

# 新奇的影像序列色彩轉換演算法

王宗銘

中興大學資訊科學研究所

[cmwang@cs.nchu.edu.tw](mailto:cmwang@cs.nchu.edu.tw)

黃耀賢

中興大學資訊科學研究所

[yhuang@cht.com.tw](mailto:yhuang@cht.com.tw)

## 摘要

本文提出一個新奇的影像序列色彩轉換演算法，簡稱 ISCT 演算法(Image Sequences Color Transfer, ISCT)。ISCT 演算法之輸入為兩組影像；第一組為一張輸入影像(Input Image)  $I_1$ ，第二組為三張目標影像(Target Image)  $T_1, T_2, T_3$ 。輸出為具有色彩轉換、呈現微小變化、符合給定關連性(Correlation)等特性之影像序列(Image Sequences)  $\{S_i\}, i = 1, 2, \dots, N$ 。ISCT 演算法之步驟分為前置處理(Pre-process)、影像序列處理(Image Sequence Process)、後續處理(Post-process)等三大步驟。此演算法將輸入影像  $I_1$  與目標影像  $T_1, T_2, T_3$  以色彩轉換演算法(Color Transfer Algorithm)自動產生影像序列  $\{S_i\}$ 。我們以兩個模型對實驗系統作測試，測試結果顯示：ISCT 演算法確實可以自動迅速產生一系列具有微小色彩轉換變化且符合給定關連性之影像序列。總結本文，ISCT 演算法僅需使用四張影像即可自動、有效的產生符合使用者所規範的影像序列。ISCT 演算法可應用於媒體製作、視訊產生、剪輯等領域。

**關鍵詞：**影像序列，色彩轉換演算法

## 一、簡介

色彩(Color)在人類視覺的認知中扮演非常重要的角色，色彩轉換(Color Transfer)[2,6,9]是電腦圖學中的一個重要研究領域。色彩轉換常應用於影像處理(Image Processing)[8]或影像校正(Image Correction)等領域。

影像序列(Image Sequence)是由一系列單張、靜態影像所組合而成[5]。若將影像序列依序播出，基於人類視覺暫留(Persistence of Vision)之原理，此影像序列將具有動態、連續之變換效果。影像序列所能提供的資訊遠超過單張靜態影像，其應用相當廣泛，散見於視訊剪輯(Video Editing)、電腦動畫(Computer Animation)製作等領域。

研究學者雖然已提出半自動之單張影像色彩轉換演算法[6]，然而就我們所知，文獻上尚未發現針對影像序列來發展相關的色彩轉換演算法[4]。我們認為如能成功研發影像序列色彩轉換演算法，此演算法至少有三項不可忽略的優點。第一、節省成本；我們可將影像透過色彩轉移演算法產生影像序列，如此將有助於大量節省製作成本。第二、擴大涵蓋範疇；我們可將某影像序列轉換成為另一不同風格的影像序列。例如，我們可由白天到黃昏的影像序列，製作出春、夏、秋、冬等季節變化的影像序列。第三、提高附加價值；我們可利用研發的演算法來製作出現實世界所無法拍攝出來的場景。

發展此影像序列色彩轉換演算法也具有相當困難度。第一、由於人類視覺對於色彩變換具有相當的敏感度，因此所產生的影像序列必須具備連續、和緩的顏色變化方不至於產生不自然的視覺感受。第二、演算法必須儘可能的做到自動化，且所需的處理時間要盡可能的縮短。第三、演算法必須考慮能發展一個友善方便的使用者介面(User Interface)。使用者透過此介面可正確的輸入必要的參數，演算法處理後也會有立即的回饋，讓使用者可以即時檢視所產生的影像序列與各種量化數據。

本論文基於上述論點，提出一個新奇的影像序列色彩轉換演算法，簡稱 ISCT 演算法(Image Sequences Color Transfer, ISCT)。演算法首先利用一張輸入影像  $I_1$  (Input Image)，三張目標影像  $T_1, T_2, T_3$  (Target Image)，自動產生三張基底影像(Basis Images)  $B_1, B_2, B_3$ 。然後，演算法再依據使用者所給定的關連性(Correlation)特性，利用此三張基底影像產生影像序列  $\{S_i\}, i = 1, 2, \dots, N$ 。我們實現 ISCT 演算法，並發展出一個具有使用者介面(ISCT-UI)的影像序列色彩轉換實驗系統。此介面可供使用者輸入必要的參數，經系統計算後，快速的回饋色彩轉換之結果。我們以兩個模型對系統作測試，結果證實 ISCT 演算法確實能同時滿足平滑轉變性、效率性與方便性的要求，產生一系列具有色彩轉換、呈現微小變

化、符合給定關連性之影像序列。

本文架構如下：第二節簡介色彩空間與色彩轉換的相關研究；第三節說明影像序列色彩轉換原理與流程；第四節展示將此轉換方法運用於 2 個不同的測試模型所產生的執行結果與分析；最後總結本文並建議未來工作。

