Topic 9:

Visibility

- Elementary visibility computations:

  Clipping

  Backface culling

- Algorithms for visibility determination

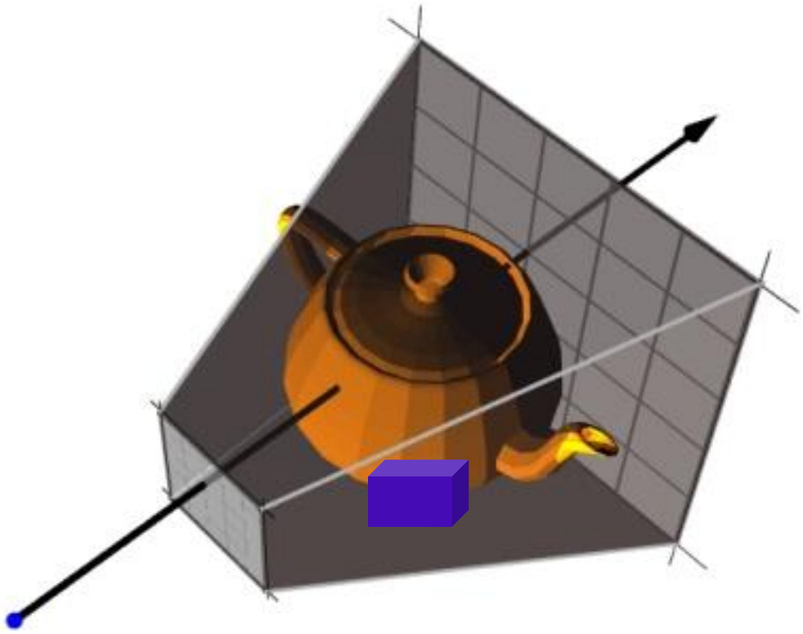
  Z-Buffer

  Painter’s algorithm

  Space partitions: BSP, AABB, OOBB, octrees

Visibility Problem

What is NOT visible?



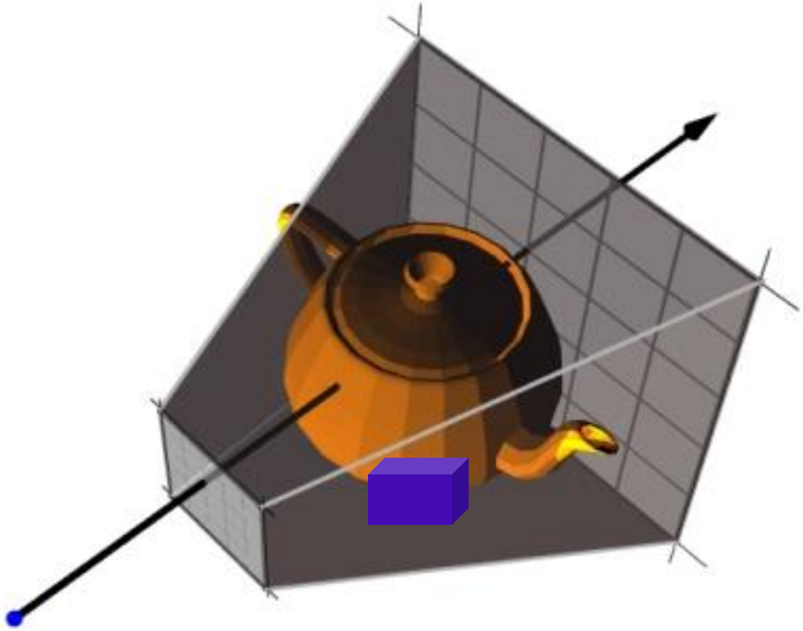
Visibility Problem

What is NOT visible?

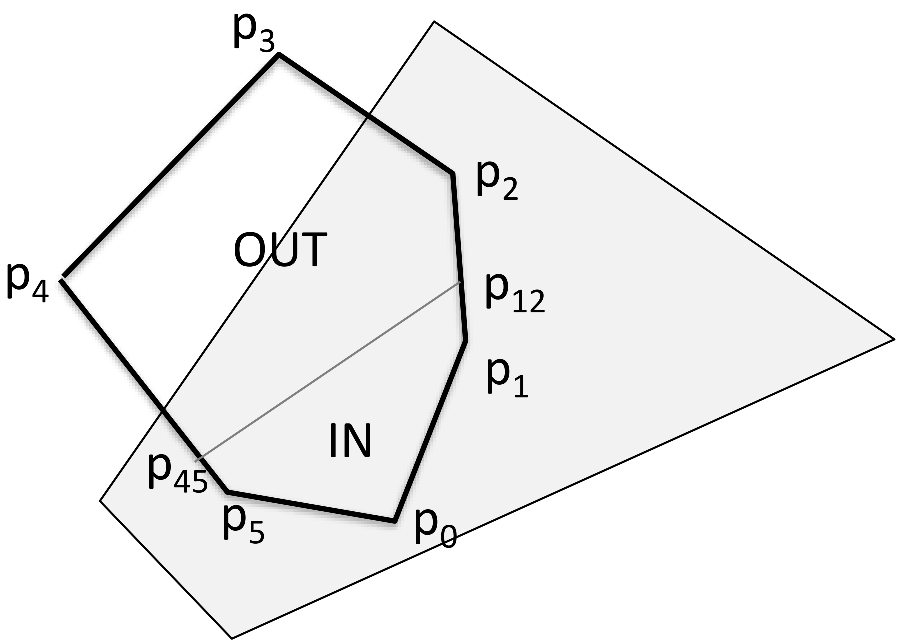
primitives outside of the field of view

back-facing primitives

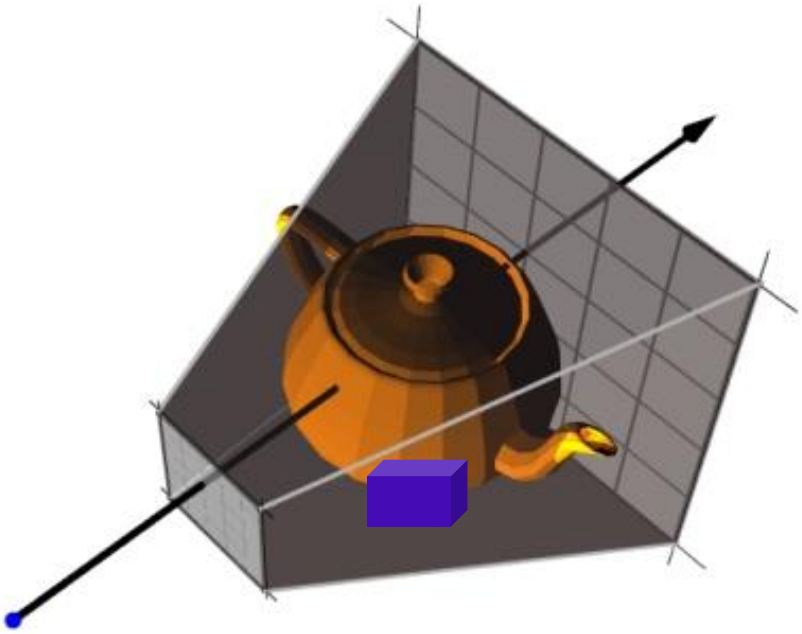
primitives occluded by other objects closer to the camera



