
Processamento de Imagens com CUDA

Caio Braz

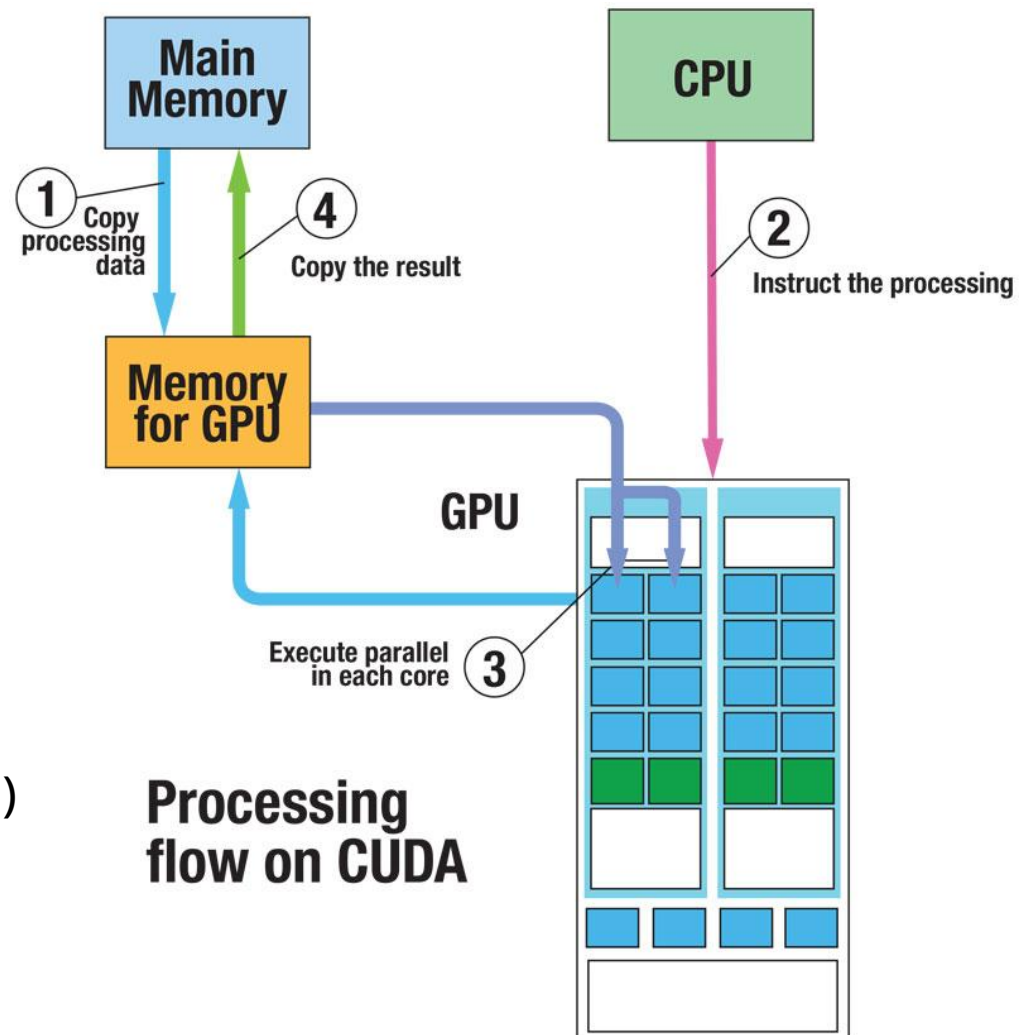
26 de junho de 2015

CUDA

Plataforma GPGPU lançada pela NVIDIA em junho de 2007

Suporte a OpenACC e OpenCL

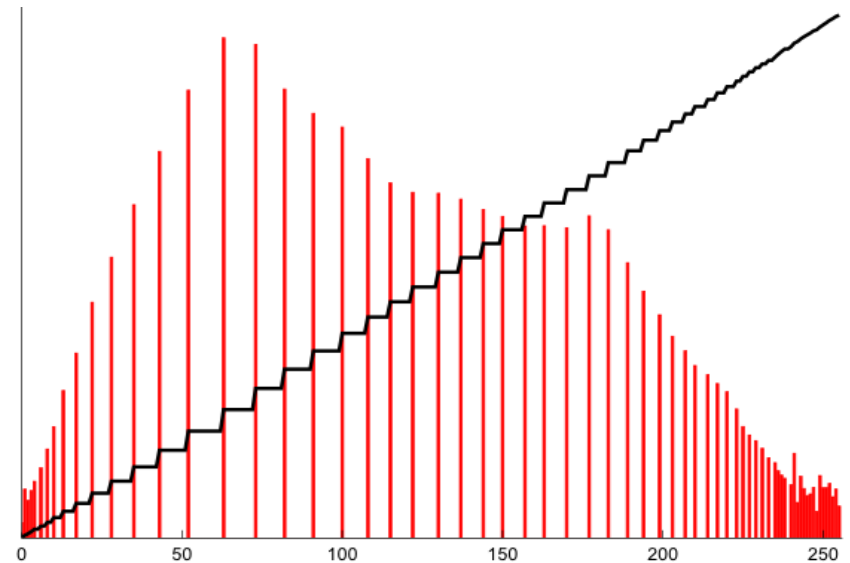
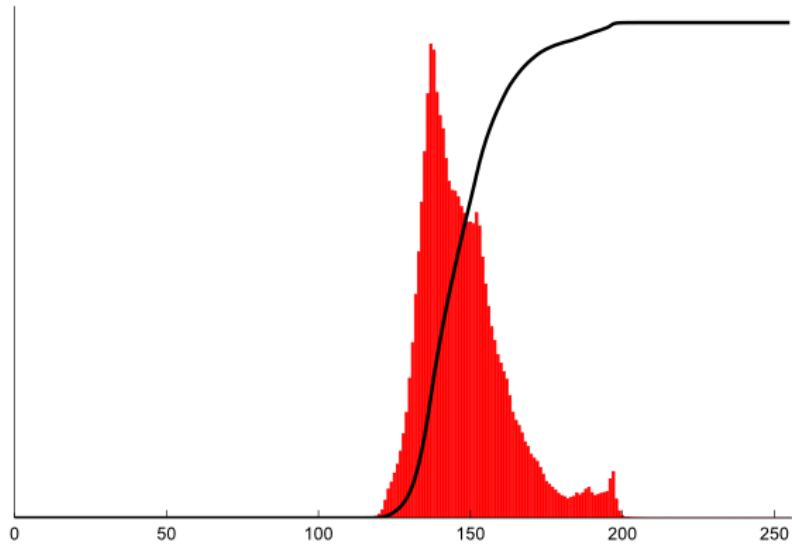
Modelo de execução SIMD
(Single Instruction Multiple Data)



Processamento de Imagens

- Transformada discreta de wavelet (DWT)
 - Transformada discreta de cosseno (DCT)
 - Transformada rápida de Fourier (FFT)
