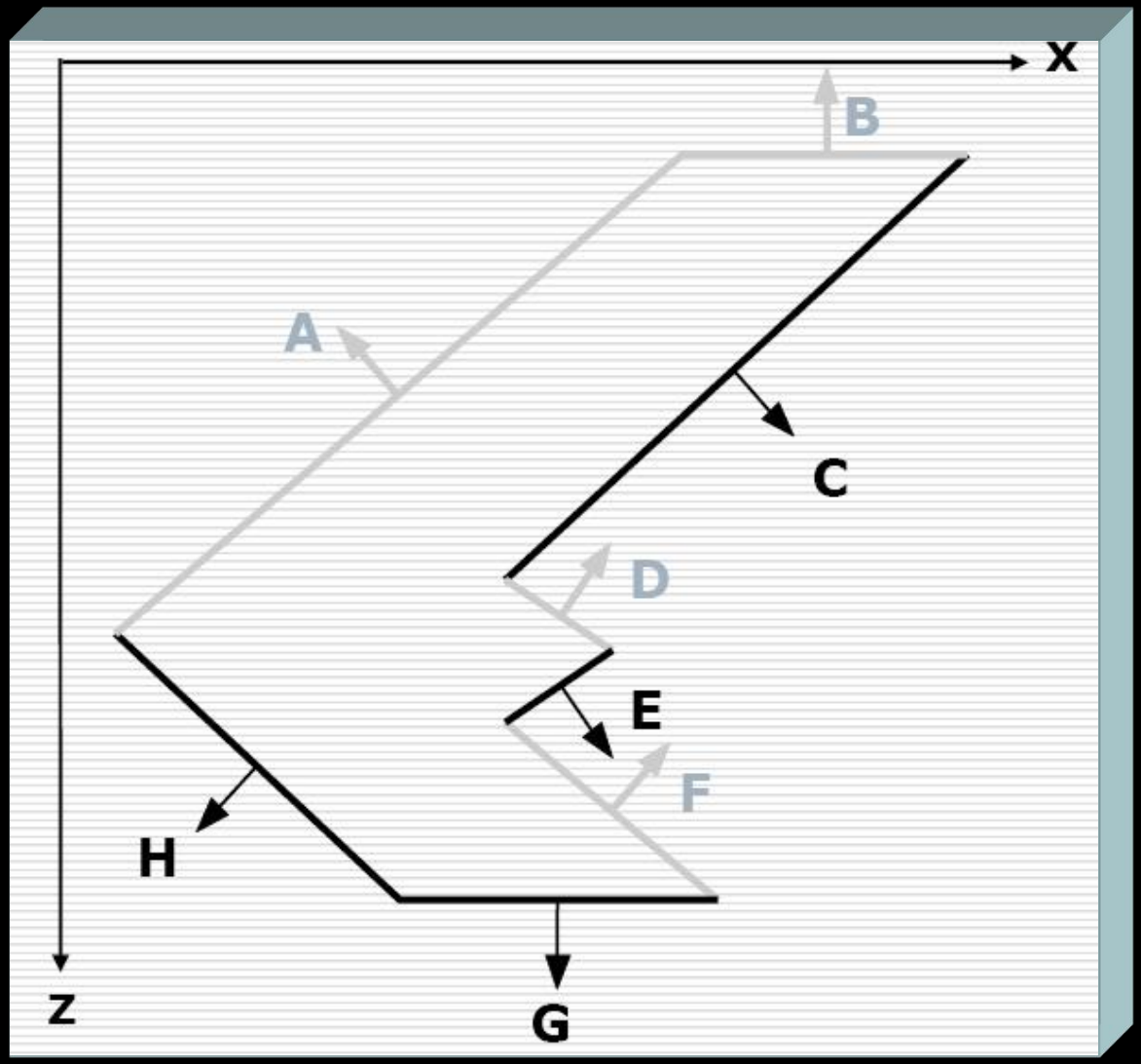
Visible-Surface Determination

Prof. Lizhuang Ma
Shanghai Jiao Tong University

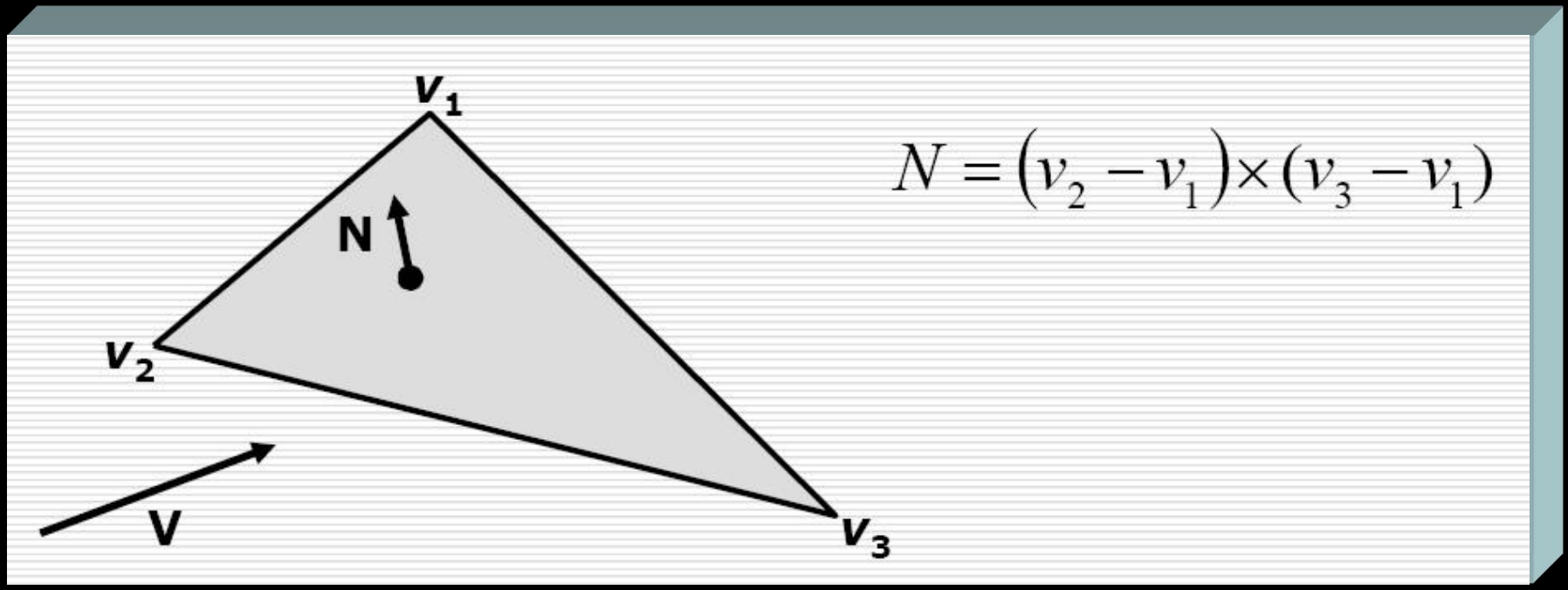
Contents

- 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



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



Contents

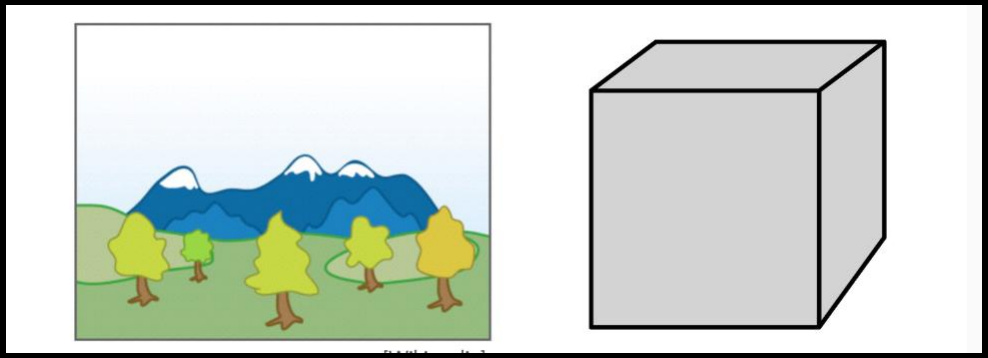
- Back-Face Culling
- The Depth-Sort Algorithm
- Binary Space-Partitioning Trees
- The z-Buffer Algorithm
- Scan-Line Algorithm
- Visible-Surface Ray Tracing
- Warnock’s Algorithm