## 二、相關研究

目前影像序列研究主要集中於影像序列分析(Image Sequence Analysis)[5, 10]與影像序列應用等。文獻上僅有針對單張影像[2,6,9]，但並無針對影像序列做色彩轉換之研究。由於單張影像之轉換係在  $l\alpha\beta$  色彩空間(Color Space)[7]中完成，故本節將首先說明  $l\alpha\beta$  色彩空間(Color Space)，接著回顧單張影像色彩轉換演算法，以便能進一步的瞭解色彩轉換演算法之精義，並據此推導出新的影像序列色彩轉換演算法。

### 2.1 $l\alpha\beta$ 色彩空間

色彩空間是電腦描述與儲存影像顏色資訊之方式[8]。各種色彩空間以不同的維度(Dimension)來處理顏色，例如 RGB 與 LMS 色彩空間。

RGB 與 LMS 等色彩空間有一個共同的特性，就是不同頻道在色彩空間中存在著相依(Correlation)的關係，如圖 1 所示。圖 1 係由一張影像中隨機取出 1000 個像素點(Pixel)，並畫出這些像素點的 L-M 與 L-S 關係圖。由圖顯示 L-M、L-S 均為線性相依，此相依特性致使我們在改變某一頻道(Channel)的數值時，無法將改變僅侷限於該頻道。

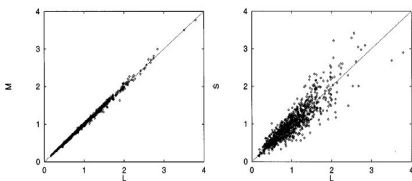


圖 1 由影像隨機選取 1000 個 pixel 的 L-M 與 L-S 分佈圖[7]

1983 年 Buchsbaum 等人研究發現：若利用一些線性轉換，則可將人類視覺系統所接受的顏色刺激轉換成 3 個互無關連的頻道[1]。1998 年 Ruderman 延續 Buchsbaum 等人之研究，提出一個新的色彩空間 -  $l\alpha\beta$  色彩空間[7]。 $l\alpha\beta$  色彩空間中的  $l$  頻道表亮度(Luminance)， $\alpha$  頻道代表黃色、藍色所組成的頻道， $\beta$  頻道則帶表紅色、綠色所組成的頻道。圖 2 係由影像中隨機選取 1000 個像素點，

並將顏色由 RGB 色彩空間轉換至  $l\alpha\beta$  色彩空間後，繪出  $l-\alpha$ 、 $l-\beta$ 、 $\alpha-\beta$  的相互關係分佈圖。由此圖我們可明顯看出： $l-\alpha$ 、 $l-\beta$ 、 $\alpha-\beta$  具有垂直或水平的分佈情形，而非對角線之分佈狀況。因此，此三個頻道具有非常低的關連性，此不同於 RGB、LMS 色彩空間之相依情形。

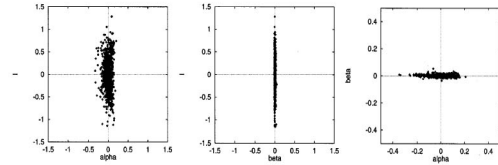


圖 2 由影像隨機選取 1000 個 pixel 的  $l-\alpha$ 、 $l-\beta$ 、 $\alpha-\beta$  分佈圖[7]

利用  $l\alpha\beta$  色彩空間中各個頻道低相依性之性質，若我們想對影像做色彩校正時，我們僅需將原先數值所在之色彩空間先行轉換至  $l\alpha\beta$  色彩空間，便可針對各個頻道的數值作校正，然後我們僅需再將最後之數值轉換回原先的色彩空間即可完成校正手續。

### 2.2 單張影像色彩轉換演算法

Reinhard 於 2001 年提出了利用統計學的技巧，來達成將原始影像(Source Image)的顏色特徵轉移到另一張目標影像(Target Image)的單張影像色彩轉換演算法(Color Transfer between Images) [6]。

整個轉換步驟首先將原始影像與目標影像先由 RGB 色彩空間轉變至  $l\alpha\beta$  色彩空間。接著演算法計算原始影像與目標影像的所有像素(Pixel) 在  $l$  頻道  $\alpha$  頻道與  $\beta$  頻道各別的平均值(Mean)與標準差(Standard Deviation)。然後將原始影像每一個像素的  $l\alpha\beta$  值各別減去原始影像  $l\alpha\beta$  頻道的平均值，接著乘上目標影像與原始影像的標準差比值，再各自加上目標影像的  $l\alpha\beta$  頻道的平均值如算式 1 所示，最後將處理後的原始影像由  $l\alpha\beta$  色彩空間轉變回 RGB 色彩空間，目標影像的顏色已順利被轉換至原始影像中。

$$l_i = \frac{\sigma_l^i}{\sigma_{BaseImage1}^l} (l_{BaseImage1} - \langle l \rangle_{BaseImage1}) + \langle l \rangle_i$$

$$\alpha_i = \frac{\sigma_\alpha^i}{\sigma_{BaseImage1}^\alpha} (\alpha_{BaseImage1} - \langle \alpha \rangle_{BaseImage1}) + \langle \alpha \rangle_i \quad (1)$$

$$\beta_i = \frac{\sigma_\beta^i}{\sigma_{BaseImage1}^\beta} (\beta_{BaseImage1} - \langle \beta \rangle_{BaseImage1}) + \langle \beta \rangle_i$$