Polygon Clipping (wrt to a single plane)



Input	Output
edge (p_k, p_{k+1})	
in, in	p_{k+1}
in, out	$p_{intersect}$
out, out	
out, in	$p_{intersect}$, p_{k+1}



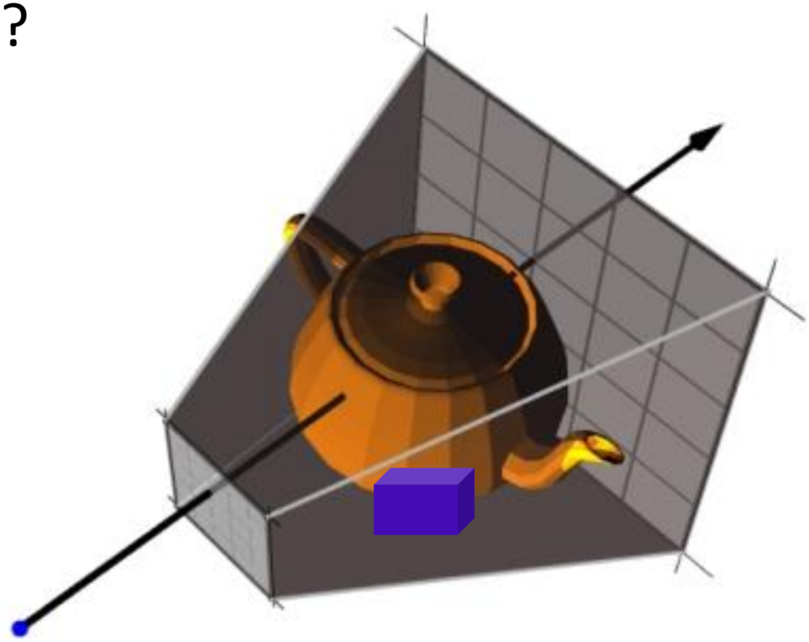
Polygon Clipping (wrt to a volume)

Clip with respect to each plane of the volume in sequence!

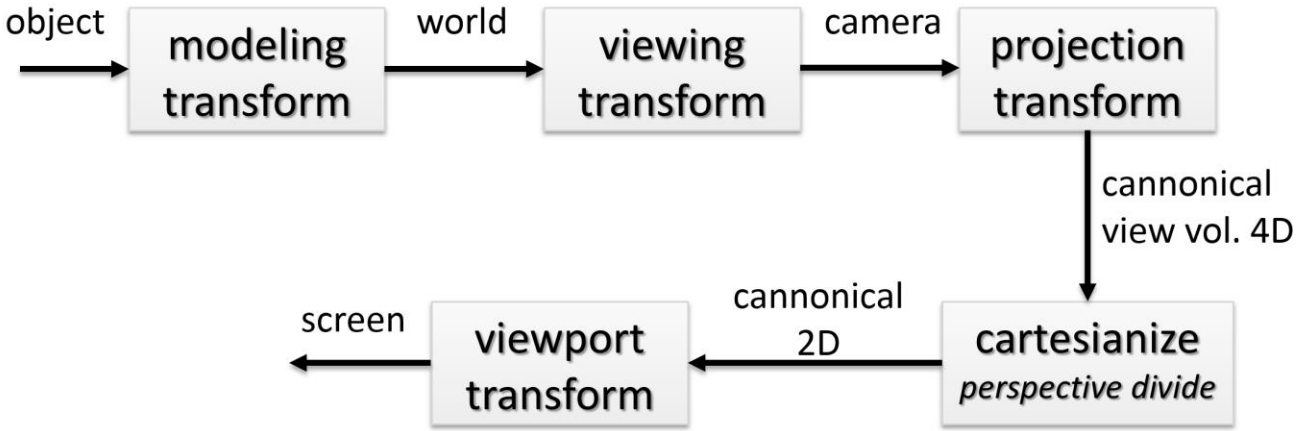
Does the order of the planes matter?

Does it work for concave polygons?

Does it work for concave volumes?



Polygon Clipping (when to clip?)