 - Filtro de Sobel
 - Equalização de histograma
-

Equalização de Histograma



Equalização de Histograma

Algoritmo simples: basta calcular a função distribuição acumulada (fda) da imagem e substituir o valor de cada pixel seguindo a fórmula:

$$I(p) = \left(\frac{fda(p) - fda_{min}}{P - fda_{min}} \times (L - 1) \right)$$



Imagem original



Imagem com histograma equalizado

Equalização de Histograma - CUDA

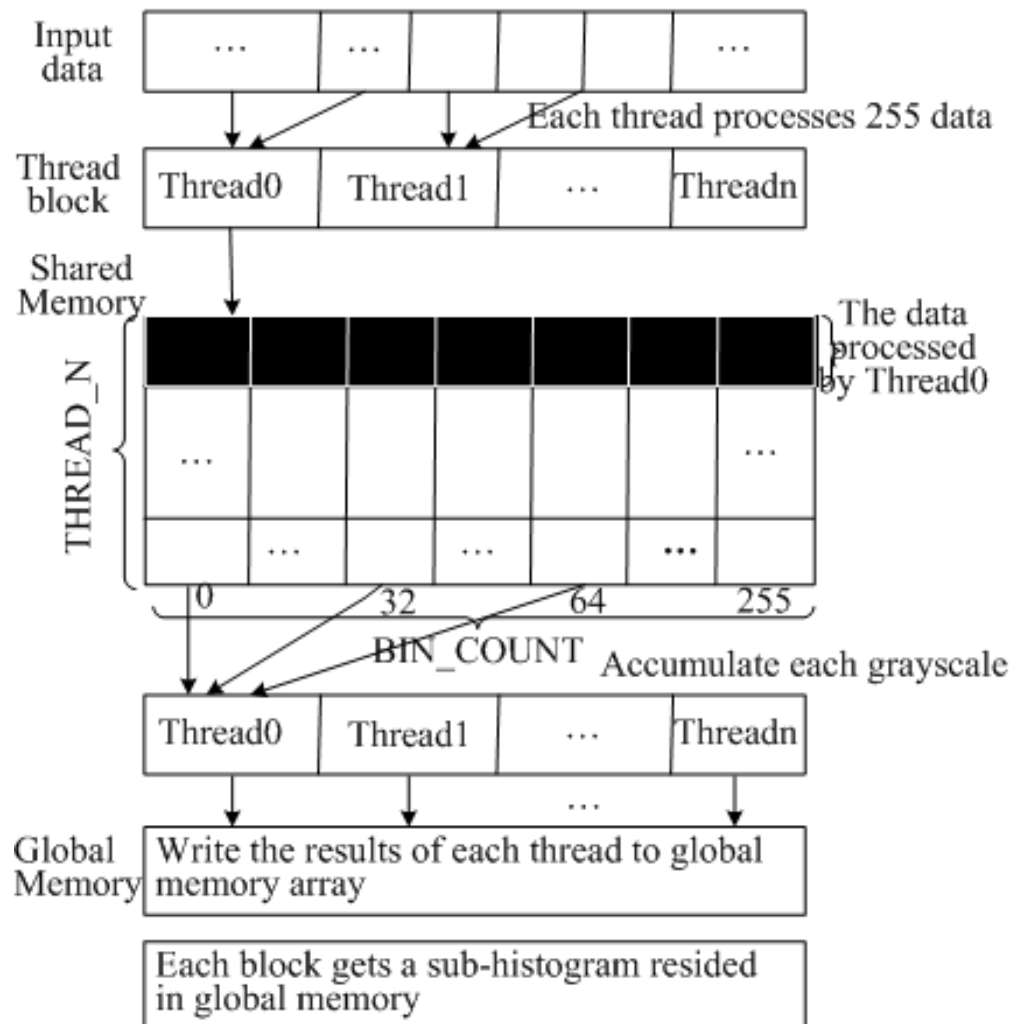
Usando n threads por bloco.

