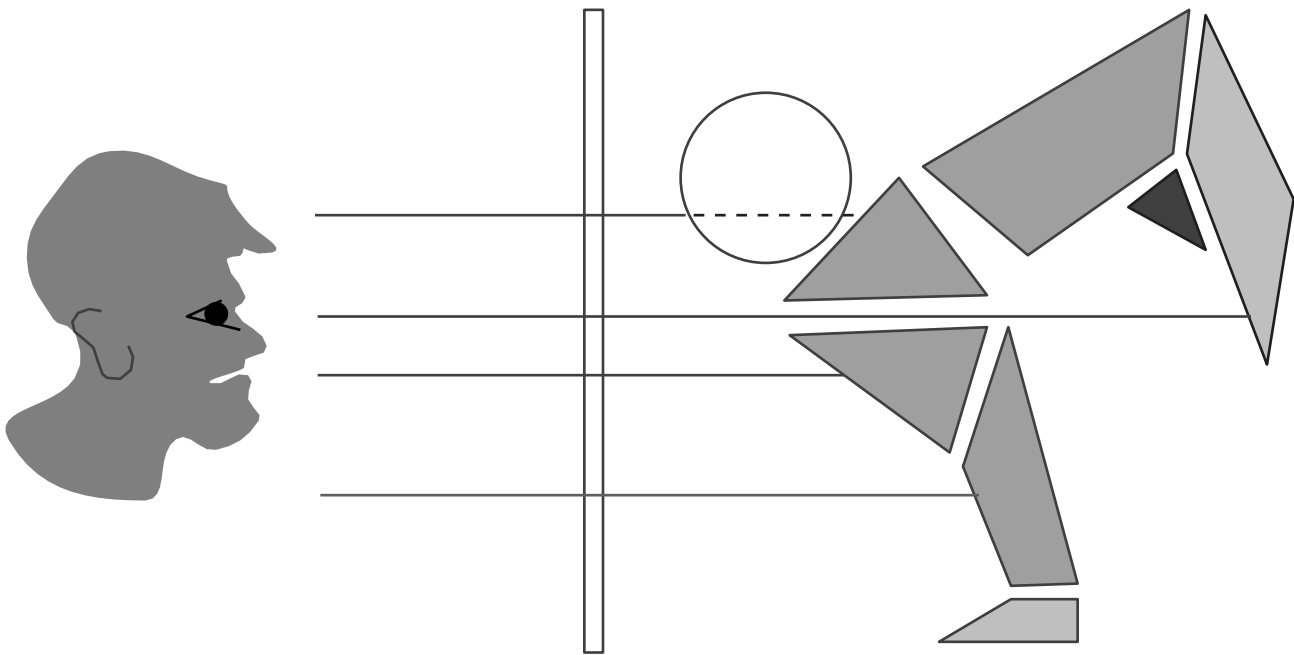


Visible–Surface Detection

Identifying those parts of a scene that are visible from a chosen viewing position, and only process (scan–convert) those parts



Two approaches:

1. Object–space methods
2. Image–space methods

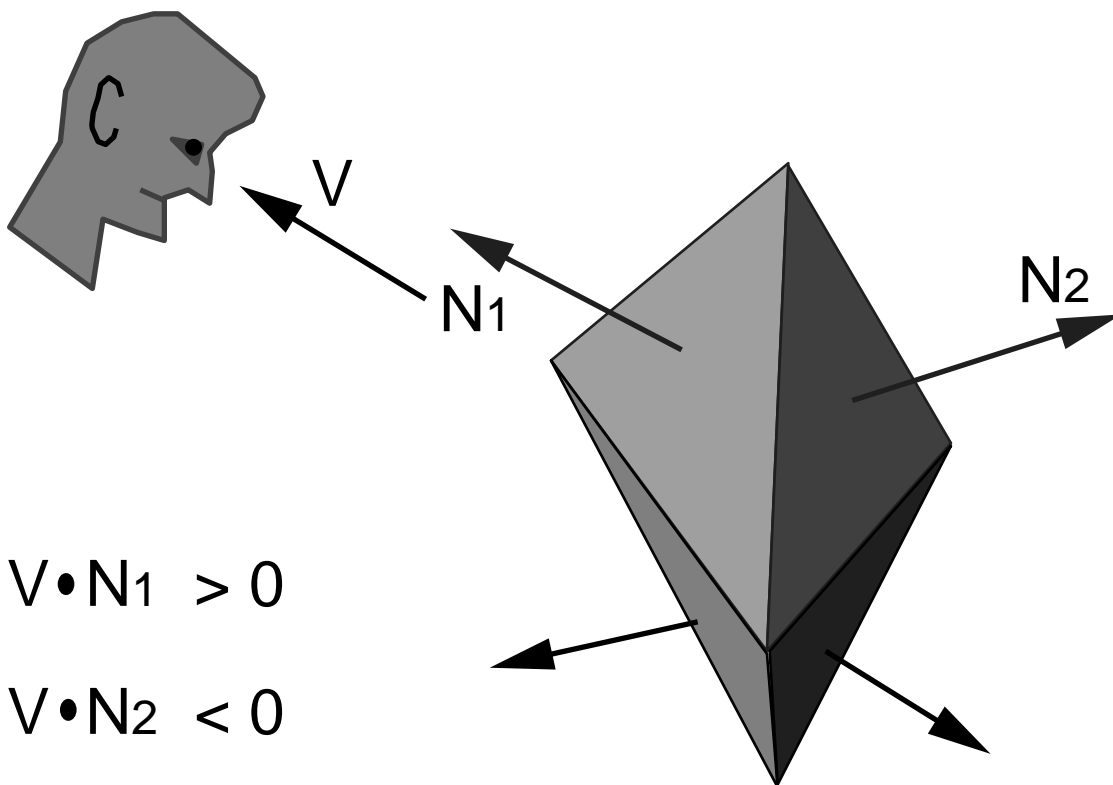
Two main strategies:

