

Uso de Graphics Processing Unit (GPU) na Restauração de Imagens de Microscopia de Força Atômica com Regularização de Tikhonov

Augusto G. Almeida

Instituto Federal do Rio de Janeiro, IFRJ

27197-000, Pinheiral, RJ

E-mail: augusto.ifrj@gmail.com

Dalmo Stutz

Universidade do Estado do Rio de Janeiro – Instituto Politécnico

28630-050, Nova Friburgo, RJ

E-mail: stutz@iprj.uerj.br

Geraldo A. G. Cidade

Universidade Federal do Rio de Janeiro – Instituto de Biofísica Carlos Chagas Filho

21941-902, Bonsucesso, RJ

E-mail: gcidade@biof.ufrj.br

Antônio J. da S. Neto

Universidade do Estado do Rio de Janeiro – Instituto Politécnico

28630-050, Nova Friburgo, RJ

E-mail: ajsneto@iprj.uerj.br

Resumo: Neste trabalho são expostos os resultados de uma implementação que tem por objetivo restaurar imagens estáticas provindas de microscópios de força atômica, com processamento realizado em Graphics Processing Unit (GPU). O uso de GPUs para processamentos diversos apesar de ainda ser novidade no campo da ciência vem demonstrando resultados com excelentes performances na velocidade de processamento. A esse novo campo da ciência foi dado o nome de General-Purpose computing on Graphics Processing Unit (GPGPU). Devido à natureza das GPUs o processamento difere do tradicional em vários aspectos, neste trabalho foi utilizada a biblioteca Compute Unified Device Architecture (CUDA) que torna a tarefa de implementação em GPU muito próxima da forma como é feita em CPU.

1. INTRODUÇÃO

O Microscópio de Força Atômica (AFM – Atomic Force Microscope [1]) é uma tecnologia que permite a aquisição de imagens em escala nanométrica de quase todo tipo de superfície, inclusive de materiais não-condutores (polímeros, cerâmicas, amostras biológicas, etc.). Entretanto, devido às características do equipamento e das dimensões de varredura, as imagens obtidas pelo AFM apresentam uma relação sinal/ruído pobre, causada tanto por borramentos quanto por ruídos aditivos, provenientes da própria instrumentação. Para recuperar estas imagens ou, ao menos, minimizar-lhes os efeitos de degradação, faz-se necessário o emprego de técnicas de restauração de imagens, tais como: morfologia matemática, transformada de Fourier (FFT), deconvolução, etc.

Visando melhorar o desempenho do algoritmo desenvolvido por Cidade, Silva Neto e Roberty [5] vem sendo implementada uma estratégia com processamento paralelo [6,7]. Neste trabalho é feita a implementação em GPU do algoritmo [5] que é de natureza serial.

2. MODELO DE DEGRADAÇÃO

O modelo do processo de degradação de uma imagem de AFM é dado por um operador B que descreve a função de borramento, também conhecido como Função de Espalhamento

Pontual (PSF - *Point Spread Function*), que juntamente com um termo de ruído aditivo η , opera sobre uma imagem x (imagem original), produzindo a imagem y observada pelo AFM (Fig. 1).

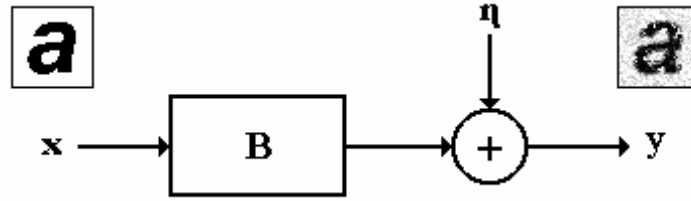


Fig. 1 - Modelo do processo de degradação da imagem obtida pelo AFM.

A partir do modelo podemos descrever o processo de degradação através da seguinte expressão geral

$$y = B * x + \eta \quad (1)$$

Dada a dificuldade em se obter uma representação exata para o operador B , é aqui assumido que ele seja uma representação matemática do tipo gaussiana dada por [4]

$$B \propto e^{-\frac{d^2}{\sigma^2}} \quad (2)$$

onde d é a distância dos pontos vizinhos a um ponto P da imagem e σ^2 é a variância que representa diferentes ponteiros usadas para fazer a varredura da amostra e produzir a imagem obtida no AFM.

Desprezando-se o ruído aditivo e redefinindo a Eq. (1) ponto a ponto, teremos

$$y(i, j) = \sum_{k=-N}^N \sum_{l=-N}^N b(k, l)x(i+k, j+l) \quad (3)$$

onde $i, j = 1, 2, \dots, M$ são as coordenadas do ponto $y(i, j)$ em uma imagem de dimensão $M \times M$, N é um valor arbitrário que define de forma discreta as dimensões de B ($\dim B = 2N + 1$), aqui expresso por $b(k, l)$ e $x(i+k, j+l)$ equivale às intensidades dos termos vizinhos ao ponto (i, j) .

