

Programovatelné zpracování vrcholů a pixelů v DirectX 8

Ondřej Španěl

Bohemia Interactive Studio

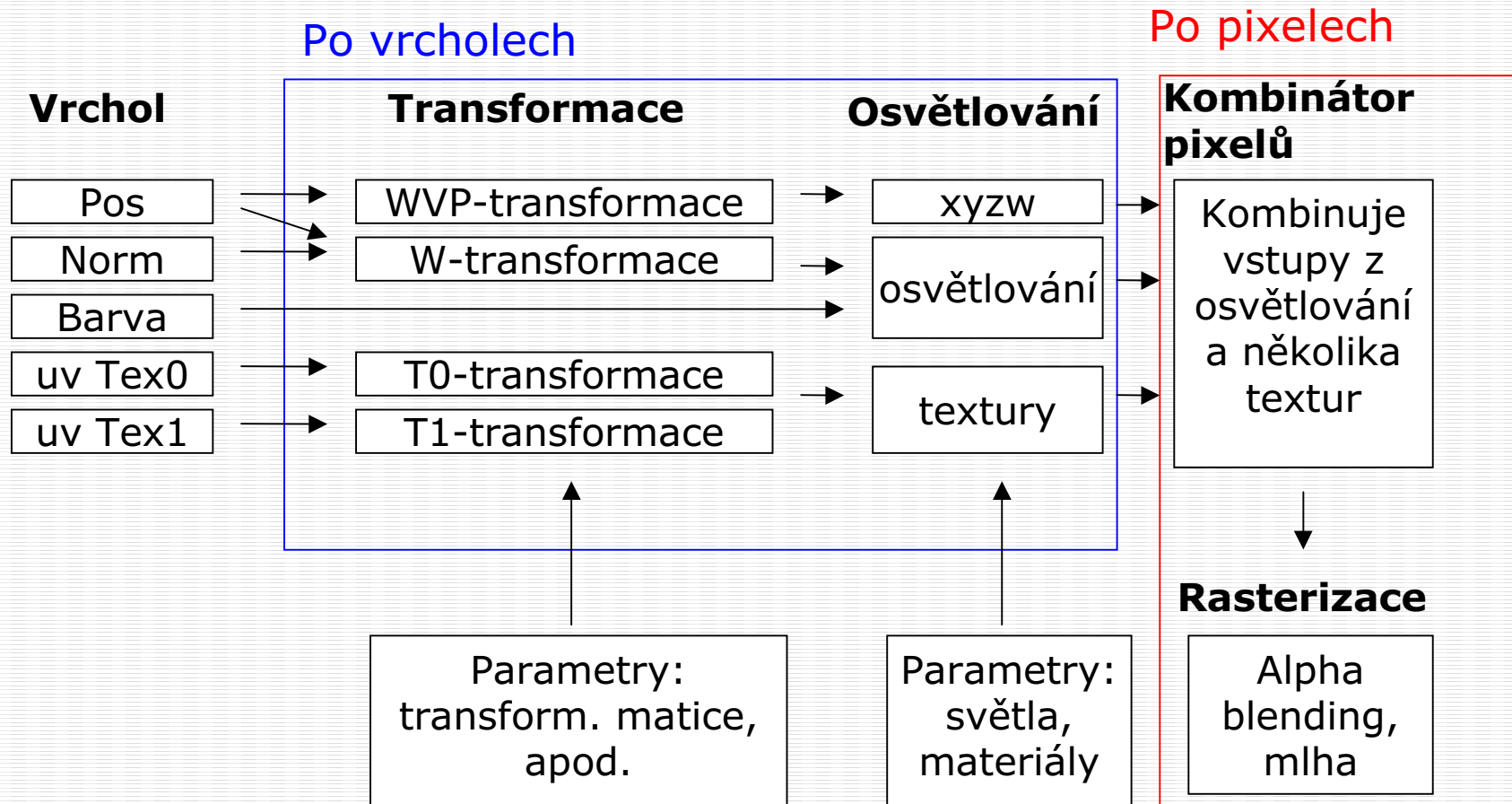
www.bistudio.com

Jen pro DirectX?

- V DirectX od verze 8
- Lze čekat obdobné funkce v OpenGL, zatím existuje jako nVidia specifické rozšíření.
- Stínovače vrcholů poměrně stabilní
- Stínovače pixelů se pravděpodobně budou ještě hodně měnit – současná podoba není moc obecná.

