

國立交通大學

資訊科學系

碩士論文

利用 Linear Programming 進行色域對應

Images Dependent Gamut Mapping Algorithm by Linear
Programming

研究生：李杰翰

指導教授：荆宇泰 博士

中華民國九十四年七月

利用 Linear Programming 進行色域對應

學生：李杰翰

指導教授：荊宇泰 博士

國立交通大學資訊科學系



摘要

因為不同的顯示器有不同的顯示能力，所以在顯示彩色圖片時往往會因為顯示設備的能力不足，發生無法顯示需要顏色的問題。在我的論文裡，我將以傳統的色域對應(Gamut Mapping)方法為基礎，利用設計 Linear Programming 問題，希望透過考慮顏色與顏色之間的關係後，能找出最適當的方法來顯示這些顏色

Images Dependent Gamut Mapping Algorithm by Linear Programming

Student : Cheih-Han Lee

Advisor : Dr. Yu-Tai Ching

Institute of Computer and Information Science
National Chiao Tung University



Because every display device has its own display characteristic, it's a popular situation that there are some colors in the image can't be exactly reproduce by a specific display device. In my thesis, I propose a gamut mapping method base on design a linear programming problem to find a good correspondence between colors in different display device.

致謝

謝天謝地



摘要	I
英文摘要	II
致謝	III
圖表目錄	VI
方程式目錄	VII
第一章 導論	1
1.1 簡介.....	1
1.2 術語.....	1
1.3 色域對應的基本流程.....	3
2 第二章 背景及文獻回顧	5
2.1 色彩空間 (Color Space)	5
2.1.1 CIE 1931 XYZ Color Space	6
2.1.2 CIE Uniform 色彩空間	8
2.2 顏色的差異(Color Difference).....	13
2.2.1 色塊間的顏色差異.....	13
2.2.2 圖片間的顏色差異.....	14
2.3 顯示設備的特性.....	14
2.3.1 Cubic Interpolation	15
2.3.2 Polynomial Fitting	16
2.3.3 色域的邊界描述法(Gamut Boundary Descriptor)GBD.....	16
2.4 色域對應.....	17
2.4.1 目標.....	17
2.4.2 傳統常用的色域對應法.....	17
3 第三章 使用Linear Programming進行色域壓縮	20
3.1 利用Linear Programming進行色域對應	20
3.1.1 從最佳化的觀點看傳統的色域對應.....	20
3.1.2 顏色與顏色之間的關係限制.....	22
3.2 實做方法.....	25
3.2.1 (Control Point)	25
3.2.2 控制點的Objective Function.....	27
3.2.3 總結.....	28
4 第四章 實驗與討論	30
4.1 實驗.....	30
4.2 結果討論與未來展望.....	32



圖表目錄

圖 1-1 色域對應的基本流程.....	3
圖 2-1 CIE STANDARD COLORIMETRIC OBSERVER COLOR MATCHING FUNCTION	6
圖 2-2 CIE XY CHROMATICITY DIAGRAM.....	7
圖 2-3 CIE XY CHROMATICITY DIAGRAM裡有相同色彩差異的”色對”	8
圖 2-4 CIE U’V’ CHROMATICITY DIAGRAM裡有相同色彩差異的”色對”	9
圖 2-5 CIE U’V’ CHROMATICITY DIAGRAM.....	9
圖 2-6 CIE COLOR SPACE	10
圖 2-7 CIE LAB COLOR SPACE	12
圖 2-8 利用CUBIC INTERPOLATION內插顏色的特性	15
圖 2-9 GAMUT MAPPING DIRECTION(A) SARA (B) TAYLOR (C) LAIHANEN (D) MACDONALD	18
圖 2-10 MAPPING METHOD.....	19
圖 3-1 LEAST AND MAX COMPRESSION AMOUNT	21
圖 3-2 LINEAR CONSTRAINT	24
圖 3-3 CONTROL POINT	26
圖 3-4 THE DEFINITION OF ALPHA, THETA, AND RADIUS	27
圖 3-5 RESULT OF SEGMENT MAXIMA GBD.....	27
圖 3-6 THE FLOWCHART OF MY GAMUT MAPPING ALGORITHM	29
圖 4-1 EIZO FLEXSCANE L568(右)以及PANASONIC TC-20LA1TN(左)色域在AB平面上的正交投影.....	31
圖 4-2 EIZO FLEXSCANE L568(右)以及PANASONIC TC-20LA1TN(左)色域在LA平面上的正交投影.....	31
圖 4-3 使用傳統LAIHANEN的方法決定顏色壓縮的方向，(左)使用CLIPPING的對應結果，(右)使用LINEAR COMPRESSION的結果。	32
圖 4-4 BEFORE GAMUT MAPPING.....	33
圖 4-5 CLIPPING METHOD	33
圖 4-6 LINEAR COMPRESS METHOD	34
圖 4-7 USE LUMINANCE DIFFERENCE AS CONSTRAINT	34
圖 4-8 USE PROJECTION VECTOR AS CONSTRAINT	35

方程式目錄

(EQUATION 1)	6
(EQUATION 2)	7
(EQUATION 3)	8
(EQUATION 4)	10
(EQUATION 5)	11
(EQUATION 6)	13
(EQUATION 7)	13
(EQUATION 8)	13
(EQUATION 9)	13
(EQUATION 10)	15
(EQUATION 11)	16
(EQUATION 12)	20
(EQUATION 13)	21
(EQUATION 14)	21
(EQUATION 15)	22
(EQUATION 16)	22
(EQUATION 17)	23
(EQUATION 18)	23
(EQUATION 19)	23
(EQUATION 20)	24
(EQUATION 21)	24
(EQUATION 22)	25
(EQUATION 23)	26
(EQUATION 24)	28
(EQUATION 25)	28



第一章 導論

1.1 簡介

我們都知道色彩在人類文化裡的重要性。而從布匹的染色，紙張的印刷，到在 CRT 或 LCD 監視器上顯示圖片，這些不同的顏色顯示”設備”所使用的顯示方法都有所差異，自然的所能產生的顏色也都有不同的特性。因此，如何在各式不同的顯示設備上重現出想要的顏色結果一直以來都是重要研究的主題。

在計算機科學的領域裡，一般提到顏色我們都會使用最簡單的 RGB 表示法來表示。不過相同的 RGB 值在每個不同的顯示設備上會有不同的詮釋方式，最後呈現出來的結果往往也不盡相同。所以若想在不同的顯示設備上都能得到自己期望的顏色，必須有一套方法來對不同顯示設備的 RGB 值進行顏色校正(Calibration)。

假設已知顯示設備對 RGB 的詮釋結果，在我的論文裡，我將嘗試以傳統的色域對應(Gamut Mapping)[1][2]演算法為基礎，再配合我設計的最佳化目標—Objective Function 及線性不等式來進行 Linear Programming，最佳化想要重現的圖片色彩結果。希望經由考慮顯示設備間的顯示能力以及顏色與顏色之間的關係，在不同的顯示設備間提供的 RGB 對應關係，讓我們可以在不同顯示設備上能得到盡量相似的圖片顏色顯示結果。

1.2 術語

在討論色域對應前，我將先定義後面討論所需要用的基本的術語，以利在接下來的討論中清楚描述問題。

圖片(Image)：

圖片為一個二維陣列，陣列中的每個元素(Entry)紀錄該位置 RGB 值。

原始顯示設備(Original Display Device)：

原始顯示設備為正確顏色的參考設備，色域對應的目標是要顯示出和原始顯示設備相似的圖片結果。

重製顯示設備(Reproduction Display Device)：

重製顯示設備代表我們最後希望顯示圖片的設備，可能是 CRT 監視器，或著是彩色印表機等。

色域(Gamut)：

色彩空間裡所有顯示設備所能表現出來的顏色。

色域邊界(Gamut Boundary)：

顯示設備中能表現出的顏色”極限”。

色域邊界描述子 GBD(Gamut Boundary Descriptor)：

用來描述色域邊界的資料結構，提供色域對應時所需要的基本幾何運算。

色調(hue)：

色調代表人們所感知的顏色種類。色調相同代表我們認為他是屬於同一種的顏色，如紅色，藍色，綠色，黃色等等。

色度(Chroma)：

色度代表人們所感知的顏色濃度。色度越高代表顏色的純度越高。色度零則是色域裡完全無色的的灰階軸。

正確性(Accuracy)：

圖片在原始及重製顯示設備間的顏色相似程度。



1.3 色域對應的基本流程

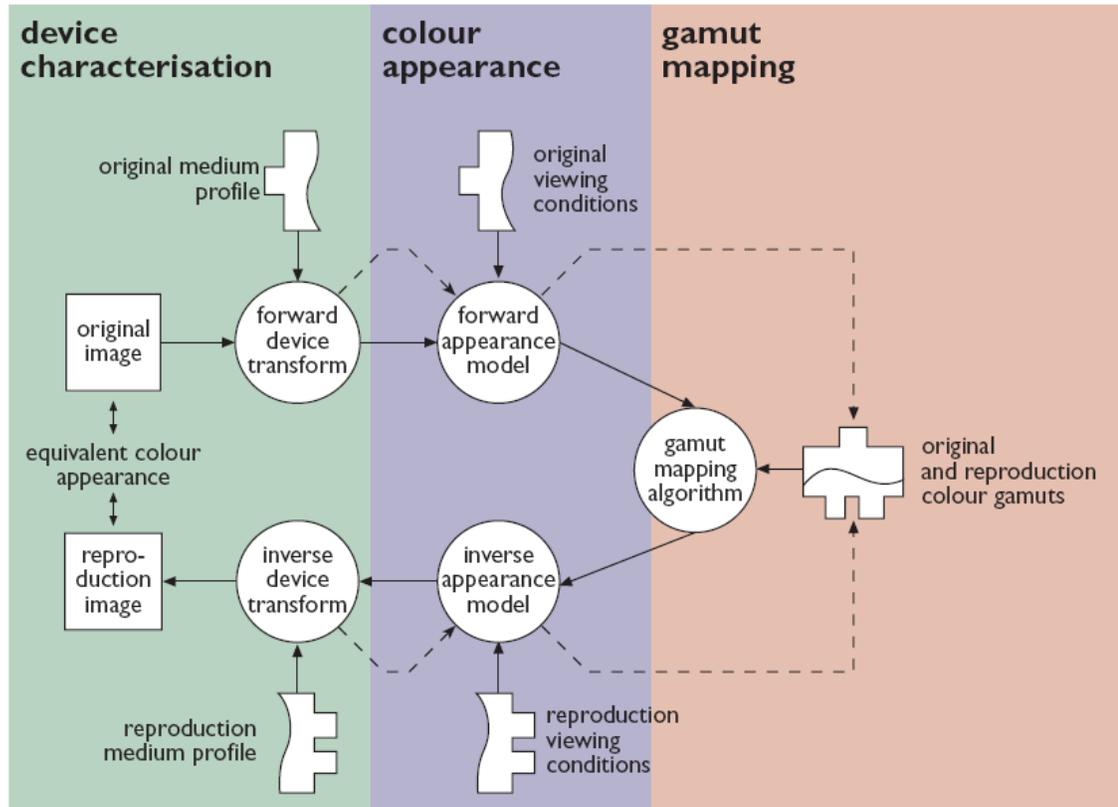


圖 0-1 色域對應的基本流程

色域對應的目的在於求出兩個顯示設備的色域間的對應關係。Figure 1.1 是 Morovic[1]所提出的利用色域對應進行色彩重製的基本流程。色彩重製的流程可以分成三大部分—計算設備的特性(Device-Characterization)，了解顏色的表現結果(Color-Appearance)，及進行色域對應(Gamut-Mapping)。

