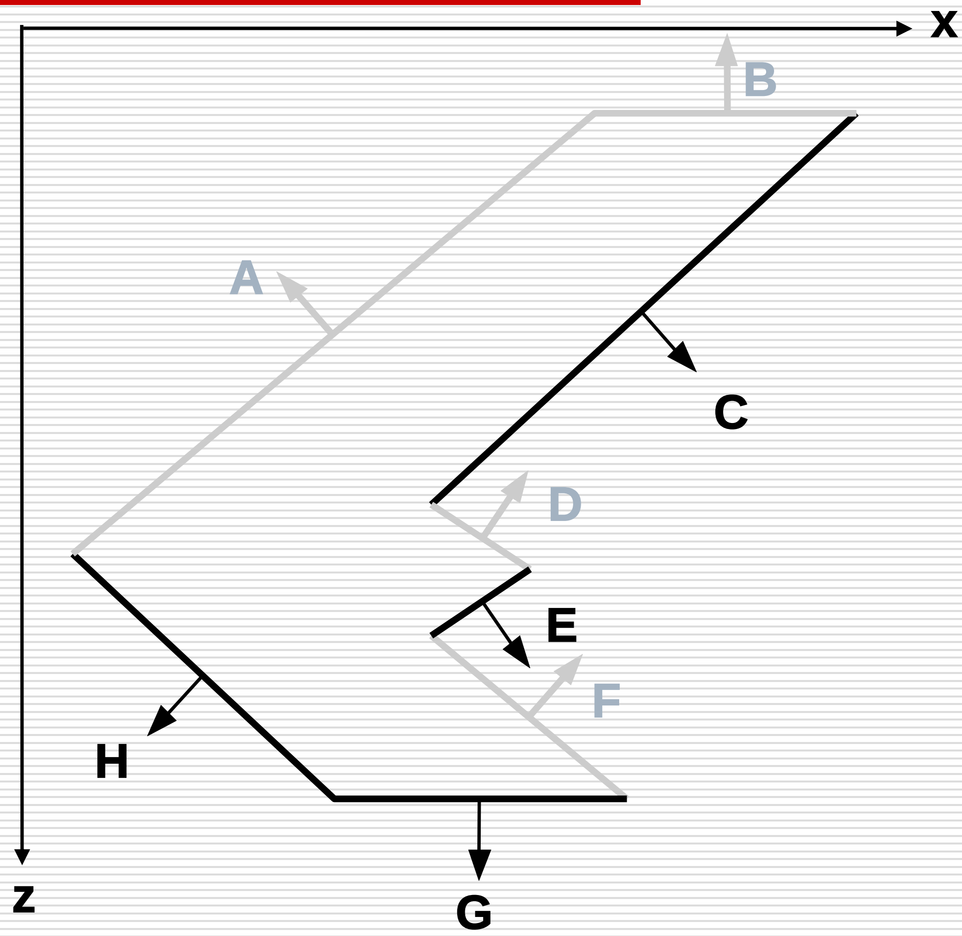
Computer Graphics

Jeng-Sheng Yeh 葉正聖
Ming Chuan University
(modified from Bing-Yu Chen's slides)

Visible-Surface Determination

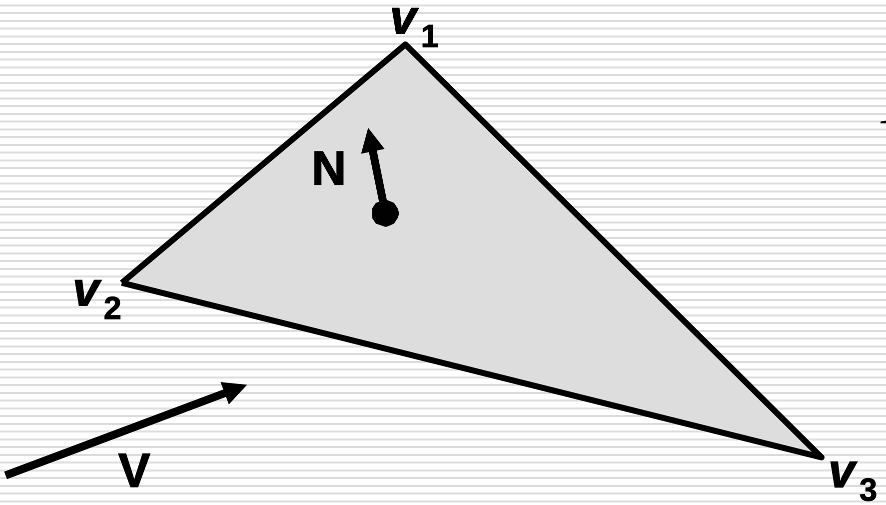
- Back-Face Culling
- The Depth-Sort Algorithm
- Binary Space-Partitioning Trees
- The z-Buffer Algorithm
- Scan-Line Algorithm
- Visible-Surface Ray Tracing
- Warnock's Algorithm

Back-Face Culling = Front Facing



Back-Face Culling = Front Facing

- use cross-product to get the normal of the face (not the actual normal)
- use inner-product to check the facing



$$N = (v_2 - v_1) \times (v_3 - v_1)$$

List-Priority Algorithms

- The Painter’s Algorithm
- The Depth-Sort Algorithm
- Binary Space-Partitioning Trees

