Computer Vision HW1 Report

B04507009 電機四 何吉瑞

Code usage:

Conventional:

python3 rgb2gray.py --source\_path [the path of bgr image folder] --target\_path [the folder path to save grayscale images] --img\_name [the filename of .png image]

Advanced:

python3 adv\_rgb2gray.py --source\_path [the path of bgr image folder] --target\_path [the folder path to save grayscale images] --img\_name [the filename of .png image] --already\_done [0 for rerunning and 1 for testing] --use\_ans [1 for using opencv\_contrib] --r\_factor [the ratio of radius to sigma\_color] --save\_img [save all images]

The red part is used only if the user wants to rerun the code. For testing, it’s only required to adjust the black args.

Example command:

python3 --source\_path ../testdata --target\_path ../result --img\_name 0a.png

1. Conventional RGB2GRAY

After rerunning the code, a grayscale image will be generated in target\_path. I use Y = B + G + R for conversion. The reference is the function cvtColor() in OpenCV. After testing, the results of my conversion and that in OpenCV are same. An example is shown in Fig 1.





(a) (b) (c)

Fig 1. (a) Original BGR image. (b) Grayscale image from my conversion. (c) Grayscale image from the conversion of OpenCV.

2. Advanced RGB2GRAY

After rerunning the code, three folders will be generated. One is for grayscale candidate images, one is for the result of using candidate images as guided images for joint bilateral filter, and one is for the nine 2D surfaces for the cost of each selected parameter.

Tied ballot:

I generate all candidate images with tied ballot. For example, if the voted results are [4, 4, 3, 3, 2, …]. The code will generate four candidate images.

I check the filtered images with the result from function in OpenCV\_contrib. The pixel wise difference is at most 0.4, which I think is slight enough. The difference may because of the contour in OpenCV is a circle instead of a square.

Here we discuss all results.

0a

|  |  |
| --- | --- |
| Weight | votes |
| Y = 1.0B + 0.0G + 0.0R | 9 |
| Y = 0.0B + 0.0G + 1.0R | 9 |

The results are a little weird since the weight of green seems stronger than that of red. All the 2D surfaces are similar to Fig 2.

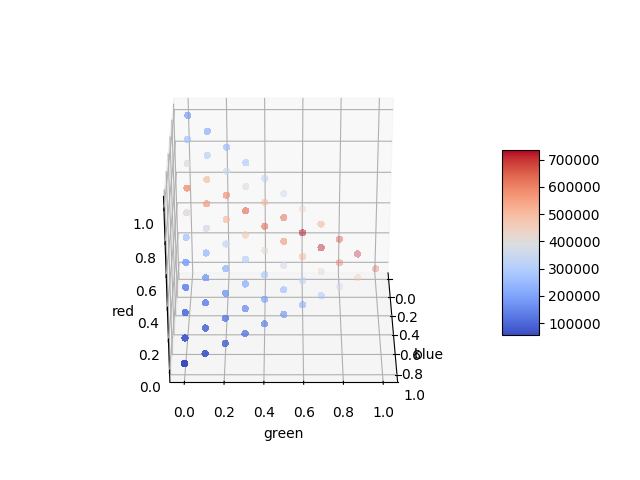


Fig 2. A 2D surface of candidates images for 0a.

Fig 3. The candidate images of 0a

0b

|  |  |
| --- | --- |
| Weight | votes |
| Y = 0.0B + 0.2G + 0.8R | 3 |
| Y = 0.0B + 0.0G + 1.0R | 3 |
| Y = 0.0B + 1.0G + 0.0R | 2 |
| Y = 0.1B + 0.2G + 0.7R | 2 |

Fig 4. The candidate images of 0b

The image is majorly composed of red and green, so the results are reasonable.

0c

|  |  |
| --- | --- |
| Weight | votes |
| Y = 0.3B + 0.4G + 0.3R | 3 |
| Y = 0.0B + 0.0G + 1.0R | 2 |
| Y = 0.3B + 0.5G + 0.2R | 1 |
| Y = 0.3B + 0.3G + 0.4R |
| Y = 0.2B + 0.6G + 0.2R |
| Y = 0.2B + 0.5G + 0.3R |
| Y = 0.1B + 0.5G + 0.4R |

5 candidate images of 0c have tied ballot, which is also out of my expectation. A possible reason is that the local minimum shifts a little under different parameters. For example, (0.3, 0.4, 0.3), (0.3, 0.5, 0.2), and (0.3, 0.3, 0.4) are close to each other.

Fig 5. The candidate images of 0b

