

Projeto 4: Segmentação de imagens

SuperComputação 2018/2

Igor Montagner

Este projeto aplicará os conceitos de GPGPU vistos em aula em um problema real de segmentação de imagens. Os objetivos de aprendizagem trabalhados serão:

1. utilização de ferramentas para processamento de dados em GPU para resolução de um problema complexo.
2. profiling de código GPU usando `cudaEvent`.

Parte 0 - segmentação de imagens

Uma segmentação de imagem é sua divisão em regiões *conexas* (sem interrupções) contendo objetos de interesse ou o fundo da imagem. A segmentação iterativa de imagens é baseada na adição de marcadores em objetos de interesse para diferenciá-los do fundo (que contém todos os outros objetos e plano de fundo da imagem). A figura abaixo exemplifica este tipo de interação. Uma demonstração deste tipo de técnica é mostrada [neste vídeo](#).

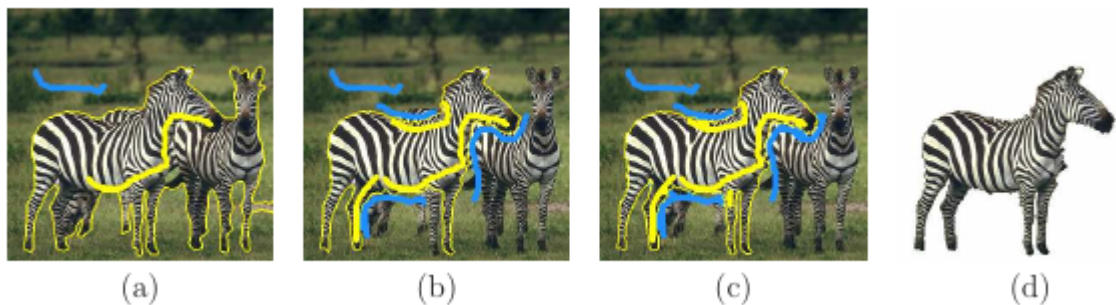


Figure 1: Segmentação de uma zebra pela adição de marcadores de frente (em amarelo) e fundo (em ciano)

Um algoritmo simples e muito eficaz para este problema é a IFT (Image Foresting Transform), que representa a imagem usando um grafo e trata a segmentação de imagens como um problema de caminho mínimo entre os vértices (pixels) semente e todos os outros pontos da imagem. Cada pixel (vértice) está conectado com os 4 pixels (vértices) adjacentes. O custo de cada aresta pode ser dado de duas maneiras:

1. Magnitude da diferença entre os valores dos pixels na imagem original;
2. Magnitude da diferença entre os valores dos pixels na imagem de bordas (gradiente morfológico ou laplace);

O algoritmo básico para resolver este problema é uma modificação do [algoritmo de Dijkstra](#) para a determinação de caminhos mínimos vista em Redes Sociais. Cada pixel v pertence a uma semente/objeto com semente s_i se o caminho de custo mínimo de v até s_i possui menor custo que o caminho de custo mínimo até todas outras sementes/objetos $s_j, j \neq i$.

Como visto no vídeo acima, a segmentação de imagens de maneira iterativa necessita de resultados quase instantâneos para que o processo seja agradável para o usuário. Neste contexto iremos usar bibliotecas de processamento de grafos em GPU para tentar acelerar este processo para imagens de alta resolução. Nossos objetivos são

1. Implementar o processo completo em um algoritmo sequencial **eficiente**;
2. Usar ferramentas de GPU para acelerar cada etapa do processo sequencial.
3. Quantificar o tempo que cada etapa do processamento leva e comparar com a implementação sequencial.
 - Criação do grafo / cópia de dados

- Algoritmos de caminho mínimo
- Criação da imagem final segmentada.

Apesar de retomar parte do conteúdo de Redes Sociais, uma parte significativa do conteúdo para a criação do programa sequencial é nova e requer estudo do material passado na seção de Referências. Os materiais já estão em ordem de facilidade de leitura.

