第四章 彩色影像處理軟體演算法

4.1 自動色階演算法

自動色階的演算必定會相依於景物的內容以及亮度層。如下圖所示，由兩圖可以觀察到，從左圖的色階分佈到右圖的色階分佈確實是將整個影像的色階都拉扯開了，而其改變是線性的變化。

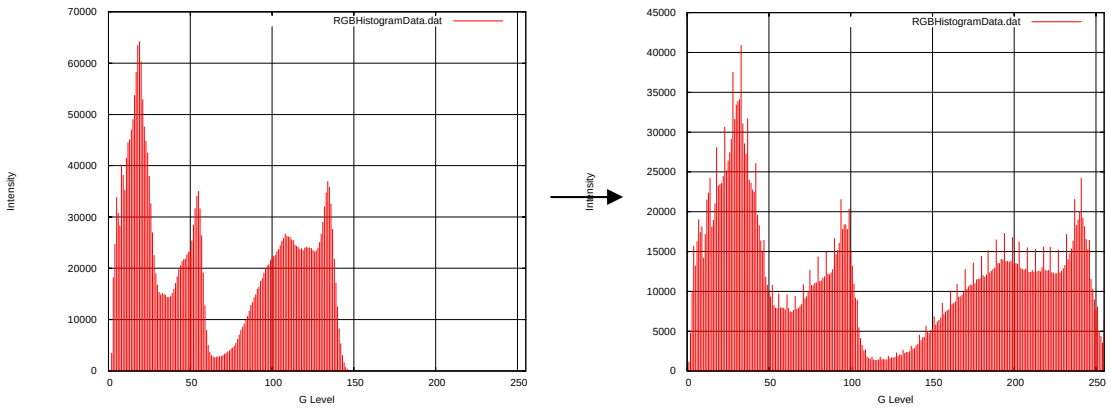

圖 4-1 自動色階處理前後的色階分佈

以下便自動色階的演算法步驟：

（a） 計算整張原始資料影像中所有三元素的統計圖 $H_i(0 \le i \le \Phi)$。

（b） 找尋控制點 α 以及 β 並使它們滿足下式（4-1），其找尋方式便是分別累加整張影像中從最暗與最亮的前百分之零點一以及百分之九十九的點，在兩控制點間的動態範圍便拉撐到整個系統的數值呈現範圍，當系統為 8 位元時，Ω 為 255，系統為 10 位元時，則 Ω 為 1023，依此類推。

$$\left(\sum_{i=0}^{\alpha-1} H_i\right)/\Omega \le 0.001 \le \left(\sum_{i=0}^{\alpha} H_i\right)/\Omega$$
$$\left(\sum_{i=0}^{\beta-1} H_i\right)/\Omega \le 0.999 \le \left(\sum_{i=0}^{\beta} H_i\right)/\Omega \qquad (4\text{-}1)$$

（c） 針對每個像素的原始 *RGB* 數值拉撐為新值 V^{STH}，於是將整張影像上所有的點都套用到自動色階的數學式（4-2）中。

$$V^{STH} = \begin{cases} (V - \alpha^*) \times (\Phi / (\beta - \alpha)) & \alpha \neq \beta \\ V & \alpha = \beta \end{cases} \quad (4\text{-}2)$$

（d） 當兩個控制點被計算出來後，若影像中有小於控制點 α 時，則將其值直接指定為影像數值系統中最小的值，即 0。反之，若影像中有大於控制點 β 時，則將其值直接指定為影像數值系統中最大的值。