Reinhard 所提出的色彩轉換演算法有其優點。第一、簡潔性。僅需原始影像與目標影像，即可將色彩予以轉移。第二、即時性。色

彩轉移所需要的計算時間短暫。第三、自動完成性。色彩轉移演算法可以自動完成，無需使用者的介入。然而相對的，此演算法亦有兩個主要缺失。其一、演算法屬於高勞力處理 (Labor-Intensive Processing)。使用者需針對影像不斷嘗試錯誤，這種作法屬於高勞力性處理，無從預估其成功的時間。其二、無可修正性。當使用者輸入原始、目標影像時，在執行色彩轉移前並無法得知色彩轉移是否成功。若色彩轉移不成功，Reinhard 並未提出任何方法予以修正或補救。

我們認為 Reinhard 所提出的色彩轉換演算法無法適用於影像序列的色彩轉移除了無可修正性因素使然外，最主要的困難在於吾人需要尋得適合的目標影像序列，方能對此原始、目標影像序列的每張畫面做 1 對 1 的色彩變換。此外影像序列色彩轉換應存在若干的彈性空間，若我們僅能尋得數張目標影像，我們也應能發展適合的演算法，並在此條件下能成功的完成影像序列的色彩轉移。據此，我們提出了利用一張輸入影像與三張目標影像的概念做影像序列的色彩轉移演算法。我們在下一節中詳述我們所發展的影像序列色彩轉換演算法。在第四節中，我們將透過兩組測試模型來驗證色彩轉移的結果。

### 三、影像序列色彩轉換演算法

本節敘述影像序列色彩轉換演算法之原理，並說明所需的輸入、輸出資訊與轉換步驟。色彩轉換影像序列的原理如圖 3 所示；使用者給定一張輸入影像  $I_I$  與三張目標影像  $T_1$ 、 $T_2$ 、 $T_3$ ，經由色彩轉換後可產生三張基底影像  $B_1$ 、 $B_2$ 、 $B_3$ 。透過此三張基底影像與這些基底影像之間不同的影像色調變化方式，即可使用影像序列色彩轉換演算法來產生其他非在給定時間內的成圖影像，且所產生的影像序列與基底影像間依使用者的設定而有一定的關連（線性、非線性等）。因為 3 點決定一條曲線，所以製作一段色彩變換具有曲線變化特性之影像序列至少需要三張基底影像。此三張

基底影像可由輸入影像與三張目標影像作色彩轉換產生，亦可由輸入影像調整色調產生。而使用者只要將所產生的不同影像序列串連即可製作出無限長度的色彩轉換影像序列。

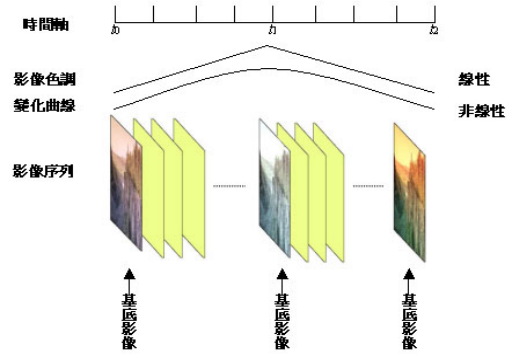


圖 3 影像序列色彩轉換原理

影像序列色彩轉換演算法可分為前置處理、影像序列處理與後續處理 3 個步驟。

前置處理步驟首先將輸入影像  $I_I$  色調與三張目標影像  $T_1$ 、 $T_2$ 、 $T_3$  轉換至  $l\alpha\beta$  色彩空間  $I_{I(RGB)} \Rightarrow I_{I(l\alpha\beta)}$ ， $T_{1(RGB)} \Rightarrow T_{1(l\alpha\beta)}$ ， $T_{2(RGB)} \Rightarrow T_{2(l\alpha\beta)}$ ， $T_{3(RGB)} \Rightarrow T_{3(l\alpha\beta)}$ 。

在色彩空間轉換部分，我們可以透過簡單的矩陣運算如算式 (2)(3)(4) 所示，即可將三張基底影像由 RGB 色彩空間轉換至  $l\alpha\beta$  色彩空間。