The Painter's Algorithm

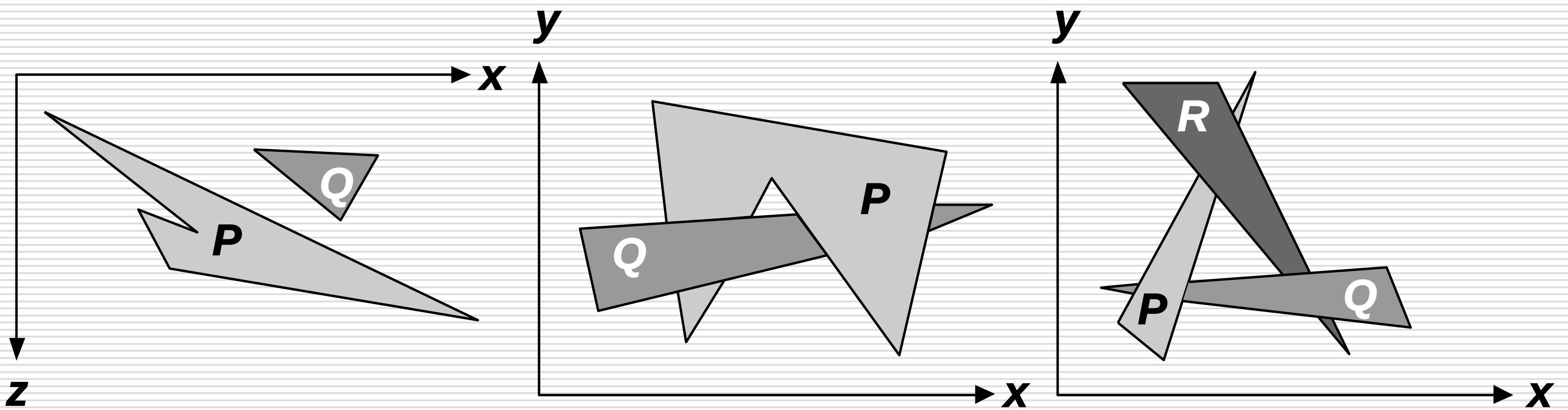
- for the planes with constant *z*
- not for real 3D, just for 2½D

- **sort** all polygons according to the smallest (farthest) *z* coordinate of each
- **scan convert** each polygon in ascending order of smallest *z* coordinate (i.e., back to front)

The Depth-Sort Algorithm

- **sort** all polygons according to the smallest (farthest) *z* coordinate of each
- resolve any ambiguities that sorting may cause when the polygons' *z* extents **overlap**, **splitting** polygons if necessary
- **scan convert** each polygon in ascending order of smallest *z* coordinate (i.e., back to front)

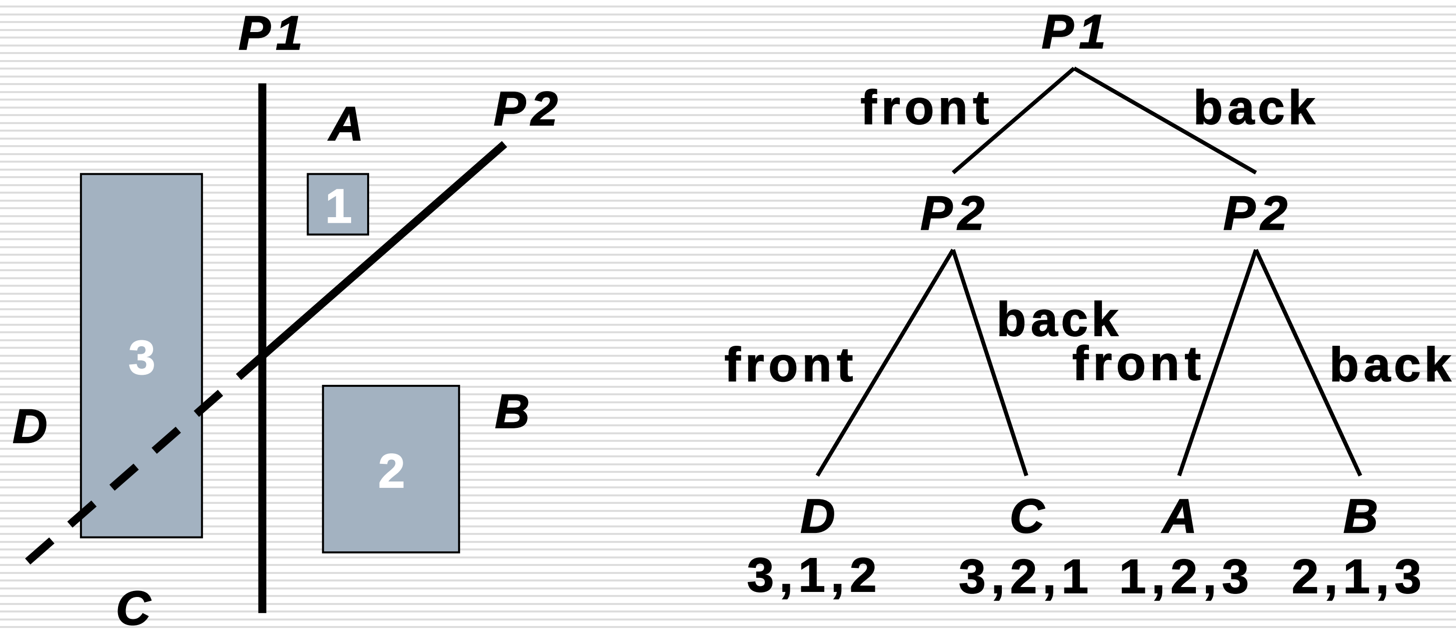
Overlap Cases



Overlap Detection

- Do the polygons'x not overlap?
- Do the polygons'y not overlap?
- Is P entirely on the opposite side of Q's plane from the viewpoint?
- Is Q entirely on the same side of P's plane as the viewpoint?
- Do the projections of the polygons onto the (x,y) plane not overlap?

