



國立高雄應用科技大學

100-101 年度獎勵科技大學及技術 學院 典範科大計畫

產學及研究成果轉專題製作教材

—(Low brightness image enhancement)

低亮度影像強化

低亮度影像強化

(科系) (指導教授) (學生)
電機系 黃敬群

蘇建穎
陳信樺

第一章 摘要

本專題中我們預備增強紀錄器的影像顯示功能提升紀錄器在監控錄影的功效，現今，雖然已有許多擁有良好視野的紀錄器，當然價錢也隨著效能而更昂貴。

計畫中，我們將嘗試透過軟體演算法的實作，或搭配取像硬體的改善，來修正夜間影像照度不足或產生的影像品質不良的缺陷。

此外，本專題將以記錄器為實驗平台，在一定硬體成本的控制下，透過軟體演算法的改進，提升紀錄器的取像品質，我們一開始嘗試過全域處理的演算法來處理影像，例如：伽瑪校正以及直方圖等化的方式來修整影像，結果我們發現動態範圍過大的影像無法適用，接著，我們學習了 Retinex 演算法，此種方法屬於局部影像處理，可以針對亮度值較低的像素提高亮度，並且不會使原本亮度值高的部分更亮，根據 Retinex 又分了許多種方法，例如 SSR、MSR、IMSR 等方法，用這些方法來對影像進行，另外也學習了自適應局部綜合方法的彩色圖像非線性增強方法與有效的快速算法增強光線較暗的圖像方法，自適應局部綜合方法的彩色圖像非線性增強方法結合了自適應亮度增強為亮度信息處理和動態範圍壓縮，自適應對比度增強為保存的細節和逼近與原始圖像的色調這兩種方法，另一方法快速算法增強光線較暗的圖像使用低燈光的倒置視頻增強算法應用來快速得到結果。

第二章 模擬軟體

2-1 MATLAB

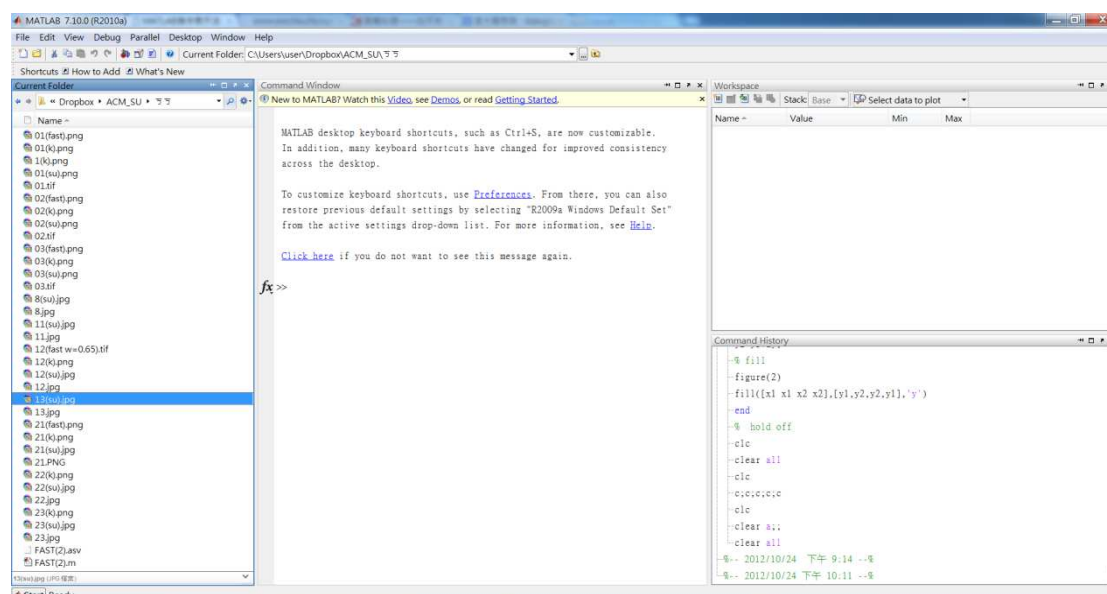


圖 2-1 MATLAB 使用介面

2-1-1 MATLAB 之簡述

MATLAB 是 MATrix LABoratory 的縮寫，是一款由美國 The MathWorks 公司出品的商業數學軟體。MATLAB 是一種用於演算法開發、資料視覺化、資料分析以及數值計算的高階技術計算語言和互動式環境。除了矩陣運算、繪製函式/資料影像等常用功能外，MATLAB 還可以用來建立使用者介面及與呼叫其它語言（包括 C，C++和 FORTRAN）編寫的程式。儘管 MATLAB 主要用於數值運算，但利用為數眾多的附加工具箱（Toolbox）它也適合不同領域的應用，例如控制系統設計與分析、影像處理、訊號處理與通訊、金融建模和分析等。另外還有一個配套軟體包 Simulink，提供了一個視覺化開發環境，常用於系統類比、動態/嵌入式系統開發等方面。

2-2 MATLAB 之影像處理指令簡介

1. MATLAB 常用指令

MATLAB語法	指令意義	備註
help	求助功能	
demo	執行範例程式	
who	列出記憶體中所有變數	
what	列出 .M 檔	
size	行的維度	
length	向量長度	
clear	清除記憶空間之資料	
~c	中止執行	

圖 2-2 常用指令圖

2-3 android

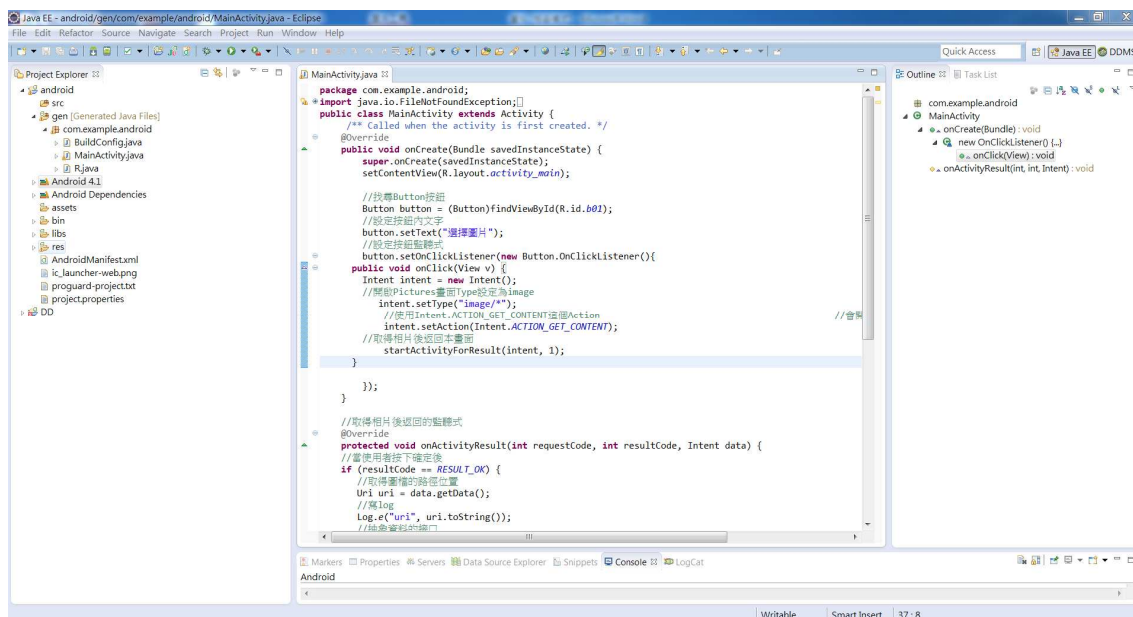


圖 2-4 android 使用介面

2-3-1 android 簡述

Android 以 Linux 為核心的 Android 行動平台，而在早期的 Android 應用程式開發中，通常通過在 Android SDK（Android 軟體開發包）中使用 Java 作為編程語言來開發應用程式。開發者亦可以通過在 Android NDK（Android Native 開發包）中使用 C 語言或者 C++ 語言來作為編程語言開發應用程式。同時 Google 還推出了適合初學者編程使用的 Simple 語言，該語言類似微軟公司的 Visual Basic 語言。此外，Google 還推出了 Google App Inventor 開發工具，該開發工具可以快速地構建應用程式，方便新手開發者，開發者開發完成式之後，還可以執行手機模擬器，來運行程式進行測試。

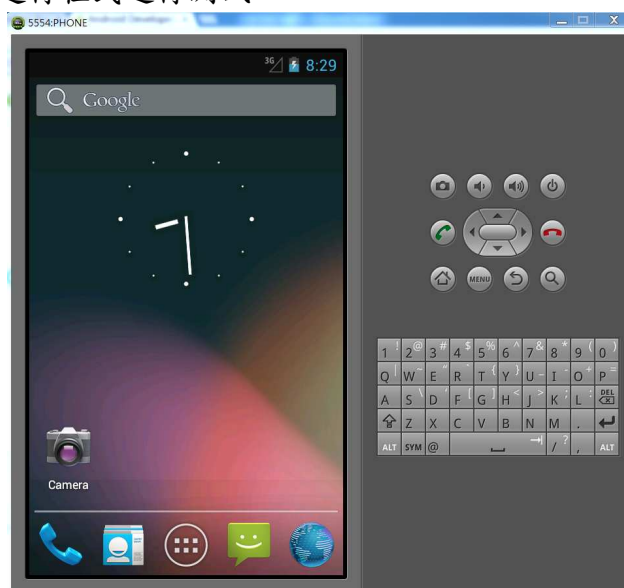


圖 2-5 手機模擬器

第三章 伽瑪校正

3-1 伽瑪校正之相關知識

對圖像的伽瑪曲線進行校正，以對圖像進行非線性色調校正的方法，檢驗出圖像信號中的深色部分和淺色部分，並使兩者比例增大，從而提高圖像對比度效果。

傳統的伽瑪校正公式(1)如下式：

$$Output = 255 \left(\frac{Input}{255} \right)^{\gamma} \quad (1)$$

其中 Input 是輸入影像的 RGB 值

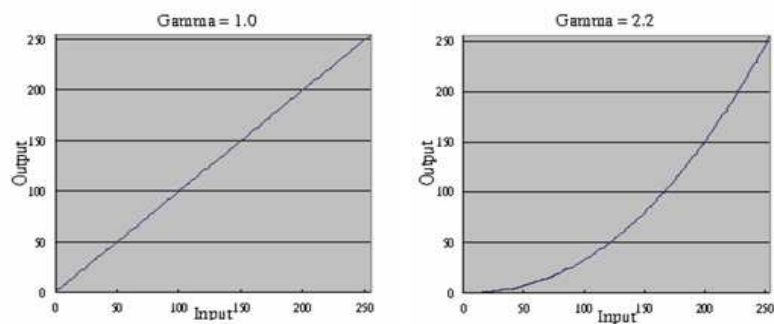


圖 3-1 Gamma 對應 RGB 顏色示意圖

3-2 伽瑪校正之模擬結果

1. 低亮度圖形



圖 3-2 原始圖形



圖 3-3 經過伽瑪校正後圖形

2. 夜間圖形



圖 3-4 原始圖形



圖 3-5 經過伽瑪校正後圖形

第四章 直方圖均衡化

4-1 直方圖之相關知識

直方圖:對於彩色影像而言，直接使用此方法處理影像是非常不容易的事，只有將影像的基本色調給分別處理分析才能知道影像的資訊狀況與色彩分布狀況。

4-2 直方圖均衡化之相關知識 [13]