object

modeling transform

world

viewing transform

camera

projection transform

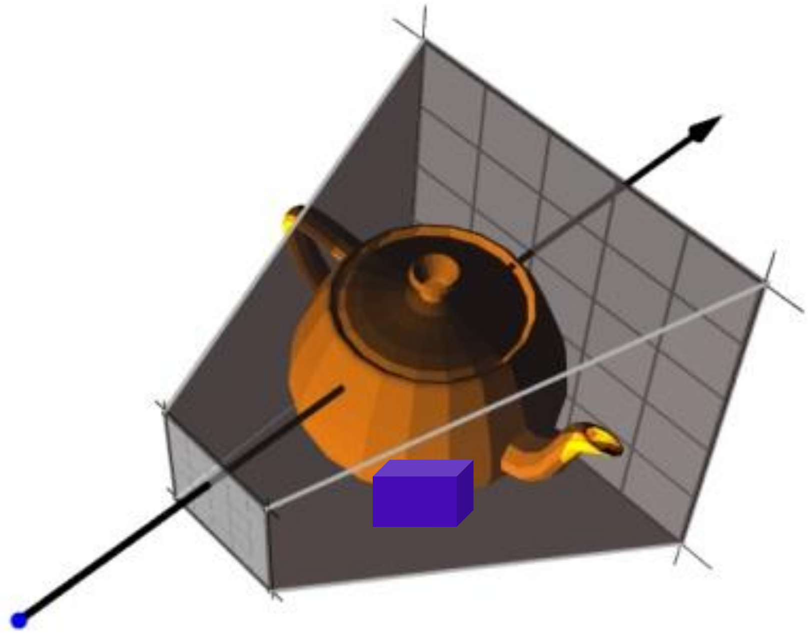
cannonical view vol. 4D

cartesianize *perspective divide*

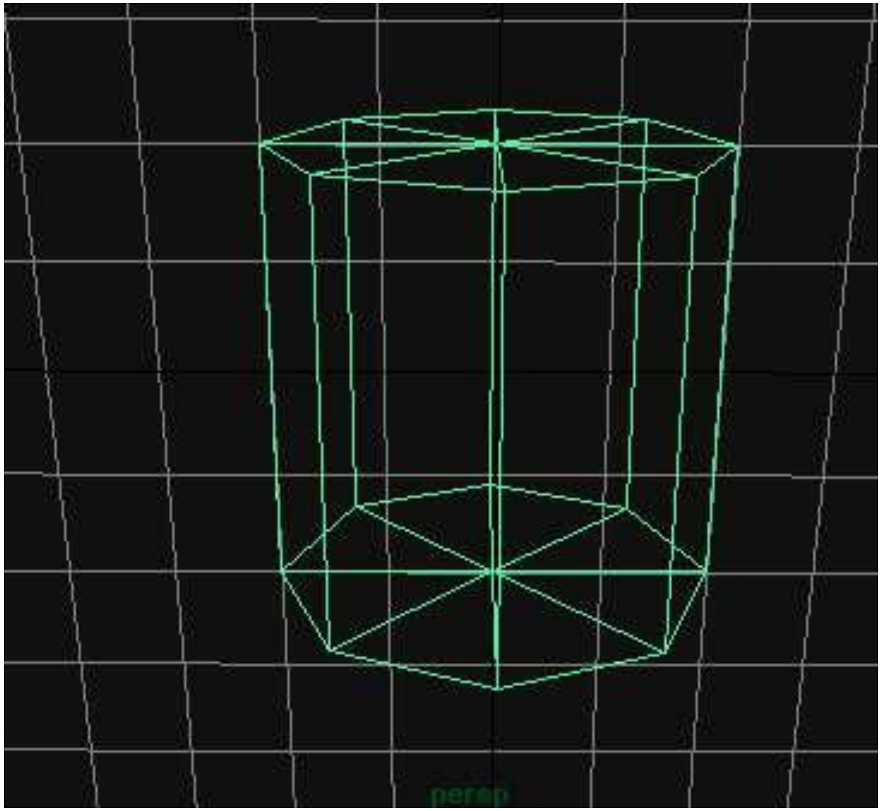
cannonical 2D

viewport transform

