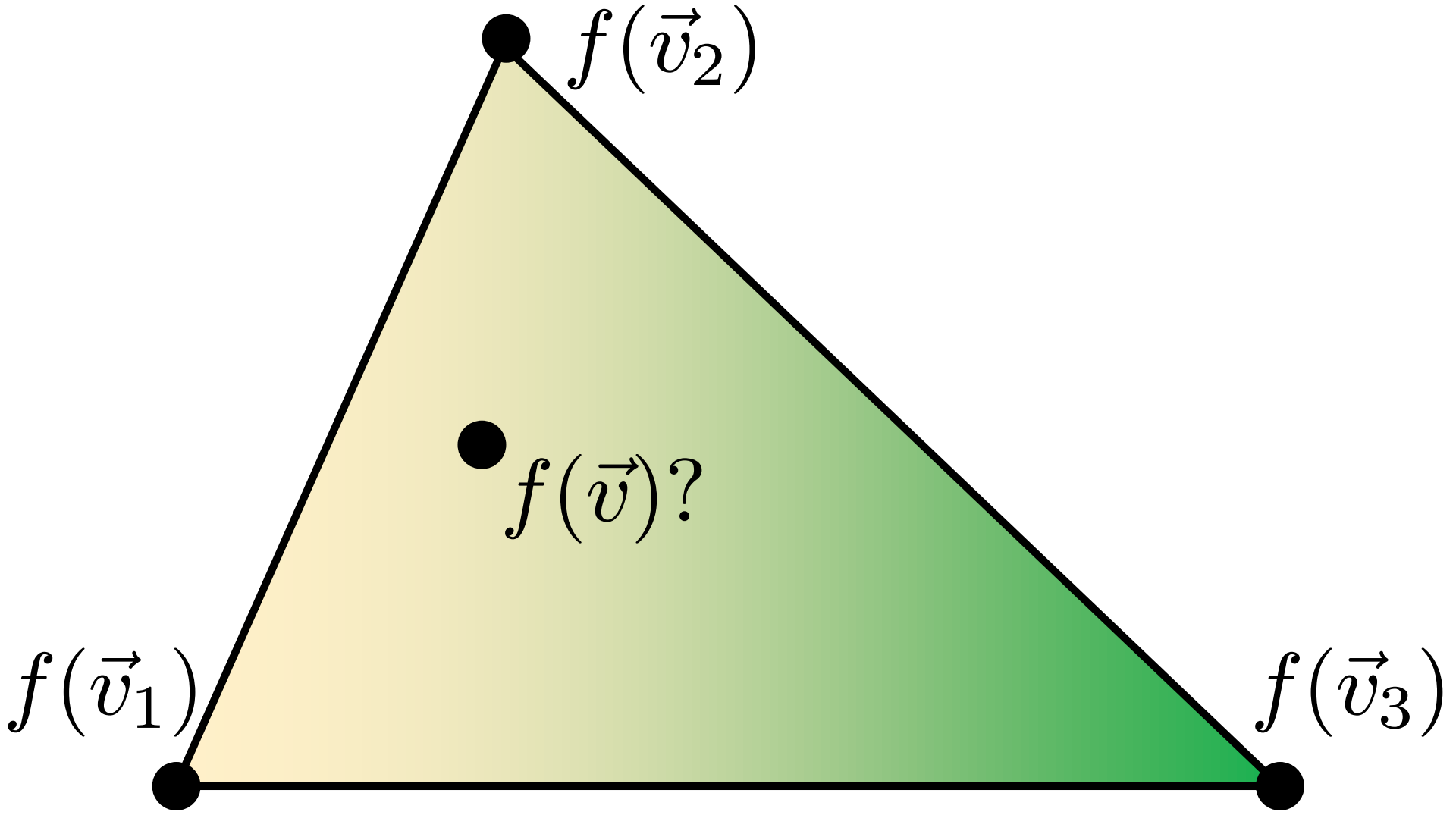
Flat shading is boring.