Binary **S**pace-**P**artitioning Trees

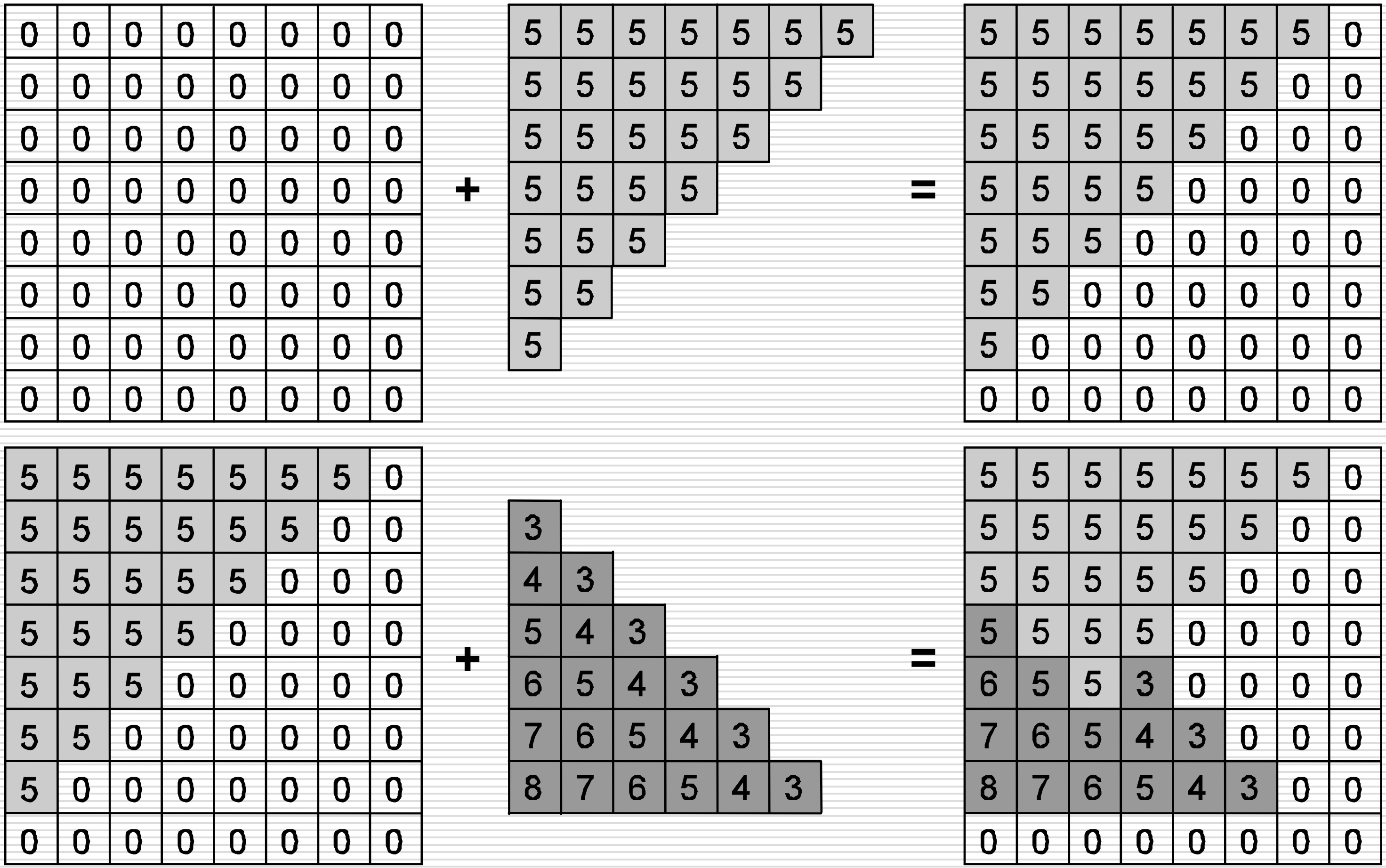


- extremely efficient for static objects

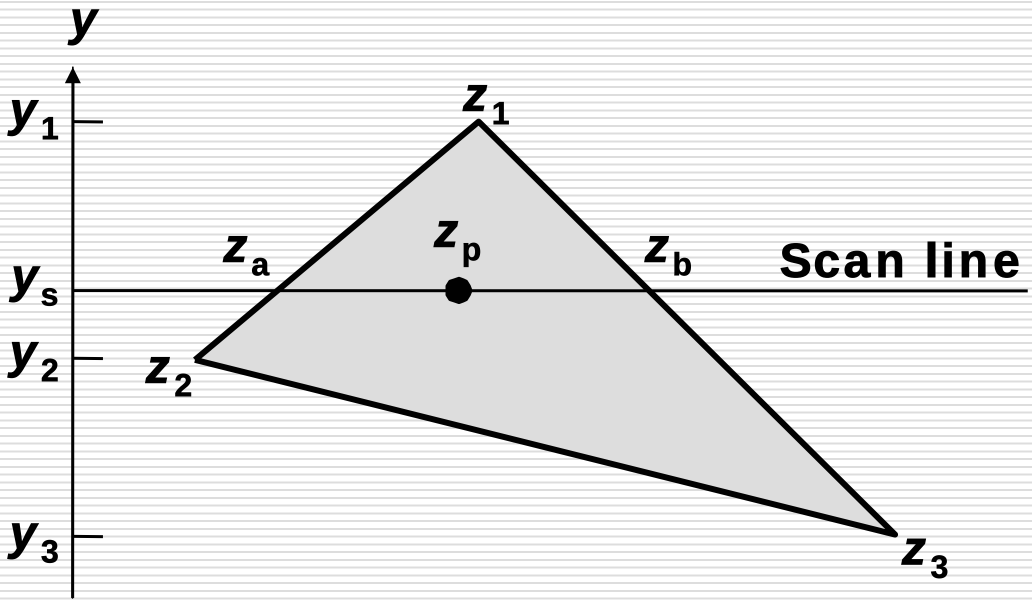
The z-Buffer Algorithm

```
void zBuffer() {
    int pz;
    for (each polygon) {
        for (each pixel in polygon's projection) {
            pz=polygon's z-value at (x,y);
            if (pz>=ReadZ(x,y)) {
                WriteZ(x,y,pz);
                WritePixel(x,y,color);
            }
        }
    }
}
```

