

# Assignment 1:

## Advanced Color-to-Gray Conversion

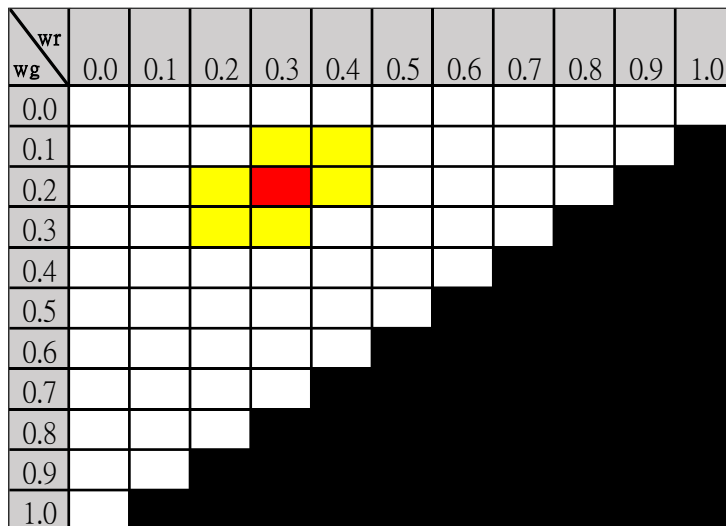
系級: 電機四  
姓名: 林棋祥  
學號: b04505004

- 輸入圖片: 1a.png, 1b.png, 1c.png
- main.py 主要包含了以下幾個作業要求的函數:

Function Name	Description	Parameters		Returns	
rgb2gray	conventional rgb2gray	I	Input Image: RGB image array	I_gray	Grayscale image of type 'float'
		wr	weight for red		
		wg	weight for green		
		wb	weight for blue		
JBF	joint bilateral filter	I	Input Image: RGB image array with pixel values between 0~1	I_filtered	Filtered image of type 'float' with pixel values between 0~1
		G	Guide Image: Grayscale/RGB image array with pixel values between 0~1		
		k	kernel size		
		sigma_s	sigma for spatial kernel		
		sigma_r	sigma for range kernel		
rgb2gray_X	advanced rgb2gray	I	Input Image: RGB image array with pixel values between 0~255	I_gray_x	List of top 3 voted grayscale images with pixel values between 0~255


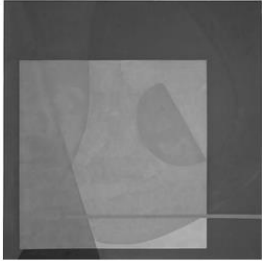

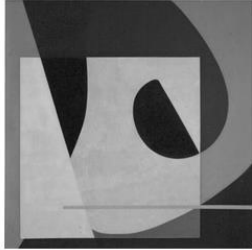











Note: 此次作業用到了 skimage 及 numpy 套件，煩請事先安裝。

- 要確認某個點是否為 local minima，需要在  $w_r + w_g + w_b = 1$  的平面上與相鄰的六個點比對數值大小，為此，我先將所有的點計算在  $w_r$ ,  $w_g$  的平面上，如下圖所示:



若想確認  $(w_r, w_g, w_b) = (0.2, 0.3, 0.5)$  是否為 local minima，只需要比對  $(0.2, 0.3)$  (意即圖上的紅色方格) 上下左右、右上和左下共六個點 (意即圖上的黃色方格) 即可。不難理解在  $w_r, w_g$  平面上的這六個點就是在  $w_r + w_g + w_b = 1$  平面上與  $(0.2, 0.3, 0.5)$  相鄰的六個點。此方法也可以透過 numpy 運算而達成，因此不會耗費太多運算時間。

4. 輸出結果:  
(wr, wg, wb)

Image Name	Input	Conventional	Advanced 1	Advanced 2	Advanced 3
1a.png			 (1, 0, 0) votes=7	 (0, 0, 1) votes=6	 (0, 0.2, 0.8) votes=3
1b.png			 (0, 0.6, 0.4) votes=3	 (0.1, 0.5, 0.4) votes=3	 (0.7, 0.1, 0.2) votes=2
1c.png			 (0.6, 0.4, 0) votes=4	 (0.8, 0.2, 0) votes=2	 (0.5, 0.4, 0.1) votes=1

為求排版圖片經過縮放。在 Output images 資料夾中有清晰的原圖。

5. 心得:

這次的作業結果看起來滿成功的。然而在時間上，我已盡量用 numpy 的矩陣運算，跑一張 joint bilateral filter 的時間仍需要 6 秒。也許當初應該用 Pytorch 實現，就可以用 GPU 加速了。