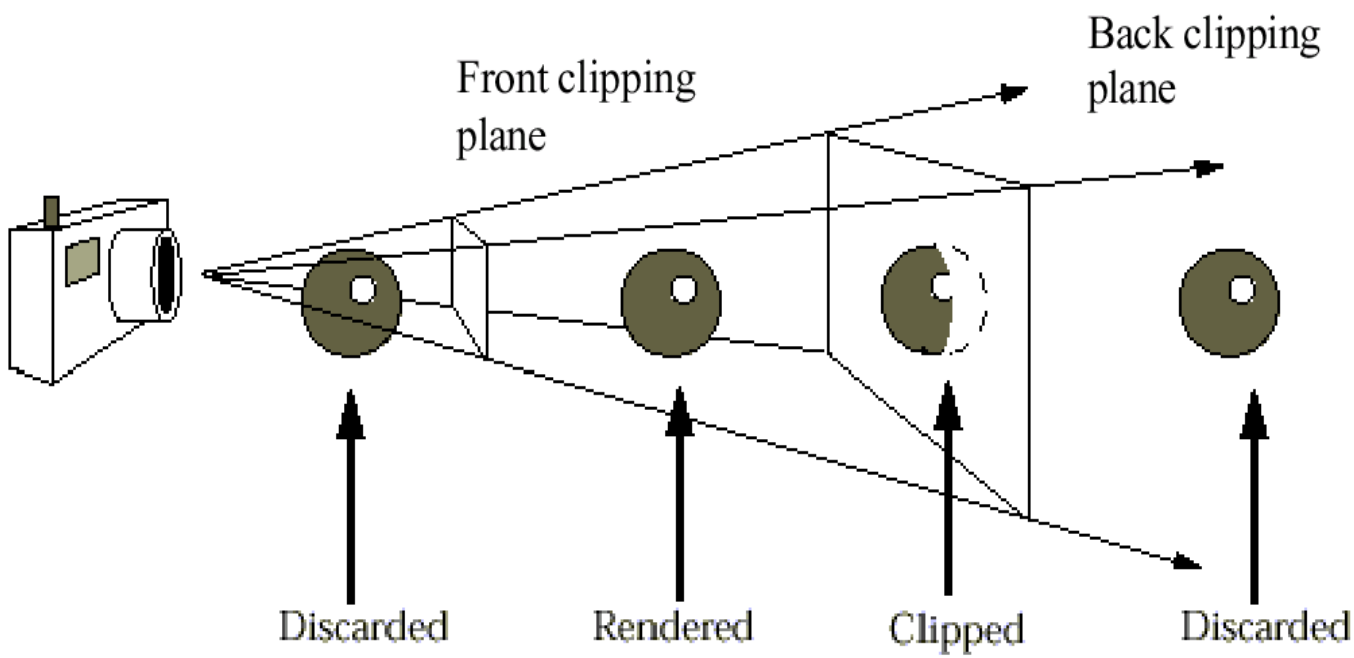
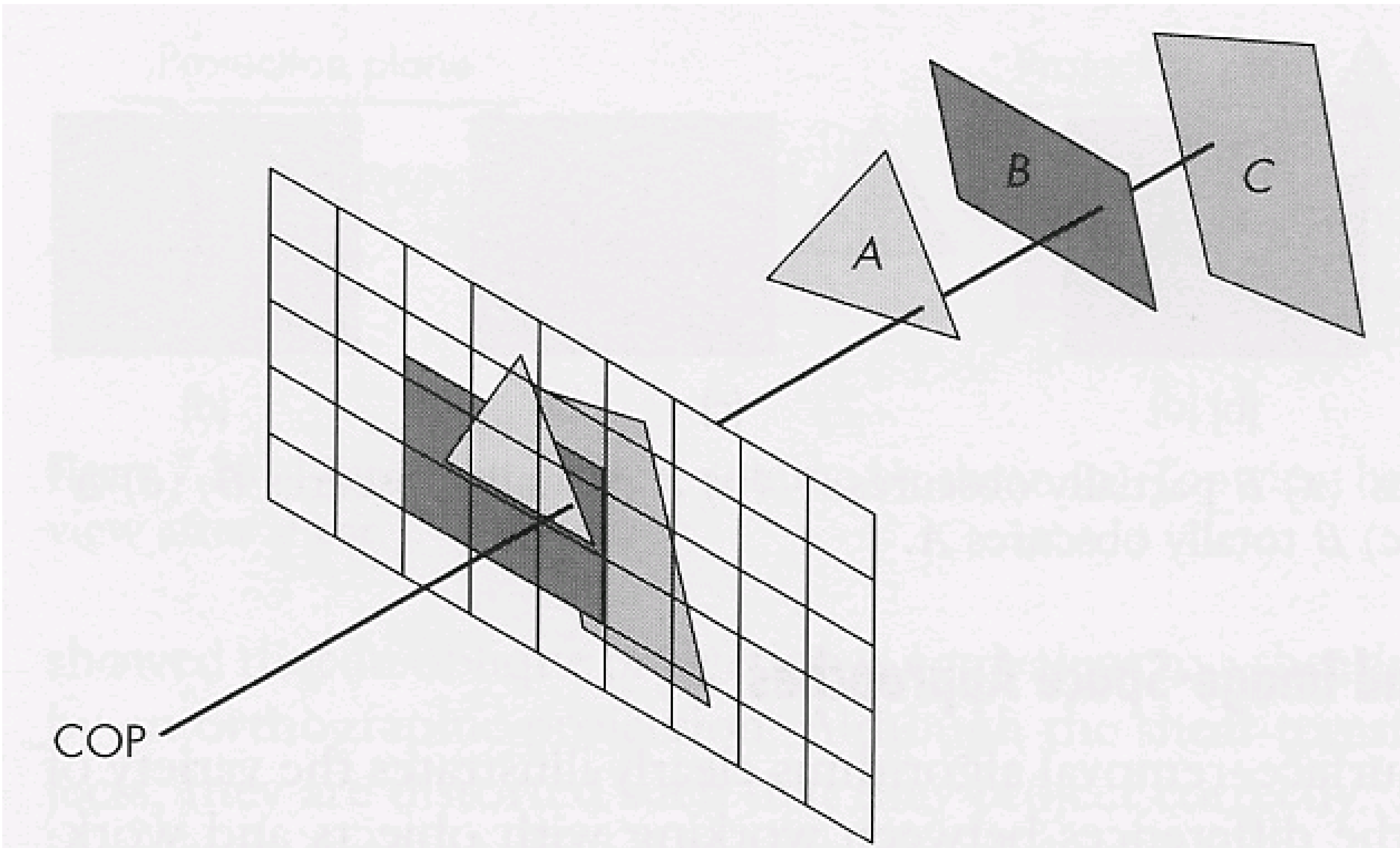
3D Culling and Clipping



