**影像處理作業**

**Noise Removal**

**base on Image Averaging**

系級：資工四A

學號：102502521

姓名：高聿緯

2017年 1月5日

**目錄**

[**實作部分** 2](#_Toc503130134)

[**1.** **使用語言** 2](#_Toc503130135)

[**2.** **使用資料** 3](#_Toc503130136)

[**3.** **使用演算法介紹** 3](#_Toc503130137)

[**4.** **實作步驟** 3](#_Toc503130138)

[**結果** 4](#_Toc503130139)

[**1.** **灰階Lena** 5](#_Toc503130140)

[**2.** **彩色Lena** 5](#_Toc503130141)

[**3.** **彩色的還原影像放大** 5](#_Toc503130142)

[**結論** 6](#_Toc503130143)

[**心得** 7](#_Toc503130144)

**實作部分**

1. **使用語言**

Python with OpenCV library

1. **使用資料**

lena\_grey.jpg(512\*512\*3三個channel都是同數值)

lena\_color.jpg (512\*512\*3)

1. **使用演算法介紹**

當影像的噪點在時間上彼此無相關，而且平均為零，那麼我們可以將不同有噪點的影像相加，再平均，就能將一些雜訊去除。

設雜訊為n(x, y)，原始影像為f(x, y)，若將影像平均，則，又因為與時間無關，所以。

1. **實作步驟**
2. 將原始影像讀入，分為灰階影像 (只讀一個channel)

與彩色影像(3個channel)。

1. 分別對兩種影像加入10000個雜訊點，而雜訊加入的方法是從512\*512的像素中，隨機挑選出10000個像素，再從

-255到255之間選一個數值，直接加入R、G、B三個Channels中，由於數值與時間無關又平均為零，所以可以使用影像平均法去雜訊。

1. 分別儲存一張有雜訊的影像，再陸續生成另外19張有雜訊的影像，總和取平均之後看差異並且儲存起來。
2. 特別注意的是，像素儲存型態為unsigned integer 8，要轉為其他較大的型態才能正常的總和。

(原本是numpy.uint8，這裡我利用astype轉為integer，當要輸出影像之前，再將型態轉為numpy.uint8)

1. 對於彩色影像，我先將第一張雜訊圖的色彩空間轉為VHS，把20張雜訊圖的V加總取平均，再將第一張的H與S加回來試著還原原始影像。
2. 輸出的影像都存在output資料夾中。

**結果**

以下的圖分別分為灰階與彩色圖，從左到右為：原始影像，雜訊影像與還原影像，可以看到灰階還原的效果比較好，而彩色影像雖然有去掉一些雜訊，但是效果不這麼明顯，就算使用2000張有雜訊的圖去做平均，依然還是有幾個雜訊無法去除。

1. **灰階Lena**



1. **彩色Lena**



1. **彩色的還原影像放大**

可以明顯的看出，還原的效果並沒有很好，還有幾個小雜訊留著，所以這個方式不太適用於彩色影像。

****

**結論**

灰階影像使用影像平均來去除噪點效果非常好，但是對於彩色影像來說，因為色彩是根據RGB三個色彩的比例調配出來的，不能分成獨立三個Channels去做平均再和成起來，這樣顏色比例會跑掉，而造成色偏。

而對於彩色影像，為了避免色偏，我先將色彩空間RGB轉成VHS，再針對V(明度)去做影像平均來實現噪點去除的效果，根據結果來看，效果並不像在灰階影像上處理來得好，我想應該是有些色彩資訊被藏在HS之中，所以效果不是那麼的好。

**心得**

這次影像處理的實作，雖然只利用了簡單的影像平均來達成雜訊去除的工作，不過在灰階影像上，卻達到了不錯的效果，10000個像素的雜訊，僅僅利用了20張雜訊圖去做平均，就能達到不錯的還原。

而彩色影像，就結果來看，確實是有消除一些雜訊，但是對於無法去除、比較頑固的雜訊，我想應該可以利用非監督式機器學習，去區分”雜訊點”與”非雜訊點”，再利用”區塊的收縮放大”來解決，利用鄰近的像素來抹去雜訊，以達到去除雜訊的效果。