4.2 自動白平衡演算法

本論文提出的自動白平衡的演算法和過去文獻的比較，表現較好的原因在於：

（a） 該方法校正其自動色溫方程式時完全考慮該感應器之特性。

（b） 為了防止大色塊會左右自動白平衡的計算，其演算過程，有做邊緣偵測（16×16 區塊）而不是單純地抽取出像素的邊緣，如此一來，區塊上的雜訊會較少。

（c） 其計算速度較 Color By Correlation 演算法來得快速[10]，該方法同樣也是有考慮到感應器的特性，同樣是一個相當成功的方法。

以下便是完整的自動白平衡的細節演算法步驟，其輸入一張原始影像資料，輸出為經過自動白平衡校正後的影像資料。

（a） 將原始影像分割成二維的十六乘十六區塊，$M_{i,j}(1 \le i \le m,\ 1 \le j \le n)$，如下圖所示。

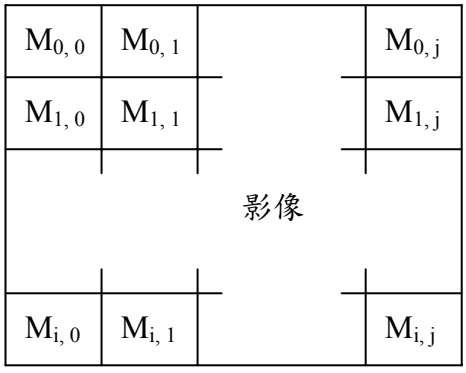

圖 4-2 切割成十六乘十六的區塊

（b） 對每個區塊$M_{i,j}$計算其紅（$R_{i,j}$），綠（$G_{i,j}$），藍（$B_{i,j}$）的平均數值。

（c） 對每個區塊$M_{i,j}$計算其顏色的差異性$D_{i,j}$ 。

$$D_{i,j}=\frac{(R_{i,j}-R_{i,j-1})^2+(G_{i,j}-G_{i,j-1})^2+(B_{i,j}-B_{i,j-1})^2}{(\frac{R_{i,j}+R_{i,j-1}}{2})^2+(\frac{G_{i,j}+G_{i,j-1}}{2})^2+(\frac{B_{i,j}+B_{i,j-1}}{2})^2} \quad (4\text{-}3)$$

（d） 對每個區塊$M_{i,j}$設立一邊界旗標$F_{i,j}$，θ為一事先定義之臨界值。

$$F_{i,j}=\begin{cases}1, F_{i,j-1}=1 \quad D_{i,j}\geq\theta \\ 0 \quad\quad D_{i,j}<\theta\end{cases} \quad (4\text{-}4)$$

如下圖便是數學式的示意圖，其中，L、R便是表示左右相鄰的區塊，而其式（4-4），計算所產生的可能性則會有下列四種情況，而各符號所代表的意涵如下說明。

（1） R 代表的是$D_{i,j}$ ，L 代表的是$D_{i,j-1}$ 。

（2） 白色方塊便是代表$D_{i,j}\leq\theta$ ，意即非邊緣區塊；反之亦然，條紋方塊便是代表$D_{i,j}\geq\theta$ ，意即邊緣區塊。

（3） 0 代表的是$F_{i,j}=0$ ，亦即該區塊並不會列入平均權重的考慮內；反之亦然，1 代表的是$F_{i,j}=1$ ，亦即該區塊會列入平均權重的計算內。

（Ⅰ） （Ⅱ） （Ⅲ） （Ⅳ）

圖 4-3 邊界區塊示意圖

其中比較值得一提的是如上圖（Ⅱ）的個案，由於當右邊的區塊被計算出$D_{i,j}\geq\theta$時，意即該區塊為邊緣區塊時，也就是該區塊會列入平均權重的計算內，根據式（4-4），此時，其左邊的區塊也要被列

入平均權重的計算內。

（e） 計算其紅色平均權重(WR)、綠色平均權重(WG)、藍色平均權重(WB)的數值。

$$WR = \sum_{i=1,j=1}^{i=m,j=n} F_{i,j} R_{i,j} / \sum_{i=1,j=1}^{i=m,j=n} F_{i,j}$$
$$WG = \sum_{i=1,j=1}^{i=m,j=n} F_{i,j} G_{i,j} / \sum_{i=1,j=1}^{i=m,j=n} F_{i,j} \quad (4\text{-}5)$$
$$WB = \sum_{i=1,j=1}^{i=m,j=n} F_{i,j} B_{i,j} / \sum_{i=1,j=1}^{i=m,j=n} F_{i,j}$$

