ČVUT FEL
Katedra počítačů

Matematické a fyzikální výpočty na grafických kartách (DirectX 9 + DirectX 10)

Bc. Jindřich Gottwald

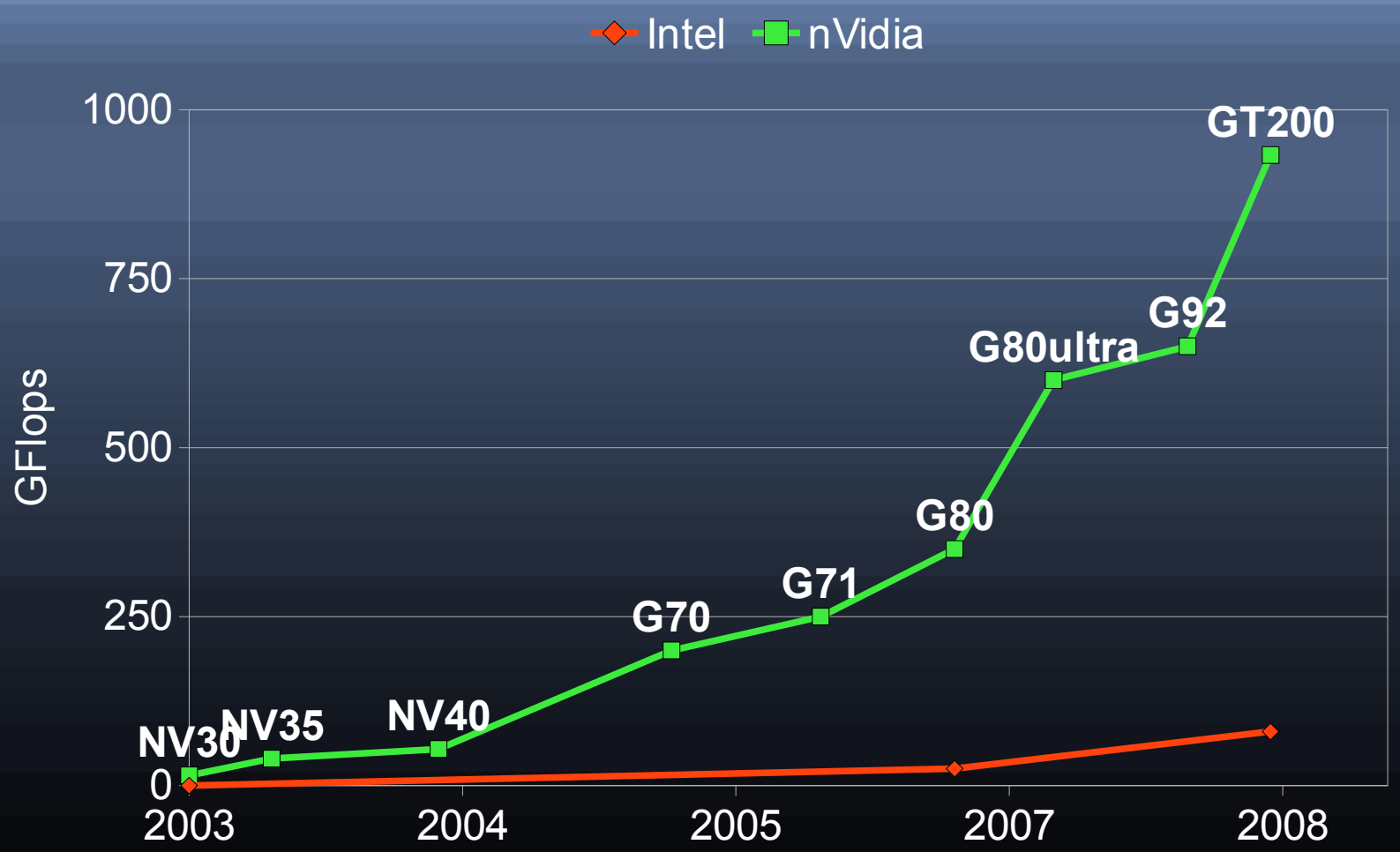
vedoucí: Ing. Ivan Šimeček, Ph.D.

