

Análise comparativa do desempenho de filtros digitais de imagens com o OpenCL na GPU Nvidia e ATI

Darlisson Marinho de Jesus¹, Raimundo Correa Oliveira

¹ Universidade do Estado do Amazonas – UEA

Av. Darcy Vargas, 1200, Parque 10 de Novembro – 69.065.020 – Manaus – AM – Brasil

{darlisson11, rcorrea.oliveira}@gmail.com

Abstract. *Graphics Processing Unit (GPU) are high performance co-processors intended, originally, to improve the use and quality of computer graphics applications. However, the GPU applications has been extended to other fields to general purpos, such as digital image processing. The main objective of this study is to compare the performance of digital filters to images in the context of GPUs from Nvidia and ATI vendors. To achieve this, we will implement the Fast Fourier Transform algorithm and the Canny edge detection filter, with OpenCL (Open Computing Language) framework, which allows the writing of code for running cross-platform on these GPUs.*

Resumo. *Unidades de processamento gráfico ou GPU (do inglês, Graphics Processing Unit) são co-processadores de alto desempenho destinados inicialmente a melhorar ou prover de capacidade gráfica um computador. Porém, as aplicações nas GPUs tem sido expandida à outras áreas para fins gerais, como o processamento digital de imagens. O principal objetivo deste trabalho é comparar o desempenho de filtros digitais de imagens no contexto das GPUs das fabricantes Nvidia e ATI. Para conseguir isso, implementaremos o algoritmo da Transformada Rápida de Fourier e o filtro de detecção de borda Canny, com o framework OpenCL (do inglês, Open Computing Language), a qual, permite a escrita de código multi-plataforma para execução nestas GPUs.*

1. Descrição do Problema

Alguns dos grandes desafios da área de processamento de imagens são a disponibilidade e o custo financeiro do acesso a um dispositivo com poder computacional suficiente para atingir grandes velocidades na execução de sistemas ligados à visão computacional e à interação homem-computador. Imagens e vídeos digitais são em essência sinais multidimensionais e, portanto, possuem um grande volume de dados [1]. O processamento paralelo pode ajudar a diminuir o tempo para o processamento deste grande volume de dados através da divisão do problema em problemas menores, os quais podem ser executados por múltiplos núcleos de processamento.

Como resultado da recente evolução dos processadores gráficos, pesquisadores, cientistas e desenvolvedores formaram uma comunidade interessada em usufruir do poder destes processadores, principalmente no contexto da computação científica.

Essa iniciativa ficou conhecida como GPGPU (do inglês, General-Purpose Computation on Graphics Processing Unit) e os fabricantes de placas de vídeo, reconhecendo o potencial deste mercado, começaram a investir em APIs e linguagens de programação de alto nível para permitir o desenvolvimento de aplicativos baseados em GPU. Entre as principais características das GPUs modernas, destacam-se o acesso rápido à memória de vídeo, clock de alta frequência e múltiplos pipelines programáveis [2]. Contudo, surge a necessidade de comparar o desempenho computacional dessas placas gráficas de diferentes fabricantes, e assim, usar essas informações como uma ferramenta para auxiliar engenheiros e cientistas na busca do melhor desempenho, aliado ao menor custo das tecnologias com o passar do tempo. Existem duas principais tecnologias utilizadas para atender a demanda de desenvolvimento de aplicações não gráficas. A primeira, o CUDA (do inglês, Compute Unified Device Architecture) [3], foi desenvolvida pela pioneira NVIDIA e permite aumentos significativos de performance computacional ao aproveitar a potência da unidade de processamento gráfico (GPU). Mas, apesar de ser amplamente utilizada no mercado, essa tecnologia é restrita para as placas gráficas desse fabricante e não permite a portabilidade das aplicações para GPUs de outros fabricantes. A outra tecnologia é o OpenCL (do inglês, Open Computing Language) [4], que é um padrão aberto para programação paralela de propósito geral em CPUs, GPUs e outros processadores. Ele consiste de uma API para coordenar computação paralela entre processadores heterogêneos e uma linguagem de programação multi-plataforma baseada na C99. O OpenCL é gerido pelo consórcio tecnológico Khronos Group e entre os desenvolvedores estão a Intel, IBM, Apple, AMD, Nvidia etc. A portabilidade da aplicação é um dos requisitos obrigatórios para que possamos comparar o desempenho de GPUs de diferentes fabricantes, por isso, usaremos o OpenCL como recurso para as implementações, pois com ele é possível escrever programas para dispositivos diferentes.