1. Sorting
2. Coherence

Visible –Surface Detection Algorithms:

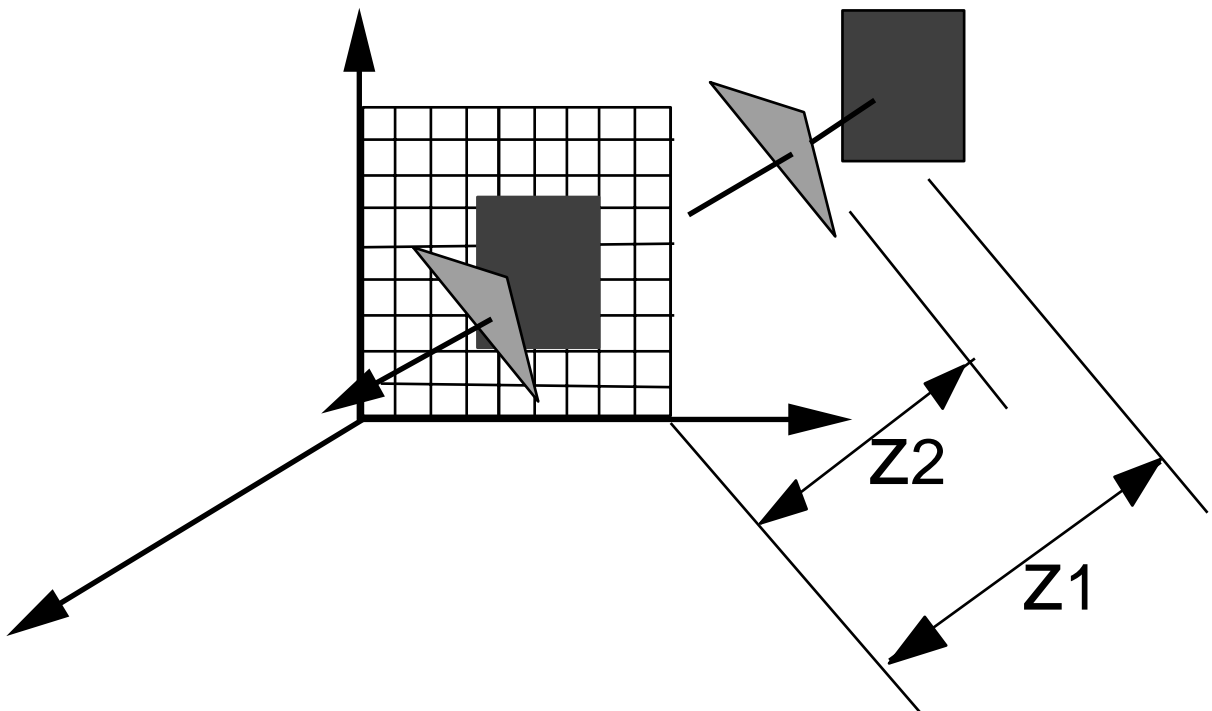
1. Back–face detection
2. Z–buffer
3. A–buffer
4. Scan–line method
5. Painter’s algorithm
6. BSP–tree method
7. Area–subdivision method
8. Octree methods
9. Ray–casting method
10. Wireframe methods

Back–face detection



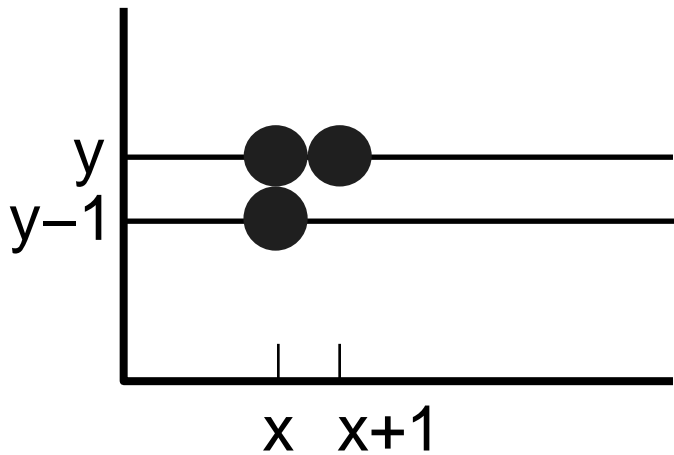
Z-buffer

- most widely used
- simple to implement
- use a depth buffer with the same resolution as the image buffer
- scanconvert each polygon and compare the z value of each resulting pixel with the value stored in the z-buffer
- Update the image buffer only when the new z value is smaller than the saved z value



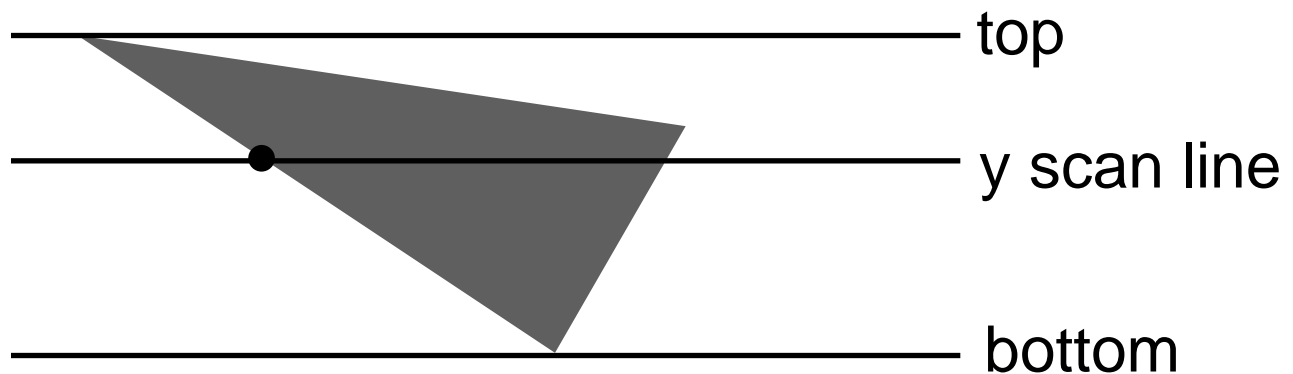
Z-buffer (cont'd)

