Z-Buffer

Use auxiliary Z-buffer to hold depth of currently active surface
0. Initially, all $z = -\infty$, surface colours are background
1. Scan convert each polygon
2. If poly's pixel's $z >$ current $z$, replace current colour with poly's colour, current $z$ with poly's $z$

Can use coherence of polygon to perform incremental update on $z$ during scan line conversion

If plane equation of poly is $Ax + By + Cz + D = 0$
$z = -D/Ax-By/C$

Given starting $z$, $z_1$ at $(x + \Delta x, y)$ is
$z_1 = z - \Delta z$ if $\Delta x = 1$
$z_1 = z - \frac{\Delta y}{C}$ if $\Delta y = 1$

Similarly, $z_1$ at $(x, y + \Delta y)$ is
$z_1 = z - \frac{\Delta x}{A}$ if $\Delta y = 1$

Advantages:
1. Simple to implement
2. Can handle objects other than polygons (but must change incremental Z computations)
3. Time to run tends to be independent of number of polygons, as $\#poly \uparrow$, size poly $\downarrow$
Disadvantages:
1. requires memory buffer
2. precision is a problem for 16, 32-bit integer buffers (z-fighting)
3. aliasing is a problem (polys cannot "share" a pixel)

A - Buffer

Similar to z-buffer, but each depth entry is a list of overlapping pixels, percentage of area covered, depth, and opacity.

Used for antialiasing, sub-pixel resolution support.