假設我們希望在重製顯示設備上重現出和原始顯示設備一致的圖片顏色結果，我們需要先了解顯示器如何詮釋它接收到的 RGB 值。因為不同的顯示設備對同一個 RGB 值的表現結果不同，所以我們需要將 RGB 表示法轉換成不會因為顯示設備改變(Device Independent)的顏色表示法。為了這個原因，在計算設備特性(Device-Characterization)的階段裡我們需要建立顯示設備的 RGB 值和新表示法—色彩空間的對應關係。

設備特性的階段得到顏色的表現結果，但是和人類感知的顏色結果可能還有一段差距。這是因為人類對顏色的感知結果跟他所處的觀視環境(Viewing condition)息息相關。所以進行顏色重製另一個要考慮的因素即是顯示設備周圍的觀視環境。

利用上述的兩個部分可以得到原始及重製顯示設備的 RGB 值和人類觀感的對應關係，並知道顯示器所能表現出的顏色範圍—色域。此時若所有想表現的顏色都屬於重製顯示設備的色域內，那麼只要再將顏色轉換回重製顯示設備上相對應的 RGB 值，那色彩重製的目標即告完成。但一般而言，因為不同的顯示設備的色域往往會有很大的不同，所以常會發生有顏色想要顯示，卻又不在重製顯示設備的色域裡。

色域對應演算法(Gamut Mapping Algorithm)的目的即是為了處理大小和形狀都不相同的各種顯示設備色域，決定這些色域間點與點的對應關係。



第二章 背景及文獻回顧

在開始說明我提出的方法之前，本章先簡述一些進行色域對應之前的必備知識。為了建立不隨顯示設備而改變的顏色表示方式，本章會將先介紹與顯示設備不相依(Device Independent)的顏色表示法，並介紹測量顏色間相似性的顏色差異方程式(Color Difference Formula)。接著說明如何從少數的樣本顏色來計算出顯示設備的色域裡所有顏色特性，最後則是說明一些其他人使用過的色域對應方式。

2.1 色彩空間 (Color Space)

我們一般常用的顏色的表示方法多不勝數，如RGB，CMYK，和HSV等等[3]。不過因為這些顏色的表示法並沒有定義和最後顏色表現結果的絕對的關係，所有在不同的顯示設備上往往對這些表示法都有不同的表現結果。同樣的RGB值在A顯示設備上與B顯示設備上看到的結果往往會有所差距。

所以若想要在不同的顯示設備上顯示相同的顏色，那首先就必須對顏色的本質和顏色彼此之間的關係有所了解，然後再建立和顏色本質有直接相關的顏色表示法。[3]中大略介紹了關於顏色的基本知識。顏色是一組綜合許多波長的可見光。而人類透過收集視網膜上的光線刺激來感知外界的光線變化。視網膜上有兩種光線受體細胞(Receptor)－桿狀體 (Rod) 和錐狀體 (Cone)。桿狀體廣泛的分布在視網膜的表面上，用來獲得視野中的東西大致的影像外型。桿狀體感知的結果雖然不帶有彩色資訊，但是在低亮度的時候遠較錐狀體反應敏感（所以當亮度很低時，人類只有灰階的視覺）。錐狀體主要落在視網膜中央部份的中央凹(Fovea)上，一般認為錐狀體可再分為三類，分別對不同波長的光線刺激有不同程度的響應程度(Response)，負責人類的顏色感知。

依照上面對人類色彩反應的了解，我們可以用類似人類對光線的反應結果來描述色彩的頻譜分布SPD (Spectral Power Distribution)。換句話說，可以利用三個不

同的權重函數 (Weighting Function) 來代表三種不同錐狀體的不同波長的響應程度。

基於上述理由，CIE(Commission Internationale del'Éclairage)在1931年和1964年定義了兩種權重函數—Standard Colorimetric Observer Color Matching function 和 Supplementary Standard Colorimetric Observer Color Matching Function—來描述人眼對色彩的反應。分別以 $x_{(\lambda)}, y_{(\lambda)}, z_{(\lambda)}$ 和 $x_{10(\lambda)}, y_{10(\lambda)}, z_{10(\lambda)}$ 表示。

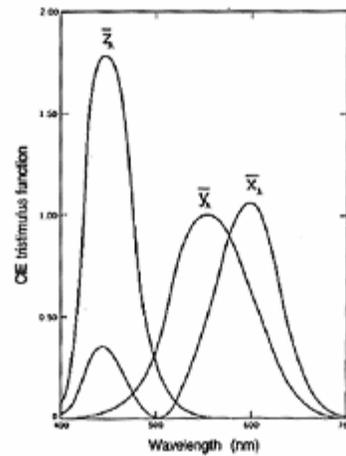


圖 2-1 CIE Standard Colorimetric Observer Color Matching Function

2.1.1 CIE 1931 XYZ Color Space

利用光線的SPD經過的Standard Colorimetric Observer Matching Function轉換後的結果，可以組成最基本的色彩空間—CIE XYZ。CIE XYZ 是一個三維空間，空間中每個點代表一個特定顏色。點的座標的三個分量分別代表該顏色的SPD透過其中一個Standard Colorimetric Observer Matching Function的對應結果。此座標可以利用下面的式子計算：

$$\begin{aligned} X &= k \sum X_{\lambda} P_{\lambda} \Delta\lambda \\ Y &= k \sum Y_{\lambda} P_{\lambda} \Delta\lambda \\ Z &= k \sum Z_{\lambda} P_{\lambda} \Delta\lambda \end{aligned}$$

(Equation 1)

P_λ 代表光線在該波長強度

$X_\lambda, Y_\lambda, Z_\lambda$ 分別代表權重函數的在該波長的權值

k 為一個縮放常數

而通常為了視覺化的方便，會利用下式將三維的CIE XYZ色彩空間對應到2維色度圖（Chromaticity Diagram）上：

$$x = \frac{X}{X + Y + Z}$$
$$y = \frac{Y}{X + Y + Z}$$

(Equation 2)

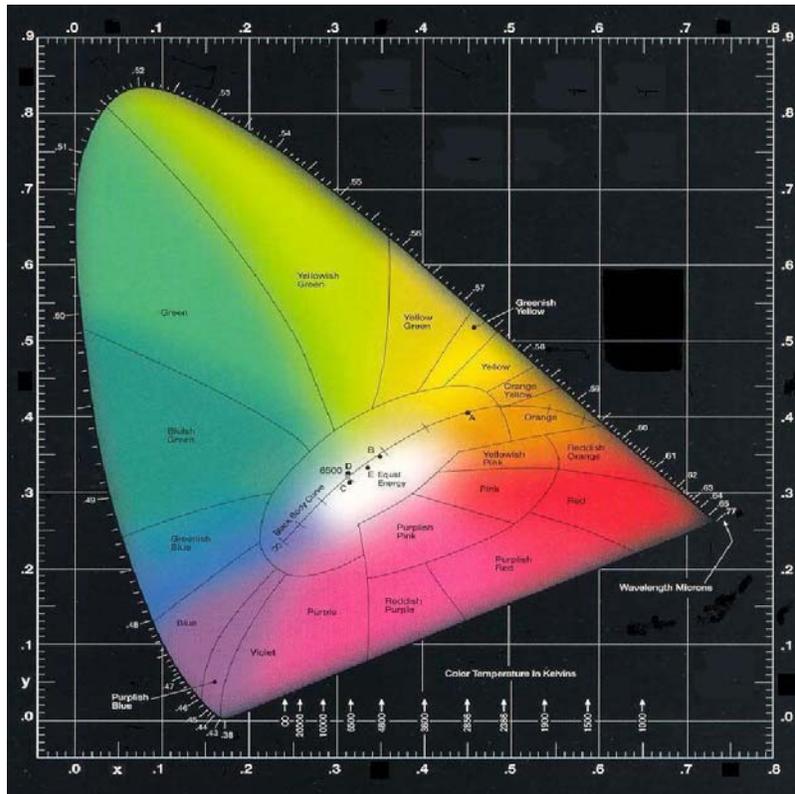


圖 2-2 CIE xy Chromaticity Diagram

不過CIE XYZ色彩空間雖然可以表達色彩對人眼的刺激，但是當要用來計算顏色之間彼此的差異時卻有嚴重的問題。因為除了兩個顏色在三種錐狀體的響應差異外，對顏色差異的解讀還需要仰賴後端的神經系統的處理（而且不幸的是，我們目前對大腦處理這方面資訊的方式的了解仍然十分有限）。所以換句話說，我們無

法直接使用兩個顏色在CIE XYZ 色彩空間中的歐氏距離(Euclidean Distance)來判斷顏色之間的差似程度。以Wright的實驗顏色結果為例，Figure 2.3 中的所有的線的兩個端點代表人們認為有都著相同的色知覺差異。可以看出，隨著顏色所在CIE XYZ 色彩空間裡的位置不同，不同的歐氏距離的卻有一樣色知覺差異。

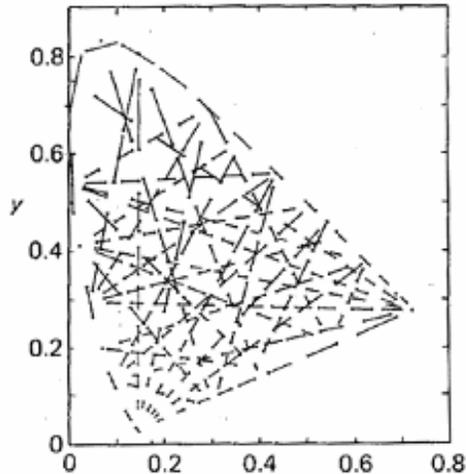


圖 2-3 CIE xy Chromaticity Diagram 裡有相同色彩差異的”色對”

2.1.2 CIE Uniform 色彩空間

2.1.2.1 CIE L^*, u^*, v^* 色彩空間

為了解決前述CIE XYZ色彩空間裡顏色間差異的測量問題，於是CIE從CIE XYZ色彩空間衍伸定義出新的均勻(Uniform)色彩空間來描述顏色。並提出一組新的色度圖—the CIE 1976 uniform chromaticity scale diagram。定義如下：

$$u' = \frac{4X}{X + 15Y + 3Z}$$

$$v' = \frac{9Y}{X + 15Y + 3Z}$$

(Equation 3)