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.3811 & 0.5783 & 0.0402 \\ 0.1967 & 0.7244 & 0.0782 \\ 0.0241 & 0.1288 & 0.8444 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2)$$

$$\begin{aligned} L &= \log L \\ M &= \log M \\ S &= \log S \end{aligned} \quad (3)$$

$$\begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix} \quad (4)$$

將輸入影像與三張目標影像轉換至  $l\alpha\beta$  色彩空間後，再以 Reinhard 的單張影像色彩轉換演算法將輸入影像與目標影像轉換成三張基底影像，即  $\{I_{I(l\alpha\beta)}, T_{1(l\alpha\beta)}\} \Rightarrow B_{I(l\alpha\beta)}, \{I_{2(l\alpha\beta)},$

$T_{2(l\alpha\beta)} \Rightarrow B_{2(l\alpha\beta)}$ 、 $\{I_{3(l\alpha\beta)}、T_{3(l\alpha\beta)}\} \Rightarrow B_{3(l\alpha\beta)}$ 。在將  $B_{1(l\alpha\beta)}$ 、 $B_{2(l\alpha\beta)}$ 、 $B_{3(l\alpha\beta)}$  交給影像序列處理步驟使用前，我們需先將此三張影像轉換至 RGB 色彩空間  $B_{1(l\alpha\beta)} \Rightarrow B_{1(RGB)}$ ， $B_{2(l\alpha\beta)} \Rightarrow B_{2(RGB)}$ ， $B_{3(l\alpha\beta)} \Rightarrow B_{3(RGB)}$ ，以檢查 Reinhard 的單張影像色彩轉換演算法有可能產生的色彩溢位問題。若有像素產生溢位情形，若小於 0，我們以 0 來取代原先像素的值，若超過 255，我們以 255 取代原先的值，最後再將修正後的三張基底影像轉換至  $l\alpha\beta$  色彩空間以供影像序列處理步驟使用。

影像序列處理步驟主要工作為將前置處理步驟所輸入的  $B_{1(l\alpha\beta)}$ 、 $B_{2(l\alpha\beta)}$  與  $B_{3(l\alpha\beta)}$  與相關設定產出影像序列  $S_i(l\alpha\beta)$  ( $1 < i < N$ )，以供後續處理步驟使用。影像序列處理步驟呼叫了計算平均值與標準差 (CalMeanVar())、計算轉換曲線參數 (CalFunctionParameter()) 與產生影像序列 (CreateSi()) 三個功能函數。此 3 個功能函數其執行功能說明如下：

1. 計算平均值與標準差功能函數：此函數計算三張基底影像在  $l\alpha\beta$  色彩空間中所有像素的  $l$  頻道、 $\alpha$  頻道與  $\beta$  頻道的平均值與標準差，以供計算轉換曲線參數功能函數使用。

2. 計算轉換曲線參數功能函數：此函數依據使用者所選定整個影像序列之不同色彩轉換方式，針對  $l\alpha\beta$  三個頻道，各別計算所需之轉換曲線參數。例如選擇色彩轉換方式為線性變化， $B_{2(l\alpha\beta)}$  在影像序列中為第  $M$  張影像， $B_{3(l\alpha\beta)}$  在影像序列中為第  $N$  張影像，則以基底影像間的色調變化計算通式  $y = ax + b$ ，將之前計算平均值功能函數所計算的  $B_{1(l\alpha\beta)}$  與  $B_{2(l\alpha\beta)}$  在  $l$  頻道的平均值  $B_{1\text{-mean}}$  與  $B_{2\text{-mean}}$  以  $(x_1, y_1) = (0, B_{1\text{-mean}})$  與  $(x_2, y_2) = (M, B_{2\text{-mean}})$  帶入計算通式  $y = ax + b$ ，可得  $a$  與  $b$  的值，其意義為影像  $B_{1(l\alpha\beta)}$  至影像  $B_{2(l\alpha\beta)}$  之間在  $l$  頻道方面平均值的變化方程式。同理可得  $B_{2(l\alpha\beta)}$  至影像  $B_{3(l\alpha\beta)}$  之間在  $l$  頻道方面平均值的變化

方程式。以此方法類推可得在  $l\alpha\beta$  頻道的所有平均值與標準差的變化方程式。

若使用者選擇色調轉換曲線為拋物線，則利用拋物線通式  $y = ax^2 + bx + c$  與 3 組資料值  $(x_1, y_1)$ 、 $(x_2, y_2)$  與  $(x_3, y_3)$  解聯立方程式可得  $a$ 、 $b$  與  $c$  的通式為：

$$a = \frac{(y_1 - y_2)(x_1 - x_3) - (y_1 - y_3)(x_1 - x_2)}{(x_1 - x_2)(x_1 - x_3)(x_2 - x_3)}$$

$$b = \frac{(y_1 - y_2)(x_1^2 - x_3^2) - (y_1 - y_3)(x_1^2 - x_2^2)}{(x_1 - x_2)(x_1 - x_3)(x_2 - x_3)} \quad (5)$$

$$c = \frac{y_1(x_1 - x_2)(x_1 - x_3)(x_2 - x_3) + x_1x_3(y_1 - y_2)(x_1 - x_2) - x_1x_2(y_1 - y_3)(x_1 - x_2)}{(x_1 - x_2)(x_1 - x_3)(x_2 - x_3)}$$