List-Priority Algorithms

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

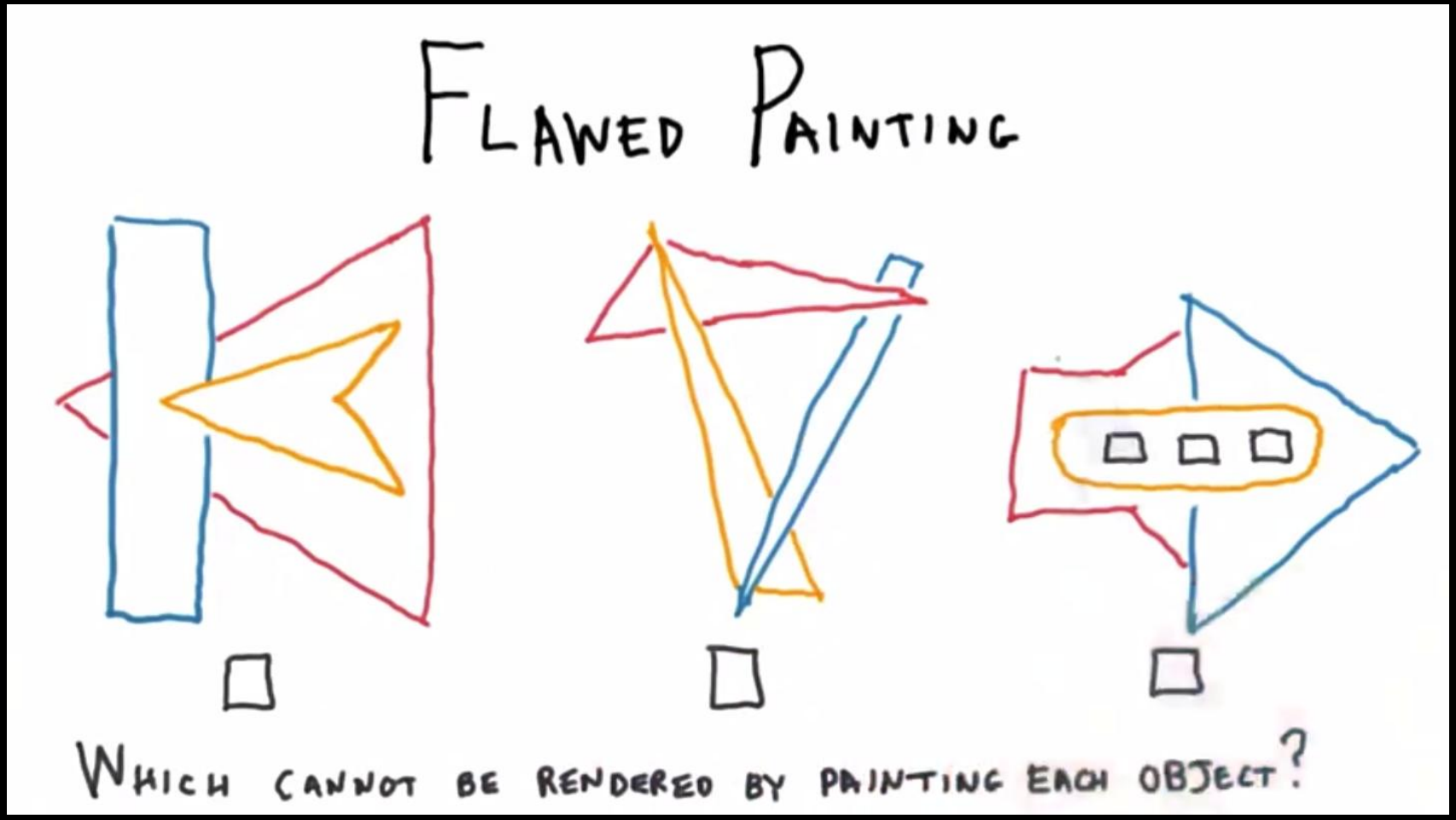
The Painter's Algorithm

- 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 Painter's Algorithm

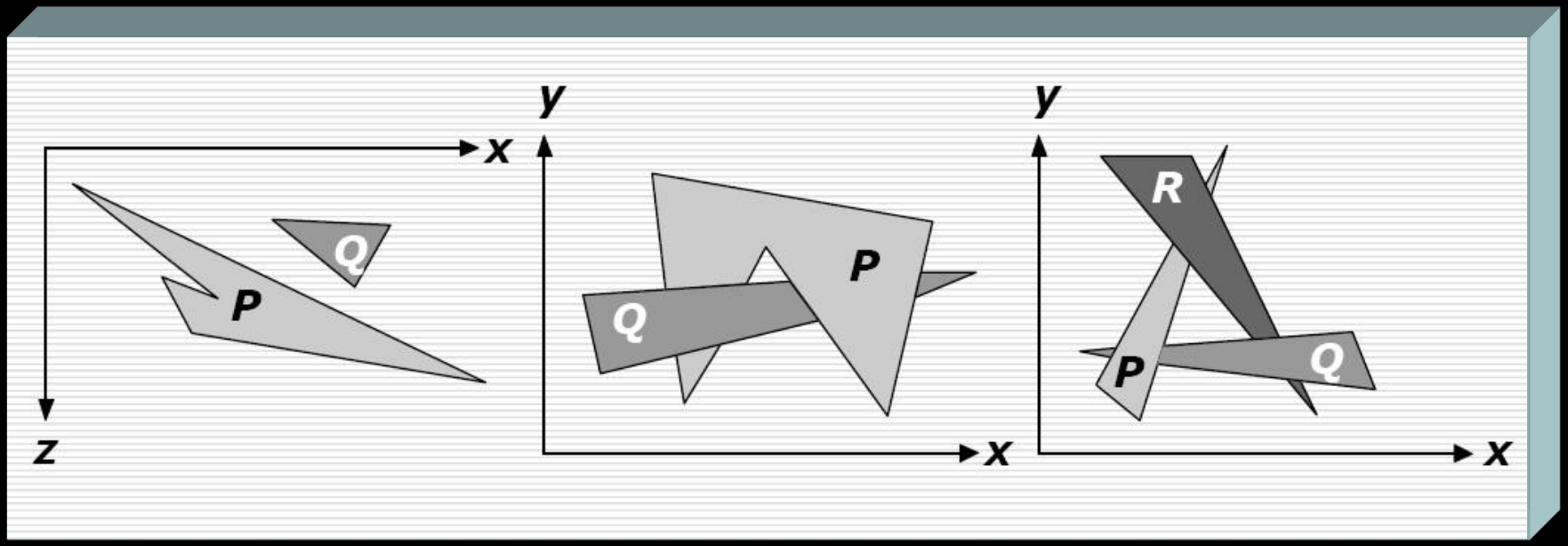