而直方圖均衡化是影像是過高的亮度或者是過低的亮度又或者是過低的對比度的話，影像的基礎色在處理分析圖表內之中的分布圖將會聚集在某一處，直方圖均衡化的目標就是為了讓這些色彩平均分布在影像上的處理影像之中一種方法，處理後的影像會是高對比容易讓肉眼觀賞的影像。

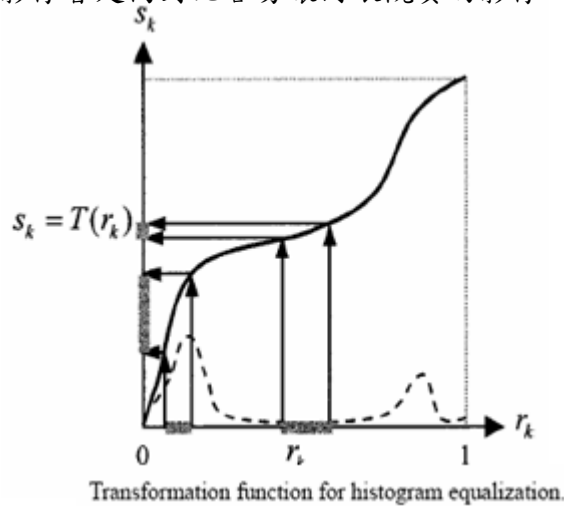


圖4-1 直方圖均衡化示意圖

直方圖均衡化公式(2)如下式:

$$s_k = T(r_k) = \sum_{j=0}^k P_r(r_j) = \sum_{j=0}^k \frac{n_j}{n} \quad (2)$$

$$0 \leq r_k \leq 1 \text{ and } k = 0, 1, \dots, L-1$$

$$P_r(r_j) = \frac{n_j}{n}: \text{輸入圖像的 } j \text{ 階層的概率密度函數 (PDF)}$$

n : 在輸入圖像的總像素數

n_j : j 階層的輸入像素數

對照圖4-1算法:

- (i) 將影像從檔案從三維轉成二維的陣列，並且計算出RGB三色的直方圖表和Y值的直方圖表。
- (ii) 將算法(i)所得到的直方圖表計算得出累進分布表(CDF)。

(iii) 將算法(ii)中所得的結果計算得出轉換表。

(iv) 將算法(iii)中所得的結果將原始影像做轉換得出新的高對比度影像，並且計算轉換後的亮度(Y)值直方圖表。

4-3 直方圖均衡化之模擬結果

1. 低亮度圖形



圖 4-2 原始圖形

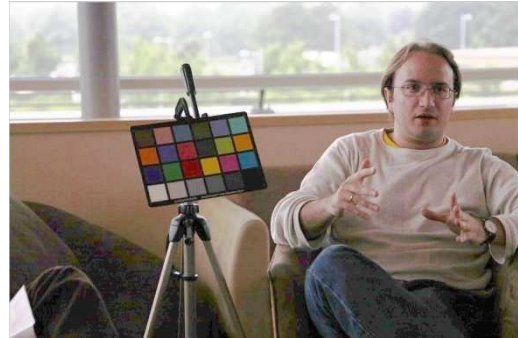


圖 4-3 經過直方圖均衡化後圖形

2. 夜間圖形



圖 4-4 原始圖形



圖 4-5 經過直方圖均衡化後圖形

第五章 Retinex 方法

5-1 Retinex 之相關知識 [2][3][6][7]

過去此演算法常用於數位相機自動白平衡中，它除了具有色彩恆常性，亦包含了強化影像的效果，對於光線對影像所造成的影響，Retinex具有良好的處理效果。而Retinex演算法最早是由Edwin Land (1964)所提出來的，這是Edwin Land在工作時所發現的，他利用投影機把影像投射在白色的布幕上，並且把一個紅色的濾片鏡放在鏡頭前，當透過這個紅色的濾波片觀看彩色影像時，發現到從紅色濾色片中所看見的影像，並不是整張影像都呈現紅色的，還是有包含其他色彩成分。由此可推知，當人眼觀察到一個物體時，所看見的顏色，並不只是由物體反射出來光譜能量而已，而是藉由此物體與鄰近物體之間的色彩和明度值比較後而得到的色彩感覺，然而這樣的色彩感知能力是經由人腦中的視網膜(Retina)與大腦皮層所處理的。因此，在人眼視覺中，藉由視網膜取得影像的刺激值，再經由大腦皮層解譯視網膜所取得的色彩資訊，進而得知物體正確的顏色，以如此的運作方式，使人們得以感受到真實的影像，所以Edwin Land 將此演算法取名為Retinex。而針對此演算法的研究也發展許久，這些研究都是針對影像的光源影像去計算與改良，而其共通點就是以影像中，任一像素(pixel)為中心，以擴散的概念來尋找、比較中心像素的周圍環境。

Retinex 最主要的目的就是利用影像表面的明度值，在每個獨立的感光受器(人眼的感光細胞或視攝影器材的感光元件)與周圍的感光受器之間做比較，衡量出人眼所感知到的影像。然而物體會被我們人眼所看見主要是藉由物體本身的反射率，和光線所導致我們人眼看見的樣子，因此便可以用一個數學式來表示物體在人眼中成像的模型

如下 Retinex 演算法的基本概念公式(3):

$$I'(x_i) = I(x_i) + \log(\rho^{x_i}) - \log(\rho^{x_j}) \quad (3)$$

$I'(x_i)$ 為影像中的像素， $\log(\rho^{x_i})$ 為計算中像素的照明參數， $\log(\rho^{x_j})$ 為計算中像素周圍的像素的照明參數，由此概念發展出各種演算法。

在過去的研究中可以發現，大部分的環境光源會分布在低頻的部分，因此Retinex 裡所需要的亮度值便可以透過低通濾波的方式取得整張影像的光源部分。

5-2 Single Scale Retinex 方法 [2]

在此演算法中，最簡單的即是單一規模的高斯函數，稱為Single Scale Retinex (SSR)，此作法只需考慮周圍的八個像素點，或是以等比例的方式向外擴張，公式如下所示(4)：

$$R(x, y) = \log I(x, y) - \log [F(x, y) * I(x, y)] \quad (4)$$

$I(x, y)$ 是影像的灰階值， $*$ 代表卷积(Convolution)運算， $F(x, y)$ 是高斯周圍函數(Gaussian Surround Function)，而高斯函數的計算如下(5)：

$$F(x, y) = K e^{\frac{-(x^2 + y^2)}{c^2}} \quad (5)$$

在上式(5)高斯函數的當中， $x^2 + y^2$ 是與中心點的距離，而 c 值是常數，用來決定高斯函數所包含的範圍，另外 K 是常數，滿足下列公式(6)：

$$\iint F(x, y) dx dy = 1 \quad (6)$$

所以 c 值的大小，決定了高斯函數計算後的結果，進而影響了Retinex演算法的處理效果，當 c 值越大的時候，所包含的像素範圍較大，可以保留較多的色彩資訊， c 值較小時，所包含的像素範圍較小，雖然可以強化影像的邊緣資訊，但是所保留的色彩資訊較少，如此一來會造成無法兩全其美的結果，不是保留的色彩資訊太少，亦或是沒有強化影像的邊緣資訊，之所以又產生MSR演算法來改進SSR的美中不足。

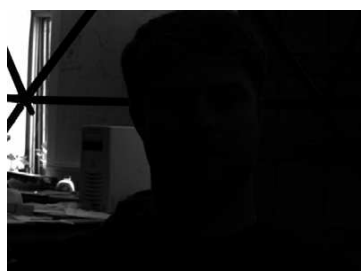


圖5-1 原始影像



圖5-2 SSR處理(c=2)



圖5-3 SSR處理(c=150)

圖5-1，圖5-2和圖5-3所展示的三張影像，即是一張原本被陰影所遮蔽的人臉影像，經由SSR處理後，可以發現原本被陰影所遮蔽的人臉部分，利用周圍環境的像素點使其被強化出來。此外，也可以觀察到若 c 值越大，影像所蘊含的色彩資訊也就越多，五官較為清晰，但邊緣資訊卻相對的較為模糊，如圖5-3。

5-2-1 Single Scale Retinex 之模擬結果

1. 低亮度圖形



圖 5-4 始圖形



圖 5-5 經過 SSR 後圖形

2. 夜間圖形



圖 5-6 原始圖形



圖 5-7 過 SSR 後圖形

5-3 Multi Scale Retinex 方法 [1][2][5]

由於 SSR 沒辦法兼顧色彩與邊緣資訊又發展出 MSR 演算法，Woode11^[3]提出了利用不同規模的高斯函數，發展出新的演算法，稱為 Multiple-Scale Retinex (MSR)，運作的方式是以不同的權重，給予各個不同大小的高斯函數，同時對影像的邊緣與色彩資訊做保留，而圖(2)與圖(3)為 SSR 與 MSR 計算範圍的示意圖，黑色的區塊為目前所要計算的像素點。