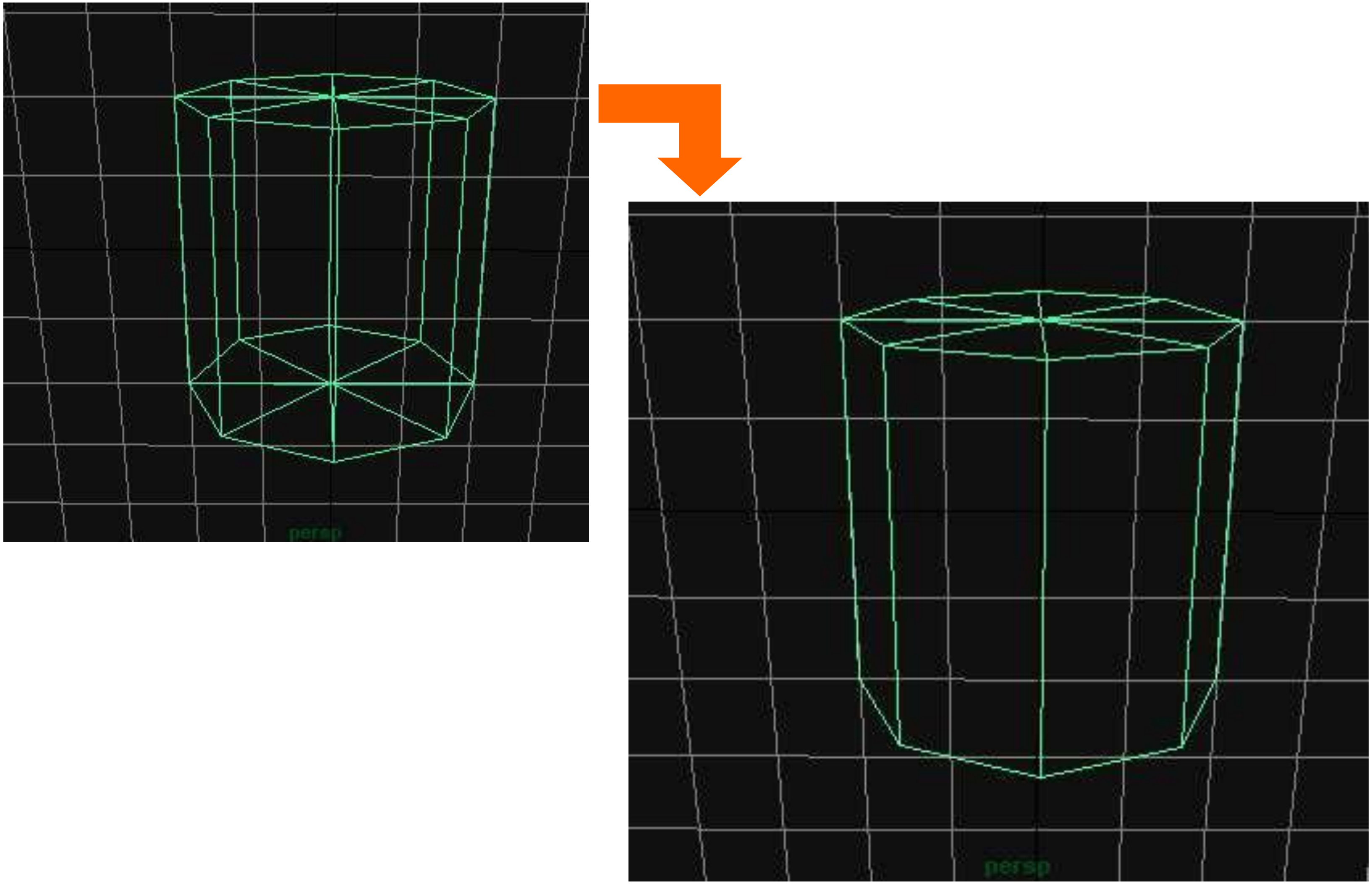
screen



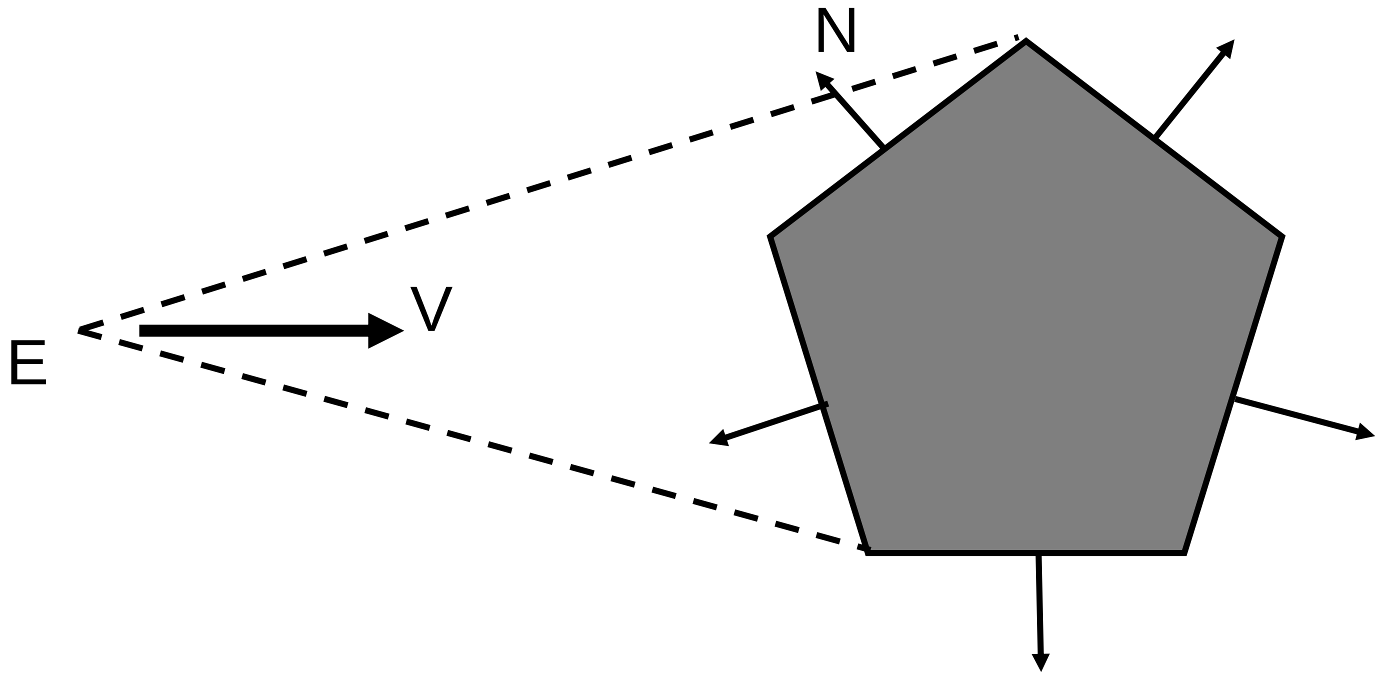
Backface culling



Backface culling

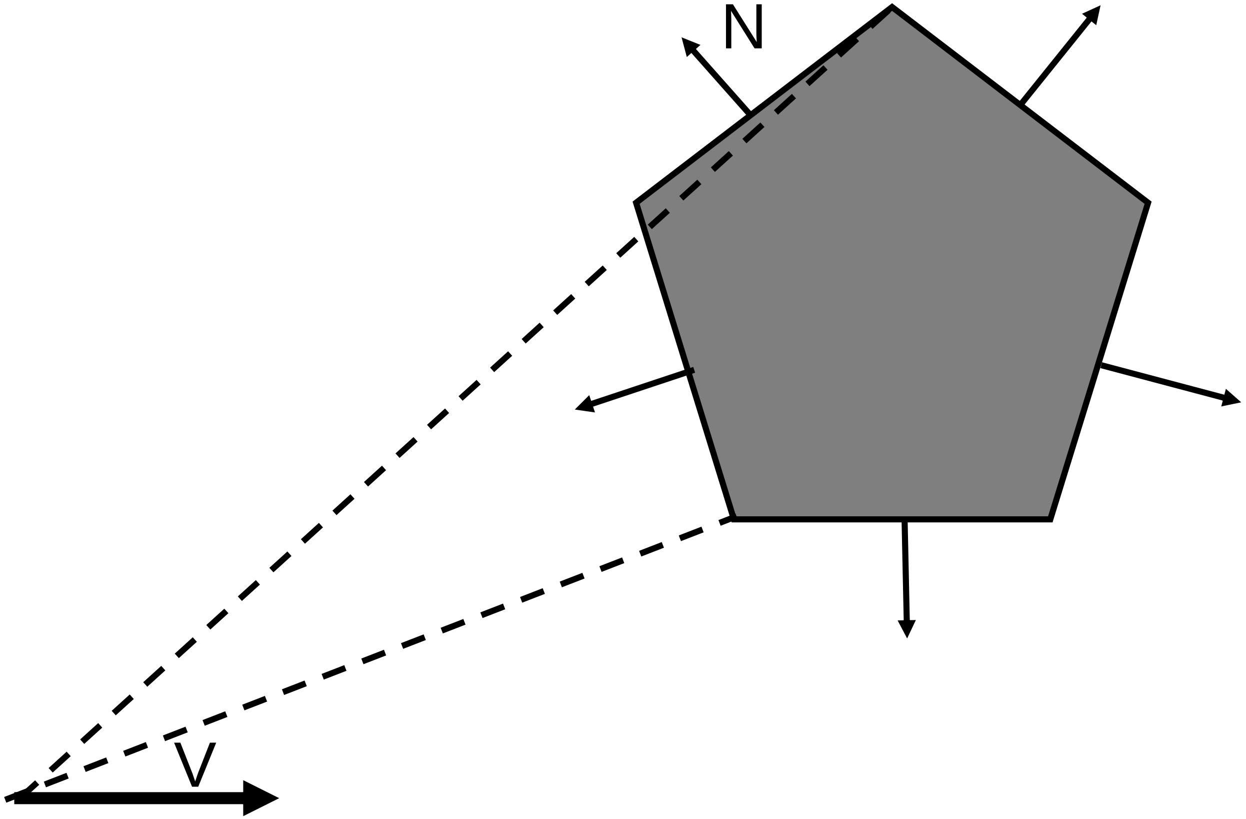


Backface culling



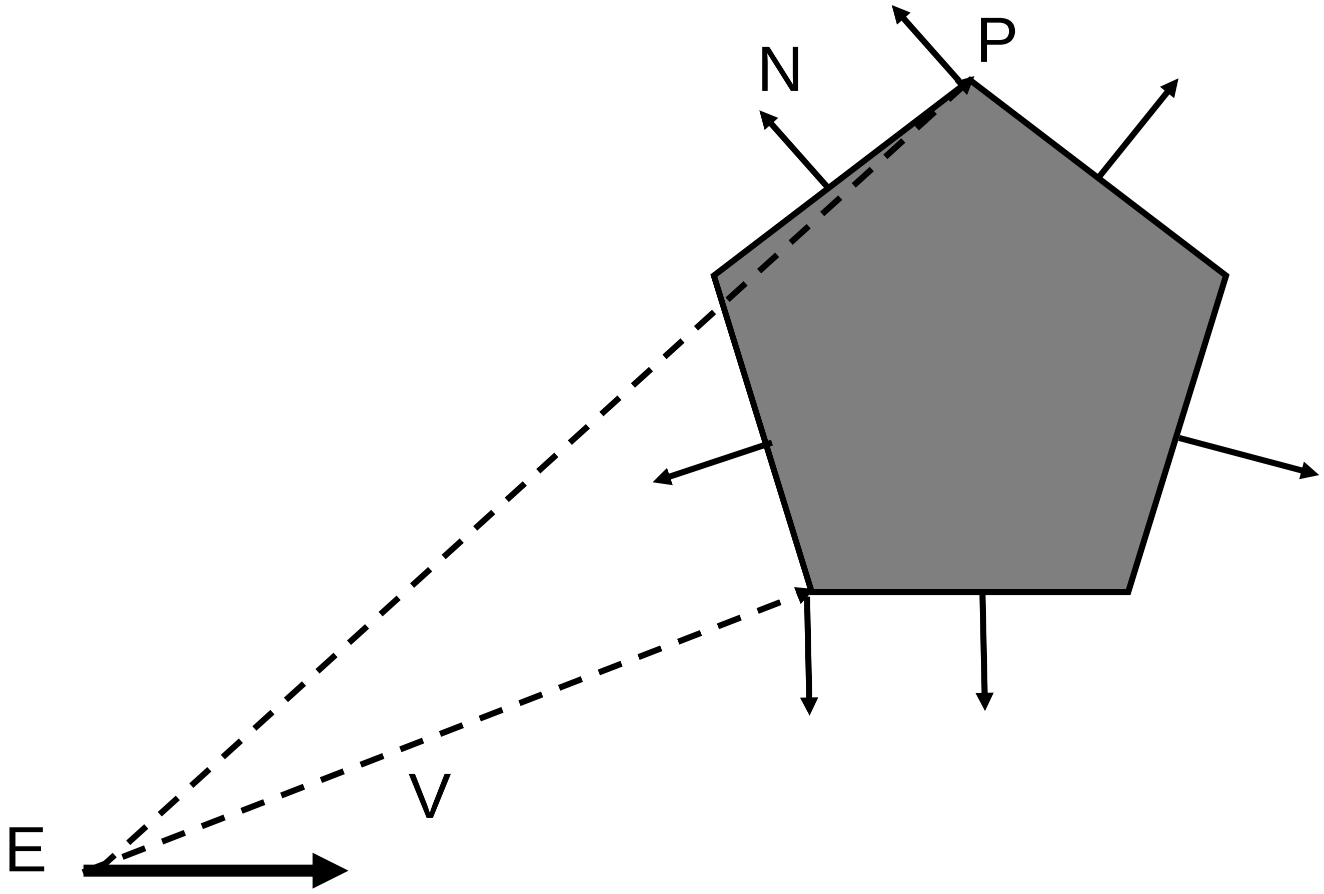
Backface culling

N.V > 0 is a back face?