Hidden Surface Elimination

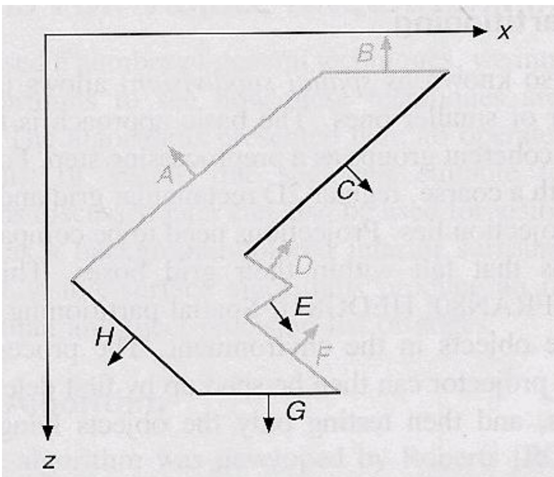


Popular Algorithms

- Back-Face Culling
- Z-Buffer (Depth-Buffer) Algorithm
  - Frame Buffer (Color Buffer) 와
  - Z-Buffer (Depth-Buffer) 를 사용
- BSP Tree 알고리즘

(BSP: Binary Space Partitioning)

Back-Face Detection



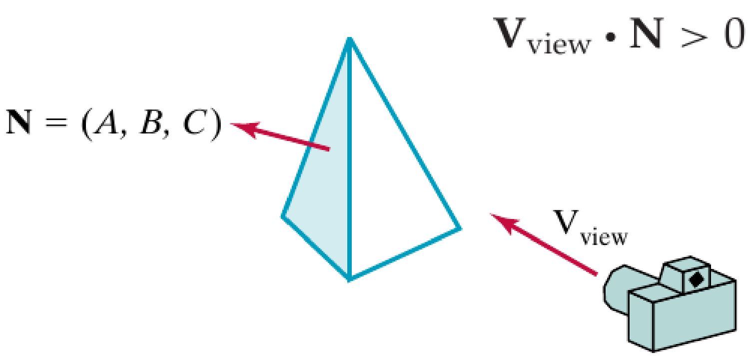


FIGURE 9–1 A surface normal vector **N** and the viewing-direction vector $\mathbf{V}_{\text{view}}$.

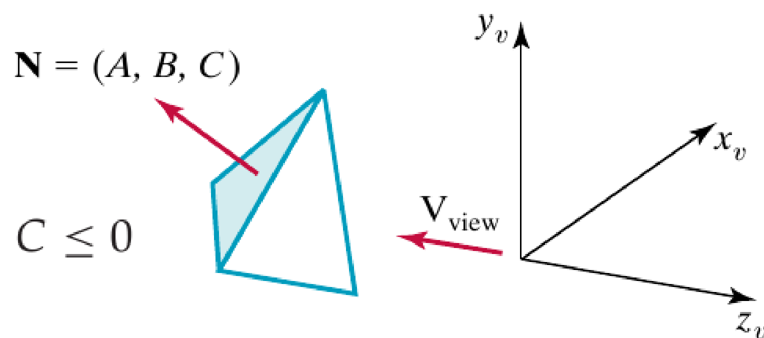
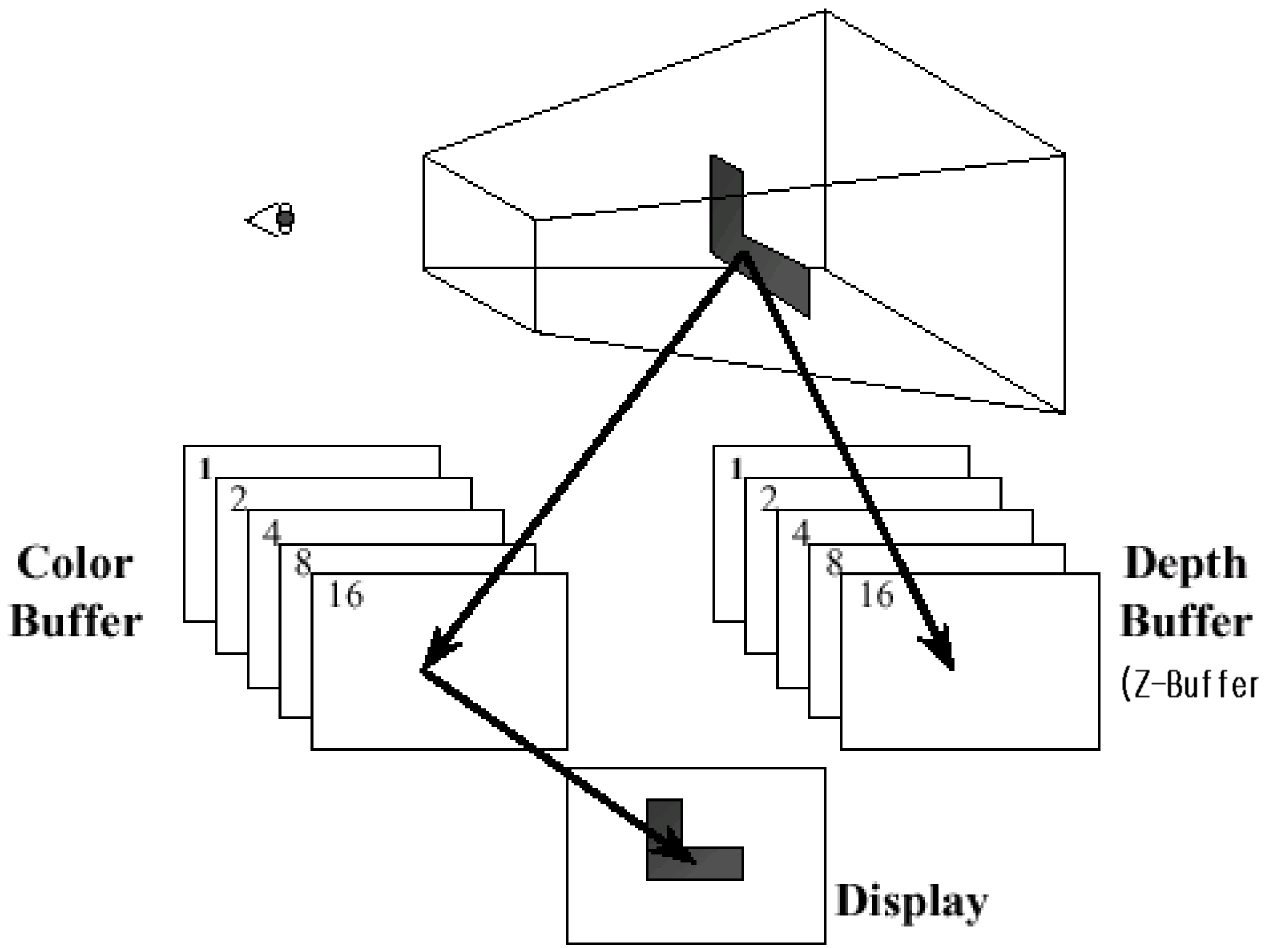