2009

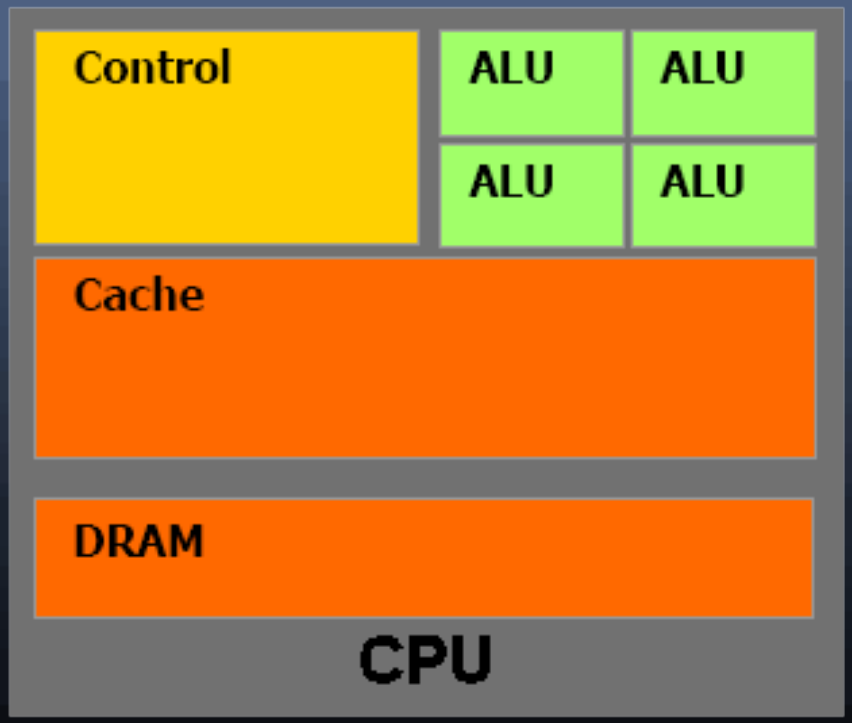
Co je GPGPU?

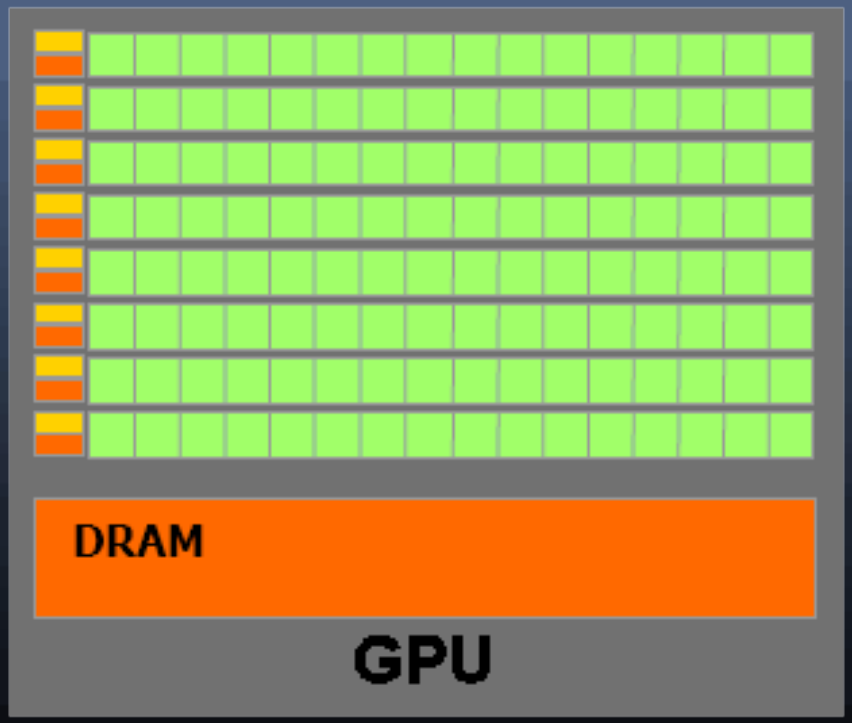
- **G**eneral **P**urpose computing on **G**raphics **P**rocessing **U**nit
- obecné výpočty na grafických kartách
- využití shader jednotek grafické karty k negrafickým účelům
- http://www.gpgpu.org (od roku 2002)

Proč GPGPU?