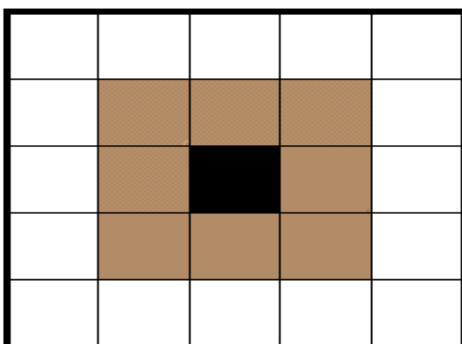


圖5-8 SSR示意圖(只有一個濾波)

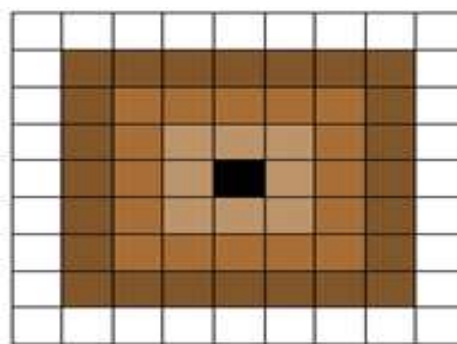


圖5-9 MSR示意圖(多個濾波)

在圖(5-9)中，有三個不同的顏色區域，為三個規模的高斯函數，每個高斯函數都給予不同的權重值，而權重值的總和等於 1，經由這樣的方式，改善了

SSR 演算法因為 c 值所造成的缺點，小的高斯函數保留影像的邊緣資訊，大的高斯函數保留影像色彩資訊，以此方式將同時達到保留邊緣資訊，和保留色彩資訊的效果。

$$R(x, y) = \sum_{K=1}^K W_K (\log I(x, y) - \log [F(x, y) * I(x, y)]) \quad (7)$$

如上式(7)，其中 W_k 代表第 k 個 SSR 的權重值，且 $\sum_{K=1}^k W_K = 1$ ， $F(x, y)$ 是高斯函數，即每個單一高斯函數乘上各自的權重值之後再加總起來，如下所示(8)：

$$R_{MSR}(x, y) = \sum_{K=1}^K W_K R_{SSR}(x, y) \quad (8)$$



圖5-10原始影像



圖5-11為經過MSR處理後



圖5-12原始影像



圖5-13經過MSR 處理後

圖5-10與圖5-12分別代表不同的兩個人，在不同的光線變化下，經由MSR處理過後所得到的結果。可以明顯的觀察到，在加入了不同大小的高斯函數與給予不同權重並加大了濾波器的大小，考慮了更多周圍環境的像素點之後，人臉陰影的部份更清晰的被強化出來。

以上實驗皆針對於灰階影像來作處理，如果要對彩色影像，Retinex 在做運算時，必須將 RGB 三個色頻分開來處理，分別的對每個色頻做強化，再合併這

三個色彩空間的資訊作輸出。但是將 RGB 三個色頻合併輸出影像之後，會造成下列這些缺點：

- (1) 輸出時，顏色重現可能無法呈現出正常的顏色，而且看起來會不自然。
- (2) 會造成整張影像飽和度下降，產生影像變灰的感覺。
- (3) 若是原始影像本身有偏向某個主色調，在經過MSR 處理後，反而會有色偏的情況發生。
- (4) 造成光暈現象。
- (5) 當影像中明度區域佔大多數時，對於比例較小的暗部做增強時，效果不是那麼明顯。
- (6) 在做影像增強時，影像中暗部或陰影中的雜訊也會跟著增強。

5-3-1 Multi Scale Retinex 之模擬結果

1. 低亮度圖形



圖 5-14 原始圖形

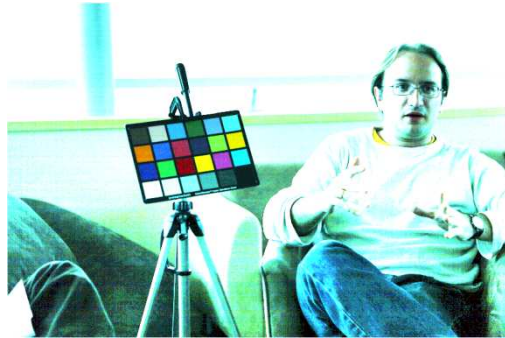


圖 5-15 經過 MSR 後圖形

2. 夜間圖形



圖 5-16 原始圖形



圖 5-17 經過 MSR 後圖形

5-4 Multi Scale Retinex with color Restoration 方法 [2]

由於 MSR 這方法缺點是無法完整保存原圖的色彩資訊，所以接著發展出這種方法，其概念是，考慮到原始影像的 RGB 資訊，再藉由 MSR 的結果在進行調整，將進行 MSR 之後 RGB 色頻的資訊拉回到原本影像的 RGB 色頻值一定程度，如公式 (9)：

$$R'_{MSR_i}(x, y) = R_{MSR_i}(x, y) \times I'_{i(x, y)} \quad (9)$$

其中 $I'_i(x, y)$ 是由下列公式(10)給定，根據 RGB 資訊在影像上的比例作調整：

$$I'_i(x, y) = \beta \log\left(\alpha \frac{I_i(x, y)}{\sum_{i=1}^3 I_i(x, y)}\right) \quad (10)$$

$I'_i(x, y)$ 代表 R, G, B 的像素值，而 α 、 β 代表常數。

經過了 MSRCR 的計算後，就可以克服 MSR 對於彩色影像所產生的那些問題。

5-4-1 MSRCR 之模擬結果

1. 低亮度圖形



圖 5-26 原始圖形

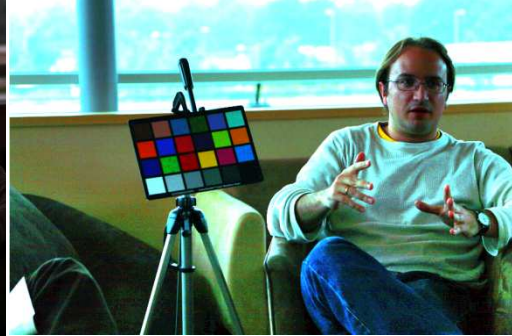


圖 5-27 經過 MSRCR 後圖形

2. 夜間圖形



圖 5-28 原始圖形



圖 5-29 經過 MSRCR 後圖形

5-5 Integrated Multi Scale Retinex with gamut mapping [9]

此演算法是利用 gamut boundary 調整影像亮度增強後的亮度和色彩飽和度，而亮度增強則是利用Retinex的imsr方法做增強，圖八為此演算法的流程圖(5-18)：

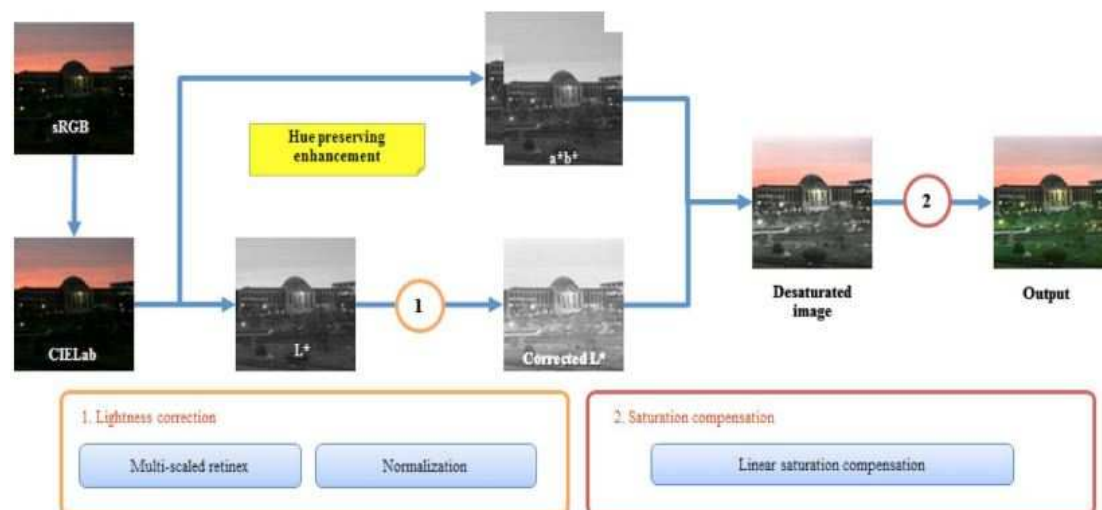


圖5-18 MSRCR流程圖

對一張輸入的影像做以下的處理步驟：

- (i) 將RGB影像轉到CIELAB，得到L*亮度值
 - (ii) 將亮度用imsr做增強
 - (iii) 利用gamut boundary對增強後的亮度做調整
 - (iv) 將原本影像的a*, b*和調整後的L*合起來，並做色彩飽和度拉升
- 以下將為每個步驟做一一詳細的說明

(i)將RGB影像轉到CIELAB，得到L亮度值

先將每個像素點RGB空間數值分別算出X, Y, Z

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} R_{sRGB} \\ G_{sRGB} \\ B_{sRGB} \end{bmatrix} \quad (11)$$

再利用X, Y, Z分別算出每個像素點的L*, a*, b*

$$L^* = \begin{cases} 116\left(\frac{Y}{Y_n}\right)^{1/3} & \text{if } \frac{Y}{Y_n} > 0.008856 \\ 903.3\left(\frac{Y}{Y_n}\right) & \text{if } \left(\frac{Y}{Y_n}\right) < 0.008856 \end{cases} \quad (12)$$

$$a^* = 500 \left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right) \quad (13)$$

$$b^* = 200 \left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right) \quad (14)$$

$$f(x) = \begin{cases} x^{1/3} & \text{if } x > 0.008856 \\ 7.787x + \frac{16}{116} & \text{if } x < 0.008856 \end{cases} \quad (15)$$