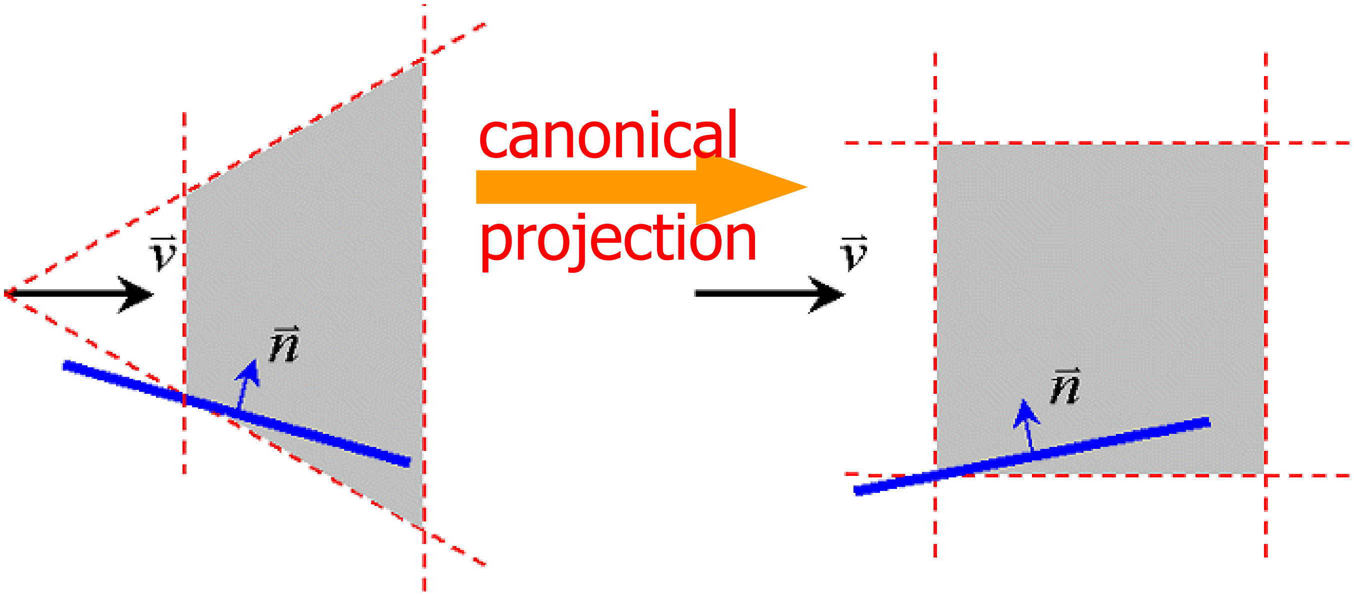
Backface culling

N.(P-E) >0



Backface culling (when to cull?)

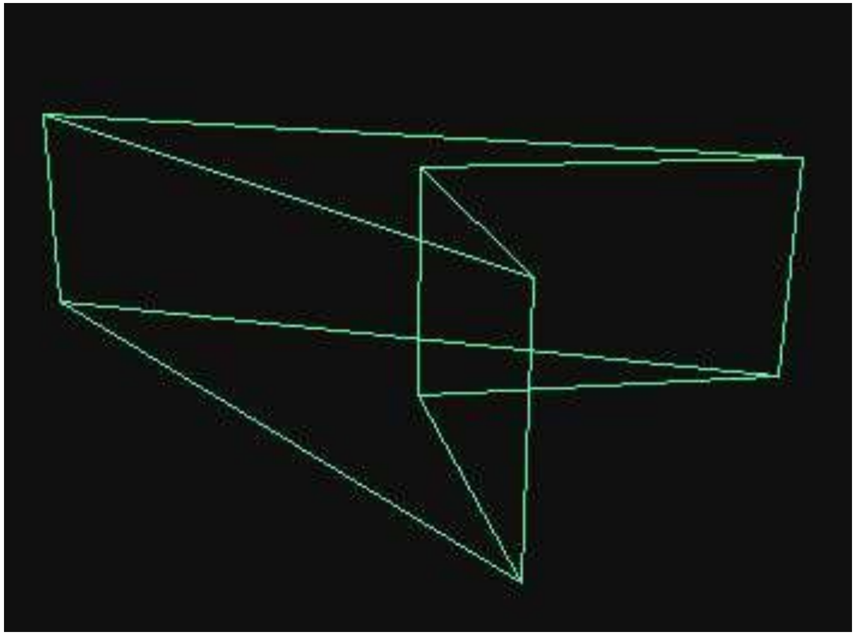
Where in the graphics pipeline can we do backface culling?


