

Análise Comparativa do Desempenho de Unidades de Processamento Gráfico Nvidia e ATI através de filtros digitais de imagens

Darlisson Marinho de Jesus¹
Raimundo Corrêa de Oliveira¹

¹Engenharia de Computação
Universidade do Estado do Amazonas - UEA

Julho - 2013

Sumário

- Introdução
 - Descrição do Problema
 - Justificativa
 - Objetivo Geral
 - Objetivos Específicos
- Metodologia
- Desenvolvimento
 - Arquitetura da GPU Moderna
 - A Linguagem OpenCL
 - Filtro Sobel
 - Filtro Passa-baixa
- Resultados
- Referências

Descrição do Problema

- Certos problemas, como dinâmica de fluidos, simulação de colisões, processamento sísmico, comparação de imagens e modelagem climáticas, exigem muito dos recursos de cálculo, pois possuem alto custo computacional.
- Essas áreas da computação de alto desempenho estão adotando modernas unidades de processamento gráfico para resolver problemas de cálculo em grande escala [3].

Justificativa

- Diante disso, como auxiliar engenheiros e cientistas a escolherem as GPUs que forneçam o melhor desempenho?
- É necessário avaliar quais GPUs fornecem o melhor desempenho para soluções de computação paralela, e assim, auxiliar no projeto das soluções para a computação de alta performance.

Objetivo Geral

- Comparar o desempenho das Unidades de Processamento Gráfico das fabricantes NVIDIA e ATI, através do Processamento Digital de Imagens com os filtros Passa-baixa e o filtro Sobel para detecção de borda implementados na linguagem OpenCL.

Objetivos Específicos

- Determinar os indicadores de desempenho que permitam avaliar as rotinas destes filtros no contexto das Unidades de Processamento Gráfico;
- Avaliar a arquitetura das GPUs da Nvidia e ATI, buscando identificar as diferenças que podem afetar no desempenho das implementações;
- Implementar o algoritmo do filtro Passa-Baixa na linguagem OpenCL e obter os dados de desempenho das GPUs Nvidia e ATI;
- Implementar o algoritmo do filtro para detecção de borda Sobel na linguagem OpenCL e obter dados de desempenho das GPU Nvidia e ATI;

Objetivos Específicos

- Determinar os indicadores de desempenho que permitam avaliar as rotinas destes filtros no contexto das Unidades de Processamento Gráfico;
- Avaliar a arquitetura das GPUs da Nvidia e ATI, buscando identificar as diferenças que podem afetar no desempenho das implementações;
- Implementar o algoritmo do filtro Passa-Baixa na linguagem OpenCL e obter os dados de desempenho das GPUs Nvidia e ATI;
- Implementar o algoritmo do filtro para detecção de borda Sobel na linguagem OpenCL e obter dados de desempenho das GPU Nvidia e ATI;

Objetivos Específicos

- Determinar os indicadores de desempenho que permitam avaliar as rotinas destes filtros no contexto das Unidades de Processamento Gráfico;
- Avaliar a arquitetura das GPUs da Nvidia e ATI, buscando identificar as diferenças que podem afetar no desempenho das implementações;
- Implementar o algoritmo do filtro Passa-Baixa na linguagem OpenCL e obter os dados de desempenho das GPUs Nvidia e ATI;
- Implementar o algoritmo do filtro para detecção de borda Sobel na linguagem OpenCL e obter dados de desempenho das GPU Nvidia e ATI;

Objetivos Específicos

- Determinar os indicadores de desempenho que permitam avaliar as rotinas destes filtros no contexto das Unidades de Processamento Gráfico;
- Avaliar a arquitetura das GPUs da Nvidia e ATI, buscando identificar as diferenças que podem afetar no desempenho das implementações;
- Implementar o algoritmo do filtro Passa-Baixa na linguagem OpenCL e obter os dados de desempenho das GPUs Nvidia e ATI;
- Implementar o algoritmo do filtro para detecção de borda Sobel na linguagem OpenCL e obter dados de desempenho das GPU Nvidia e ATI;

Métodos

- Revisão bibliográfica;
- Análise e experimentação da linguagem OpenCL;
- Desenvolvimento dos Filtros Digitais (Sobel e Passa-baixa);
- Coleta das imagens para serem processadas;
- Coleta e análise dos dados;

Métodos

- Revisão bibliográfica;
- Análise e experimentação da linguagem OpenCL;
- Desenvolvimento dos Filtros Digitais (Sobel e Passa-baixa);
- Coleta das imagens para serem processadas;
- Coleta e análise dos dados;