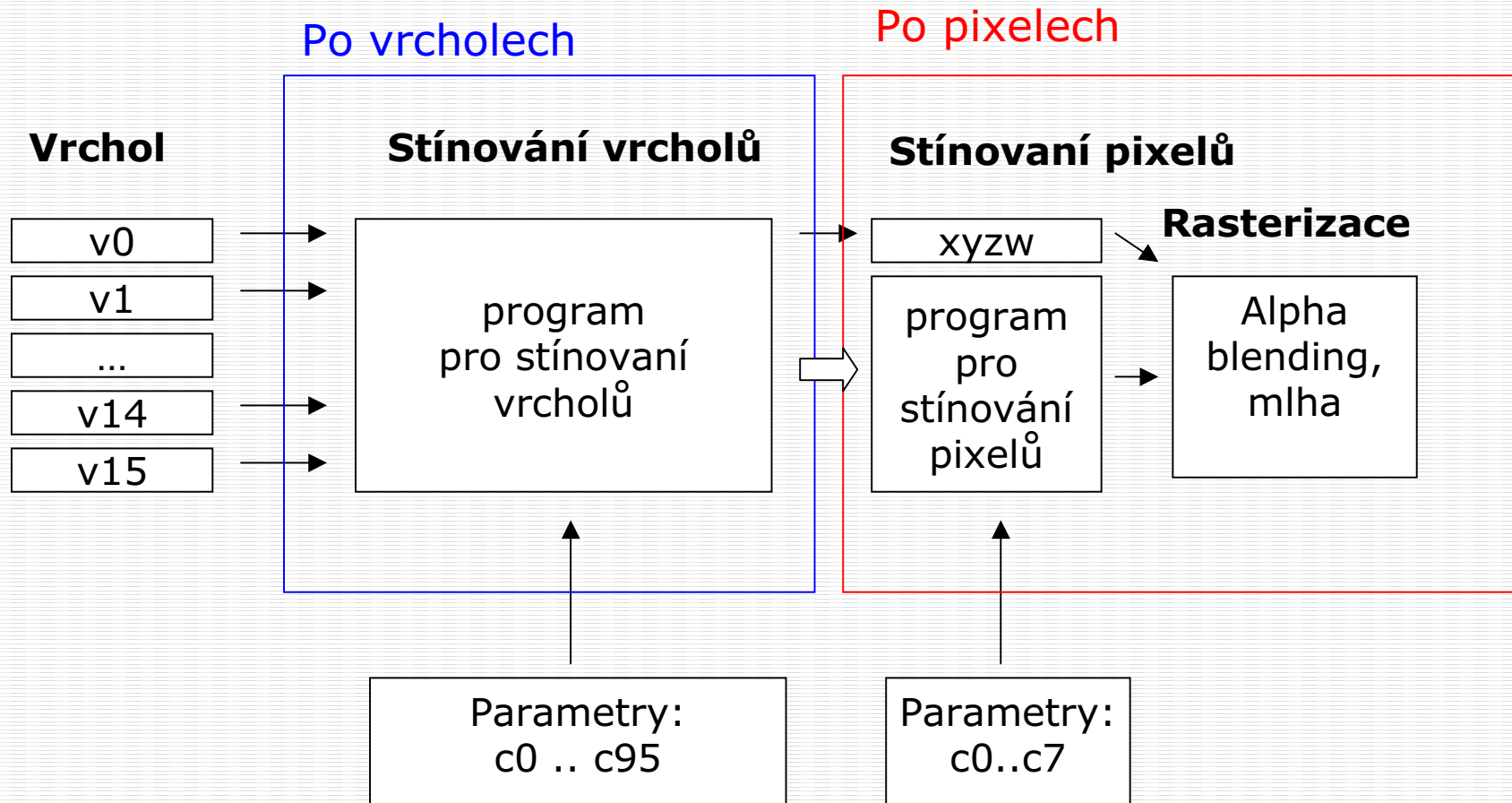
Pevně nastavené zpracování

(Fixed function processing)



Programovatelné zpracování

(Programmable processing)



Stínování vrcholů

(Vertex shader)

- Vstupy: vn - 16 vektorů
- Konstanty: cn - 96 vektorových konstant
- Výstupy: $oPos$, oDn , oTn , $oFog$
- Dočasné registry: an - 1 skalár, rn - 12 vektorů
- Instrukce: add , mul , mad , $dp3$, $dp4$
Příklad: $mad\ r0,\ r3,\ r4,\ r0$

Stínování pixelů

(Pixel shader)

- Vstupy: vn - 2 vektory, tn - 4 vektory
- Konstanty: cn - 8 vektorových konstant
- Výstup: $r0$
- Dočasné registry: rn - 2 vektory
- Adresování textur: tex , $texcoord$
- Instrukce: add , mul , mad , $dp3$, lrp
Příklad: $mad\ r0,\ r3,\ r4,\ r0$

