

Processamento de Imagens usando CUDA

Caio de Moraes Braz

Monografia apresentada à disciplina
MAC5742 - Introdução à Computação Paralela e Distribuída

Prof. Alfredo Goldman vel Lejbman

Sumário

[1 - Introdução](#)

[2 - Processamento de Imagens](#)

[3 - Um pouco sobre CUDA](#)

[4 - Algoritmos usando CUDA](#)

[5 - Bibliotecas](#)

[6 - Conclusões](#)

[7 - Referencias](#)

1 - Introdução

Neste trabalho, nosso interesse é discutir o estado da arte da criação e implementação de algoritmos paralelos de processamento de imagens, especialmente usando a tecnologia CUDA da NVIDIA, que usa o poder de processamento das GPUs para outros propósitos além de computação gráfica.

Veremos um pouco sobre processamento de imagens além de uma introdução aos conceitos de CUDA e alguns detalhes sobre o que foi encontrado como sendo o estado da arte na área.

2 - Processamento de Imagens

Processamento de imagens é a área da ciência da computação que engloba qualquer forma de processamento de dados no qual a entrada e saída são imagens tais como fotografias ou quadros de vídeo. A maioria das técnicas envolve o tratamento da imagem como um sinal bidimensional, em geral uma matriz de pixels, no qual são aplicados padrões de processamento de sinais.

Atualmente, imagens tem ganhado mais importância devido à grande necessidade de visualização de dados, como por exemplo em aplicações de bioinformática e aplicações que dependem de big-data.

Muitos algoritmos nesta área podem ser considerados clássicos já, como detecção de bordas por gradiente e a transformada de watershed, que é um algoritmo de segmentação de imagens. Muitos destes algoritmos podem se beneficiar muito de implementações paralelas para

3 - Um pouco sobre CUDA

Lançado em junho de 2007, o CUDA (sigla para *Compute Unified Device Architecture*) é uma plataforma de computação paralela e um modelo de programação inventados pela NVIDIA com o intuito de usar o poder de processamento das GPUs para processamento geral (GPGPU).

As GPUs NVIDIA trabalham no modelo SIMD (Single Instruction Multiple Data), isto é, todos os processadores da GPU executam a mesma instrução em cada passo.

Devido ao número elevado de processadores, um algoritmo que possa ser paralelizado para GPU pode ter um ganho de performance muito alto, como veremos na sequência.

4 - Algoritmos usando CUDA

Nos artigos pesquisados sobre implementações em CUDA de algoritmos de processamento de imagens, era comum os autores não entrarem em detalhes de implementação, muitas vezes explicando apenas em um alto nível simplificado, dificultando o entendimento e a reprodutibilidade dos mesmos. Outro fato interessante é o de que os artigos que mostram estas implementações não são novos, sendo o mais recente de 2011.

Um dos poucos que teve uma explicação um pouco mais detalhada, ainda que de alto nível, foi o artigo [1], no qual conseguimos entender como foi feita a equalização de histograma.

Outros algoritmos que foram citados na bibliografia são a transformada discreta do cosseno (DCT), usada na compressão JPEG, a transformada discreta de wavelet (DWT), a transformada rápida de Fourier (FFT), remoção de nuvens, detecção de bordas por Sobel, entre outros.

Na maioria dos casos, os ganhos relatados pelos autores foram de pelo menos 40x, sendo que em alguns casos, como na remoção de nuvens, os ganhos chegaram a até 80x. Lembrando que estes ganhos não estão levando em conta a comunicação entre a memória da GPU e da CPU.

5 - Bibliotecas

Durante a revisão bibliográfica sobre o tema, foi notado que não haviam muitos artigos sobre algoritmos em CUDA depois de 2011, como salientado na seção anterior. Um dos motivos disso pode ser o desenvolvimento de bibliotecas para CUDA facilitando seu uso e por consequência a implementação de algoritmos na GPU. Algumas bibliotecas encontradas foram:

- GPU4VISION (Graz University of Technology)

Biblioteca criada pelo Institute for Computer Graphics and Vision da Graz University of Technology com intuito de promover avanços no campo de visão computacional. A página do projeto é <http://gpu4vision.icg.tugraz.at/>.

- OpenVIDIA (University of Toronto)

O projeto OpenVIDIA, criado pela Universidade de Toronto, implementa algoritmos de visão computacional e processamento de imagens em uma ou mais GPUs NVIDIA. A página do projeto é <https://www.openhub.net/p/openvidia>.

- MinGPU: A minimum GPU library for Computer Vision

A MinGPU é uma biblioteca que contém do modo mais minimalista possível, todas as funções necessárias para converter código serial existente para código GPU.

A ideia principal é facilitar a implementação por pessoas que nunca tiveram contato com as implementações em GPU antes.

A biblioteca funciona com GPUs NVIDIA e ATI (AMD) e a página do projeto é <http://vision.eecs.ucf.edu/MinGPU/>.

- NPP - NVIDIA Performance Primitives

A NPP é uma coleção de algoritmos em GPU para processamento de sinais (imagens, vídeo e áudio) que entregam performances de 5x a 10x maior que os códigos CPU.

A página do projeto é <https://developer.nvidia.com/npp>.

6 - Conclusões

Assim como sabemos que não é trivial paralelizar códigos para CPU, temos ainda mais desafios quando queremos usar GPUs. Os maiores desafios relatados pelos pesquisadores são a alocação de dados nas threads e a comunicação entre o *host* (CPU) e o *device* (GPU).

Na alocação, a ideia é dividir os dados e as tarefas de modo que nenhuma thread tenha trabalho a mais que outras. Criar algoritmos para isso é um dos desafios encontrados.

Na parte de comunicação é encontrado o maior gargalo de desempenho das implementações em GPU, pois todos os dados devem migrar da memória principal para a memória de vídeo e de volta para a memória principal. A ideia sempre é realizar o mínimo de comunicações possível, embora isso seja um problema pois cada GPU tem tamanhos diferentes de memória global e compartilhada, dificultando o processo.

7 - Referencias

1. Zhiyi Yang; Yating Zhu; Yong Pu, "Parallel Image Processing Based on CUDA," *Computer Science and Software Engineering, 2008 International Conference on* , vol,3, no,, pp,198,201, 12-14 Dec, 2008
2. In Kyu Park; Singhal, N,; Man Hee Lee; Sungdae Cho; Kim, C,W,, "Design and Performance Evaluation of Image Processing Algorithms on GPUs," *Parallel and Distributed Systems, IEEE Transactions on* , vol,22, no,1, pp,91,104, Jan, 2011
3. Nan Zhang; Yun-shan Chen; Wang, Jian-Li, "Image parallel processing based on GPU," *Advanced Computer Control (ICACC), 2010 2nd International Conference on* , vol,3, no,, pp,367,370, 27-29 March 2010