

具可逆能力的色彩轉換演算法

A Reversible Color Transformation Algorithm

王宗銘、李鴻昌、黃明隆、蔡宗佑、黃彥翔、陳昱仁、曾楷倫、黃彥人
國立中興大學資訊科學研究所
cmwang@cs.nchu.edu.tw

摘要

本文提出一個具有可逆能力的色彩轉換演算法。我們的演算法有別於過去學者提出的色彩轉換演算法，我們結合資訊隱藏的技術，將色彩轉換過程中所需的重要變數隱藏於結果影像中，而當使用者對於色彩轉換的結果不甚滿意或是遺失了原始的來源影像時，可以透過我們提出的演算法將結果影像還原成原始的來源影像，可再進行下一次的色彩轉換。根據試驗結果，我們提出的方法確實能夠將結果影像還原。演算法具三項優點：(1) 記憶體空間的節省；(2) 更多次的色彩轉換；(3) 快速的還原時間。演算法適合被使用於行動裝置或是即時系統上。

關鍵字：可逆的色彩轉換 (Reversible Color Transformation)、資訊隱藏 (Information Hiding)。

一、簡介

在日常生活中，人類藉由視覺所獲得知的資訊佔所有獲取資訊的百分之八十以上，其中，色彩對視覺的影響力，更是佔有相當的一席之地，例如我們所使用的衣服，家具，甚至是科技產品，對於產品本身色彩上的設計均越趨講究，藉此提高銷售成績，此外，在許多的研究領域中，電腦圖學、視訊多媒體、影像處理等，常有與色彩有關的研究成果

被提出，由以上可知，與色彩相關的研究確實有其必要性與重要性[14]。

生活週遭的環境變化往往是影響我們最深的，在同一景致中，依時間的不同，早晨與黃昏；依天氣的不同，晴天或陰天；依季節的不同，春、夏、秋、冬，均會呈現多樣性的變化，據此，Reinhard 等人於 2001 年首次提出影像間的色彩轉換 (Color Transfer between Images) 技術[9]，透過使用一不同的目標影像，達到對同一場景的影像作多樣性的變換，隨後也有學者陸續投入與色彩轉換相關之研究 [3, 4, 5, 10, 12, 13, 15]，總結相關文獻，色彩轉換演算法，其主要的概念為企圖將來源影像 (Source Image) 與目標影像 (Target Image) 經過一轉換機制的處理後，以自動將目標影像上的色彩特徵轉換至來源影像上，以使得結果影像的色彩分佈相似於目標影像的色彩分佈，更可進一步使單張的來源影像搭配多張不同的目標影像，分別得到不同色彩風格的結果影像。

不過，目前在文獻上並沒有學者探討關於色彩轉換之後，將結果影像還原的問題，因此色彩轉換後必須保留原始的來源影像，才能於使用者不滿意轉換結果時，再次使用其他目標影像來進行色彩轉換，不過，若使用者遺失最初的來源影像，將永遠無從得知來源影像原來的風貌，無法使用同樣的原始影像與多張不同的目標影像進行色彩轉換以茲比較，此外，未來若是將色彩轉換技巧嵌入行動裝置，或是即時系統中，如數位相機，可拍照的行動電話以及 PDA，因此儲存空間的問題是可以預見的，若可以直接還原結果影像至原來的風貌，將可以有效

的降低所需的儲存空間。

本論文首創一個具可逆能力的色彩轉換演算法 (A Reversible Color Transformation Algorithm)，我們的主要目標為企圖使已經由色彩轉換後的結果影像具有可還原為原始來源影像的能力。我們的演算法架構簡單易懂且處理速度上亦很快速，這對於未來將此技術應用於行動裝置上時有莫大的幫助。

由於我們企圖使經由色彩轉換後的結果影像具有可還原的能力，因此，必需要將色彩轉換過程中，一些重要的變數隱藏於結果影像上，主要的參數有來源影像與目標影像的平均值 (Mean) 與標準差 (Standard Deviation)，如此一來，即可在結果影像需要還原時，將所需變數取出，再進一步經過色彩還原運算式的計算，便能夠獲得原始的來源影像。

2004 年, Ni 等人提出一具有較低遺失率與較佳隱藏效果的影像資訊隱藏演算法[8]，由於此演算法的特性符合於我們的基本需求：低遺失率與隱藏效果佳；不過，在低遺失率部份無法勝任我們的需求，我們期望隱藏至影像上的變數必須能被百分之百地完全取出，針對此目標，透過展頻 (Spread Spectrum) 技術及投票法 (Ballot box) 與 Ni 的方法結合，將原始影像及目標影像的一些重要變數隱藏至色彩轉換的結果影像中，經由我們的試驗，改善後的資訊隱藏方法，能夠確實地將資訊藏入影像中，並且亦能夠完整地取出，取出後的數值與原始的數據僅有微小的誤差，因此在實行結果影像的反運算處理後，所獲得的原始來源影像與原先我們所使用的來源影像，他們之間的 PSNR 值均大於 30dB 以上，因此在視覺上看不出有任何太大的差異性。

我們結合色彩轉換與資訊隱藏技術提出一具可逆能力的色彩轉換演算法，此演算法可以解決當失去原始來源影像時，而無法得知原始來源影像，其色彩分佈的問題，更可節省儲存空間的容量，因為僅需要最後的結果影像便能夠獲得原始的來源影像，解決了色彩轉換演算法不可逆性的缺點。