The z-Buffer Algorithm



The z-Buffer Algorithm

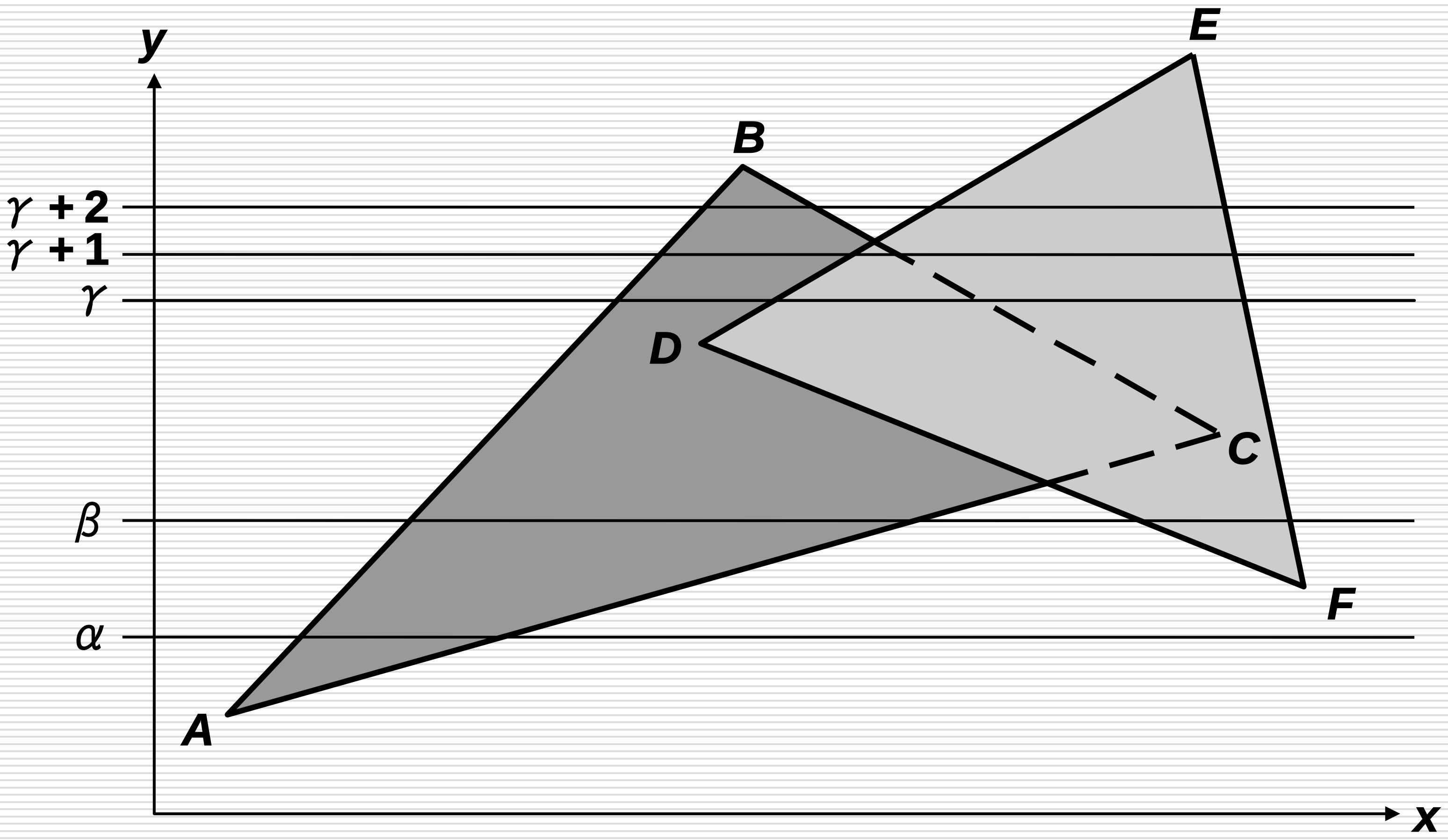


$$z_a = z_1 - (z_1 - z_2)\frac{y_1 - y_s}{y_1 - y_2}$$

$$z_b = z_1 - (z_1 - z_3)\frac{y_1 - y_s}{y_1 - y_3}$$

$$z_p = z_b - (z_b - z_a)\frac{x_b - x_p}{x_b - x_a}$$

Scan-Line Algorithm



Scan-Line Algorithm

ET entry

x	y_{max}	Δx	ID	● →

PT entry

ID	Plane eq.	Shading info	In-out

AET contents

Scan line	Entries			
α	*AB*	*AC*		
β	*AB*	*AC*	*FD*	*FE*
$\gamma, \gamma+1$	*AB*	*DE*	*CB*	*FE*
$\gamma+2$	*AB*	*CB*	*DE*	*FE*