因此我們以  $(0, B_{1\text{-mean}}) = (x_1, y_1)$ 、 $(M, B_{2\text{-mean}}) = (x_2, y_2)$  與  $(N, B_{3\text{-mean}}) = (x_3, y_3)$  帶入上述(5)的式子，可得參數  $a$ 、 $b$  與  $c$  的值與影像  $B_{1(l\alpha\beta)}$  至影像  $B_{3(l\alpha\beta)}$  之間在  $l$  頻道方面平均值的拋物線變化方程式。同理亦可得  $l\alpha\beta$  頻道的所有平均值與標準差的曲線變化方程式。

3. 產生影像序列功能函數：此函數依據之前計算轉換曲線參數功能函數所計算的成圖影像  $l$  頻道、 $\alpha$  頻道與  $\beta$  頻道之平均值與標準差的曲線變化方程式，將第  $i$  張影像以  $x_i = i$ ，( $1 < i < N$ ) 帶入曲線變化方程式中，可求出所有  $l$  頻道、 $\alpha$  頻道與  $\beta$  頻道的平均值  $\langle l \rangle_i$ 、 $\langle \alpha \rangle_i$ 、 $\langle \beta \rangle_i$  與標準差  $\sigma_i^l$ 、 $\sigma_i^\alpha$ 、 $\sigma_i^\beta$ ，帶入色彩轉換通式，即可產生成圖影像  $S_i(l\alpha\beta)$  ( $1 < i < N$ )。整個影像轉換步驟可用下列虛擬碼表示：

```
Image_Sequence_Process(B1(lαβ), B2(lαβ), B3(lαβ))
{
    CalMeanVar();
    CalFunctionParameter();
    While(No. of Image Sequence <= N)
    {
        CreateSi();
    }
    return Si(lαβ);
}
```

後續處理步驟將影像序列處理步驟所產生在  $l\alpha\beta$  色彩空間的影像序列  $S_{i(l\alpha\beta)}$  ( $1 < i < N$ ) 轉換回 RGB 色彩空間的影像序列  $S_{i(RGB)}$  ( $1 < i < N$ )，轉換方法可透過算式(2)(3)(4)的反運算即可轉換回 RGB 色彩空間，並可視需要直接輸出影像序列或轉換成各種視訊 ( Video ) 格式輸出。

透過上述影像序列色彩轉換演算法，使用者只要輸入一張輸入影像，選擇基底影像間要產生多少的成圖影像，並選擇色調變化的方式，即可自動產生一連串成圖影像，且成圖影像的色調變化依據使用者所選擇的變化方式。此演算法可以最少的成本產生影像序列，且可將所產生的不同影像序列連結，製作出無限長度的色彩轉換影像序列。以下我們以兩組測試模型來驗證此演算法的實際執行結果。

#### 四、結果

由於影像序列色彩轉換至今並無相關研究，亦無標準測試模型，在此我們以峽谷與綠色隧道兩組測試模型驗證執行結果。其中峽谷模型我們以色彩轉換法將輸入影像轉換成三張可代表峽谷早上、中午及黃昏的基底影像，使得所產生的影像序列具有由早上經過中午到黃昏的時光飛逝效果，並將之與人類視覺對於自然場景早中晚光影認知作比較。而綠色隧道模型同樣由輸入影像經色彩轉換成三張具有黃色、綠色與紫色色調的基底影像，所產生的影像序列具有由黃色轉至綠色最後至紫色的超現實效果。我們將檢驗此產生的影像序列色彩變換是否自然，可將此種影像序列運用於電影動畫等特效中。本節首先表列出測試過程中所使用的發展測試平台，接著說明我們所開發的影像序列色彩轉換系統使用者介面功能，然後以上述兩組測試模型，來驗證影像序列色彩轉換演算法的執行結果。我們將列出所產生的影像序列畫面， $l\alpha\beta$  色彩空間的轉換曲線，執行時間與畫面差異比較表。畫面差異部分我們以 RMS 差異值[3]來表示影像序列前後兩張畫面的差異情形，RMS 差異值計算方式如下：

$$RMS = \sqrt{\sum pixels [\bar{L}_{reference} - \bar{L}_{image}]^2 / N} \quad (6)$$