本文其他章節的架構如下：第二節說明相關的研究，包括色彩轉換演算法以及資訊隱藏的概念與演算法；第三節詳細敘述我們提出的演算法，第四

章展示並且說明我們的試驗結果，第五章則總結本文並且進一步地提出未來的研究方向。

二、相關工作

2.1. 色彩轉換

近來色彩轉換技術可分為兩大類：1：彩色影像間的色彩轉換；2：黑白轉成彩色的色彩轉換。

關於彩色影像間的色彩轉換，文獻上有幾篇文章，分別為 [3, 4, 9, 10, 12, 15]。Reinhard 等人透過一個有較低相依性的色彩空間，透過統計的技巧，達成影像間的色彩轉換，她們的演算法快速且有效 [9, 10]。Chang 等人提出一新的演算法實行色彩轉換，她們使用 11 種基本的顏色類別作色彩的轉換，因此需要先對來源與目標影像做色彩分類，再做色彩的轉換，因此計算較為繁複，需較長的計算時間 [3, 4]。Zhang 的論文探討如何選取主要區域 (Principal Region)，來建構出一個至多二次的多項式貼覆函數 (Polynomial Mapping Function)，據此，在一群影像中，可以兼顧顏色的外觀，達成快速的影像校正 [15]，該技術主要的貢獻為色彩校正，因此，該論文不在我們討論的範疇內。Wang 基於 Reinhard 的演算法，考慮時間的變化，透過轉換曲線的變化，成功地為同一場景模擬於不同時間的景色變化 [12]。

黑白轉彩色的相關文獻主要以 Welsh [13] 等人於 2002 年提出之論文為主。Welsh 的目的是重新賦予黑白圖片新的生命，企圖將彩色影像之色彩導入黑白影像，使得結果影像具有黑白影像之特徵，亦具有彩色影像之色彩，Welsh 亦提出予許使用者手動定義影像區塊 (Swatch)，透過這些區塊進行色彩的匹配跟調整處理，來提升轉換的效果，儘管 Welsh 的方法獲致了不錯的成果，但是黑白轉彩色必須將原先一個的灰階頻道，擴展為三個的彩色頻道。由於不足的資訊需要一些複雜的手動額外處理，因此，Welsh 所提的演算法不在我們探討的範圍內。2004 年，Chen 以 Welsh 的技術為基礎，提出

一個效果更佳的技术為灰階影像上色[5]，不同於 Welsh 技术之處為，Chen 利用影像分割(Image Metting)的技术，將灰階影像中的物件分解，此處分解物件的意義相當於 Welsh 所使用的取樣區塊(Swatch)，被分解後的物件再分別予以指定所希望轉換的色彩，因此效果更優於 Welsh 提出的方法。

總結相關的文獻資料，被提出的技术均只探討如何變換色彩，無相關文獻探討如何將色彩還原的問題，有鑑於此，我們結合資訊隱藏的技术，期望能夠發展出一個具有還原影像能力的色彩轉換演算法。

2.2. 資訊隱藏

近年來，由於網際網路的蓬勃發展，使得人們得以輕鬆地藉由網路，傳遞各式各樣的資訊。無論聲音、影像、文字、指令等，凡是可數位化的資訊，皆可透過網路傳遍世界各地。其中有些資訊是珍貴的智慧財產，為了保護它們，許多人便在其中秘密藏入可辨識其智慧財產權的資訊，這些技术被統稱為資訊隱藏(Data Hiding)。

根據[1]對資訊隱藏研究的分類，可以將資訊隱藏的研究區分為二大領域，分別為偽裝學(Steganography)及數位浮水印(Digital Watermarking)。

偽裝學主要著重於資訊隱藏方面，其目的主要是保護被隱藏的資訊而非數位媒體本身，且通常希望藏入的資訊愈多愈好，而浮水印之目的則是保護數位媒體本身，如 3D 模型、2D 影像或是文件等，因此浮水印通常須具備一定程度之抵抗攻擊的能力，我們擬提出之可逆的色彩轉換演算法，則是屬於偽裝學領域。

若是以資訊嵌入數位媒體後之視覺效果來作分類，那麼上述兩個領域皆可再區分為可視(Visible)及不可視(Imperceptible)二類。可視浮水印通常是直接在數位媒體中嵌入浮水印，可明顯的由肉眼辨別出這些嵌入的浮水印。不可視浮水印則無法用肉眼察覺，必須使用特定的演算法來進行計算，方可取

出並驗證浮水印的存在及正確性，其中，不可視浮水印依照藏入的資訊是否具備抵抗破壞的能力，則可再區分為兩類，分別為強韌性(Robust)及易碎性(Fragile)技术，強韌性技术主要運用於浮水印相關的著作權保護，而易碎性常用於驗證保護之數位媒體是否經過竄改。

現今與資訊隱藏的相關技术可說是玲瑯滿目，根據 Katzenbeisser 等人對於浮水印技术的分類整理，大致可將這類技术分成六大類[7]：

1. 置換方式(Substitution Systems)
2. 轉換領域 (Transform Domain)
3. 展頻 (Spread Spectrum)
4. 統計法(Statistical Methods)
5. 失真技术(Distortion Techniques)
6. 產生遮蓋物件法(Cover Generation Methods)

本文引用 Ni 等人於 2004 年所提出的影像資訊隱藏演算法來建構出我們的演算法，Ni 的演算法屬於上述分類中的統計法，係利用原始影像之統計特質來嵌入資訊，我們採用此演算法主要考量有三點，1.適用於 JPEG (Joint Photographic Experts Group) 影像格式；2.效率高：此演算法不需繁雜的計算過程，3.影像被藏入資訊後，平均仍保有 39dB 以上之 PSNR(Peak Signal to Noise Ratio)值，且不會有明顯的雜訊出現[8]，但是，Ni 的技术在取回資訊的部份，儘管有較低的遺失率，不過，我們期望藏入的資訊必須能夠被完整地取出，因此我們需要對 Ni 的技术做改善，以期能夠百分之百地取出資訊。

三、演算法

本文擬提出一具有可逆能力的色彩轉換演算法，我們以 Reinhard 等人提出的色彩轉換演算法為基礎[9]，其演算法主要的概念為計算兩張欲相互轉換的影像在 $l\alpha\beta$ 色彩空間中的平均值與標準差，透過此二個變數使得來源影像上的色彩分佈類似於目標影像的分布，由此可以知道，若企圖將轉換後的

影像結果，使其還原成為原來的來源影像，我們必須保存影像在 $\alpha\beta$ 色彩空間中總共 12 個平均值與標準差(來源與目標影像的每一頻道均具有平均值與標準差)，因此我們必須將此 12 個重要參數隱藏於色彩轉換後的結果影像中。

