Final Project

第二組： 葉雨婷 110368404 \ 電子碩二

顏郁芩 110368155 \ 電子碩二

謝瑞榆 110618053 \ 自動碩二

目錄

問題分析與解決思路

問題分析

- **目標** 對一組可能存在雜訊、亮度不均和經過濾鏡處理的圖像進行圖像修復與重排序，並展示結果
- **存在挑戰**

I. 圖像修復：圖像組圖像可能遭受損壞（雜訊、亮度不均和經過濾鏡處理）

II. 圖像排序：找到衡量圖像間的距離的方法

解決思路

- 即時讀取測試圖像目錄
- 視頻幀調亮度並去雜訊
- 視頻幀排序，輸出TXT檔
- 計算SRCC與MSE_Average
- 最終圖像集生成mp4
- 輸出計算程式運行時間

實現功能

即時讀取測試圖像目錄

目的：獲取數據集目錄和正確排序文件路徑

所用框架：

（1）shlobj頭文件中的SHBrowseForFolder函數

功能：調用Shell，顯示一個對話框，使用戶能選擇Shell文件夾

（2）shlobj頭文件中的SHGetPathFromIDList函數

功能：將選擇的文件夾的項標識符列表轉換為文件系統路徑

參考資料

https://learn.microsoft.com/zh-cn/windows/win32/api/shlobj_core/

https://www.cnblogs.com/qingtian224/p/5566901.html

計算SRCC與MSE_Average

SRCC

- Spearman's rank correlation coefficient，取值範圍為[-1, 1]，值為1則是完全正相關，為-1則是完全負相關
- 計算公式

$$SRCC = \left| 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \right|$$

- d_i 的計算方法：
  (1) Rank距離
  (2) 索引距離

計算SRCC與MSE_Average

Rank距離計算方法

- Rank距離

Xi	Yi
106	7
86	0
100	27
101	50
99	28
103	29
97	20
113	12
112	6
110	17

Xi	Yi	xi	yi	di	di^2
86	0	1	1	0	0
97	20	2	6	-4	16
99	28	3	8	-5	25
100	27	4	7	-3	9
101	50	5	10	-5	25
103	29	6	9	-3	9
106	7	7	3	4	16
110	17	8	5	3	9
112	6	9	2	7	49
113	12	10	4	6	36

參考資料：https://zhuanlan.zhihu.com/p/339079547

計算SRCC- 索引距離計算方法

- 索引距離

ID	RANK1	RANK2	D	D^2
1	10	10	0	0
2	9	7	5	25
3	8	6	2	4
4	7	2	-2	4
5	6	8	-2	4
6	5	4	3	9
7	4	9	1	1
8	3	3	0	0
9	2	5	4	16
10	1	1	0	0

計算MSE_Average

$$\mathrm{MSE}_{Average} = \frac{1}{N}\sum_{x=1}^{N-1}\frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I_x(i,j) - K_x(i,j)]^2$$