The Painter's Algorithm



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)

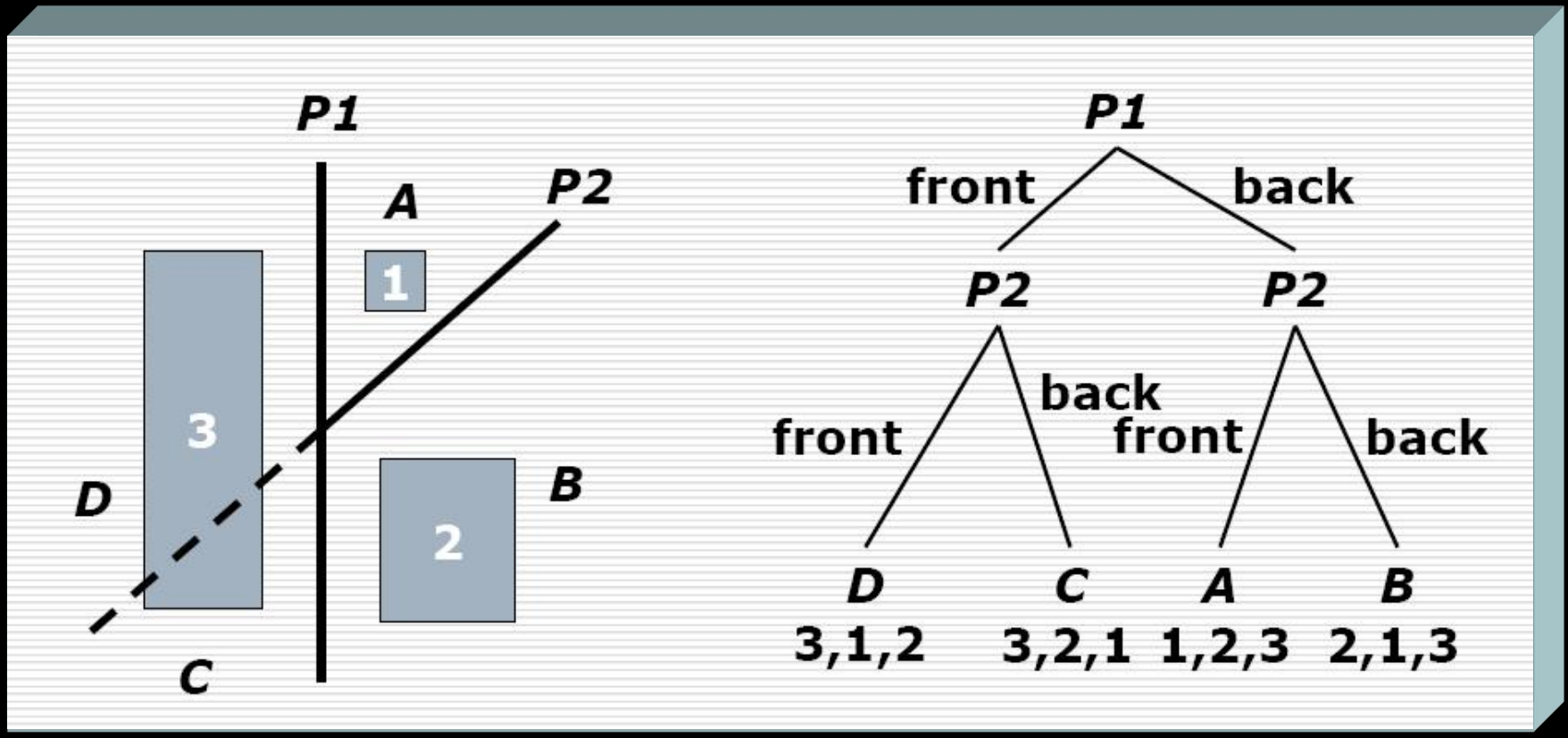
Overlap Cases



Overlap Detection

- If $z_{min}(P) < z_{min}(Q) < z_{max}(P)$
- 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