- ET = edge table
- PT = polygon table
- AET = active-edge table

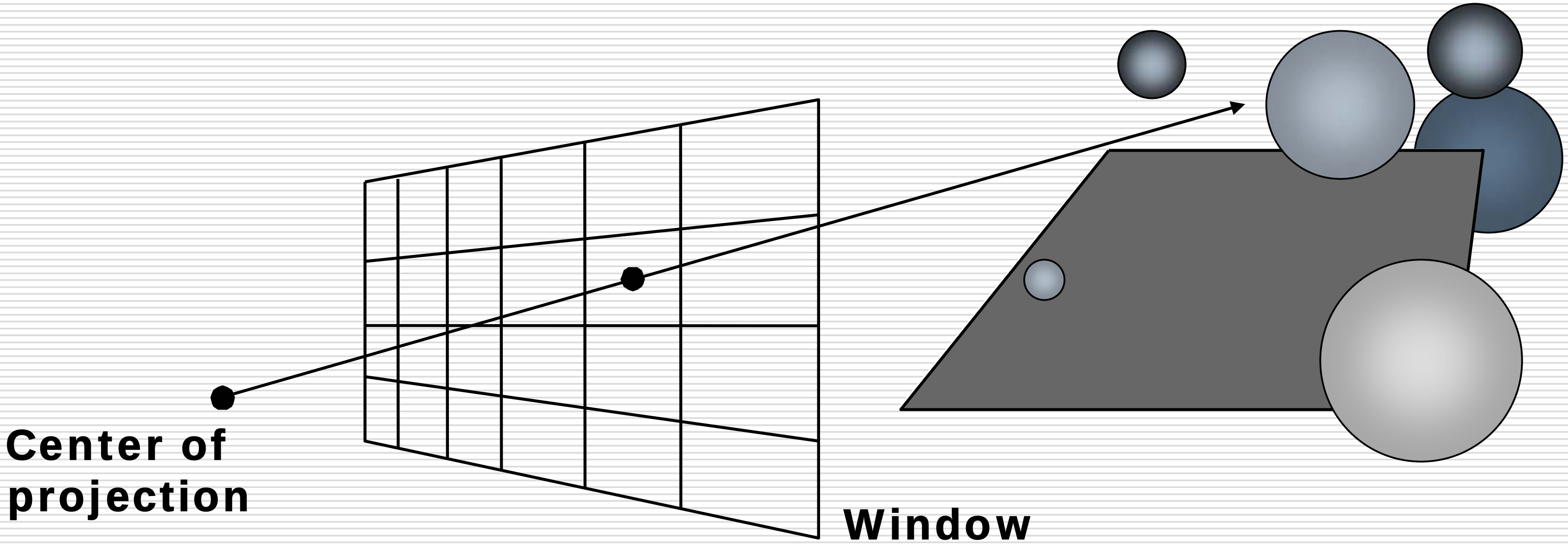
General Scan-Line Algorithm

```
add surfaces to surface table (ST);
initialize active-surface table (AST);

for (each scan line) {
    update AST;

    for (each pixel on scan line) {
        determine surfaces in AST that project to pixel;
        find closest such surface;
        determine closest surface's shade at pixel;
    }
}
```

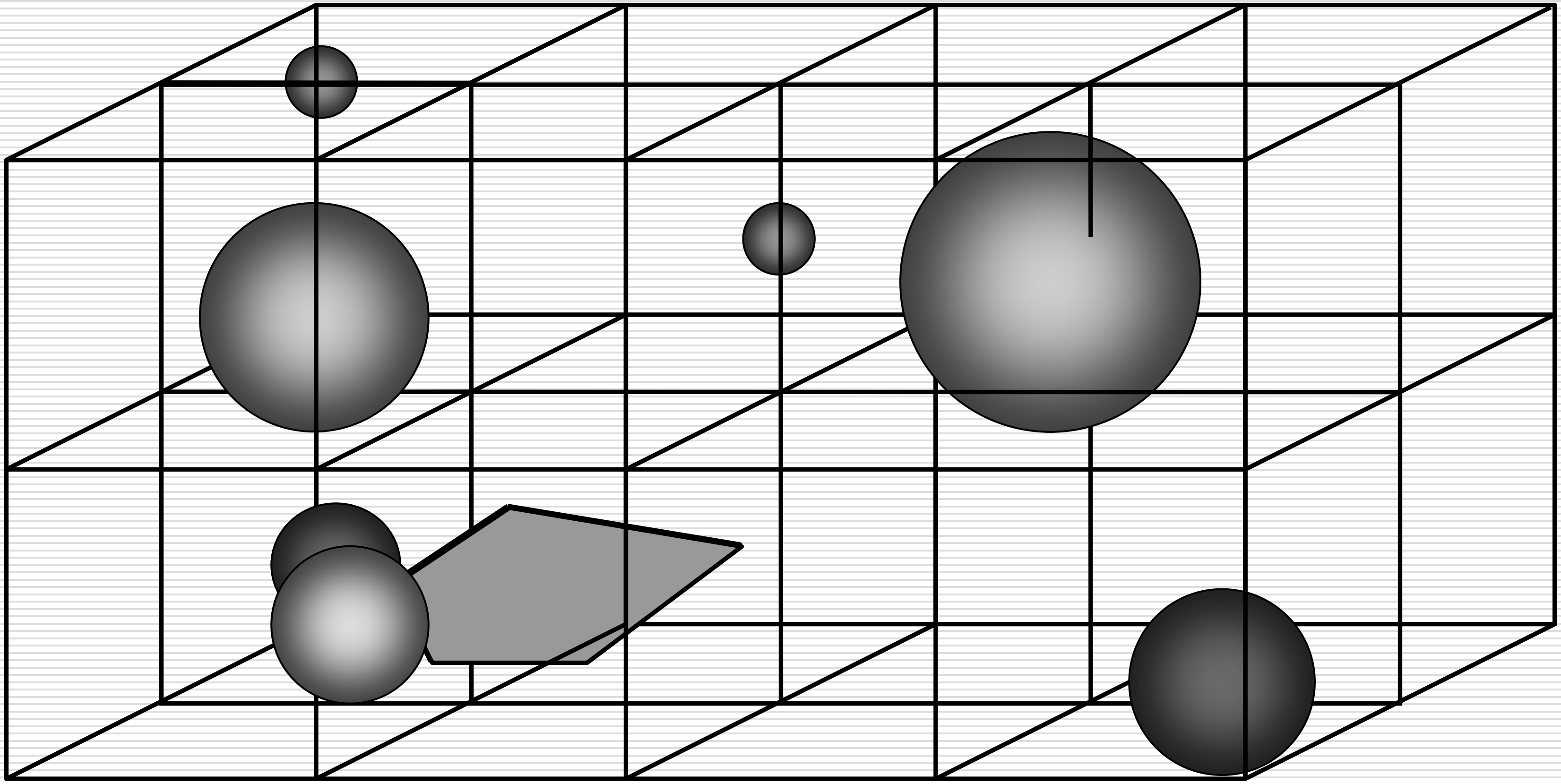
Ray Tracing = Ray Casting

```
select center of projection and window on viewplane;
for (each scan line in image) {
    for (each pixel in scan line) {
        determine ray from center of projection through pixel;
        for (each object in scene) {
                if (object is intersected and is closest considered thus far)
                        record intersection and object name;
        }
        set pixel's color to that at closest object intersection;
    }
}
```