由於必須將資訊隱藏於影像中，因此我們必須尋求一隱藏性較好的資訊隱藏技術，不過，因為我們企圖於影像上隱藏資訊，有兩點重要的事項必須要注意：

1. 由於我們的目的是企圖使色彩轉換後的結果影像能夠具有還原的能力，因此隱藏至影像上的資訊必需能夠被完整地取出，必須有抵擋攻擊的能力，例如影像的壓縮，如此才能使得還原後的影像近乎與原始的來源影像相同，換句話說，即還原後影像與原始來源影像間須具有較高的 PSNR 值(至少需 30dB 以上)。

2. 將資訊隱藏於影像上後，該影像效果必須使

我們的視覺系統無法察覺出有任何的異樣，例如產生雜訊或是產生不符合視覺經驗的結果。

對於現存的資訊隱藏技術，我們發現，Ni 於 2004 年提出一個效果不錯且低遺失率的影像資訊隱藏技術[8]，因此隱藏上述 12 個重要變數於影像上的部份，我們以 Ni 的技術為基礎，不過，如上述中的第一點，被隱藏的資訊必須能被完整取回，而 Ni 的方法儘管效果不錯，但是資訊取回的遺失率依然太高，為了解決此一問題，我們結合展頻技術與投票法於 Ni 的方法中，企圖提高被取回資訊的正確率。我們的演算法主要區分為五個步驟，其流程如圖 1 所示。此五個步驟分別為 1：色彩轉換； 2：資料編碼； 3：資訊隱藏； 4：資訊取出； 5：色彩還原。為了明顯區分步驟彼此間的先後順序，在圖中我們以五個灰色虛線區塊作為區分；接著，再以下小節中我們將詳細說明我們提出的演算法之五個步驟的方法及其概念。

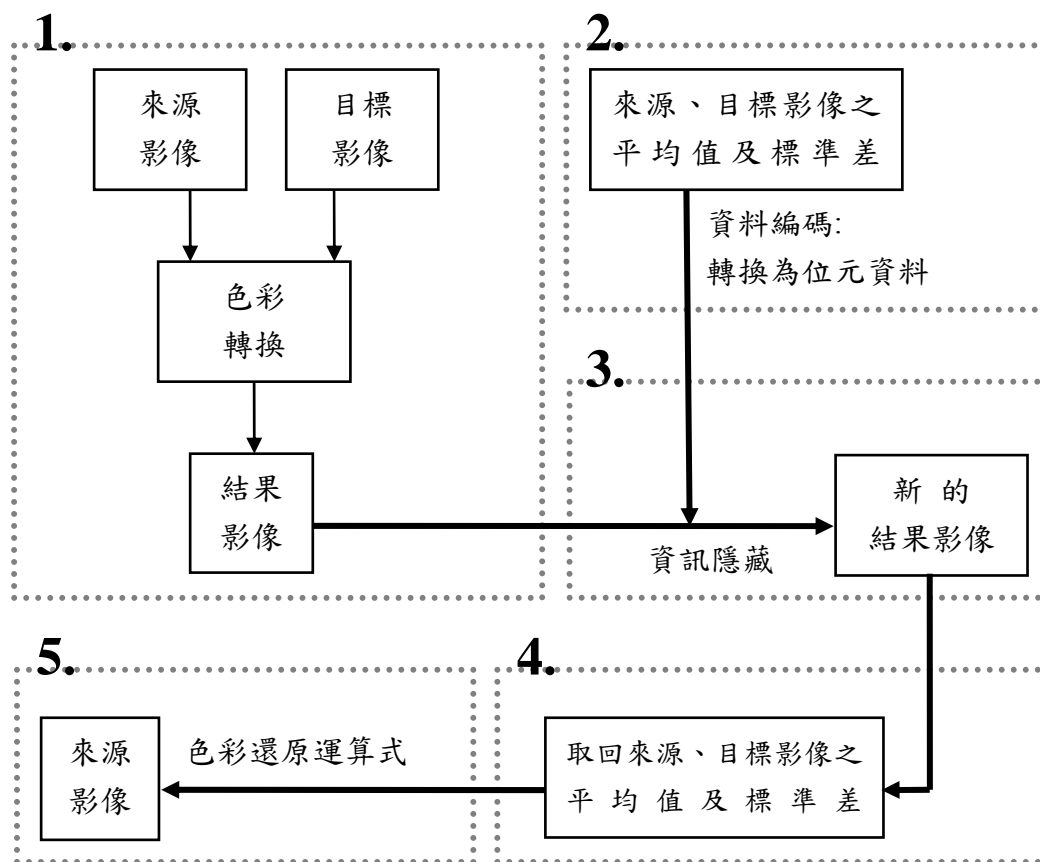


圖 1：具可逆能力的色彩轉換演算法流程圖。

3.1 色彩轉換演算法

Buchsbaum 等人於 1983 年提出一份關於色彩轉換的研究報告，他們發現可以利用一些線性的方法將人類視覺系統所能接受的三刺激色，紅、綠、藍，轉換成 3 個互無關聯性的色彩頻道 (color channel)[6]，隨後，1998 年，Ruderman 以 Buchsbaum 的研究為基礎，發展出了一個新的色彩空間， $l\alpha\beta$ 色彩空間[11]，此色彩空間的 3 個色彩頻道之間具有相當低的關聯性，圖 2 為影像隨機選取 1000 個像素點後轉換到 $l\alpha\beta$ 色彩空間後所繪得的關係分佈圖，由左至右分別是： $l-\alpha$ 、 $l-\beta$ 、 $\alpha-\beta$ 的色彩關係分佈圖，其中 l 為無色差頻道 (Achromatic Channel)， α 與 β 則互為對立色頻道 (Opponent Channels)，分別為黃-藍色頻道 (Yellow-Blue Channel) 與紅-綠色頻道 (Red-Green Channel)，圖 3 為影像轉換至 $l\alpha\beta$ 色彩空間中，3 個頻道的顯像圖。