FIGURE 9–2 A polygon surface with plane parameter $C < 0$ in a right-handed viewing coordinate system is identified as a back face when the viewing direction is along the negative z_v axis.

Color Buffer and Depth Buffer



Depth-Buffer Algorithm

1. Initialize the depth buffer and frame buffer so that for all buffer positions (x, y),

```
depthBuff (x, y) = 1.0,    frameBuff (x, y) = backgndColor
```

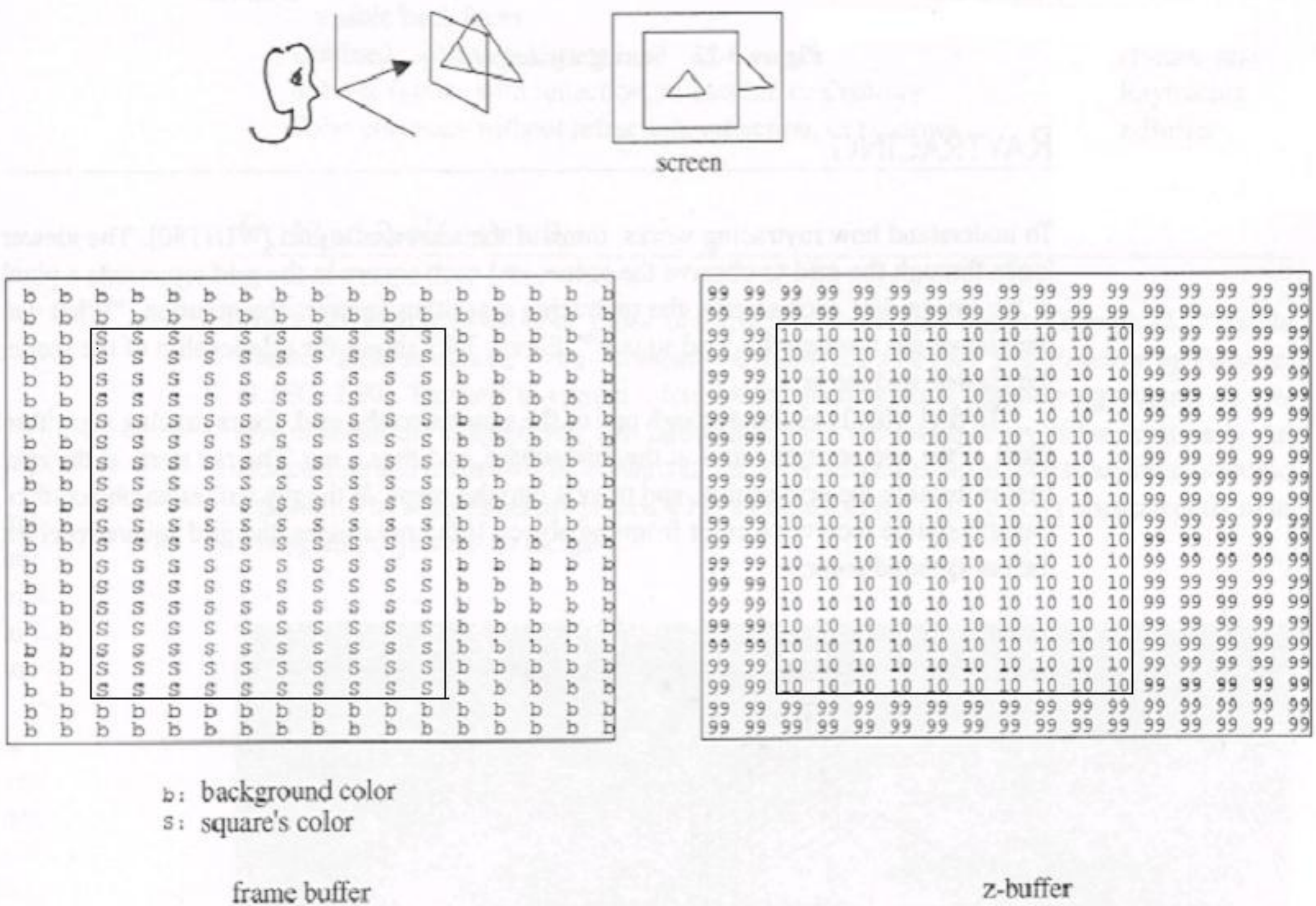
2. Process each polygon in a scene, one at a time.