(ii)將亮度用imsr做增強

由於Retinex基本理論在前面已經介紹過，以下直接列出imsr的數學式加以說明：

$$LR_{\text{merge}}^*(x,y) = \alpha \frac{L^*(x,y)}{SR_{\text{merge}}^*(x,y)} \quad (16)$$

$$SR_{\text{merge}}^*(x,y) = \sum_{m=1}^M w(\sigma_m) SR_m^*(x,y, \sigma_m) \quad (17)$$

$$SR_m^*(x,y, \sigma_m) = G_m(x,y) \otimes L^*(x,y) \quad (18)$$

從公式中可以看出，這跟MSR的理論相似，不同的是對一張色彩影像，他只將影像中的亮度部分提取出來做處理，並且不像MSR利用log做運算，因此在MSR的log相減，在這邊便會變成相除。

(iii)利用gamut boundary對增強後的亮度做調整

在CIELAB空間的gamut boundary示意圖如下所示：

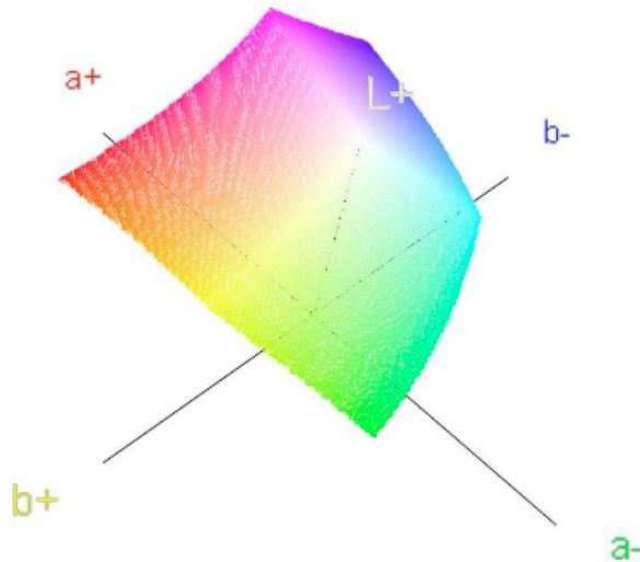


圖5-19 gamut boundary 示意圖

對於為什麼在亮度經過強化後還需要做強化過後的調整，將以下面的圖片說明：

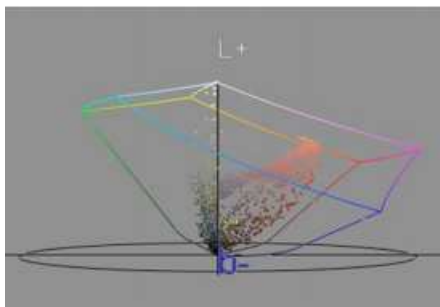


圖5-20

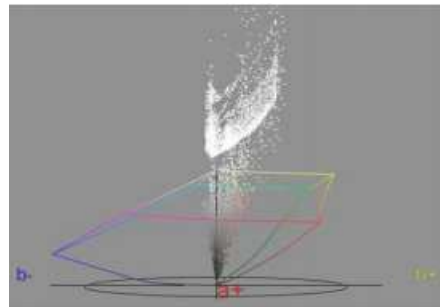


圖5-21

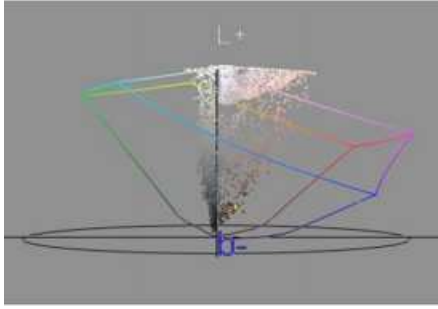


圖5-21

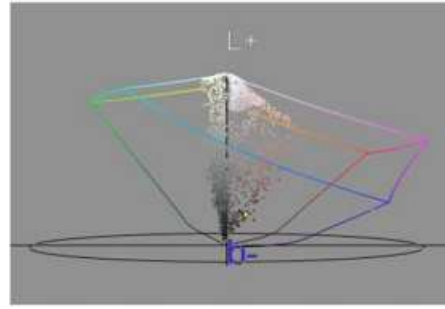


圖5-22

圖5-20是一張輸入影像再CIELAB空間的分布圖，圖5-21是輸入影像經過IMSR之後的結果分布圖，圖5-21是圖5-20經過正規化之後的樣子，到這邊可以從圖5-21中發現到，即使原本經過加強亮度的影像貝正規化之後，還是有一些點分布在邊界之外(超出邊界)，這就是為什麼需要對亮度強化後的影像做亮度調整的原因。因此做完亮度調整之後如圖5-22，所有的點都邊界之內。

在剛剛上述中，可以發現兩個問題：

- (1)如何取得正規化的標準(並不是單純取最大亮度值)
- (2)如何對超出邊界的亮度值做調整

將會在下面做詳細的討論

(1) 如何取得正規化的標準

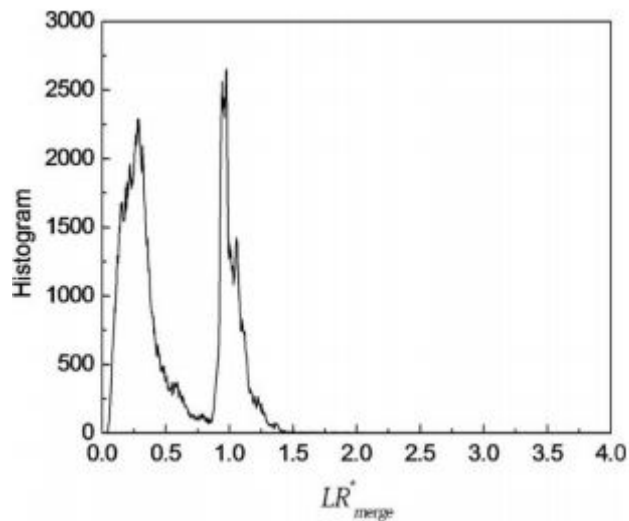


圖5-23 亮度值經過IMSR處理後的直方圖

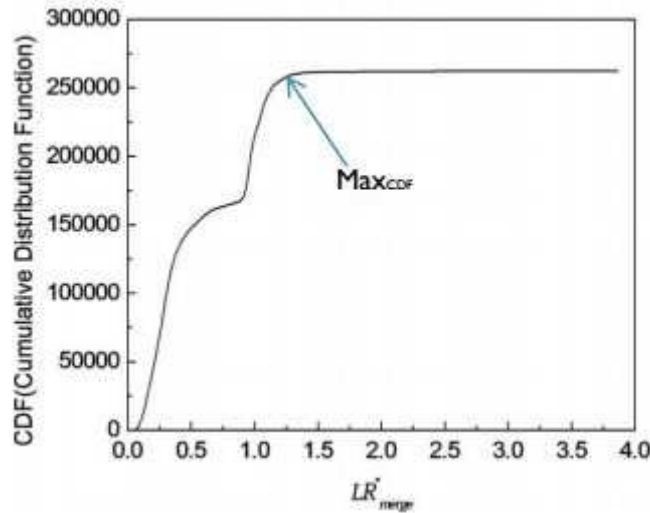


圖5-24 亮度值經過IMSR處理後的累積直方圖(CDF)

對於某些影像來說，可能會發生大部份的亮度值分布於強度比較小的地方，只有少部分某幾個點的值是分布在強度很大的地方，在這種情況下，如果我們把最大的值當作正規化的標準，那麼勢必大部分分布在強度比較小的亮度值會被正規化的很小，甚至趨近於0。因此為了改善這種情況，我們必須利用亮度值的累積直方圖(CDF)來選擇正規化的標準。

如圖5-24所示，我們取當斜率趨近於0的地方當作是正規化的標準值，雖然取到的值有可能不是整張影像的最大值，但這樣才能避免有些值被壓抑的太低。

(2) 如何對超出邊界的亮度值做調整

在我們做完亮度的正規化之後，利用圖九的gamut boundary和每個像素點已知的 a^* , b^* 找出每個像素點所對應到 L^* 的邊界值，對每個正規化後的亮度值分別乘上所相對應的邊界值，將每個點都保持在邊界以內。公式如下：

$$LR_{correct}^*(x, y, \sigma_m) = Gb_{Hat}(a^*, b^*, x, y) \frac{LR_{merge}^*(x, y, \sigma_m)}{Max_{CDF}} \quad (19)$$

公式中，等號左邊是我們調整完後的結果， Max_{CDF} 是找出的正規化標準值，而 LR_{merge}^* 則是原本亮度最 IMSR 處理後的結果，所以後面的分數是正規化的動作，而 $Gb_{Hat}(a^*, b^*, x, y)$ 則是每個像素點所對應的 L^* 邊界。

(iv) 原本影像的 a^* , b^* 和調整後的 L^* 合起來，並做色彩飽和度拉升

在這邊也是利用 gamut boundary 的概念對飽和度做拉升，原本影像的亮度值 L^* 再經過 IMSR 處理過後，亮度值會改變，相對水平對應到的邊界也會跟著改變，因此 a^* 和 b^* 也必須等比例的隨著邊界值不同做調整，這邊要注意的是 a^* , b^* 必須等比例做調整否則色調會變得跟原本影像不同。