Reinhard 根據 $l\alpha\beta$ 色彩空間的特性，於 2001 年提出了以統計學的技巧來達成將目標影像的色彩特徵轉換至來源影像上，她們稱此技巧為：影像間的色彩特徵轉換演算法[9]，演算法的實作僅有幾個簡單的步驟，首先，透過 LMS 色彩空間將來源影像與參考影像從 RGB 色彩空間轉換至 $l\alpha\beta$ 色彩空間，演算式(1)為 LMS 色彩空間的轉換矩陣：

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.3811 & 0.5783 & 0.0402 \\ 0.1967 & 0.7244 & 0.0782 \\ 0.0241 & 0.1288 & 0.8444 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

然後利用演算式(2)將 LMS 色彩空間轉換至 $l\alpha\beta$ 色彩空間，演算式如下：

$$\begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} \log_{10} L \\ \log_{10} M \\ \log_{10} S \end{bmatrix} \quad (2)$$

當影像均被轉換至 $l\alpha\beta$ 色彩空間中後，再利用演算式(3)來轉換兩張影像間色彩特徵。此處符號 $l, \alpha,$

β 的意義為色彩空間的三個頻道， σ 表示影像的標準差， m 表示影像的平均值，上標符號 l, α, β 表示需分別對 l, α, β 頻道做處理，下標 $FinalResult$ 定義為轉換後的結果， $source$ 為來源影像， $target$ 為欲轉換的參考影像，最後再將所得到的結果從 $l\alpha\beta$ 色彩空間轉換回至 RGB 色彩空間。

$$\begin{aligned} l_{FinalResult} &= \frac{\sigma_{target}^l}{\sigma_{source}^l} \times (l_{source} - m_{source}^l) + m_{target}^l \\ \alpha_{FinalResult} &= \frac{\sigma_{target}^\alpha}{\sigma_{source}^\alpha} \times (\alpha_{source} - m_{source}^\alpha) + m_{target}^\alpha \quad (3) \\ \beta_{FinalResult} &= \frac{\sigma_{target}^\beta}{\sigma_{source}^\beta} \times (\beta_{source} - m_{source}^\beta) + m_{target}^\beta \end{aligned}$$

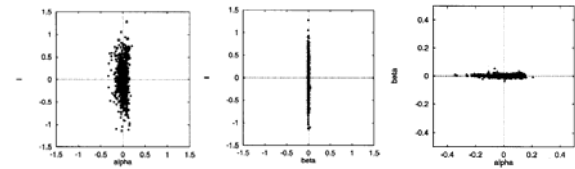


圖 2：影像中隨機選取 1000 個像素經由轉換後 $l-\alpha$ 、 $l-\beta$ 、 $\alpha-\beta$ 色彩關係分佈圖，由圖中可很清楚地得知此 3 個色彩頻道間僅具些微的關聯性 [11]。

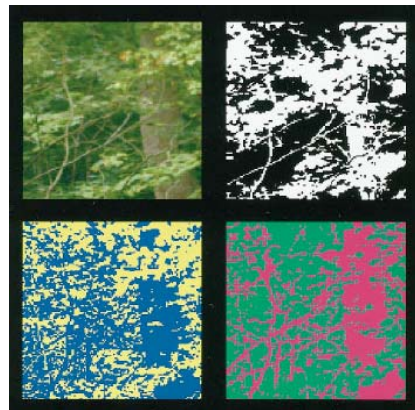


圖 3：左上圖為原始的測試影像 (RGB)，右上圖為無色差頻道 (Achromatic Channel)， l ；左下圖為黃-藍色頻道 (Yellow-Blue Channel)， α ；右下圖為紅-綠色頻道 (Red-Green Channel)， β [11]。

3.2 資訊編碼

資訊在藏入影像前，必須事先經過編碼的處理。而我們所隱藏至結果影像上的資訊為目標影像和來源影像， $\alpha\beta$ 三頻道各別的平均值和標準差，總共有 12 個數值。我們以 l 頻道的平均值為例作說明，這 12 個數值的編碼處理均以此為核心。

根據 $\alpha\beta$ 色彩空間與 RGB 之間的特性，由 RGB 轉到 $\alpha\beta$ 的數值，範圍將大致落於 -5 與 5 之間。假設轉換的值为 x ，因此我們估計 x 以 39 位元來進行編碼儲存，格式如圖 4 所示，其中，第 1 個位元表示正負符號，第 2 到 4 個位元表示個位數字的值，第 5 到第 39 個位元則為小數部份，小數部份我們以 35 個位元來進行編碼處理，主要目的就是為了保持小數點以下 10 到 11 位的精確度，以減少運算過程中所累積產生的誤差。由於我們須隱藏的資料共有 12 個數值，每個值皆以 39 個位元編碼，總計藏入的位元總共有 468 個位元 (12×39)，編排的格式如圖 5 所示。

3.3 資訊隱藏

本節是演算法過程中相當重要的一個步驟，主要說明我們如何將資訊藏入至結果影像中，要是處理的不妥善，將導致資訊無法完全的取出而造成錯

誤。因此我們結合了展頻技術以及投票法，利用一般影像中鄰近像素之間的相依性及重覆性所形成特性，在 JPEG 影像格式中藏入位元值，本節裡，我們將分為二部份來說明，第一是演算法之核心理論，第二是位元嵌入策略。