##### 4.1 影像序列色彩轉換介面

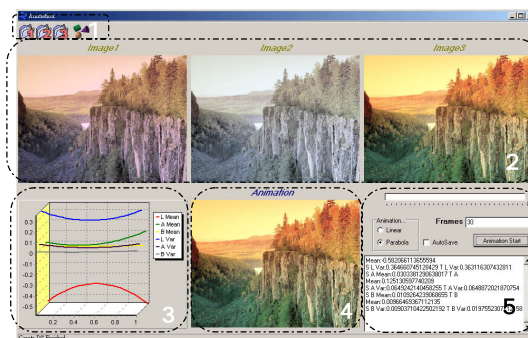






圖 4 影像序列色彩轉換系統使用者介面

圖 4 為我們開發的影像序列色彩轉換系統使用者介面，透過此介面使用者可輸入基底影像，設定轉換方式與成圖影像數目以產生整個影像序列。以下我們分 5 個部分說明使用者介面功能：

- 1. 工具列:** 工具列有 4 個按鈕，其中 3 個    為基底影像輸入按鈕，第 4 個按鈕  為色彩空間轉換按鈕。
- 2. 基底影像區：** 顯示所輸入之基底影像。
- 3. 轉換曲線顯示區：** 顯示整個影像序列從第一張影像至最後一張影像的  $l\alpha\beta$  平均值與標準差變化情形。此 6 條曲線分別代表  $l$  色彩空間平均值與標準差變化曲線， $\alpha$  色彩空間平均值與標準差變化曲線， $\beta$  色彩空間平均值與標準差變化曲線。
- 4. 影像序列動畫顯示區：** 顯示所產生的影像序列動畫。
- 5. 設定功能區：** 設定功能區提供轉換曲線設定、畫面個數設定、轉換過程訊息顯示功能、畫面捲軸功能、自動儲存設定與啟動按鈕等功能。

##### 4.2 峽谷測試模型

在峽谷測試模型中，我們的輸入影像如圖 5 為一張峽谷影像，經由與圖 5 中三張目標影像色彩轉換後成為三張具有早上，中午與黃昏色調的基底影像，其解析度為 320x240 像素。

我們設定早上基底影像與中午基底影像間產生十三張成圖影像，中午基底影像與黃昏基底影像間產生十四張成圖影像，整個影像序列加上基底影像共有三十張影像。而色彩轉換方式部分，採取線性與拋物線兩種變化方式。



圖 5 峽谷輸入影像(上層), 三張目標影像(中層), 轉換後早晨, 中午與黃昏色調基底影像(下層)

經過影像序列色彩轉換後, 產生如圖 6 (採用線性轉換) 與圖 7 (採用拋物線轉換) 所示的三十張影像序列, 其中外框為黑線的影像為輸入的基底影像。



圖 6 峽谷模型影像序列(線性轉換)



圖 7 峽谷模型影像序列(拋物線轉換)

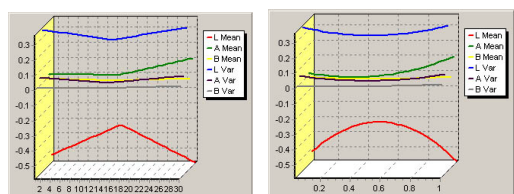


圖 8 峽谷模型線性轉換(左)與拋物線轉換(右)在  $L\alpha\beta$  色彩空間的轉換曲線

我們以 RMS 差異值來驗證輸出影像序列中相鄰畫面的差異。

表 1 峽谷模型 RMS 差異表(線性轉換)

$i$	$RMS(S_i-S_{(i-1)})$	$i$	$RMS(S_i-S_{(i-1)})$	$i$	$RMS(S_i-S_{(i-1)})$
2	2.958501	12	3.194869	22	4.172726
3	3.101741	13	3.177671	23	4.125185
4	3.170869	14	3.207430	24	4.026994
5	3.147431	15	3.227489	25	3.886681
6	3.182105	16	4.472552	26	3.802374
7	3.152728	17	4.520751	27	3.741695
8	3.230110	18	4.539006	28	3.649553
9	3.192140	19	4.581950	29	3.591635
10	3.184265	20	4.460069	30	3.559347
11	3.221118	21	4.395413		

表 2 峽谷模型 RMS 差異表(拋物線轉換)

$i$	$RMS(S_i-S_{(i-1)})$	$i$	$RMS(S_i-S_{(i-1)})$	$i$	$RMS(S_i-S_{(i-1)})$
2	5.823286	12	1.571937	22	3.91657
3	5.616491	13	1.191709	23	4.410389
4	5.328909	14	0.911679	24	4.764669
5	4.917811	15	0.865514	25	5.036223
6	4.390394	16	1.075023	26	5.276625
7	3.853068	17	1.43243	27	5.482553
8	3.348807	18	1.864812	28	5.651544
9	2.872907	19	2.338442	29	5.800963
10	2.418215	20	2.844931	30	5.929304
11	1.986686	21	3.38894		

