SPP – CUDA Lab

TECHNISCHE UNIVERSITÄT DARMSTADT

Zia Ul Huda

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

- Task – Edge detection in images

Black & White Image Conversion

TECHNISCHE UNIVERSITÄT DARMSTADT

- Purpose: Detection of edges in an image is a frequent problem in computer vision
- Input:
  - A m x n color image in RGB format
- Output:
  - A m x n black & white image with boundaries of objects clearly shown

Grayscaling

TECHNISCHE UNIVERSITÄT DARMSTADT

- Definition:
  - Conversion of a color image into corresponding black & white image
- Many techniques:
  - Averaging all RGB values
  - Desaturation
  - Decomposition

- We use **Colorimetric technique** for this project
  - Gray = 0.2126 R + 0.0722 B + 0.7152 G

Convolution Filters

TECHNISCHE UNIVERSITÄT DARMSTADT

- Definition:
  - Convolution is a process of adding each pixel to its neighbors weighted by some matrix values
  - Useful for performing different filters on images

- Example:

$$\left(\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} * \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}\right)[2,2] = (i*1)+(h*2)+(g*3)+(f*4)+(e*5)+(d*6)+(c*7)+(b*8)+(a*9)$$

Where * is the convolution operation and [2,2] is the location of the pixel which is being added to its neighbors.

Gaussian Blur

TECHNISCHE UNIVERSITÄT DARMSTADT

- Gaussian blur is used to blur the image using Gaussian function
- We use it to smooth the gray-scaled image from previous step to improve the quality of edge detection
- We use a 3x3 convolution filter for this operation

Sobel Filter

TECHNISCHE UNIVERSITÄT DARMSTADT

- Sobel filter is used to detect edges in the input image

- A 3x3 convolution filter is used

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * A \qquad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$

and the resultant pixel is computed as:

$$G = \sqrt{G_x^2 + G_y^2}$$

Example

TECHNISCHE UNIVERSITÄT DARMSTADT

Original

Grayscale

Gaussian blur

Sobel

General Remarks

TECHNISCHE UNIVERSITÄT DARMSTADT

- We provide sequential version of a C implementation of the edge detection
  - You need to implement CUDA kernels of appropriate functions
  - Todos are provided in the template file at the appropriate places

- The input image is in bmp format
  - Reading and writing of a bmp image is already provided

Tasks

TECHNISCHE UNIVERSITÄT DARMSTADT

- Task 1 – Compute the appropriate grid size to provide to cuda kernel calls
- Task 2 – Allocate appropriate memory on the device and initialize it to zero
- Task 3 – Allocate memory for the input image on device and copy the input image to the device memory
- Task 4 – Implement `cuda_grayscale` kernel

Tasks

TECHNISCHE UNIVERSITÄT DARMSTADT

- Task 5 – Implement `cuda_applyFilter` function
- Task 6 – implement `cuda_gaussian` kernel
- Task 7 – Implement `cuda_sobel` kernel
- Task 8 – Transfer the newly computed images to the main memory from the device memory at the appropriate places and write them to the disk for verification
- Task 9 – Free all of the allocated memory on the device