⇒ **A-buffer method:—**

Z-buffer method can only find one visible surface at each pixel position.
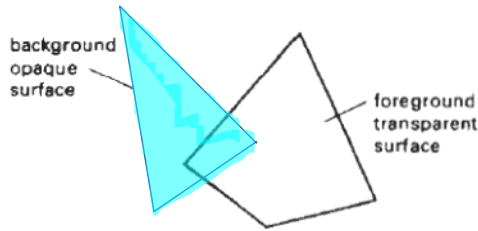
⇒ applicable only for opaque surfaces



Figure 13-8
Viewing an opaque surface through a transparent surface requires multiple surface-intensity contributions for pixel positions.

**A-buffer method:—** (Anti-aliasing, areacoverage, accumulation-buffer)

This method expands the Z-buffer so that each position in the buffer can reference a linked list of surfaces. ⇒ More than one surface intensity can be taken into consideration at each pixel.

Each position in the A-buffer has two fields:—
① Depth field :— stores a positive or negative real number.
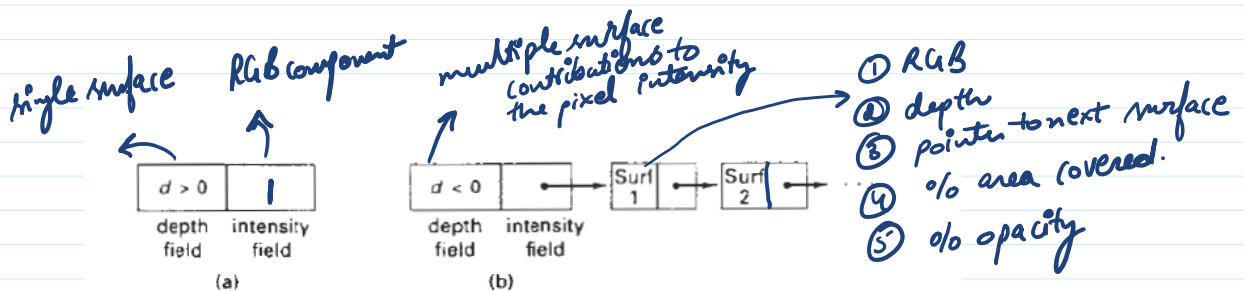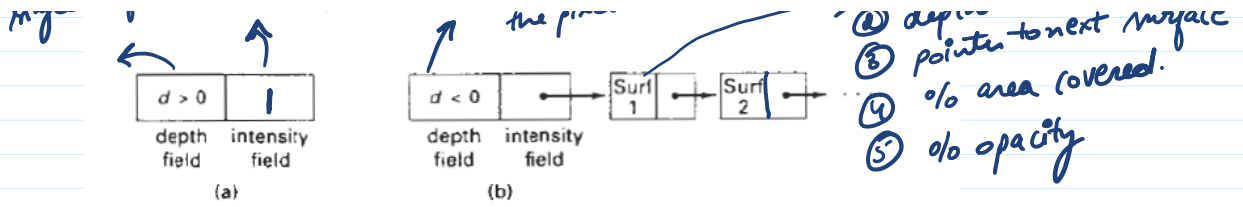② Intensity field :— stores surface intensity info. or a pointer value.



single surface    RGB component    multiple surface contributions to the pixel intensity

| d > 0 | | |
| depth field | intensity field | |
| (a) | | |

| d < 0 | → Surf 1 → Surf 2 → |
| depth field | intensity field | |
| (b) | | |

① RGB
② depth
③ pointer to next surface
④ % area covered.
⑤ % opacity

Figure 13-9

my →    ↑              ↑  the p...          ① d...
                                            ② pointu to next surface
| d > 0 | 1 |        | d < 0 |→•→| Surf 1 |→•→| Surf 2 |→...    ③ % area covered.
depth   intensity      depth   intensity                        ④ % area covered.
field   field          field   field                            ⑤ % opacity
     (a)                    (b)

Figure 13-9
Organization of an A-buffer pixel position: (a) single-surface overlap of
the corresponding pixel area, and (b) multiple-surface overlap.

⟹ **Scan-line method :—**

Dealing multiple polygons at once.

Let all the polygon tables are set up for
the surfaces i.e. edge table & polygon table.



→ flags

            $S_1$      $S_2$
             ↑          ↑
Scan Line 1   $\overline{AB, BC,}$ $\overline{EH FG}$

Scan Line 2   $\underbrace{AD, EH}_{S_1},$ $\underbrace{BC, FG}_{S_2}$
Scan Line 3