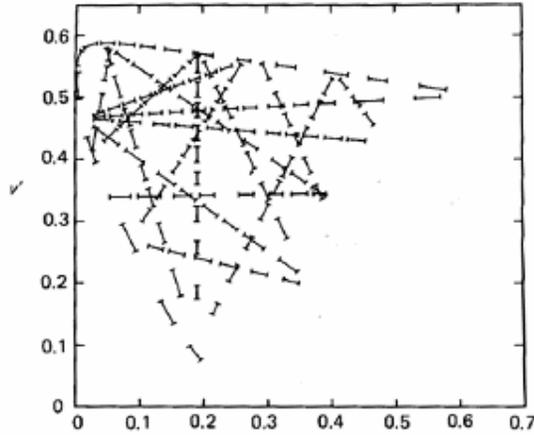


圖 2-4 CIE u'v' Chromaticity Diagram 裡有相同色彩差異的”色對”

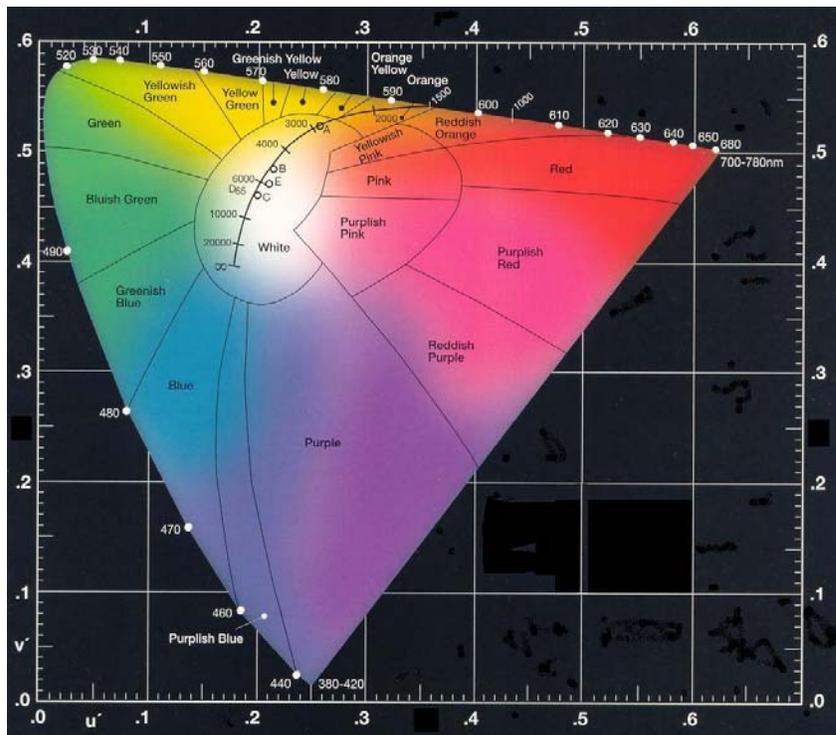


圖 2-5 CIE u'v' Chromaticity Diagram

新的色度圖可以進一步利用來定義新的三維的色彩空間—CIE L^*, u^*, v^* [4]。並且現今被廣泛應用在照明和CRT顯示器等利用加色混合(Additive Mixing)來產生顏色的色彩工業上。由CIE XYZ轉換成CIE L^*, u^*, v^* 的方法定義如下：

$$\begin{aligned}
 L^* &= 116(Y/Y_n)^{1/3} - 16 && \text{if } (Y/Y_n) > 0.008856 \\
 L^* &= 903.3(Y/Y_n) && \text{if } (Y/Y_n) \leq 0.008856 \\
 u^* &= 13.L^*(u'-u'_n) \\
 v^* &= 13L^*(v'-v'_n)
 \end{aligned}$$

(Equation 4)

Index n代表顏色顯示環境中參考白點(Reference White)的座標

另外，我們能由 L^*, u^*, v^* 三個分量定義出一些常用的顏色特性：

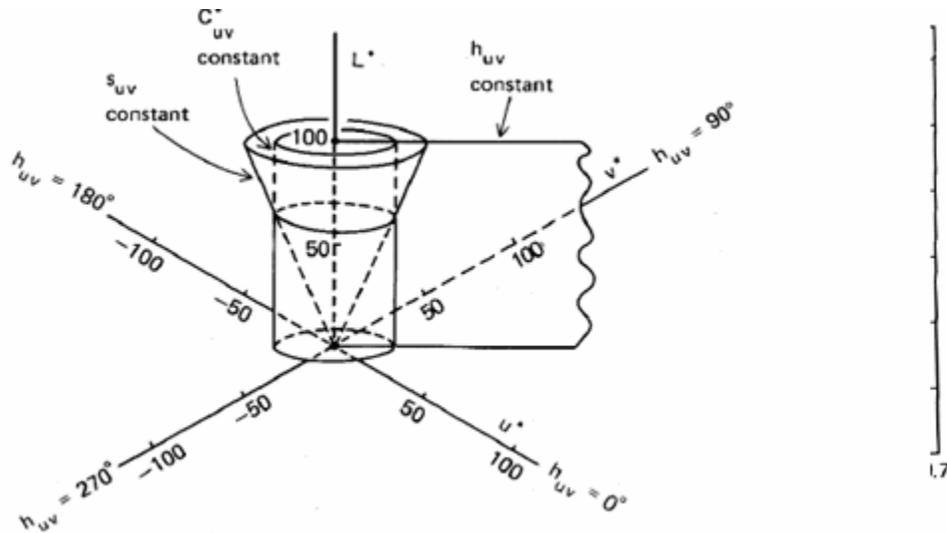


圖 2-6 CIE Color Space

L^* 亮度(Lightness) :

用來預測顏色和參考白點的相對亮度，大小由0到100，分別代表黑點和白點。

CIE LUV 色調(Hue Angle) :

$h_{uv} = \arctan(v^* / u^*)$ 用來預測色彩的色調。範圍由0度到360度。

CIE LUV 色度 (Chroma):

$C_{uv} = (u^{*2} + v^{*2})^{1/2}$ 用來預測顏色的色度。範圍由0開始到無限大。

CIE LUV 飽和度 (Saturation):

$S_{uv} = C_{uv}/L^*$ 在考量亮度以後，用來預測顏色的”強度”。範圍由0開始到無限大。

2.1.2.2 CIE Lab

除了CIE L^*, u^*, v^* ，另一個因為同樣目的被提出來的常用色彩空間是CIE Lab。CIE Lab目前被廣泛應用在印刷，染料，或美術設計等等利用減色混合 (Subtractive Mixing) 來產生顏色的色彩工業上。由CIE XYZ轉換成CIE Lab的方法定義如下：

$$\begin{aligned}
 L^* &= 116(Y/Y_n)^{1/3} - 16 \quad \text{if } (Y/Y_n) > 0.008856 \\
 L^* &= 903.3(Y/Y_n) \quad \text{if } (Y/Y_n) \leq 0.008856 \\
 a^* &= 500[f(X/X_n) - f(Y/Y_n)] \\
 b^* &= 200[f(Y/Y_n) - f(Z/Z_n)]
 \end{aligned}$$

where if $f(N/N_n)^{1/3} > 0.0086 (N \in \{X, Y, Z\})$ then

$$f(N/N_n) = (N/N_n)^{1/3}$$

otherwise

$$f(N/N_n) = 7.787(N/N_n) + 16/116$$

(Equation 5)

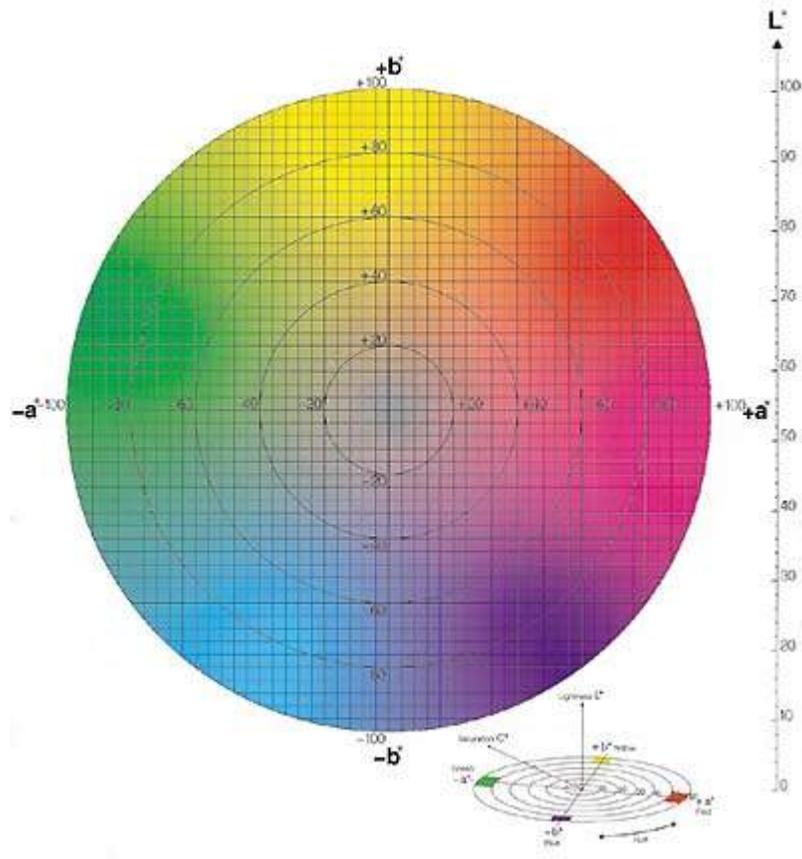


圖 2-7 CIE Lab Color Space

將2.1.2.1裡CIE L^*, u^*, v^* 色彩空間定義的色調和色度預測方程式中的 u^* 以 a^* 代換， v^* 以 b^* 代換，可以定義出屬於CIE Lab 的色調和色度預測值— h_{ab} 和 C_{ab} 。不過CIE Lab並未如CIE L^*, u^*, v^* 般定義飽和度(Saturation)，也沒定義色度圖(Chromaticity Diagram)。

必需一提的是，雖然相較於CIE XYZ，CIE L^*, u^*, v^* 和CIE Lab在計算顏色間差異上有很大的進步，但是都並非”完美”。而所定義的色調，色度的預測值也僅只是對我們理想中色調和色度所做的”預測”，並非百分之百的正確。此外，CIE L^*, u^*, v^* 和CIE Lab 兩者之中究竟該選擇何者來使用也沒有絕對的優劣之分，通常只是就習慣來選擇[2]。

2.2 顏色的差異(Color Difference)

2.2.1 色塊間的顏色差異

繼續來討論描述顏色之間差異的方法。上節2.1.2所提的均勻色彩空間對顏色差異的測量做了初步的改善。所以在CIE L^*, u^*, v^* 及CIE Lab裡測量顏色差異性最直接的方法即是使用歐氏距離(Euclidean Distance)來計算兩個顏色之間的差異，例如CIE LUV Color Difference和CIE Lab Color Difference：

$$\Delta E_{uv}^* = \left[(\Delta L^*)^2 + (\Delta u^*)^2 + (\Delta v^*)^2 \right]^{1/2} \quad (\text{Equation 6})$$

$$\Delta E_{ab}^* = \left[(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2 \right]^{1/2} \quad (\text{Equation 7})$$

