HW3: Sobel (CUDA)

Due: Mon, 2020/5/4 23:59

Problem Description

This homework helps you understand the basic concepts in CUDA.

> *The sobel operator is used in image processing and computer vision, particularly within edge detection algorithms where it creates an image emphasising edges.*

In this homework, you are given the sequential (CPU) code of a 5x5 variant of the sobel operator, and asked to **parallelize it with CUDA**. Refer to the appendix for the information of the CPU version.

Input Format

The input file is a PNG image with 3 color channels: RGB.

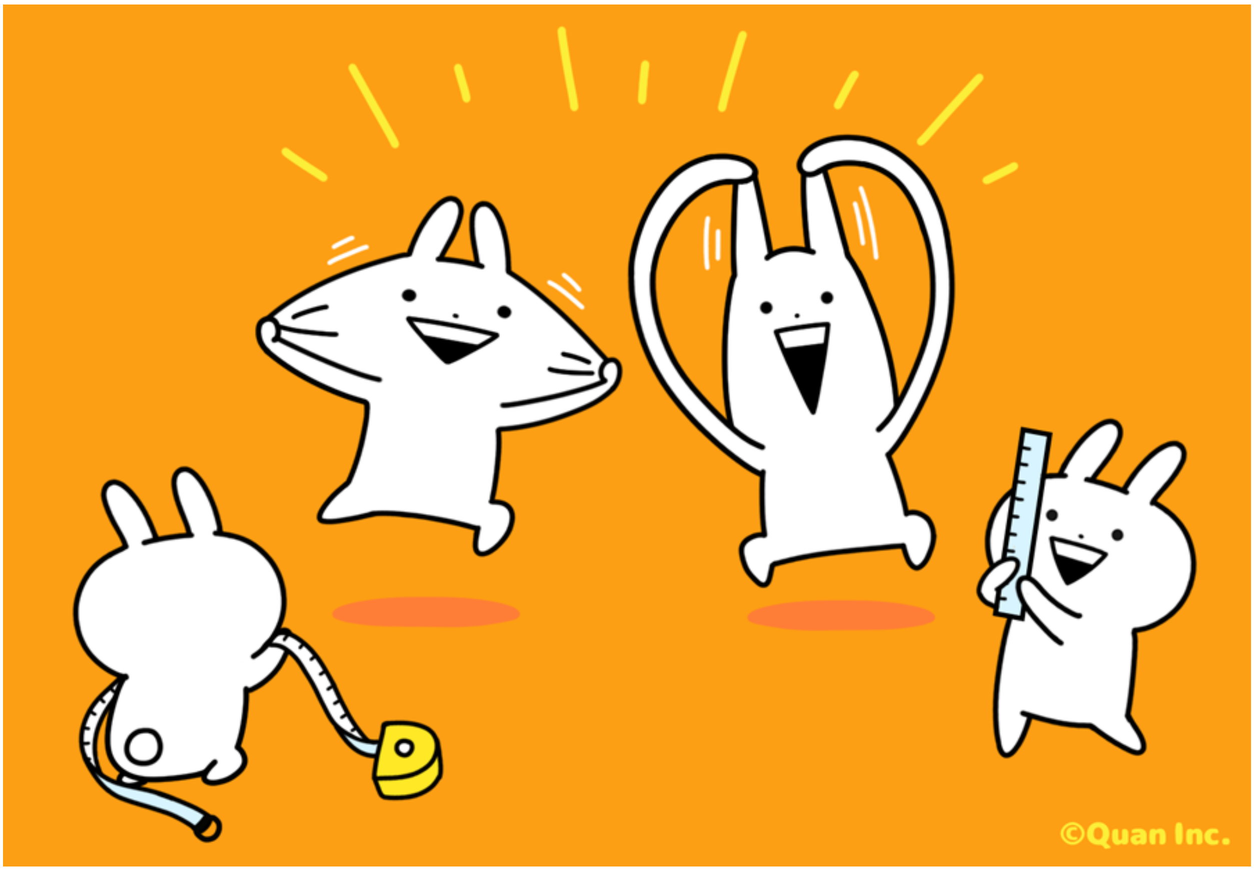
Output Format

The output file is a PNG image with 3 color channels: RGB.