最終圖像集生成mp4

目的：將重排序的圖片集轉成MP4格式

所用框架：opencv中的VideoWriter

（1）功能：設置參數後，將讀取的圖像寫入視頻文件中

（2）參數設置

fps：2, 表示每秒的幀數

size：與圖像大小保持一致

color：TRUE，彩色幀

編碼：CAP_OPENCV_MJPEG

輸出計算程式運行時間

目的：記錄程序運行時間

程序運行開始記錄，最後所有功能截止，最後以時分秒形式輸出

視頻幀亮度調整和去噪

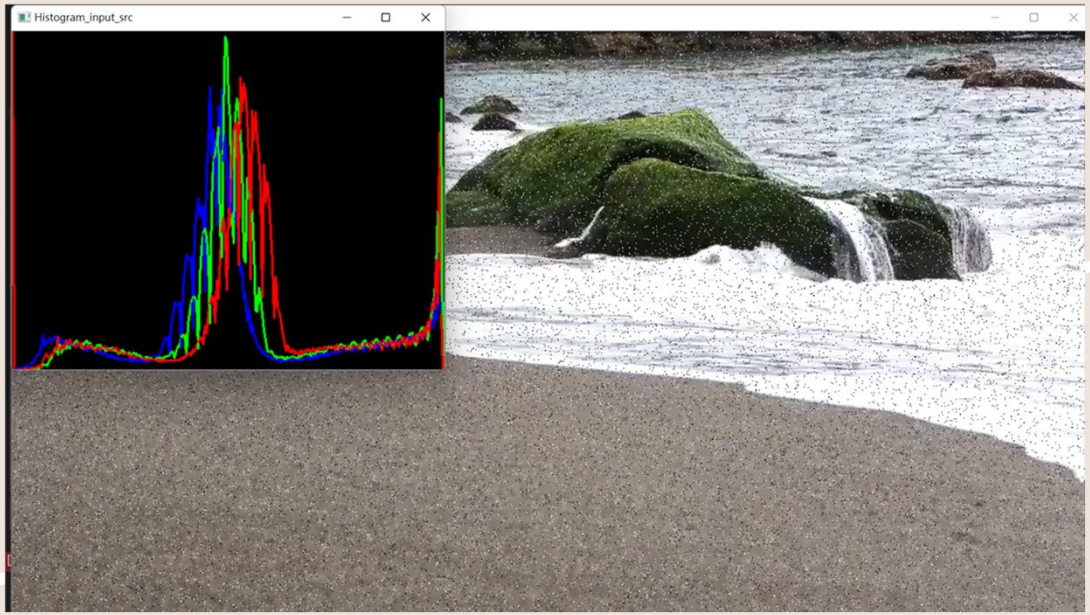
椒鹽&
彩色雜訊

亮度不均

濾鏡

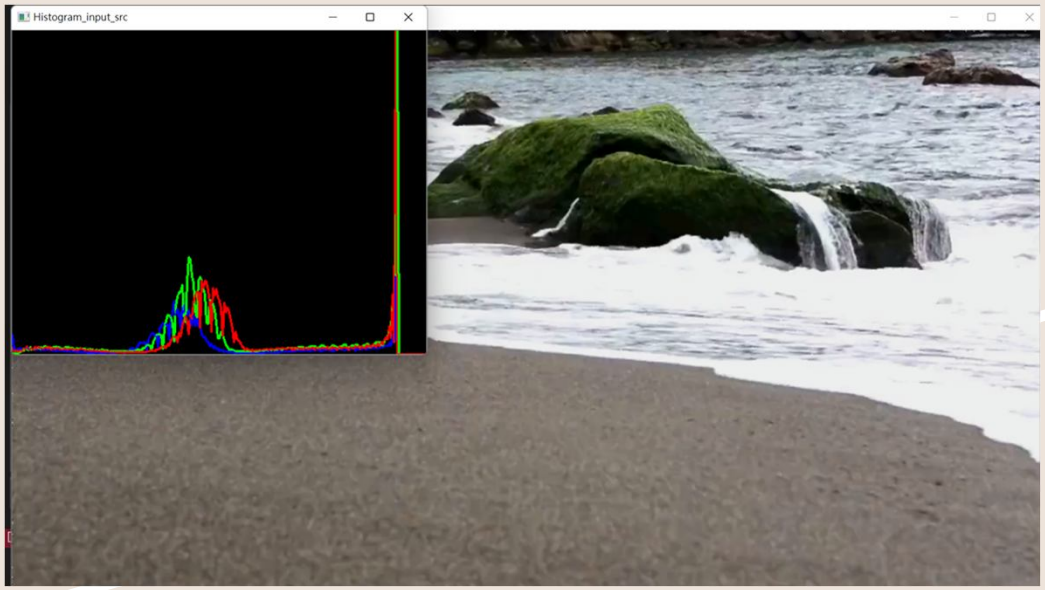
去除雜訊

判斷: 判斷B通道的最大最小值，如果pixel在值等於0的時候出現最多次，就執行去噪音處理



去除雜訊

處理:中值濾波



去除雜訊

Open source code:

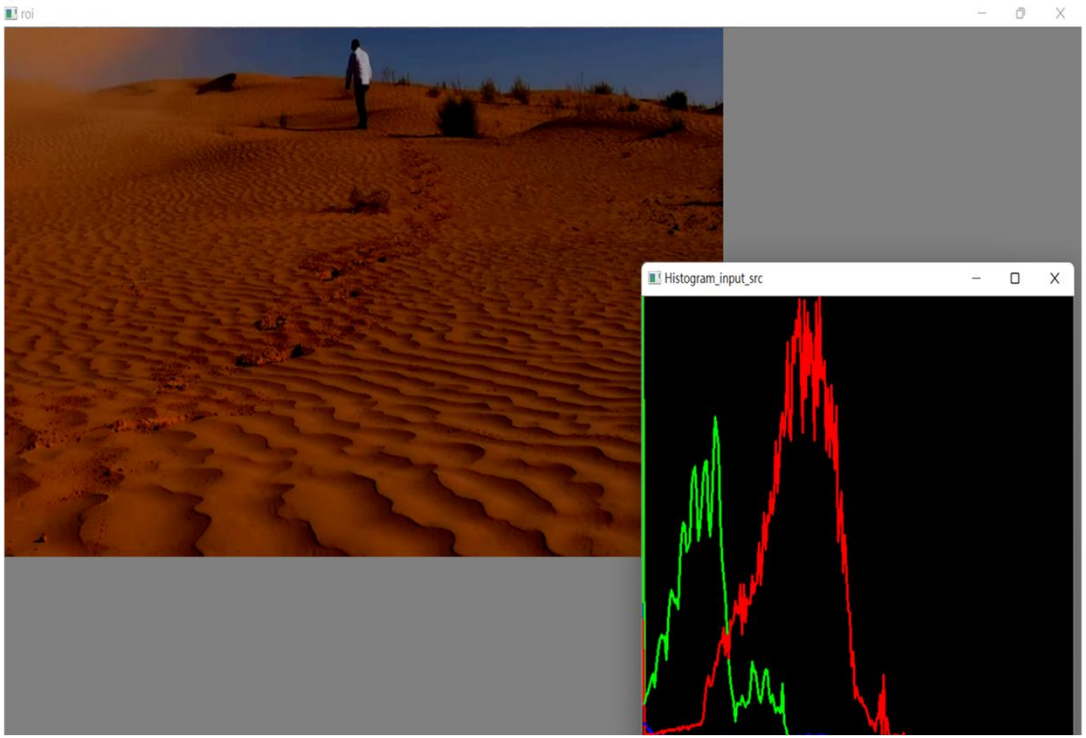
```
Mat b_hist, g_hist, r_hist;
calcHist( &bgr_planes[0], 1, 0, Mat(), b_hist, 1, &histSize, histRange, uniform, accumulate );
calcHist( &bgr_planes[1], 1, 0, Mat(), g_hist, 1, &histSize, histRange, uniform, accumulate );
calcHist( &bgr_planes[2], 1, 0, Mat(), r_hist, 1, &histSize, histRange, uniform, accumulate );
```