z values can be calculated incrementally



$$Ax + By + Cz + D = 0$$

$$z' = \frac{-A(x+1) - By - D}{C} = z - \frac{A}{C}$$



$$x' = x - \frac{1}{m}$$

$$z' = \frac{-A(x - (1/m)) - B(y-1) - D}{C} = z + \frac{A/m + B}{C}$$

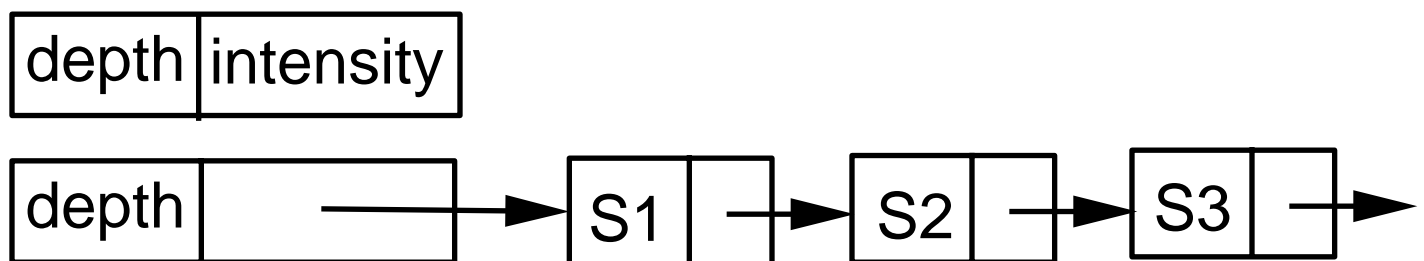
Z-buffer (cont'd)

- Large amount of memory space is needed.
What to do if short of memory space?
- Aliasing problem!
- Handle only opaque surfaces
- Other use of z-buffer

A-buffer

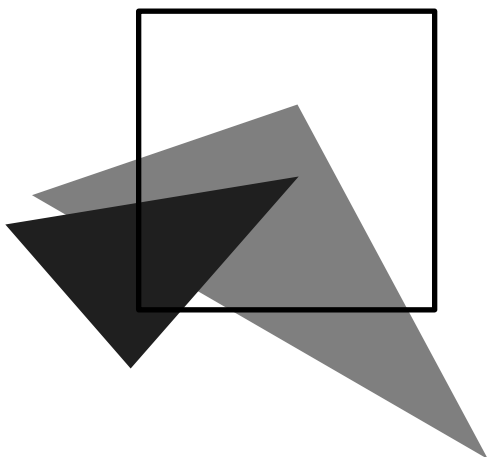
(antialiased, area-average, accumulation buffer)

- deal with non-opaque surfaces
- more than one surface intensity can be taken into consideration at each pixel position



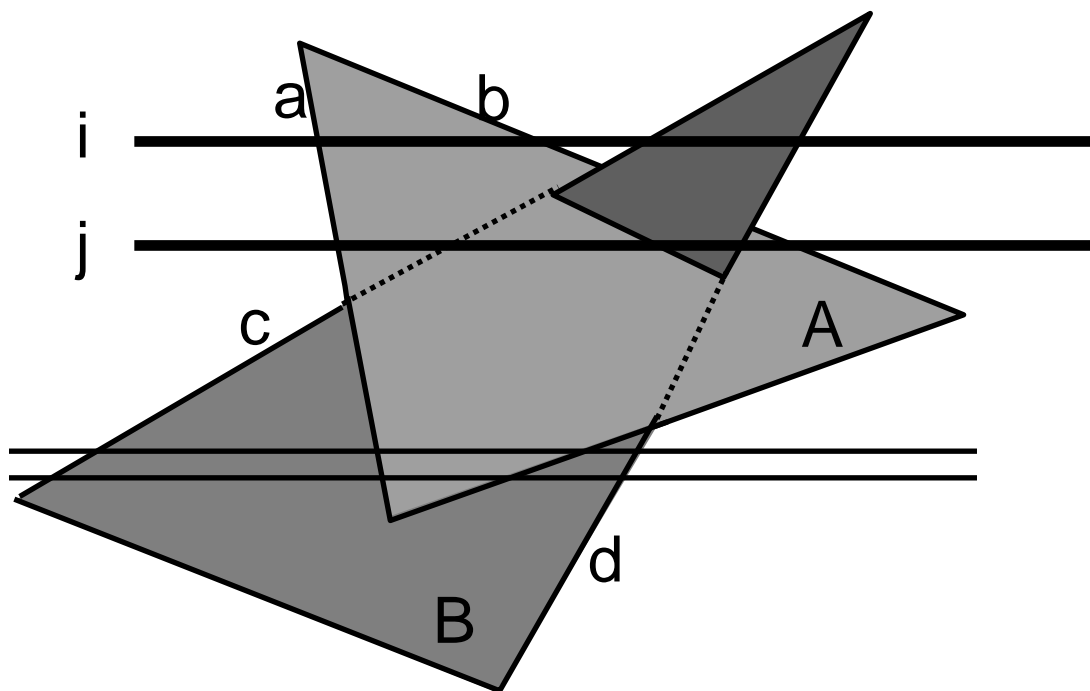
Each surface data field includes:

- RGB intensity components
- opacity parameter (percent of transparency)
- depth
- surface identifier
- percent of area coverage
- other surface-rendering parameters



Scan-Line Algorithm:

- combine polygon scan conversion with hidden surface removal
- scanline by scanline rather than polygon by polygon



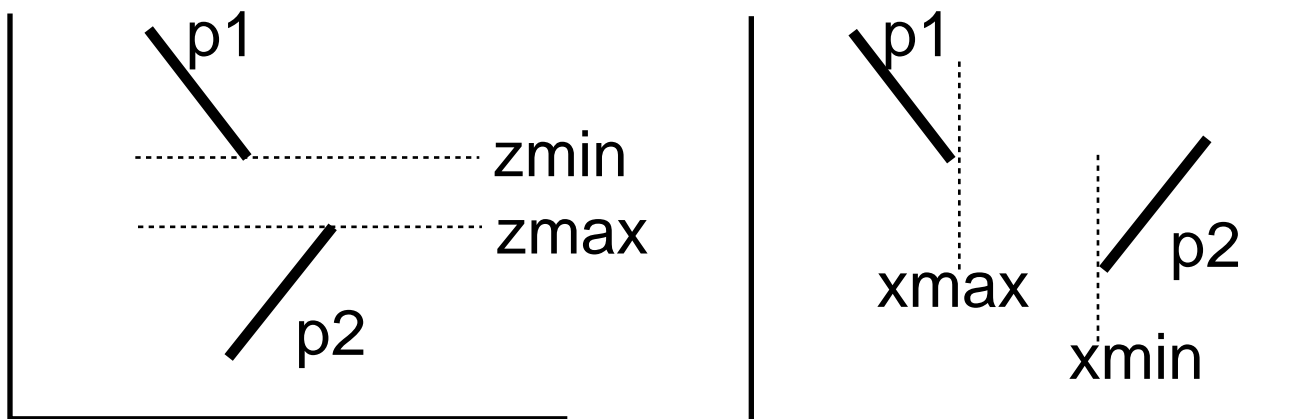
- perform depth calculation only when more than one polygon intersects with the current scanline
- require sophisticated data structures to quickly locate the polygon/edges intersecting with current scanline

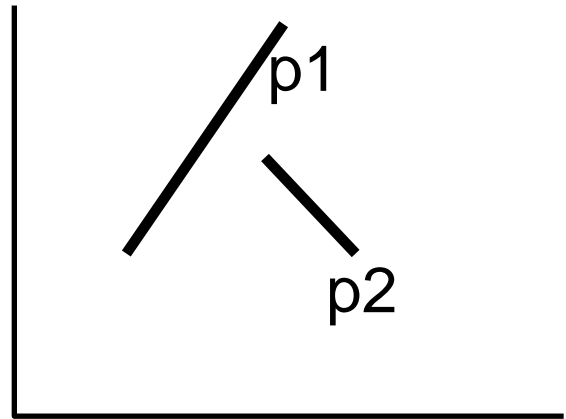
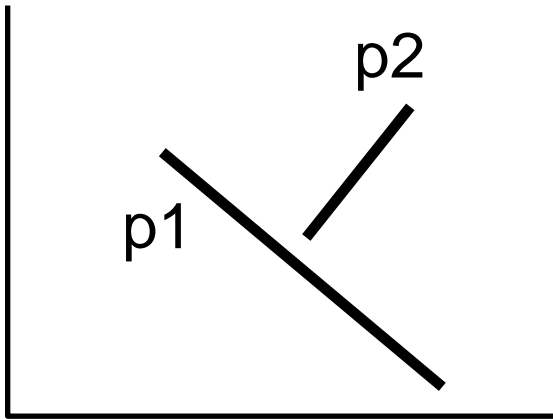
Painter's algorithm:

- depth sort the polygons
- an object space algorithm
- analogous to the way an oil painter might render a scene
- scan-convert polygons in back-to-front order

How to depth sort?

1. sort all polygons according to the minimum z coordinate of each
2. resolve any ambiguities this may cause when polygons' z extents overlap (splitting polygons if necessary)



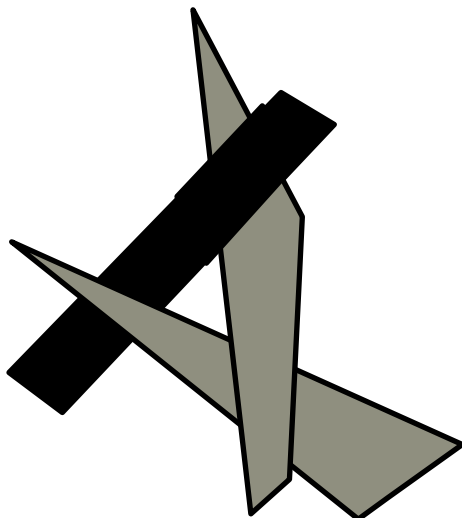


Five tests

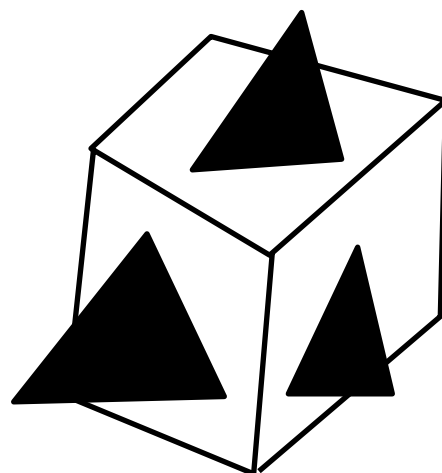
1. do the polygons' x extents not overlap?
2. do the polygons' y extents not overlap?
3. Is P1 entirely on the opposite side of P2 plane from the view point
4. Is p2 entirely on the same side of P1's plane as the viewpoint?
5. Do the projections of the polygons onto the xy plane not overlap?