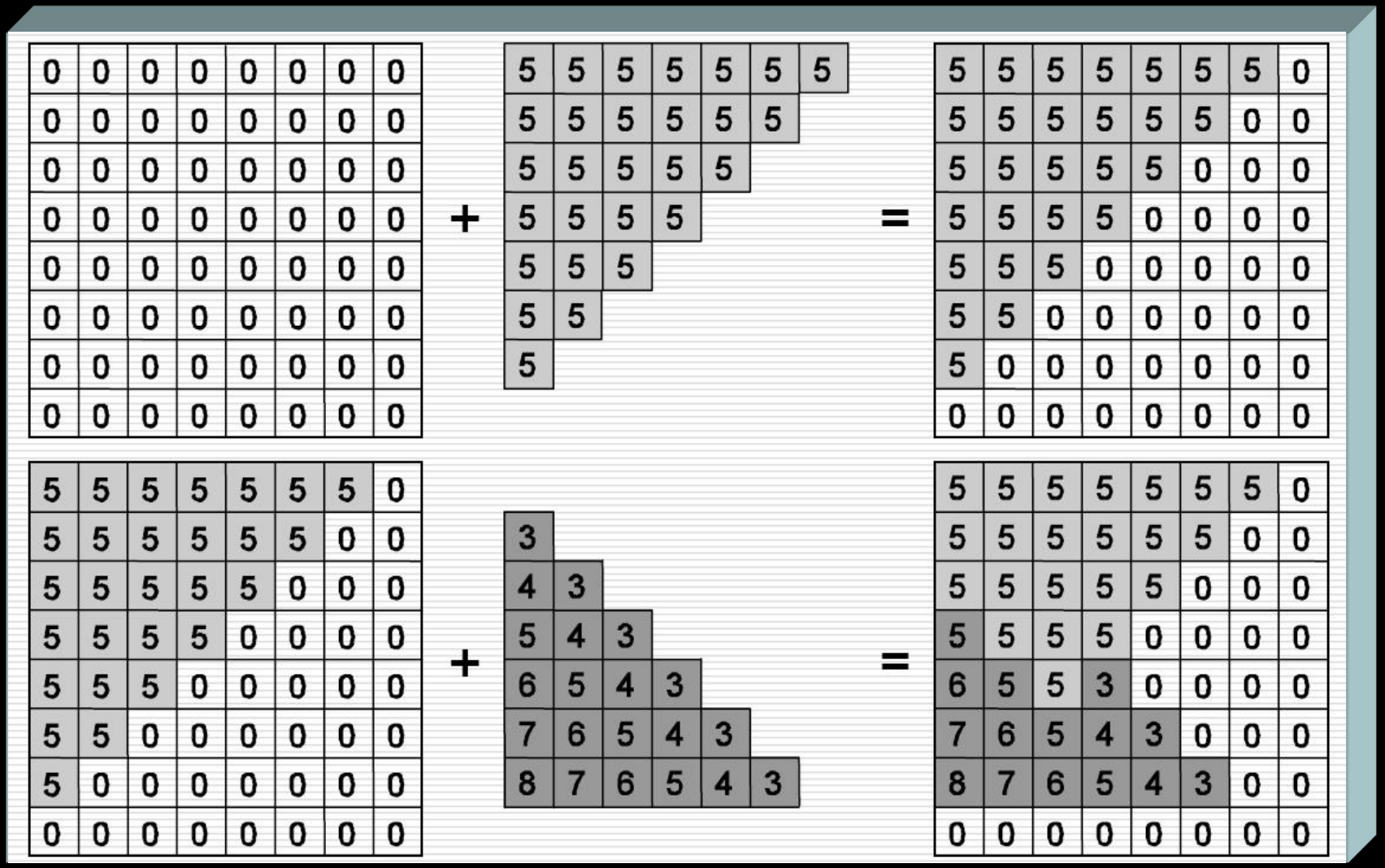
Contents

- 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

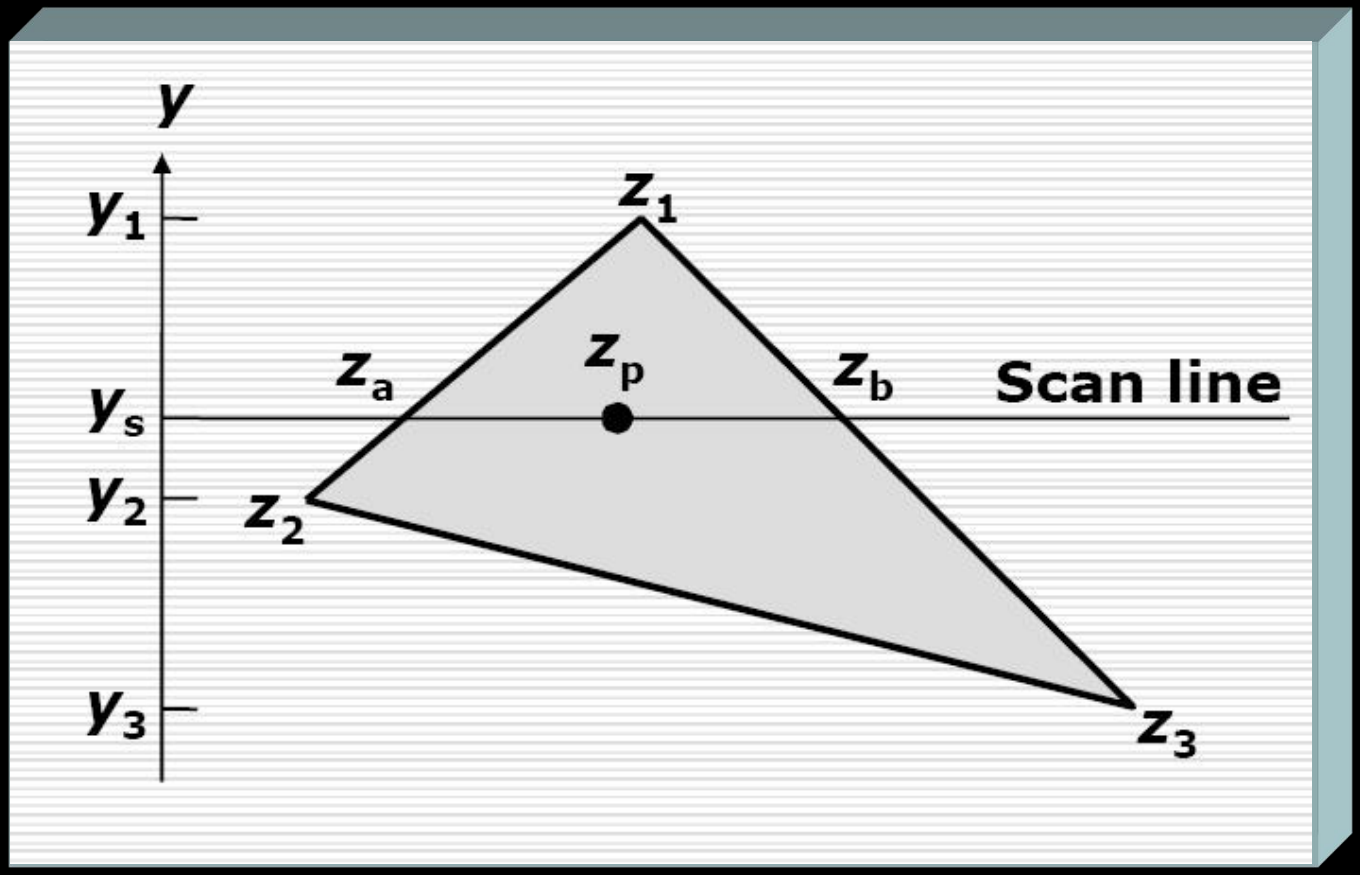
The z-Buffer Algorithm

```
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



