

CS542200

Parallel Programming

CHEN-CHUN CHEN

2016/12/08

Outline

1. Problem Description
2. Preparation
3. Your Tasks
4. Reminder

Outline

1. Problem Description
2. Preparation
3. Your Tasks
4. Reminder

Problem Description

Edge Detection: Identifying points in a digital image at which the image brightness changes sharply.



Sobel Operator

Used in image processing and computer vision, particularly within edge detection algorithms.

Uses two 3x3 kernels g_x , g_y which are convolved with the original image to calculate approximations of the derivatives - one for horizontal changes, and one for vertical.

At each point in the image, the result of the Sobel operator is either the corresponding gradient vector or the norm of this vector.

The convolution matrix

g_x, g_y are isotropic 3x3 Image Gradient Operator

$$g_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, \quad g_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Each pixel consists of 3 values R, G, B

How it works

For each output pixel, we need to refer 9 input pixels around to determine its value

Each color channel is convolved with g_x and g_y

c is a constant

$$G_x = (g_x * A) \times c, G_y = (g_y * A) \times c$$

12	100	32	55
7			

12	100	32	55
7	12		

12	100	32	55	
7	12	248		

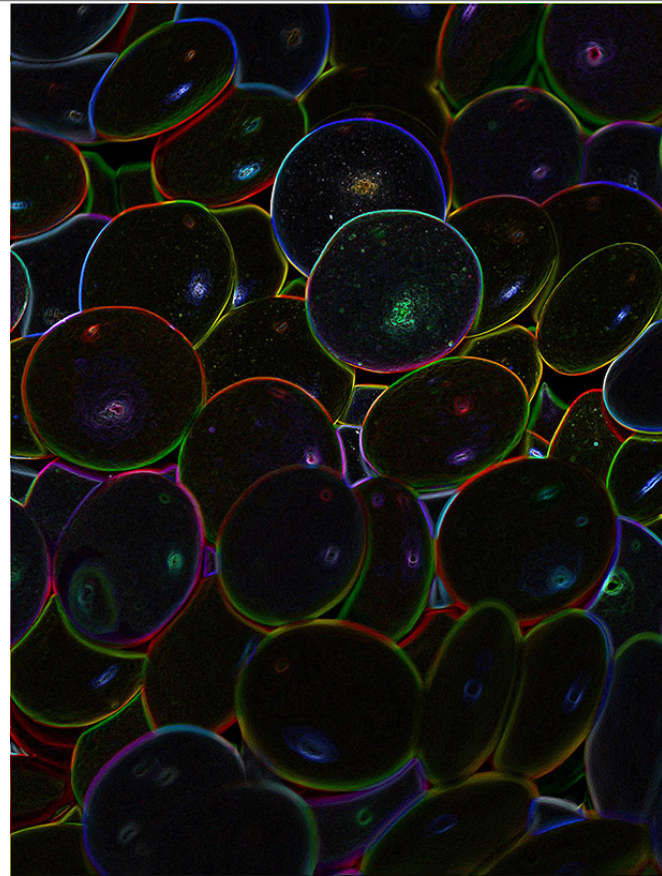
5x5 variation

We use this kernel instead of the 3x3 one in this lab.

$$g_x = \begin{pmatrix} -1 & -2 & 0 & 2 & 1 \\ -4 & -8 & 0 & 8 & 4 \\ -6 & -12 & 0 & 6 & 12 \\ -4 & -8 & 0 & 8 & 4 \\ -1 & -2 & 0 & 2 & 1 \end{pmatrix},$$

$$g_y = \begin{pmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

Sample Result



Outline

1. Problem Description
2. Preparation
3. Your Tasks
4. Reminder

Preparation

Login to the server

Copy shared.tar.gz to the server

- `$ cp -r /home/pp2016/shared/lab2 ~/`

You should be able to see these files:

- `$ cd ~/lab2`
- `$ ls`
 - `sobel.cu`
 - `Makefile`
 - `(sth).bmp`
 - `(sth)_out.bmp`
 - `job_cuda.sh`

Sobel Image Filter

The program `sobel.cu` is an example code of Sobel image filter.

`./sobel INPUT_FILE`

- **INPUT_FILE**: the input BMP file

The sample output file is of name `*_out.bmp`

For example, You can verify your output by

- `cmp {YOUR_OUTPUT} candy_out.bmp`
- You can also download the file and see the result!

Compile and run

Please refer to the **PP2016_Lab2_tutorial.pptx** slide for detail instructions

For this lab, a makefile has been provided
You can simply run *make* to compile

You are recommended to use `job_cuda.sh` to submit you gpu jobs

Outline

1. Problem Description
2. Preparation
3. Your Tasks
4. Reminder

Working Items

Task 1: Turn `sobel()` into kernel function

- Add `cudaSetDevice`, `cudaMalloc`, `cudaMemcpy`, ... etc.
- Relabel index to combination of `threadIdx`, `blockIdx`, ... etc.

Task 2: Put `mask[][][]` into shared memory

- Add `__shared__` , `__syncthreads()`
- Access through shared memory instead of global memory

Task 3: Use pinned memory

- Add `cudaMallocHost`

Outline

1. Problem Description
2. Preparation
3. Your Tasks
4. Reminder

Reminder

See comments in `sobel.cu` for HINTS

Please finish it before leaving

If you cannot finish all tasks in time,
you can finish it at home

- Deadline: 12/11 (Sun) 23:59:59

Submission:

- Directly copy **`sobel.cu`** and **`Makefile`** to
`~/homework/Lab2/` (at pp31)