Intel
nVidia
GFlops
1000
750
500
250
0
2003
2004
2005
2007
2008
NV30
NV35
NV40
G70
G71
G80
G80ultra
G92
GT200

HW rozdíl CPU a GPU





Rozdílný přístup CPU a GPU

CPU:

- optimalizován pro sekvenční zpracování

  (cache, predikce skoků)
- univerzální

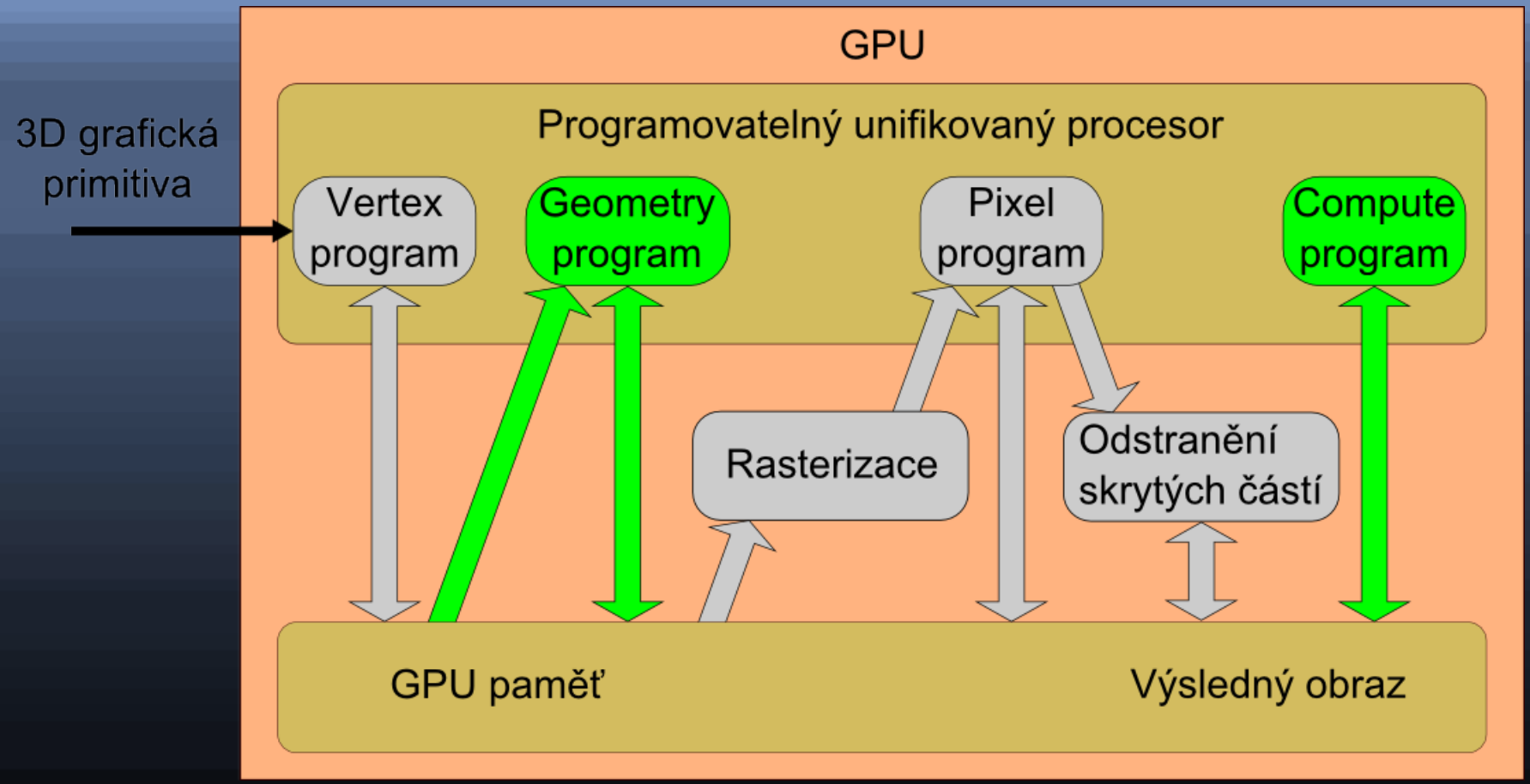
GPU:

- vysoce paralelní (MSIMD)
- "jednoúčelové"
- nemá bitové operace
- nemá integer
- omezená přesnost
- omezené sdílení dat

Jak na GPGPU?