Parte 1 - Implementação

Esta seção descreve quais conceitos serão atribuídos para cada parte do trabalho implementada. Todos os conceitos são incrementais. As sementes serão recebidas (inicialmente) pela linha de comando com o seguinte formato:

```
n_sementes_frente n_sementes_fundo
x_0 y_0
x_1 y_1
....
```

Conceito I

O projeto não foi entregue ou o que foi entregue não alcança todos os requisitos para um conceito **D**

Conceito D

Para obter este conceito seu programa deverá estar implementado usando a biblioteca **nvGraph** para a busca de caminhos mínimos. Você precisará

- ler a imagem
- receber um ponto para semente de frente e um ponto para semente de fundo
- montar o grafo a partir da imagem original usando **nvGraph** e calcular os caminhos mínimos.
- criar uma imagem máscara com cor branca para a frente e preta para o fundo

Você também deverá criar medições de desempenho (usando **cudaEvent**) para as seguintes etapas do seu programa:

1. montagem do grafo
2. cálculo dos caminhos mínimos
3. montagem da imagem segmentada
4. tempo total do programa.

Conceito C

Este conceito requer

- a implementação de um filtro de bordas usando CUDA **C** e construção do grafo usando a imagem de bordas
- aceitar mais de uma semente para frente e mais de uma semente para fundo na sua implementação em GPU **e no programa sequencial**

Conceito C+

Como mostrado no artigo [5], frameworks de processamento de grafos em GPU possuem desempenhos muito diferentes. Selecione um segundo framework de grafos e faça uma nova versão do algoritmo de segmentação, desta vez usando este novo framework para calcular os caminhos mínimos. Você deve incluir a resposta a seguinte pergunta:

Qual dos grafos usados no artigo para comparar os desempenhos é mais parecido com os grafos de imagens que estamos analisando? Qual o desempenho do framework escolhido em relação ao **nvgraph**?

Conceito B+

Para obter este conceito você deverá implementar o algoritmo descrito em *A new GPU-based approach to the Shortest Path problem* em *CUDA C* e comparar seus resultados (desempenho e saída) com as alternativas anteriores.

Conceitos A e A+

Cada um dos itens abaixo vale 1 ponto e pode ser feito se o conceito **B+** já foi alcançado.

- Implementação sequencial do algoritmo de *Dijkstra* mais versões SIMD e OpenMP.
- Implementou uma ferramenta gráfica para o posicionamento das sementes e apresenta de maneira gráfica os resultados. Seu programa deverá permitir a segmentação interativa da imagem de maneira natural.

Parte 2 - relatório

Você deverá entregar um relatório mostrando

1. os resultados de segmentação para um conjunto de imagens simples (que pode ser compartilhado entre vocês)
2. testes com imagens de alta resolução, incluindo medições de tempo
3. comparação com um programa sequencial disponibilizado no blackboard

A rubrica de avaliação é a mesma que a usada nos projetos anteriores.

Parte 3 - avaliação e burocracias

A data de entrega está, provisoriamente, colocada para o dia **27/11**. Sua nota final será composta por 50% da sua implementação, 40% do seu relatório e 10% em qualidade de projeto/código. Você precisará de conceito **C** em todas as três partes, caso contrário sua nota final será **D**.

Parte 4 - referências

1. Slides sobre IFT - Paulo Miranda - [link para download](#)
2. Proposta de tutorial - *Graph-Based Image Segmentation*, Alexandre Falcão, Thiago Spina, Paulo Miranda, Fábio Cappabianco, SIBGRAPI 2010. [link para download](#)
3. Falcão, Alexandre X., Jorge Stolfi, and Roberto de Alencar Lotufo. "The image foresting transform: Theory, algorithms, and applications." *IEEE transactions on pattern analysis and machine intelligence* 26.1 (2004): 19-29. [link para download](#)
4. Ortega-Arranz, Hector, et al. "A new GPU-based approach to the shortest path problem." *High performance computing and simulation (HPCS), 2013 international Conference on*. IEEE, 2013. [link para download](#)
5. Aasawat, Tanuj Kr, Tahsin Reza, and Matei Ripeanu. "How well do CPU, GPU and Hybrid Graph Processing Frameworks Perform?." *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 2018. [link para download](#).