Homework 3 - Parallel Computing

- Course : **Operating Systems**
- Deadline : **2017/04/27 23:59**
- Demo : **2017/04/28 14:30 - 16:30**
- Contact : Che-Pin Chang (張哲彬), TA, jalex.cpc@gmail.com

Overview

After practicing the Beowulf cluster and MPICH in the last assignment, we are goning to get more examples in parallel computing. In this assignment, we would build an image processing service, which can manipulate assigned images in multi-processer computing .

Tutorials

Setup parallel computing environment

In this part, we are going to build a service environment, which can handle image files and execute a program in parallelism.

1. Create a 3 VMs as **master**, **slave1**, and **slave2**.
2. Install **the Beowulf cluster**. Please refer steps in part1 of **Homework 2**.
3. Install OpenCV. You can follow this link.
4. Validation

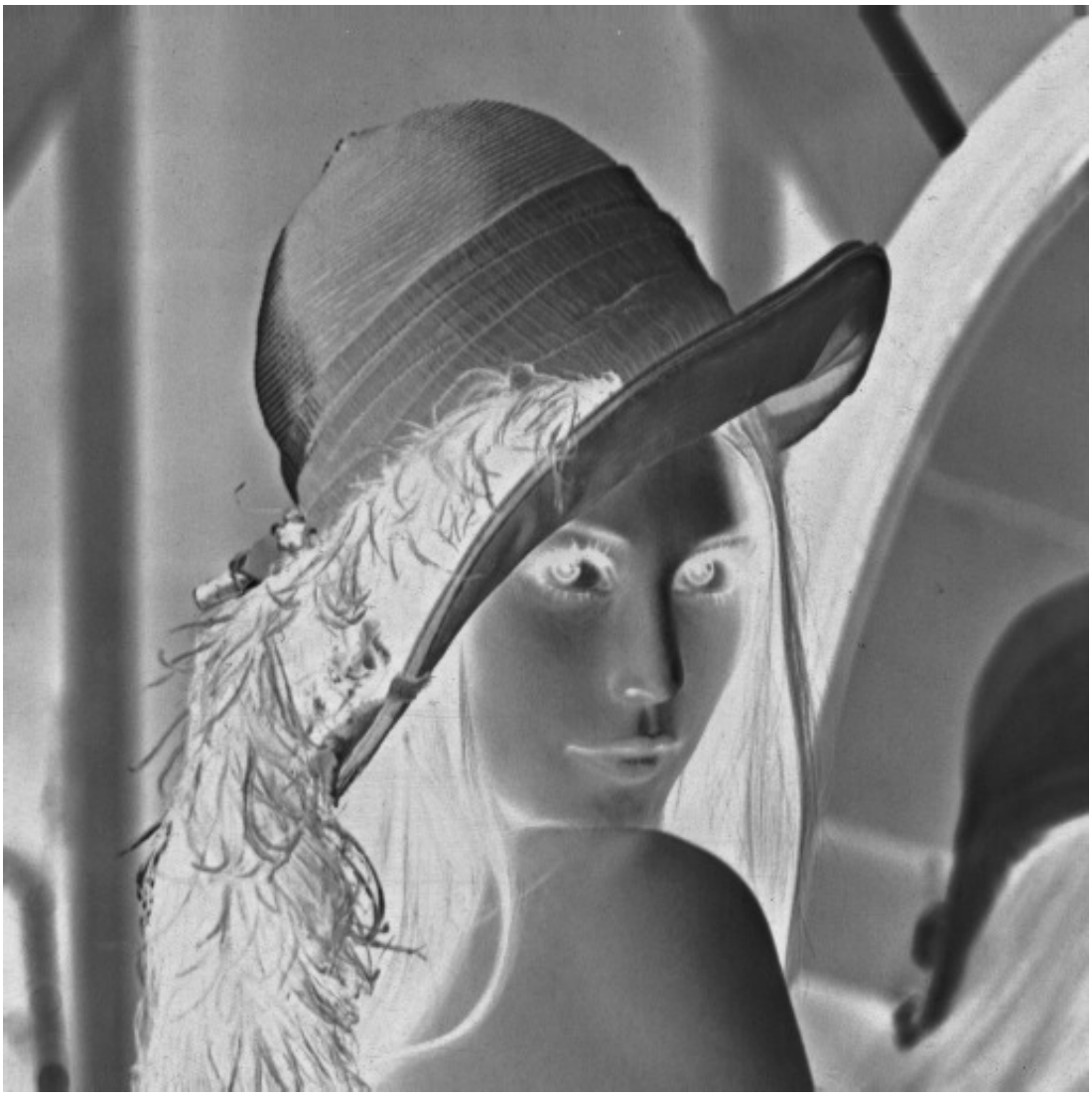
    1. Download attached source code, which includes a sample makec++ code, a makefile, and two images.
    2. Compile the code with the custom MPI C++ compiler.