Troublesome situations:

Cyclic overlap

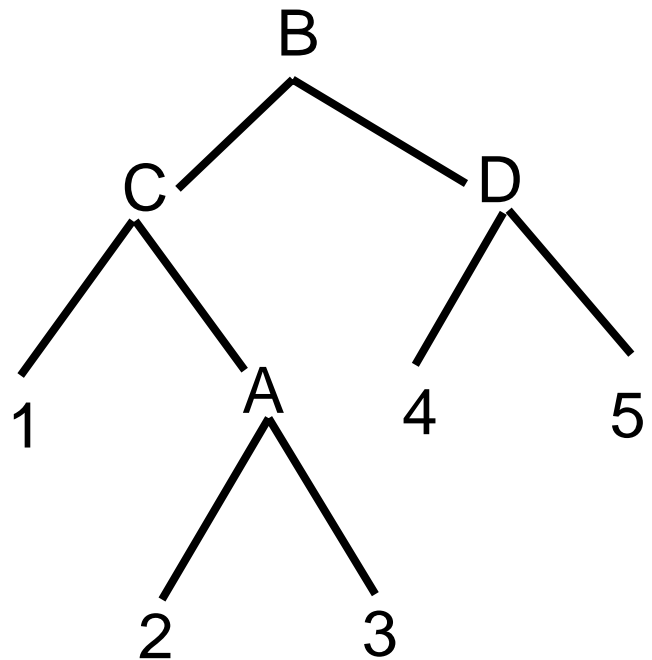
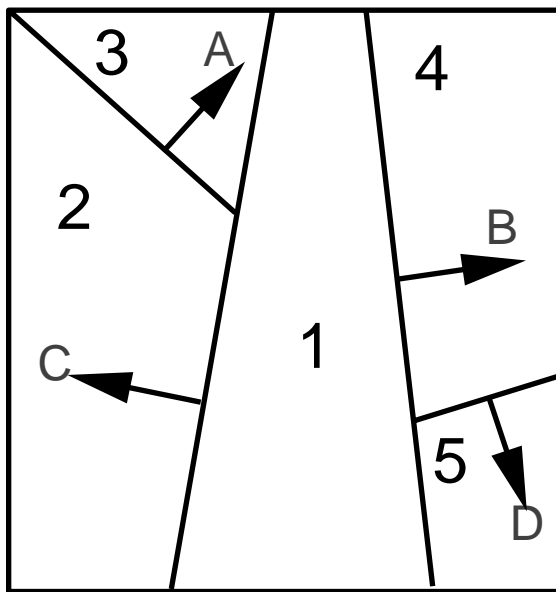


Piercing polygon



BSP-tree Method

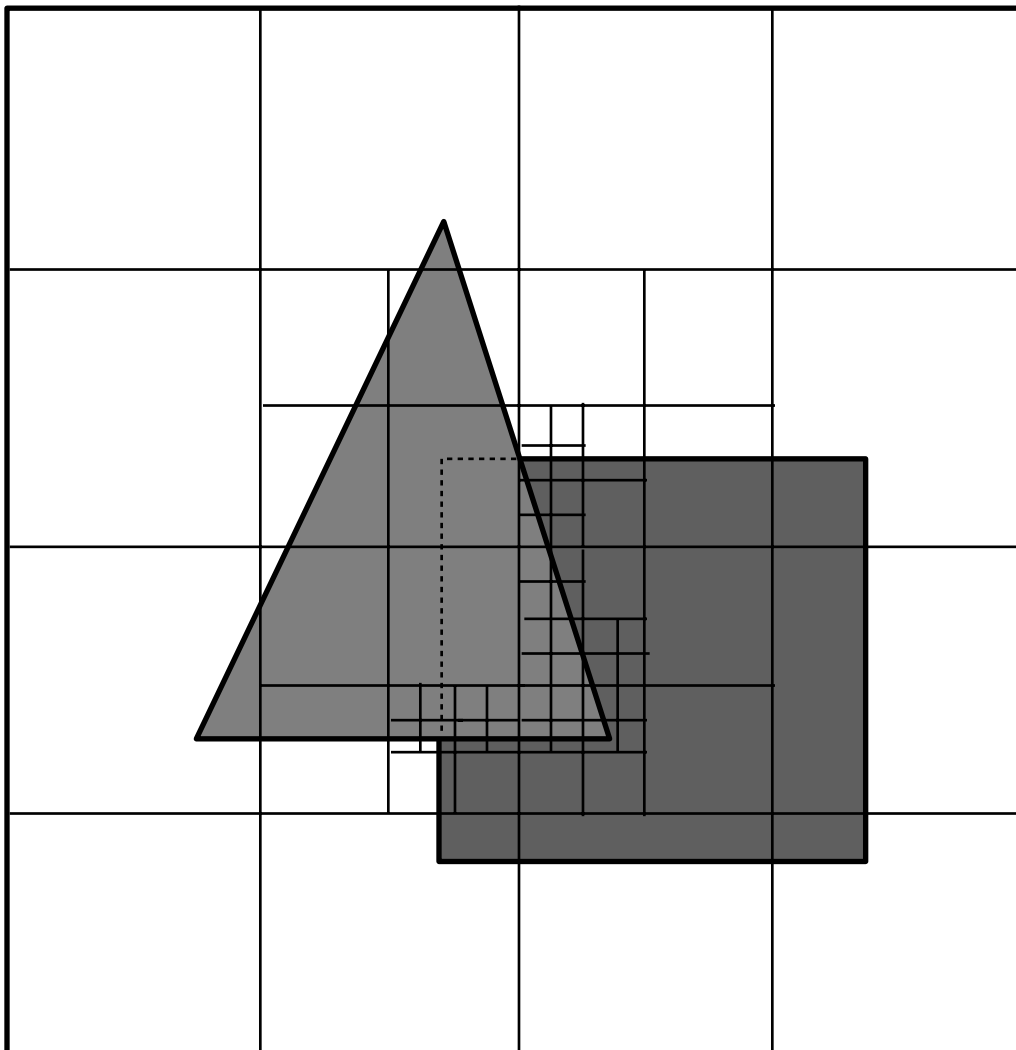
- binary space partitioning tree
- storing relative positioning information