              $\underbrace{S_1, S_2,}$
                    ↓
                 depth
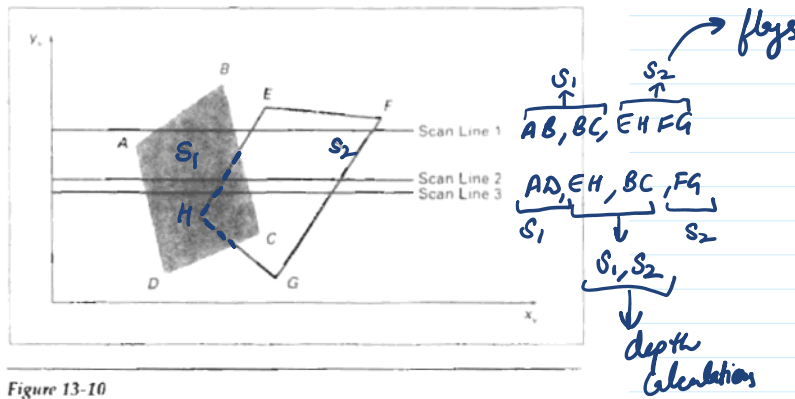                 Calculations

Figure 13-10
Scan lines crossing the projection of two surfaces, $S_1$ and $S_2$, in the
view plane. Dashed lines indicate the boundaries of hidden surfaces.

⟹ Any number of overlapping polygon surfaces
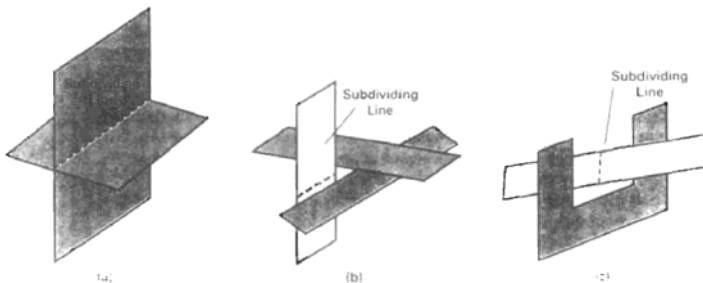can be processed with scan-line VSD method.
using flags.



Figure 13-11
Intersecting and cyclically overlapping surfaces that alternately obscure one another.

⟹ **Depth Sorting / Painter's Algorithm :—**

Image-space + Object Space Operations

① Surface are sorted in the order of decreasing depth.

② Surfaces are scan converted in order, starting with surface of greatest depth

Let surface S is the surface with the greatest depth.

S is compared with to other surfaces in the list.
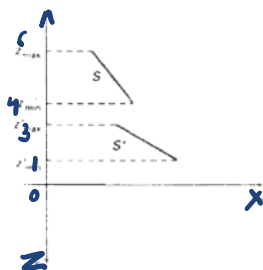
For each surface that overlaps with S,



Figure 13-12
Two surfaces with no depth overlap.

The tests are listed in the order of increasing difficulty.

**TEST 1 :-** The bounding rectangles in the xy plane for the two surfaces do not overlap.

**TEST 2 :-** Surface S is completely behind the overlapping surface relative to the viewing position.

**TEST 3 :-** The overlapping surface is completely in front of S relative to the viewing position.

**TEST 4 :-** The projections of the two surfaces onto the view plane do not overlap.
( checking for bounding boxes overlapping )
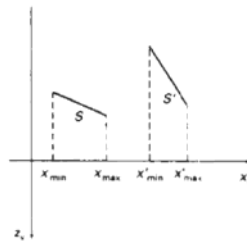


Figure 13-13
Two surfaces with depth overlap but no overlap in the x direction.
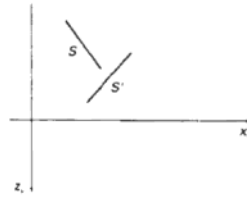


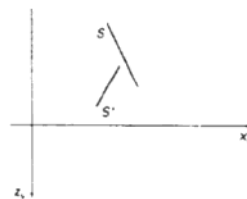Figure 13-14
Surface S is completely behind ("inside") the overlapping surface S'.



Figure 13-15
Overlapping surface S' is completely in front ("outside") of surface S, but S is not completely behind S'
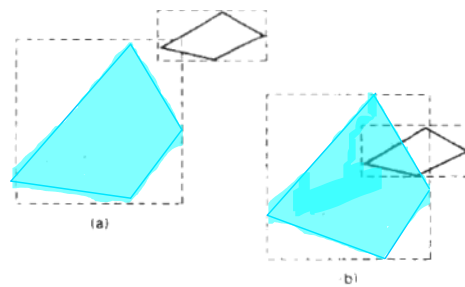


Figure 13-16
Two surfaces with overlapping bounding rectangles in the xy plane.

These tests are performed in the order listed and the next overlapping is proceeded as soon as we find one of these tests is true.

If all the overlapping surfaces pass at least one of these tests, none of them is behind S. No reordering is then necessary and S is scan converted.
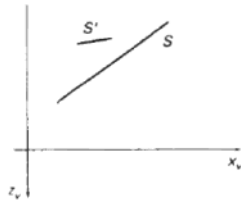
*Figure 13-17*
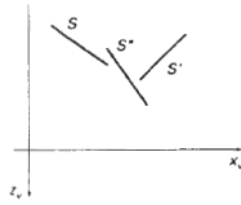Surface S has greater depth
but obscures surface S'.



*Figure 13-18*
Three surfaces entered into
the sorted surface list in the
order S, S', S" should be
reordered S', S", S.

All tests failed for an overlapping surface S', we interchange surfaces S and S' in the sorted list.