canonical projection

@alec: Would be nice to redo this image

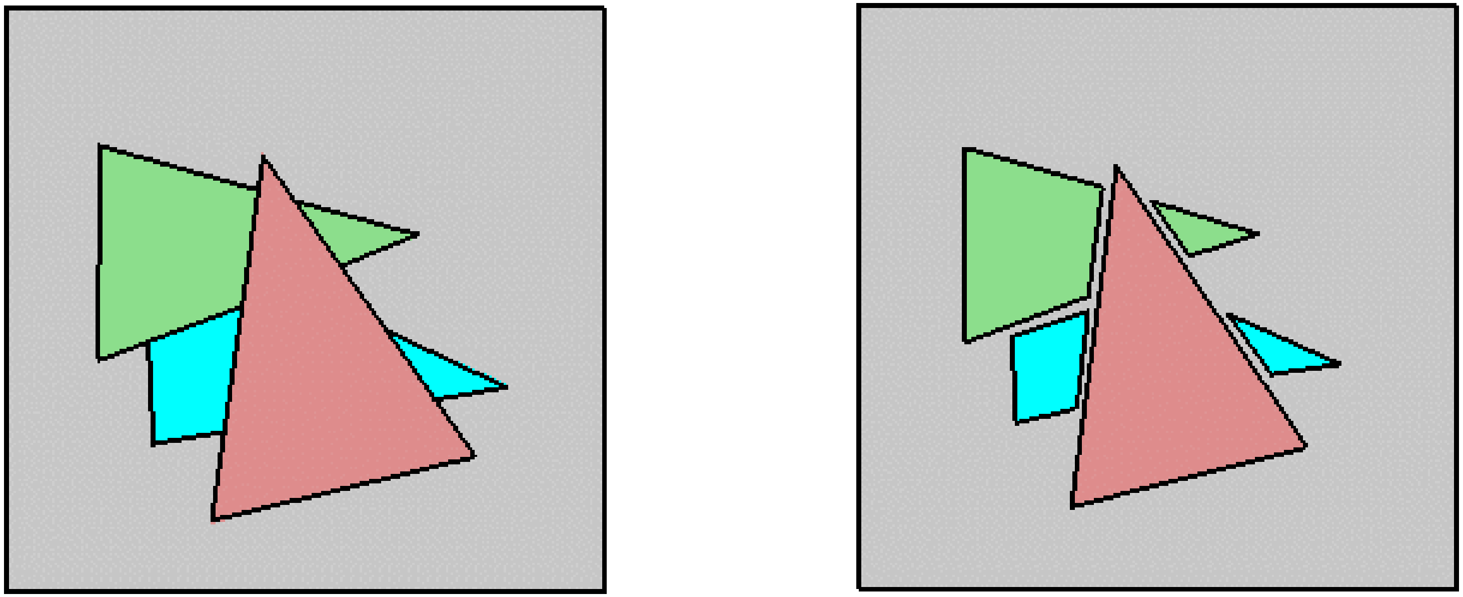
Occluded faces

Does backface culling always determine visibility completely for a single object?



Occluded faces

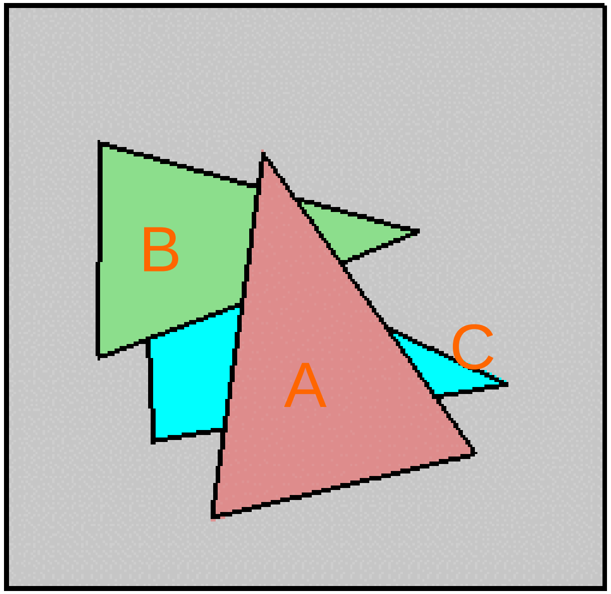
In typical scenes some polygons will overlap, we must determine which portion of each polygon is visible to eye!

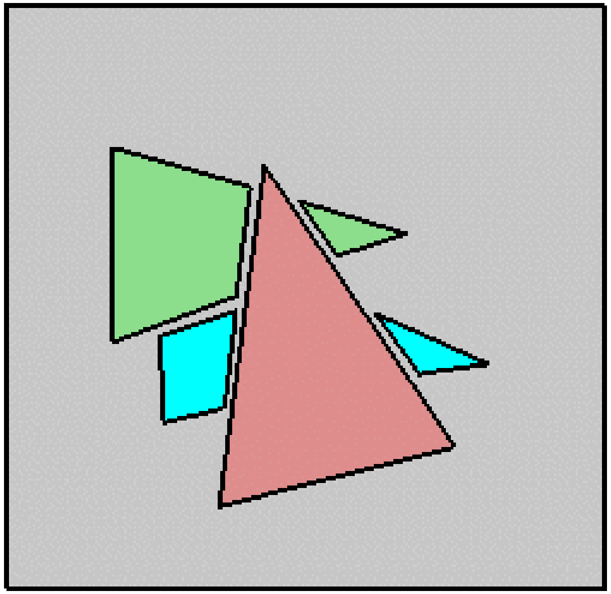


Painters Algorithm

Sort primitives in Z.

Draw primitives back to front (CBA).

