

Otimização de Sistema de Identificação de Pontos de Bifurcação em Vasos Sanguíneos de Imagens de Retina usando GPU

Juliana A. Guimarães, Flavio S. L. De Souza, Karin S. Komati

Coordenadoria de Informática – Instituto Federal do Espírito Santo (IFES) Campus Serra Rodovia ES-010, Km 6.5 – 29.164-231 – Serra – ES – Brasil

{juju.guimaraes_flavio.lamas@gmail.com, kkomati@ifes.edu.br}

1. Introdução

A retina possui artérias e veias que podem ser estudadas computacionalmente. Com técnicas de processamento digital de imagens (PDI) é possível segmentar as veias e artérias, e conhecer seu comprimento e diâmetro, dentre outras medidas. Sua morfologia indica a presença de diversas doenças, tais como hipertensão e diabetes, e é útil para determinar seu avanço. A imagem colorida da retina passa por várias etapas, a primeira etapa é segmentação, separando quais pontos da imagem pertencem aos vasos sanguíneos e quais pontos não, gerando uma imagem binária (*black & white*).

Trabalho anterior, (Guimarães, Souza e Komati, 2012), já identificava os pontos de bifurcação, isto é, os pontos na "árvore sanguínea" onde ocorre a separação de um vaso em dois ou mais ramos, a partir da imagem segmentada da retina. Neste trabalho, o objetivo é avaliar o tempo de execução dos algoritmos de esqueletização em duas formas: com e sem o uso da unidade de processamento gráfico (GPU – *Graphic Processor Units*). Para tanto, utilizou-se o *Compute Unified Device Architecture - Cuda* (NVIDIA, 2012), que é uma plataforma de computação paralela criado pela NVIDIA, que permite aumento de performance computacional ao aproveitar a potência da GPU.

2. Solução Proposta

Um algoritmo de esqueletização reduz os vasos sanguíneos da imagem já segmentada a uma linha fina. A esqueletização elimina distorções no contorno, e mantém as propriedades geométricas e topológicas da imagem, permitindo a obtenção de propriedades críticas como pontos extremos e conexões.

Utilizou-se o método Zhang-Suen (Corrêa e Festa, 2005), cuja ideia é decidir se um determinado *pixel* será eliminado analisando o valor dos seus oito vizinhos. Há duas regras para decidir se o *pixel* deve ou não ser removido:

- a primeira regra diz que o *pixel* somente pode ser apagado se o número de conectividade do mesmo for igual a um. Isto significa que o *pixel* é conectado somente a uma única região. Se um *pixel* possuir o número de conectividade igual a dois então, duas regiões conectadas poderão se separar e isto não pode ocorrer;
- A segunda regra é que o *pixel* somente pode ser apagado se este tiver mais de um e menos de sete vizinhos. Esta regra assegura que os *pixels* resultantes foram retirados sucessivamente das bordas da região da imagem, e não de suas partes internas.

As duas regras do algoritmo de Zhang-Suen são analisadas em duas sub-iterações, onde uma sub-iteração exclui os *pixels* de contorno norte e oeste, e a outra sub-iteração exclui os restantes, sul e leste. É importante ressaltar que os *pixels* só devem ser eliminados no final da sub-iteração. Se no final da segunda sub-iteração não existirem *pixels* para serem eliminados, então a esqueletização está completa e o programa pára. Algoritmos de esqueletização paralelos divididos em duas sub-iterações preservam a conectividade. Caso contrário, um simples exemplo de um retângulo horizontal que possui dois *pixels* de largura pode vir a desaparecer durante o processo de afinamento.

Como a decisão de remover ou não um *pixel* na sub-iteração atual é baseada somente no resultado da sub-iteração anterior, todos os *pixels* podem ser analisados independentemente e de uma forma paralela a cada sub-iteração. A estratégia de paralelização adotada neste trabalho foi tratar a concorrência para cada linha da imagem e sincronizada a cada sub-iteração. Testes iniciais que paralelizavam cada *pixel* se mostrou pior que a abordagem por linha.

Para encontrar os pontos de bifurcação, o algoritmo analisa a quantidade de vizinhos de cada *pixel* da imagem esqueletizada. Se o número de vizinhos for maior que dois, o *pixel* está em um ponto de bifurcação, se o número de vizinhos for igual a um o *pixel* é um ponto terminal e se o *pixel* tiver exatamente dois vizinhos ele está em uma linha, não sendo necessário marcá-lo na imagem original. A Figura 1 mostra os resultados intermediários das etapas da solução.