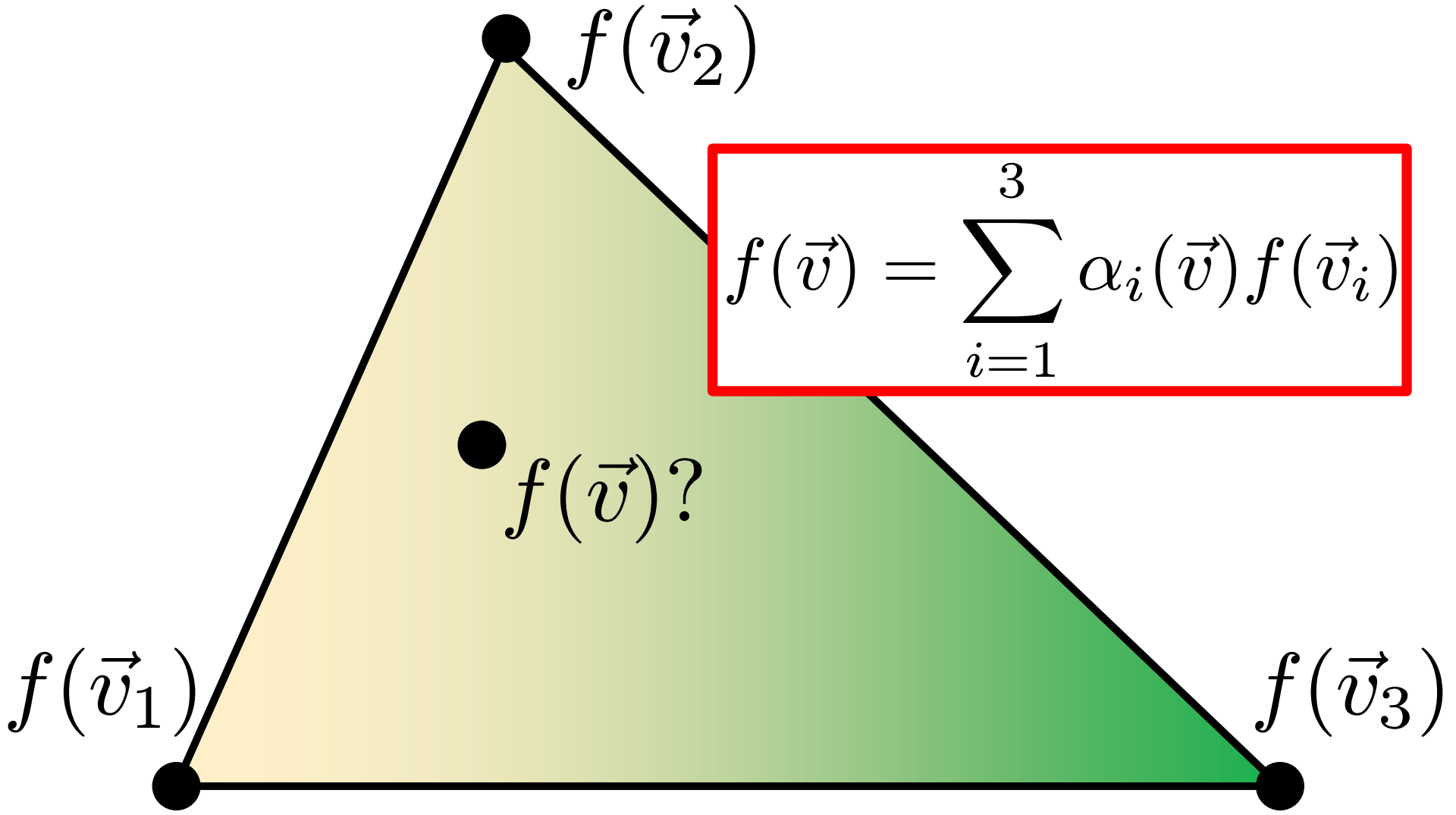
Depth values aren't flat.

http://www.implovator.com/wp-content/uploads/2010/11/teapot_flat.png
http://upload.wikimedia.org/wikipedia/commons/4/4e/Z_buffer.svg