除了歐氏距離外，由於CIE LUV和CIE Lab並非完美，CIE在1994年提出另外一個新的顏色差異測量方式—CIE94 Color Difference Formula來彌補在CIE Lab色彩空間裡直接使用歐氏距離時的一些缺點。定義如下：

$$\Delta E_{94} = \sqrt{\Delta L_{ab}^{*2} + \left(\frac{\Delta C_{ab}}{1 + 0.045C_1^*} \right)^2 + \left(\frac{\Delta H_{ab}}{1 + 0.015C_1^*} \right)^2} \quad (\text{Equation 8})$$

C_1^* 表示欲計算差異的表準顏色的色度

在上述的方式之外之外，Katoh[5]在他發展的色域對應法裡直接將顏色間亮度，色度，和色調的差異給予不同大小的權重值，來計算尋找最相似的顏色。使用的方程式如下：

$$\Delta E_w = \sqrt{\left(\frac{\Delta L^*}{K_L} \right)^2 + \left(\frac{\Delta C^*}{K_C} \right)^2 + \left(\frac{\Delta H^*}{K_H} \right)^2} \quad (\text{Equation 9})$$

K_h, K_L, K_c 由使用者自行決定。由於亮度和色調一般認為對影響顏色是否相似影響較大，所以在使用時一般選擇設定 $K_c \geq K_L \geq K_h$ 。讓亮度和色調的差異佔較大的比重。Katoh的建議是使用 (K_h, K_L, K_c) 使用 (1,1,2) 或著是 (1,2,2)。

2.2.2 圖片間的顏色差異

前面所介紹的計算顏色差異方法提供了分析兩個顏色間差異性的工具。當將這些計算工具用在分析複雜圖片在兩個不同顯示設備上顯示出來的顏色差異時，一般是將像素(Pixel)一對一的計算在兩個顯示設備上的顏色差異，然後在觀察這些差異值的統計資料，如最大值，最小值，和平均值等等。

不過事實上，人們很早就發現影響我們對顏色感知的除了顏色本身的特性之外，顏色周遭的環境(Surround)也會有影響。以日常生活中常見的例子來說，超市裡販賣的肉類旁常會擺上一些綠色的裝飾物作為襯托。這是因為在綠色的對比下，可以使得紅色的肉類感覺起來更為鮮豔。Zhang 和 Wandell[6]注意到這個事實，於是提出了 S-CIE Lab(Spatial Extension of CIE Lab)，將影像在一個像素對一個像素(Pixel by Pixel)計算差異前先進行前處理(Preprocess)。先將影像中的顏色從 CIE XYZ 作線性轉換到新的對比色彩空間 ACC。ACC 色彩空間裡的三個分量(Components)分別代表無顏色資訊(緊隨亮度而變)的 Achromatic，紅色綠色對比程度的 Red-Green，以及藍色黃色對比程度的 Blue-Yellow。接著再利用 Low Pass Filter 或 Band Pass Filter 分別處理影像在 ACC 的資訊，最後再轉至 CIE Lab 色域空間。用傳統的方式計算兩張圖片的 Pixel 間顏色差異。

2.3 顯示設備的特性

在2.1節定義了一些色彩空間之後，我們已經可以成功的描述出每個顏色的特性。不過以目前一般的全彩顯示器為例，每個顯示器可以有一千六百多萬個顏色，所以想要對顯示器的每個顏色一一做測量幾乎是件不可能任務。因此，如何測量少數個固定的樣本顏色(Sample Color)再計算出顯示器所有可用顏色在色彩空間裡的

位置是想要進行色域對應時不可或缺的一部份。下面簡介眾多方法中最常見的兩種-Cubic Interpolation 和Polynomial Fitting。

2.3.1 Cubic Interpolation

從固定的樣本顏色的特性來計算其他所有顏色的特性最直觀的方法，就是利用內插法內插出樣本點之外的顏色資訊。想要內插三維空間裡的資訊最簡單的方法就是利用Cubic Interpolation。想利用Cubic Interpolation內插顯示設備每個RGB值在CIE XYZ色彩空間裡的位置時，首先要將 $256 \times 256 \times 256$ 的RGB空間切成等大小的Cubic，再把每個Cubic的頂點當作測量的樣本顏色去測量其在CIE XYZ裡的位置，位於樣本顏色之間的點在CIE XYZ裡的位置則利用三次線性內插出出來。圖2-8是Morovic在[2]裡的例圖。

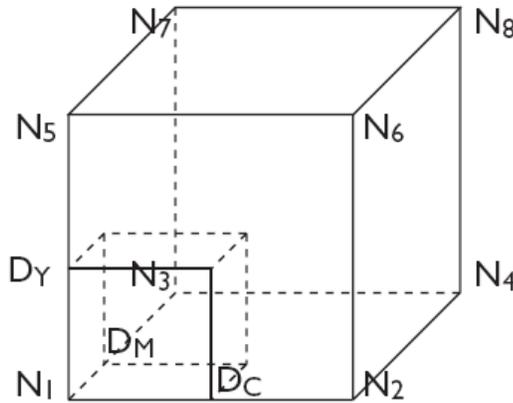


圖 2-8 利用 Cubic Interpolation 內插顏色的特性

內插公式可以整理如下：

$$\begin{aligned}
 N = & (1 - D_C)(1 - D_M)(1 - D_Y)N_1 + D_C(1 - D_M)(1 - D_Y)N_2 + \\
 & (1 - D_C)D_M(1 - D_Y)N_3 + D_CD_M(1 - D_Y)N_4 + \\
 & (1 - D_C)(1 - D_M)D_YN_5 + D_C(1 - D_M)D_YN_6 + \\
 & (1 - D_C)D_MD_YN_7 + D_CD_MD_YN_8 \\
 N \in & \{X, Y, Z\}
 \end{aligned}$$

(Equation 10)

相較於其他用來計算顯示設備所有顏色特性的方法，使用內插法通常可以得到更正確的結果。不過較於其他的計算法也需要更多的樣本顏色，進而需要更多的測量時間。除此之外，在使用上內插法還需要一個足夠大的LUT(Look up table)來紀錄樣本顏色的特性。

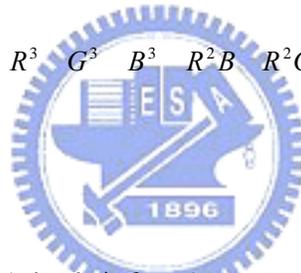
2.3.2 Polynomial Fitting

除了內插法，另一個也常用的方式就是利用一組多項式來計算一個顯示設備上RGB 和 CIE XYZ 間的對應關係。Jackson[7]利用一個三元三次多項式來代表RGB和CIE之間的對應函數。

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = BC$$

$$C = [R \ G \ B \ R^2 \ G^2 \ B^2 \ R^3 \ G^3 \ B^3 \ R^2B \ R^2G \ G^2R \ G^2B \ B^2R \ B^2G \ GB \ RB \ RG]$$

B 為一個 3×18 的矩陣



(Equation 11)

假設RGB到CIE XYZ的對應關係是一個三元三次多項式。在測量數個樣本顏色以後，可以利用Least Square method [8]找出使的Least Square Error最小的三元三次多項式。Polynomial Fitting方法的正確度取決於使用了多少的樣本顏色來來估計參數，以及所選的樣本顏色的代表性。

2.3.3 色域的邊界描述法(Gamut Boundary Descriptor)GBD

利用前述的方法後，我們可以得到顯示設備上所有能顯示的顏色的特性。不過我們還需要適當的資料結構和演算法才能在色域對應時幫我們處理這些顏色。大多數的色域對應演算法最需要用的到的資訊是顯示設備的顯示極限。換句話說，我們需要能有效率的計算出顯示設備的色域的邊界，並且支援一些所需的幾何關係運算。因此，配合所需的幾何運算不同，不同的色域對應演算法有各自適合的

GBD。[9]以及[10]提出了一些不同的GBD。而我則稍微修改[11][12]的方法來時做我的色域對應演算法。方法容後述之。

2.4 色域對應

2.4.1 目標

最後進入我希望解決的色域對應的部分。簡單的來說，色域對應的目標是在考慮顯示設備的色域之後，對兩個顯示設備提供一組適當的顏色對應關係，使的圖片在重製顯示設備上能顯示和在原始顯示上相似的結果。理想上，如果所有我們希望表現的顏色都在重製設備的色域裡，那麼可以完美解決這問題。但因為不同的顯示設備擁有的色域都不盡相同，所以常會有許多顏色在原始顯示設備的色域範圍內，卻在重製顯示設備色域之外。另外，因為最後希望能在重製顯示設備上顯示的是複雜的圖片，而不僅只是簡單的一個顏色，所以在進行色域對應時也不能只考慮單一顏色的對應結果。下面是 [1]中所提及MacDonald所提出的幾個基本的色域對應目標。



- 1.保持影像的灰階變化，並且希望能得到最大的對比值。
- 2.減少重製顯示設備色域範圍之外被Clipping掉顏色
- 3.盡量減少顏色的Hue角變化
- 4.在允許的範圍內，傾向於有較高的飽和度
- 5.顏色之間的相互關係比顏色的真實值更重要

2.4.2 傳統常用的色域對應法

在設計色域對應演算法時，需要考慮的主要有兩大點：

1. 對應方向(Mapping Direction)- 決定顏色該往什麼方向移動，使色域外的顏色能落在色域內

2. 對應方法(Mapping Method)–移動的方向決定後，決定顏色往該方向移動的量

以如何決定對應的方向來說，因為認為色調(Hue)代表顏色的種類，相同的色調屬於相同的顏色，所以多數的色域對應演算法都將顏色的色調值當作不應變動的常數，而只對亮度(Lightness)和色度(Chroma)做變動 [18]。舉例來說 [13][14][15][16]，Sara認為色域對應時顏色的色調和亮度都不應改變，所以將顏色往色度較小的地方壓縮到色域的範圍之內(Figure 2.9 a)，Taylor決定當顏色落在色域外時，選擇色域內最相似的點來代替，Leihanen選擇將顏色往整個無色軸的中心(色度為零，亮度五十)點移動，MacDonald和Morovic則將顏色往擁有最大色度時的無色軸作移動。Figure 2.9為圖例。