2. Trabalhos Relacionados

Purcell [5] realizou uma das primeiras experiências envolvendo GPUs programáveis ao explicar a modelagem de um algoritmo de ray tracing para GPUs. Seu trabalho demonstrou que o algoritmo escrito em assembly para a NVIDIA GeForce 3, no processador de fragmentos, conseguia ter um desempenho maior em GPU. Este trabalho publicado nos primórdios da era GPGPU demonstrou o potencial do hardware gráfico programável como um dispositivo para computação genérica e paralela. Uma comparação do desempenho do CUDA em contraste com o OpenCL foi feita no trabalho dos autores em [6]. Por meio da implementação de uma aplicação científica com elevado custo computacional, simulação Monte Carlo de um sistema de spin, eles mediram e compararam o tempo de transferência de dados da GPU para a CPU e vice-versa. Os tempos de execução dos kernels e os tempos de execução do ciclo de vida completo das execuções, tanto para o CUDA quanto para o OpenCL. Neste trabalho os testes restringiram-se a uma GPU da própria Nvidia e os autores concluíram que o CUDA era a melhor escolha para aplicações de alta performance. Em [7], os autores propuseram algoritmos eficientes para o processamento linear de imagem, explorando as extensões SIMD (do inglês Single Instruction, Multiple Data) fornecidas em processadores AMD e Intel. O trabalho experimental e os resultados obtidos sugeriram que implementações baseadas em OpenCL proporcionaram

um rendimento mais baixo, uma média de 1,8 vezes, do que implementações equivalentes que utilizam diretamente o SIMD intrínsecos suportado pelo compilador Intel. O interessante neste trabalho é a aplicação heterogênea que se pode fazer com o OpenCL, usado tanto para GPUs quanto para processadores multi-cores. Nosso trabalho faz contribuições interessantes, pois até o presente momento não se tem registro de uma análise comparativa do desempenho de uma aplicação com o OpenCL entre placas gráficas da Nvidia e AMD/ATI. Além disso, este trabalho contém implementações de filtros digitais para imagens em GPU usando o OpenCL.

3. Objetivo Geral Comparar o desempenho das GPUs dos fabricantes NVIDIA e AMD/ATI, por meio do Processamento Digital de Imagens com a Transformada Rápida de Fourier e o filtro de detecção de borda Canny. Assim, identificaremos qual GPU apresenta o melhor desempenho quando se utiliza o framework OpenCL como recurso computacional para a implementação destas rotinas.

3. Objetivo Geral

Comparar o desempenho das GPUs dos fabricantes NVIDIA e AMD/ATI, por meio do Processamento Digital de Imagens com a Transformada Rápida de Fourier e o filtro de detecção de borda Canny. Assim, identificaremos qual GPU apresenta o melhor desempenho quando se utiliza o framework OpenCL como recurso computacional para a implementação destas rotinas.

3.1. Objetivos Específicos

- Identificar os indicadores de desempenho que permitam avaliar comportamento das rotinas destes filtros;
- Avaliar o paradigma de programação das GPUs da Nvidia e AMD/ATI, buscando identificar as diferenças que podem afetar no desempenho das implementações;
- Implementar o algoritmo da FFT (do inglês, Fast Fourier Transform) e obter os dados de desempenho;
- Implementar o algoritmo do filtro para detecção de borda Canny e obter dados de desempenho;

4. Metodologia

Os procedimentos adotados para atingir os objetivos propostos serão formados de duas etapas: pesquisa bibliográfica e pesquisa experimental. A pesquisa bibliográfica será constituída de material publicado em periódicos, jornais, revistas, congressos e também a partir de livros. A revisão bibliográfica que será contemplará o tema Processamento Digital de Imagens a fim de esclarecer os principais conceitos de filtragem digital, tanto no domínio do tempo, quanto no domínio da frequência e especificamente quanto as rotinas implementados neste trabalho, a Transformada Rápida de Fourier e o filtro de detecção de borda Canny.

A etapa seguinte será o estudo de Modelos de Computação Paralela, onde identificaremos qual a classificação da arquitetura de processamento computacional usaremos, de acordo com a Taxionomia de Flynn[8], é uma etapa importante, pois, sabendo essa classificação entenderemos melhor como deverá acontecer a concorrência dos processos e o controle de fluxo da memória.

Identificar as instruções que poderão ser paralelizáveis e que se tornará no programa a ser executado nos diversos núcleos da GPU é uma etapa importante deste trabalho, assim, será necessário um estudo das especificações do OpenCL: linguagem, API, runtime e modelos de programação, e também das arquiteturas das GPUs Nvidia e AMD/ATI. Realizaremos testes de desempenho com estas implementações nestas GPUs e então analisaremos e compararemos os resultados.

Para as implementações dos filtros, usaremos o Microsoft Visual Studio 2010 professional para realizar as implementações dos filtros digitais em OpenCL no sistema operacional Microsoft Windows. Os experimentos contarão com um determinado número de amostras, que garantirão uma precisão mais alta nos resultados finais. Por fim, faremos uma análise e interpretação dos resultados para avaliar o desempenho das rotinas e determinar em qual GPU eles possuem o melhor desempenho.

5. Cronograma

A seguir, o cronograma para realização das tarefas.

Referências