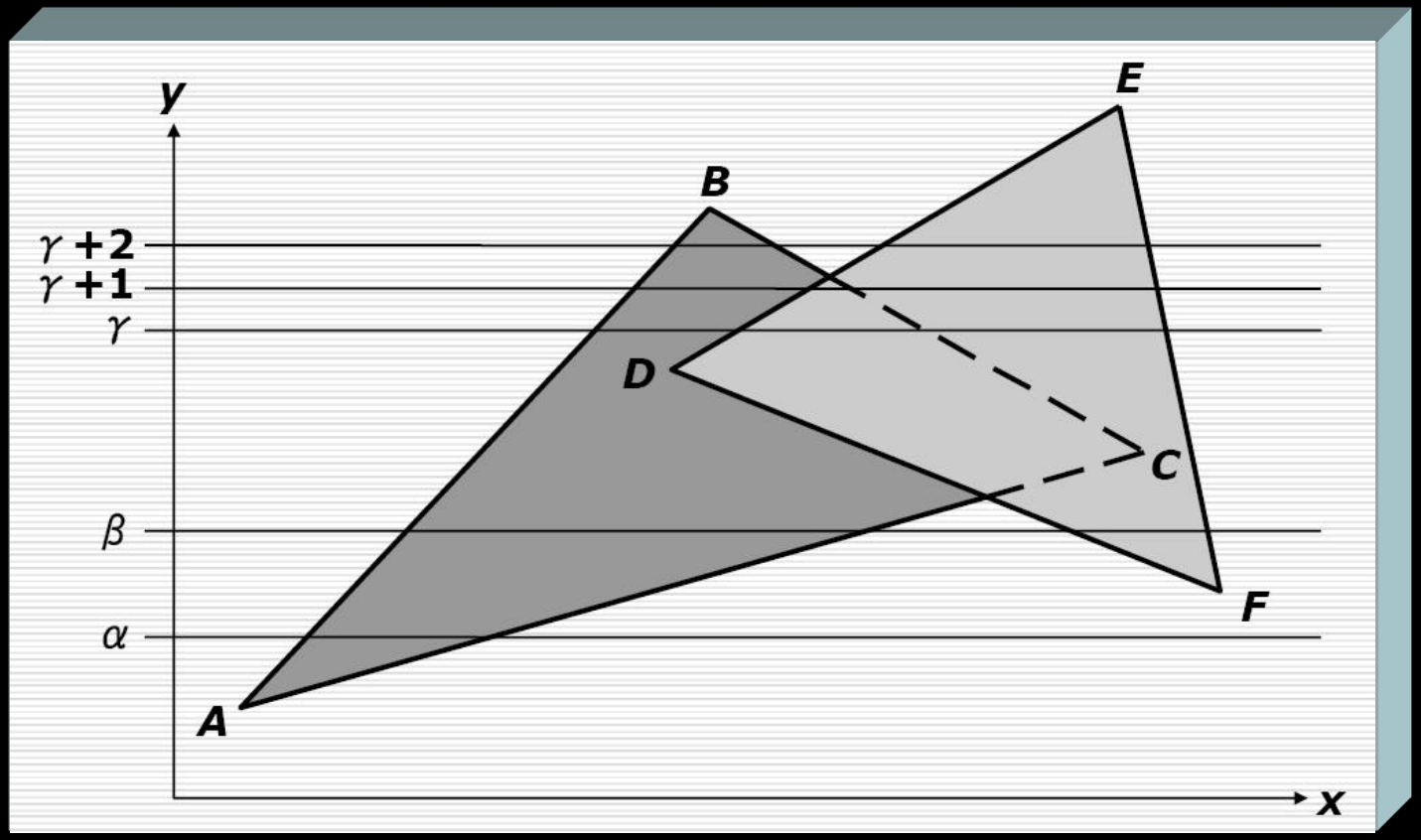
The z-Buffer Algorithm



Contents

- 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

Scan-Line Algorithm



Scan-Line Algorithm

ET entry:

x	y_{max}	dx	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;
   }
}
```