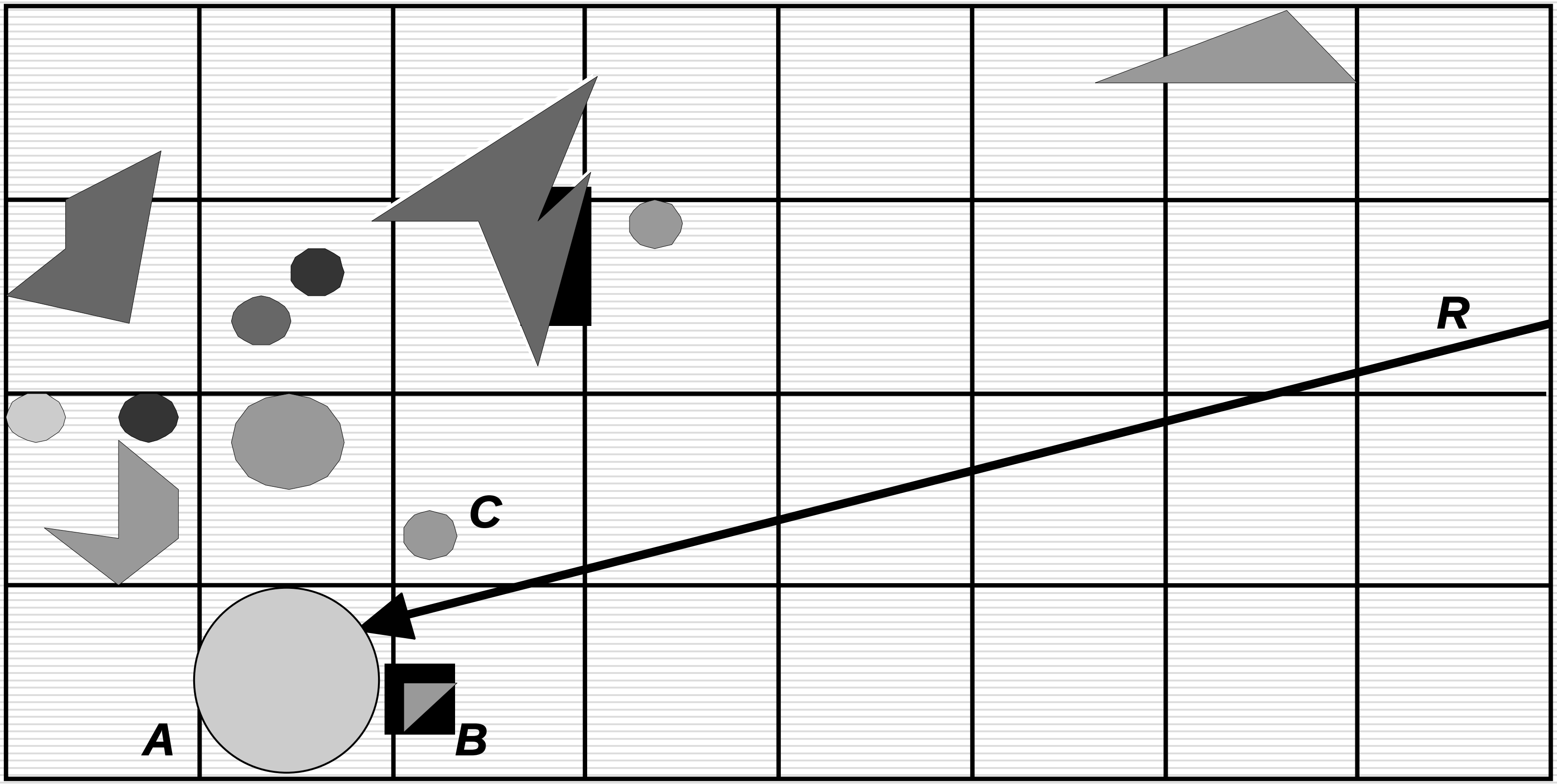
Ray Casting



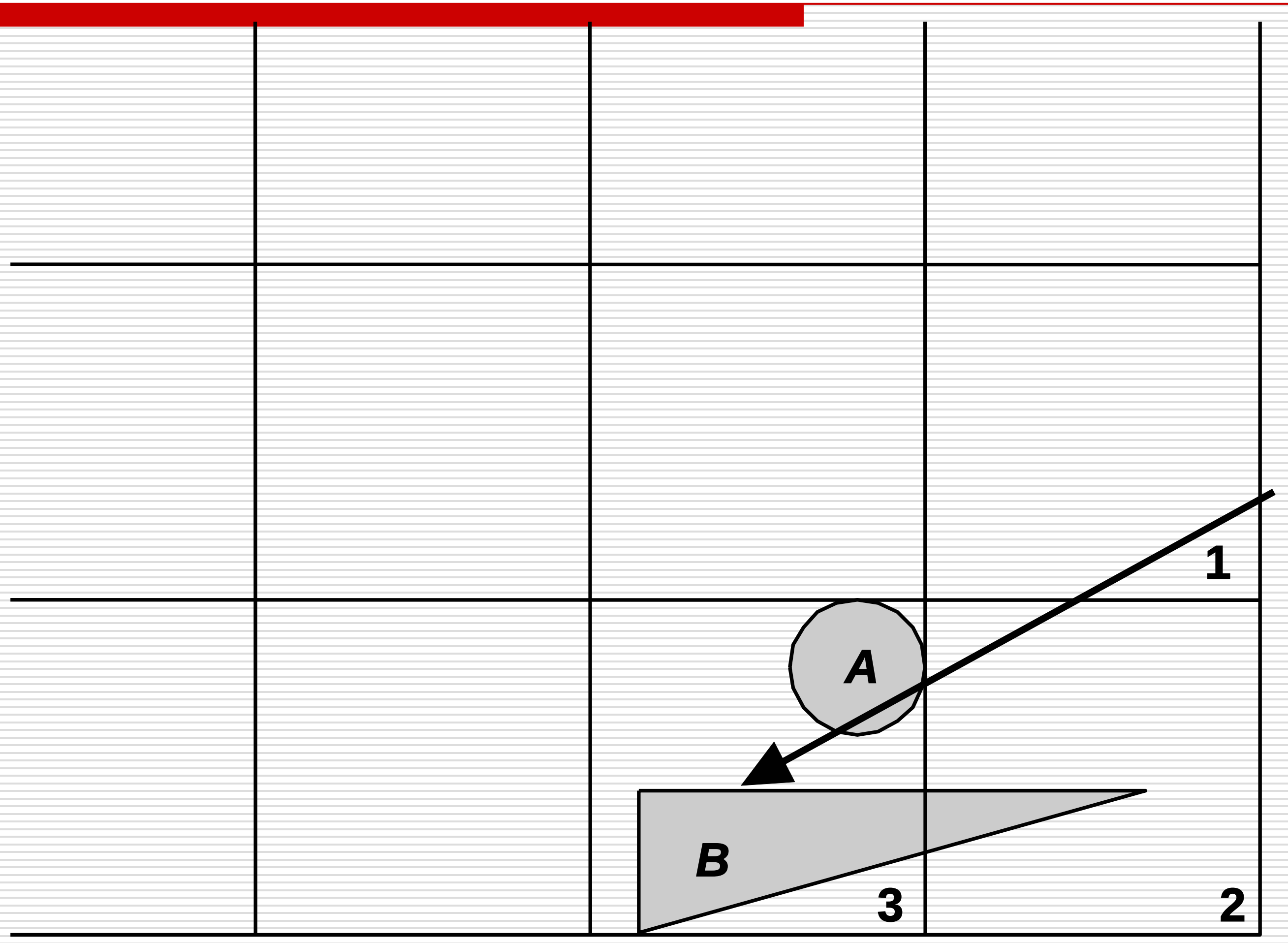
Spatial Partitioning



Spatial Partitioning

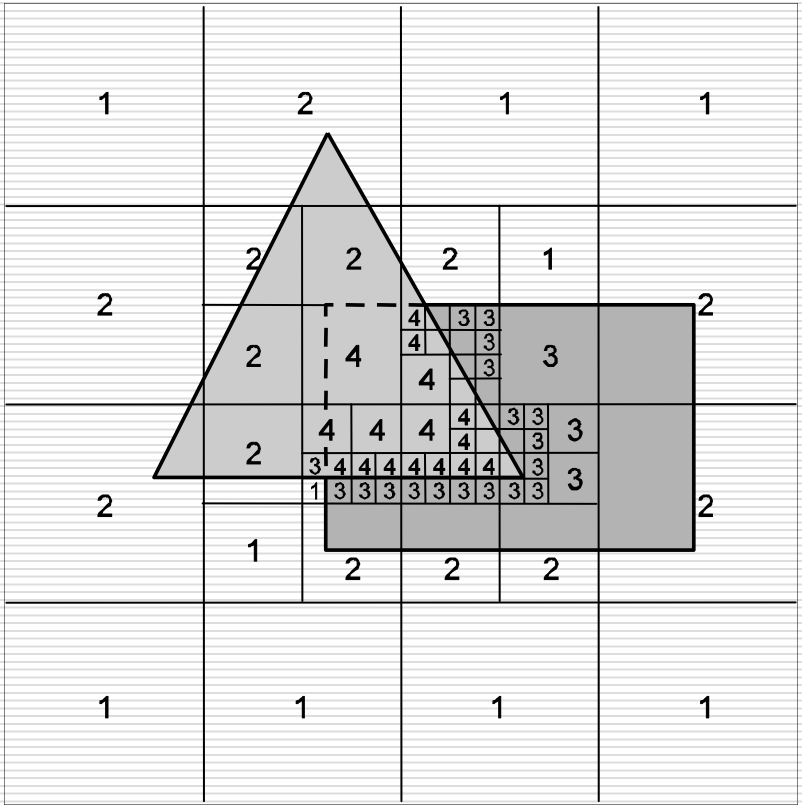


Spatial Partitioning



Warnock's Algorithm

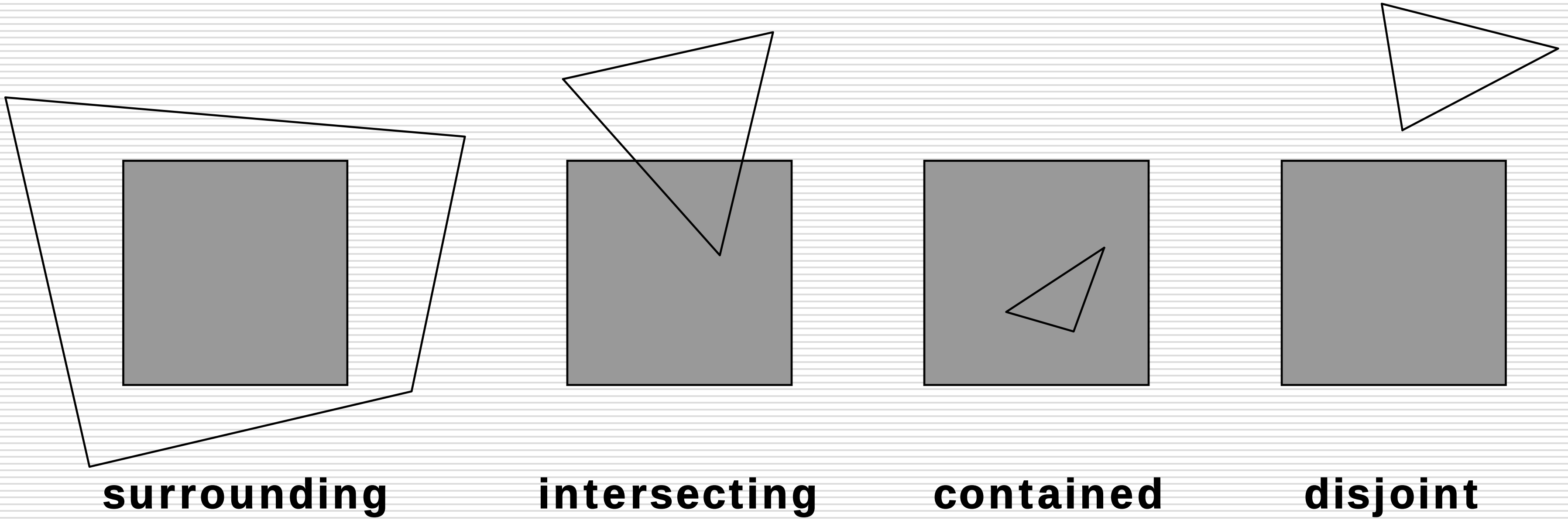
- a area-subdivision algorithm



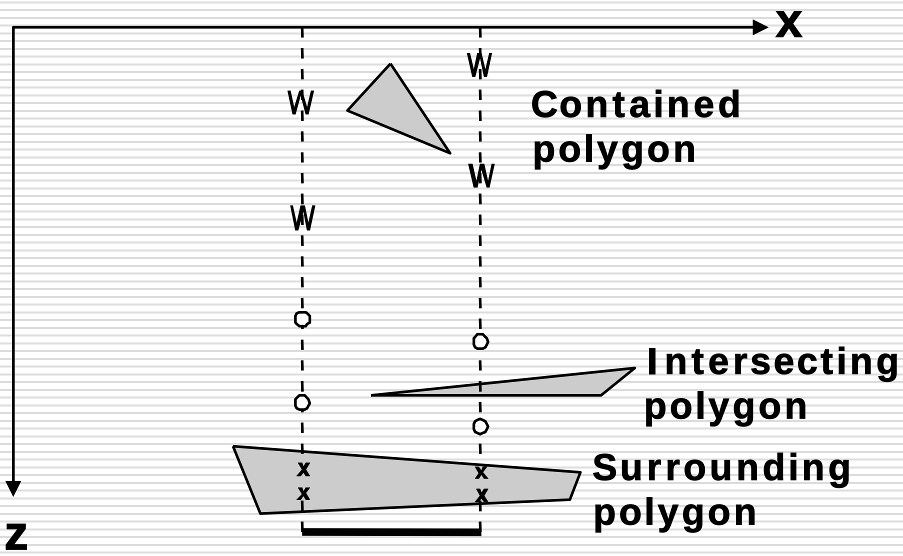