        ```
        $ cd source
        $ make
        ```

    3. Run the program by mpirun.

        ```
        $ mpirun -n 1 -host master ./test lena_gray.jpg
        ```

    4. You would see a generated image file, named gray_negative.jpg. And it is an inverted image of the original one.



5. If you meet an error like "libopencv_imgcodecs.so.3.2: cannot open shared object file". Please add enviroment path **LD_LIBRARY_PATH** in the command.

```
$ mpirun -np 1 -host master -x LD_LIBRARY_PATH=/usr/local/lib/ ./test lena
_grey.jpg
```

Image Processing with grayscale

In this part, we are goning to write our own image processing functionalitices, which includes negative, rotation, and Gaussian blur.

1. **Negative** is a fuction that invert pixel value in grayscale. The attached smaple code is the negative function with a single thread. You could modify it to parrallel version with MPICH.
2. **Rotation** is a funtcion that make specific image rotate to 90° right.

3. **Gaussian blur** is a smooth function that blur a image by a particular filter matrix, which values are distributed as Gaussian distribution.

    1. Refer this material for more datails.
    2. For simplicity, you can directly use a $3x3$ filter matrix like :

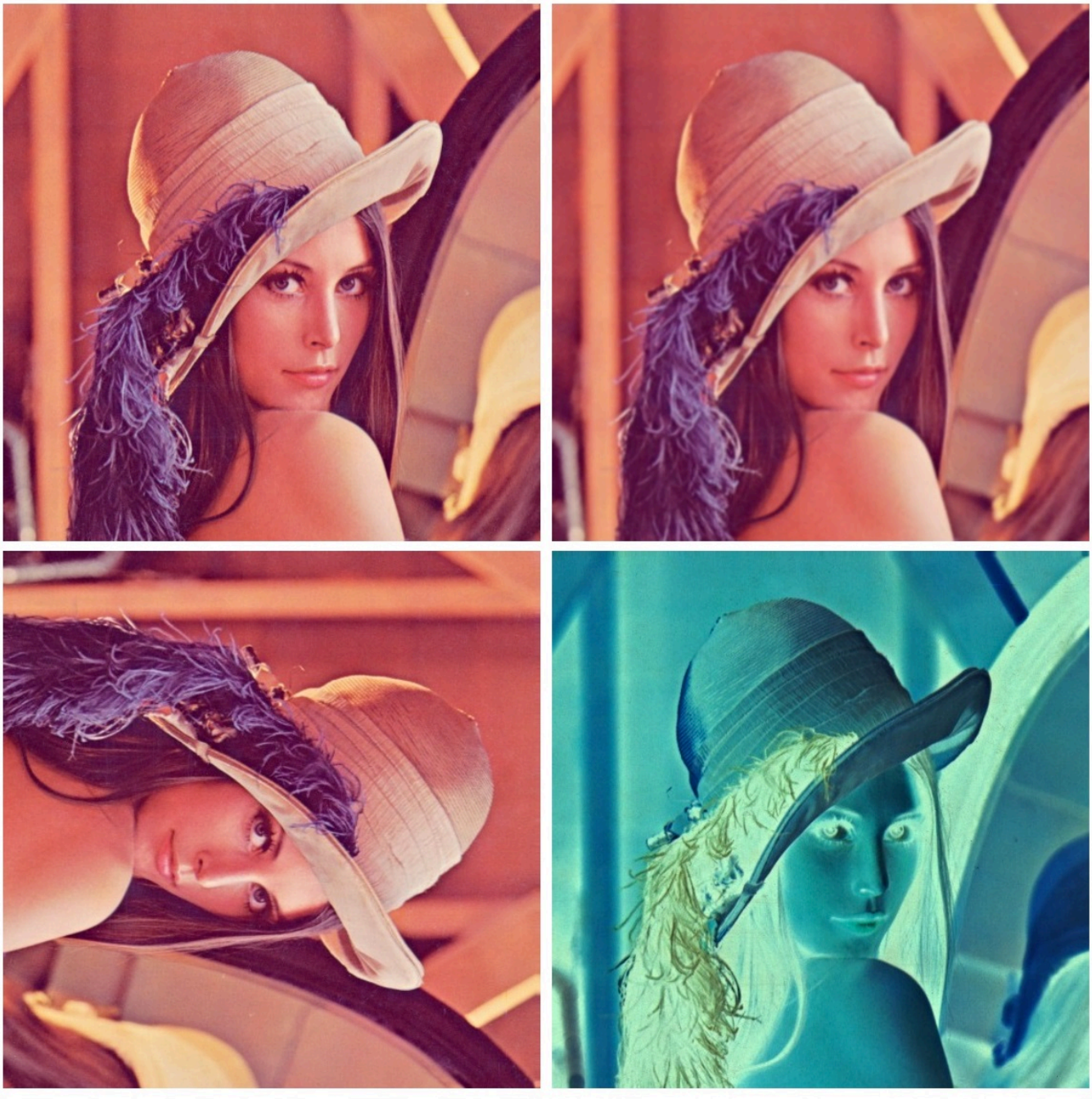
$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

4. In this part, you **can not** use built-in libraries in OpenCV to achieve above functions. But you can use them to validate your codes.

Image Processing with colorscale

This part is similar to part2. We add extra 3 functions to handle color images. Also, you **can not** use built-in libraries in OpenCV to achieve required functions.