   - For each projected (x, y) pixel position of a polygon, calculate the depth z (if not already known).
   - If z < `depthBuff (x, y)`, compute the surface color at that position and set

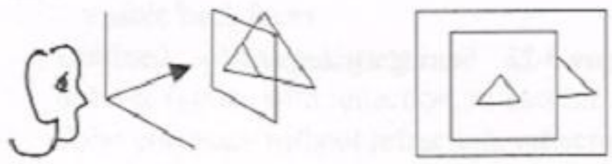
```
depthBuff (x, y) = z,    frameBuff (x, y) = surfColor (x, y)
```

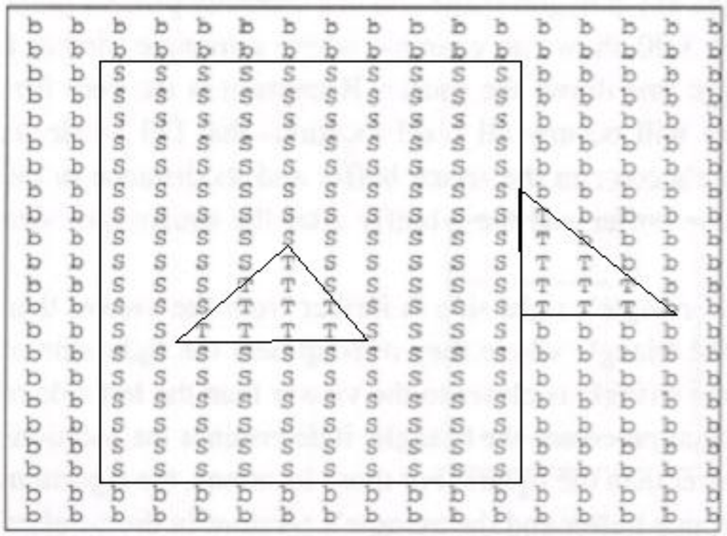
After all surfaces have been processed, the depth buffer contains depth values for the visible surfaces and the frame buffer contains the corresponding color values for those surfaces.

Z-Buffer Algorithm



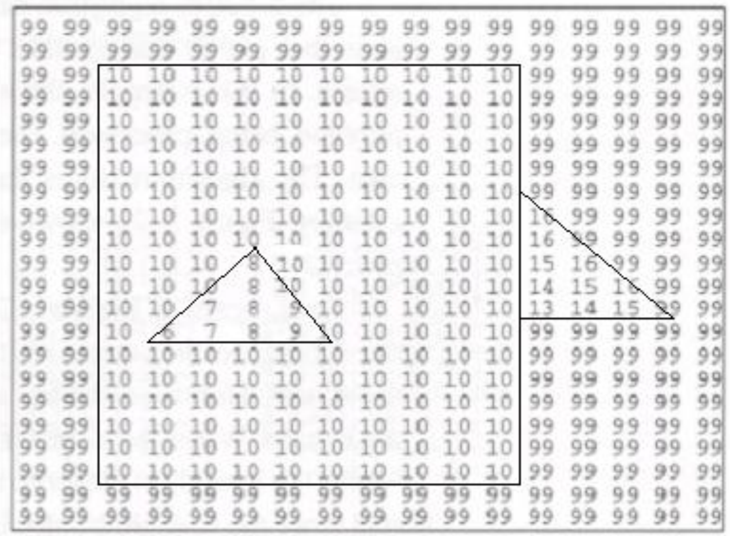
Z-Buffer Algorithm





b : background color
S : square's color
T : triangle's color

frame buffer



z-buffer

Depth Computation for Planes

$$z = \frac{-Ax - By - D}{C}$$

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

$$z' = z - \frac{A}{C}$$

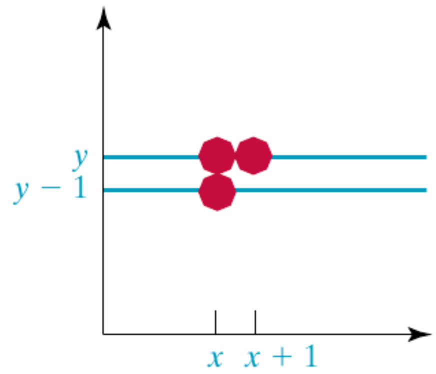


FIGURE 9-5 From position (x, y) on a scan line, the next position across the line has coordinates $(x+1, y)$, and the position immediately below on the next line has coordinates $(x, y-1)$.

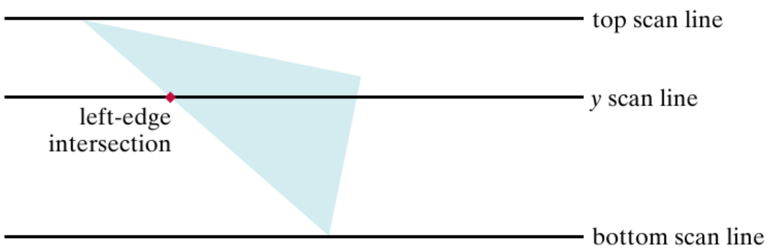


FIGURE 9-6 Scan lines intersecting a polygon surface.

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

$$z' = z + \frac{A/m + B}{C}$$

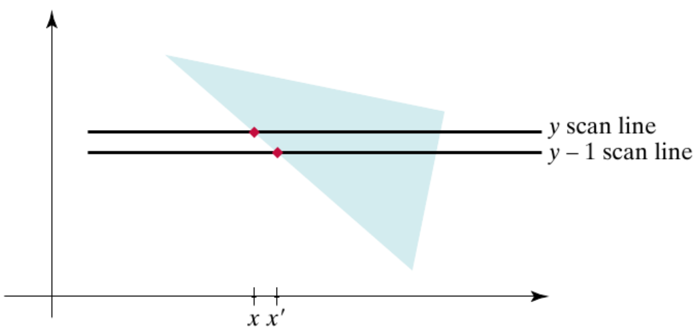


FIGURE 9-7 Intersection positions on successive scan lines along a left polygon edge.

Accumulation Buffer

FIGURE 9–8 Viewing an opaque surface through a transparent surface requires multiple color inputs and the application of color-blending operations.