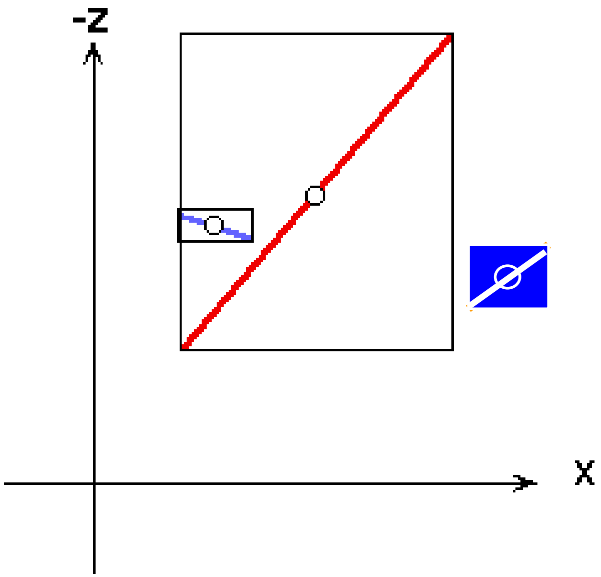




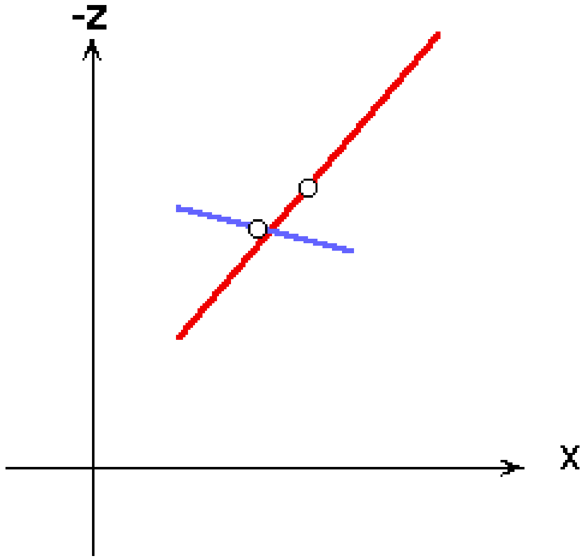
Painters Algorithm

Problems

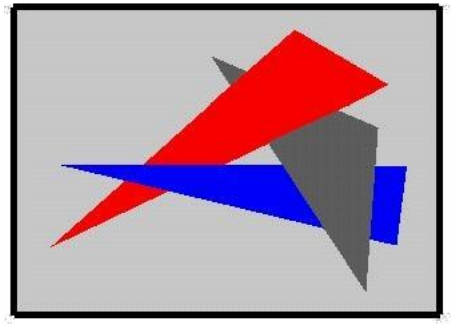
- Large faces



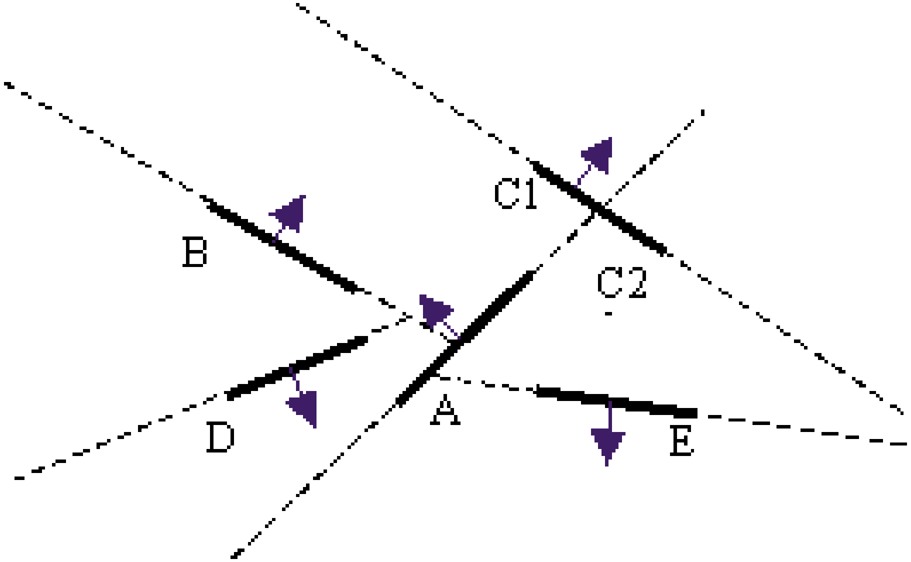
- Intersecting faces

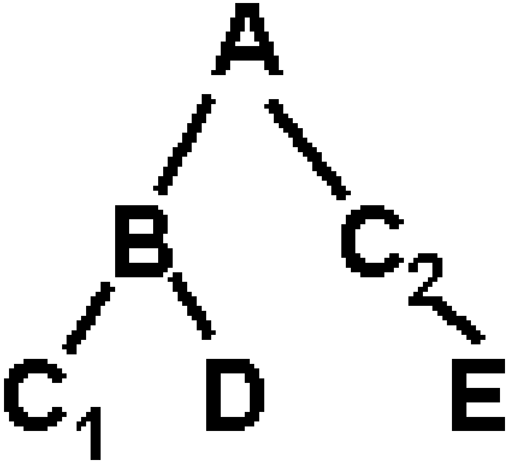


- Cycles

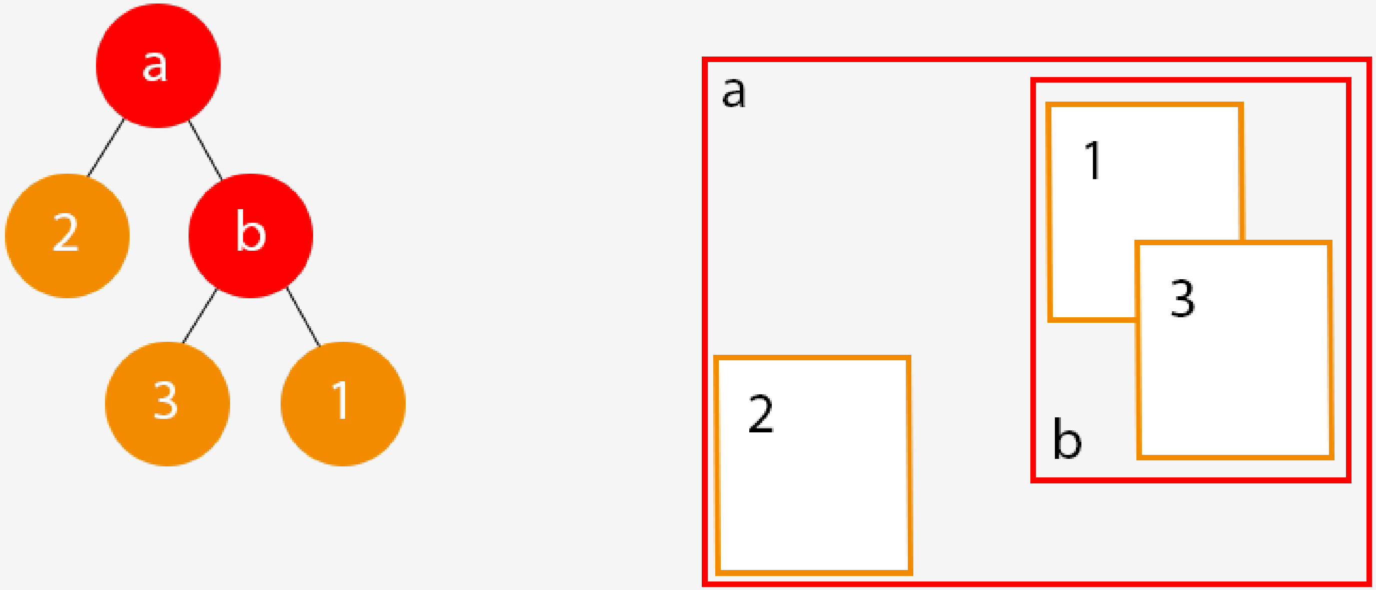


BSP tree

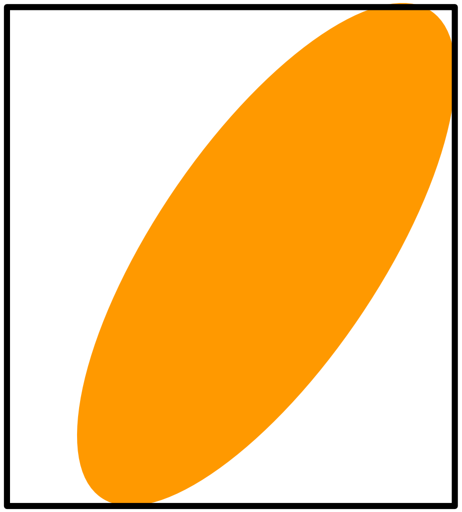




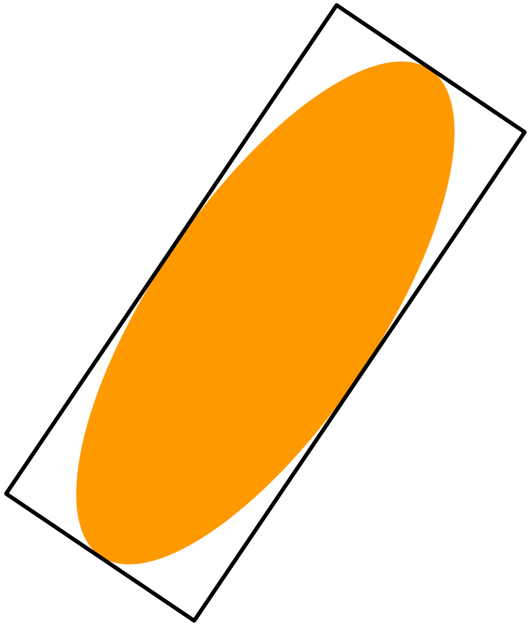
AABB tree



OBB

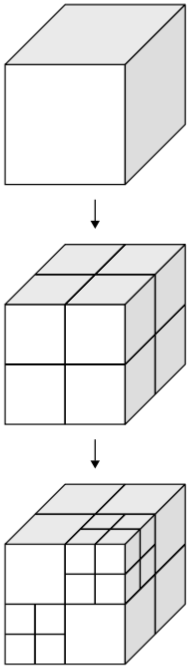


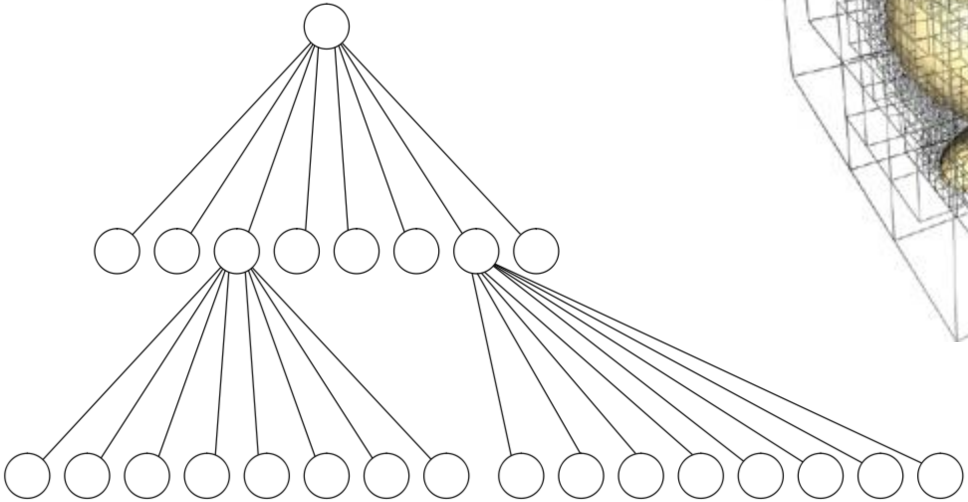
AABB

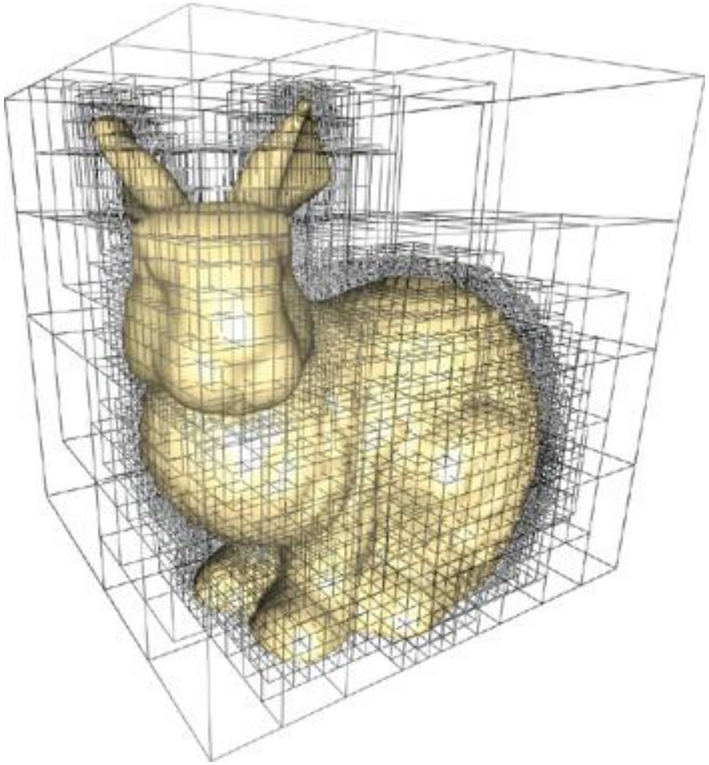


OBB

Octree



