Computer Graphics Charles University

Z-buffer

CGG MFF UK Praha

pepca@cgg.mff.cuni.cz

http://cgg.mff.cuni.cz/~pepca/

Z-buffer

- **Drawing to a buffer**
  - Video RAM, raster printer with a buffer
- **Area filling**
  - Even with patterns
- **No sorting necessary**
- **Correct drawing in all situations**
  - Intersecting polygons, cyclic coverage, ...

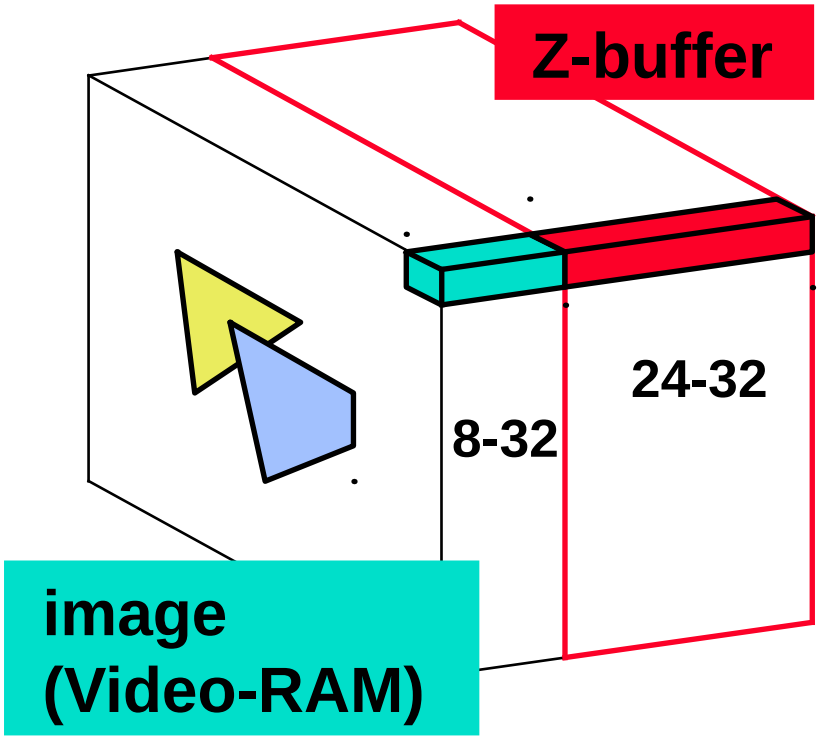
Depth buffer

For each pixel we store:

- **colour** (Video-RAM)
- **depth** = distance from observer (**Z-buffer**)

float: simple (newer GPUs)

integer: fast (24-32 bits), problems with accuracy



Algorithm:

1. **initialisation:**
   - Video-RAM := *background colour*
   - Z-buffer := "*infinity*"
2. **Writing all objects to the buffer:**
   - Writing to individual pixels (polygon filling)
   - Depth test

```
WritePixel ( x, y, z, colour : integer );
  if z < Zbuf[x,y] then begin
   Zbuf[x,y] := z;
   PutPixel(x,y,colour);
  end;
```

Z-buffer Advantages

- **Simple calculations**
  - Integer arithmetic
  - HW implementation: 500k to 100M polygons/s
- **No sorting necessary**
- Correct rendering of **non-standard configurations**
- **Not limited** to drawing of **planar surfaces**
  - Any object that can be broken down into pixels works (as long as a depth for the pixel can be computed)

Z-buffer Disadvantages

- **Memory requirements** (this used to be a problem!)
  - 1024 x 768 x 24 bits = 2.3 MB
  - 1920 x 1200 x 32 bits = 8.8 MB
  - **Erasing the memory** at the beginning of each frame
- In most cases, some pixels in Video-RAM are **written to many times due to overdraw** (inefficient)

Saving Memory (outdated)

- **Drawing in strips**
  - Z-buffer only for image strips
  - Several iterations needed to complete picture (one rendering pass for each strip)
  - Clipping
- **„Single line Z-buffer”**
  - Every scanline is done separately
  - Higher efficiency: a list of active objects is maintained

End

Further information:

- **J. Foley, A. van Dam, S. Feiner, J. Hughes**: ***Computer Graphics, Principles and Practice***, 668-672
- **Jiří Žára a kol.**: ***Počítačová grafika***, principy a algoritmy, 298-300