**Computer Vision HW1 Report**

Student ID: B10705034

Name: 許文鑫

**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)

當threshold低的時候，可以看到除了貓咪邊界有key points以外，貓咪身上的花紋以及鬍鬚上也有，但是當threshold調高後，邊界以外的區域key points都減少了。

**Part 2.**

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

|  |  |
| --- | --- |
| Gray Scale Setting | Cost (1.png) |
| cv2.COLOR\_BGR2GRAY | 1207800 |
| R\*0.0+G\*0.0+B\*1.0 | 1439568 |
| R\*0.0+G\*1.0+B\*0.0 | 1305962 |
| R\*0.1+G\*0.0+B\*0.9 | 1386155 |
| R\*0.1+G\*0.4+B\*0.5 | 1277423 |
| R\*0.8+G\*0.2+B\*0.0 | 1127895 |

|  |  |
| --- | --- |
| Gray Scale Setting | Cost (2.png) |
| cv2.COLOR\_BGR2GRAY | 183852 |
| R\*0.1+G\*0.0+B\*0.9 | 78490 |
| R\*0.2+G\*0.0+B\*0.8 | 86425 |
| R\*0.2+G\*0.8+B\*0.0 | 187521 |
| R\*0.4+G\*0.0+B\*0.6 | 128826 |
| R\*1.0+G\*0.0+B\*0.0 | 110861 |

* **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)

在low cost的灰階圖中，紅色葉子和草地的intensity的差距比較明顯，而high cost的灰階圖中，葉子和草地的差距不大，葉子就比較不明顯。

|  |  |  |
| --- | --- | --- |
| 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)

Low cost的灰階圖可以比較明顯的看出原圖各個顏色的邊界，high cost的就很不明顯。

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

因為計算的過程會將兩個kernel相乘，所以我把exp(a) \* exp(b) 的部分都用 exp(a + b)處理，不過也只有快一點點。