如下圖 5-25 所示：

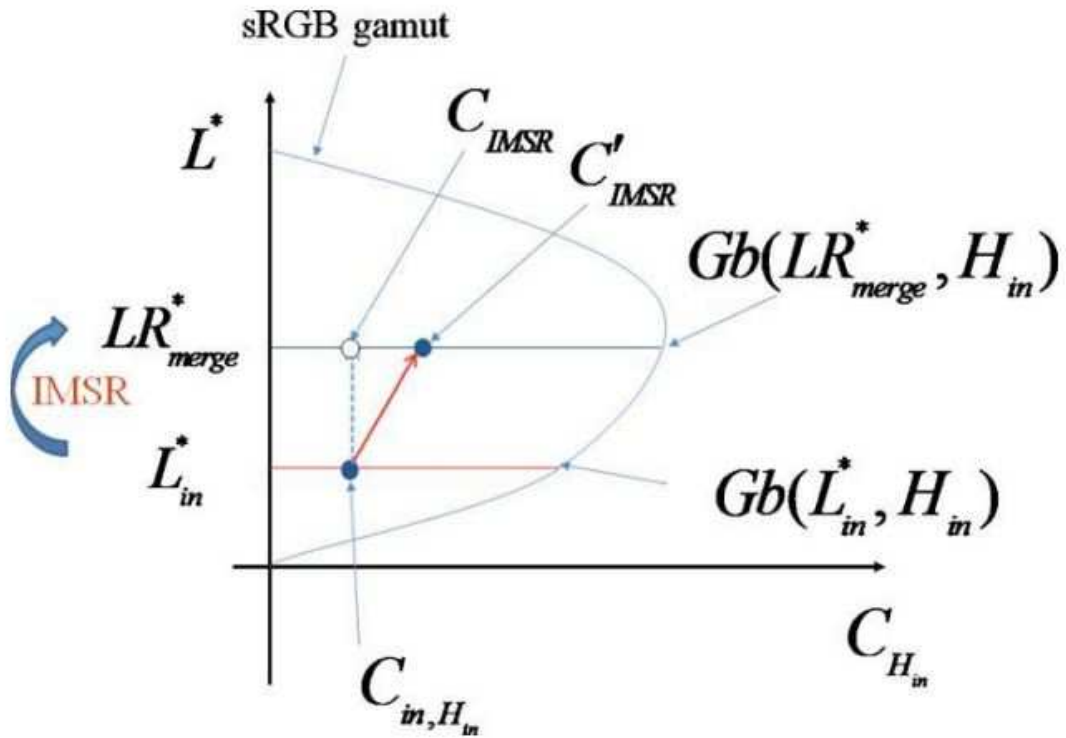


圖5-25

L_{in}^* 是輸入影像的亮度值， LR_{merge}^* 是 IMSR 處理過後的亮度值，從圖中可以發現到，所對應的邊界值如上述已經改變，因此每個像素點原本所對應的 a^* , b^* 必須依照邊界值的變化，做等比例的改變，Gb 即是不同 L^* 所對應的邊界值。因此飽和度拉升的公式如下：

$$G'_{IMSR}(x, y, H_{in}) = \frac{Gb(LR_{merge}^*)(x, y, H_{in})}{Gb(L_{in}^*)(x, y, H_{in})} C_{in}(x, y, H_{in}) \quad (20)$$

G'_{IMSR} 是我們做完飽和度拉升的結果， C_{in} 則是每個像素點原本的 a^* 和 b^* 。

5-6 Integrated Multi Scale Retinex 方法 [1]

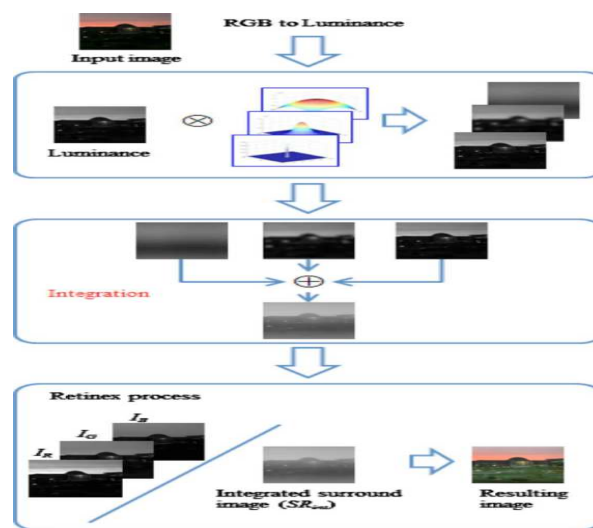


圖 5-30 IMSR 流程圖

為了保持色彩的平衡又發展出 IMSR 這方法, 其中理念是將亮度單獨取出作調整, 下列為程序的公式(21):

$$SR_{sum}(x, y) = A \frac{I_i(x, y)}{S_{sum}(x, y)} \quad (21)$$

其中 I 是 RGB 輸入影像, i 是 RGB 通道的索引值, A 為增益係數。

$$S_{sum}(x, y) = \sum_{m=1}^M w(\sigma_m) S_m(x, y, \sigma_m) \quad (22)$$

$$S_m(x, y, \sigma_m) = G_m(x, y) \otimes Y(x, y) \quad (23)$$

$$\sum_{m=1}^M \omega(\sigma_m) = 1 \quad (24)$$

$G_m(x, y)$ 為高斯濾波器, Y 為影像的亮度資訊, $\omega(\sigma_m)$ 為權重。

5-6-1 IMSR 之模擬結果

1. 低亮度圖形

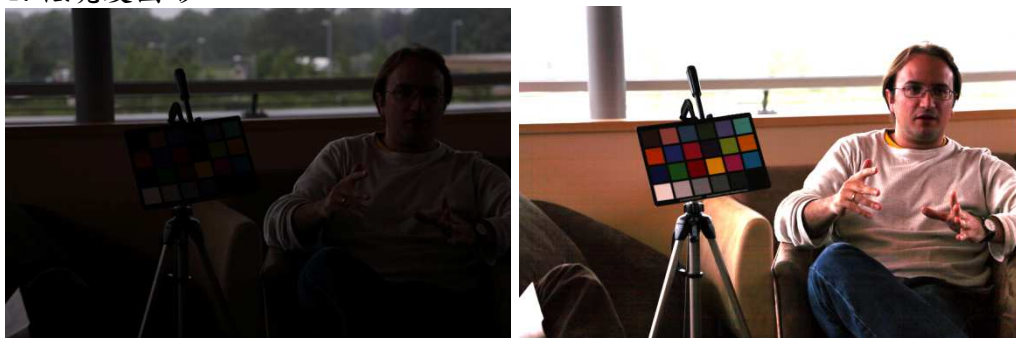


圖 5-31 原始圖形

圖 5-32 經過 IMSR 後圖形

2. 夜間圖形



圖 5-33 原始圖形



圖 5-34 經過 IMSR 後圖形

5-6-2 Integrated Multi Scale Retinex (with blur filter)方法

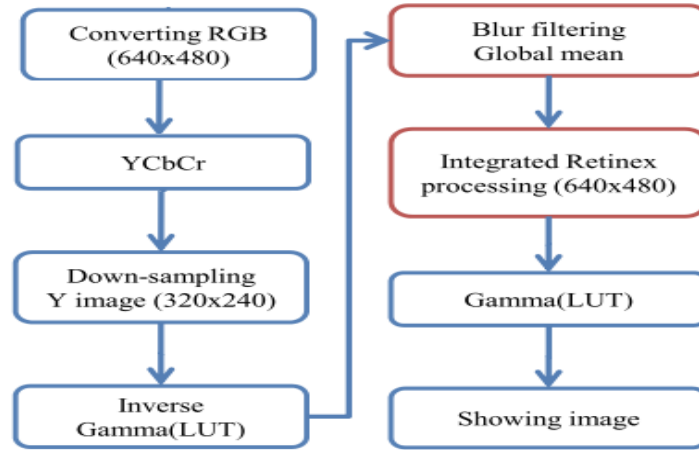


圖 5-35 IMSR 簡化版的流程圖

因為 IMSR 所需的計算時間過長，進而衍生出此方法。本方法採用一個縮小 2 倍的影像當作過濾的樣本並且用 YCrCb 得出亮度樣本，對此亮度樣本作模糊濾波(Sblur)，此模糊濾波(Sblur)採用的是 3*3 的模糊濾鏡以便減少計算時間，之後再計算整張影像的亮度平均值(GM)，下一步再把模糊影像加全域平均值(GM)得到模糊影像(SGM)，之後利用 IMSR 來得到所要的結果。

下列是簡化版 IMSR 的公式

$$S_{\text{blur}}(x, y) = B(x, y) \otimes Y_D(x, y) \quad (25)$$

$$GM = \frac{\sum_{n=0}^N \sum_{m=0}^M S_{\text{blur}}(n, m)}{NM} \quad (26)$$

$$SR_{\text{BM}}(x, y) = A \frac{I_i(x, y)}{SGM_{\text{blur}}(x, y)} \quad (27)$$

$$SGM_{\text{blur}}(x, y) = w_1 S_{\text{blur}}(x, y) + w_2 GM \quad (28)$$

因為本方法會使用到伽瑪校正以計算每一像素是很花時間，為了減少程式執行時間我們可以事先做出查找表來減少所需時間。

Y values	Gamma values	iGamma values
0	0	0
1	12	0
2	21	0
3	28	0
4	33	1
⋮	⋮	⋮
253	254	1005
254	254	1014
255	254	1024

圖 5-24 gamma correction 查找表

5-6-3 IMSRBF 之模擬結果

1. 低亮度圖形



圖 5-25 原始圖形



圖 5-26 經過 IMSRBF 後圖形

2. 夜間圖形



圖 5-27 原始圖形



圖 5-28 經過 MSRBF 後圖形

第六章 自適應局部綜合方法的彩色圖像非線性增強

6-1 自適應局部綜合方法之相關知識[8][11][12]

此方法目標是在低亮度下數位圖像增強，該演算法綜合自適應和鄰里依賴方式的非線性增強(AINDANE)，此兩方式分別為自適應亮度增強用於非線性全圖強度轉換，另一方式為自適應對比度增強用於調整每個像素的強度與相鄰像素。

