Z-buffer

The **Z-buffer** or **depth buffer** algorithm [Catmull, 1974] is probably the simplest and most widely used.

Here is pseudocode for the Z-buffer hidden surface algorithm:

```
for each pixel (i,j) do
    Z-buffer [i,j] ← FAR
    Framebuffer[i,j] ← <background color>
end for
for each polygon A do
    for each pixel in A do
        Compute depth z and shade s of A at (i,j)
        if z > Z-buffer [i,j] then
            Z-buffer [i,j] ← z
            Framebuffer[i,j] ← s
        end if
    end for
end for
```

Q: What should FAR be set to?

Far clipping plane depth
– Big number

Z-buffer: Analysis

- Easy to implement? Y
- Easy to implement in hardware? Y
- Incremental drawing calculations (uses coherence)? Y
- Pre-processing required? N
- On-line (doesn't need all objects before drawing begins)? Y
- If objects move, does it take more work than normal to draw the frame? N
- If the viewer moves, does it take more work than normal to draw the frame? N
- Typically polygon-based? Y
- Efficient shading (doesn't compute colors of hidden surfaces)? N
- Handles transparency? N
- Handles refraction? N

A over B over C

$$\alpha_A A + (1-\alpha_A)(\alpha_B B + (1-\alpha_B)C) \neq$$

$$\alpha_B B + (1-\alpha_B)(\alpha_A A + (1-\alpha_A)C)$$