- applicable to data of arbitrary dimensions
- at each stage, the space is partitioned into two parts of arbitrary size
- each subdivision plane (or line) can have an arbitrary orientation
- view-dependent operations can be applied appropriately to the resulting two half-spaces

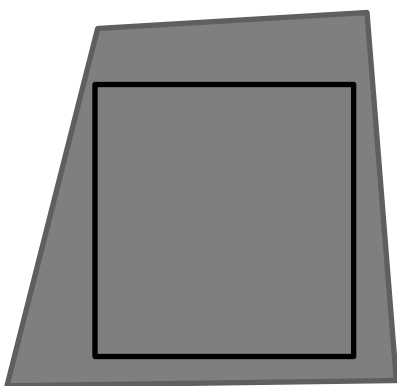
Area Subdivision (Warnock's algorithm)

- try to make an easy decision about which polygon is visible in a section of the image. If a decision cannot be made, subdivide the area recursively until one can be made.
- work at image precision for subdivision, and at object precision for depth comparison

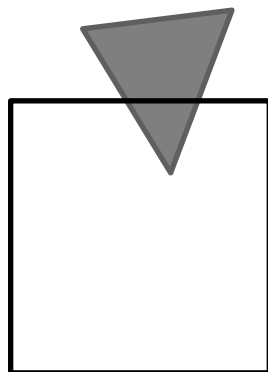


For each area do the following tests

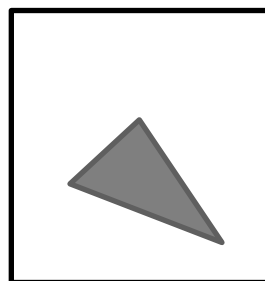
1. are all polygons disjoint from area?
if yes, display background color
2. only one intersecting or contained polygon.
if yes, fill with background color, and then
draw contained polygon or intersecting portion
3. one single surrounding polygon, no intersecting
or contained polygons.
if yes, draw area with that polygon's color
4. more than one polygon is intersecting,
contained in, or surrounding, but only one
polygon is surrounding the area and is in front
of others
if yes draw area with that front polygon's color
5. otherwise, subdivide the area into 4 equal areas
and recurse



