ISLab 暑假作業

前言

各位學弟妹大家好，歡迎你們加入溫馨的 ISLab！由於我們實驗室的研究方向與數位影像處理十分相關，所以在此提供一個題組幫助大家簡單熟悉一下相關的基礎知識、操作方法以及做研究的感覺。

在影像處理的眾多方法當中，最簡單的方式就是直接對原始影像像素的數值做基本的數值運算，如加減乘除等來得到結果影像，比如常見的空間域(Spatial domain)的濾波(Filtering)就是屬於這一類的方法，因此後面的練習將會圍繞在空間域濾波這個主題上。

那麼何謂空間域濾波？簡單來說就是對於影像上的像素，都會參考其鄰近的像素數值來輸出想要的結果，比如線性濾波器(Linear filter)就是將鄰近的像素數值做線性組合(Linear combination)後再輸出結果，用數學語言描述的話可以得到底下的式子

$$g(i,j) = \sum_{p=-h}^{h} \sum_{q=-w}^{w} c(p,q) \cdot f(i+p, j+q)$$

其中$f(i,j)$和$g(i,j)$分別為原始影像與結果影像在位置(i,j)的像素數值，$c(p,q)$為線性組合的係數，w和h代表我們參考以(i,j)為中心、大小為$(2w+1) \times (2h+1)$的鄰近區域。當所有的$c(p,q)$都為$\frac{1}{(2w+1)\times(2h+1)}$時，這個濾波器即為均值濾波器(Mean filter)，比如底下即為大小是3×3的均值濾波器。

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

接著大家就會想問，線性濾波器可以有什麼應用？因為影像的品質與取像的環境與設備有關，而在某些情境下(比如低光源的環境等)容易受雜訊的干擾，所以有個重要的議題就是如何去除雜訊(Noise reduction、denoise)，而線性濾波器就是其中一個常見的方法。底下將會簡單說明均值濾波器去除雜訊的原理。

一般的雜訊屬於加性雜訊(Additive noise)，也就是疊加在像素數值上的隨機變數(random variable)，其中最常見的雜訊是加性高斯白雜訊(Additive white Gaussian noise)，其意義是指這個隨機變數是來自於一個平均值(mean)為零(Zero-mean)的高斯分布(Gaussian distribution、normal distribution)，所以在此假設

遭遇到的雜訊為加性高斯白雜訊，如下所示

$$f(i,j) = h(i,j) + n(i,j)$$
$$n(i,j) \sim N(x|0,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$$

其中f為受雜訊干擾的原始影像，h為沒有受到雜訊干擾的影像，n為雜訊，$N(x|0,\sigma)$為平均值是 0、變異量(variance)為σ^2的高斯分布。可以看出去除雜訊的目的就是希望從f重建出h，但因為受隨機的雜訊影響，所以h的可能性會很多，因此有一種重建出h的法則稱為最大似然(Maximum likelihood)法則，其概念是希望找出一個h_{ML}滿足它在雜訊干擾後有最高的可能性剛好會等於原始影像f，以條件機率(conditional probability)的語言可以描述成如下

$$h_{ML} = \arg\max_{g} p(f|g)$$

同時因為每個像素所受的雜訊是獨立(independent)的，而且由於在以(i,j)為中心、大小為$(2w+1)\times(2h+1)$的區域中的像素數值通常來自於同個物體，所以能假設影像h在這個區域有相同的像素數值y，因此條件機率$p(f|h)$可以寫成如下

$$p(f|h) = \prod_{\substack{-h\le p\le h \\ -w\le q\le w}} N(f(i+p,j+q)|y,\sigma)$$

最後就可以基於最大似然法則與影像具有「空間相似性」(區域內的像素數值相似)的假設，得出h_{ML}來估計未受雜訊汙染的影像h

$$y = h_{ML}(i,j) = \frac{1}{(2w+1)\times(2h+1)} \sum_{p=-h}^{h} \sum_{q=-w}^{w} f(i+p,j+q)$$

即為均值濾波後所得到的結果。從上述的分析可以觀察到在影像的紋理(texture)較少時，越容易滿足「空間相似性」的前提，因此這時均值濾波器能夠有效地去除雜訊，但在畫面的邊界(edge)、紋理較多時可能就會表現較差(比如細節損失)，所以線性組合係數的設計就是影響處理效果的關鍵，比如高斯濾波器(Gaussian filter)、雙邊濾波器(Bilateral filter)等都是常見的改進方式。

在演算法的設計上，除了透過理論分析、實驗來確保影像處理的有效性(effectiveness)，計算的效率(efficiency)也是十分重要的，所以接著將會給一些練