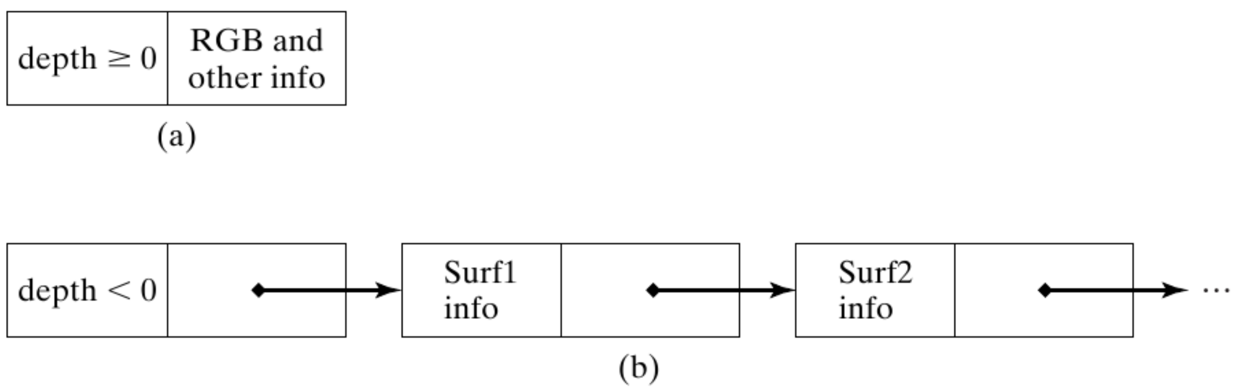
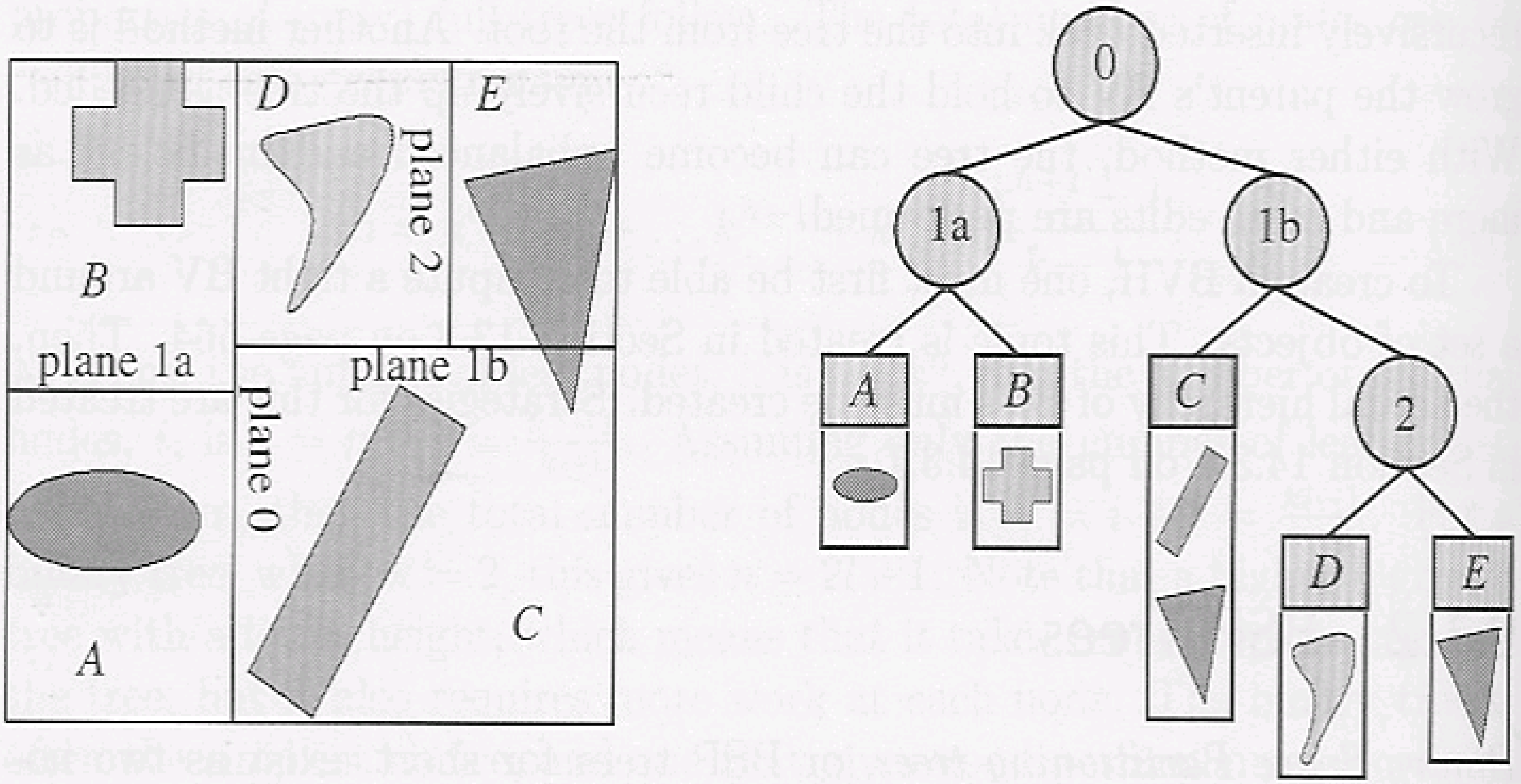
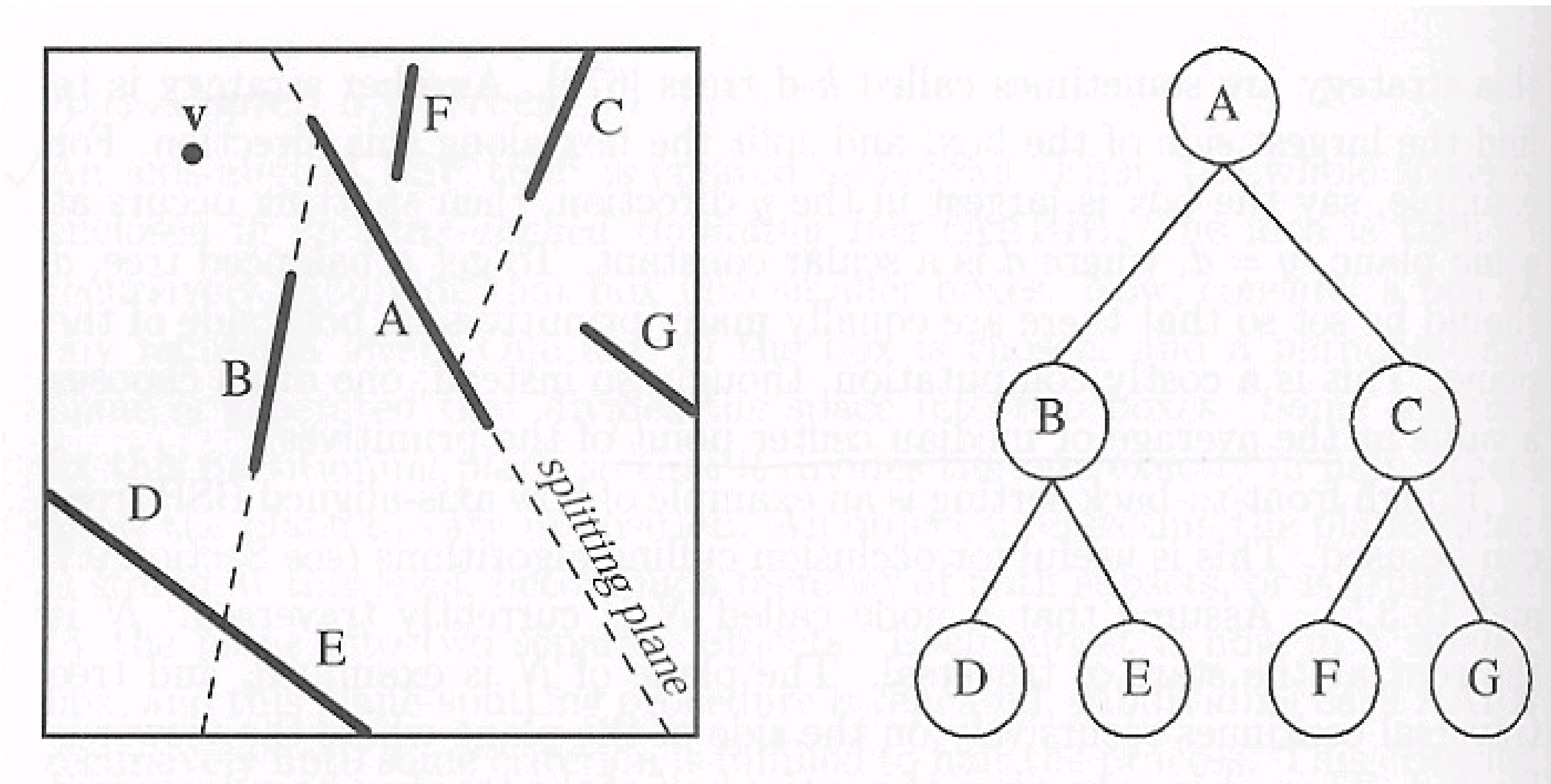