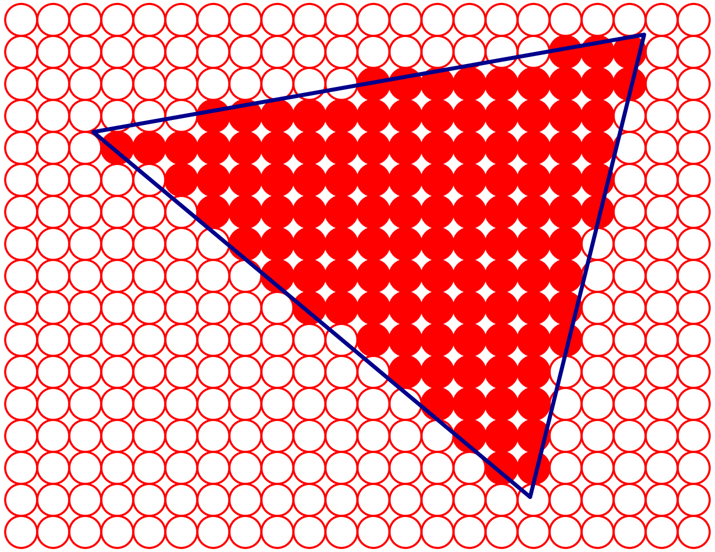


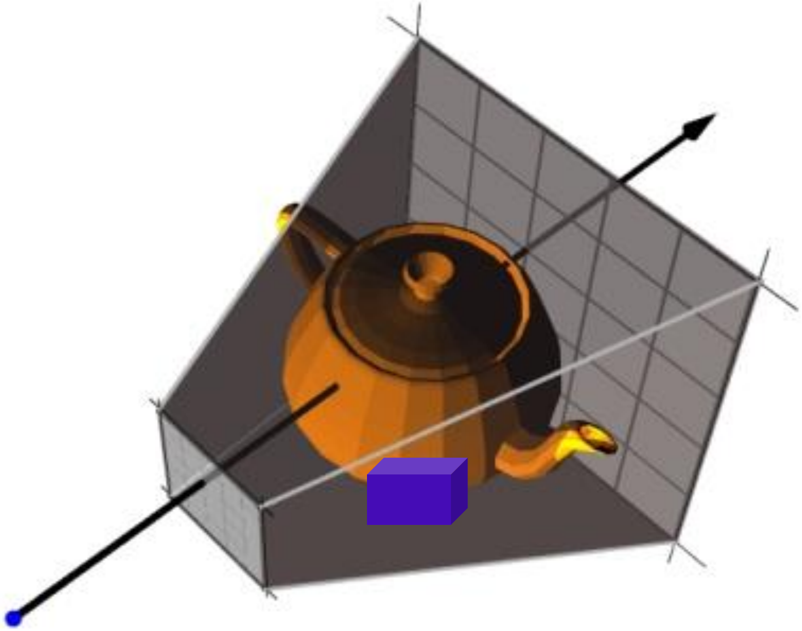


Visibility Problem: Z-buffer, A-buffer

Z-buffer: rasterize each polygon in the scene, keeping track of the polygon closest to the eye at each pixel.

A-buffer: accumulate pixel contribution to handle transparent polygons.





Visibility Algorithms

Image space algorithms

- Operate in display terms pixels.
- Visibility resolved to display resolution
- Examples: Z-buffer, ray-tracing
- O(n*resolution)

Object Space algorithms

- Analytically compute visible fragments
- Examples: painters algorithm, BSP
- $O(n^2)$