Métodos

- Revisão bibliográfica;
- Análise e experimentação da linguagem OpenCL;
- Desenvolvimento dos Filtros Digitais (Sobel e Passa-baixa);
- Coleta das imagens para serem processadas;
- Coleta e análise dos dados;

Métodos

- Revisão bibliográfica;
- Análise e experimentação da linguagem OpenCL;
- Desenvolvimento dos Filtros Digitais (Sobel e Passa-baixa);
- Coleta das imagens para serem processadas;
- Coleta e análise dos dados;

Métodos

- Revisão bibliográfica;
- Análise e experimentação da linguagem OpenCL;
- Desenvolvimento dos Filtros Digitais (Sobel e Passa-baixa);
- Coleta das imagens para serem processadas;
- Coleta e análise dos dados;

Métodos

- Revisão bibliográfica;
- Análise e experimentação da linguagem OpenCL;
- Desenvolvimento dos Filtros Digitais (Sobel e Passa-baixa);
- Coleta das imagens para serem processadas;
- Coleta e análise dos dados;

Métodos - medidas de desempenho

- **Tempo Médio de Execução do Kernel** - corresponde ao tempo, em milisegundos, em que o *kernel* permanece em execução na GPU
- **Taxa Média de Transferência Dados da Memória** - corresponde ao número de bytes por unidade de tempo transmitidos entre a memória do computador e a memória da placa gráfica de forma bi-direcional.

Materiais

Equipamentos:

- Três PC/x64 com Sistema Operacional Microsoft Windows 7 Professional 64 Bits, Processador *Intel® Core™ 2 Duo E7400* 2.80GHz e 4 GB de memória RAM.
- Três Unidades de processamento gráfico. A Tabela 1 apresenta as especificações de hardware detalhados dessas GPUs.

Materiais Cont.

Modelo	Geforce GT 520	Geforce 210	Radeon HD 6450
Processadores de Stream	48	16	160
Clock do processador	810 MHz	589 MHz	750 MHz
Arquitetura da GPU	Tesla	Fermi	Caicos
Memória	—	—	—
Clock da memória	900 MHz	533 MHz	1066 MHz
Tamanho da memória	1024 MB	512 MB	1024 MB
Interface da memória	64-bit	64 -bit	64-bit
Largura de Banda (GB/sec)	14.4	8.0	8.5
Tipo de memória	DDR3	DDR3	DDR3

Tabela : Resumo das especificações das GPUs Nvidia e ATI

Materiais - Desenvolvimento.

As seguintes ferramentas foram utilizadas para compilação dos programas:

- Microsoft Visual Studio 2010 versão 10.0.30319.1 RTMRel
- Microsoft .NET Framework versão 4.5.50709 RTMRel
- OpenCL-GPU: Pacote AMD APP SDK versão 2.8.1 e driver de vídeo ATI Catalyst versão 12.104 no Windows 7 32 bits.
- OpenCL-GPU: Pacote Nvidia Cuda Toolkit versão 5.0 e driver de vídeo versão 320.18 no windows 7 64 bits.

Materiais - Amostras.

- Usamos 32 imagens no formato *PGM* (do inglês, *Portable Gray Map*), divididas em 8 amostras para cada uma das seguintes dimensões (em pixels): 256x256, 512x512, 1024x1024 e 2048x2048.
- Estas imagens foram obtidas no banco de dados de imagens proposto em [2].

Materiais - Análise e Coleta dos Dados

As seguintes ferramentas foram utilizadas para coleta das métricas de desempenho dos programas e análise:

- Nvidia Nsight Visual Studio Edition versão 3.0
- AMD CodeXL versão 1.2
- Software R versão 3.0.1