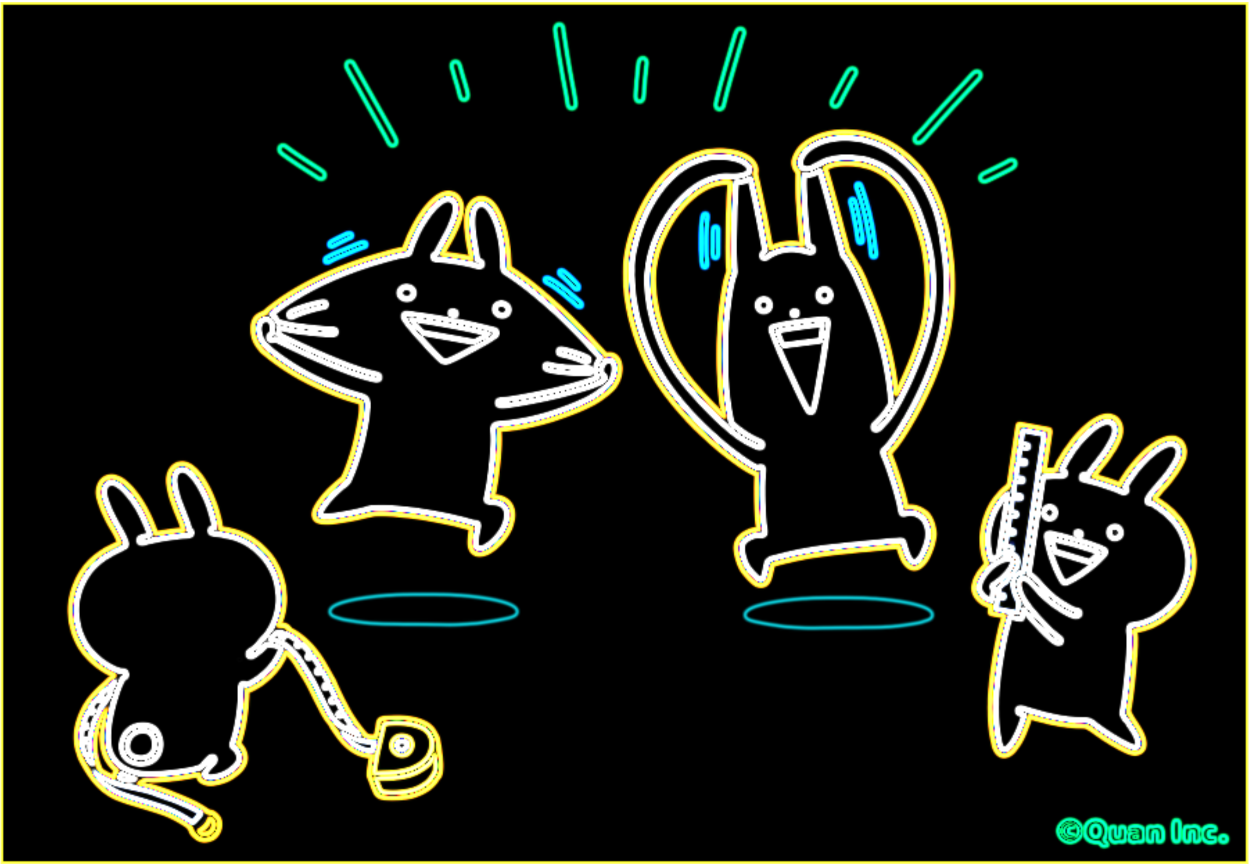
Your output is considered correct if at least 99.8% of the pixels are identical with the provided sequential version.

Your output is considered incorrect if the dimensions of the output image is incorrect.

Example Input



Example Output



Compilation

We use ninja to build your code. The default ninja file for this homework is provided at `/home/ipc20/ta/hw3/build.ninja`. If you wish to change the compilation flags, include `build.ninja` in your submission.

To use ninja to build your code, make sure `build.ninja` and `hw3.cu` is in the working directory, then run `ninja hw3` on the command line and it will build `hw3` for you. To remove the built files, run `ninja -t clean`.

We will compile your code with the following command:

```
ninja hw3
```

Execution

Your code will be executed with a command equalviant to:

```
srun -ppp --gres=gpu:1 ./hw3 input.png output.png
```

The time limit for each test case is 30 seconds.

Report

Answer the following questions, in either English or Traditional Chinese.

1. How did you parallelize the code?
    - Which CUDA APIs did you use?
    - Which functions are ported to a CUDA? How did you distribute the workload to blocks and threads?
2. Which optimization techniques did you apply to your code?
3. What's the difference between `cudaMalloc` and `cudaMallocManaged`? When will you pick one over another?
4. Pick any image that is not in the sample test cases, run your implementation with the image, and showcase both the input and output in your report.
5. (Optional) Any suggestions or feedback for the homework are welcome.

Submission

Upload these files to iLMS:

- `hw3.cu` – the source code of your implementation.
- `build.ninja` – optional. Submit this file if you want to change the build command.
- `report.pdf` – your report.

Please follow the naming listed above carefully. Failing to adhere to the names above will result to points deduction. Here are a few bad examples: `hw3.CU`, `HW3.cu`, `report.docx`, `report.pages`.

Grading

1. (40%) Correctness. Propotional to the number of test cases solved.
2. (30%) Performance. Based on the total time you solve all the test cases. For a failed test case, 75 seconds is added to your total time.
3. (30%) Report.

Appendix

Please note that this spec, the sample test cases and programs might contain bugs. If you spotted one and are unsure about it, please ask on iLMS 討論區!

Sequential (CPU) Version

The reference C++ implementation is at `/home/ipc20/ta/hw3/sobel.cc`. You can copy the code and `build.ninja` and run `ninja sobel`. The refernce code follows the same input/output format as your homework, and you can start implementing your version by copying it to `hw3.cu`.

Sample Testcases

The sample test cases are located at `/home/ipc20/ta/hw3/samples`.

Output validation

`/home/ipc20/ta/hw3/hw3-diff` can be used to compare two images.

For example, to compare your output with the answer, you may use:

```
/home/ipc20/ta/hw3/hw3-diff out.png /home/ipc20/ta/hw3/samples/c-1x.out.png
```

Judge

The `hw3-judge` command can be used to automatically judge your code against all sample test cases, it also submits your execution time to the scoreboard so you can compare your performance with others.

The scoreboard is here.

To use it, run `hw3-judge` in the directory that contains your code `hw3.cu`. It will automatically search for `build.ninja` and use it to compile your code, or fallback to the TA provided `/home/ipc20/ta/hw3/build.ninja` otherwise. If code compiliation is successful, it will then run all the sample test cases, show you the results as well as update the scoreboard.

> *Note:* `hw3-judge` *and the scoreboard has nothing to do with grading. Only the code submitted to iLMS is considered for grading purposes.*

Type `hw3-judge --help` to see a list of supported options.

Judge Verdict Table

Verdict	Explaination
internal error	there is a bug in the judge
time limited exceeded+	execution time > time limit + 10 seconds
time limited exceeded	execution time > time limit
runtime error	your program didn’t return 0 or is terminated by a signal
no output	your program did not produce an output file
wrong answer	your output is incorrect
accepted	you passed the test case