Ci1009 Lab 3 – ERE2 (2020) - versão 1.0
CUDA Image Blur

(baseado no GPU Teaching Kit -- Accelerated Computing
Modificado por W.Zola/UFPR)

Objetivo

O objetivo deste laboratório é implementar um algoritmo eficiente para desfoque de imagem. Cada pixel das imagens, tanto de entrada quanto de saída, tem 3 bytes, que são quantidades de Vermelho (R), verde (G) e azul (B). O bitmap de uma imagem de n x m pixels é uma matriz (array) 2D de pixels.
Seu kernel deve percorrer o bitmap da imagem de entrada e produzir o bitmap de saída, para cada pixel use um filtro de caixa 5x5 para desfocar a imagem original e produzir o pixel correspondente na imagem borrada (de saída).

O tamanho (largura e altura) do filtro está definida no código com:

```
#define BLUR_SIZE 5
```

Pré-requisitos

Antes de iniciar este laboratório, certifique-se de que:
• Você concluiu o lab2

Formato das imagens

As imagens de entrada e de saída estão em formato PPM P6 (RGB) e podem ser lidas pela função wbImport(…) que está em wb4.h (nossa biblioteca).
Os alunos podem criar suas próprias imagens de entrada, caso queiram, mas o diretório Dataset e DatasetW contém os casos de testes que queremos verificar para diversos tamanhos de imagem.

A maneira mais fácil de criar a imagem é por meio de ferramentas externas.
No Unix, existem programas para converter imagens para PPM.

A wblib já tem uma função para ler a imagem e para gravar as imagens PPM.

Instruções

Edite o código código para realizar o seguinte:
• alocar memória do dispositivo
• copiar a memória do host para o dispositivo
• inicializar o bloco de threads e as dimensões da grade do kernel
• invocar kernel CUDA
• copiar os resultados do dispositivo para o host
• desalocar memória do dispositivo

As instruções sobre onde colocar cada parte do código estão demarcadas pelas // @@ linhas de comentário no código.

Instruções de configuração local

A versão mais recente do (programa) modelo/esqueleto para este laboratório (código-fonte e instruções) está em ~wagner/ci1009-ere2-2020/labs/lab3 na máquina orval ou macalan.

Você deve fazer uma cópia de cada diretório para sua conta pessoal
e trabalhar com sua cópia.
O professor lhe dará instruções sobre onde/como entregar sua solução final, provavelmente enviando um e-mail com:

assunto: [ci1009-ERE2] lab3

anexando um arquivo **lab3-uid.tar.gz** com sua solução, onde:
uid é o seu uid (nome) na máquina orval.

O executável gerado como resultado da compilação de sua solução de laboratório deve ser executado usando o seguinte comando:

./ImageBlur -i **<**input.ppm**>** -o **<**output.pbm**>**

onde:
<input.ppm> é a imagem (do dataset), e
<output.pbm> é a imagem de saida da **solução** , também já está no dataset

o programa deve invocar a função wbSolution(…) para verificar se a saída do seu kernel confere com a solução existente no dataset.

Code Template

O modelo de código no diretório do laboratório é sugerido como ponto de partida para estudantes. O código lida com a importação e exportação, bem como o verificação da solução.

Os alunos devem inserir seu código é
as seções demarcadas com // @@.

Espera-se que os alunos não alterem a biblioteca, ou seja, apenas adicionem o seu código para fazer a transformação de imagem e para ativar as funçõess de CUDA necessárias.

Neste ponto, o wblib original não é funcional, então o professor está fornecendo uma versão "work-arround" da lib, p wb4.h

Modificada por W.Zola/UFPR