Materiais - Coleta e Análise dos Dados

- CodeXL 1.2 da AMD

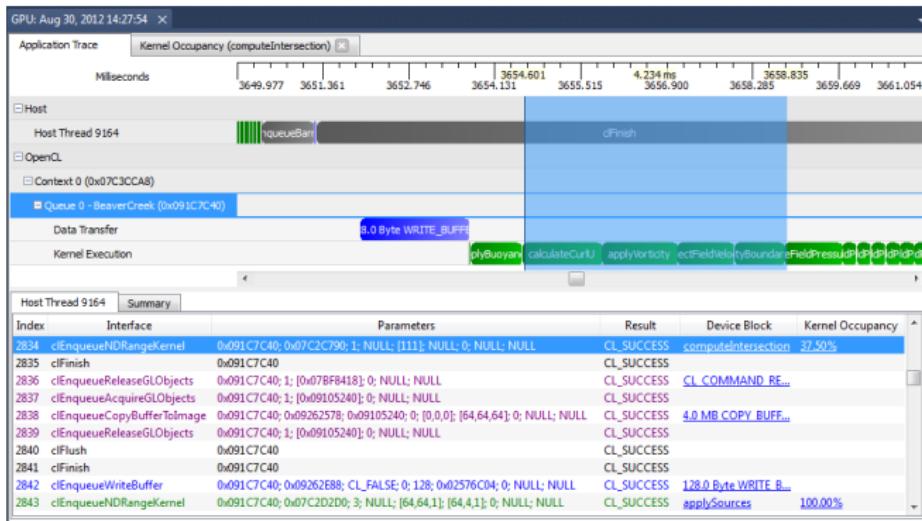


Figura : AMD CodeXL - Permite o profiler das aplicações em escritas em OpenCL nas GPUs ATI

Materiais - Coleta e Análise dos Dados

- Nvidia Nsight 3.0

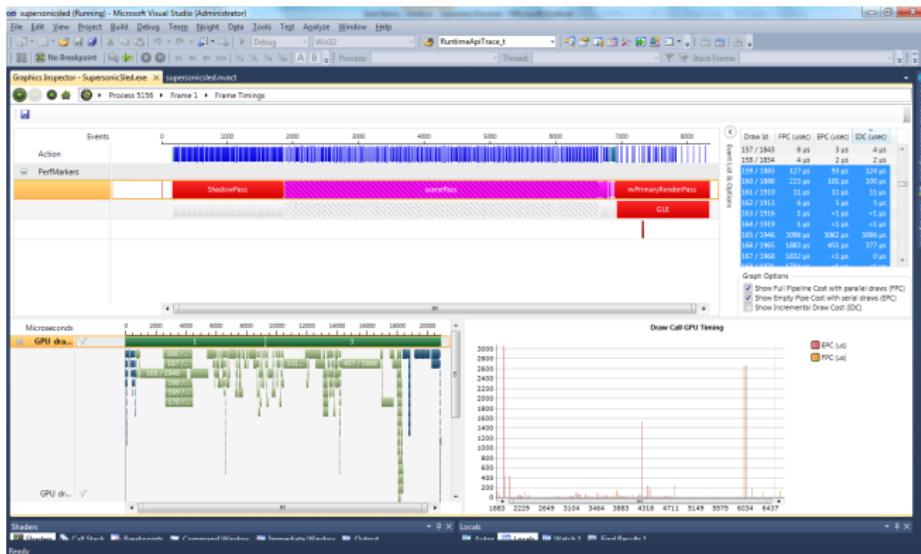


Figura : Nvidia Nsight - Permite o profiler das aplicações em escritas em OpenCL nas GPUs Nvidia

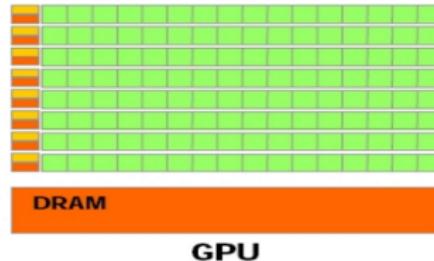
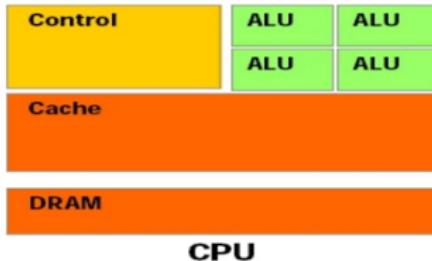
Histórico das GPUs

- **1981 MDA** Monochrome Display Adapter - IBM
 - permitiu o PC exibir 80 colunas e 25 linhas no monitor.
- **1988 VGA Wonder** - ATI
 - capaz de produzir imagens em 16-bits de cor.
- **1996 Voodoo 1** - 3DFX
 - primeira a fornecer uma interface de programação.
- **1997 Riva 128** - Nvidia
 - primeira placa com suporte a Direct 3D.
- **1998 Voodoo 2** - 3DFX
 - suporte a processamento paralelo com SLI (Scan-Line Interleave).
- **1999 GeForce 256 DDR** - Nvidia,
 - primeira placa 3D com suporte total a DirectX 7, primeira GPU de fato.
- **2001 GeForce 3** - Nvidia
 - primeira GPU totalmente programável.

