**Computer Vision HW1 Report**

Student ID: R11921041

Name: 蔣沅均

**Part 1.**

* **Visualize the DoG images of 1.png.**

|  |  |  |  |
| --- | --- | --- | --- |
|  | DoG Image (threshold = 3) |  | DoG Image (threshold = 3) |
| DoG1-1.png |  | DoG2-1.png |  |
| DoG1-2.png |  | DoG2-2.png |  |
| DoG1-3.png |  | DoG2-3.png |  |
| DoG1-4.png |  | DoG2-4.png |  |

* **Use three thresholds (1,2,3) on 2.png and describe the difference.**

|  |  |
| --- | --- |
| Threshold | Image with detected keypoints on 2.png |
| 1 |  |
| 2 |  |
| 3 |  |

(describe the difference)

Threshold低的時候，找到的keypoints比較多，但可能比較不精準；Threshold高的時，只能找到較少的keypoints，可能會遺漏某些重要的keypints，因此尋找適合的threshold對於後續應用很重要。

**Part 2.**

* **Report the cost for each filtered image.**

|  |  |
| --- | --- |
| Gray Scale Setting | Cost (1.png) |
| cv2.COLOR\_BGR2GRAY | 1207799 |
| R\*0.0+G\*0.0+B\*1.0 | 1439568 |
| R\*0.0+G\*1.0+B\*0.0 | 1305961 |
| R\*0.1+G\*0.0+B\*0.9 | 1390952 |
| R\*0.1+G\*0.4+B\*0.5 | 1278834 |
| R\*0.8+G\*0.2+B\*0.0 | 1127294 |

|  |  |
| --- | --- |
| Gray Scale Setting | Cost (2.png) |
| cv2.COLOR\_BGR2GRAY | 183850 |
| R\*0.1+G\*0.0+B\*0.9 | 77797 |
| R\*0.2+G\*0.0+B\*0.8 | 85864 |
| R\*0.2+G\*0.8+B\*0.0 | 188065 |
| R\*0.4+G\*0.0+B\*0.6 | 128150 |
| R\*1.0+G\*0.0+B\*0.0 | 110862 |

* **Show original RGB image / two filtered RGB images and two grayscale images with highest and lowest cost.**

|  |  |  |
| --- | --- | --- |
| Original RGB image (1.png) | Filtered RGB image and Grayscale image of  Highest cost | Filtered RGB image and Grayscale image of  Lowest cost |
|  |  |  |
|  |  |  |

(Describe the difference between those two grayscale images)

Cost較高的圖片，整體都偏暗，較難區分出紅葉與背景綠地。

Cost較低的圖片則很明顯可以看出紅葉跟綠地分屬深淺，且在綠地中不是綠色的部分也有被凸顯出來。

|  |  |  |
| --- | --- | --- |
| Original RGB image (2.png) | Filtered RGB image and Grayscale image of  Highest cost | Filtered RGB image and Grayscale image of  Lowest cost |
|  |  |  |
|  |  |  |

(Describe the difference between those two grayscale images)

Cost較高的圖片，整體對比度較低。而cost較低的圖片中，不同色塊間的邊緣分界非常明顯。

* **Describe how to speed up the implementation of bilateral filter.**

因為每個window的spatial kernel都是一樣的，因此可以在迴圈外預先算好，這樣就不會重複計算。

另外每個pixel在個別計算值的時候，window內的計算可以利用vectorization減少迴圈的使用。