Computer Vision: from Recognition to Geometry HW1

R07942063 電信所碩一 郭冠軒

Input Images

0a.png, 0b.png, 0c.png

Toolkit Requirements

imageio (https://pypi.org/project/imageio/)

Execution Command

python3 hw1_r07942063.py –d < testdata directory > -o < output image directory >

[Description]

The program requires the path of the “testdata” directory provided in this homework, and the output images will be generated in the path of the output image directory, which should also be given by the user.

If the path mentioned above is not given, the program will use the default arguments.

[Default Arguments]

< test data directory > ./testdata/

< output image directory > ./output/

[Example command]

python3 hw1_r07942063.py –d ./testdata/ -o ./output/

Implementation Details

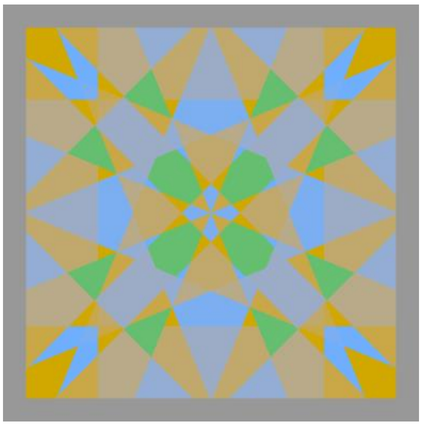
My local minima selection strategy is as follows:

First, I use the original image as the input image I and 66 grayscale images g^i as guide images to apply joint bilateral filtering. For a pair of input image and guide image, I create two windows with the size $(6\sigma_s + 1, 6\sigma_s + 1)$ to perform convolution on the input image, one window refers to G_r, and the other refers to G_s. During the convolution process, I do not pad on the border of the input image, so the size of the filtered images will be $(H - 6\sigma_s,\ W - 6\sigma_s)$, where H and W are the height and the width of the input image, respectively.

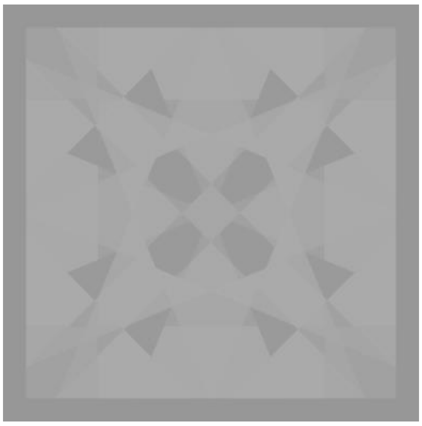
Then, we may use the bilateral filtered image I^I and the joint bilateral filtered images I^{g^i} to calculate the matching loss. I use a dictionary to store each weight combination and the corresponding matching loss. The key of the dictionary is the tuple of the grayscale weight combinations, and the value is the matching loss. After recording every pair of weight and loss, I check the loss of the neighbors of each weight to determine whether the weight is a local minimum.

Output Images

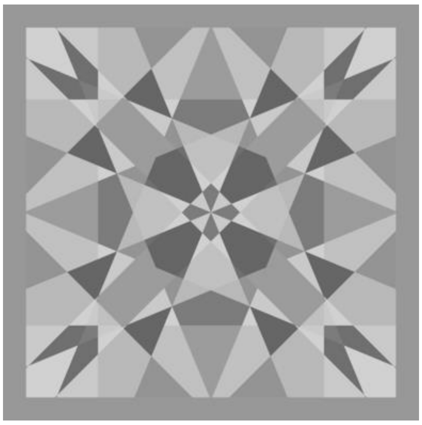
Original Image
(0a.png)



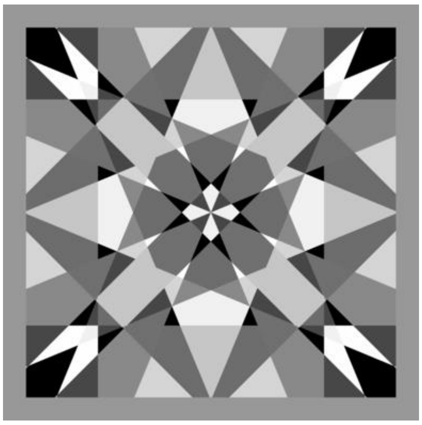
Conventional Color-to-Gray Conversion
[R, G, B] = [0.299, 0.587, 0.114]



[R, G, B] = [1.0, 0.0, 0.0]



[R, G, B] = [0.0, 0.0, 1.0]



Original Image
(0b.png)

Conventional Color-to-Gray Conversion
[R, G, B] = [0.299, 0.587, 0.114]

[R, G, B] = [1.0, 0.0, 0.0]

[R, G, B] = [0.8, 0.2, 0.0]

[R, G, B] = [0.7, 0.2, 0.1]

Original Image
(0c.png)

The luminance generated by a physical device is generally not a linear function of the applied signal. A conventional CRT has a power-law response to voltage: luminance produced at the face of the display is approximately proportional to the applied voltage raised to the 2.5 power. The numerical value of the exponent of this power function is colloquially known as *gamma*. This nonlinearity must be compensated in order to achieve correct reproduction of luminance.

As mentioned above (*What is lightness?*), human vision has a nonuniform perceptual response to luminance. If luminance is to be coded into a small number of steps, say 256, then in order for the most effective perceptual

Conventional Color-to-Gray Conversion
[R, G, B] = [0.299, 0.587, 0.114]

The luminance generated by a physical device is generally not a linear function of the applied signal. A conventional CRT has a power-law response to voltage: luminance produced at the face of the display is approximately proportional to the applied voltage raised to the 2.5 power. The numerical value of the exponent of this power function is colloquially known as *gamma*. This nonlinearity must be compensated in order to achieve correct reproduction of luminance.

As mentioned above (*What is lightness?*), human vision has a nonuniform perceptual response to luminance. If luminance is to be coded into a small number of steps, say 256, then in order for the most effective perceptual

[R, G, B] = [0.3, 0.4, 0.3]

The luminance generated by a physical device is generally not a linear function of the applied signal. A conventional CRT has a power-law response to voltage: luminance produced at the face of the display is approximately proportional to the applied voltage raised to the 2.5 power. The numerical value of the exponent of this power function is colloquially known as *gamma*. This nonlinearity must be compensated in order to achieve correct reproduction of luminance.

As mentioned above (*What is lightness?*), human vision has a nonuniform perceptual response to luminance. If luminance is to be coded into a small number of steps, say 256, then in order for the most effective perceptual

[R, G, B] = [1.0, 0.0, 0.0]

The luminance generated by a physical device is generally not a linear function of the applied signal. A conventional CRT has a power-law response to voltage: luminance produced at the face of the display is approximately proportional to the applied voltage raised to the 2.5 power. The numerical value of the exponent of this power function is colloquially known as *gamma*. This nonlinearity must be compensated in order to achieve correct reproduction of luminance.

As mentioned above (*What is lightness?*), human vision has a nonuniform perceptual response to luminance. If luminance is to be coded into a small number of steps, say 256, then in order for the most effective perceptual

[R, G, B] = [0.4, 0.5, 0.1]

The luminance generated by a physical device is generally not a linear function of the applied signal. A conventional CRT has a power-law response to voltage: luminance produced at the face of the display is approximately proportional to the applied voltage raised to the 2.5 power. The numerical value of the exponent of this power function is colloquially known as *gamma*. This nonlinearity must be compensated in order to achieve correct reproduction of luminance.

As mentioned above (*What is lightness?*), human vision has a nonuniform perceptual response to luminance. If luminance is to be coded into a small number of steps, say 256, then in order for the most effective perceptual