Histórico das GPUs

- **2006 GeForce 8800** - Nvidia
 - *o CUDA é lançado e a era da GPGPU começa.*
- **2009 Radeon HD 5970** - ATI
 - *implementou a tecnologia CrossFireX com suporte para até 4 GPU.*
- **2013 GeForce GTX 690** - Nvidia
 - *placa gráfica mais rápida já criada, 2 GPUs kepler e 3.072 cuda cores.*

CPU X GPU



- Própria para tarefas sequenciais
- Cache eficiente
- Maior quantidade de memória principal
- Número de cores de 1 ordem de grandeza
- 1, 2 threads por core

- Própria para tarefas com paralelismo de dados
- Maior (capacidade) operações de ponto flutuante por segundo
- Alto throughput de memória
- Dezenas de multiprocessors
- Múltiplas threads por core

Visão geral da arquitetura Nvidia Tesla

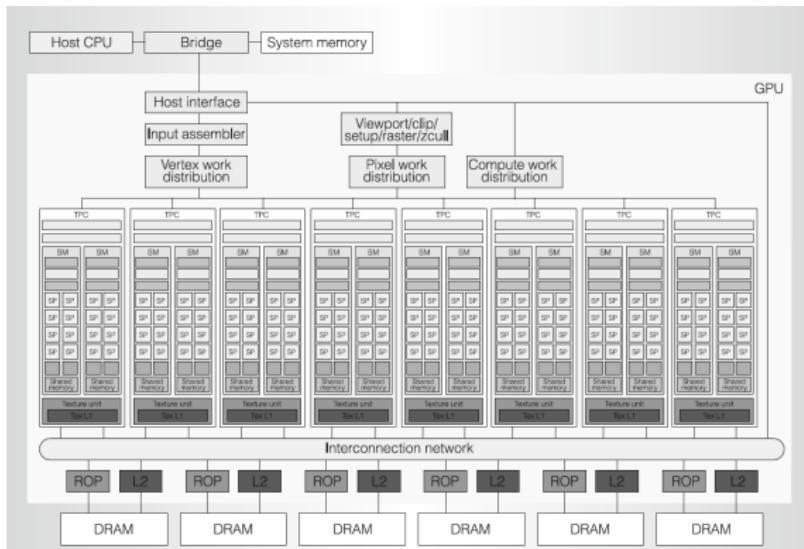


Figura : Visão Geral da arquitetura Nvidia Tesla.Fonte:([1])

Detalhes da arquitetura Nvidia Tesla

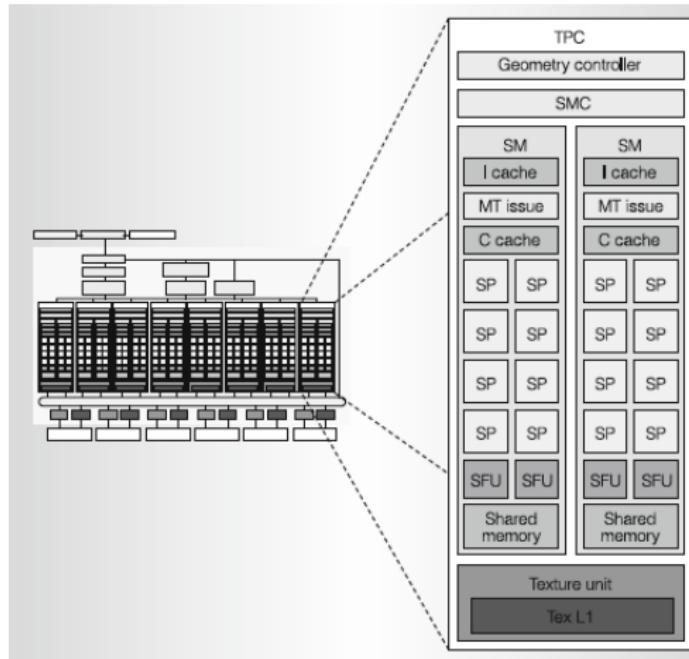


Figura : Detalhes da arquitetura Nvidia Tesla.Fonte:([1])