FIGURE 9–9 Two possible organizations for surface information in an A-buffer representation for a pixel position. When a single surface overlaps the pixel, the surface depth, color, and other information are stored as in (a). When more than one surface overlaps the pixel, a linked list of surface data is stored as in (b).

Axis-Parallel BSP Tree

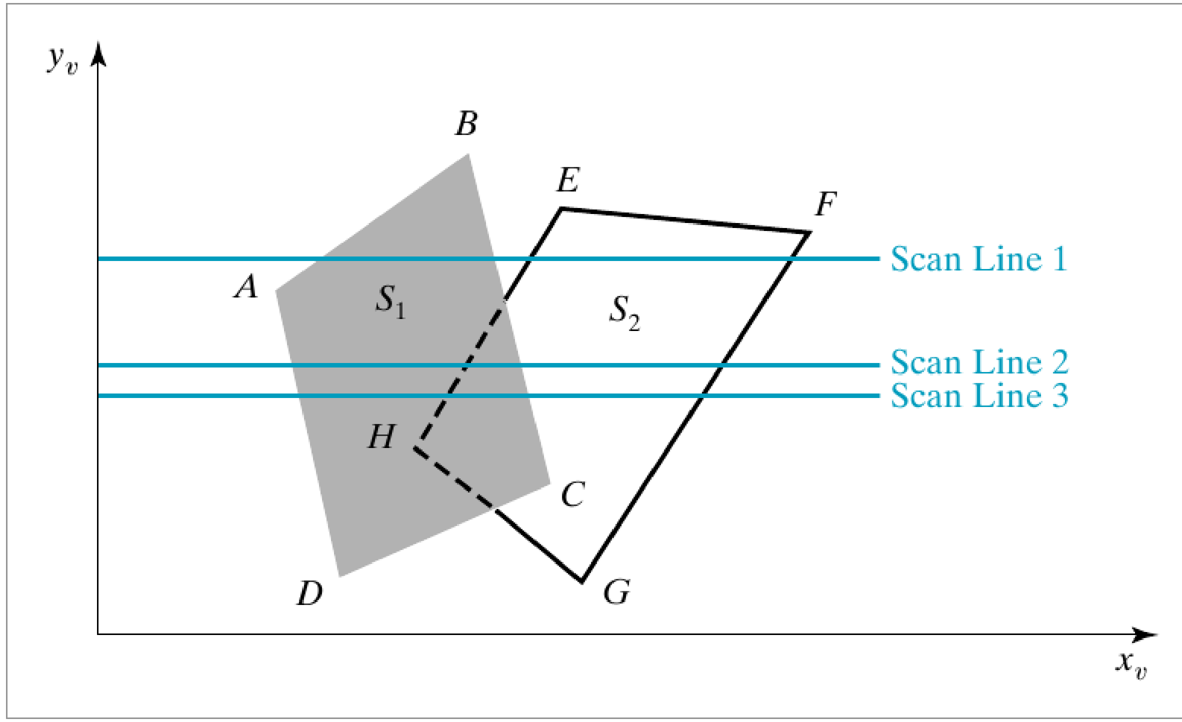


General BSP Tree



Scan-Line Method

FIGURE 9-10 Scan lines crossing the view-plane projection of two surfaces, S_1 and S_2. Dashed lines indicate the boundaries of hidden surface sections.