Cada thread processa L pixels e monta seu histograma local.

Todas as threads fazem uma redução de modo que cada thread guarde valores específicos de intensidade.

Cada thread agora monta na memória global o histograma local do bloco.

Após todos os blocos executarem, temos os sub-histogramas de cada bloco.

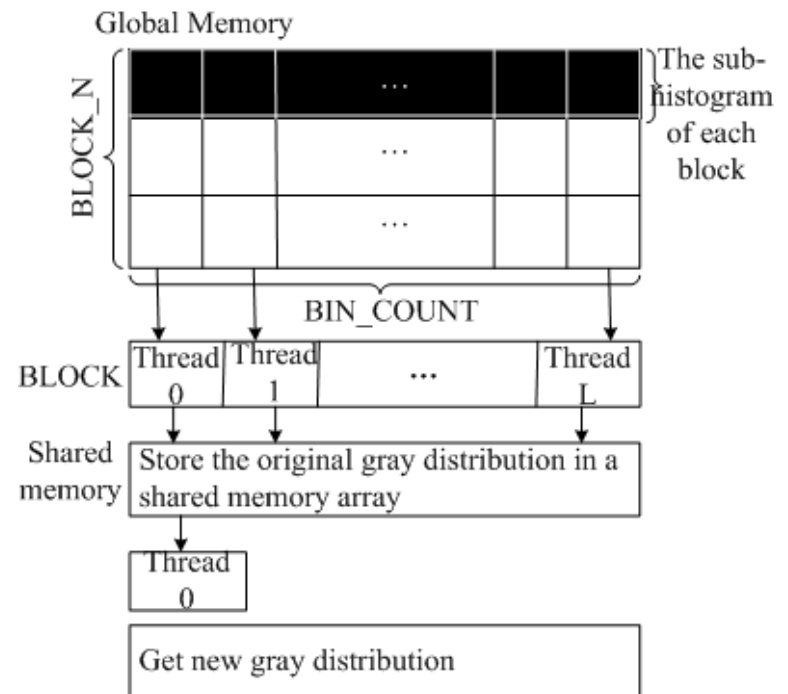


Equalização de Histograma - CUDA

Cada bloco reduz seu sub-histograma para criar o histograma global original da imagem.

Cada thread acumula um único nível de cinza

Finalmente, cada thread processo esse único nível de cinza e o substitui pelo valor adequado.



Resultados

Tamanho da imagem	CPU(ms)	GPU(ms)	Ganho(CPU/GPU)
5120x5120	1746,40	37,29	46,83
4096x4096	1126,38	24,12	46,70
3072x2304	479,11	10,57	45,33
2048x2048	280,50	6,50	43,15
1600x1200	128,51	3,25	39,54
1280x1024	91,44	2,46	37,17
1024x816	54,60	1,79	30,50
512x512	17,53	0,93	18,85
320x408	8,83	0,97	9,10

Desafios

- Alocação de dados nas threads

Balanceamento de trabalho nas threads aumenta a performance.

- Comunicação com o *host device*

Barramento de comunicação é o maior gargalo do processo, portanto pensar em maneiras de otimizar a comunicação entre *host* e *device* é importante!

Bibliotecas

- GPU4VISION (Graz University of Technology)
<http://gpu4vision.icg.tugraz.at/>
 - OpenVIDIA (University of Toronto)
<https://www.openhub.net/p/openvidia>
 - MinGPU: A minimum GPU library for Computer Vision
<http://vision.eecs.ucf.edu/MinGPU/>
 - NVPP - NVIDIA Performance Primitives
<https://developer.nvidia.com/npp>
-

Referências

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
 4. https://en.wikipedia.org/wiki/Histogram_equalization
 5. http://www.nvidia.com/object/imaging_comp_vision.html
-