3. TÉCNICA DE RESTAURAÇÃO BASEADA NO FUNCIONAL DE REGULARIZAÇÃO DE TIKHONOV

Partindo da Eq.(3), na busca de uma estimativa da imagem real, aqui denotada por \hat{x} , procura-se minimizar o funcional de regularização de Tikhonov, dado por [5]

$$Q(\hat{x}) = \| y(i, j) - \sum_{k=-N}^N \sum_{l=-N}^N b(k, l).\hat{x}(i+k, j+l) \|^2 + \alpha S \quad (\alpha > 0; i, j = 1, 2, \dots, M) \quad (4)$$

onde αS é o termo de regularização, que irá tratar da parte ruidosa da imagem, α o parâmetro de regularização e S corresponde a uma família de funções de regularização construída com distâncias de Bregman [2], baseados no funcional q -discrepância [3,4].

$$S = D_q(\hat{x}, \bar{x}) = \frac{1}{1+q} \sum_{i=1}^{M \times M} \sum_{j=1}^{M \times M} \left\{ \hat{x}_{i,j} \left[\frac{(\hat{x}_{i,j})^q - (\bar{x}_{i,j})^q}{q} \right] - (\bar{x}_{i,j})^q (\hat{x}_{i,j} - \bar{x}_{i,j}) \right\} \quad (5)$$

onde \bar{x} representa um valor de referência que pode ser a própria imagem borrada obtida pelo AFM e q é um parâmetro ajustável.

Na Fig. 2 é representado o algoritmo serial de restauração da imagem que é baseado na minimização do funcional dado pela Eq. (4), onde ω é o fator de relaxação, $\Delta\hat{x}^{t,0} = 0, \hat{x}^0$ um valor inicial estimado, γ um fator de atenuação ($0 < \gamma < 1$), t e c são contadores de iteração, e

$$F_{rs} = -2 \left\{ \sum_{i=1}^M \sum_{j=1}^M \left[y(i, j) - \sum_{k=-N}^N \sum_{l=-N}^N b(k, l) \hat{x}(i+k, j+l) \right] b(r-i, s-j) \right\} + \frac{\alpha}{q} (\hat{x}_{rs}^q - \bar{x}_{rs}^q) \quad (5)$$

$$\frac{\partial F_{rs}}{\partial \hat{x}_{mn}} = 2 \sum_{k=-N}^N \sum_{l=-N}^N b(k, l) b(r-m+k, s-n+l) + \alpha \hat{x}_{rs}^{q-1} \delta(r, m) \delta(s, n) \quad (6)$$

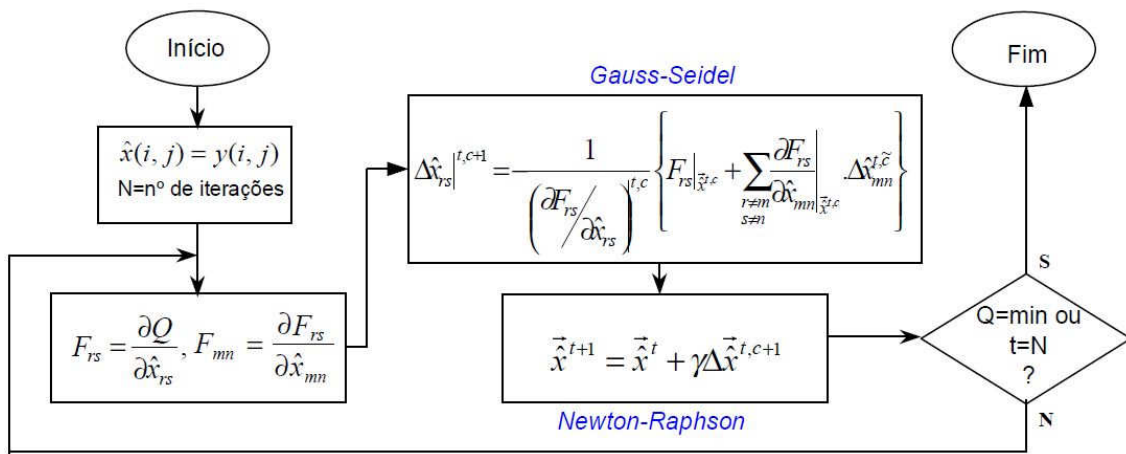


Fig. 2 - Fluxograma do algoritmo serial de restauração de imagens de AFM, baseado na Minimização do Funcional de Regularização de Tikhonov [4].

4. PROCESSAMENTO EM GPU

O atual estágio de evolução dos hardwares voltados para geração e manipulação de imagens (hardwares estes conhecidos comumente como placas gráficas), permite que processamentos outrora feitos em CPU possam ser realizados em GPU. Uma GPU consiste num processador que está embutido em uma placa gráfica cuja principal função via uma programação conhecida como *shader* é processar vértices (*vertex shader*) e pontos (*pixel shader*).