Visão geral da arquitetura Nvidia Fermi

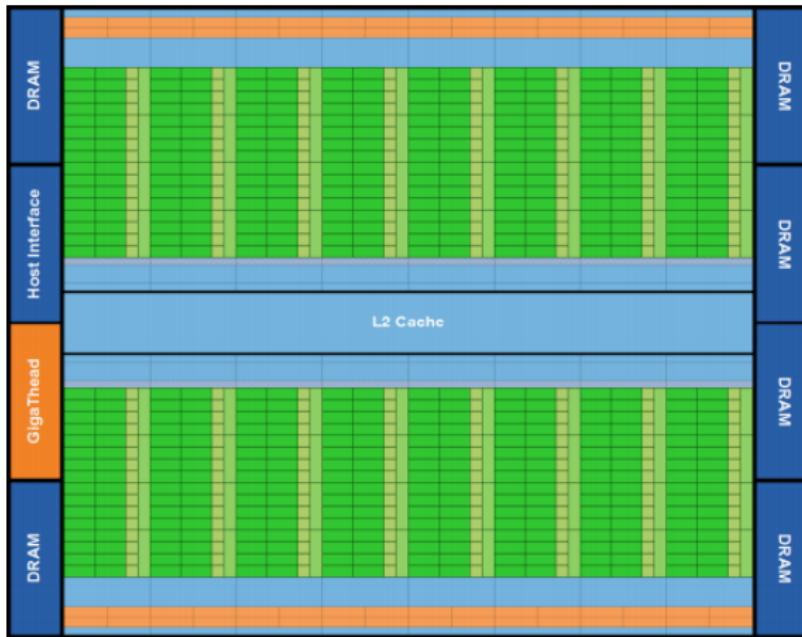


Figura : Visão geral da arquitetura Nvidia Fermi

Detalhes da arquitetura Nvidia Fermi



Figura : Detalhes da arquitetura Nvidia Fermi

Diferenças Fermi x Tesla

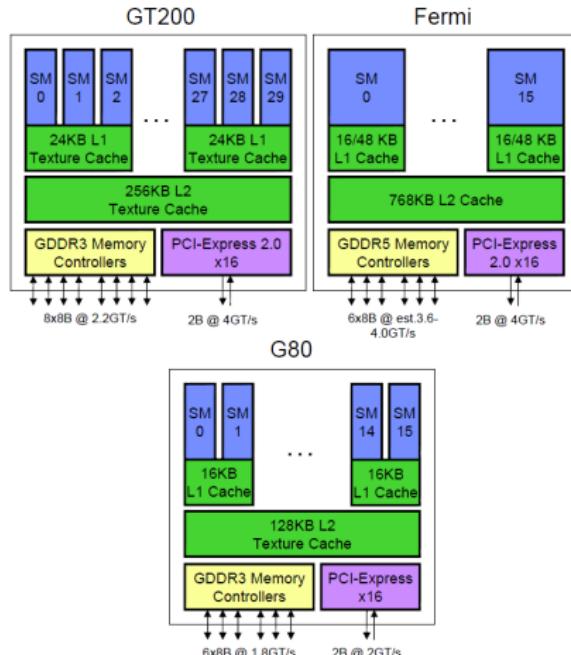


Figura : Diferenças entre a Fermi e a Tesla

Visão geral da arquitetura ATI Caicos

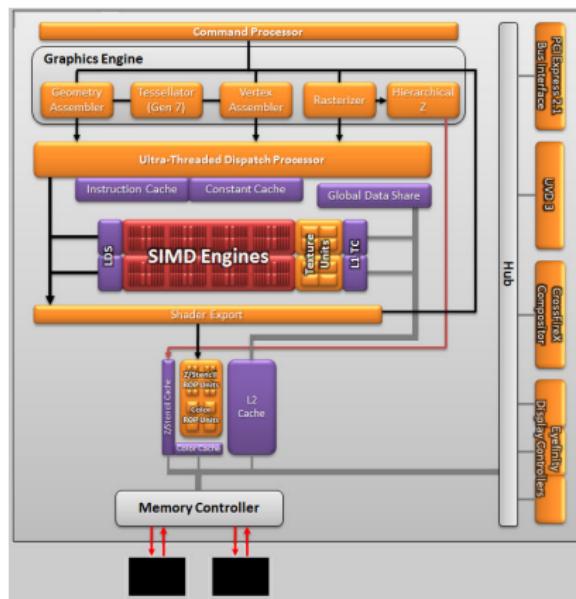
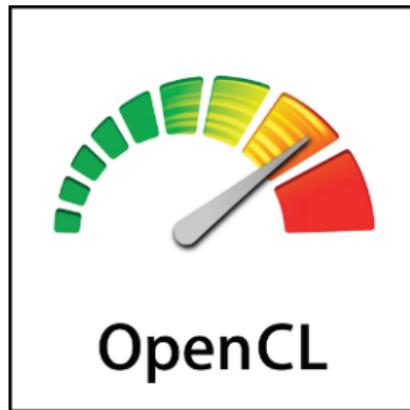


Figura : Visão geral da arquitetura ATI Caicos

GPGPU

- **General-purpose computing on Graphics Processing Units**
 - Técnica de uso de GPU para computação de propósito geral
- Linguagens/API's
 - Brook
 - Brook+
 - OpenCL
 - CUDA