執行時間部分三十張影像採用線性與拋物線轉換的平均執行時間介於 0.56 秒至 0.57 秒間。由表 1 與表 2 的 RMS 差異值可以清楚看到在峽谷測試模型所產生的影像序列中, 若是採用線性變化方式, RMS 差異值變化不大, 可是若是採用拋物線變化方式, 則 RMS 差異值變化由差異大遞減至第十四與十五張時差異最小, 然後再遞增, 與  $L\alpha\beta$  色彩空間的轉換曲線變化方式相當。

選擇不同曲線將影響影像序列色彩變化情形, 以峽谷測試模型為例, 我們從圖 6 與圖 7 的影像序列畫面可看出, 若選擇線性變化,

影像序列的色彩變化過程比較單調，色調由基底影像 1 的色調轉換至基底影像 2 的色調再轉至基底影像 3。若選擇拋物線的變化方式，則如圖 7 所示，影像序列色彩變化方式較接近人類視覺認知的由早到晚的色彩變化方式，即在接近早上與黃昏時色彩變化差異較大，其餘時間差異不大的特性。

### 4.3 綠色隧道測試模型

在綠色隧道測試模型中，我們的輸入影像為圖 9 最上層之綠色隧道影像，經由圖 9 中間的三張影像色彩轉換後，色調調整成為圖 9 最下層三張具有黃色、綠色與紫色色調的影像，其解析度為 320x240 像素。我們設定黃色色調基底影像與綠色色調基底影像間產生十三張成圖影像，綠色色調基底影像與紫色色調基底影像間產生十四張成圖影像，整個影像序列加上基底影像共有三十張影像。而色彩轉換方式部分，採取線性與拋物線兩種變化方式。

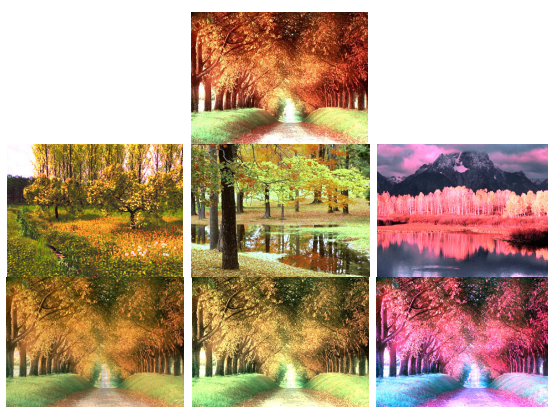


圖 9 綠色隧道輸入影像（上層），三張目標影像（中層）與轉換後色調偏黃、色調偏綠與色調偏紫的基底影像（下層）

經過影像序列色彩轉換後，產生如圖 10（採用線性轉換）與圖 11（採用拋物線轉換）所示的三十張影像序列，外框為黑線的影像為輸入的基底影像。

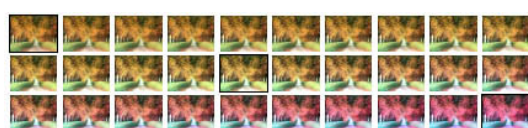


圖 10 綠色隧道模型影像序列(線性轉換)

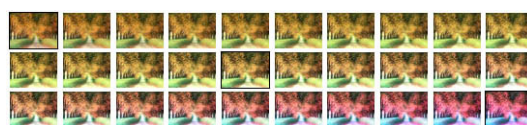


圖 11 綠色隧道模型影像序列(拋物線轉換)

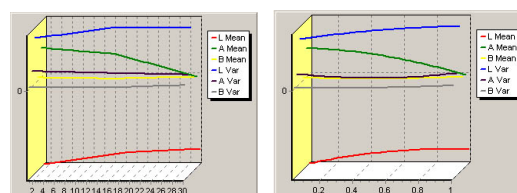


圖 12 綠色隧道模型線性轉換（左）與拋物線轉換（右）在  $L\alpha\beta$  色彩空間的轉換曲線

表 3 綠色隧道模型 RMS 對照表（線性轉換）

$i$	$RMS(S_i-S_{(i-1)})$	$i$	$RMS(S_i-S_{(i-1)})$	$i$	$RMS(S_i-S_{(i-1)})$
2	2.956044	12	3.316249	22	3.775561
3	2.789507	13	3.167478	23	3.880527
4	2.806237	14	3.217171	24	3.790841
5	2.822103	15	3.155787	25	3.833524
6	3.019843	16	3.694782	26	3.837386
7	3.105009	17	3.754618	27	3.817879
8	3.138371	18	3.733139	28	3.838692
9	3.212190	19	3.832319	29	3.821272
10	3.188631	20	3.808452	30	3.887153
11	3.228999	21	3.841877		

表 4 綠色隧道模型 RMS 對照表(拋物線轉換)