而自適應亮度增強為亮度信息處理和動態範圍壓縮，自適應對比度增強為保存的細節和逼近與原始圖像的色調。

此兩方式的結合可以使圖像因亮度調整而導致色彩失衡後為了使圖像的色彩不失去原本色彩所以使用自適應對比度增強來保留原始圖像的色調。

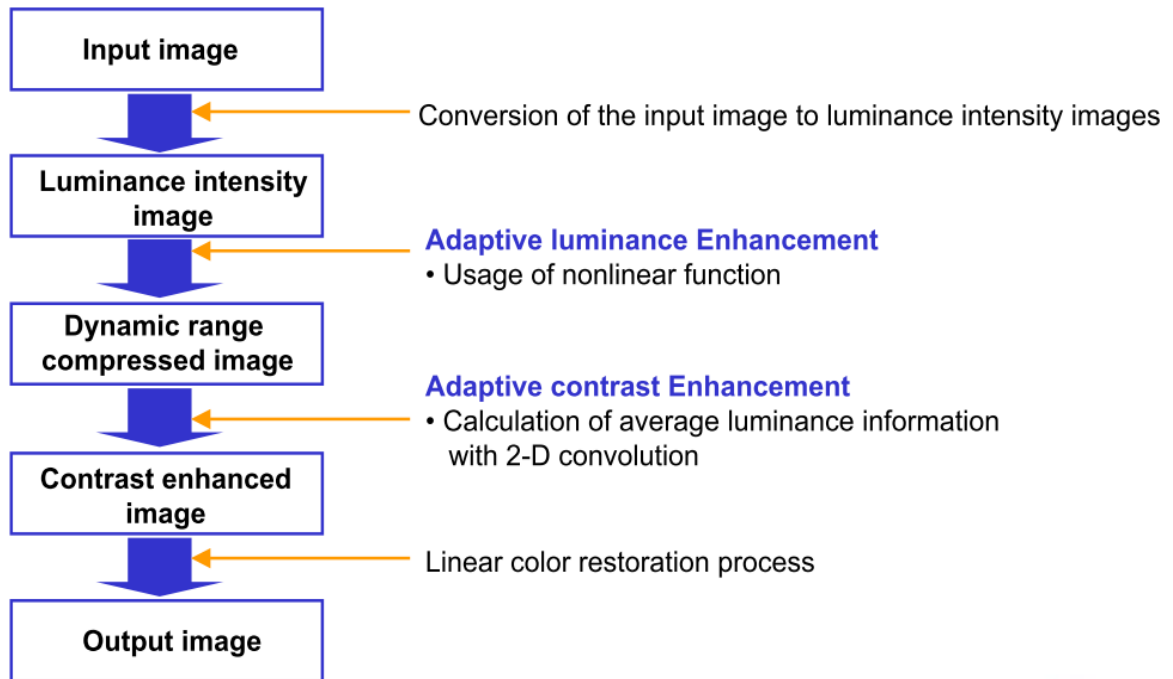


圖 6-1 流程圖

第一部分為了使圖像亮度增強但我們不希望在高亮度區域也曾強太多所以我們使用自適應亮度增強方法來改善此問題：

亮度信息轉換方法：

$$I(x, y) = \frac{76.245I_R(x, y) + 19.685I_G(x, y) + 29.071I_B(x, y)}{255} \quad (29)$$

$$\text{正規化: } I_n(x, y) = \frac{I(x, y)}{255} \quad (30)$$

$$\text{增強亮度強度的動態範圍壓縮: } I'_n = \frac{I_n^{(0.75z+0.25)} + (1-I_n)0.4(1-z) + I_n^{(2-z)}}{2} \quad (31)$$

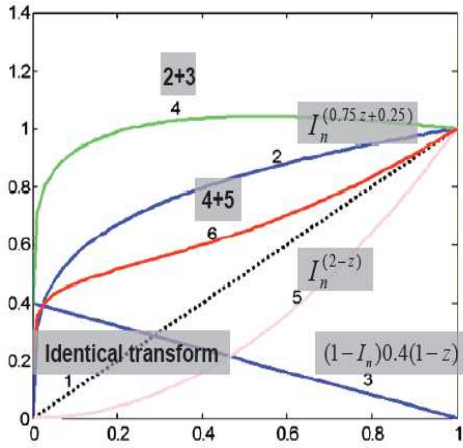


圖 6-2 亮度動態範圍壓縮圖

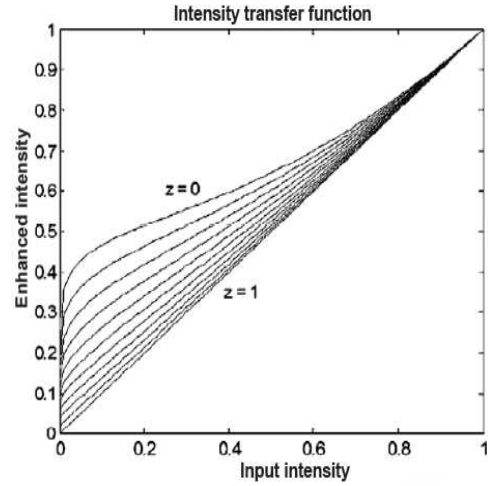


圖 6-3 Z 指數不同亮度動態圖

$$z = \begin{cases} 0, & \text{for } L \leq 50 \\ \frac{L-50}{100}, & \text{for } 50 < L \leq 150 \\ 1, & \text{for } L > 150 \end{cases} \quad (32)$$

Z 是圖像依賴參數

L 是對應於 0.1 的 CDF 的強度水平

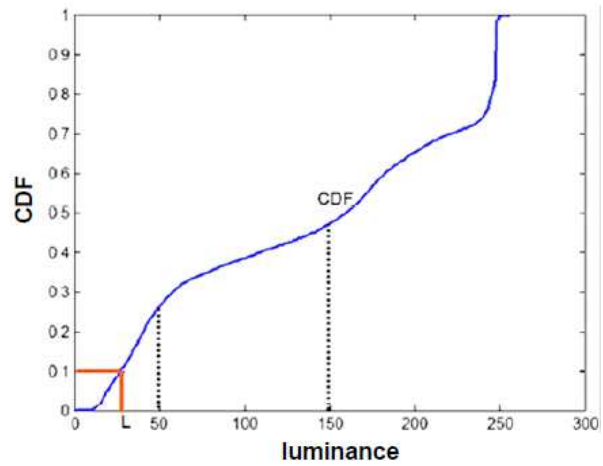


圖 6-4 亮度累積分布圖

第二部分因為第一部分亮度調整過後使得圖像的色彩有所偏差為了使色彩逼近原始圖像所以我們使用自適應對比度增強來改善第一部分的色彩問題：濾波方式我們採用高斯濾波：

$$G_1(x, y) = K \exp \left[\frac{-(x^2 + y^2)}{c_1^2} \right] \quad (33)$$

其中 c_1 是不同尺度

對圖像做二維離散空間卷積：

$$I_{conv,1}(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(m, n) G_1(m + x, n + y) \quad (34)$$

找尋圖像對比度增強的像素強度 $E_1(x, y)$ ：

$$E_1(x, y) = r_1(x, y)^p = \left[\frac{I_{conv,1}(x, y)}{I(x, y)} \right]^p \quad (35)$$

p 是一個圖像相關的參數用來調整的對比度：

$$p = \begin{cases} 3 & \text{for } \sigma \leq 3 \\ \frac{27-2\sigma}{7} & \text{for } 3 < \sigma < 13 \\ 1 & \text{for } \sigma \geq 13 \end{cases} \quad (36)$$

σ :是全域的標準偏差

中心環繞對比度增強: 中心的像素值和卷積結果的比較，拉了較高的中心像素值高於平均水平，反之亦然

$$S_i(x,y) = 255I'_n(x,y)^{E_i(x,y)} \quad (37)$$

$$S(x,y) = \sum_i w_i S_i(x,y) \quad (38)$$

w_i 是為每一個輸出的權重因子 $w_i = 1/n, i = 1, 2, \dots, n$

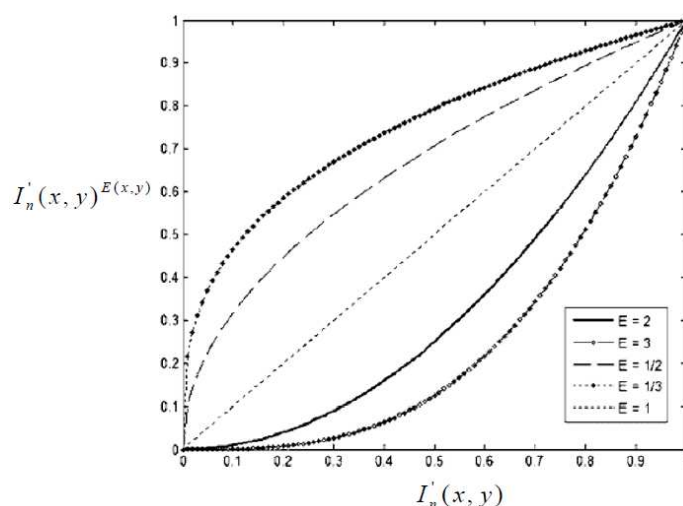


圖 6-5 不同的權重因子曲線圖

色彩還原: 線性色彩增強的彩色圖像還原過程

$$S_j(x,y) = S(x,y) \frac{I_j(x,y)}{I(x,y)} \lambda_j \quad (39)$$

$j=r, g, b$ 表示的 R, G, B 的光譜帶

S_r, S_g, S_b 是增強的彩色圖像的 RGB 值

λ : 手動調整的增強的彩色圖像的色調。常數，並小於但接近 1

6-2 自適應局部綜合方法之模擬結果

1. 低亮度圖形



圖 6-6 原始圖形



圖 6-7 經過自適應局部後圖形

2. 夜間圖形



圖 6-8 原始圖形



圖 6-9 經過自適應局部後圖形

第七章 有效的快速算法增強光線較暗的圖像方法

7-1 有效的快速算法增強光線較暗的圖像之相關知識 [10]

這篇方法主要的概念是對一張很暗的影像對他做 inverse，再對這張 inverse 的影像做除霧的動作，最後再反向回來達到增強影像亮度的效果。這個概念源自於作者發現，暗的影像經過反相後，某些特性跟一張有霧的影像相似。

為何要採取此方法是因為低亮度圖像經 inverse 後會變成大部分都是白色的影像，我們假設白色的影像是霧氣所遮罩若去除霧氣則得到沒有霧氣的影像，最後再把結果導致就可以得到原本影像的調整之後的圖像。