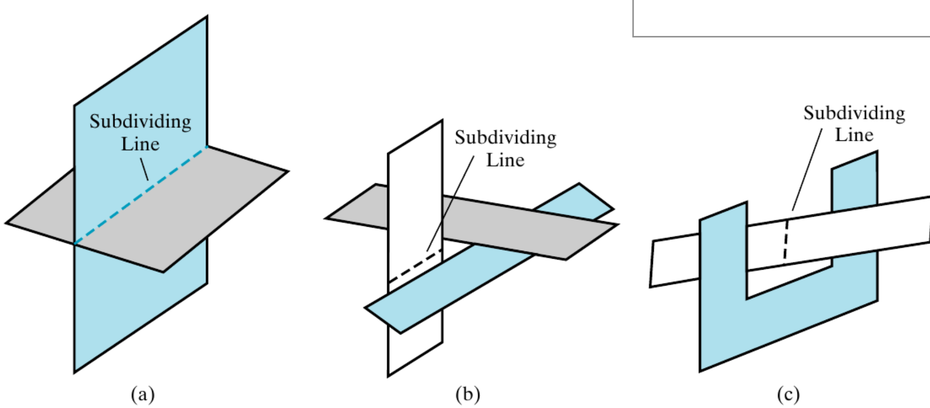
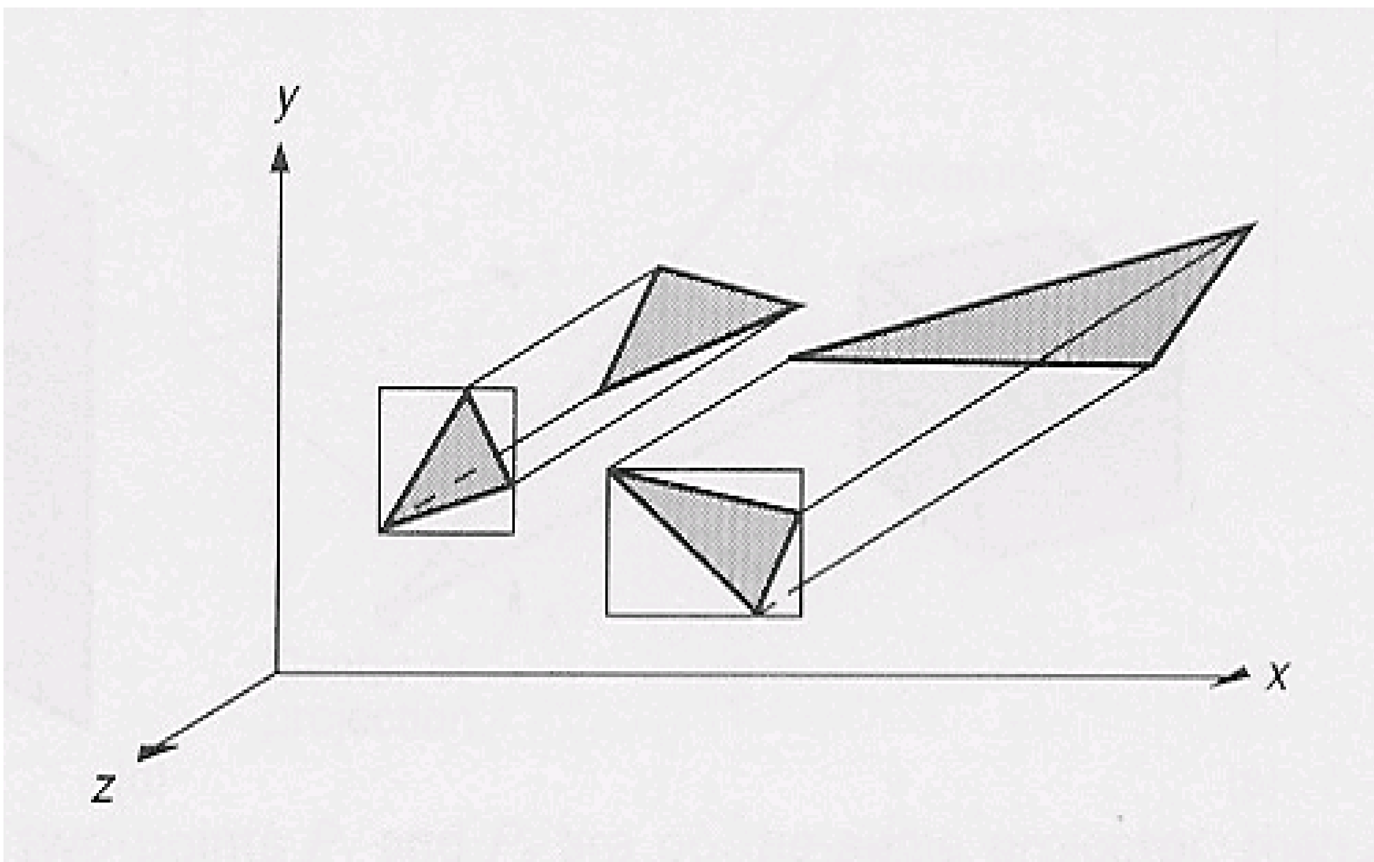