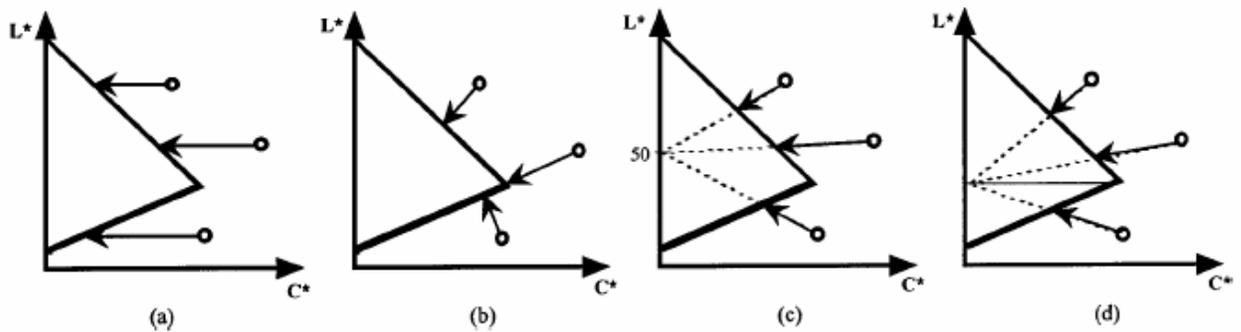


圖 2-9 Gamut Mapping Direction(a) Sara (b) Taylor (c) Laihanen (d) MacDonald

對應方向決定後，接著考慮對應的方法—顏色該往對應的方向移動多少的量。對應方法可以分為三大類型-(1)Clipping，(2)Linear Compression 和 (3)Nonlinear Compression。Clipping最大的特性在於Clipping不變動在重製設備色域之內的顏色。只有當想要顯示的顏色在色域之外時，選定某個色域邊界(Gamut Boundary)上的顏色作為對應。而Linear 和 Nonlinear Compression方法則須考慮整個重製設備的色域。為了想保持顏色之間的相對關係，所以無論欲顯示的顏色是否在色域內，對應到的顏色都可能會有變化。

在決定顏色要對應的方向決定後，一個簡單的對應方法範例如下圖Figure 2.10

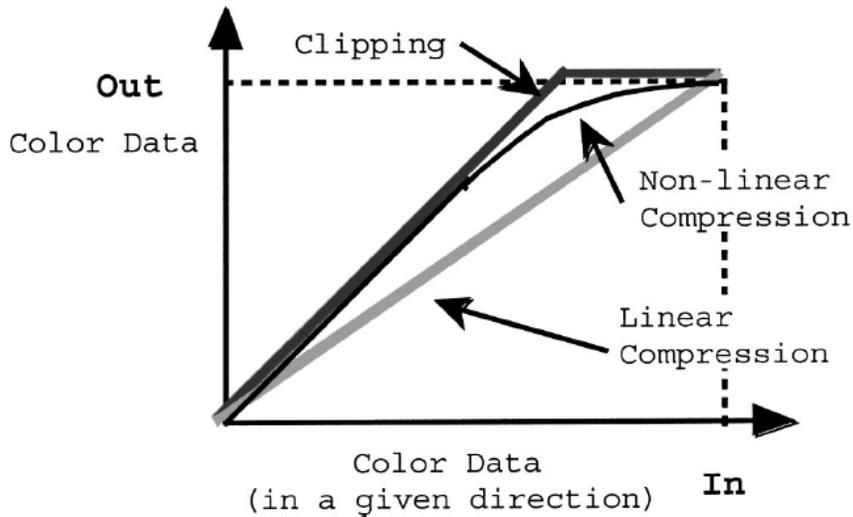


圖 2-10 Mapping Method

Clipping和Compression出來的結果最大差異在於—如果使用Clipping法，因為在色域內的顏色對應後的結果不會改變，所以在色域內的顏色表現出的效果較佳，整張圖像的飽和度也較高。不過當顏色在色域時，有可能會被對應到同一個點上，所以會使影像中的Gradient消失，失去一些影像裡的細節。而Compression法則反之。

本節2.4.1開頭所述色域對應的第二個目標即是在希望在Compression以及Clipping間達成一個平衡。一般認為當兩個顯示設備的色域差異很大的時候，使用Compression法的結果較佳。而當顯示設備間色域差異小的時候，則會傾向於使用Clipping方法。

第三章 使用 Linear Programming 進行色域壓縮

在 2.4.2 裡，我提到了影響色域對應結果的兩大部分一對應方向以及對應方法。不過這些方法並未考慮原始顏色之間的相對關係，所以最後得到的結果可能會違反最初所定的色域對應目標第五點—顏色之間的相互關係比顏色本身確實的值更重要。我在之前在 2.2.2 裡作過討論，人類對於顏色感知的結果除了顏色的本身之外，還受到週遭其他顏色的影響。所以我將考慮顏色之間的關係，使的色域對應前後顏色彼此間的關係能夠保持一致。

3.1 利用 Linear Programming 進行色域對應

3.1.1 從最佳化的觀點看傳統的色域對應

色域對應最基本的目標，就是希望色域對應前和色域對應後顏色改變的量能越少越好。假設在進行色域對應時每個顏色移動的方向是固定的，則對每個固定顏色，色域對應前後的顏色差異可以表示成下式：

$$\begin{aligned} & \sqrt{\frac{\nabla L^2}{W_L} + \frac{\nabla h^2}{W_h} + \frac{\nabla C^2}{W_c}} \\ &= \sqrt{\frac{(L_o - (L_o + \alpha V_L))^2}{W_L} + \frac{(h_o - (h_o + \alpha V_h))^2}{W_h} + \frac{(C_o - (C_o + \alpha V_C))^2}{W_C}} \\ &= \sqrt{\frac{\alpha^2 (V_L)^2}{W_L} + \frac{\alpha^2 (V_h)^2}{W_h} + \frac{\alpha^2 (V_C)^2}{W_C}} \\ & \text{assume } \alpha \geq 0, \text{ then} \\ &= \alpha \sqrt{\frac{(V_L)^2}{W_L} + \frac{(V_h)^2}{W_h} + \frac{(V_C)^2}{W_C}} \end{aligned}$$

(Equation 12)

L ， h ， C 分別代表亮度(Lightness)，色調(Hue)，以及色度(Chroma)

W_L ， W_h ， W_c 分別代表差異方程式對亮度，色調，及色度的不同權值(Weight)

V_L, V_h, V_c 則分別代表移動向量在 L, h, C 的分量

Index o 表示對應前的原始顏色

α 則代表顏色被移動的量

明顯的，當顏色完全不改變時（即 $\alpha=0$ ），色域對應前後的顏色差距最小，不過為了在重製顯示設備上顯示顏色， α 至少需要滿足下面條件：

$$I_i + \alpha V_i \in \text{Gamut of Reproduction media}$$

I_i 代表像素 i 的顏色原本在色彩空間中的位置

V_i 代表像素 i 的顏色在色域對應時移動的方向向量

(Equation 13)

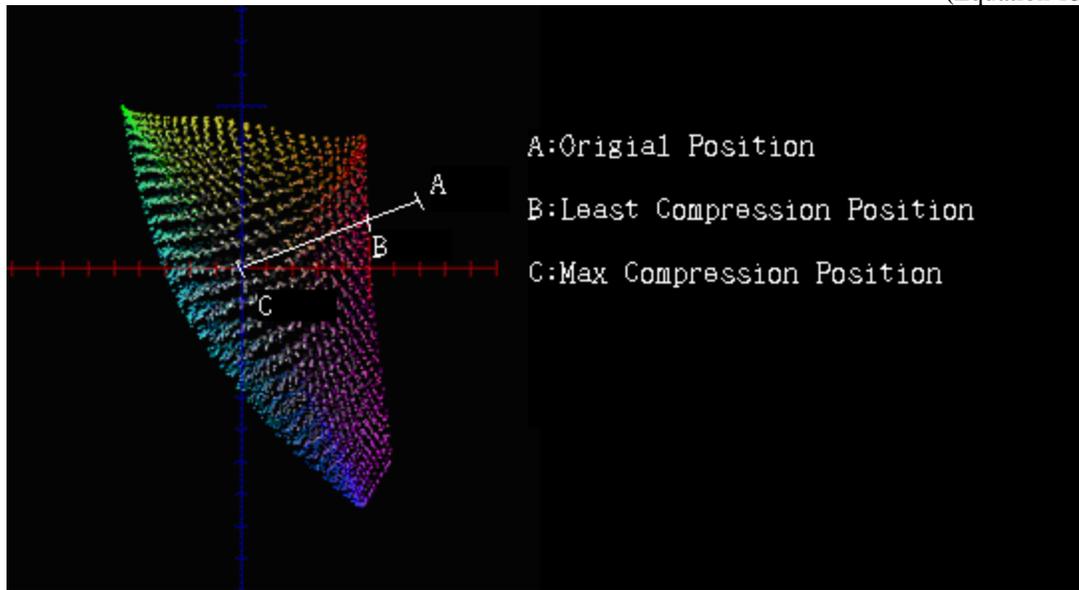


圖 3-1 Least and Max Compression Amount

以 Figure 3.1 為例，在考慮重製顯示設備的色域後，可以得到下式：

$$C_{LeastCompressAmount} \leq \alpha_i \leq C_{MaxCompressAmount}$$

(Equation 14)

不難看出，如果利用上述的 Objective Function(Equation 3.1)及線性不等式(Equation 3.3)進行 Linear Programming[8][17]來計算每個顏色的壓縮量，不難看出

會得到傳統使用 Clipping 方式的結果。在顏色重製設備色域內的顏色不會移動，而在色域外的顏色則對應到往顏色對應方向最接近的色域表面。

3.1.2 顏色與顏色之間的關係限制

了解色域對應最基本的要求和目標後，接著進一步考慮顏色與顏色之間關係。觀察圖片中像素間顏色間的 Gradient，我們的目標是讓色域對應後像素間的 Gradient 能夠盡量和對應前保持一致的趨勢。

兩個像素之間的 Gradient 可以表示如下：

$$\begin{aligned} Gr_{ij}^{before} &= I_i - I_j \\ Gr_{ij}^{after} &= (I_i + \alpha_i V_i) - (I_j + \alpha_j V_j) \end{aligned}$$

(Equation 15)

Gr_{ij}^{before} 以及 Gr_{ij}^{after} 為色域對應前和色域對應後像素 i 與像素 j 間的 Gradient 向量

I_i 與 I_j 為像素 i 和 j 在 CIE Lab 或 CIE LUV 中的位置

V_i 與 V_j 為像素 i 和 j 在進行色域對應時的移動方向向量

α_i 與 α_j 為像素 i 和 j 在進行色域對應演算法時移動的量

為了希望保持色域對應前後兩個 Gradient 的趨勢一致，我嘗試使用兩種不同的方式來限定對應前後的 Gradient 關係。第一種是考慮色域對應後的 Gr^{after} 在色域對應前的 Gr^{before} 的投影量。首先要求 Gr^{after} 和 Gr^{before} 的夾角於 $-\frac{\pi}{2}$ 和 $\frac{\pi}{2}$ 之間，使的

Gr^{after} 在 Gr^{before} 上的投影向量的方向不會與 Gr^{before} 相反，則：

$$\begin{aligned} \because Gr^{after} \bullet Gr^{before} &= \|Gr^{after}\| \times \|Gr^{before}\| \times \cos \theta \\ \therefore -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2} &\Leftrightarrow Gr^{after} \bullet Gr^{before} \geq 0 \end{aligned}$$