OpenCL - Open Computing Language



**"Padrão aberto para a
programação paralela de sistemas
heterogêneos"**

OpenCL - Open Computing Language

Principais características:

- Provê interface homogênea para a exploração da computação paralela heterogênea
 - Abstração do hardware
 - CPU's (AMD, ARM, IBM, Intel), GPU's (AMD, ARM, Intel, Nvidia), APU's, CBE, DSP's, FPGA's
- Padrão aberto
 - Especificação mantida por vários membros gerenciada pelo grupo Khronos
- Alto desempenho
 - Possui diretivas de baixo nível para uso eficiente dos dispositivos
 - Alto grau de flexibilidade

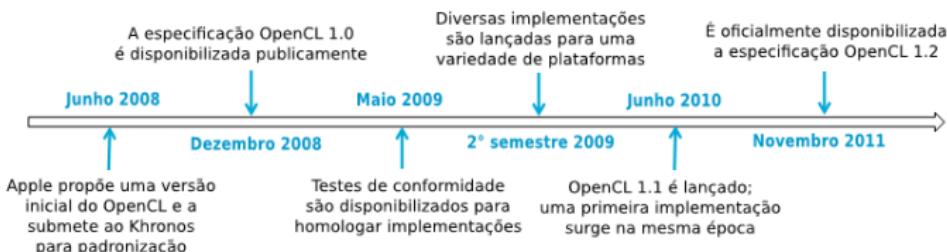
OpenCL - Características

- Multi-plataforma
 - Disponível em várias classes de hardware e sistemas operacionais
- Código portável entre arquiteturas e gerações
- Especificação baseada nas linguagens C e C++

OpenCL - História

História

- **2003:** GPUs começam a adquirir características de propósito geral: a era da programabilidade
- **2003-2008:** Cenário GP-GPU fragmentado, com várias soluções proprietárias
- **2008:** Apple elabora o rascunho inicial da especificação do OpenCL



- **2013:** OpenCL 2.0 é lançado

OpenCL - Contribuidores



Figura : Contribuidores para o OpenCL em 2013

Filtro Sobel

O filtro Sobel calcula o gradiente da intensidade da imagem em cada ponto, dando a direcção da maior variação de claro para escuro e a quantidade de variação nessa direcção, através de duas matrizes 3x3, que são convoluídas com a imagem original para calcular aproximações das derivadas - uma para as variações horizontais G_x e uma para as verticais G_y .

Máscara de Sobel 3x3

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

A magnitude do gradiente é dado por:

$$|G| = \sqrt{G_x^2 + G_y^2}$$

Código Fonte em OpenCL

Ver item A.1 do Apêndice A

Filtro Passa Baixa no domínio da Frequência

Filtro passa-baixas ideal - Um filtro passa-baixas 2-D ideal é aquele cuja função de transferência satisfaz a relação:

$$H(u, v) = \begin{cases} 1, & \text{se } D(u, v) \leq Do, \\ 0, & \text{se } D(u, v) > Do. \end{cases}$$

onde Do é um valor não-negativo (corresponde à frequência de corte de um filtro 1-D), e $D(u, v)$ é a distância do ponto (u, v) à origem do plano de frequência; isto é,

$$D(u, v) = \sqrt{(u - P/2)^2 + (v - Q/2)^2}$$

onde, P e Q são, respectivamente, a largura e a altura da imagem.

Quanto menor o raio Do , menor a frequência de corte e, portanto, maior o grau de borramento da imagem resultante.