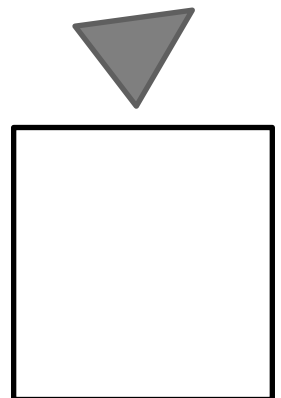
surrounding



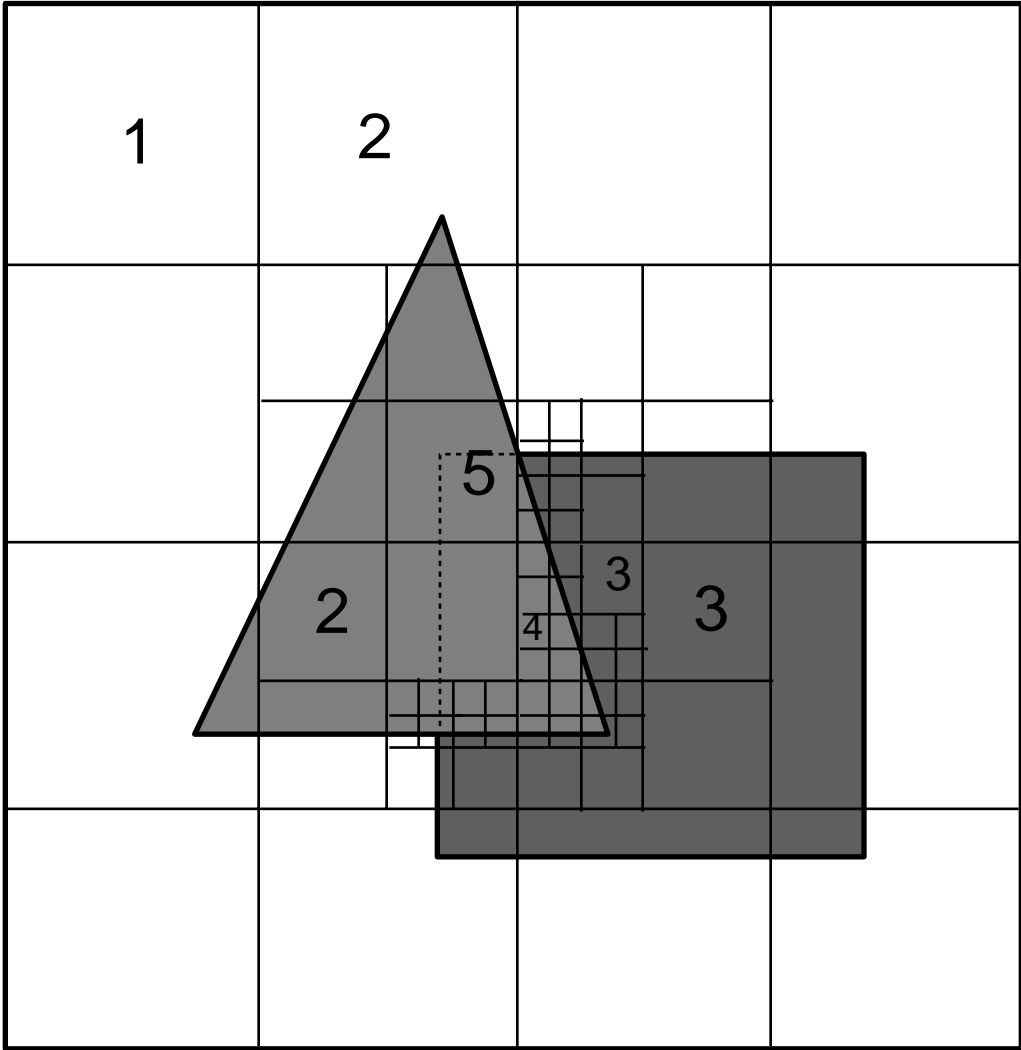
intersecting



contained

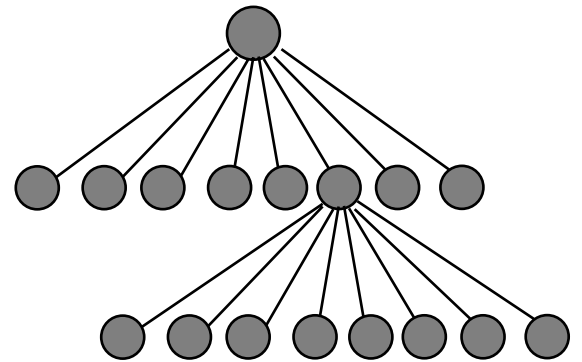
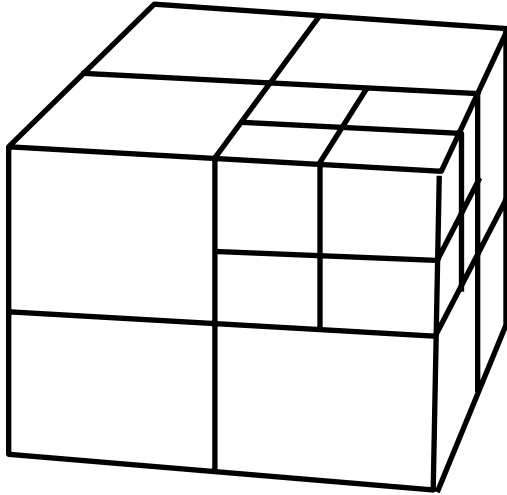


disjoint

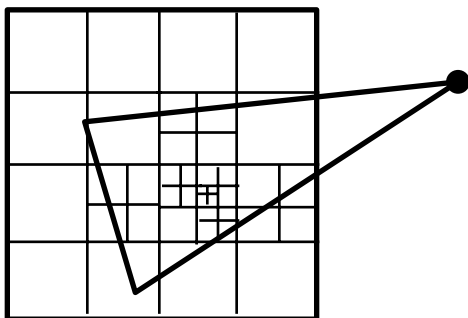


Octree Methods

- an octree is constructed by recursively partitioning the space into eight subspaces each of which is called an octant.

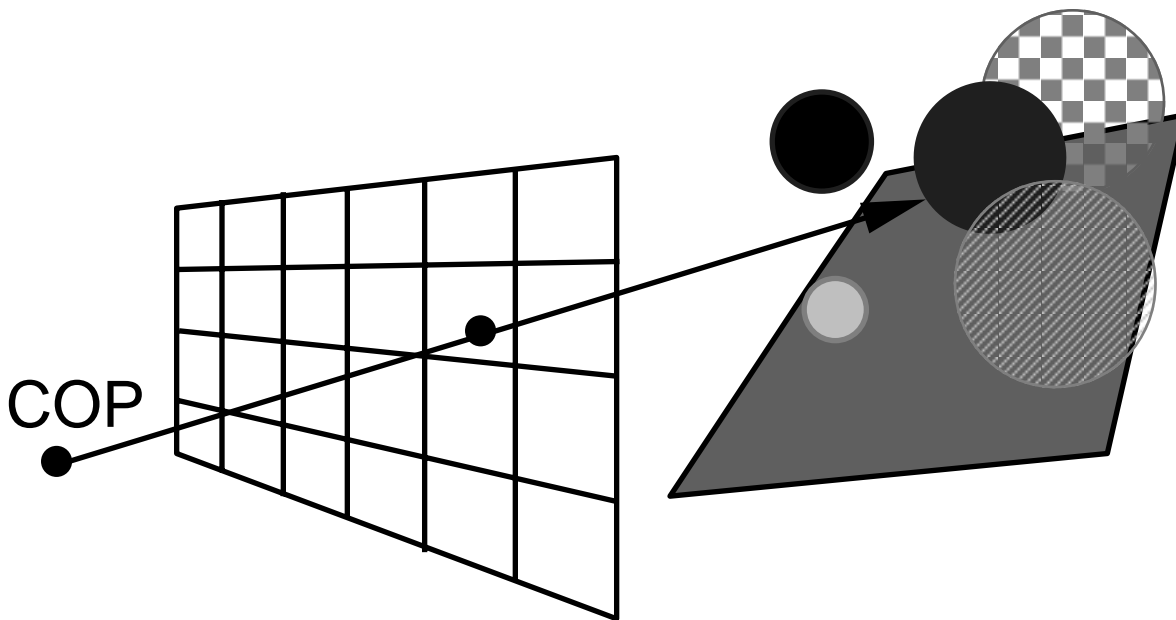


- regular partitioning
- spatial presorted and octants can be visited with a tree traversal
- back-to-front or front-to-back
- hierarchical Z-buffering for occlusion culling



Ray-casting Method

- as a visibility detection method
- only cast rays from pixel positions along the projection path
- effective for scene with curved surfaces