(Equation 16)

又 Gr^{after} 在 Gr^{before} 的投影向量可以表示成下式

$$\begin{aligned}
& \text{Pr oj}(Gr^{after}) \\
&= \|Gr^{after}\| \times \frac{Gr^{after} \bullet Gr^{before}}{\|Gr^{after}\| \times \|Gr^{before}\|} \times \frac{1}{\|Gr^{before}\|} \times Gr^{before} \\
&= \frac{Gr^{after} \bullet Gr^{before}}{\|Gr^{before}\|^2} \times Gr^{before}
\end{aligned}$$

(Equation 17)

所以若要求 Gr^{after} 在 Gr^{before} 的投影量不應大於原始的 Gr^{before} ，可以得到下面結果：

$$\begin{aligned}
& \|\text{Pr oj}(Gr^{after})\| \leq \|Gr^{before}\| \\
& \Rightarrow \left\| \frac{Gr^{after} \bullet Gr^{before}}{\|Gr^{before}\|^2} \times Gr^{before} \right\| \leq \|Gr^{before}\| \\
& \Rightarrow \frac{Gr^{after} \bullet Gr^{before}}{\|Gr^{before}\|^2} \times \|Gr^{before}\| \leq \|Gr^{before}\| \\
& \Rightarrow \frac{Gr^{after} \bullet Gr^{before}}{\|Gr^{before}\|} \leq \|Gr^{before}\| \\
& \Rightarrow Gr^{after} \bullet Gr^{before} \leq \|Gr^{before}\|^2
\end{aligned}$$

(Equation 18)

總結上面兩個限制，得到：

$$0 \leq Gr^{after} \bullet Gr^{before} \leq Gr^{before} \bullet Gr^{before}$$

(Equation 19)

最後，將 Gr^{before} 以及 Gr^{after} 以 I 和 V 代換，得對移動量 α 的線性不等式：

$$\begin{aligned}
0 &\leq (I_i + \alpha_i V_i - I_j - \alpha_j V_j) \cdot (I_i - I_j) \leq (I_i - I_j) \cdot (I_i - I_j) \\
\Rightarrow 0 &\leq (I_i - I_j) \cdot (I_i - I_j) + (I_i - I_j) (\alpha_i V_i - \alpha_j V_j) \leq (I_i - I_j) \cdot (I_i - I_j) \\
\Rightarrow -(I_i - I_j) \cdot (I_i - I_j) &\leq (I_i - I_j) (\alpha_i V_i - \alpha_j V_j) \leq 0 \\
\Rightarrow -(I_i - I_j) \cdot (I_i - I_j) &\leq (I_i - I_j) \cdot \alpha_i V_i - (I_i - I_j) \cdot \alpha_j V_j \leq 0
\end{aligned}$$

(Equation 20)

除了較簡單的限制 Gr^{after} 在 Gr^{before} 上的投影量，另外一個嚴格的限制方式是要求 Gradient 的每一個分量在色域對應前後保持一致。將像素的顏色從 CIE Lab 或 CIE LUV 色彩空間轉換成以亮度(Lightness)，色調(Hue)，以及色度(Chroma)的表示法後。作下面的限制：

$$\begin{aligned}
&\text{if } I_{Li} - I_{Lj} \geq 0 \text{ then } 0 \leq (I_{Li} + \alpha_i V_{Li}) - (I_{Lj} + \alpha_j V_{Lj}) \leq I_{Li} - I_{Lj} \\
&\text{else } I_{Li} - I_{Lj} < (I_{Li} + \alpha_i V_{Li}) - (I_{Lj} + \alpha_j V_{Lj}) \leq 0 \\
&\text{if } I_{hi} - I_{hj} \geq 0 \text{ then } 0 \leq (I_{hi} + \alpha_i V_{hi}) - (I_{hj} + \alpha_j V_{hj}) \leq I_{hi} - I_{hj} \\
&\text{else } I_{hi} - I_{hj} < (I_{hi} + \alpha_i V_{hi}) - (I_{hj} + \alpha_j V_{hj}) \leq 0 \\
&\text{if } I_{ci} - I_{cj} \geq 0 \text{ then } 0 \leq (I_{ci} + \alpha_i V_{ci}) - (I_{cj} + \alpha_j V_{cj}) \leq I_{ci} - I_{cj} \\
&\text{else } I_{ci} - I_{cj} < (I_{ci} + \alpha_i V_{ci}) - (I_{cj} + \alpha_j V_{cj}) \leq 0
\end{aligned}$$

L, h, C 分別代表亮度，色調，和色度三個分量。

(Equation 21)

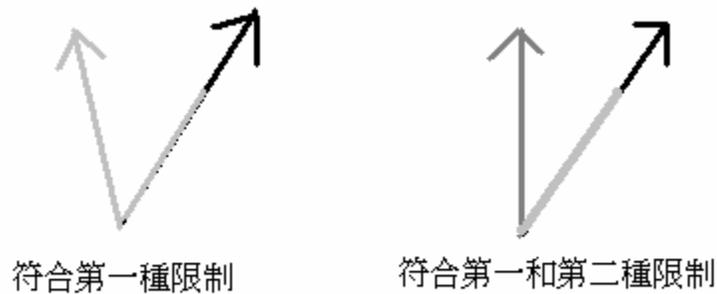


圖 3-2 Linear Constraint

有了這些線性不等式之後，即能對希望最佳化的 Objective Function 進行 Linear Programming。在 3.1.1 時，因為顏色的壓縮量之間不會互相影響，所以每個顏色的最佳化是獨立的。不過現在因為考慮到顏色與顏色之間的關係，所以 Objective Function 的設計會影響最後的壓縮結果。假設所有的顏色在色域對應時都一樣重要，那麼要最佳化的 Objective Function 即為所有顏色對應後差異的合。不過我們也可以讓每個顏色依照出現的頻率作最佳化，越常見的顏色希望他越能保持不變。所以以最佳化一張圖片的色域對應結果來說，可以使用下面的 Objective Function

$$\sum_{\forall i \in \text{image}} \left(\alpha_i \sqrt{\frac{(V_{Li})^2}{W_L} + \frac{(V_{hi})^2}{W_h} + \frac{(V_{Ci})^2}{W_C}} \right)$$

(Equation 22)



3.2 實做方法

3.2.1 (Control Point)

理論上，我們可以直接使用 3.1 提出的方法進行色域對應，計算出每個顏色的移動量。不過以現今大多數的顯示設備而言，要對色域內每個顏色都進行這些步驟幾乎是件不可能任務。以一台全彩 24bit 的顯示器來說，就有一千六百多萬個變數要進行 Linear Programming。所以實做上需要使用少量的控制點(Control Point)來代替其他色域內的顏色進行色域對應。我將利用 Morovic 和 Lue[11]的 Segment Maxima GBD(Gamut Boundary Descriptor)作為基礎設計我的控制點，再對控制點進行 Linear Programming。

Segment Maxima GBD 使用一個固定大小的矩陣來表示色域的邊界。矩陣裡記錄色彩空間中每個區間(Segment)裡最極端的顏色。區間的切割方式依使用的色域對應方法來決定，目標是使的進行色域對應所需要的一些基本幾何運算能最有效率的達成。以 2.4.2 裡 Leihanen 所提的想法—將顏色往 CIE Lab 的無色軸中心點（色度零，亮度五十）做移動為例，一個建議的方法是利用無色軸中心作為參考點，把

色彩空間依照對參考點的仰角 α 以及對 X 軸的夾角 θ 做等分的切割出許多區間後，再用矩陣記錄下每個區間裡和參考點的距離 r 最遠的顏色。 α ， θ ， r 的計算方式如下：

$$r = \left[(L - L_{reference})^2 + (a - a_{reference})^2 + (b - b_{reference})^2 \right]^{\frac{1}{2}}$$

$$\alpha = \tan^{-1} \left(\frac{(b - b_{reference})}{(a - a_{reference})} \right)$$

$$\theta = \tan^{-1} \left[\frac{(L - L_{reference})}{\left((a - a_{reference})^2 + (b - b_{reference})^2 \right)^{\frac{1}{2}}} \right]$$

(Equation 23)

Figure 3.4 為 α ， θ ， r 的圖例，Figure 3.5e 右邊是我測量出的 EIZO FlexScanL568 液晶監視器的色域，左邊則是使用上述 Segment Maxima GBD 表示 EIZO FlexScanL568 色域的結果。

在利用 Segment Maxima GBD 找出代表色域裡每個區間的顏色之後，我再在這些顏色和參考點的相連的直線上依照長度做等分的切割，在色域內也細分出新的區間，將這些區間的所有頂點作為我進行色域對應時的控制點。