OpenCV Website link : OpenCV: Histogram Calculation

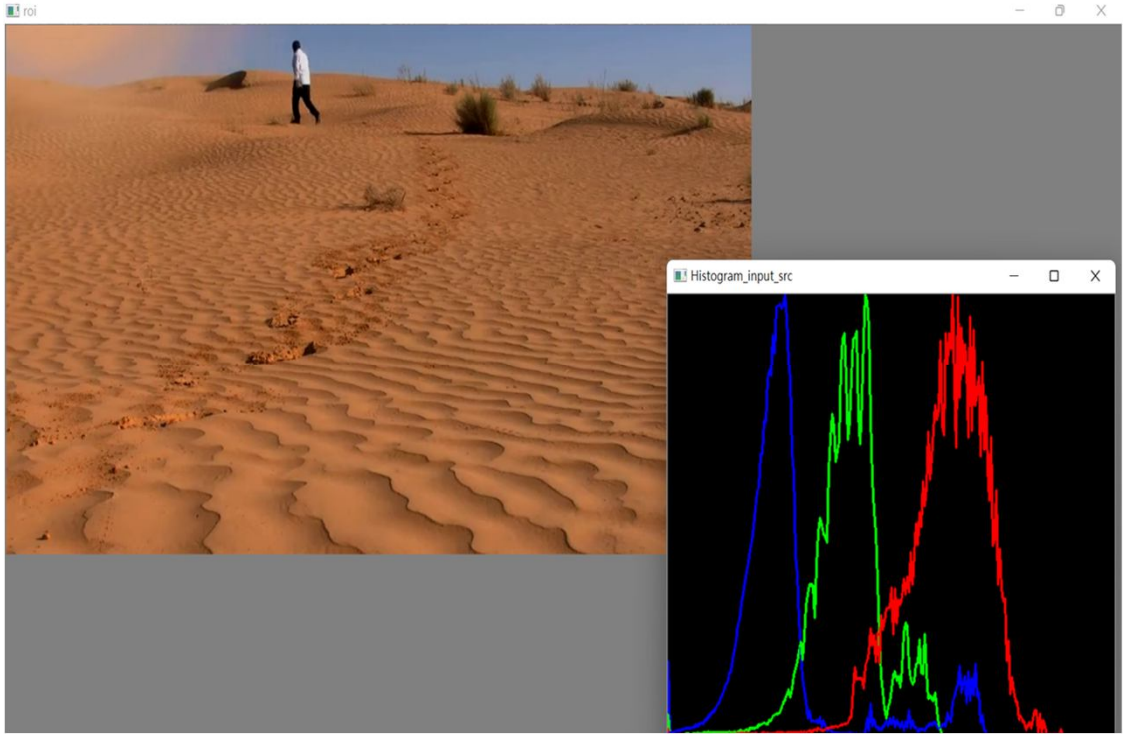
```
normalize(b_hist, b_hist, 0, histImage.rows, NORM_MINMAX, -1);
normalize(g_hist, g_hist, 0, histImage.rows, NORM_MINMAX, -1);
normalize(r_hist, r_hist, 0, histImage.rows, NORM_MINMAX, -1);
```