Jednoduchý stínovač vrcholů

- Transformace a difúzní osvětlení

```
dp4 r0.x, v0, c[0]
```

```
dp4 r0.y, v0, c[1]
```

```
dp4 r0.z, v0, c[2]
```

```
dp4 r0.w, v0, c[3]
```

```
mov oPos, r0
```

```
dp3 r1.x, v1, c[5] ; směr světla
```

```
max r1, r1.x, c[6] ; c[6] = (0,0,0,0)
```

```
add oD0, r1.x, c[7] ; ambient
```

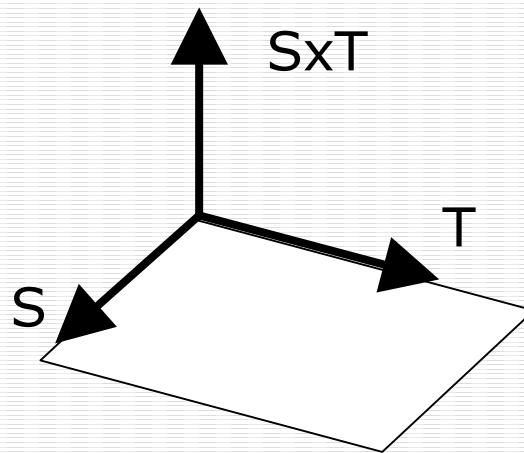
Jednoduchý stínovač pixelů

- Simulace tyčinek a čípků

```
tex t0                ; vezmi texel
mul r0, v0, t0        ; v0 - výstup z osvětlení
dp3_sat r1,r0,c[0]    ; urči jas, c0 = citlivost oka
add_sat r1.a,r1.a,c[1].a ; c1.a udává míru efektu
lrp r0.rgb, r1.a, r0, r1 ; interpolace
```


Osvětlování po pixelech

- Normály jsou definované v textuře (rgb->xyz)
- Tečný prostor je definován ve vrcholech
- Normály v textuře jsou v tečném prostoru $S, T, S \times T$



Stínovač pixelů pro osvětlování po pixelech

- Vstupy stínovače pixelů:
 - t0 – souřadnice v mapě normál
 - t1 – normální textura
 - v0 – vektor směru světla transformovaný do prostoru S,T,SxT

```
tex t0                ; vezmi normálu
tex t1
dp3_sat r0, v0_xb2, t0_bx2 ; N . L
add_sat r0, c0         ; ambient
mul r0, t0            ; moduluj texturou
```

Skeletální animace

- Každý vrchol je přivázán k nějaké kosti
- Ve skutečných aplikacích je přivázán k několika kostem s různými váhami
- Póza kostry je definována polem matic udávajících polohu a orientaci jednotlivých kostí

Stínovač vertexů pro skeletální animaci

- Vrcholy musí obsahovat informaci o tom, k jaké kosti jsou přivázané:
 - v0 – pozice
 - v1 – normála
 - v2 – index matice

Program pro skeletální animaci

```
mov r0.x ; v2.x           ; index matice
mul a0.x , r0, c20        ; c20 = počet matic *3

dp4 r0.x, v0, c[a0.x+30]  ; transformace pozice
dp4 r0.y, v0, c[a0.x+31] ; matice uloženy na c30 a dále
dp4 r0.z, v0, c[a0.x+32]

dp4 r1.x, v1, c[a0.x+30]  ; transformace normály
dp4 r1.y, v1, c[a0.x+31]
dp4 r1.z, v1, c[a0.x+32]

; v r0 a r1 nám zbyde pozice normála,
;tu transformujeme a osvětlujeme obvyklým způsobem
```

Spolupráce s osvětlováním po pixelech

- Vrcholy musí obsahovat vektory S,T:
 - v0 – pozice
 - v1 – normála
 - v2 – index matice
 - v3 – vektor S
 - v4 – vektor T
- $S \times T$ se dá kdykoliv spočítat

Spolupráce s osvětlováním po pixelech

■ Transformace tečného prostoru

dp4 r2.x, v3, c[a0.x+30] ; transformace S

dp4 r2.y, v3, c[a0.x+31]

dp4 r2.z, v3, c[a0.x+32]

dp4 r3.x, v4, c[a0.x+30] ; transformace T

dp4 r3.y, v4, c[a0.x+31]

dp4 r3.z, v4, c[a0.x+32]

mul r4, r2.yzxw, r3.zxyw ; výpočet SxT

mad r4, -r2.zxyw, r3.yzxw

Více informací

DirectX 8 SDK, ke stažení na:

msdn.microsoft.com/directx

Vývojářské stránky výrobců grafických karet:

nVidia:

<http://www.nvidia.com/>

ATI:

<http://www.ati.com/>

Dotazy