Warnock’s Algorithm

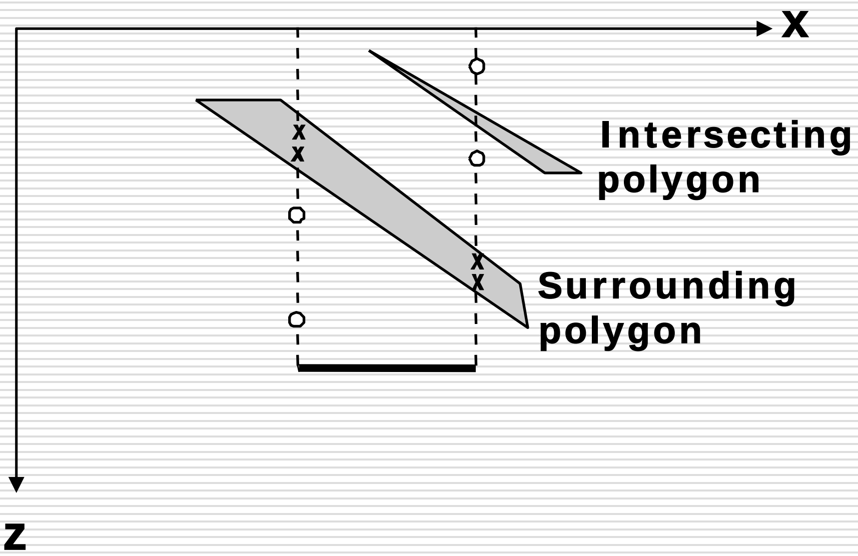
1. all the polygons are disjoint from the area
2. there is only one intersecting or only one contained polygon
3. there is a single surrounding polygon, but no intersecting or contained polygons
4. more than one polygon is intersecting, contained in, or surrounding the area, but one is a surrounding polygon that is in front of all the other polygons

Warnock's Algorithm



Warnock's Algorithm





Performance of Four Algorithms for Visible-Surface Determination

Algorithm	Number of Polygons		
	100	2,500	60,000
Depth sort	1	10	507
z-buffer	54	54	54
Scan line	5	21	100
Warnock area subdivision	11	64	307