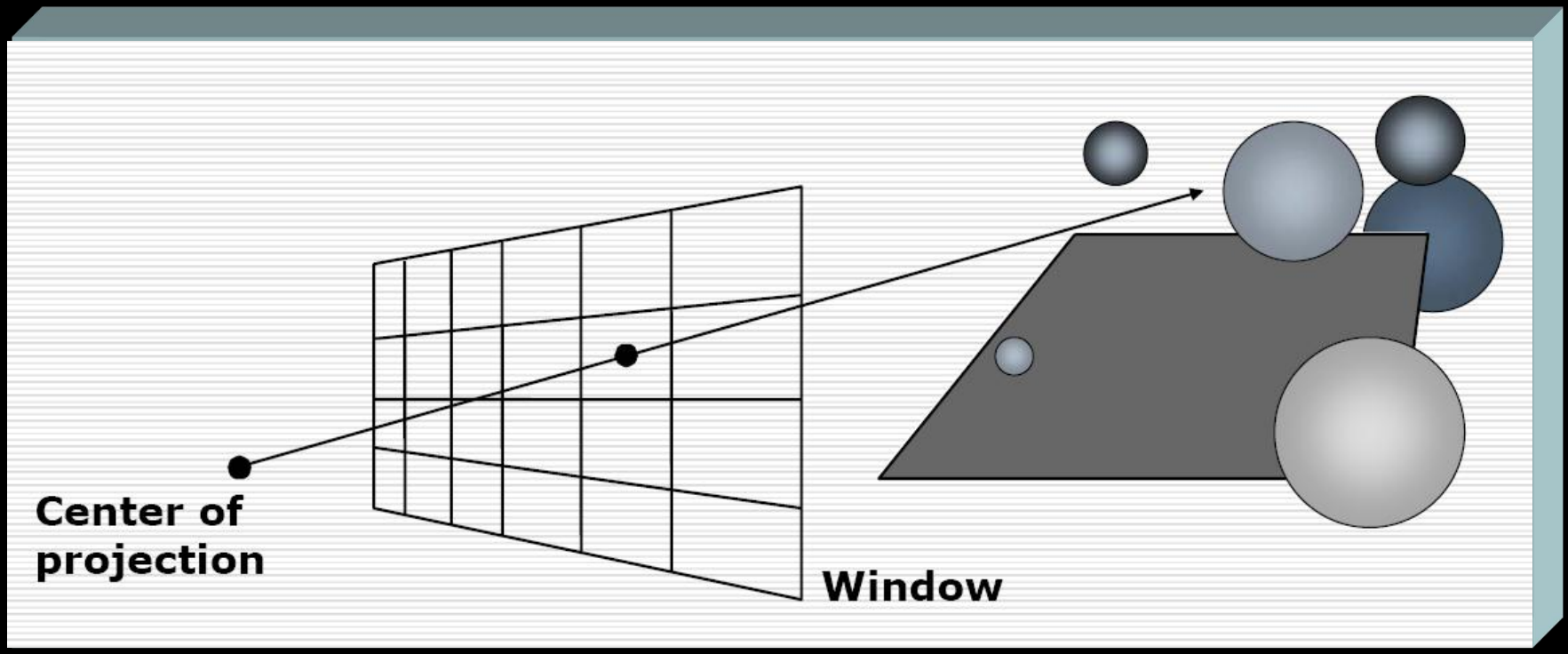
Contents

- 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

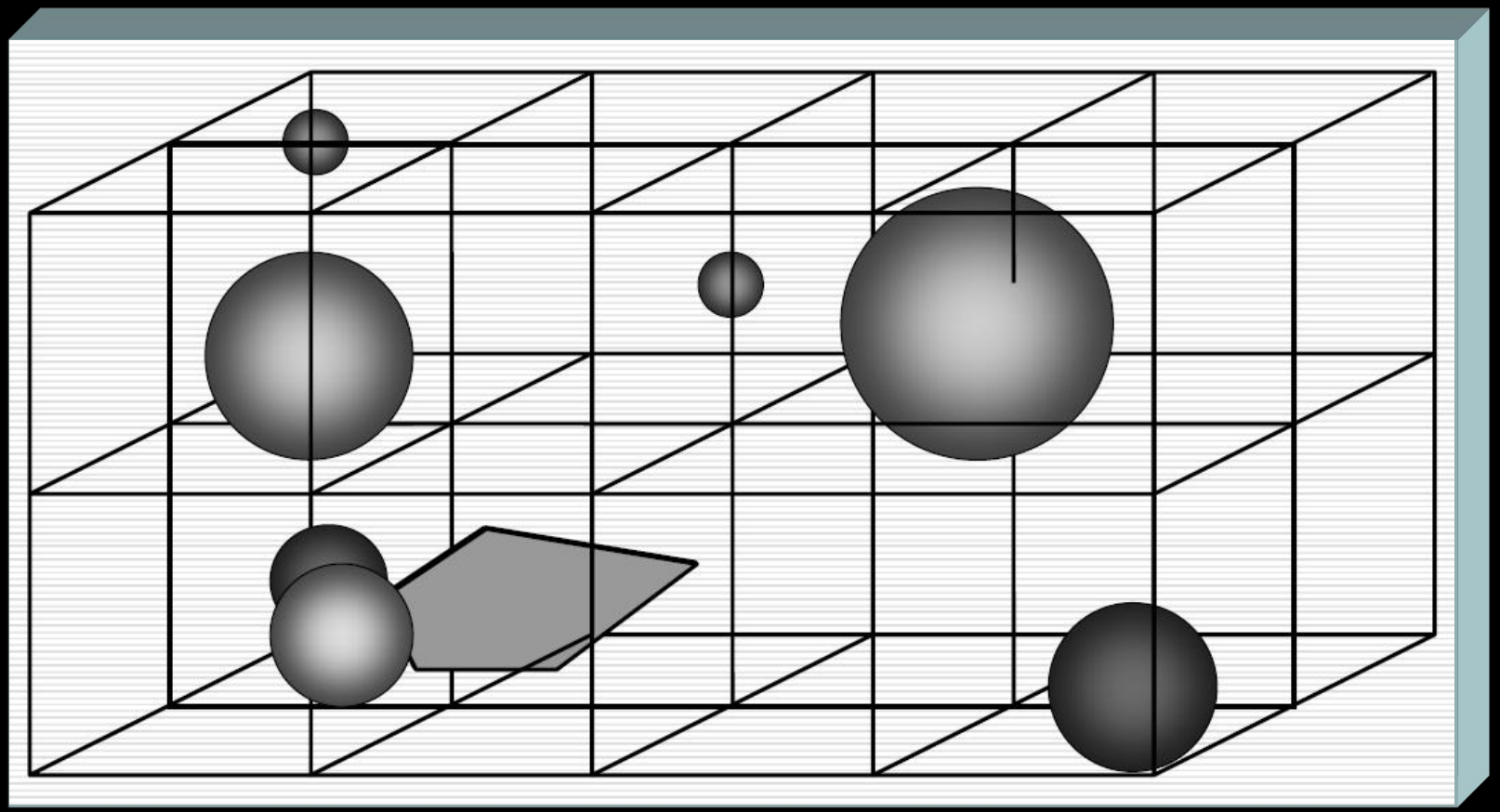
Ray Tracing = Ray Casting

```
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;
}
```