3.3.1 核心理論

首先我們導入 Ni 論文中資訊隱藏的概念，先將影像切割成數個互不重疊、互不相依，大小為 8×8 的影像區塊。並於此區塊內，以交叉間隔的方式，再分割成兩種(以 A 和 B 來表示)不重疊的子區塊，共 64 塊。其中區塊 A 以 '+' 來標示，區塊 B 以 '-' 標示，其表示方式如圖 5 所示。此外需注意一點，我們事先刪除極端值(如 RGB 值為 0 或是 255 的像素)的區塊，這麼一來可以預防程式執行中所產生的錯誤，例如嵌入資訊後造成 RGB 值發生溢位/下溢(over/underflow)等問題。而每個 8×8 區塊，我們計算區塊中每一對像素(即每一對 '+'，共 32 對)之差異值，接著算出此區塊中所有差異值的平均，以 α 表示。如演算式(4)所示，其中 n 值為 32。

$$\alpha = \frac{1}{n} \sum_{i=1}^n (a_i - b_i) \quad (4)$$

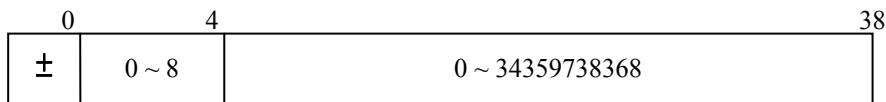


圖 4：資料位元的編碼方式。

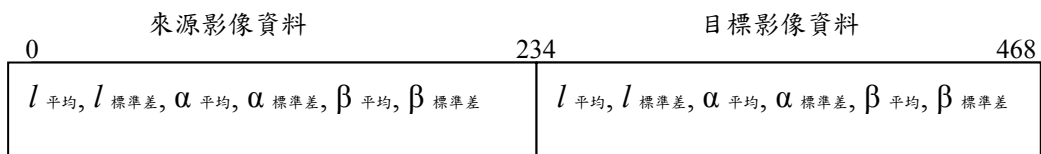


圖 5：資料編排格式。

一般我們常見的影像中鄰近相素通常具有相當高的相依性及重覆性，因此區塊中所計算出來的 α 值大多趨近於 0，其分佈如圖 6 所示。

+	-	+	-	+	-	+	-
-	+	-	+	-	+	-	+
+	-	+	-	+	-	+	-
-	+	-	+	-	+	-	+
+	-	+	-	+	-	+	-
-	+	-	+	-	+	-	+
+	-	+	-	+	-	+	-
-	+	-	+	-	+	-	+

圖 5：8x8 區塊，分為二個子區塊的交叉方式。

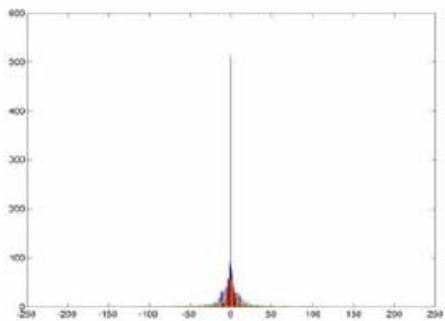


圖 6：影像中 α 值的分佈情形。

3.3.2 位元藏入策略

接下來討論 α 值位於不同的範圍內，要用何種方式來處理：

狀況 1：當 α 值位於預設的門檻值內(本文使用小於 5 的數值作為門檻值)，此時若要藏入的位元值為 1 時，我們便將 α 值往左或是往右移動，使其越過門檻值，如圖 7 所示。移動 α 值的方式是，藉此來調整兩個子區塊的差異值。若藏入的位元為 0，則不須對此區塊進行變動。

狀況 2：同理，當 α 值位於事先設定的門檻值外，此時若要藏入的位元值為 0 時，我們便將 α 值往左或往右移動，使其 α 值縮小

至門檻值之內，調整方式與狀況 1 相同，同樣針對兩個子區塊中的每一個像素，加或減去一個固定值。但若藏入的資訊為 1，則不進行變動。

在狀況 2 時，我們的作法與 Ni 作法略有不同，原始作法是當 α 值落在門檻值之外時，不管原本欲藏入的位元值為 0 還是 1，皆以 1 作為嵌入位元，而資訊取出後，再藉由錯誤訂正碼(Error Correction Code, ECC)進行修正。因為我們的目的與 Ni 不盡相同，Ni 的目標是希望藏入更多的資訊，而我們則是每次藏入的資訊量皆一致(即 468 bits)，因此我們針對此部份略作修改，期望能達到我們所預期的成果。

由於我們期望所提出的演方法能夠適用於使用較為普遍的 JPEG 格式上，因此我們必須考慮到 JPEG 格式的破壞性壓縮問題，此一問題可能致使取出後的資訊為了進行錯誤檢查，而必須藏入更多的錯誤檢查碼，來進行錯誤偵測，於此，我們決定以展頻技術搭配投票法來取代狀況 2 的部份。所謂的展頻技術，即使用者想隱藏資訊，然而將同樣的資訊藏許多份。而投票法是當在進行資訊取出的工作時，若某個 bit 在取出時，因為傳送或壓縮時失真的關係，造成每份的此 bit 不完全相同，導致資訊取出的結果可能造成錯誤，因此我們就把出現最多次數的 bit 值當作是正確的，藉由這兩樣技術的搭配我們即可以完善的處理狀況 2 的情形。如以下這個例子：

要藏入的原始資訊：00010110
 複製三份藏入： 00010110(第一份)
 00010110(第二份)
 00010110(第三份)
 傳送或壓縮時失真，使得某些 bit 錯誤
 01010110(第一份)
 00100110(第二份)
 00010010(第三份)
 利用投票法決定： 00010110
 →得到正確結果。

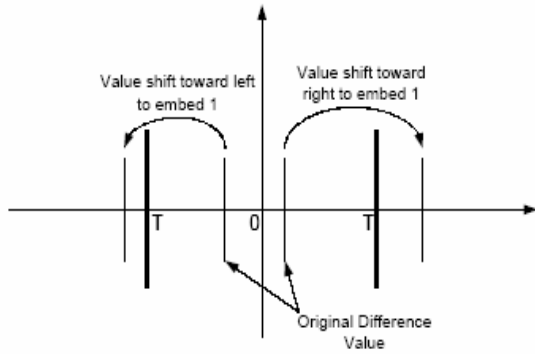


圖 7：位元嵌入策略。

3.4 資訊取出

資訊藏入之後，最關鍵的部份為須絲毫無誤的把資訊地成功截取出來，僅能容許微小的差異性；位元取出的步驟和藏入順序相反。當我們得到一張已存入資訊的影像時，首先將影像切割成不重疊的 8x8 區塊，如上圖 5，並計算出每個區塊的 α 值，並且由算出的 α 值可以得知此區塊嵌入之位元為何，再加上我們搭配了展頻技術及投票法，因此一個位元值通常須由 3 到 5 個區塊來決定最終值為 0 或 1，如此便能夠大大地提升資訊的正確率。