（f） 計算 $\log(WG/WR)$ 以及 $\log(WG/WB)$ 。

（g） 從 $\log(WG/WR)$ ，$\log(WG/WB)$ 找尋其投影點（R_p，B_p）以便校正色溫曲線係數，如下圖所得知，經由 $\log(WG/WR)$ 以及 $\log(WG/WB)$ 所座落在該平面的點A，而P點即是被投射在色溫曲線上的修正光源位置點，而 L_{CTC} 即是經由實驗數據所得。

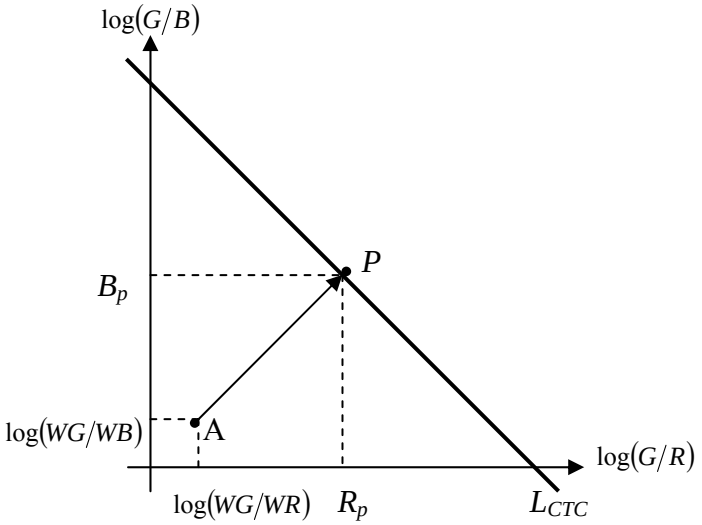

圖 4-4 色溫曲線投影映射光源點

提出的自動白平衡方法乃是經由採集七十張在不同場景以及不同光源下的實驗結果，如下圖為針對特定感應器的物理特性所得到的色溫曲線，其中水平軸為 $\log(G/R)$ ，垂直軸為 $\log(G/B)$ 。

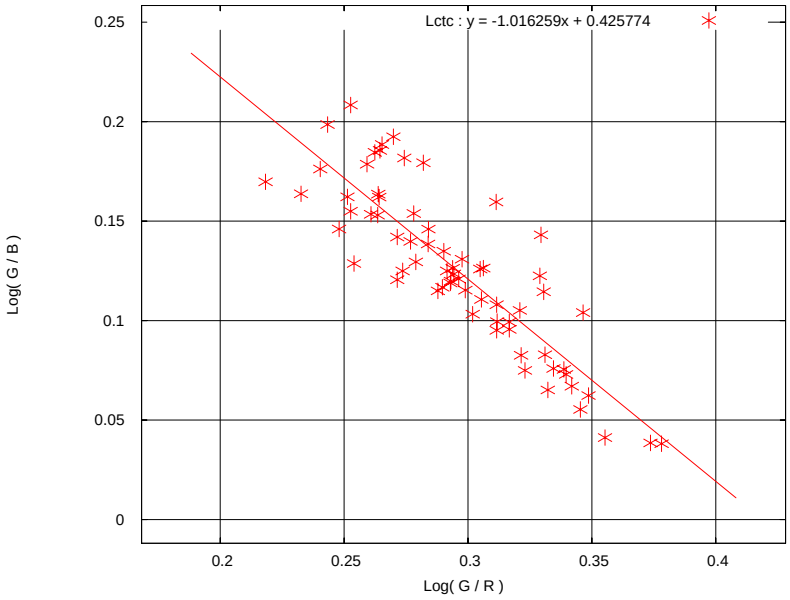

圖 4-5 色溫曲線

而其色溫曲線的求得乃是經由線性迴歸所計算得來的。然而線性迴歸其定義為：若是兩個變數 X 和 Y 會互相影響，則我們可以透過兩個變數間的關係，找到代表此關係的方程式，再藉其一變數及方程式來預測另一變數，而此方法即稱為迴歸分析。式（4-6）便為該直線方程式，β 為其斜率，α 為其截距。