OpenCV Website link : OpenCV: Operations on arrays

亮度不均

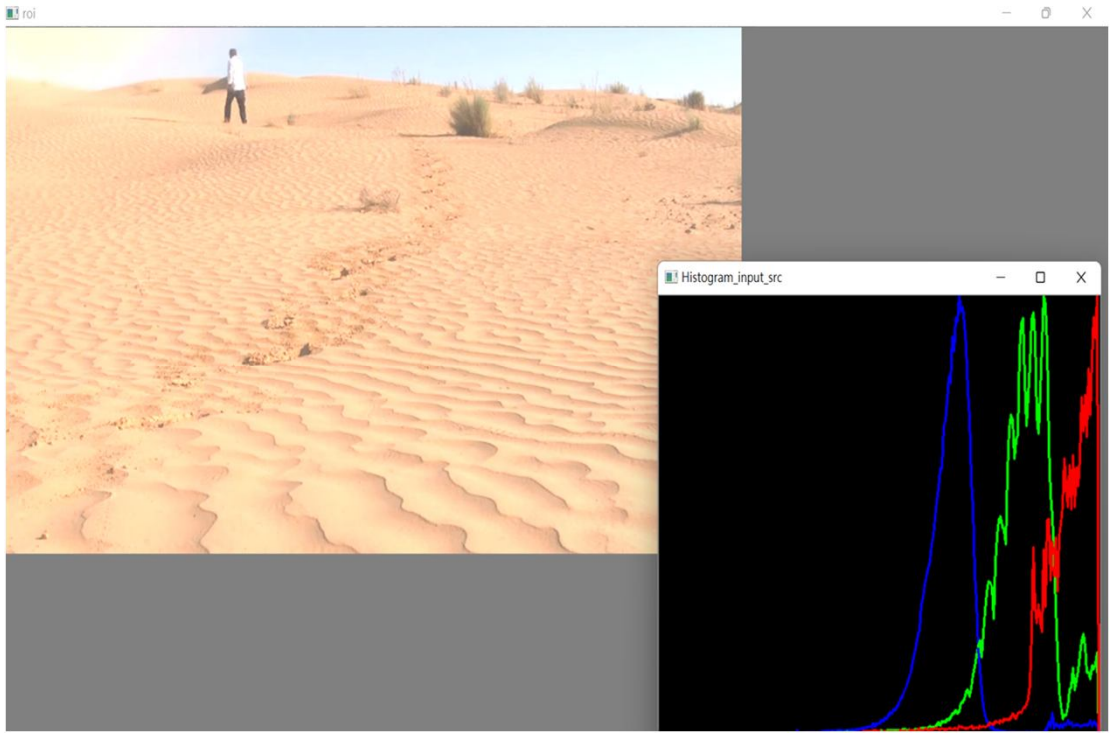


過暗

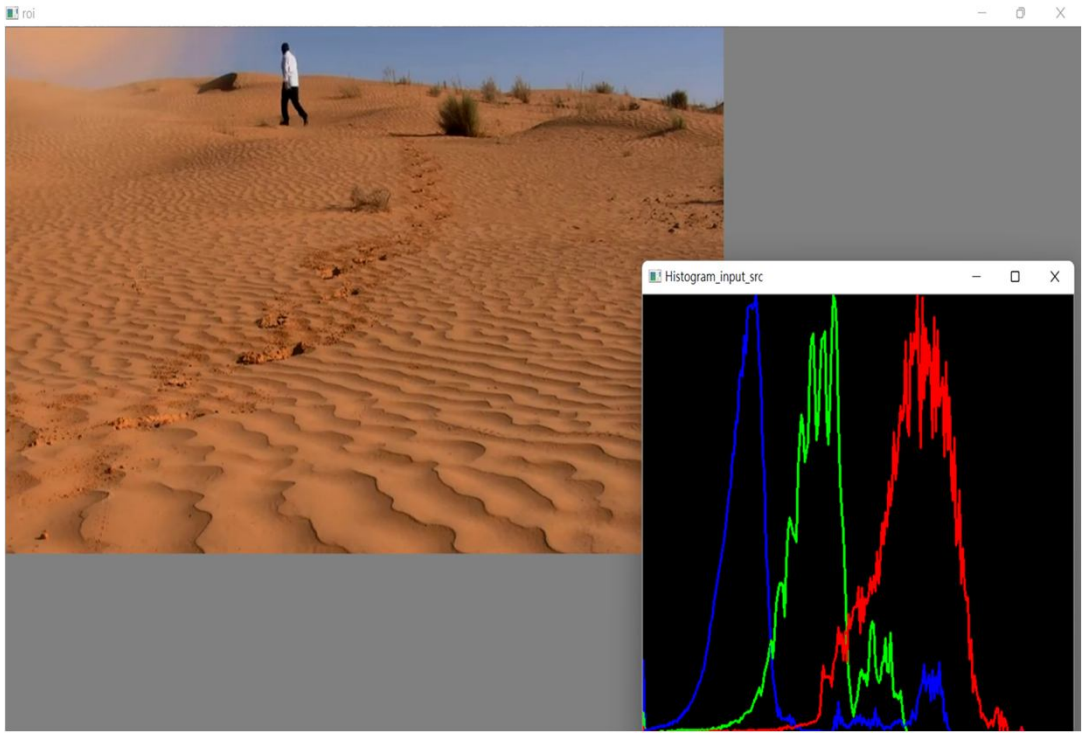


正常

亮度不均



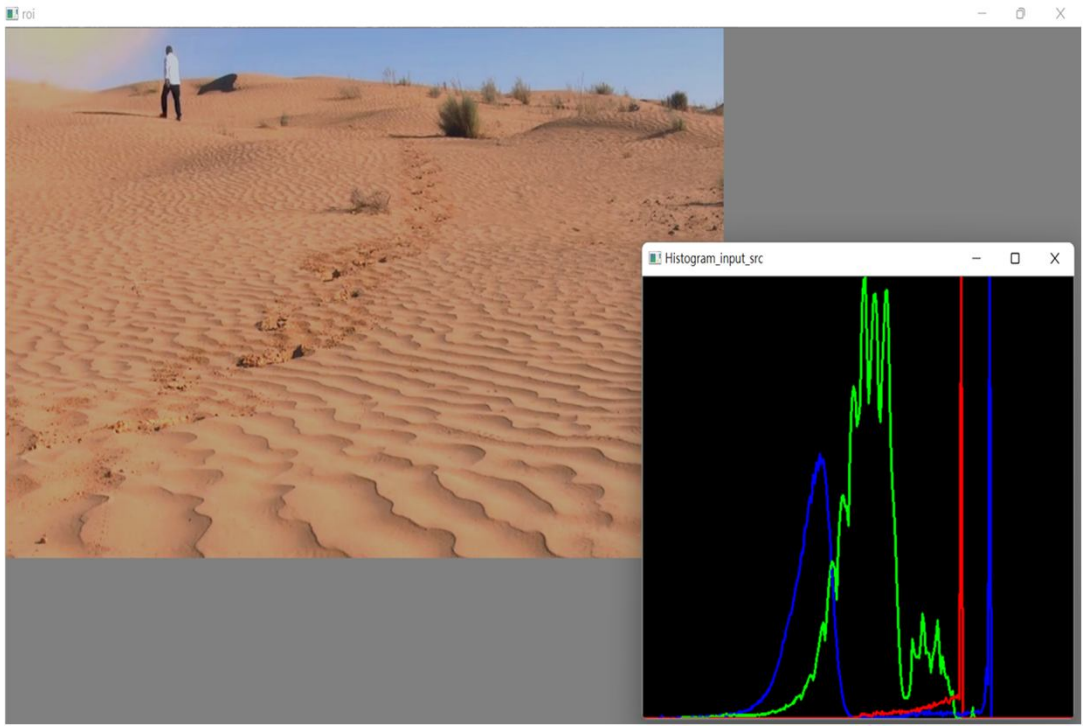