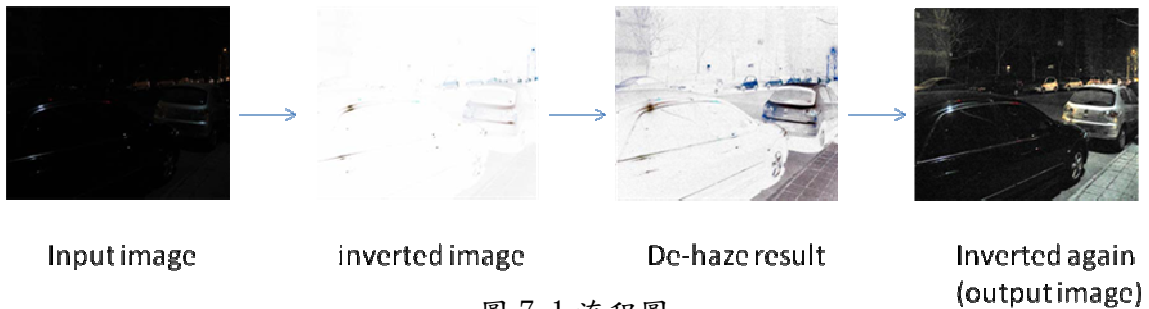


圖 7-1 流程圖

接下來將會分下面幾個部分對此篇演算法做介紹：

- (i) 低光源的inverse影像和有霧影像的相關性
- (ii) 除霧演算法

(i) 低亮度的inverse影像和有霧影像的相關性



圖7-2 低亮度的inverse影像和有霧影像相關圖

圖7-2中，上排是一般低亮度的影像，中間是inverse之後的結果，而最後一

排是有霧的影像。

從圖中可以稍微發現到inverse影像中白色的影像就好比霧氣，這是人眼的直覺感官，接著以影像的像素值直方圖來顯示它的相似性：

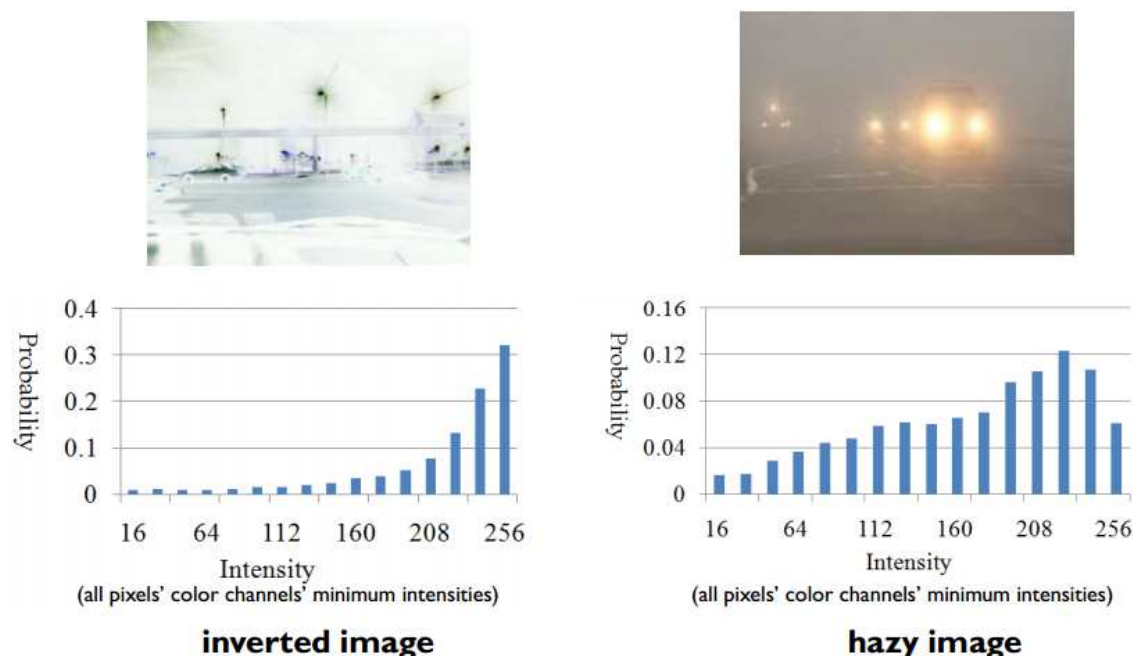


圖7-3

從圖7-3中會發現，無論是有霧的影像或是inverse的影像，大部分的像素值會分布在像素值高的地方，因為白色的部分占據了大部分的影像。

(ii) 除霧演算法

在介紹演算法之前必須先了解一個重要的基礎觀念 “dark channel”。輸入的一張彩色影像，每個像素都有三個通道RGB分別的值，而dark channel是利用一個window遮罩，先window內每個像素點三個channel中最小的值當作每個像素點的強度值，再把window內最小的值當作是中心的值，以此類推找出原本影像中每個像素點所對應的dark channel圖。

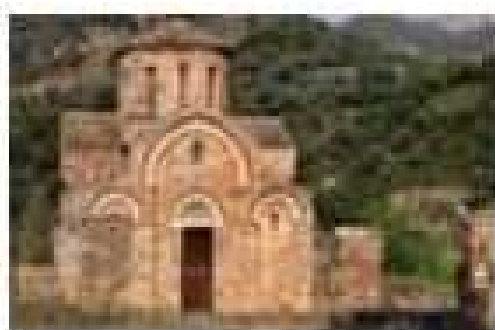


圖 7-4 原始圖形



圖 7-5 取 RGB 之間的最小值圖形

上圖中，左上影像是一般清晰的影像，右上則是一張有霧的影像，左下跟右下則分別是他們的dark channel結果。從中我們可以發現到，左邊影像的dark

channel大不分是低像素值，而右邊影像再有霧的部分則多是高像素的值。這是因為在有霧的影像中，白色霧所占的像素值都偏高，即使分別取三個channel的最小值，像素值還是很大。dark channel的公式如下所示：

$$J^{\text{dark}}(\mathbf{x}) = \min_{c \in \{r, g, b\}} \left(\min_{y \in \Omega(\mathbf{x})} (J^c(y)) \right) \quad (40)$$

J : haze-free image

$\Omega(\mathbf{x})$: a **local patch** centered at $\mathbf{x}$

接下來介紹除霧演算法的基本model模型：

$$R(\mathbf{x}) = J(\mathbf{x})t(\mathbf{x}) + A(1 - t(\mathbf{x})) \quad (41)$$

公式中， I 指的是原本有霧的影像， J 則是除霧後清晰的影像， A 指的是大氣光， t 則是霧的介質參數(霧化程度)， $\mathbf{x}$ 指的是影像中不同的像素點。從公式中可以看出，影像中每個像素點都是由 $t(\mathbf{x})$ 去決定它的霧化程度，如果某一點的霧化越嚴重，那麼它的 t 勢必會很小， A 大氣光對於他的影像會占多數，而原本清晰的影像所占的比重相對會減少，因此會有霧化模糊的情形。

在除霧的模型中，我們要求的是 J (清晰的影像)，而 A (大氣光)和 t (霧的介質參數)是我們不知道所必需求的。因此以下將會介紹兩個部分：

- (i) 求出整張影像的 A
- (ii) 求出每個像素點的 t

(i) 求出整張影像的 A

求 A 有不同的求法，其中一種是直接取整張影像中最亮的點當作是大氣光，但是這麼做的缺點是，如果影像中包含白色的車子，或任何白色的物體，那麼 A 值便錯誤，因此在這邊是利用前述的dark channel圖來取得，如下圖7-4所示：

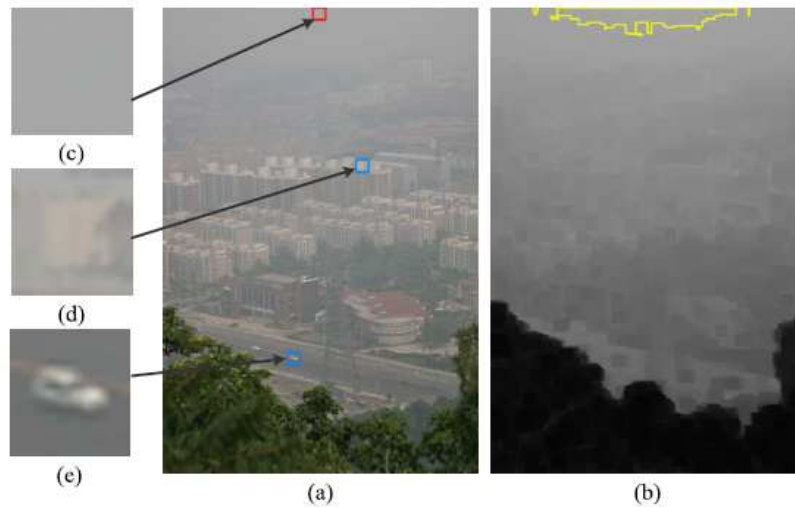


圖 7-6 大氣光選取方式比較圖

(a) 輸入的圖像。(b)黑暗的通道和最陰霾不透明的地區。(c)修補程序我們的方法自動獲取大氣光。(d)及(e):這兩個補丁，包含像素的亮度比空氣亮。

那為什麼要使用 J^{dark} 的通道？

在原始圖像中，最亮的像素一輛白色轎車或一幢白色的建築物，因此所得出的大氣光亮度是不對的，所以才使用 J^{dark} 通道來找尋大氣光。

(ii) 求出每個像素點的 t

$$\min_{y \in \Omega(x)} (I^c(y)) = \tilde{t}(x) \min_{y \in \Omega(x)} (J^c(y)) + (1 - \tilde{t}(x))A^c \quad (42)$$

上述的公式是以除霧演算法的基本模型變換而來，不同的是這邊先以 local patch 來看，並將兩邊分別取 channel 中最小值，而 C 指的是分別的三個 channel。接下來便逐一的推導：

$$\min_{y \in \Omega(x)} \left(\frac{I^c(y)}{A^c} \right) = \tilde{t}(x) \min_{y \in \Omega(x)} \left(\frac{J^c(y)}{A^c} \right) + (1 - \tilde{t}(x)) \quad (43)$$

上述公式中將兩邊的三個 channel 同除以 channel 中個別的 A (大氣光)：

$$\min_c \left(\min_{y \in \Omega(x)} \left(\frac{I^c(y)}{A^c} \right) \right) = \tilde{t}(x) \min_c \left(\min_{y \in \Omega(x)} \left(\frac{J^c(y)}{A^c} \right) \right) + (1 - \tilde{t}(x)) \quad (44)$$