當我們取回正確的資訊後，再利用演算式(5)將藏有資訊的結果影像還原成原始的來源影像。

3.5 色彩還原

於經由色彩轉換後的結果影像上取出資訊後，緊接著便是對該影像做還原色彩的處理；由於我們是以 Reinhard 提出的色彩轉換演算法為基礎，因此在色彩還原的部份，我們僅需對 Reinhard 演算法中負責處理色彩轉換的部份做反運算的處理即可獲得原始的來源影像，演算式如式子(5)，注意，等式的左邊即為我們所欲得到的一還原後的來源影像。

$$\begin{aligned}
 l_{source} &= \frac{\sigma_{source}^l}{\sigma_{target}^l} \times (l_{FinalResult} - m_{target}^l) + m_{source}^l \\
 \alpha_{source} &= \frac{\sigma_{source}^\alpha}{\sigma_{target}^\alpha} \times (\alpha_{FinalResult} - m_{target}^\alpha) + m_{source}^\alpha \\
 \beta_{source} &= \frac{\sigma_{source}^\beta}{\sigma_{target}^\beta} \times (\beta_{FinalResult} - m_{target}^\beta) + m_{source}^\beta
 \end{aligned}
 \tag{5}$$

四、實驗結果分析

本節我們分為兩個部份來說明我們所提出的演算法，首先，比較與展示 Zhicheng 與經由我們改進過後的驗算法結果，以茲證明我們提出的方法其可行性；緊接著，展示色彩轉換還原後的結果，並且輔以 PSNR 數量化反轉後的影像品質。我們所使用的平台為 Intel Pentium4 2.0 GHz 處理器，記憶體為 768 MB，所有程式以 PHP 5.04 搭配 GD Library 並於 OpenBSD 3.7 系統下開發及執行。

由於原 PSNR 公式僅定義於灰階影像，本論文實驗皆為 RGB 之彩色影像，因此將原 PSNR 公式稍作修改，如式(6)：

$$\begin{aligned}
 MSE &= \left(\frac{1}{H \times W \times 3} \right) \sum_{y=1}^H \sum_{x=1}^W \sum_{c=1}^3 (v_{ysc} - v_{ysc}')^2 \\
 PSNR &= 10 \log_{10} \left(\frac{255^2}{MSE} \right)
 \end{aligned}
 \tag{6}$$

4.1 資訊隱藏

實驗中，我們以 8 張解析度為 1600x1200 的影像進行測試。首先，在第一組的實驗裡，我們直接以 Ni 的演算法嵌入 468 個位元，並將結果配合 PSNR 值，列於表格 8。由表格 1 可以明顯的觀察出直接使用此演算法藏入資訊後的影像，仍然保有相當高的 PSNR 值，但是由於影像經過 JPEG 格式壓縮[2]，使得取出後的位元資訊若沒有經過錯誤檢查碼或投票法的修正，錯誤率可能高達 34%，也就是在 468 個位元中，有高達 162 個位元發生錯

誤，如表格 1 中的 Sample4。

我們將表格 1 的實驗結果影像展示於圖 8 及 9，其中圖 8-(a)為使用的原始影像，圖 8-(b)為藏入資訊後的結果影像，為了視覺化原始影像與藏入資訊後的結果影像之差別，我們將兩張影像相減，結果如圖 8-(c)，但由於差異並不明顯，因此我們將差異放大 10 倍展示於圖 8-(d)。圖 9 則為 Sample2~8 之實驗結果，同樣也展示了兩張影像差異值放大 10 倍後的結果。

觀察圖 8 及圖 9，被嵌入資訊後的影像不會產生明顯的雜點，而且仍然保持著 40dB 以上的 PSNR 值，視覺上幾乎感受不出影像被藏入資訊，但由於 JPEG 在成圖時須經過壓縮，使得取回的資訊時不全正確，造成日後無法正確還原影像，因此我們為 Zhicheng 的演算法加入展頻技術與投票法，企圖提高資訊提取的正確性；緊接著展示經由我們修正過後的結果表格 2 列出 Sample1~8 展頻的倍數及 PSNR 值。

圖 10 則為實驗結果影像，在我們運用展頻技術配合投票法，展頻三倍或五倍之後，即可使得取回的資訊正確率接近於 100%，數值上僅有些微的誤差，在圖 10 的結果中，我們更發現即使經過展頻，藏入資訊後的結果影像也不會有明顯的雜訊產生，並且仍然保持著相當高的 PSNR 值，視覺上亦是幾乎看不出影像被藏入資訊。經由上述實驗，證明我們的方法可以確保資訊隱藏後取出資料之正確性，並且保有相當高的 PSNR 值。

4.2. 色彩還原

在此部分，展示三組色彩轉換還原之結果，我們先以兩組 Reinhard[9]論文的日出模型及城堡模型進行實驗，再以先前的 Sample4 進行實驗。

圖 11 為 Reinhard 論文的日出模型，以我們的演算法進行色彩轉換後並藏入資訊的結果影像為圖 11-(c)，接著把結果影像進行還原，還原之後的結果為圖 11-(d)，觀察還原後的影像，我們提出的方法確實成功的還原其色彩，使其色彩分佈與來

源影像相同，並具有 33.4dB 的 PSNR 值。圖 12 與 13 為另兩組試驗模型，排列方式與日出模型相同，此兩個模型還原後仍具有 45.8dB 與 39.15dB 的 PSNR 值。

根據實驗結果證實，我們提出的演算法，可以成功的還原結果影像之色彩，回復成原來的來源影像，使其色彩分佈與來源影像相同，於還原後依然保有較高的影像品質。

五、結論與未來工作

本文提出一具可逆能力的色彩轉換演算法，首創讓色彩轉換後之結果影像具備了色彩還原的能力。我們運用色彩轉換演算法配合資訊隱藏的技術，將來源影像及目標影像在 $\alpha\beta$ 色彩空間的平均值與標準差嵌入結果影像中，經由實際測試與分析顯示，具可逆能力的色彩轉換演算法確實不僅成功的使得色彩轉換後的結果影像還原成原始影像，具有相當不錯之成效，其 PSNR 值均維持在 30dB 以上。