過亮



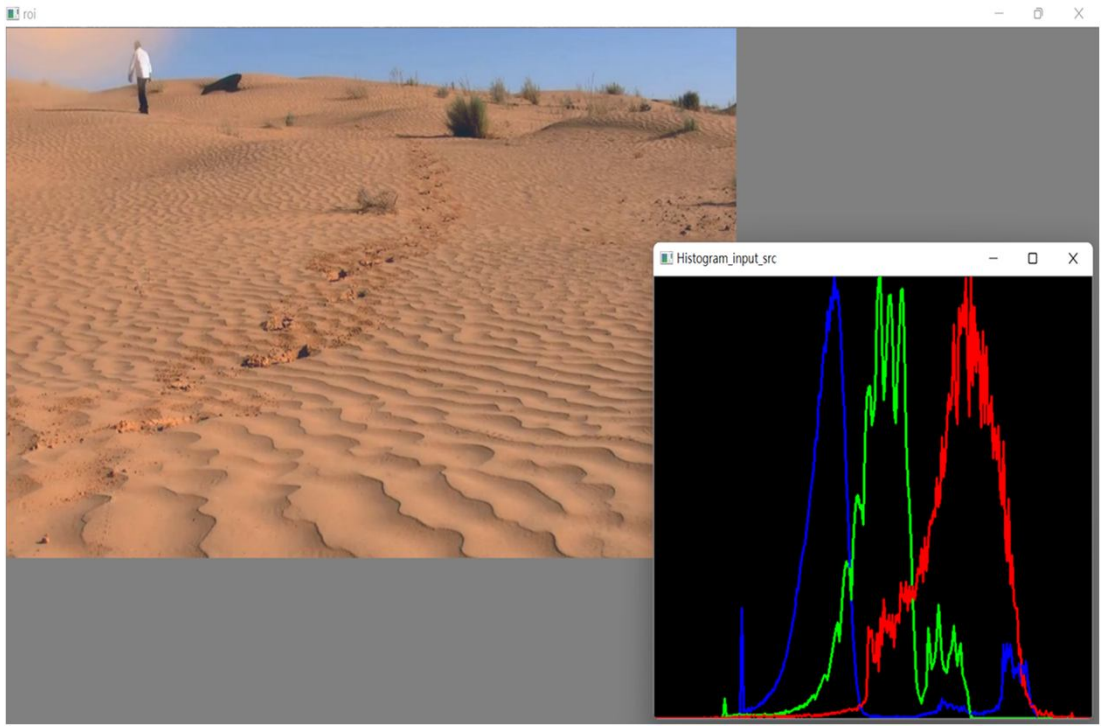
正常

亮度不均-調整

計算該張圖該通道與全部平均的差值，並進行加減。讓每個通道維持在平均值



原本過亮的圖



原本過暗的圖

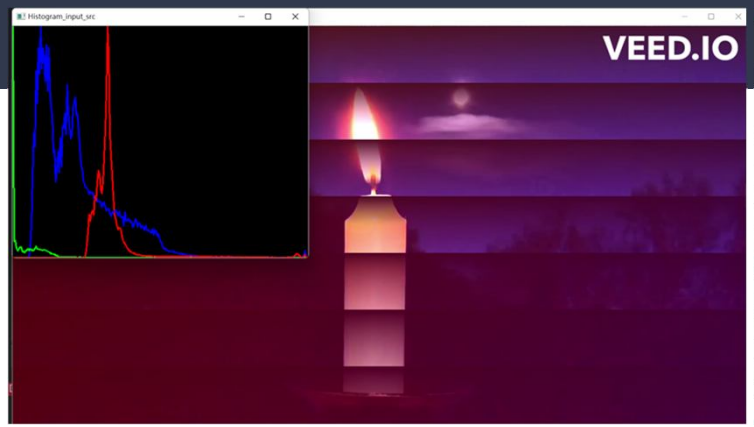
亮度不均 - Open source code