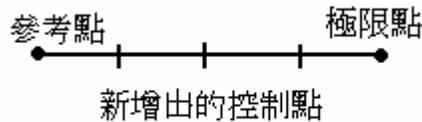


圖 3-3 Control Point

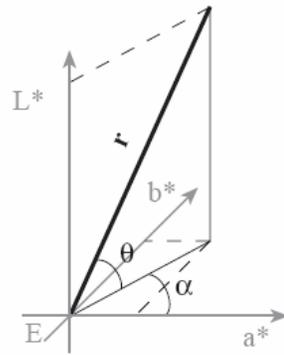


圖 3-4 The Definition of alpha, theta, and radius

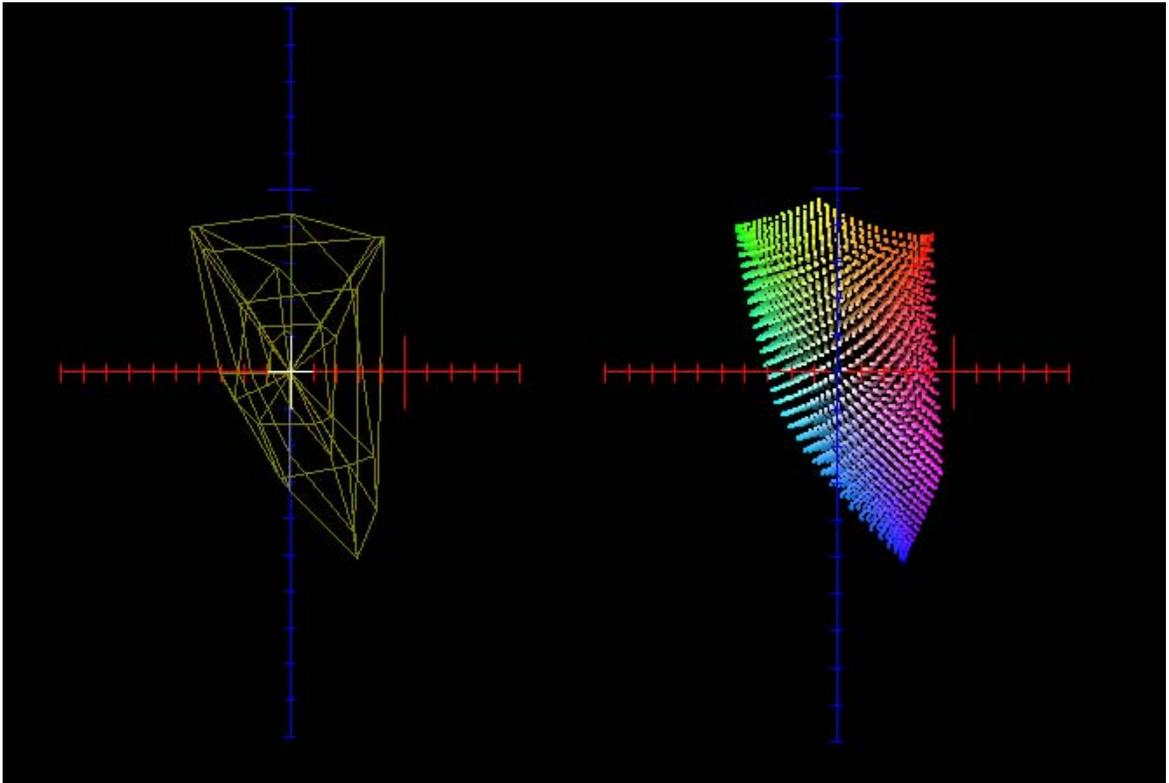


圖 3-5 Result of Segment Maxima GBD

3.2.2 控制點的 Objective Function

定義完控制點和區間後，我將利用每個顏色所在區間的八個控制點以 α ， θ ， r 做三維的線性內插來計算每個顏色的移動量。所以每個顏色的移動量可以表示成周圍八個控制點的移動量乘以特定的權重值後相加。對每個顏色 i 而言的色域對應後的移動量 α_i 可以表示成

$$\alpha_i = \sum_{j \in \text{Control Point}} WI_j^i \alpha_j$$

α_j 是 Control Point j 的移動量

WI_j^i 是 Control Point j 對顏色 i 所佔的 Interpolation 比例

(Equation 24)

則 3.1.2 裡所使用的 Objective Function 可以改成：

$$\sum_{\forall i \in \text{image}} \left(\left(\sum_{j \in \text{Control Point}} WI_j^i \alpha_j \right) \sqrt{\frac{(V_{Li})^2}{W_L} + \frac{(V_{hi})^2}{W_h} + \frac{(V_{Ci})^2}{W_C}} \right)$$

(Equation 25)

3.2.3 總結

最後總結第二與第三章裡所述的架構，說明整個我進行的色域對應流程。不失其一般性，假設原始以及重製顯示設備都接收 RGB 訊號，但是在每個 RGB 值的表現結果不盡相同。則若希望圖片能在重製顯示設備上顯示和原始顯示設備相似的顏色結果，那色域對應需要提供一組由原始顯示設備的 RGB 值到重製顯示設備 RGB 值的對應關係。Figure 3.6 表示我進行色域對應的流程圖。

首先先測量出原始和重製顯示設備中樣本顏色的 RGB 值所對應的 CIE XYZ，再利用 Cubic Interpolation 計算出色域內全部的顏色，並轉換到 CIE Lab 或著 CIE LUV 色彩空間。接著利用 3.2 裡介紹的方式完成步驟 3 到 9，得到每個顏色的新位置，最後再反查重製顯示設備上 RGB 和 CIE XYZ 的對應關係，求出原始顯示設備

裡每個顏色在重製顯示設備上該使用的 RGB 值。

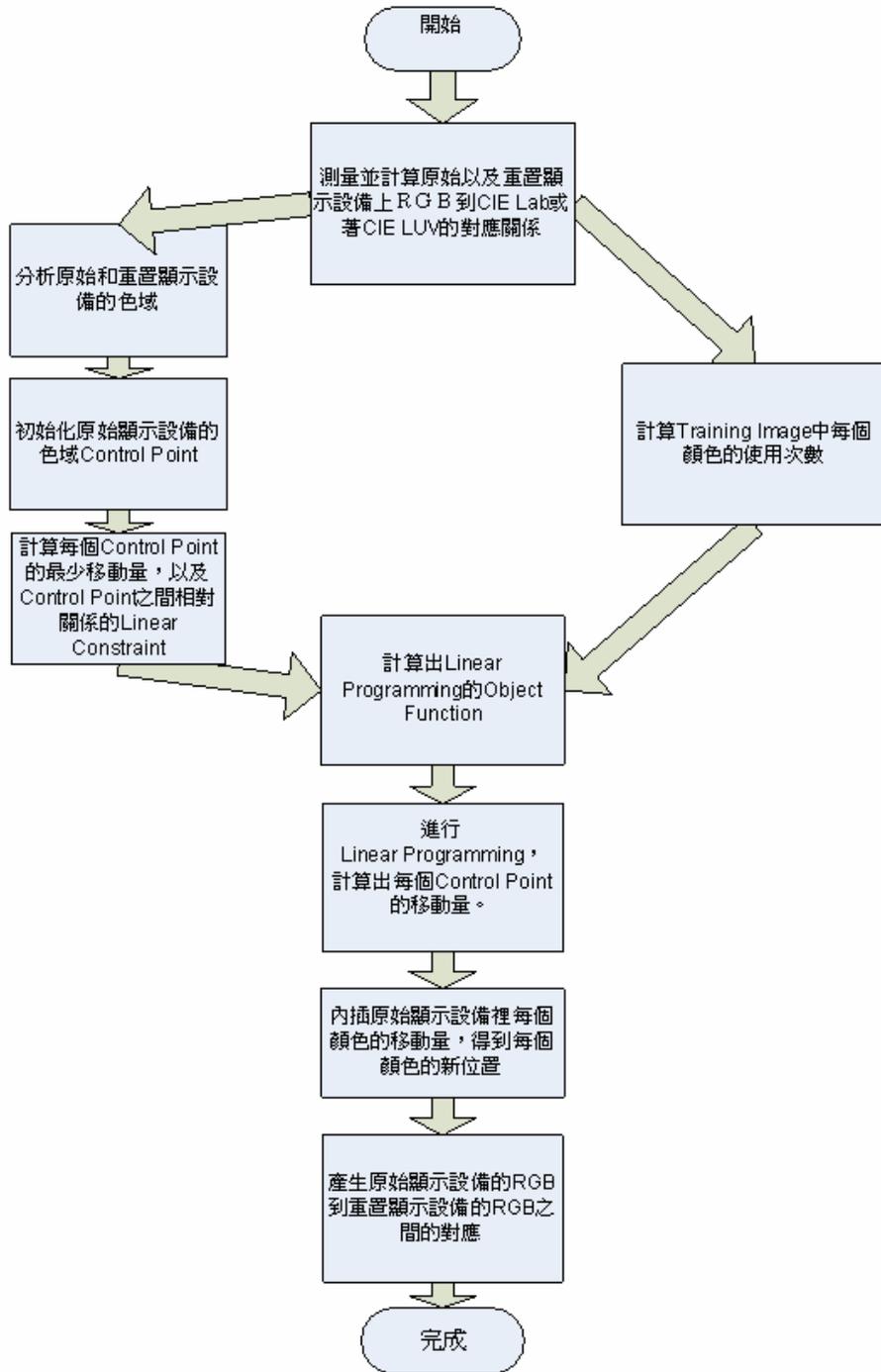


圖 3-6 The Flowchart of my gamut mapping algorithm

第四章 實驗與討論

4.1 實驗

最後，我實驗在色域較小的 EIZO FlexScane L568 LCD 電腦監視器上模擬擁有較大色域的 Panasonic TC-20LA1TN 液晶電視。EIZO FlexScane L568 LCD 和 Panasonic TC-20LA1TN 的訊號源由 Asus 製造的 ATI RADEON 9600 VIVO 顯示卡負責輸出。按照第三章末節的流程，首先我將顏色在 RGB 空間上均勻的取 $18 \times 18 \times 18$ 的樣本點，再使用 GretagMacbeth 的 i1 Display Color Meter 和它所提供的 SDK 來測量樣本點在 CIE XYZ 空間所對應到的值。而樣本點以外的顏色資訊則使用 2.3.1 的 Cubic Interpolation 得到。

得到顏色對應的 CIE XYZ 資訊後，接著再將所有顏色轉換到 CIE Lab 色彩空間上。一般在色彩空間轉換時，會傾向於使用兩台顯示器各自的最亮點((R,G,B)=(255,255,255))作為參考白點(Reference White)。不過因為我想將兩台顯示器直接放在一起比較色域對應結果，所以在轉換至 CIE Lab 時我統一使用 EIZO FlexScane L568 上的最亮點作為參考白點。

Figure 4.1 及 Figure 4.2 為這兩個顯示設備在 CIE Lab 的色域比較圖。右邊為 EIZO FlexScane L568。左邊則為 Panasonic TC-20LA1TN。Figure 4.1 是色域在 ab 平面上的正交投影(Orthogonal Projection)，Figure 4.2 則是在 La 平面上的正交投影。須注意的是，在 Figure 4.1 及 Figure 4.2 中，點的颜色並不是該點在 CIE Lab 實際顯示出的顏色，僅只是用來代表該點對應的 RGB 值而已。

一般的電視和電腦監視器在色域上最大的差別在於電視的色域要求較高的亮度(Lightness)以及色度(Chroma)。觀察 Figure 4.1 及 Figure 4.2，可以看出 Panasonic TC-20LA1TN 與 FlexScane L5685 之間差異也大致呈現這種趨勢。不過或許是希望展現出更亮麗的顏色，Panasonic TC-20LA1TN 色域裡最亮的白色在出廠時的調校似乎經過抑制，使的亮度反而低於色域裡綠色的部分，這點和一般顯示器的特性有很大的差異。另外，觀察整個色域的分佈，我們可以發現 Panasonic TC-20LA1TN 的顏色整體而言較偏向紅色，尤其是藍色的部分事實上已經偏向於紫色。