再將兩邊取 local patch 的最小值，最後我們可以利用 dark channel 的特性：

$$J^{\text{dark}}(x) = \min_c \left(\min_{y \in \Omega(x)} (J^c(y)) \right) = 0 \quad (45)$$

再介紹 dark channel 的概念時有描述過，對一張清晰的影像，它的 dark channel 圖每點像素值都很小，趨近於 0，因此這邊便可以假設這一項等於 0，最後公式會得出：

$$\tilde{t}(x) = 1 - \min_c \left(\min_{y \in \Omega(x)} \left(\frac{I^c(y)}{A^c} \right) \right) \quad (46)$$

因此每一點的 t 便可以用上述公式求出，此外為了讓影像在除霧後看起來比較自然，需要在可以在上述公式中加上一個參數 ω 控制：

$$\tilde{t}(x) = 1 - \omega \min_c \left(\min_{y \in \Omega(x)} \left(\frac{I^c(y)}{A^c} \right) \right) \quad (47)$$

ω 值是介於 0 和 1 之間的數值。最後再將處理後的影像在做一次 inverse 反向回來，便可以得到亮度增強後的結果。

最後把上式帶回原式得出：

$$J(x) = \frac{R(x)-A}{t(x)} + A \quad (48)$$

然而，我們發現直接應用上式方程式可能導致提高低照明領域。對進一步優化 $t(x)$ 的計算，我們專注於加強該地區的利益，如房屋，車輛等，同時避免加工背景，例如天空地區在低照明影片。 $t(x)$ 自適應，同時要調整保持其空間的連續性。所以，所得到的視頻變得光滑視覺上，我們引入了一個乘數 $P(x)$ ，並通過了廣泛的實驗，我們發現 $P(x)$ 可被設置為：

$$J(x) = \frac{R(x)-A}{P(x)t(x)} + A \quad (49)$$

$$P(x) = \begin{cases} 2t(x), & 0 < t(x) < 0.5 \\ 1, & 0.5 < t(x) < 1 \end{cases} \quad (50)$$

7-2 有效的快速算法增強光線較暗的圖像之模擬結果

1. 低亮度圖形

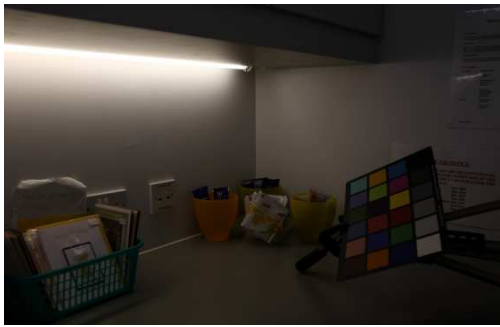


圖 7-3 原始圖形



圖 7-4 經過快速增強光線後圖形

2. 夜間圖形



圖 7-5 原始圖形



圖 7-6 經過快速增強光線後圖形

第八章 Android 手機實做

我們將影像強化的技巧與方法運用在手機上的 Android 系統，設計出一個能強化相片的功能，將所學的影像強化技巧做成實體可運用在手機上的實體程式。

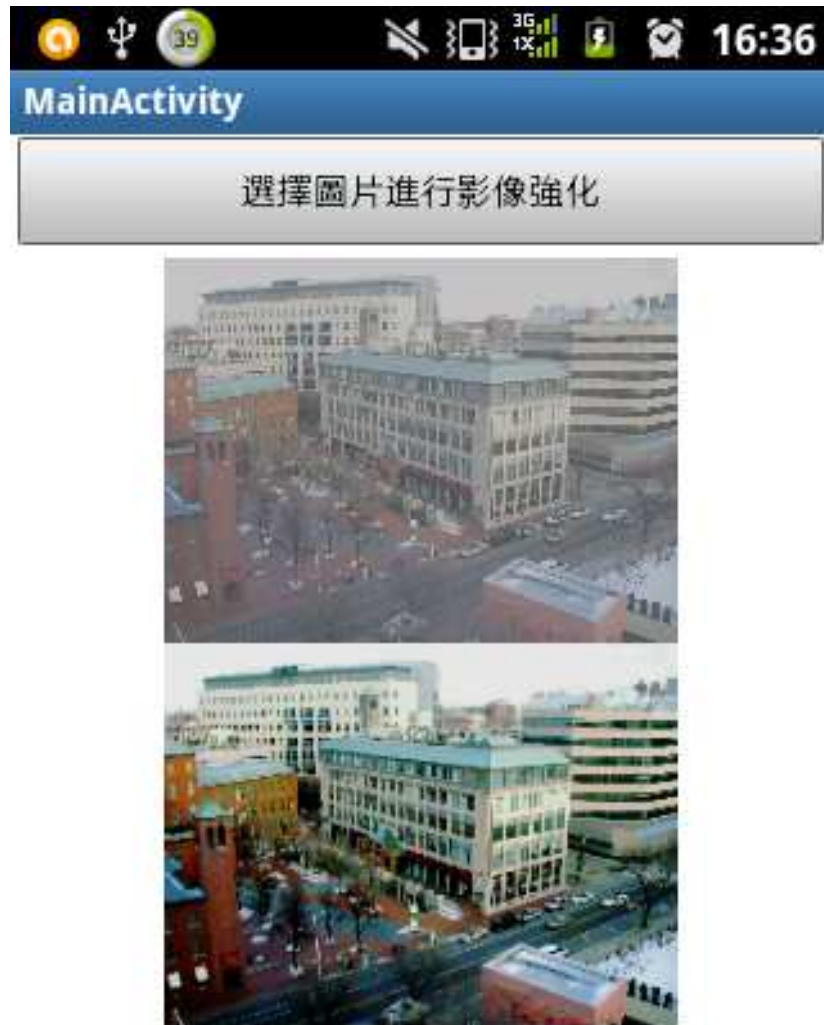


圖 8-1 手機程式圖

第九章 結論

用於低亮度影像時的各個方法都有不錯的能見度，但是用於在夜間影像時因原本圖形所擁有的色彩不夠豐富因此伽瑪校正、直方圖均衡化、直方圖均衡化、Single Scale Retinex、Multi Scale Retinex、Multi Scale Retinex with color Restoration 等方法所產生的結果圖像只是把亮度拉高沒有顧及到圖像的色彩方面所以產生了色彩錯亂的圖像，因此後來演變成 Integrated Multi Scale Retinex 方法為了保持色彩的平衡在夜間處理上有了極大的進步，但色彩還是不盡善，因此又發展出 gamut mapping 使得色彩平衡更完善。

而自適應局部綜合方法、有效的快速算法增強光線較暗的圖像方法，對於夜間影像有著更好的處理方法，它們對於亮度增強後 RGB 色彩繪因此偏移掉所產生的問題各自有不同的解決方式，而每一種方法改善的地方和程度是不一樣的，只有適不適合目前需求，沒有最完美的方法，因此，經過這上述的研究我們發現 IMSR、自適應局部綜合方法、有效的快速算法增強光線較暗的圖像方法對於夜間影像來說是最符合低亮度影像強化的需求。

參考文獻

- [1] Dae-Chul Kim, Wang-Jun Kyung and Yenong- Ho Ha, “Real-time Multi-scale Retinex to Enhance Night Scene of Vehicular Camera” , *Frontiers of Computer Vision (FCV)*, 2011 17th Korea-Japan Joint Workshop on Date of Conference: 9-11 Feb.2011.
- [2] 盧俊良, “Face Recognition Under Illumination and Facial Expression Variation” , 國立中央大學資訊工程研究所碩士論文.
- [3] Zia-ur Rahman, Daniel J. Jobson, Glenn A. Woodell, “Retinex processing for automatic image enhancement” , *College of William & Mary, Department of Computer Science, Williamsburg, VA* 23187.
- [4] Edwin H. Land, “An alternative technique for the computation of the designator in the Retinex theory of color vision” , *in Proceedings of the National Academy of Science*, 83, pp. 3078-3080, 1986.
- [5] Brian Funt, Kobus Barnard, Michael Brockington, and Vlad Cardei, “ Luminance-based multi-scale Retinex” , *Proceedings AIC Colour 97 Kyoto 8th Congress of the International Colour Association*, May 1997.
- [6] Edwin H. Land, “ The Retinex” , *American Scientist*, Vol. 52, pp. 247-264, 1964.
- [7] Edwin H. Land, “The Retinex theory of color vision” , *Proceedings of The Royal Institution of Great Britain*, 47, pp. 23-58, 1974.
- [8] Li Tao and Vijayan K. Asari, “ Adaptive and integrated neighborhood dependent approach for nonlinear enhancement of color images Journal of Electronic Imaging” , *Journal of Electronic Imaging* , Vol. 14(4), Oct. Dec. 2005.
- [9] Wang-Jun Kyung, Tae-Hyoung Lee, Choul-Hee Lee, and Yeong-Ho Ha, “ Improved color reproduction based on CIELAB color space in integrated multi-scale retinex” , *Proc. of SPIE-IS&T Electronic Imaging SPIE Vol. 7241*, 724105.
- [10] Xuan Dong, Guan Wang, Yi (Amy) Pang, Weixin Li, Jiangtao (Gene) Wen, Wei Meng, Yao Lu , “Fast Efficient Algorithm for Enhancement of Low Lighting Video ”, *Department of Computer Science, Tsinghua University, Beijing, China*.
- [11] Osman Nuri UÇAN, Capt. Ender ÖĞÜŞLÜ, “A non-linear technique for the enhancement of extremely non-uniform lighting images” , *JOURNAL OF AERONAUTICS AND SPACE TECHNOLOGIES JUNE 2007 VOLUME 3 NUMBER 2*(37-47).
- [12] Hau Ngo, Li Tao, Ming Zhang, Adam Livingston and Vijayan Asari “A Visibility Improvement System for Low Vision Drivers by Nonlinear Enhancement of Fused Visible and Infrared Video” , *Computational Intelligence and Machine Vision Laboratory Department of Electrical and Computer Engineering , Old Dominion University, Norfolk, VA* 23529.

- [13] Hee-won Lee, Se-Jung Yang, Byung-Uk Lee, “Color compensation of histogram equalized images” , *Dept. of Electronics Engineering, Ewha W. University, Seoul* 120-750, *Korea*