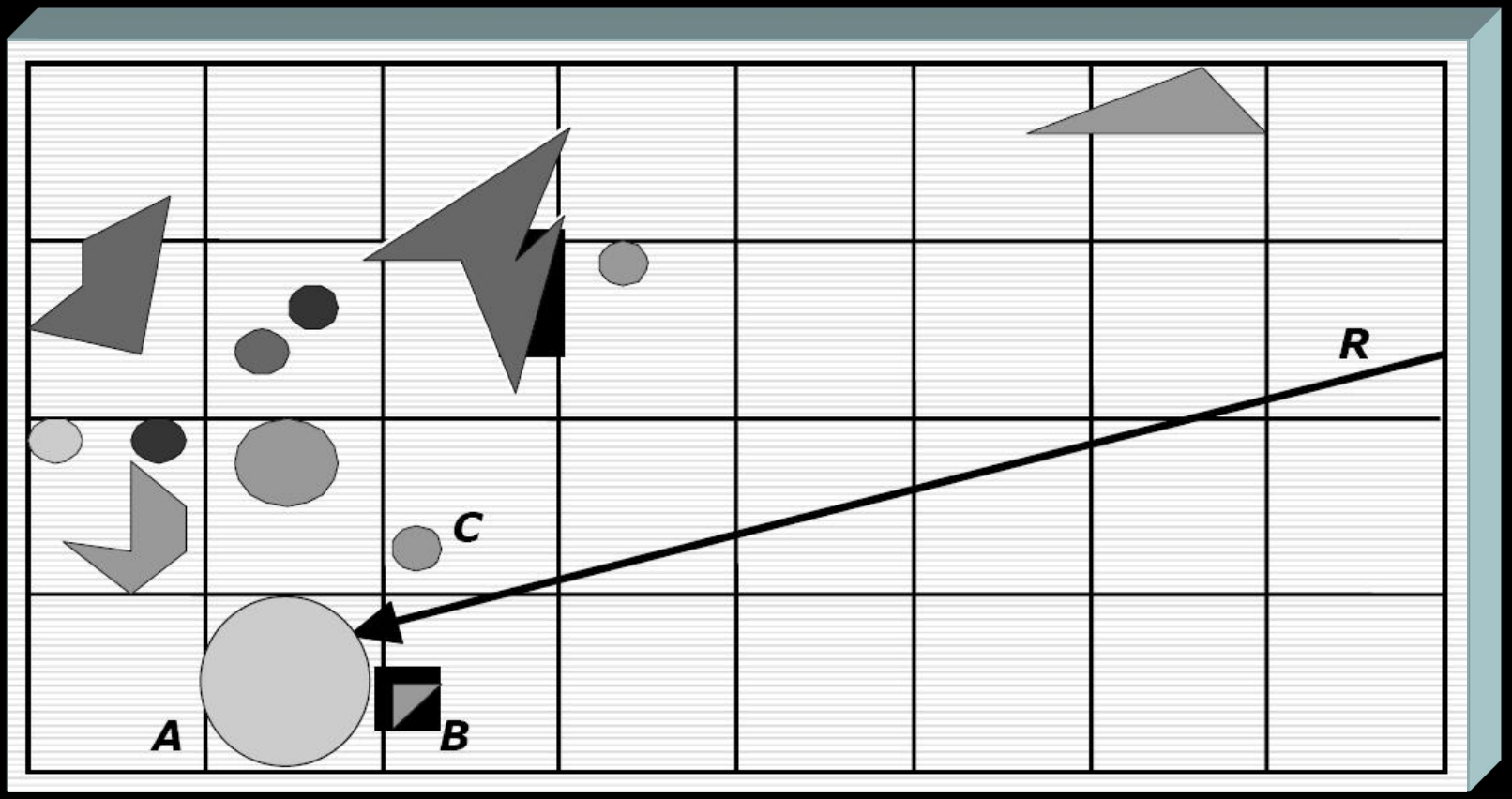
Ray Tracing



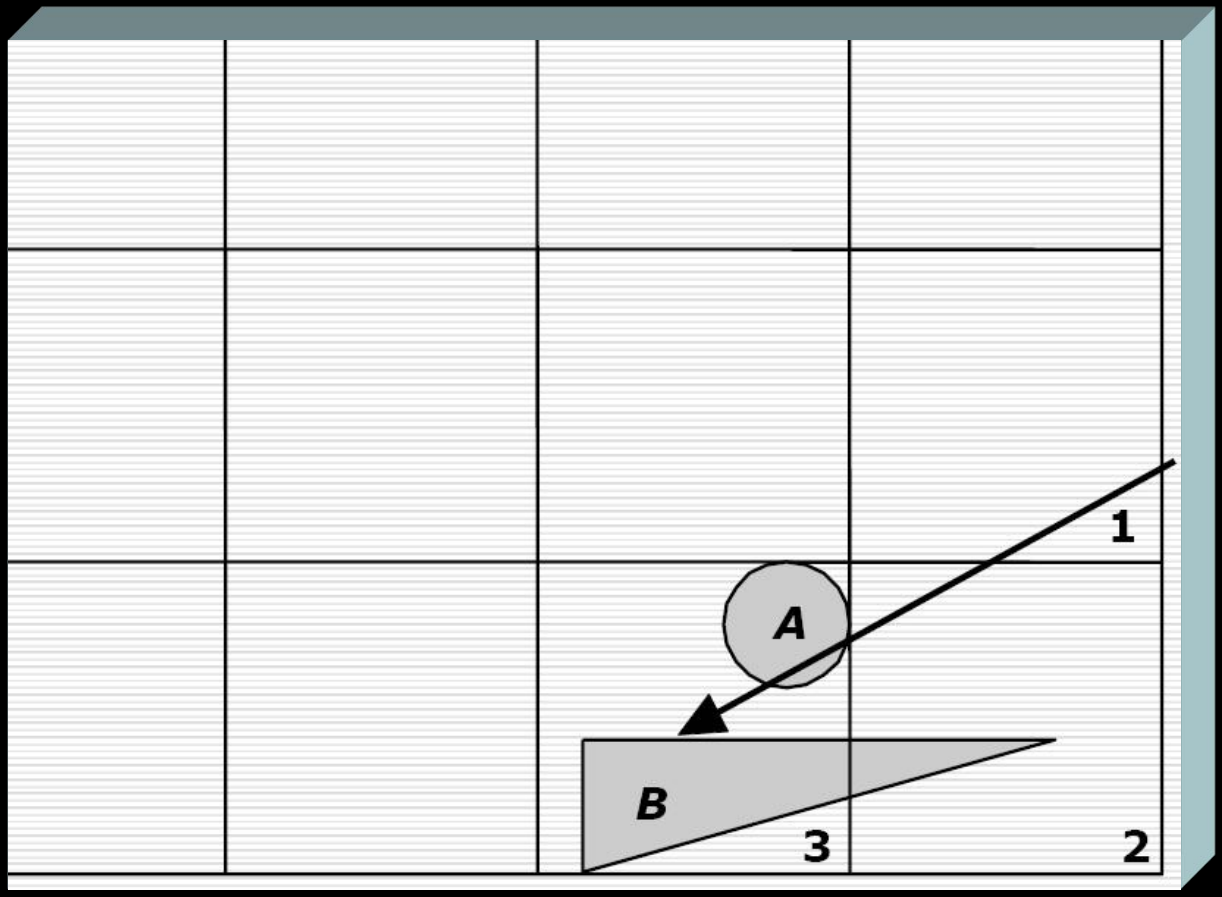
Spatial Partitioning



Spatial Partitioning



Spatial Partitioning

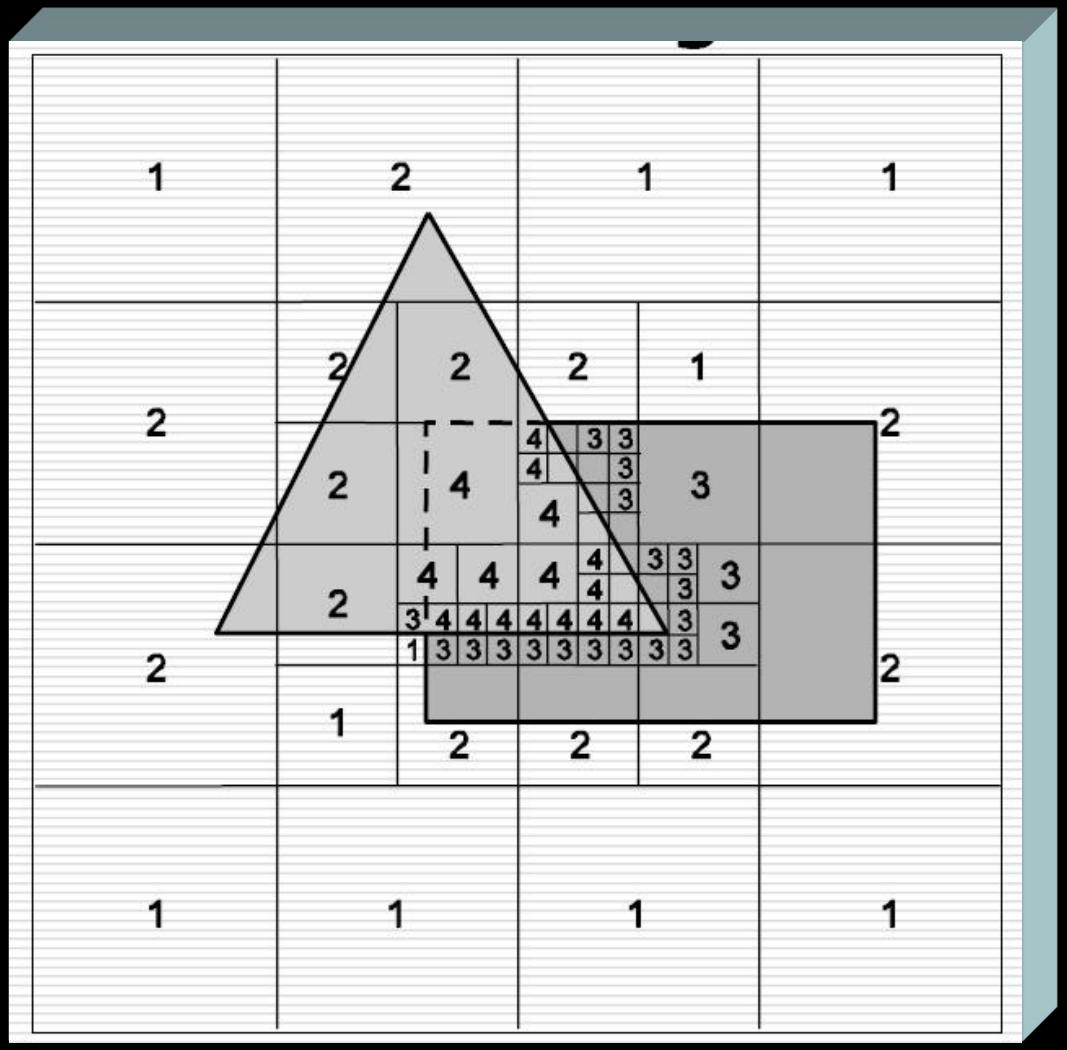


Contents

- 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

