Insper

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. implementação de um algoritmo (Dijkstra) baseado em sua descrição formal.
3. comparação de desempenho entre a implementação sequencial simples e uma implementação usando GPU em alto desempenho.

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.

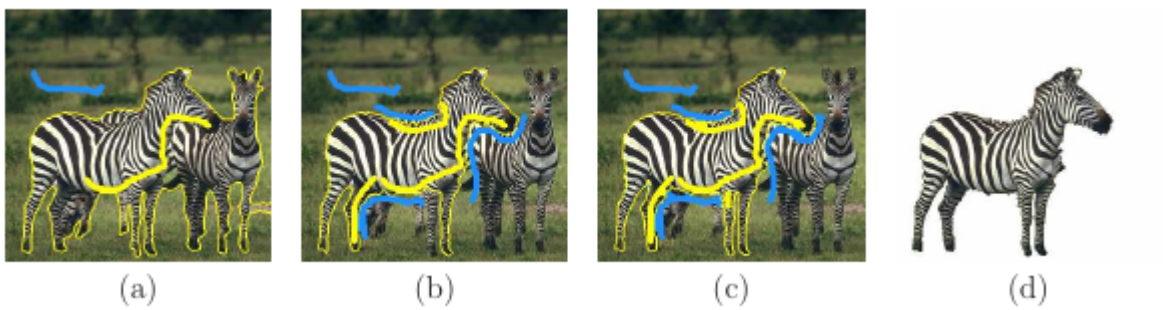

Figure 1: Segmentação de uma zebra pela adição de marcadores de frente (em amarelo) e fundo (em ciano)

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 no 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 materias 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.

Conceito I

O projeto não foi entregue ou o que foi entregue não funciona nem no sequencial.

Conceito D

O projeto entregue carrega imagens, implementa um filtro de bordas (gradiente morfológico ou laplace), aplica o algoritmo de Dijkstra e interpreta seus resultados para obter uma segmentação da imagem. Basicamente, este conceito envolve fazer uma implementação totalmente sequencial do processo. Você pode supor que existe somente uma semente para frente e uma semente para o fundo. Seu resultado final deverá ser uma imagem como a mostrada na parte 0.

Conceito C

O caminho entre as sementes e cada pixel é calculado usando a biblioteca nvGraph. O filtro de bordas é calculado usando `cuda::thrust`.

Conceito C+

O filtro de bordas é feito usando um kernel escrito em *CUDA C*. Mais de uma semente de frente e fundo pode ser usada (tanto para o sequencial como para a versão GPU).

Conceito B/B+

Cada tarefa implementada equivale a uma parte do conceito. Se quiser fazer somente uma você pode escolher qualquer uma das duas.

1. usar outro framework para o problema dos caminhos mínimos (referência 5).
2. melhorar seu sequencial com openmp e/ou simd.

Em ambos os casos você deve descrever como cada item foi implementado no seu relatório e mostrar os ganhos de desempenho em relação ao sequencial e/ou ao programa apresentado no conceito **C+**.

Conceito A

Implementou o artigo *A new GPU-based approach to the Shortest Path problem* em *CUDA C* e comparou seu desempenho com as alternativas anteriores.

Conceito A+

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.

Este conceito pode ser feito se você implementar corretamente ao menos o conceito **C+**.

Parte 2 - relatório

Você deverá entregar um relatório comparando o desempenho de sua implementação sequencial com sua implementação baseada em GPU. Neste projeto a simples medição de tempo total do programa resultará em conceito **D** no relatório. Você deverá separar as partes do processo e medir o desempenho de suas implementações em cada parte de maneira a quantificar as melhoras (ou pioras) de desempenho.

Para as partes em *GPU* deverá ser usado o framework *cudaEvent* visto em sala de aula.

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.