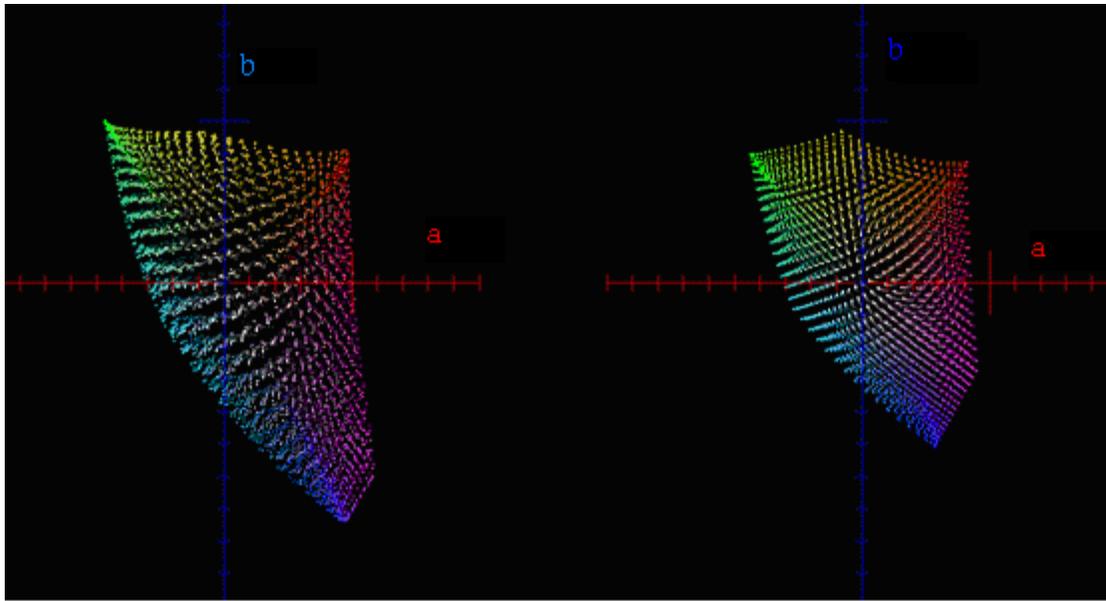


圖 4-1 EIZO FlexScan L568(右)以及 Panasonic TC-20LA1TN(左)色域在 ab 平面上的正交投影

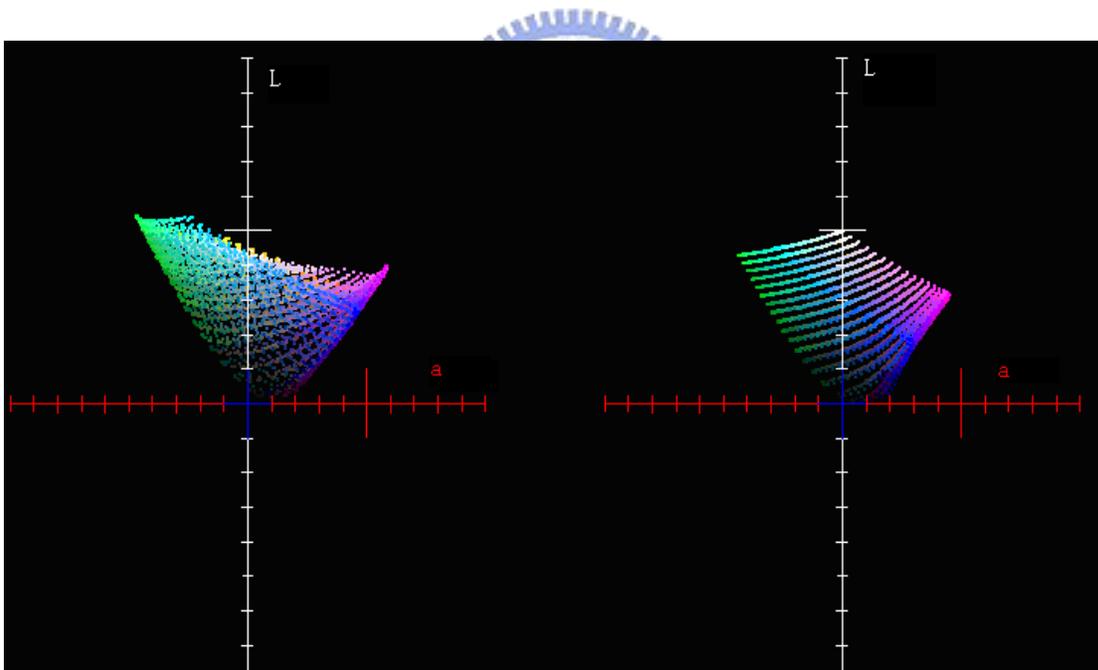


圖 4-2 EIZO FlexScan L568(右)以及 Panasonic TC-20LA1TN(左)色域在 La 平面上的正交投影

為了將 Panasonic TC-20LA1TN 的顏色以 EIZO FlexScan L568 顯示，我需要將 Panasonic TC-20LA1TN 的色域壓縮到 EIZO FlexScan L568 色域內。我選擇 Laihanen[15]所建議的移動方向讓顏色作移動，利用在第三章中提出的方法，我使用 $8 \times 8 \times 8$ 的控制點來進行前面設計的 Linear Programming，計算出每個顏色該移動

的量。一般而言，在 AMD Atholon 1.3G，1GB Ram 的電腦上使用 Simplex method[8][17]解我定義的 Linear Programming 問題可以在十分鐘以內完成。相對於完成整個色域對應流程所需要的時間，這樣算是非常快速的。

4.2 結果討論與未來展望

最後，Figure 4.3 到 Figure 4.10 是我使用 Laihanen[15]的建議作為移動方向，再分別利用傳統的 Clipping 方法，Compression 方法，以及第三章裡所提出的兩種方法(Equation 20 與 Equation 21)來計算顏色移動量得到的最後結果。用我提出的方法可以讓顏色間的 Gradient 不會如使用 Clipping 方法一樣大幅改變或消失，也不會像 Linear Compression 方法一樣讓圖片整體的飽和度大幅降低。

不過，因為色域對應的結果是針對當初所定的重製顯示設備- EIZO FlexScane L568，所以目前要在其他顯示設備上觀察色域對應的結果有先天性的困難。如何適當的視覺化色域對應的結果仍然需要繼續的研究。

另外，使用 Linear Programming 來進行色域對應最佳化雖然較穩定而且快速，不過為了滿足線性的要求也降低了設計色域對應演算法的自由度。尋找其他適當的最佳化目標和方法是未來可以繼續努力的目標。

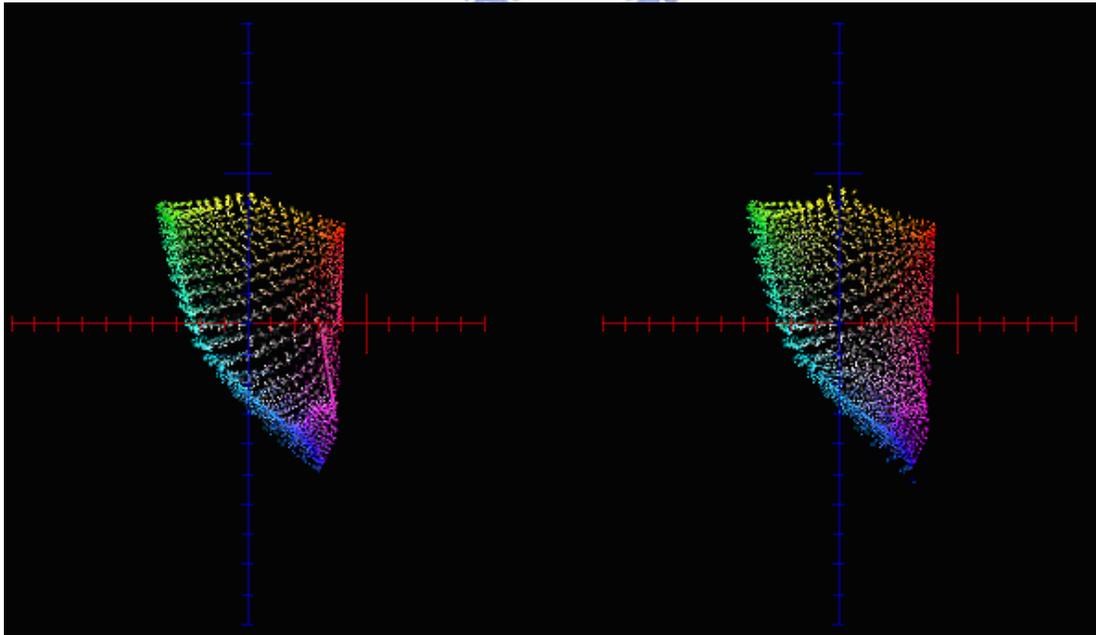


圖 4-3 使用傳統 Laihanen 的方法決定顏色壓縮的方向，(左)使用 Clipping 的對應結果，(右)使用 Linear Compression 的結果。



図 4-4 Before Gamut mapping



図 4-5 Clipping method



圖 4-6 Linear Compress method



圖 4-7 Use Equation 21 as Constraint



圖 4-8 Use Equation 20 as Constraint



參考文獻

1. J. Morovic and M. R. Luo, "The Fundamental of Gamut Mapping: A Survey", *Journal of Imaging Science and Technology*, vol 45, pp 283, 2001.
2. J. Morovic, "To Develop a Universal Gamut Mapping Algorithm", PhD thesis, University of Derby, Oct 1998.
3. R. C. Gonzales and R. E. Woods, "Digital Image Processing, Second Edition", 2002.
4. Hunt, "The reproduction of Colour in photography, printing, and television, Fourth Edition", pp 114, 1987.
5. N. Katoh and M. Ito, "Three-Dimensional Gamut Mapping Using Various Color Difference Formula and Color Spaces", *Journal of Electronic Imaging*, vol 8, pp 365, Oct, 1999.
6. X. Zhang and B. A. Wandell, "A Spatial Extension of CIELab for Digital Color Image Reproduction", <http://white.stanford.edu/html/xmei/scielab/scielab3/scielab3.html>
7. A.J. Jackson et al, "Aspects of Colour Management, Part II Characterisation of Four-Colour Imaging Devices and Colour Gamut Compression", *Colour Research and Application*, vol 23, 1998.
8. W.H. Press et al, "Numerical Recipes in C++, The art of Scientific Computing, Second Edition", Chapter 10, pp 343, 2002.
9. P.G. Herzog, "Analytical Color Gamut Representation", *Journal of Image Science and Technology*, vol 40, pp 516, 1996.
10. K.M. Braun et al, "Viewing Technology for Cross-Media Image Comparisons", *Color Research and Application*, vol 21, pp 6, 1996.
11. J. Morovic, M.R. Luo, "Calculating Medium and Image Gamut Boundaries for Gamut Mapping", *Color research and application*, vol 25, no 6, pp 394, Dec, 2000.
12. R. Saito and H. Kotera, "Image-Dependent Three-Dimensional Gamut Mapping Using Gamut Boundary Descriptor", *Journal of Electronic Imaging*, vol 13, pp 630, July, 2004.
13. J.J. Sara, "The automated reproduction of pictures with non-reproducible colors", PhD thesis, MIT, 1984.
14. J.M. Taylor et al, "Tektronix HVC: A uniform perceptual color system", *SID 89 digest*, 1998.

15. P. Laihanen, “Colour reproduction theory based on the principal of colour science”, *Proc IARAIGAI Conf*, pp 1, 1987.
16. L.W. MacDonald and J. Morovic, “Assessing the effects of gamut compression in the reproduction of fine art paintings”, *Proc IS&T/SID Color Imaging Conf*, pp 194, 1995.
17. T.H. Cormen, “Introduction to Algorithm”, MIT press, Chapter 29, pp 770, 2001.
18. S.K. Naik and C.A. Murthy, “Hue-Preserving Color Imaging Enhancement Without Gamut Mapping Problem” ,*IEEE Transaction on Image Processing*, vol 12, Dec ,2003.