習題幫助大家實際去加強理論與實作之間的連結。

作業內容

1. 前置作業
   完成 OpenCV 的安裝與環境設定，詳情可參考附檔說明或者是底下的連結。
   http://ppt.cc/Fk3r
2. 讀檔
   將附檔中的文字檔都個別轉成一張圖(共四個文字檔)，文字檔可由 word 或 txt 檔來開啟，檔案格式如下:

   資料從 DATA ascii 後面開始，前三欄都是浮點數，色彩資訊在最後一個浮點數裡(每行共四欄資訊)。由於浮點數是 32 bit 的格式，而 RGB 分別都是 8 bit 的數值，所以轉換的過程為
   整數 I (32bit)
   0x00000000
   放入 R
   0x000000RR
   放入 G
   0x0000RRGG
   放入 B
   0x00RRGGBB
   轉浮點數

   因此目標是從檔案中的浮點數擷取出像素的色彩資訊 RGB，並存放在 OpenCV 的影像容器中。關於影像容器，可以參考底下的網頁
   http://yester-place.blogspot.tw/2008/07/iplimage2.html
   http://docs.opencv.org/doc/tutorials/core/mat_the_basic_image_container/mat_the_basic_image_container.html

3. 實作線性濾波器
   請以第 2.題所讀到的結果來進行實驗。

   3.1 均值濾波器(Mean filter) (必做題)
      3.1.1 用基本巢狀迴圈實作均值濾波器
      3.1.2 以積分影像(Integral image)實作均值濾波器
         關於積分影像，請參考底下網站
         http://www.dotblogs.com.tw/dragon229/archive/2013/01/18/87485.aspx

3.1.3　比較兩種方法在濾波器大小為7×7和11×11時的運算時間

3.2 二項式濾波器(Binomial filter) (必做題)

這是一種在硬體上利用整數運算近似高斯濾波器的方法，主要是透過二項式展開的係數(巴斯卡三角形，Pascal triangle)來產生的線性濾波器，底下分別給出大小為3×3和5×5的例子

$$\frac{1}{16}\times\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix}$$

$$\frac{1}{256}\times\begin{bmatrix}1&4&6&4&1\\4&16&24&16&4\\6&24&36&24&6\\4&16&24&16&4\\1&4&6&4&1\end{bmatrix}$$

可以觀察到除法的分母都是 2 的冪次方，因此可以用簡單的移位(shift)運算。關於二項式濾波器與高斯濾波器之間的關係，請參考底下連結

https://en.wikipedia.org/wiki/Pascal%27s_triangle#Relation_to_binomial_distribution_and_convolutions

3.2.1　用基本巢狀迴圈實作二項式濾波器

3.2.2　利用可拆分(separable)的特性加速二項式濾波器的計算

關於可拆分的特性，請參考底下連結

https://en.wikipedia.org/wiki/Separable_filter

可以發現二項式濾波器是可拆分的，以3×3和5×5的大小為例

$$\frac{1}{4}\begin{bmatrix}1&2&1\end{bmatrix}\times\frac{1}{4}\begin{bmatrix}1\\2\\1\end{bmatrix}$$

$$\frac{1}{16}\begin{bmatrix}1&4&6&4&1\end{bmatrix}\times\frac{1}{16}\begin{bmatrix}1\\4\\6\\4\\1\end{bmatrix}$$

3.2.3　比較兩種方法在濾波器大小為7×7和11×11時的運算時間

3.3 雙邊濾波器(Bilateral filter) (挑戰題)

3.3.1　實作濾波器大小為7×7和11×11的雙邊濾波器

相關細節請看底下連結

https://en.wikipedia.org/wiki/Bilateral_filter

3.3.2 與 1.和 2.比較結果並分析造成差異的原因

3.3.3 思考加速的方法並實作

4. 自由發揮

此部分需自己實作，不得使用 OpenCV 中的函式來處理影像。

可參考

https://cg2010studio.wordpress.com/2012/02/19/%E7%B8%AE%E6%94%BE%E6%BC%94%E7%AE%97%E6%B3%95-scaling-algorithm/

(不一定只能是參考連結中的縮放，也可以是旋轉之類的)

注意事項

1. 除了影像容器、視窗顯示、存出圖檔，其他 OpenCV 提供的影像操作函式皆不得使用。
2. 請附上程式碼、有實驗數據的報告(在不同影像大小、濾波器大小下的計算時間)、處理後的結果圖，以及簡單的效果、效率分析和心得。
3. 若有實作雙邊濾波器，需附上參數設定。
4. 若無法實作出 利用 二項式濾波器 可拆分的特性之加速(3.2.2)，請試著實作 利用 均值濾波器 的可拆分性特之加速。
5. 建議環境為 Win7 下的 Visual Studio。

(要在 Linux 下實作也可，但請附上編譯方式)

以上為這次練習的內容說明，Deadline 定在 8/20，如有任何問題，歡迎寄信或來 lab 討論(lab 在電資大樓 708)。

最後，祝各位學弟妹暑假充實愉快