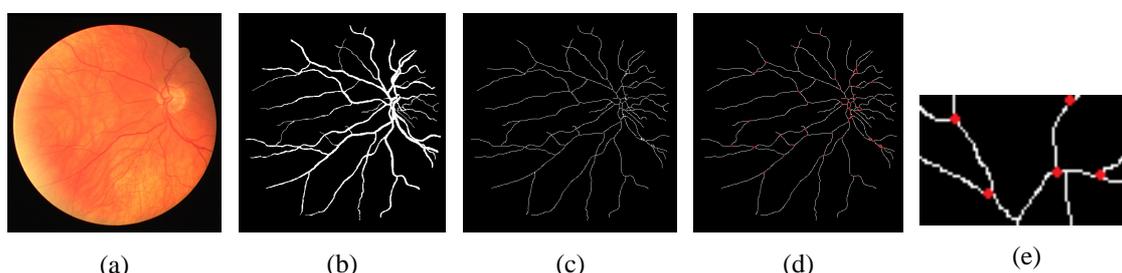


Figura 1. Exemplo das etapas da solução (a) imagem original (b) imagem da retina segmentada por um especialista (c) imagem resultado da esqueletização (d) pontos de bifurcação em vermelho (e) ampliação de uma região de (c).

O sistema foi desenvolvido em linguagem C, com a biblioteca OpenCV (Itseez, 2012), que é voltada para o desenvolvimento na área de visão computacional sob licença BSD. Gerou-se dois sistemas: um apenas com o OpenCV e outro com o OpenCV compilado com Cuda.

3. Resultados e Considerações Finais

Para testar nossa proposta, foram utilizadas algumas imagens da base de dados DRIVE (Staal et. Al, 2004). O algoritmo de esqueletização manteve a estrutura geográfica e topologia da imagem em todos os casos.

Para cada imagem de teste, executou-se 10 vezes o programa sem Cuda e 10 vezes o programa com Cuda. A Tabela 1 mostra a média dos tempos obtidos das 10 execuções em 3 imagens diferentes. Em média, o tempo sem Cuda é em torno de 3,5 vezes maior que o tempo com Cuda. Assim, verificamos que o uso de GPU realmente melhora em muito a performance do sistema.

Tabela 1. Medidas de tempo dos sistemas.

Imagem	Média de tempo sem Cuda em 10 execuções (Mt_sC)	Média de tempo com Cuda em 10 execuções (Mt_cC)	(Mt_sC)/(Mt_cC)
# 23	108.8 milissegundos	31.0 milissegundos	3,51 vezes
# 24	109.6 milissegundos	31.5 milissegundos	3,48 vezes
# 35	146.8 milissegundos	32.0 milissegundos	4,58 vezes

Como trabalho futuro, planeja-se:

- efetuar uma bateria de testes com 80 imagens para validar ainda mais o uso do Cuda como melhoria de performance;
- alterar a estratégia de paralelização, tratando a concorrência, dividindo a imagem em blocos de tamanhos diferentes para avaliar qual é a forma de melhor performance e
- comparar também a capacidade de cálculos matemáticos da CPU e da GPU afim de mostrar qual o nível de aproveitamento da GPU seu trabalho conseguiu atingir.

4. Agradecimentos

Os autores agradecem o apoio financeiro dado pelo Núcleo Incubador Serra para participação dos alunos apresentadores no congresso Computer On The Beach (COTB), e ao IFES Campus Serra pelo transporte dos alunos da caravana para participação no congresso (COTB).

References

- Corrêa, F. P. e Festa, L. M. (2005). "Avaliação de Técnicas para Afinamento de Imagens Digitais". Trabalho de Conclusão de Curso de Bacharelado em Ciência da Computação da Universidade Federal do Paraná.
- Guimarães, J. A.; Souza, F. S. L. e Komati, K. S. (2012). "Identificação de Pontos de Bifurcação em Vasos Sanguíneos de Imagens de Retina Usando Algoritmo de Esqueletização". Em: Anais do III ERI-MT 2012 Escola Regional de Informática do SBC (Regional Mato Grosso), Rondonópolis.
- Itseez (2012). "OpenCV". <http://opencv.org/>, Outubro.
- NVIDIA (2012). "Cuda". http://www.nvidia.com.br/object/cuda_home_new_br.html, Outubro.
- Staal, J. J.; Abramoff, M. D.; Niemeijer, M.; Viergever, M. A. and van Ginneken, B. (2004) Ridge based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging*, vol. 23, pp. 501-509.