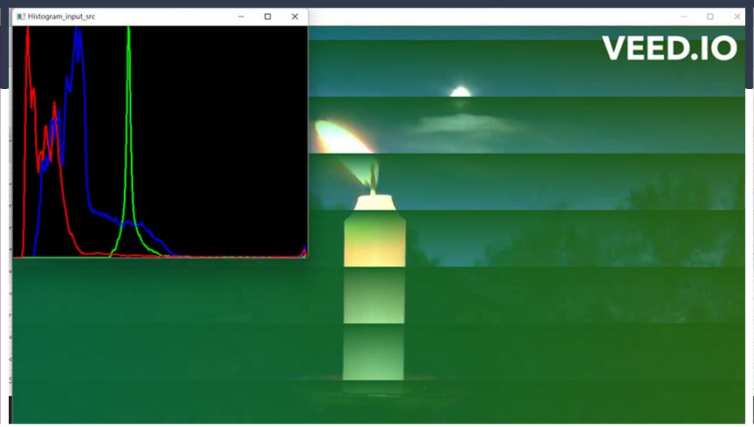
```
cv.add(src1, src2, dst, mask, dtype);
```

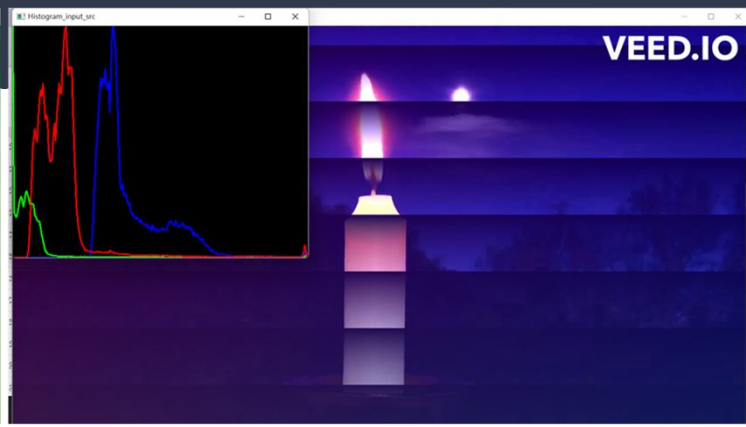
```
cv.subtract(src1, src2, dst, mask, dtype);
```