$$Y = \beta X + \alpha \tag{4-6}$$

其中式（4-7）中，$\overline{X}$ 為其 X 的平均值，而 $\overline{Y}$ 便為 Y 的平均值，n 為其資料點的個數。

$$\beta = \frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2} \tag{4-7}$$

得到其 β 之後，就可以經由式（4-8）得到 α 了，該線性迴歸直線方程式便可求得。

$$\alpha = \overline{Y} - \beta\overline{X} \tag{4-8}$$

（h） 針對影像的紅色及藍色的像素分別乘上其係數 $k_R = 2^{R_p}$ 及 $k_B = 2^{B_p}$ 來做

校正。

（i） 在做自動白平衡的演算法時，需要檢查一下是否其輸出結果有超出其色彩空間的邊界值。

4.3 自動白平衡誤差量測

正如同在先前所提到的，在a^*-b^*平面上其開根號距離$\sqrt{(a^*)^2+(b^*)^2}$代表濃度。然而，由於標準色卡上六塊灰階，在*CIELAB*上的色度值理想上應該要為零，也就是說，如果其開根號的值越小，代表影像的自動白平衡對於光源的預測越準確，如下圖所示。

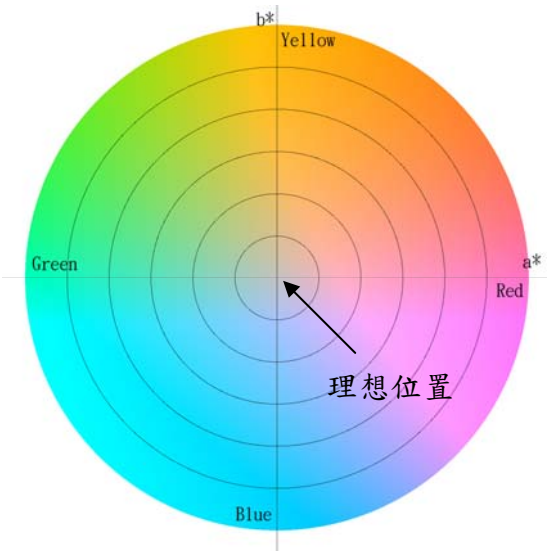

圖 4-6 a^*及b^*色彩空間

4.4 色彩校正演算法

如果原始影像的取得是在標準 D65 的光源下，則標準色卡 MacBeth 上的十八塊色塊及六塊灰塊在 *CIEXYZ* 的色彩空間便是標準的顏色[24]。

（a） 在該 D65 光源下拍攝背景中有標準色卡的場景。

（b） 根據建立在影像處理的考量下，我們可以定義出平均誤差方程式。

（c） 取得最佳化的色彩轉換矩陣M，顏色校正的問題可以經由三乘三的色

彩矩陣M來將$\sum_{i=1}^{18}|MU_i - V_i|^2$最小化，其中$U_i$及$V_i$分別代表第i個色彩區塊的原始輸入值及量測過的數值。

（d） 最後，使用最佳標準的演算法來計算該最小平方問題。如此便可以很輕易地得到轉換後的 sRGB 色彩空間，式（4-9）便為特定影像感應器的三乘三矩陣。

$$\begin{bmatrix} R_s \\ G_s \\ B_s \end{bmatrix} = \begin{bmatrix} 1.67968 & 0.49414 & 0.18554 \\ 0.21289 & 1.81250 & 0.59960 \\ 0.03906 & 0.56640 & 1.52734 \end{bmatrix} \begin{bmatrix} R_{sensor} \\ G_{sensor} \\ B_{sensor} \end{bmatrix} \quad (4\text{-}9)$$

（e） 進行色彩校正時需要防範是否其輸出有超出其色彩空間的邊界值。

4.5 色彩空間轉換演算法

色彩空間*CIELAB*是建立在色彩視覺模型上，該空間是經由色彩空間*CIEXYZ*轉換為亮度L^*以及兩色彩值a^*及b^*，其*CIEXYZ*到*CIELAB*的轉換式為式（4-10）到式（4-14），其中X_n、Y_n以及Z_n為參考白點的三刺激輸入值，R_s、G_s以及B_s分別為*sRGB*。