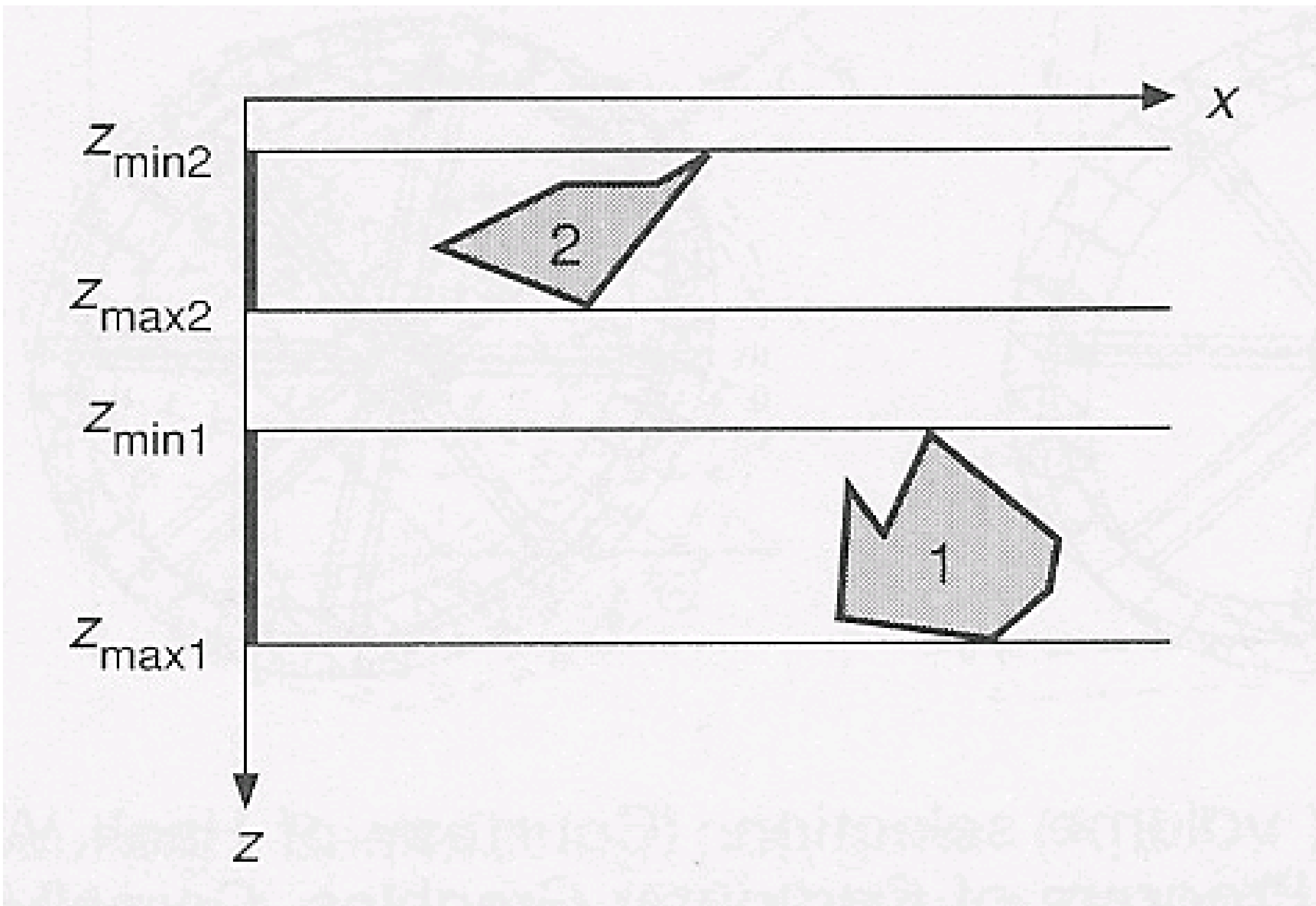
FIGURE 9-11 Intersecting and cyclically overlapping surfaces that alternately obscure one another.

Depth-Sorting Method

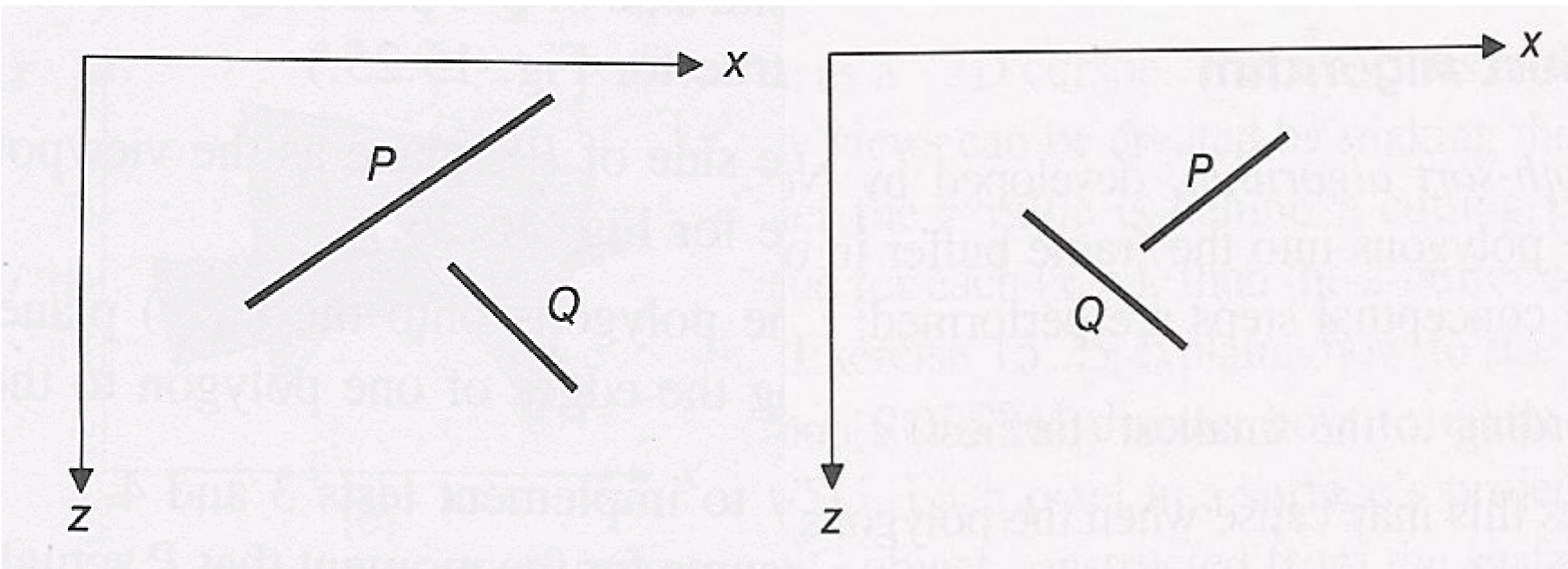
- 투영후 겹치지 않는 경우:
  아무 순서로 그려도 상관 없다.



- 깊이 범위가 다른 경우:
  먼 쪽의 물체를 먼저 가까운 쪽을 나중에.



- 앞뒤 관계가 분명한 경우:
  P를 먼저 그리고, Q를 나중에 그린다.



- 깊이가 겹치는 경우:
  (a) Q를 먼저 그리고, P를 나중에 그린다.
  (b) P를 절단하여 순서를 결정한다.
  (c) R을 절단하여 순서를 정한다.