- DirectX9, DirectX10, HLSL
- OpenGL, GLSL
- CUDA (Compute Unified Device Architecture)
- Brook, Brook+
- Cg (C for Graphics)
- Ct (C for Throughput)
- OpenCL
- DirectX11 Compute Shader, HLSL

Moderní grafická pipeline



Omezení GPU

- grafická data = textury, polygony
- jednotky zpracovávají paralelně jeden kód
- jednotky pracují nezávisle, nemohou mezi sebou sdílet data
- není možné mít jedny data zároveň pro čtení i zápis

Postup GPGPU

- textura = paměťový prostor
    - souřadnice = adresy
- obdélník o rozměrech textury poslat do vertex shaderu
- pixel shader provede kód nad každým pixelem
    - zpracovat texely textury
    - vypnout AA
- výstup pixel shaderu renderovat do jiné textury
- prohodit vstupní a výstupní textury (pingpong)

CPU implementace GE

```
for ( int t=0; t<(height-1); t++ ) // posun pivotu
{
    pivot = inMat[t][t];

    for ( int i=(t+1); i<height; i++ )  // výpočet sloupce
        tmpMat[i] = inMat[t][i] / pivot;

    for ( int i=(t+1); i<height; i++ )
        for ( int j=t; j<width; j++ ) // výpočet submatice
            inMat[j][i] = inMat[j][i] – tmpMat[i] * inMat[j][t];
}
```

GPU implementace GE

```
PS_OUTPUT psRow (VS_OUTPUT In) : COLOR
{
    PS_OUTPUT Out;
    float pivot, inMat;
    float pivotX, pivotY;

    pivotX = pivotRaw.a / pivotRaw.r; // pivot / šířka textury
    pivotY = pivotRaw.a / pivotRaw.g; // pivot / výška textury

    inMat = tex2D( mySamp, float2( pivotX, In.Tex0.y ) ).r;
    pivot = tex2D( mySamp, float2( pivotX, pivotY ) ).r;

    Out.Color.r = inMat / pivot; // výpočet sloupce

    return Out;
}
```

DirectX – na co si dát pozor?

DX9:

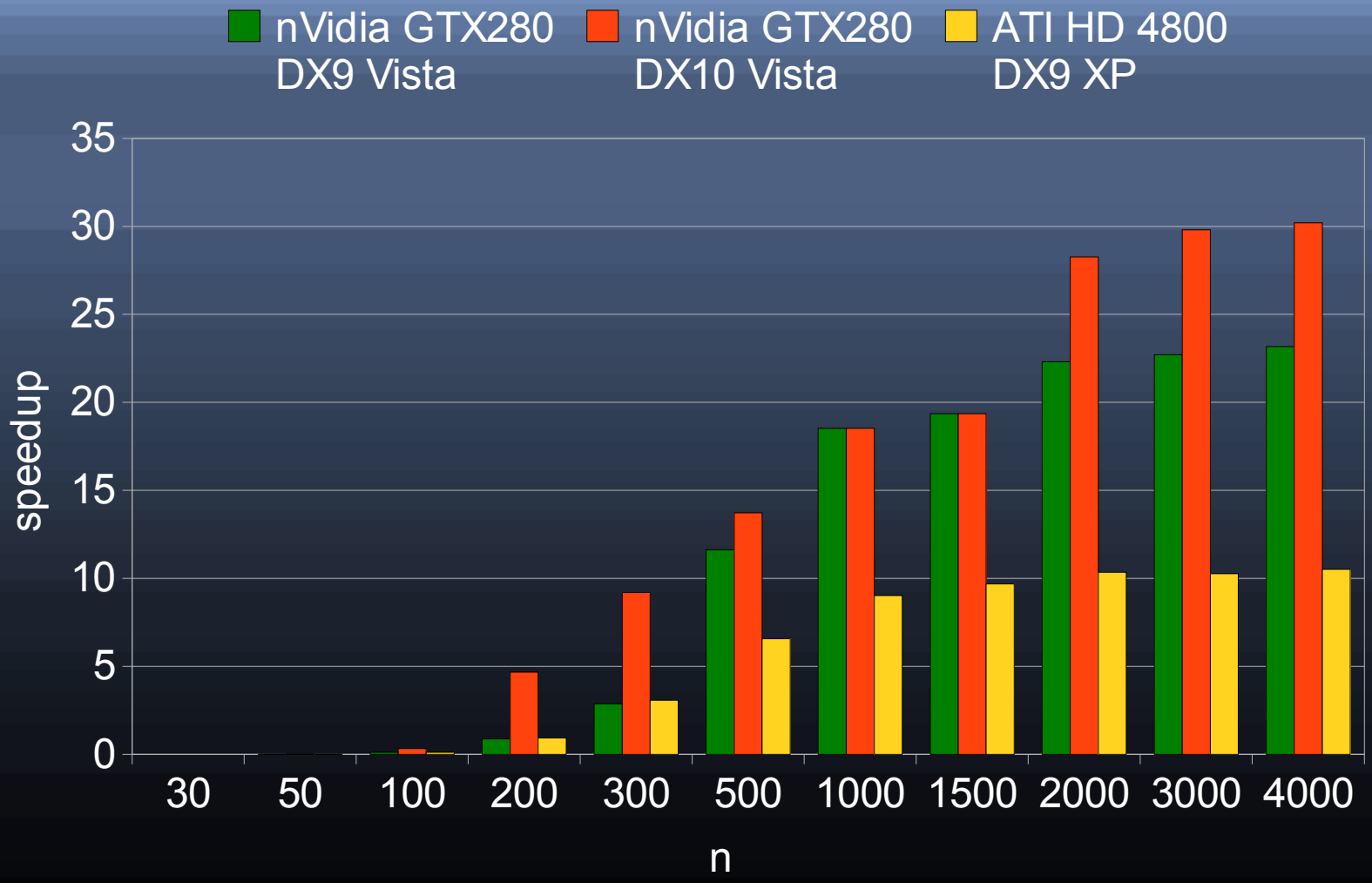
- posunutí pixelů vůči texelům o půl bodu
- normalizované souřadnice
- minimální velikost okna 32x32

DX10:

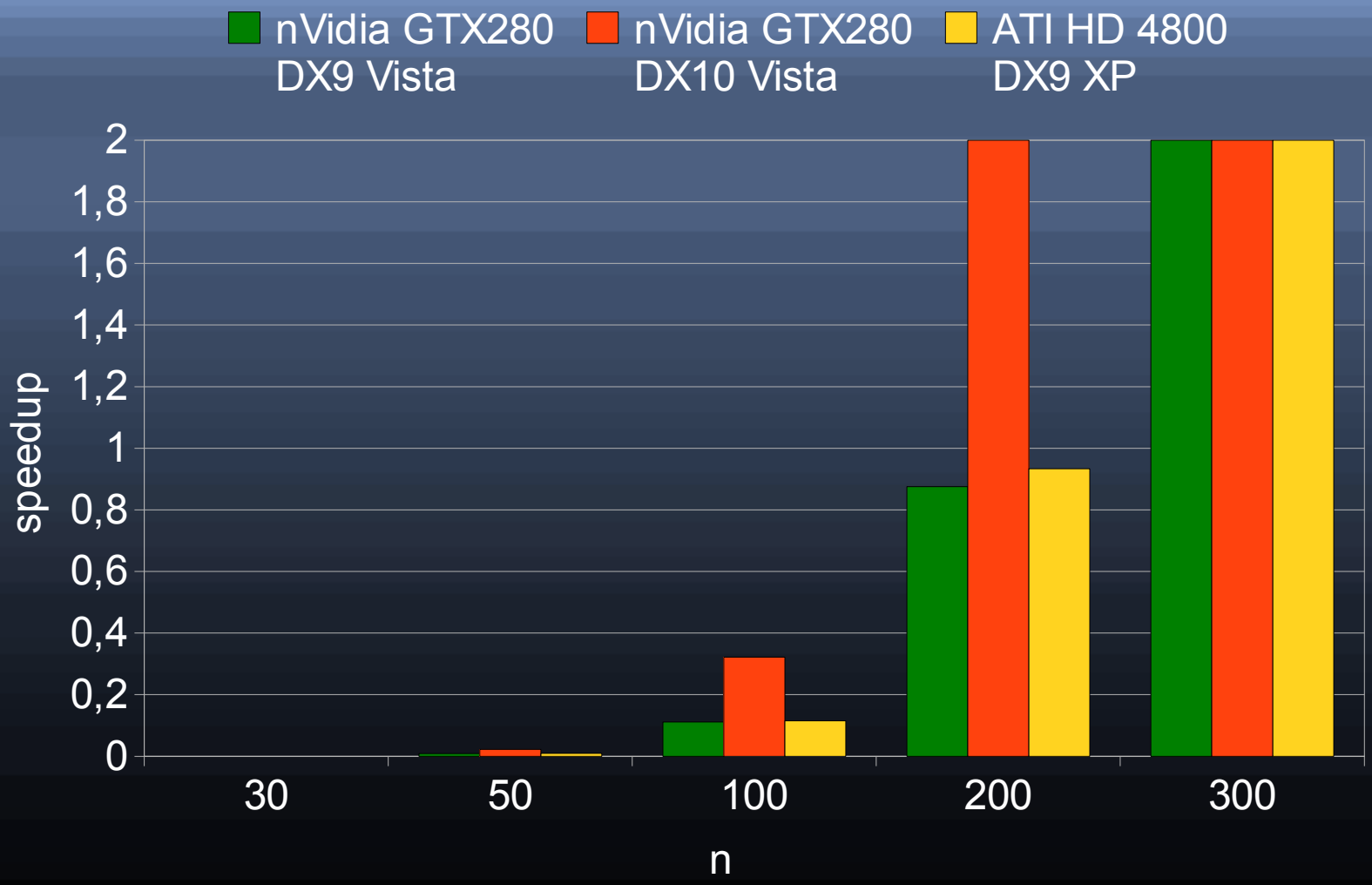
- Y souřadnice textury opačně proti viewportu
- minimální velikost okna 64x64

- uložení textury ve správné paměti

Zrychlení GE na GPU



Zrychlení GE na GPU detail



Přesnost GPU

- i když se chlubí IEEE 754, není to ani FP32
- starší karty dokonce ještě nižší přesnost

	3 / 6,5	0.0030 / 0.0065
přesná hodnota	0.46153846153846153 8	0.46153846153846153 8
FP64 CPU	0.4615384615384615 6	0.4615384615384615 6
FP32 CPU	0.4615384 6383094788	0.4615384 6383094788
FP32 nVidia GTX280; 9600M	0.4615384 9363327026	0.4615384 3402862549
FP32 nVidia Go 7600; FX 1500	0.4615384 9363327026	0.4615384 6383094788
FP32 ATI HD 2600; HD 4870	0.4615384 3402862549	0.4615384 3402862549
FP24 ATI x700	0.4615325 9277343750	0.4615325 9277343750

Závěr

- rychlost
- nepřesnost
- nepříjemnosti DX9 a DX10
- http://goldsoft.cz/gpgpugs2
- budoucnost ?

Děkuji za pozornost

Dotazy?

Výtky oponenta

- clipping/culling v kap. 2.3
  - frustrum clipping,hierarchical Z buffer, tile based
- pouze první fáze GE
- časy načtení/uložení dat, kdy se vyplatí CPU
  - graf 7 = zlom při n=200 (závisí na HW) (kap 5.4)
- tab. 4 porovnání DX9/DX10 výkyv vysvětlení
  - optimálnějším přístupem do paměti (kap 5.4)
- relativní chyba výpočtu proti CPU
  - závisí na vstupních datech

Optimalizace

- pivoting
  - omezení vzniku nepřesností
- trojúhelník místo obdélníku
  - přednačítaní sousedních texelů

Výtky vedoucího

- pouze jeden algoritmus (GE)
- použitý kompilátor a nastavení
- vztah pro tab.8