

Školitel: Ing. Tomáš Oberhuber

Vladimír Klement, Tvorba softwaru, FJFI ČVUT

IMPLEMENTACE ALGORITMŮ PRO ZPRACOVÁNÍ OBRAZU S VYUŽITÍM GPGPU

OBSAH

- 1- Grafické karty
- 2- Grafická pipeline
- 3- Numerické výpočty na grafické kartě
- 4- Zpracování obrazu
- 5- Level Set Metoda
- 6- Segmentace
- 7- Numerické schéma
- 8- Výsledky
- 9- Závěr a poděkování

GRAFICKÉ KARTY



- × Speciální typ hardwaru
- × Řádově vyšší rychlost některých datově paralelních úloh
- × Obecně se však dá využít i k jiným druhům výpočtů
- × Rozdílná architektura

POPIS GRAFICKÉ PIPELINE

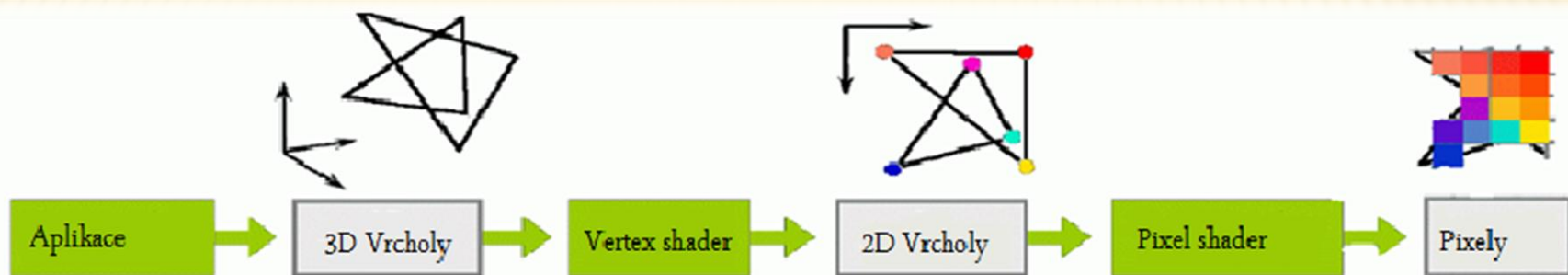


Schéma činnosti

- ✘ Transformace vrcholů (vertex shader)
- ✘ Rasterizace
- ✘ Vypočty barev jednotlivých pixelů (pixel shader)

Slovníček pojmů

Vertex – Bod v 3D prostoru

Pixel – Bod na obrazovce

Shader – Program pro grafickou kartu

NUMERICKÉ VÝPOČTY NA GRAFICKÉ KARTĚ

- ✘ Renderování může být libovolný výpočet
- ✘ Textury jsou pole
- ✘ Výsledek můžeme uložit do paměti

- ✘ OpenGL

Data = Textura 1 -> Shader -> Textura 2 = Výsledek



SHRNUÍ KLADŮ A ZÁPORŮ GRAFICKÝCH KARET

Výhody

- × Víc výpočetních jader
- × Optimalizace pro určitý druh úloh
- × Řádově vyšší rychlost

Nevýhody

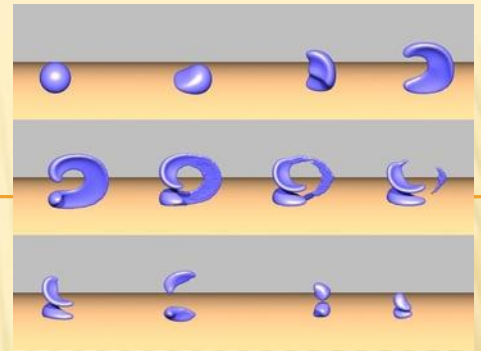
- × Nemožnost operace scatter
- × Problematické větvení
- × Nativní uložení dat je 2Dmatice

ZPRACOVÁNÍ OBRAZU

- ✘ Snímání
- ✘ Digitalizace
- ✘ Předzpracování
- ✘ Segmentace
- ✘ Porozumění obsahu



LEVEL SET METODA



- ✘ Vymyšlena Osherem a Sethianem
- ✘ Mnoho využití v dynamice tekutin
- ✘ Metoda vývoje křivky pomocí implicitní funkce
- ✘ Odvození:

$$\phi(x(t), y(t), t) = 0 \quad t \in \langle 0, T \rangle$$

$$\frac{d\phi}{dt} = \frac{\partial\phi}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial\phi}{\partial y} \frac{\partial y}{\partial t} + \frac{\partial\phi}{\partial t} = \nabla\phi \cdot \vec{\dot{x}} + \frac{\partial\phi}{\partial t} = 0$$

$$\phi_t + V|\nabla\phi| = 0 \quad V = \frac{\nabla\phi}{|\nabla\phi|} \cdot \vec{\dot{x}}$$