我們提出的演算法至少具三項優點。第一、記憶體空間的節省：我們利用具可逆能力的色彩轉換演算法，使得藏有變數的結果影像，可以藉由色彩還原運算式而得到原始的來源影像，如此僅需要儲存結果影像，而無須再儲存原始來源影像的必要，使得原本需要兩張影像的空間，只需存放一張結果影像即可，節省了記憶體空間。第二、更多次的色彩轉換：將結果影像經過色彩還原，除了獲得的原始來源影像，也能將隱藏其中的目標影像平均值與標準差取得，用來做下次色彩轉換的參考基準，以搭配多張不同的目標影像進行色彩轉換。第三、快速的還原時間：在演算法中，經過色彩還原運算式以獲取原始來源影像的時間在 5 秒以內都可完成，演算法適合被使用於行動裝置上。

儘管演算法還原的效果良好，能夠使得人類視覺系統無法察覺到其中的差異性，但是我們認為尚有的進步空間。未來的研究目標為發展視覺效果更優異的還原演算法。

表 1：不使用錯誤檢查碼的情型下，取出資訊的錯誤率可能高達 34.6%

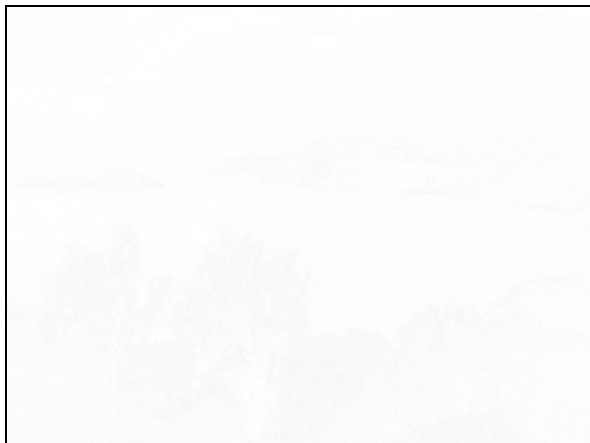
影像 (1600x1200)	可藏 bit 的 block 數	嵌入資訊之位元 總數	嵌入資訊後之 PSNR (dB)值	取出後錯誤 之位元數	錯誤率
Sample1	11,805	468	54.14	154	32.9%
Sample2	21,339	468	86.62	56	11.9%
Sample3	10,408	468	48.05	18	3.8%
Sample4	11,001	468	42.17	162	34.6%
Sample5	26,801	468	99.52	2	0.4%
Sample6	11,429	468	50.98	12	0.2%
Sample7	12,994	468	74.43	30	6.4%
Sample8	9,864	468	58.71	52	11.1%



