

Z-buffer, depth-buffer

© 1995-2019 Josef Pelikán
CGG MFF UK Praha

pepca@cgg.mff.cuni.cz
<https://cgg.mff.cuni.cz/~pepca/>



Z-buffer, depth-buffer (paměť hloubky)

Kreslení do bufferu

- video-RAM, GPU, rastrová tiskárna s bufferem

Vyplňování ploch

- lze i stínovat

Není třeba nic třídit!

Korektní vykreslení nestandardních situací

- prosekávání stěn, cyklické zákryty...



Paměť hloubky

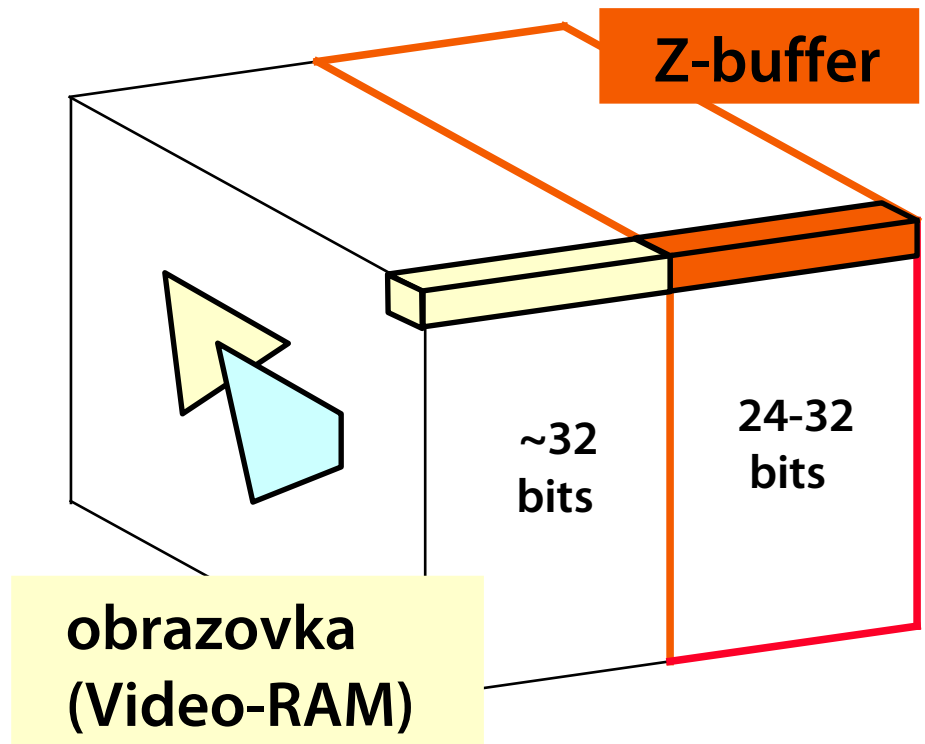
Pro každý pixel se ukládá

- **barva** (Video-RAM)
- **hloubka** = vzdálenost od pozorovatele (**Z-buffer**)

float – jednodušší (nové GPU)

Integer – rychlejší (24-32 bitů)

- pozor na přesnost!





Algoritmus

Inicializace

- fillBuffer(VideoRAM, „barva pozadí“);
- fillBuffer(Zbuf, „nekonečno“);

Zápis všech objektů do Z-bufferu

- rozložení na jednotlivé pixely (**rasterizace**, vyplňování)
- **test hloubky** jednoho pixelu:

```
void writePixel (int x, int y, float z, color color)
{
    if (z < zbuf[x, y])
    {
        zbuf[x, y] = z;
        videoRAM[x, y] = color;
    }
}
```



Výhody Z-bufferu

Jednoduchost výpočtů

- celočíselná/fixed-point aritmetika
- možnost masivní paralelizace (viz GPU)
- HW implementace 10M až 10G plošek/s

Není nutné třídění

Správné vykreslení všech obtížných situací

Nemusí se kreslit pouze rovinné plošky

- rutina pro rozklad objektu na pixely (s výpočtem hloubky z)