OpenCV Website link : OpenCV: Arithmetic Operations on Images

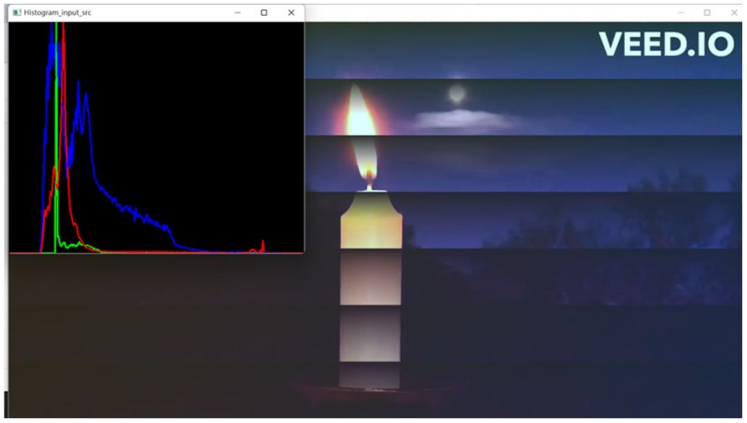
濾鏡

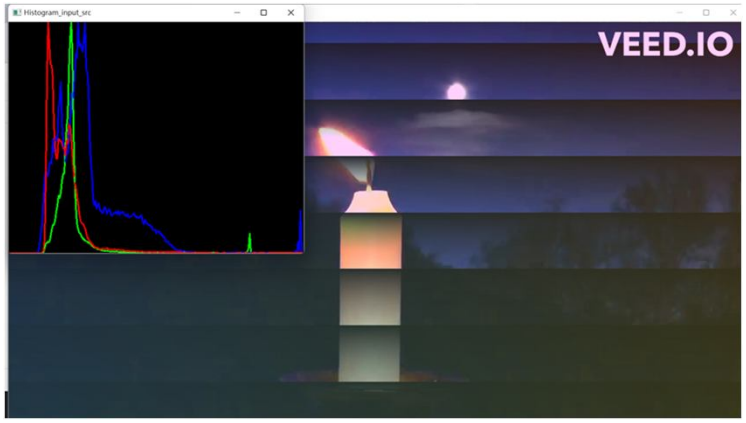


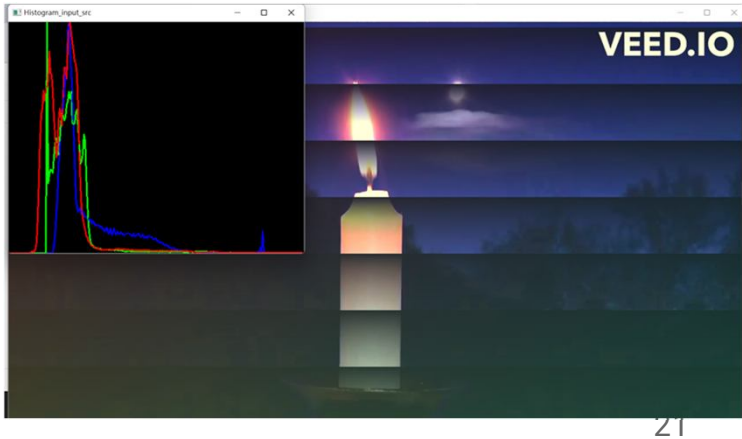




色調調整後:





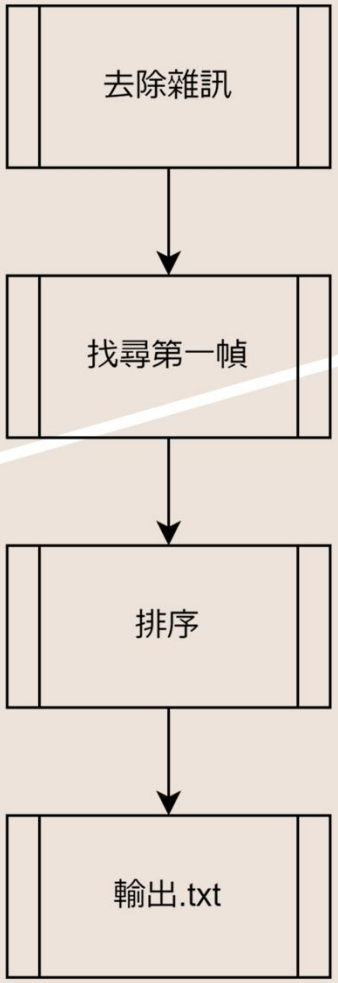


排列

輸入：已經去除雜訊的資料

處理1：利用相似度尋找第一幀
處理2：利用特徵點移動距離來排序

輸出：排列好的.txt檔案



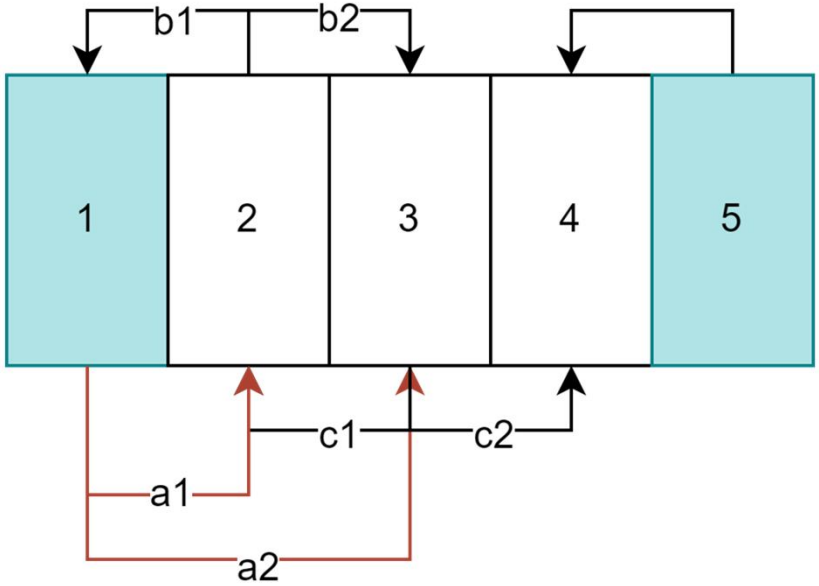
尋找第一幀-PSNR與相似度比較

利用PSNR判斷圖片間的相似程度

除了第一幀和最後一幀之外，其他幀理論上都會有兩幀與自己相似度高，而第一幀和最後一幀只會有一張與自己相似。

b1-b2和c1-c2的數值應該會比a1-a2小。

對資料集中每一幀的組合都做相同計算找第一幀或最後一幀。



參考資料：

C++ opencv 計算兩張圖像的PSNR相似度

其餘排序(1)-特徵點尋找SIFT

利用SIFT特徵點偵測方式找尋圖片特徵

問題：特徵點太多使比對時間消耗多

參考資料：

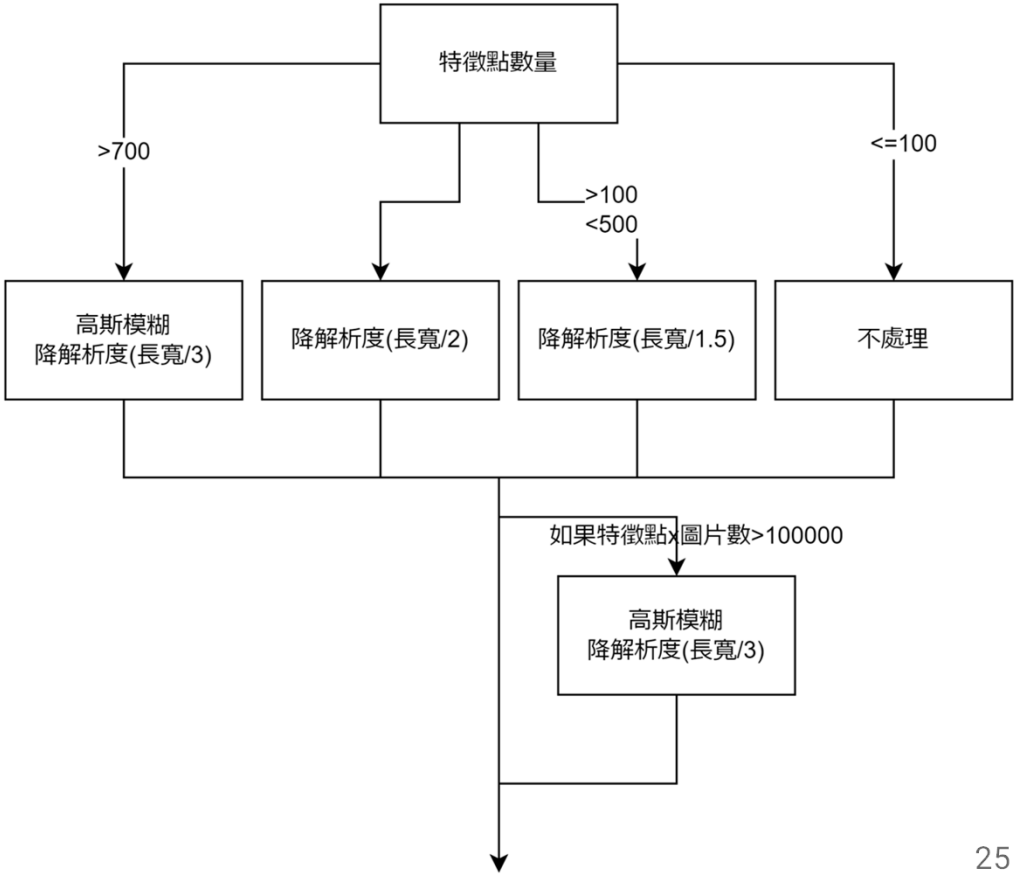
cv::SiftFeatureDetector Class Reference

圖像特徵比對(二)-特徵點描述及比對

其餘排序(2)-高斯模糊、線性resize

降低特徵點數量避免運行時間過長
(模糊或縮小之後再做特徵點偵測)

問題：正確率與速度間取捨
(特徵點越多SRCC越接近1，但時間會很長)



參考資料：
[Gaussian Blur OpenCV Tutorial C++]
[Image Resizing with OpenCV]

其餘排序(3)-特徵點配對

用opencv的matcher配對特徵點，找特徵點在兩張圖之間的座標距離

特徵點的移動距離平均最低者為下一幀

參考資料：

Feature Matching with FLANN

結果展示

數據集結果展示

	SRCC Rank	SRCC 索引	執行時間(s)	MSE_average
beach	0.398104	0.398104	298	3194.29
boat_style	0.551328	0.551328	1074	1664.39
candle_style	0.0700356	0.0700356	631	794.981
cctv	0.220956	0.220956	327	454.64
costline	0.747092	0.747092	247	3983.49
desert	0.368421	0.368421	204	3747.55
DMC	0.282018	0.282018	720	1074.4
flyout	0.874382	0.874382	337	1261.69
helltaker	0.258766	0.258766	272	735.767
PUBG	0.44473	0.44473	83	6428.86
RushPixar	0.00449294	0.00449294	427	5189.52

數據集結果展示

	SRCC Rank	SRCC 索引	執行時間(s)	MSE_average
school	0.521515	0.521515	260	4105.34
soccer_style	0.194727	0.194727	152	1540.7
TKUC	0.462024	0.462024	260	4443.74
typing	0.370771	0.370771	684	670.437
Average	0.3846	0.3846	395.2	2622.02

小組貢獻度

成員	貢獻度
葉雨婷	30%
謝瑞榆	30%
顏郁芩	40%

Thank you for your listening