1. Please use attached color image **lena.jpg** as sample input.



Web-based image processing service

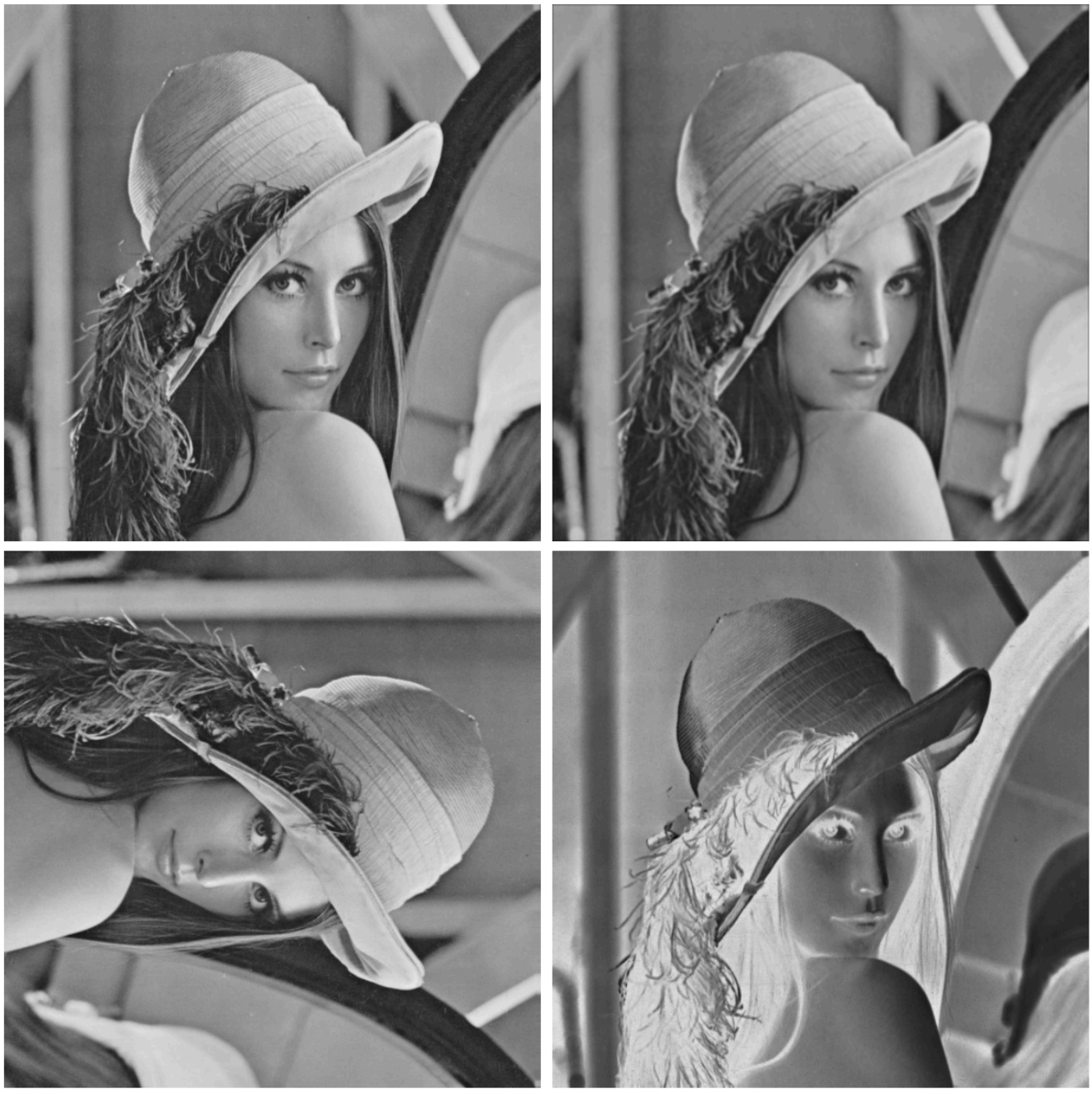
In this part, we are trying to build a web-based image processing service, which can upload a specific image and process it with different functions.

1. This service provides following features:
    1. It can upload a image file.
    2. It can choose scale's types, includes grayscale and colorscale.
    3. After pressing button "process", displays 4 image results, including 3 processed images and original one.
2. In Grayscale mode