Warnock's Algorithm

An area-subdivision algorithm

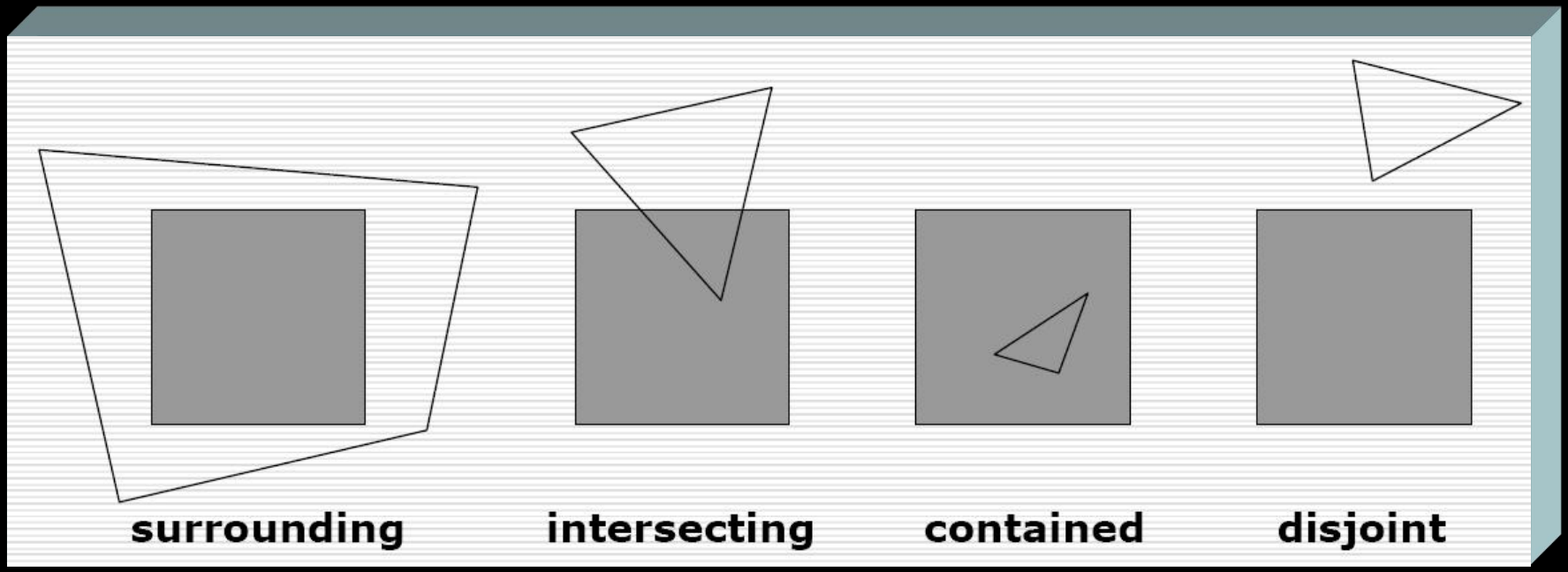


Warnock's Algorithm

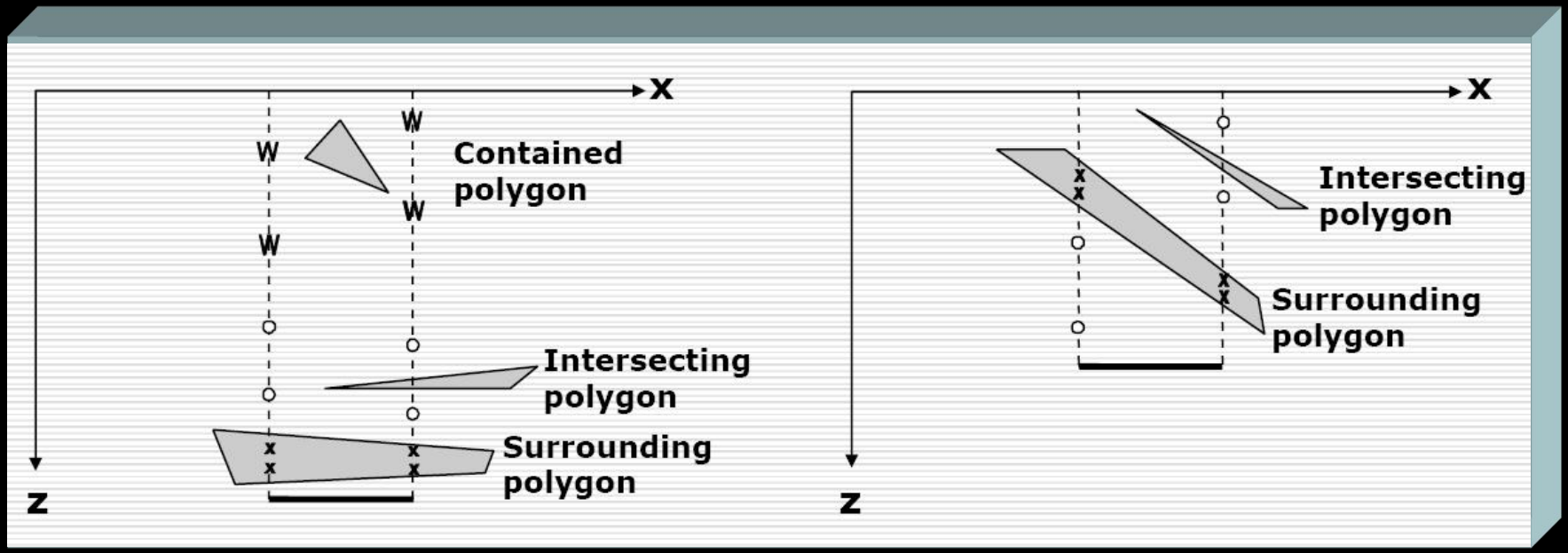
An area-subdivision algorithm

- 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