$i$	$RMS(S_i-S_{(i-1)})$	$i$	$RMS(S_i-S_{(i-1)})$	$i$	$RMS(S_i-S_{(i-1)})$
2	2.98029	12	1.878815	22	3.330789
3	2.85488	13	1.933376	23	3.51879
4	2.712035	14	2.028256	24	3.706099
5	2.556655	15	2.147701	25	3.874653
6	2.394541	16	2.288478	26	4.024879
7	2.237665	17	2.44277	27	4.149446
8	2.098009	18	2.610947	28	4.268107
9	1.984408	19	2.778395	29	4.38547
10	1.902816	20	2.958337	30	4.510337
11	1.869564	21	3.1422		

三十張影像採用線性與拋物線轉換的平均執行時間約 0.56 秒。由表 3 與表 4 的 RMS

差異值可看到綠色隧道測試模型產生的影像序列，採用線性變化方式，RMS 差異值主要在 2.7 至 3.8 間來回變化，差異不大。若是採用拋物線變化方式，則 RMS 差異值變化由差異大遞減至第十一與十二張時差異最小，然後再遞增，與  $1\alpha\beta$  色彩空間的轉換曲線變化方式相當。透過 RMS 差異值可輕易看出畫面變化情形，可是若我們直接比較圖 10 與圖 11，影像序列色調變化情形並不如圖 6 與圖 7 強烈，這主要是因為人眼視網膜對於光線亮度的接收細胞數目遠大於接收色彩細胞數目的緣故。

透過峽谷與綠色隧道兩組測試模型可驗證影像序列色彩轉換演算法確實可行，效果非常好。所產生的影像序列具有人眼不易察覺任何人造的感覺，且具有令人滿意的執行速度。

## 五、 結論與未來工作

本論文所提出的影像序列色彩轉換演算法利用一張輸入影像與三張目標影像透過少數設定便可自動快速的製作出使用者所想要的色彩變化影像序列。此演算法由兩組測試模型的執行結果可知所產生的影像序列色彩轉換效果非常好，不會有任何不自然的變化情形發生，且生產成本非常少。再者透過結合不同變化曲線所產生的影像序列，可產生長度與變化無限的影像序列，具有非常大的應用範圍。最後我們亦提供一個友善的使用者介面，使用者可透過方便的使用者介面，觀看所產生的影像序列，立即做適當的修正。總結本研究，我們認為影像序列色彩轉換將會應為我們提出的演算法變的更有效、更能擴大應用的範疇。

未來將增加更多不同的曲線變換方式，並結合人類視覺認知領域以瞭解多種曲線方程式與不同基底影像之間畫面變化與人類視覺及情緒的變化關係。此外，透過結合影像序列特效，以產生結合色彩與特效變化之影像序列

亦是嘗試的工作。未來亦將此演算法應用於電腦 3D 自然場景動畫製作；透過輸出數張影像以作為基底影像，便可製作完成整個自然場景動畫，將可大幅加速動畫製作時間。

## 參考文獻

- [1] G. Buchsbaum and A. Gottschalk, "Trichromacy, Opponent Colour Coding and Optimum Colour Information Transmission in the Retina," in *Proc. of Royal Society*, London, Ser. B 220, pp. 89-113, 1983.
- [2] G. R. Greenfield and D. H. House, "Image Re-coloring Induced by Palette Color Associations," in *Proc. of WSCG 2003*, pp. 137-145, February 2003.
- [3] E. Lafortune, *Mathematical Model and Monte Carlo Algorithm for Physically Based Rendering*, Ph. D. Dissertation, Department of Computer Science, Katholieke Universiteit Leuven, Belgium, 1996.
- [4] J. M. Pinel and H. Nicolas, "Moving Shadows Segmentation on Lambertian Surfaces in Video Sequences," in *Proc. of VCIP*, Lugano, Switzerland, pp. 235-243, July 2003.
- [5] T. R. Reed, *Digital Video Processing*, CRC Press, 2001.
- [6] E. Reinhard, M. Ashikhmin, B. Gooch and P. Shirley, "Color Transfer between Images," *IEEE Computer Graphics and Applications*, vol. 21, no. 5, pp. 34-41, September 2001.
- [7] D. L. Ruderman, T. W. Cronin and C. C. Chiao, "Statistics of Cone Responses to Natural Images: Implications for Visual Coding," *Journal of Optical Society of America*, vol. 15, no. 8, pp. 2036-2045, 1998.
- [8] S. E. Umbaugh, *Computer Vision and Image Processing*, Prentice-Hall Inc., 1998.
- [9] T. Welsh, M. Ashikhmin and K. Mueller, "Transferring Color to Greyscale Images," in *Proc. of ACM SIGGRAPH 2002*, pp. 277-280, July 2002.
- [10] G. K. Wu and T. R. Reed, "Image Sequence Processing using Spatiotemporal Segmentation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 9, no. 5, pp. 798-807, August 1999.