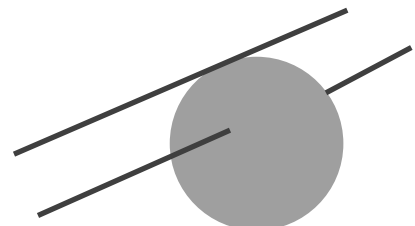


- computing intersections

e.g.:

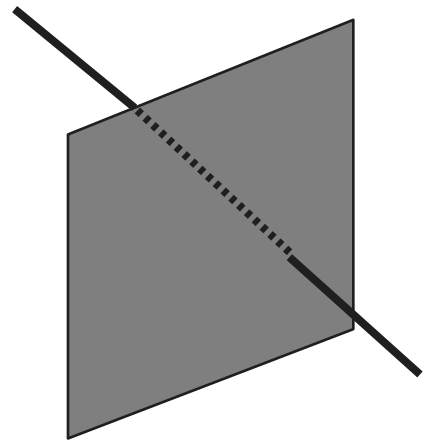
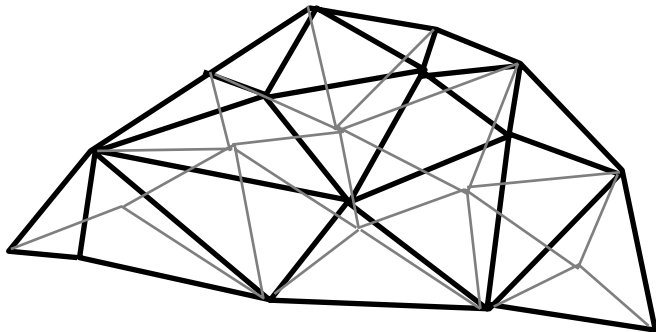
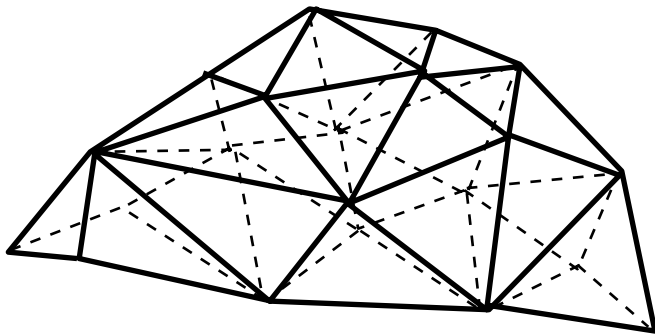
$$x = x_0 + t(x_1 - x_0) \quad y = y_0 + t(y_1 - y_0) \quad z = z_0 + t(z_1 - z_0)$$

$$(x-a)^2 + (y-b)^2 + (z-c)^2 = r^2$$

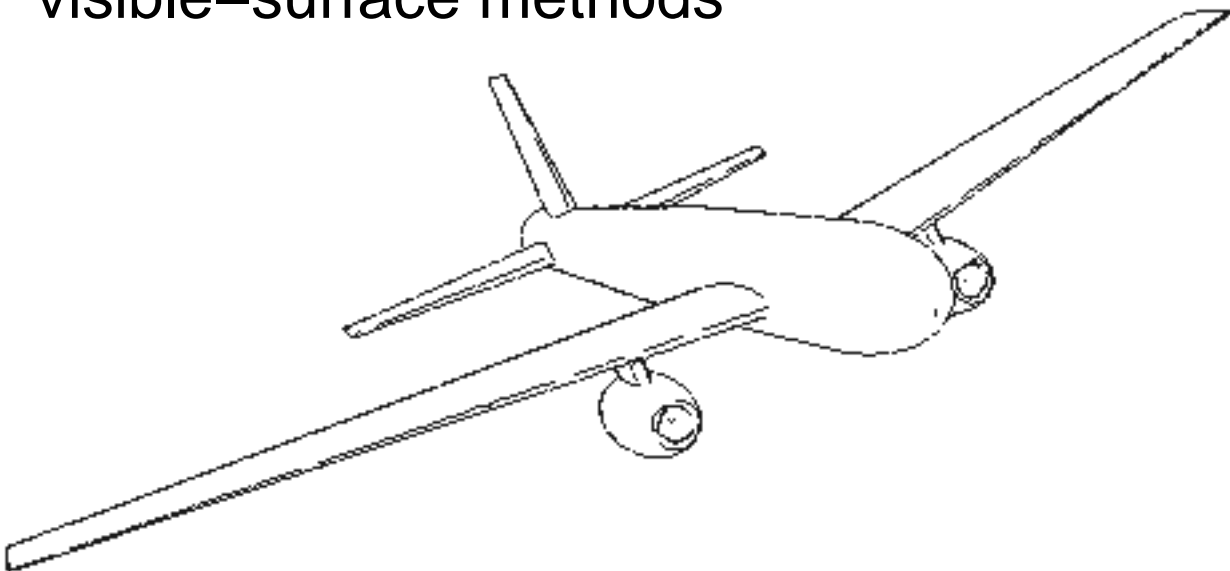


Wireframe Methods

- for drawing polygon mesh, computational mesh, outline of an object, illustration, ...
- may still want to draw the hidden lines!



- visibility testing can be achieved with variations of clipping methods and other visible–surface methods



Coherence

the degree to which parts of an environment or its projection exhibit local similarities

- Object coherence

2 entirely separated objects can be compared at the object level

- Face coherence

surface properties typically vary smoothly across a face

- Edge coherence

an edge may change visibility only where it crosses behind a visible edge or penetrates a face

- Scan–line coherence

the set of visible object spans determined for consecutive scanlines are often the same

- Area coherence

a group of adjacent pixels is often covered by the same visible face

- Depth coherence

adjacent parts of the same surface are typically close in depth

- Frame coherence

two consecutive frames are likely to be similar