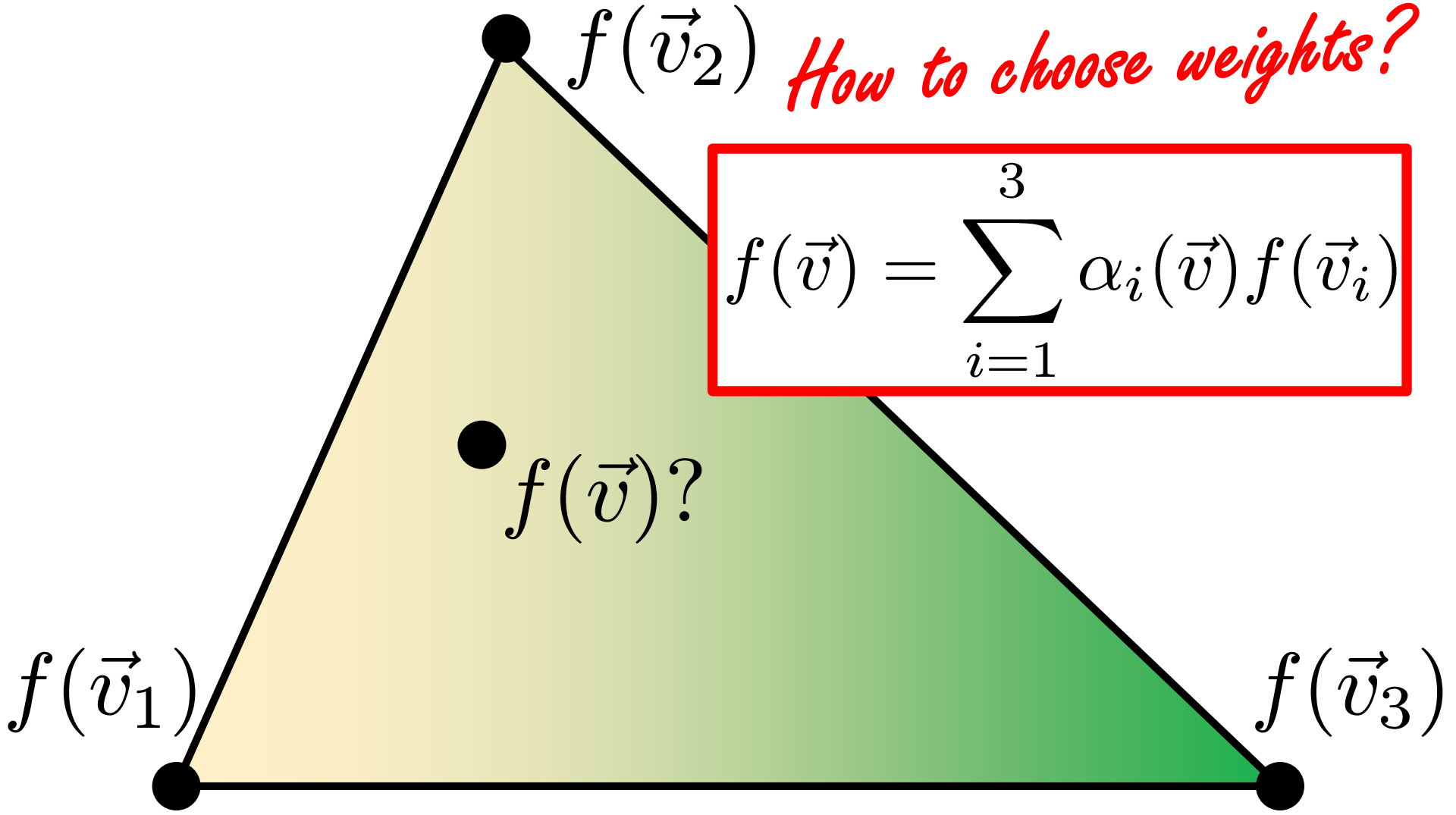
More Specific Problem



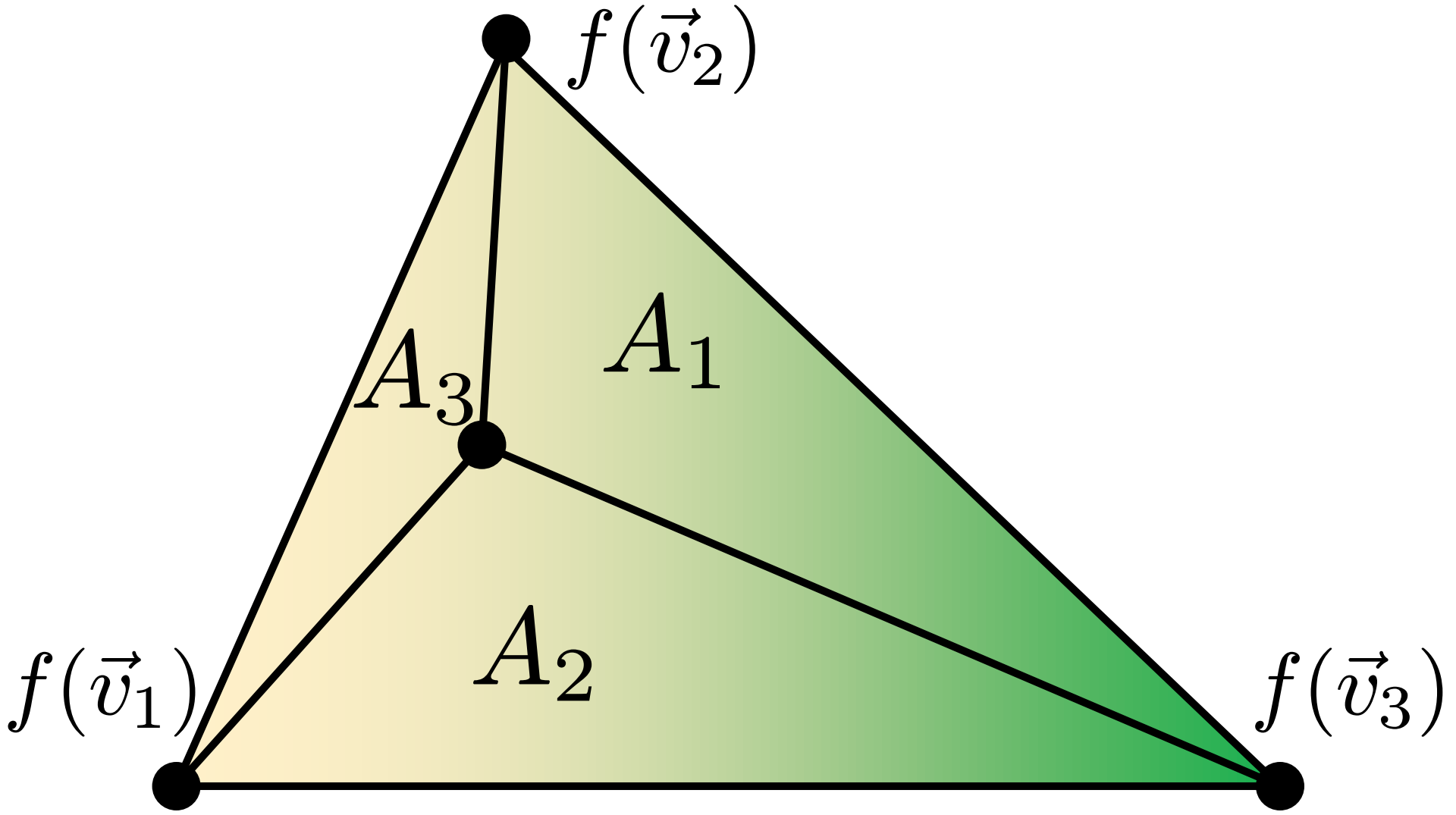
Interpolation



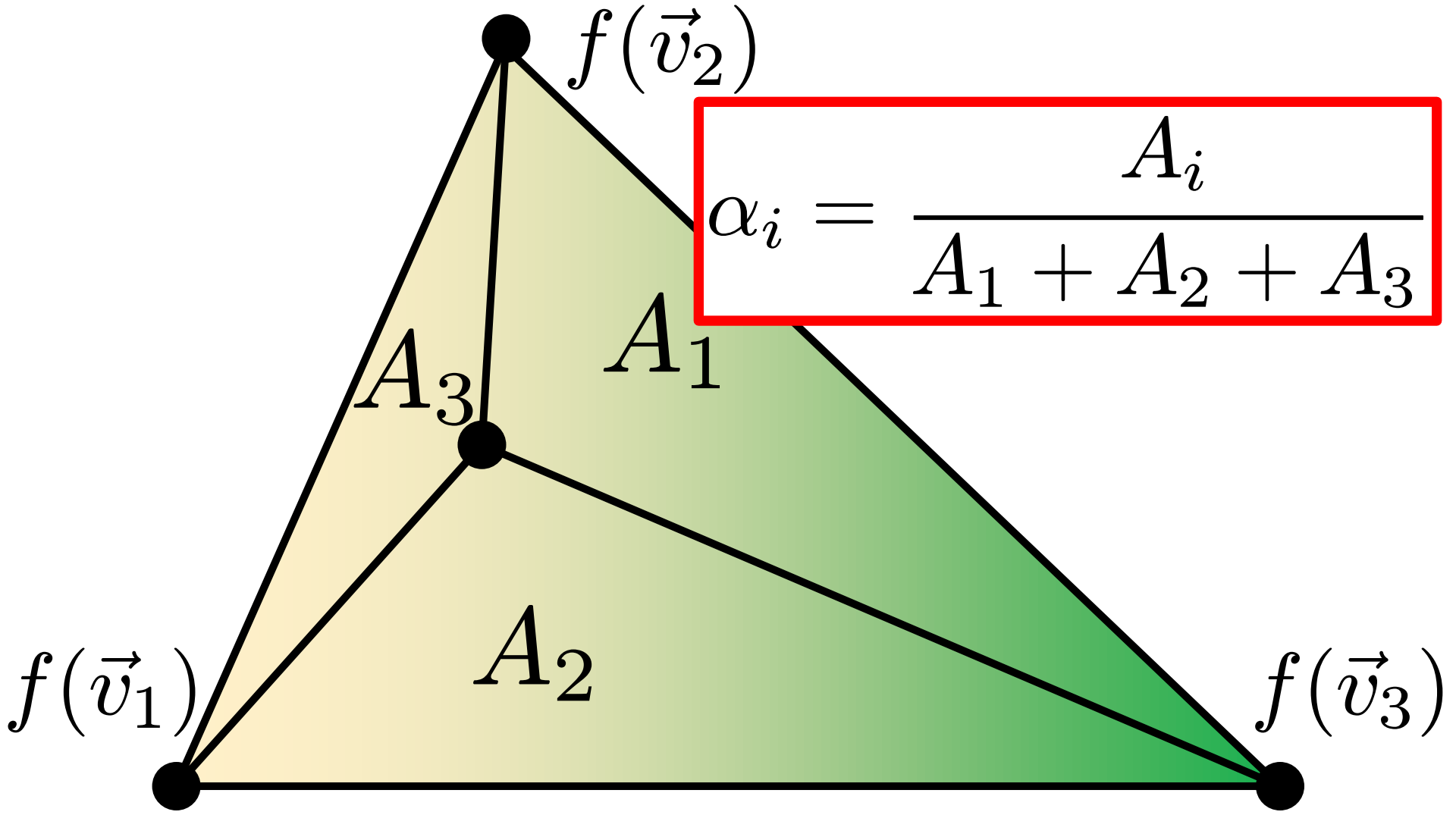
Interpolation



Barycentric Coordinates



Barycentric Coordinates



Barycentric Coordinates

Lots of nice properties:

$$\sum_{i=1}^{3} \alpha_i = 1$$

Barycentric Coordinates

Lots of nice properties:

$$\vec{v} \in \Delta(\vec{v}_1, \vec{v}_2, \vec{v}_3)$$

$$0 \cdot \quad \alpha_i(\vec{v}) \quad \cdot \quad 1 \; \forall i$$

Barycentric Coordinates

Lots of nice properties:

Can compute using linear solves, cross products, …

Barycentric Coordinates

Lots of nice properties:

Right way to interpolate depths

Barycentric Coordinates

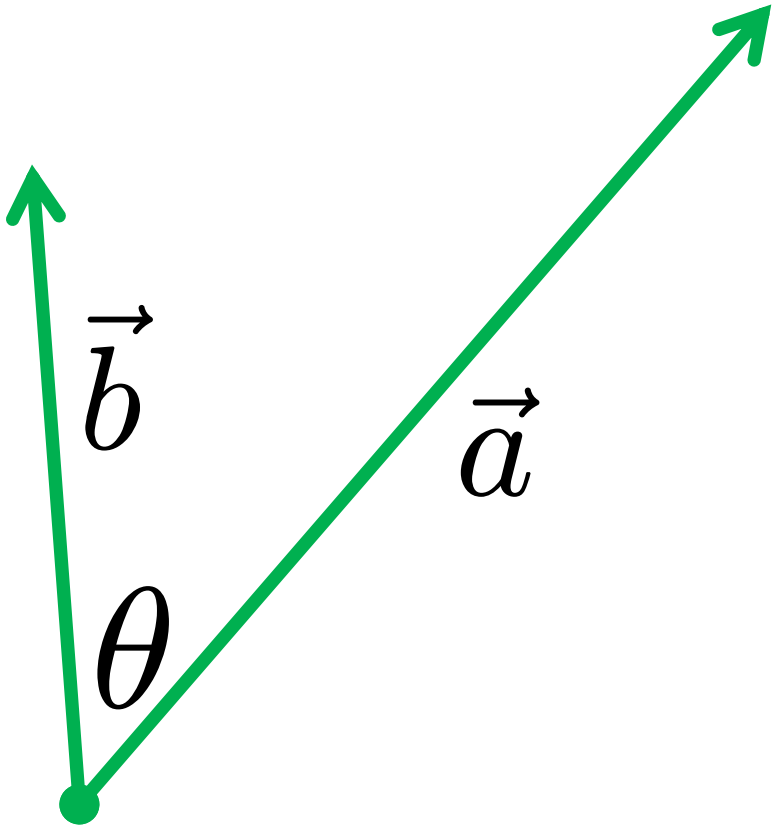
Lots of nice properties:

Right way to interpolate depths

Homework 3

Reminder: Dot Product

$$\vec{a} \cdot \vec{b} = \|\vec{a}\| \|\vec{b}\| \cos\theta = a_x b_x + a_y b_y + a_z b_z$$

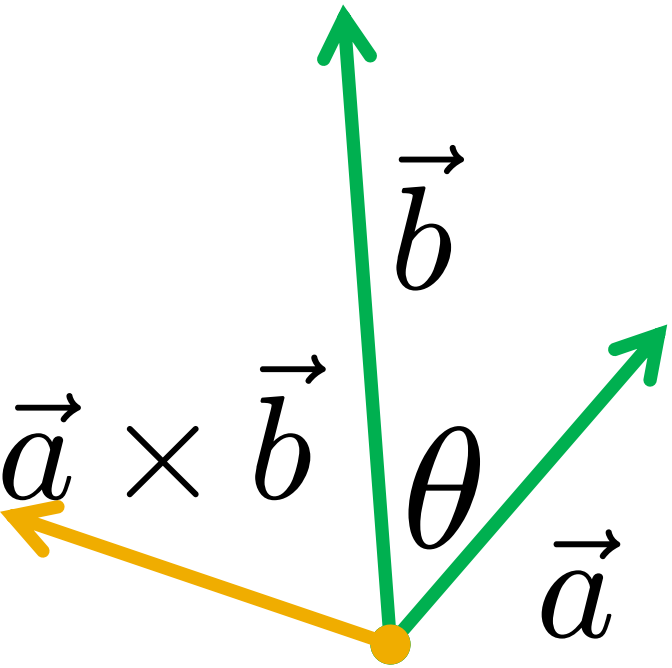


Used for:

- Projecting vectors onto other vectors
- Checking orthogonality

Reminder: Cross Product

$$\vec{a} \times \vec{b} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix}, \|\vec{a} \times \vec{b}\| = \|\vec{a}\| \|\vec{b}\| |\sin\theta|$$



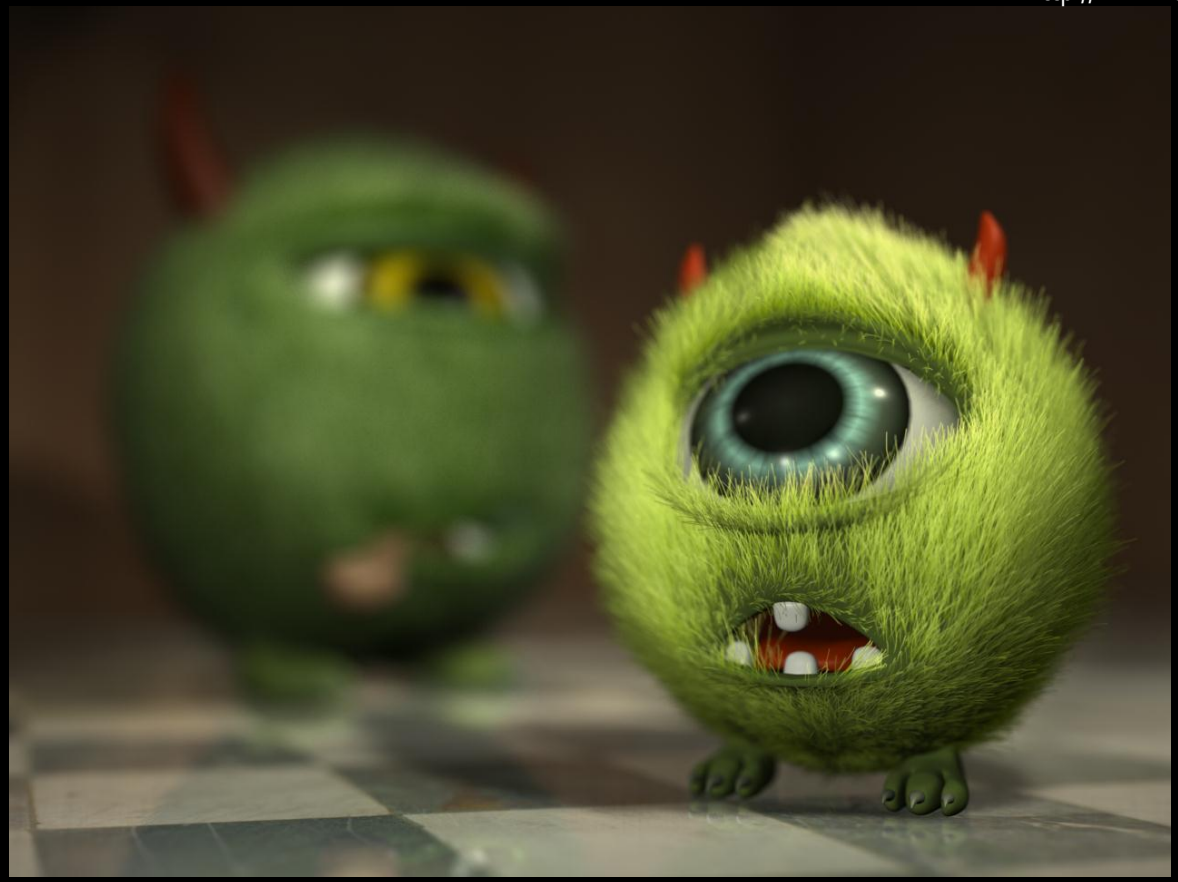
Used for:

- Finding normals
- Checking if vectors are parallel

What We Can Do Now

Rasterize 3D scenes with occlusion and per-vertex shading.

http://www.kunzhou.net/2010/mptracing.pdf



Basics of 3D Rendering

CS 148, Summer 2012
Introduction to Computer Graphics and Imaging

Justin Solomon