Nevýhody Z-bufferu

Mazání bufferu na začátku zpracování snímku

Některé pixely ve Video-RAM se **několikanásobně překreslují**

- zbytečné vyhodnocování jejich barvy (fragment shader)

„Z-fighting“

- nepříjemné artefakty při kreslení objektů ve stejné rovině
- velmi rušivé při animacích
- je to často chybou modelu nebo špatně nastavených parametrů projekce

Neřeší jednoduše **poloprůhledné 3D scény**

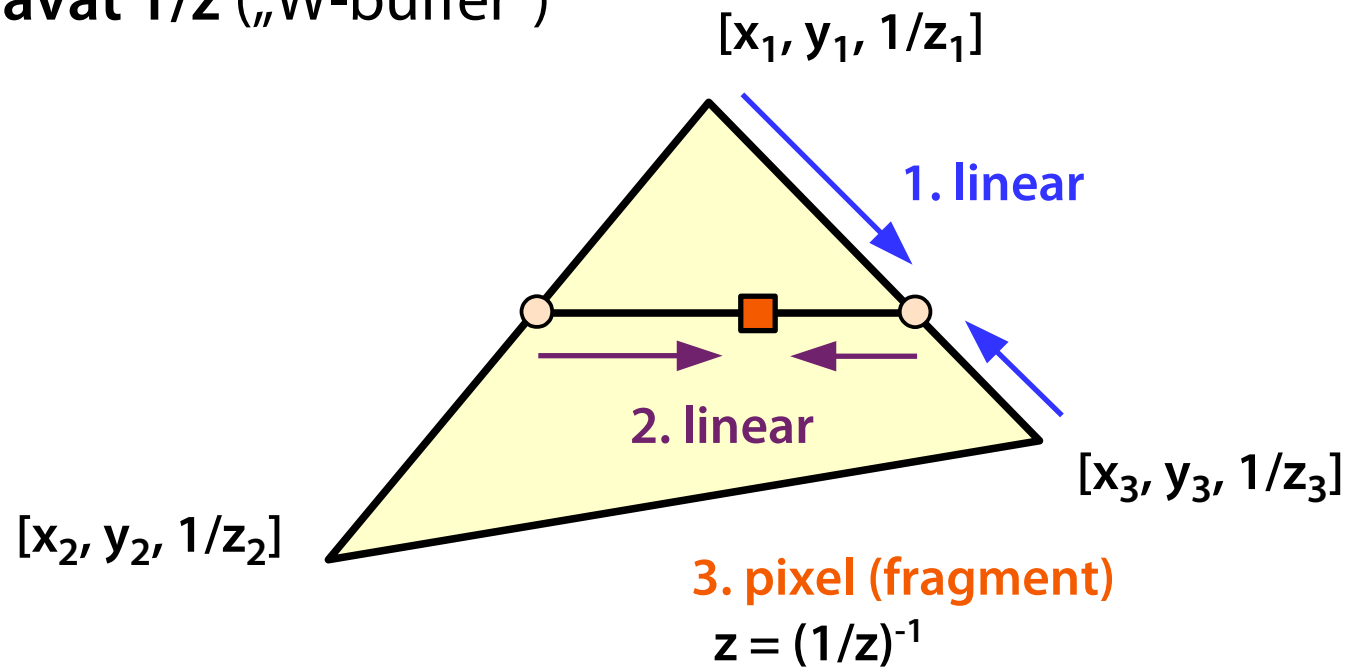
- nutnost předzpracování (3D třídění primitivů)

Perspektivně korektní interpolace hloubky



V perspektivě nemůže interpolátor (rasterizer) na ploše průmětny přímo lineárně interpolovat hloubku z

- místo toho lze interpolovat převrácené hodnoty $1/z$
- v každém pixelu můžeme z spočítat dělením nebo přímo porovnávat $1/z$ („W-buffer“)





Literatura

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

Kok-Lim Low: *Perspective-Correct Interpolation* (report, proof), University of North Carolina at Chapel Hill, 2002