**高等影像處理**

**作業#(4)**

姓名：梁仔弘

|  |  |
| --- | --- |
| 1.a | |
| **Figures** | |
| 1\_a\_cat\_hist.raw | |
| 1\_a\_cat\_catch.raw | 1\_a\_cat\_rain.raw |

|  |  |
| --- | --- |
| 1\_a\_cat\_catch\_hist.raw | 1\_a\_cat\_rain\_hist.raw |
| Discussion | |
| 原圖的histogram就已經非常好看了，而且也和rain圖的histogram很像，所以做出來的結果還是很好。至於catch這張圖片，可以看到背景幾乎全黑，所以match完的histogram偏暗做出來的圖片也就偏暗了。  最後，可以明顯看到，match rain這張圖片的效果比match catch這張圖片好很多。 | |

|  |  |
| --- | --- |
| 1.b | |
| **Figures** | |
| 1\_b\_lheRainA.raw    3 x 3 | 1\_b\_aHist.raw |
| 1\_b\_lheRainB.raw    21 x 21 | 1\_b\_bHist.raw |
| 1\_b\_lheRainC.raw    100 x 100 | 1\_b\_cHist.raw |
| Discussion | |
| 從最小的mask size來看，可以發現local histogram equalization可以把非常細節的邊框給抓出來，因為在3x3的小視窗內，只要稍微亮一點的值，就會因為equalization而被拉到非常後面(亮)，所以就會造成細節的邊界也被明顯的放大。而隨著window size越來越大，出來的效果也就會越來越像原圖。  其中最有趣的一點是21x21的那張圖，可以看到連雨傘上的流水都被清楚的框出來，這是global equalization所做不到的。 | |

|  |
| --- |
| 2.a |
| **Figures** |
| 2\_a\_blurr.raw |
| 2\_a\_roi.raw |
| Discussion |
| 我的高斯函數~(30, 0)，而window size=27x27，這是我一路從標準差=100往下調整過後最佳的結果了，因為若標準差或window size太小，模糊效果不佳，roi圈的不好；反之則會太過模糊，容易圈到太多的地方。而threshold我原本是從0.5(灰階值/255)開始，但發現圈到太多地方，所以將其設的高一些。 |

|  |
| --- |
| 2.b |
| **Figures** |
| 2\_b.raw |
| Discussion |
| 其實套用ROI遮罩最直觀的實施方法應該是利用bit-operation中的and或or來執行，不過我對這個方法不是很熟悉，所以用了條件式的方式來做。  if (roi[idx] == 255) { outputImg[idx] = inputImg[idx]; }  else { outputImg[idx] = blurredImg[idx]; } |

|  |  |  |
| --- | --- | --- |
| 3.a | | |
| **Figures** | | |
|  | Laplacian | Sobel |
| No Noise | 3\_a\_lapRoof.raw | 3\_a\_sobRoof.raw |
| With Noise | 3\_a\_lapRoofNoise.raw | 3\_a\_sobRoofNoise.raw |
| **Discussion** | | |
| **雜訊敏感度**：可以發現，Laplacian套上雜訊以後變得非常雜亂，沒有原圖那樣乾淨俐落；而Sobel則是在套上雜訊圖後，仍然可以看出明顯的輪廓，因此Sobel在抵抗雜訊上會比Laplacian好很多。  對於跨出邊界的pixel，我直接複製他隔壁的值來做處理，比如說window的右邊那個column跑出去了，我就把跑出去的值設定為他原本的位置往左移一格，如果還是沒有，那就繼續往左移，直到有值為止。  Sobel有兩個方法將合併，我使用的方法是平方開根號，因為這個比絕對值相加更精準。 | | |

|  |  |
| --- | --- |
| 3.b | |
| **Figures** | |
| 3\_b\_1.raw | 3\_b\_2.raw |
| 3\_b\_3.raw | 3\_b\_4.raw (Answer) |

|  |
| --- |
| **Discussion** |
| 流程圖    高斯模糊：為了把後面的部分細節模糊掉而使用  Laplacian：取邊框，不用Sobel是因為Sobel對於背景太過於敏感了，會把不必要的資訊都框出來。  相加：由於用模糊後取邊框，邊框不如原本的明顯，所以才會希望相加原圖來加強他的邊框，但又因為不想要太多背景被加入，所以乘上0.1。  Thresholding：其實我一開始沒想到用thresholding，是看著題目上的答案，發覺他的框非常的白才想到。 |