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R_s \\ G_s \\ B_s \end{bmatrix} \quad (4\text{-}10)$$

在執行式（4-11）之後，其L^*亮度將可得到，而其值域為 0～100 之間，純正數，沒有負值。

$$L^* = 116 f\left(\frac{Y}{Y_n}\right) - 16 \quad (4\text{-}11)$$

在執行式（4-12）之後，其a^*色度將可得到，而其值域為-86～98 之間，也就是會有負值的可能性。

$$a^* = 500\left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right) \quad (4\text{-}12)$$

在執行式（4-13）之後，其b^*色度將可得到，而其值域為-108～94 之間，

也就是會有負值的可能性。

$$b^* = 200 \times \left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right) \tag{4-13}$$

由式（4-11）到式（4-13）中可以觀察到，三個數學式都有一個共同項，也就是 f（t）的部份，其定義便為式（4-14），而其中 $t > 0.008856$ 便是為了避免有無效色域而產生的判斷式。

$$f(t) = \begin{cases} t^{1/3} & t > 0.008856 \\ 7.787t + \dfrac{16}{116} & t \le 0.008856 \end{cases} \tag{4-14}$$

由 *CIELAB* 轉換為 *sRGB* 的數學式為式（4-15）到式（4-18）定義如下，我們可以和式（4-11）到式（4-13）相互呼應，其實就是反轉換的關係而已。經由式（4-15）的可以取回 *CIEXYZ* 中 *Y* 的值域。

$$Y = \begin{cases} \left(\dfrac{L+16}{116}\right)^3 \times 256 & \dfrac{L+16}{116} > (0.008856)^{\frac{1}{3}} \\ \dfrac{\dfrac{L+16}{116.0} - \dfrac{16.0}{116.0}}{7.787} \times 256 & \dfrac{L+16}{116} \le (0.008856)^{\frac{1}{3}} \end{cases} \tag{4-15}$$

經由式（4-16）的可以取回 *CIEXYZ* 中 *X* 的值域。

$$X = \begin{cases} \left(\dfrac{a}{500} + \dfrac{L+16.0}{116.0}\right)^3 \times 256 & \left(\dfrac{a}{500} + \dfrac{L+16}{116}\right) > (0.008856)^{\frac{1}{3}} \\ \dfrac{\dfrac{a}{500} + \dfrac{L+16.0}{116.0} - \dfrac{16.0}{116.0}}{7.787} \times 256 & \left(\dfrac{a}{500} + \dfrac{L+16}{116}\right) \le (0.008856)^{\frac{1}{3}} \end{cases} \tag{4-16}$$

經由式（4-17）的可以取回 *CIEXYZ* 中 *Z* 的值域。

$$Z = \begin{cases} \left(\dfrac{L+16}{116.0} - \dfrac{b}{200}\right)^3 \times 256 & \left(\dfrac{L+16.0}{116.0} - \dfrac{b}{200.0}\right) > (0.008856)^{\frac{1}{3}} \\ \dfrac{\dfrac{L+16}{116} - \dfrac{b}{200} - \dfrac{16}{116}}{7.787} \times 256 & \left(\dfrac{L+16.0}{116.0} - \dfrac{b}{200.0}\right) \le (0.008856)^{\frac{1}{3}} \end{cases} \tag{4-17}$$

經由式（4-18）的可以取回 *sRGB* 的值域，但要小心的是，在該式的轉換過程中可能會出現負值或者超出 *sRGB* 值域的可能，所以在做色彩空間轉換的演算法時，需要檢查一下是否其輸出結果有超出其色彩空間的邊界值。

$$\begin{bmatrix} R_{sRGB} \\ G_{sRGB} \\ B_{sRGB} \end{bmatrix}_{D65} = \begin{bmatrix} 3.240479 & -1.537150 & -0.498535 \\ -0.969256 & 1.875992 & 0.041556 \\ 0.055648 & -0.204043 & 1.057311 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{D65} \quad (4\text{-}18)$$