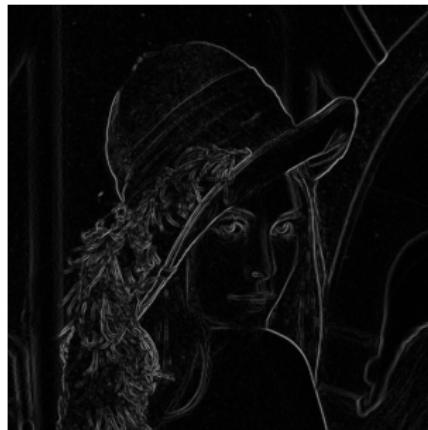
Código Fonte em OpenCL

Ver item A.2 do Apêndice A

Filtro Sobel



(a) Imagem original



(b) Imagem após a aplicação do filtro Sobel

Figura : Resultado da aplicação filtro Sobel para detecção de borda

Filtro Sobel - Tempo médio de execução

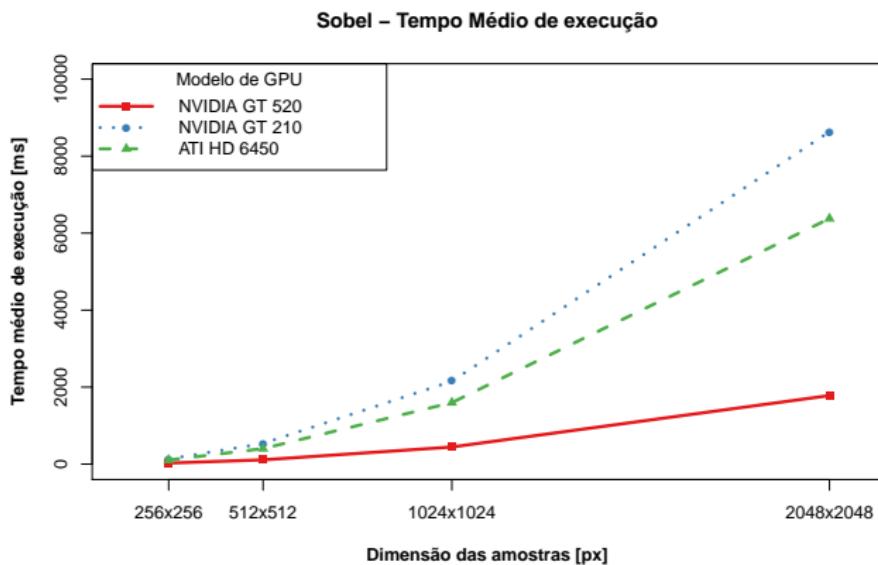


Figura : Sobel - Tempo médio de execução

Filtro Sobel - Taxa média de transferência

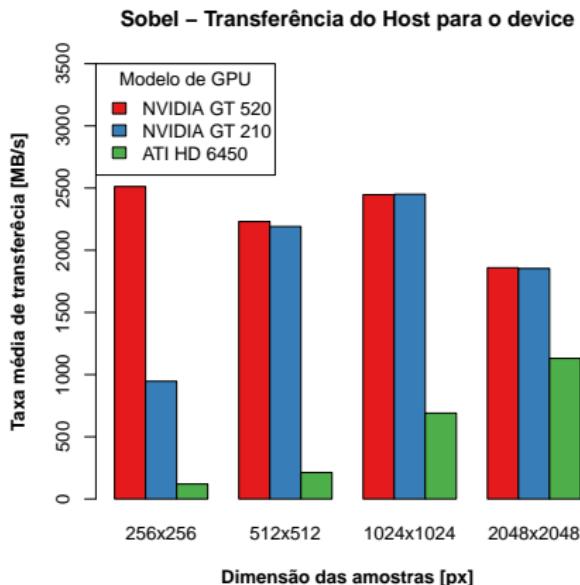


Figura : Taxa média de transferência do Host para o device

Filtro Sobel - Taxa média de transferência

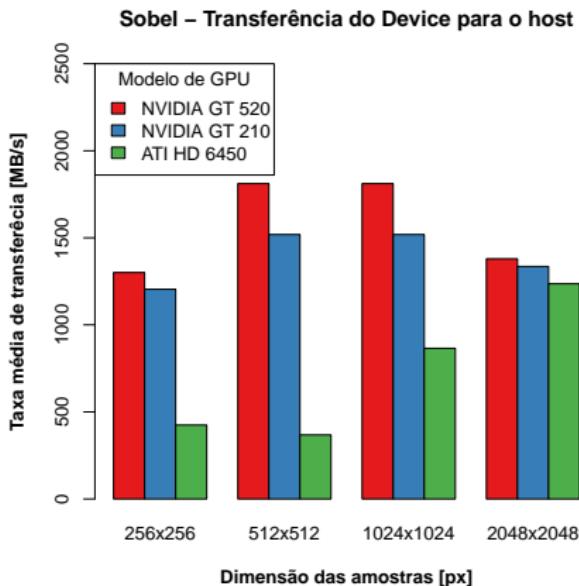


Figura : Taxa média de transferência do Device para o host

Filtro Passa-baixa



(a) Imagem original



(b) Imagem após a aplicação do filtro Passa-baixa

Figura : Resultado da aplicação do filtro Passa-baixa

Filtro Passa-baixa- Tempo médio de execução

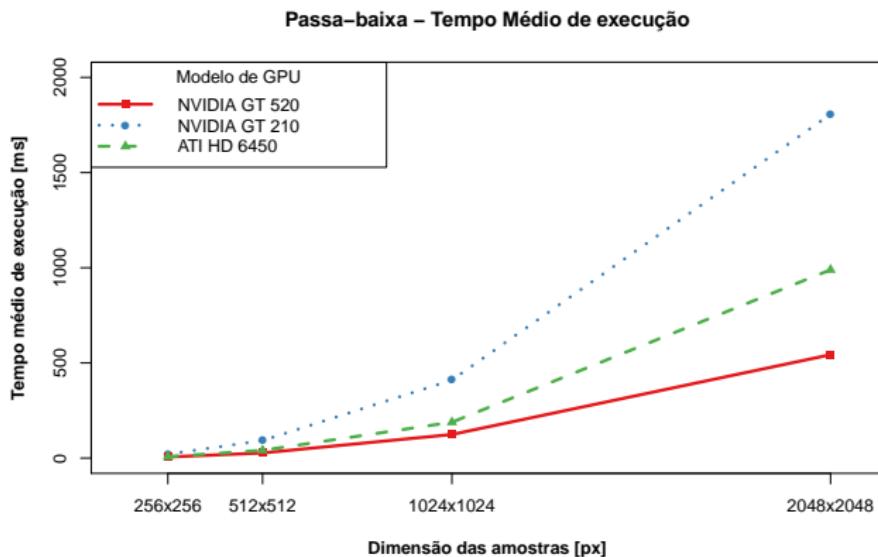


Figura : Passa-Baixa - Tempo médio de execução