São três os motivos que sugerem o uso da GPU para processamento de imagens obtidas em nanoescala com microscópio de força atômica. O primeiro é que a manipulação de imagens providas de AFM exige intensidade de processamento em cálculos com ponto flutuante, isto é justamente uma das qualidades da GPU. O segundo motivo é o preço de uma placa gráfica que é consideravelmente menor que o equivalente em CPU. O terceiro motivo é a biblioteca ou *framework* conhecida como CUDA (*Compute Unified Device Architecture*) que em arquiteturas como a GeForce 8 (ou superior) torna a tarefa de implementação muito similar a escrita em linguagem C/C++ tornando assim mais rápido o ato de implementar os algoritmos. Com o intuito de obter a velocidade de processamento em GPU, utilizamos o algoritmo citado em [5], que possui uma estrutura de processamento serial. Existe um trabalho em desenvolvimento [6,7] que trata da paralelização do código já citado. Em GPU uma das grandes vantagens no processamento está no seu alto grau de paralelismo interno, porém este paralelismo difere em alguns pontos do paralelismo tradicional.

5. RESULTADOS PARCIAIS E DISCUSSÃO

Foram utilizadas duas GPUs, uma GeForce 8400 GS e uma Quadro FX 570, e uma CPU Core 2 Duo T5670 de 1.8 Ghz, e os dados com o tempos aproximados de processamento de uma imagem podem ser vistos na Tab. 1 abaixo.

Arquitetura	Configuração					
	(I)	(II)	(III)	(IV)	(V)	(VI)
CPU	0.8	2.2	1.8	6.5	2.3	10.7
8400	2.6	6.8	5.6	19.5	7.0	32.5
Quadro FX	2.5	6.6	5.4	19.4	6.8	32.2

Tab. 1 – Tabela Resultados dos tempos de processamento (em segundos).

Na Tab. 1 as referências significam :

(I) a matriz de borramento 7×7 (linhas x colunas) e resolução da imagem 256×256 (linhas x colunas);

(II) – $7 \times 7 / 512 \times 512$;

(III) – $15 \times 15 / 256 \times 256$;

(IV) – $15 \times 15 / 512 \times 512$;

(V) – $21 \times 21 / 256 \times 256$;

(VI) – $21 \times 21 / 512 \times 215$.

6. CONCLUSÕES

Os primeiros resultados obtidos com a implementação do algoritmo em GPU apontam uma deficiência no processamento serial nesta arquitetura, porque a velocidade de processamento de uma GeForce 8400 é de aproximadamente 450Mhz e a Quadro FX 570 aproximadamente 460Mhz. Contudo o ponto forte no desenvolvimento em GPU está no seu alto grau de paralelismo, e o como já citado o paralelismo tende a diminuir o tempo total de processamento. Isso incentiva a continuação da pesquisa nesta direção visando futuramente o processamento em tempo real de imagens dinâmicas obtidas com o microscópio. Cabe ressaltar que foram utilizadas GPUs de baixo custo, e existem diversas configurações dentro de uma arquitetura, e isso influencia diretamente na velocidade de processamento. Portanto GPUs de arquitetura GeForce 8800 ou GeForce 9800 GX2 ou a recente GTX295 terão ter performance de processamento expressivamente superior a Quadro FX 570.

Agradecimentos

Os autores agradecem o apoio financeiro do CNPq, da FAPERJ e da CAPES.

Referências

1 - BINNIG, G.; QUATE, E.C.F.; GERBER, C.H., *Atomic Force Microscope*, Phys. Rev. Lett., v. 56, n. 9, p. 930-933, 1986.

2 - BREGMAN, L.M., *The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming*. Zh. vŷchisl. Mat. mat. Fiz., v. 7, n.3, p. 620-631, 1967.

3 - CARITA MONTERO, R.F.; ROBERTY, N.C.; SILVA NETO, A.J. *Absorption Coefficient Estimation in Heterogeneous Media Using a Domain Partition Consistent with Divergent Beams*, Inverse Problems in Engineering, Vol. 9, pp. 587-617, 2001.

4 - CIDADE, G.A.G.; ANTENEODO, C.; ROBERTY, N.C.; SILVA NETO, A.J. *A generalized approach for atomic force microscopy image restoration with Bregman distances as Tikhonov regularization terms*, Inverse Problems in Engineering, Vol. 8, pp. 457-472, 2000.

5 - CIDADE, G.A.G.; SILVA NETO, A.J.; ROBERTY, N.C. *Restauração de imagens com aplicações em biologia e engenharia: Problemas inversos em nanociência e nanotecnologia*, São Carlos, SP: SBMAC, 2003. (Notas em Matemática Aplicada, 1).

6 - STUTZ, D. *Restauração de imagens em escala nanométrica com funcional de regularização de Tikhonov e computação paralela*. 2004. Dissertação (mestrado), Pós-graduação em Modelagem Computacional, Universidade do Estado do Rio de Janeiro, Instituto Politécnico, Nova Friburgo.

7 - STUTZ, D.; SILVA NETO, A. J.; CIDADE, G. A. G., *Parallel Computation Approach for the Restoration of AFM Images based on the Tikhonov Regularization Method*, Microscopy & Microanalysis, Vol. 11, Supplement S03, pp. 22-25, 2005.