4.6 色彩飽和度增強演算法

色彩飽和度增強的目的在於增加色彩的飽和度，但是又必須維持每個像素的色度以及亮度這兩個要素，如同之前所描述，色彩空間 *CIELAB* 中，其數學式開根號距離以及角位置分別代表顏色的濃度及色度，如此一來，便很容易處理顏色的一致性，以及亮度，這是色彩空間如 *sRGB*，*YCbCr* 所沒有的優勢。

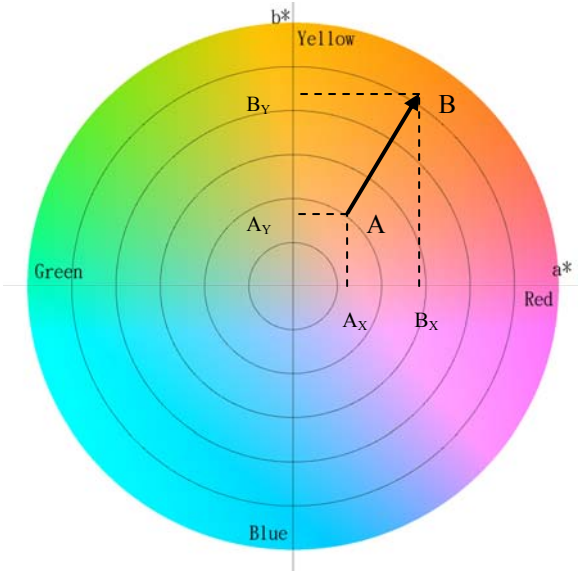

圖 4-7　a^*及b^*色彩空間

之所以可以讓所有的像素用相同的比例去處理，是因為*CIELAB*是幾乎均勻且一致性的色彩空間，以致於歐幾里德距離可以直接表現出顏色的增強效果，一旦增強了每一個像素的濃度，則其色度以及濃度的相對比率就不會有所變化，所以白平衡的結果因此就不會被色彩飽和的處理所影響。如上圖中便可以看到，線段AB座落在a^*-b^*的平面上，而A點便是原來尚未經過色彩飽和度增強的影像處理時的位置，其座標為(A_X, A_Y)；而B點便是已經經過色彩飽和度增強的影像處理時的位置，其座標為(B_X, B_Y)，可以從該a^*-b^*圓得

知，中央圓心的位置顏色最淡，而其越往外層的方向則其顏色強度越深。於是乎，水平軸方向a^*掌管的是紅色及綠色，垂直軸方向b^*掌管的是黃色及藍色。所以在a^*的色彩飽和度增強大小比率便為X_B/X_A，所以在b^*的色彩飽和度增強大小比率便為Y_B/Y_A。在做色彩飽和度增強的演算法時，需要檢查一下是否其輸出結果有超出其色彩空間的邊界值。

4.7 伽瑪校正演算法

伽瑪校正是與影像沒有相依性的操作，它是用來補償螢幕的非線性效應，預設上，我們採用伽瑪值 γ 為 0.45 來針對所有的影像進行處理，並且在硬體製作過程中，伽瑪校正的處理可以採用查表法來加速其計算速度應用於彩色影像處理流程中。因為伽瑪校正的演算式很單純，只有 $RGB = RGB^{\gamma}$，然後由於在處理較暗的像素值域時，如 0～10 之間的色域時，其最暗處會被突然拉起來，如此一來，也可能因此會增加雜訊被放大的可能性，為了能夠希望這樣的轉變不要太劇烈，所以我們將原來的伽瑪校正的數學式做一點小小的變化，如同下式（4-19）所列：

$$\delta = \frac{\Delta X^{\gamma}}{\Delta X}$$
$$RGB' = \begin{cases} RGB^{\gamma} & RGB \geq \Delta X \\ RGB \times \delta & RGB < \Delta X \end{cases} \qquad (4\text{-}19)$$