Filtro Passa-baixa - Taxa média de transferência

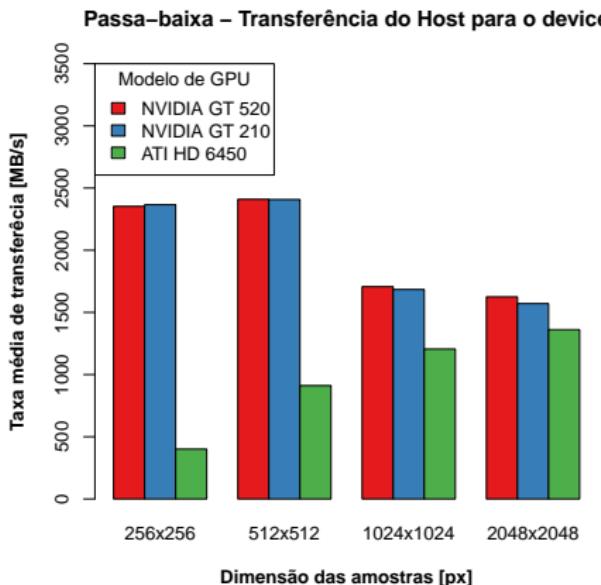


Figura : Taxa média de transferência do Host para o device

Filtro Passa-baixa - Taxa média de transferência

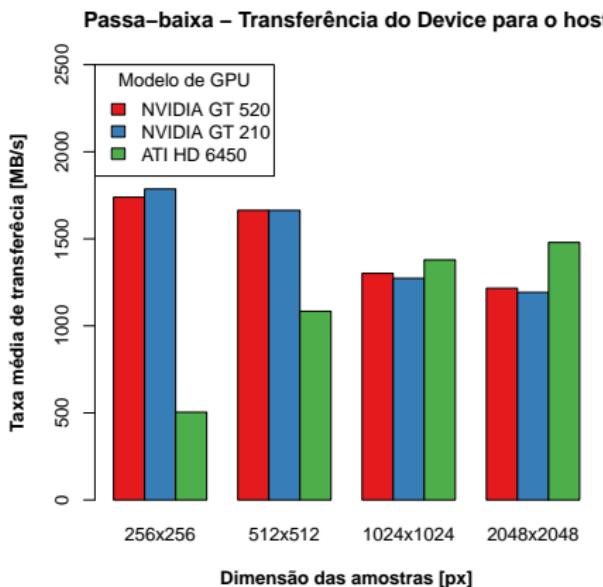


Figura : Taxa média de transferência do Device para o host

Referências

-  E. Lindholm, J. Nickolls, S. Oberman, and J. Montrym.
Nvidia tesla: A unified graphics and computing architecture.
Micro, IEEE, 28(2):39–55, 2008.
-  D. Martin, C. Fowlkes, D. Tal, and J. Malik.
A database of human segmented natural images and its application
to evaluating segmentation algorithms and measuring ecological
statistics.
In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423,
July 2001.
-  Ying Zhang, Lu Peng, Bin Li, Jih-Kwon Peir, and Jianmin Chen.
Architecture comparisons between nvidia and ati gpus: Computation
parallelism and data communications.
2012 IEEE International Symposium on Workload Characterization (IISWC), 0:205–215, 2011.