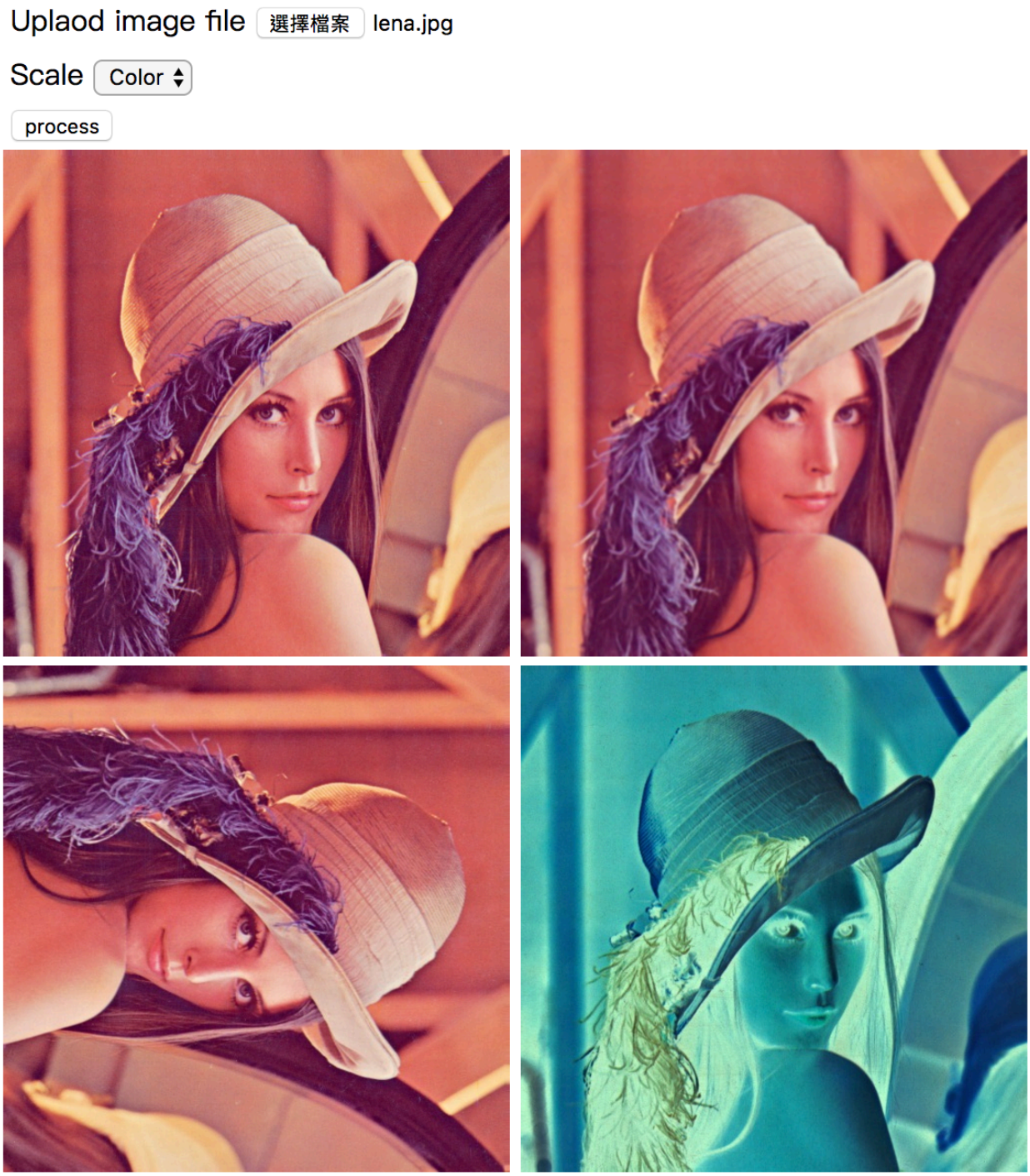
Uplaod image file 選擇檔案 lena_gray.jpg

Scale Gray

process



3. In Colorscale mode



Grades

In each part, we have prepared several oral questions to make sure that you finished the homework by yourself. If you meet the requirements and pass oral defense, you can get the following grades:

- Setup parallel computing environment **(30%)**
- Image Processing with grayscale
  - Functionalities **(15%)**
  - Parallelism **(15%)**

- Image Processing with colorscale
  - Functionalities **(15%)**
  - Parallelism **(15%)**
- Web-based image processing service **(10%)**

References

- MPI
  - https://en.wikipedia.org/wiki/Message*Passing*Interface#Example_program
  - http://mpitutorial.com/tutorials/mpi-send-and-receive/
  - https://www.mpich.org/static/docs/v3.2/www3/
- OpenCV
  - http://opencv.org/
  - http://docs.opencv.org/3.2.0/d7/d9f/tutorial*linux*install.html
- Image Processing
  - Gaussian blur
  - Negative image