由於我們採用的 γ 為 0.45，所以，其輸出的影像一定比輸入的影像較為明亮。上式中的 ΔX 的位置，則是伽瑪校正之後變化不要太劇烈的水平終點位置，而 ΔX^{γ} 則是伽瑪校正之後變化不要太劇烈的垂直終點位置。然而，事實上 ΔX 也是*RGB*的原始輸入像素值，ΔX^{γ} 則為經過伽瑪校正之後的*RGB*輸出像素值。而其條件判斷式則是判斷*RGB*的原始輸入像素值是否超過 ΔX，若是否，亦即希望變化不要太劇烈的區域則直接將*RGB*的原始輸入像素值乘上之前計算的 δ 值即可，而其校正後的數學式所產生的伽瑪曲線圖正如下圖

所示，水平軸從最小到最大為 0～1，因為其值被正規化了，因為*RGB*值域的範圍為 0～255，所以，在進行伽瑪校正影像處理時，記得要把所有輸入的*RGB*像素值都先除以 256，如此，才可以代入伽瑪校正的數學式中，而垂直軸當然就是經過伽瑪校正之後的*RGB*輸出值，其值域當然就是 0～255。

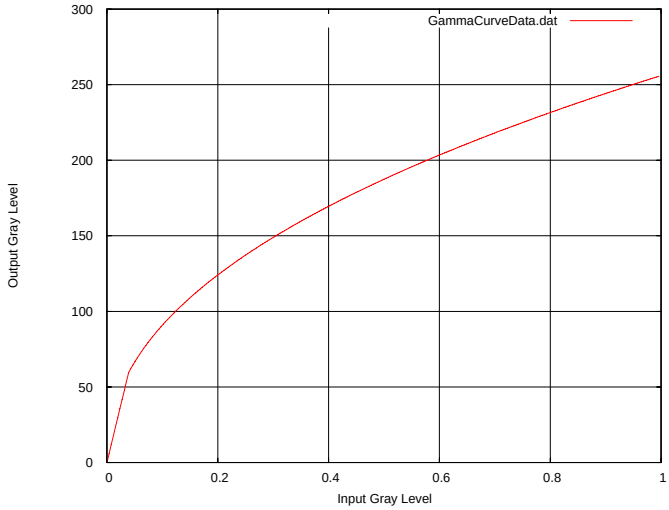

圖 4-8 伽瑪校正曲線

如下圖為伽瑪校正曲線，其伽瑪值範圍從 0.04～25.0 之間，可以觀察到，整個圖形對稱於伽瑪值為 1 時的直線，除此之外，也可以了解到小於 1 的伽瑪值將會使影像變亮；而大於 1 的伽瑪值將會使影像變暗。

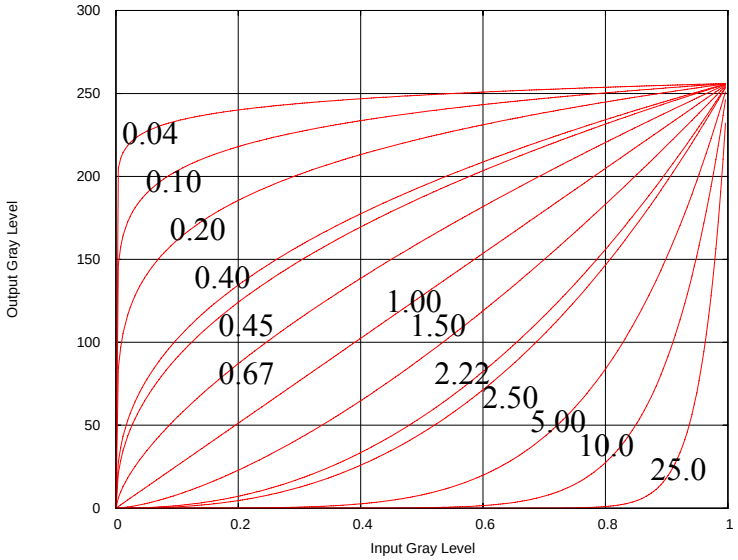

圖 4-9 伽瑪校正曲線，伽瑪值範圍 0.04～25.0