CS542200
Parallel Programming 2015
Lab2: Sobel Image Filter (with color)

Chin-Feng Lee, LSA Lab, NTHU

2015/11/25

Outline

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

Outline

1. Problem Description
2. Preparation
3. Your Tasks
4. Grading
5. 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

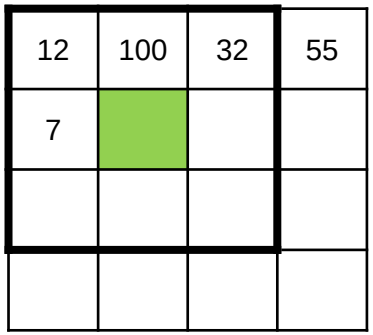
- g_x, g_y are isotropic 3x3 Image Gradient Operator

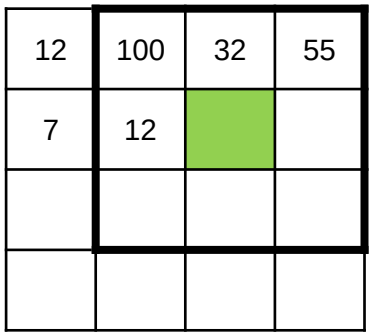
$$g_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, \qquad g_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

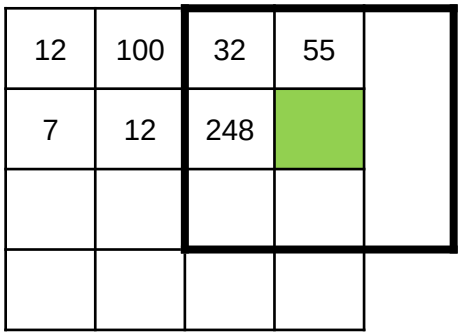
- Each pixel consists of 3 values R, G, B

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







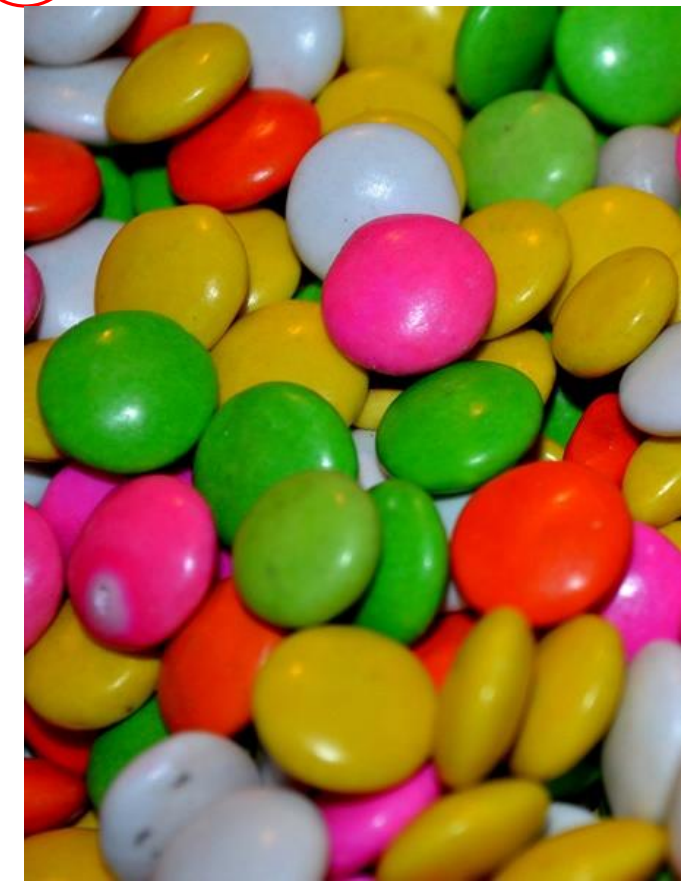
5x5 variation

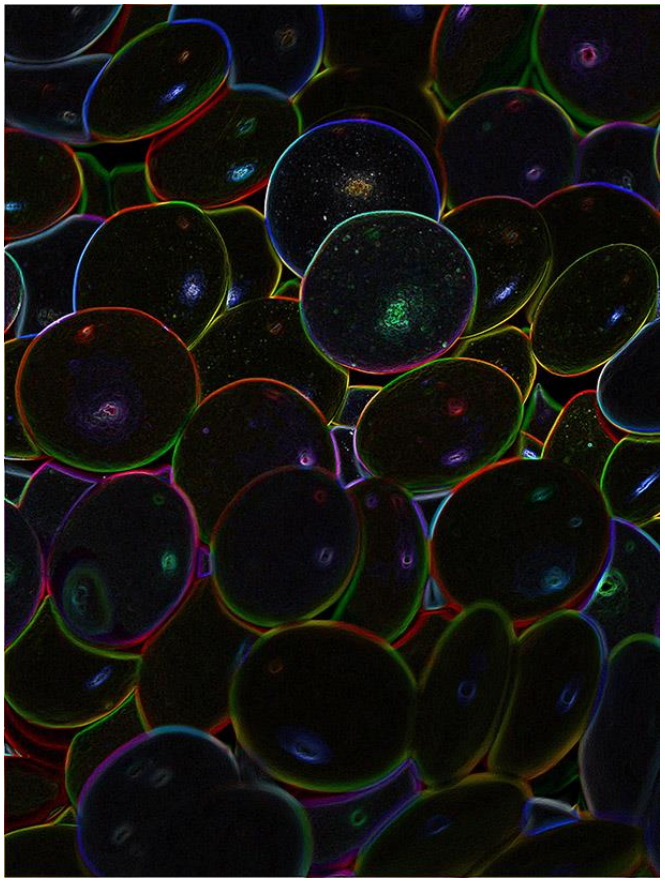
- We use this kernel instead of the 3x3 one in this lab.

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

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

Sample Result





Outline

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

Preparation

- Login to the server
- Upload shared.tar.gz to the server
  - You can use FileZilla, MobaXTerm or scp command
- Untar the file
  - `tar axf shared.tar.bz2`
- You should be able to see these files:
  - sobel.cu
  - Makefile
  - (sth).bmp
  - (sth)_out.bmp

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 **PP2015_Lab2_tutorial.pptx** slide for detail instructions

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

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. Grading
5. Reminder

Lab 2 Grading

- Total 2%
  - Attendance (0.5%)
  - Task 1: CUDA version (0.5%)
  - Task 2: Shared memory (0.5%)
  - Task 3: Pinned memory (0.5%)

Outline

1. Problem Description
2. Preparation
3. Your Tasks
4. Grading
5. 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, and submit to iLMS
  - 10% off before 11/30 (Mon) 23:59:59
  - Penalty of late submission doesn’t effect the points you got in lab time