SEGMENTACE

- ✘ Co je segmentace?
- ✘ Proces zpracování obrazu
- ✘ Modifikace level set metody



$$V = g_0 \kappa + \nabla g_0 \vec{n} \quad g_0 = \frac{1}{1 + |\nabla I|^2}$$

$$\phi_t = g_0 |\nabla \phi| \cdot \nabla \left(\frac{\nabla \phi}{|\nabla \phi|} \right) + \nabla g_0 \cdot \nabla \phi$$

- ✘ Klady a zápory použitého přístupu?

NUMERICKÉ SCHÉMA

- ✘ Metoda sítí
- ✘ Diference

$$\frac{\partial u}{\partial x} \cong \frac{u(a+t) - u(a)}{t}$$

- ✘ Numerický vzorec level set metody

$$\frac{du_{ij}}{dt} = \sqrt{\varepsilon^2 + \frac{1}{2} (|\nabla^+ u_{ij}|^2 + |\nabla^- u_{ij}|^2)} * \nabla^+ \left(\frac{1}{1 + |\Delta(\nabla^+ I_{ij})|^2} \frac{\nabla^- u_{ij}}{\sqrt{\varepsilon^2 + |\nabla^- u_{ij}|^2}} \right)$$

MERSENNOVA METODA

- ✘ Metoda 4. řádu pro typ rovnic $\frac{du_{i,j}}{dt} = f(t, u_{i,j})$
- ✘ Schopnost aktualizace kroku

Výpočet koeficientů

$$k_{i,j}^1 = \tau \cdot f(t, u_{i,j})$$

$$k_{i,j}^2 = \tau \cdot f\left(t + \frac{1}{3}\tau, u_{i,j} + \frac{1}{3}k_{i,j}^1\right)$$

$$k_{i,j}^3 = \tau \cdot f\left(t + \frac{1}{3}\tau, u_{i,j} + \frac{1}{6}k_{i,j}^1 + \frac{1}{6}k_{i,j}^2\right)$$

$$k_{i,j}^4 = \tau \cdot f\left(t + \frac{1}{2}\tau, u_{i,j} + \frac{1}{8}k_{i,j}^1 + \frac{3}{8}k_{i,j}^3\right)$$

$$k_{i,j}^5 = \tau \cdot f\left(t + \tau, u_{i,j} + \frac{1}{2}k_{i,j}^1 - \frac{3}{2}k_{i,j}^3 + 2k_{i,j}^4\right)$$

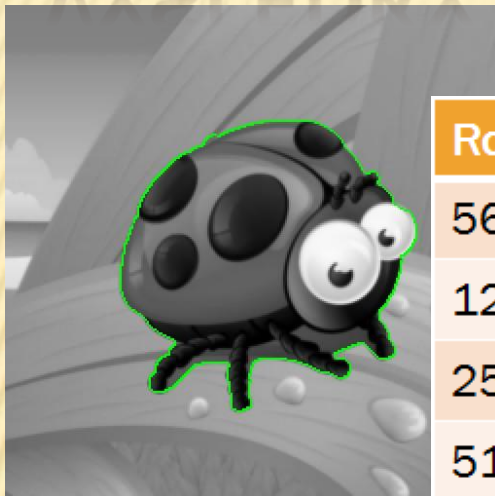
Aktualizace řešení

$$u_{i,j}(t + \tau) = u_{i,j}(t) + \frac{1}{6}k_{i,j}^1 + \frac{2}{3}k_{i,j}^4 + \frac{1}{6}k_{i,j}^5$$

Aktualizace časového kroku

$$\tau = \omega \tau \left(\frac{\varepsilon_M}{E} \right)^2$$

VÝSLEDKY MĚŘENÍ



Rozlišení	GPU	CPU	CPU/GPU	Zátěž CPU
56x56	9,762 s	7,281 s	0,74	71%
128x128	9,859 s	39,465 s	4,02	67%
256x256	10,493 s	169,523 s	16,06	60%
512x512	23,174 s	823,310 s	35,52	45%
1024x1024	70,360 s	3299,620 s	46,89	22%



KONEC

Děkuji za pozornost.