(a) 原始影像



(b) 藏入資訊後影像



(c) 差異影像，即 a-b



(d) 將差異放大 10 倍後結果

圖 8：資訊隱藏於 JPEG 影像，實驗結果(Sample 1)。

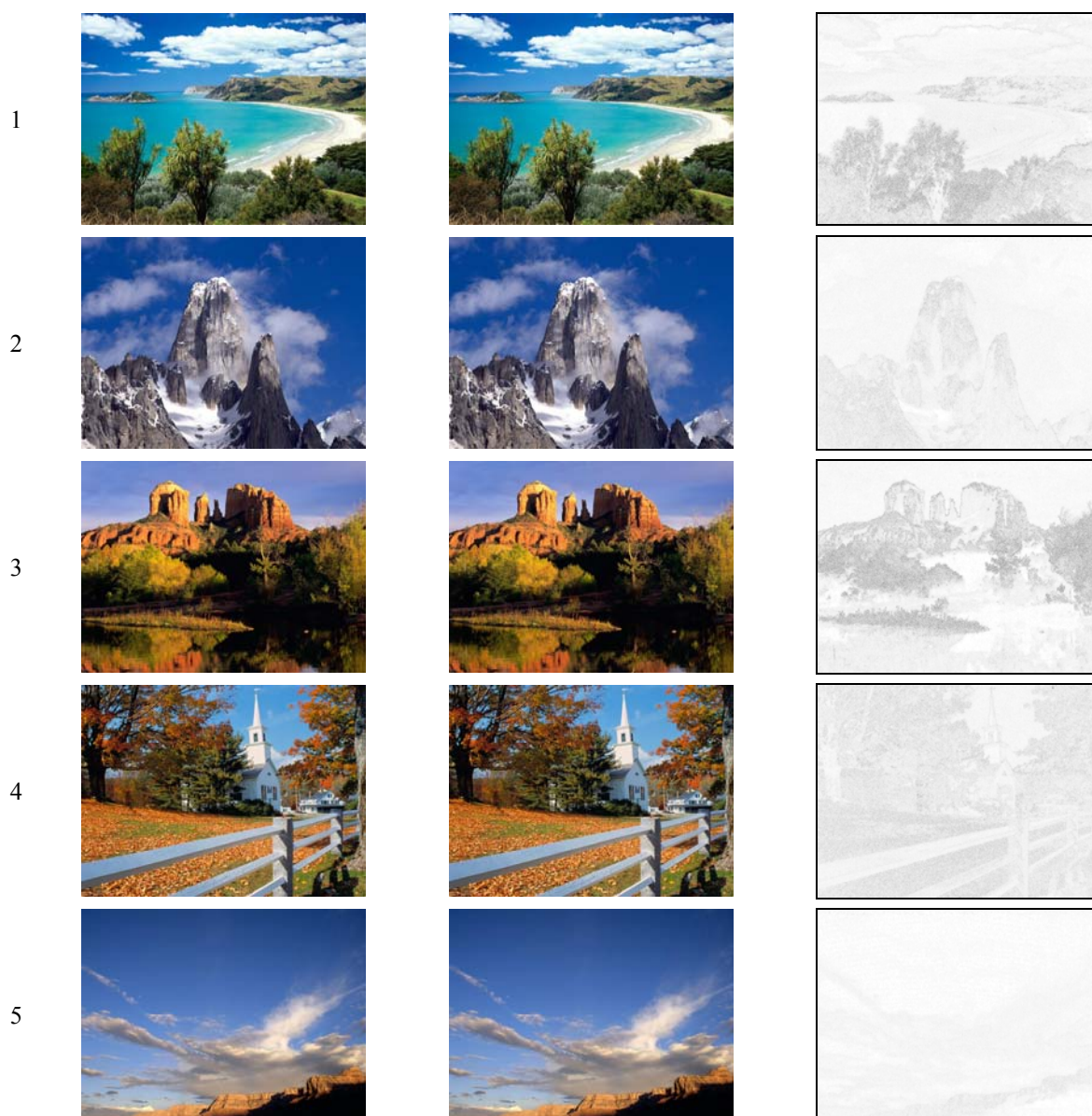


(a) 原始影像 (b) 藏入資訊後影像 (c) 差異影像，即(a-b)*10

圖 9：資訊隱藏於 JPEG 影像，實驗結果(Sample 2~8)。

表 2：達到取出位元的正確率為 100% 時，所需展頻的倍數及 PSNR 值。

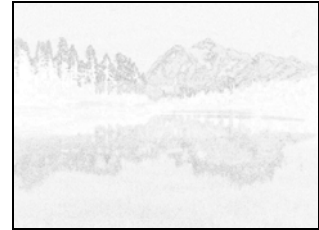
影像 (1600x1200)	嵌入資訊之位元總數	嵌入資訊後之 PSNR (dB)值	展頻 倍數
Sample1	468	49.52	5
Sample2	468	83.74	3
Sample3	468	47.33	3
Sample4	468	39.15	5
Sample5	468	90.08	3
Sample6	468	49.14	3
Sample7	468	72.64	3
Sample8	468	56.22	3



6



7



8

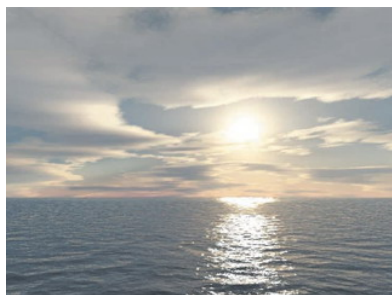


(a) 原始影像

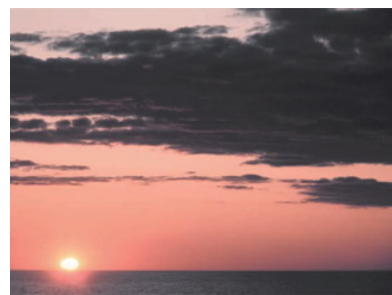
(b) 藏入資訊後影像

(c) 差異影像，即 a-b

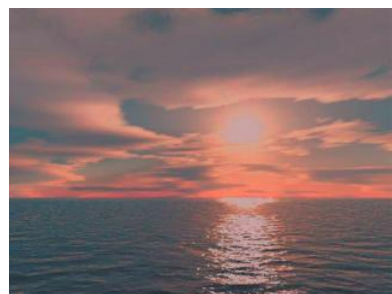
圖 10：展頻後，取出資訊正確率為 100%，所呈現之結果。



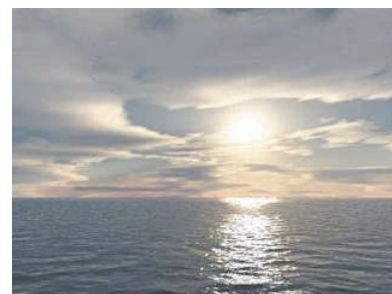
(a) 來源影像



(b) 目標影像



(c) 被藏入資訊後的色彩轉換結果

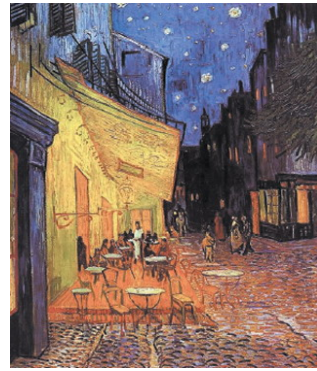


(d) 經由提取資訊後還原的結果

圖 11：日出模型還原結果，還原後 PSNR=33.4 dB。



(a) 來源影像



(b) 目標影像



(c) 被藏入資訊後的色彩轉換結果



(d) 經由提取資訊後還原的結果

圖 12：城堡模型，還原後 PSNR=45.8 dB。



(a) 來源影像



(b) 目標影像



(c) 被藏入資訊後的色彩轉換結果



(d) 經由提取資訊後還原的結果

圖 13：Sample4 模型還原結果，還原後 PSNR=39.15dB。

六、誌謝

本研究承蒙國科會專題研究計畫之經費補助 NSC 94-2213-E-005-022、NSC 94-2218-E-164-001、NSC 94-2815-C-005-026-E、NSC 93-2213-E-005-018、NSC 92-2213-E-005-021，謹此向國科會致謝。

七、參考文獻

- [1] 吳大鈞，“在影像中隱藏資料的研究”，國立交通大學博士論文，民國 89 年。
- [2] 吳炳飛、胡益強、瞿忠正、蘇崇彥，JPEG2000 影像壓縮技術，台北市，全華圖書，民國 92 年。
- [3] Y. Chang, S. Saito, and M. Nakajima, “A Framework for Transfer Colors Based on the Basic Color Categories,” in Proceedings of Computer Graphics International, pp. 176-181, 2003.
- [4] Y. Chang, S. Saito, K. Uchikawa, M. Nakajima, “Example-Based Color Stylization Based on Categorical Perception,” in Proceedings of the 1st Symposium on Applied perception in graphics and visualization, Los Angeles, California August 07-08, pp. 91-98, 2004.
- [5] T. Chen, Y. Wang, V. Schillings, and Christoph Meinel. “Grayscale Image Matting and Colorization,” in Proceedings of Asian Conference on Computer Vision, 2004.
- [6] G. Buchsbaum and A. Gottschalk, “Trichromacy, Opponent Colour Coding and Optimum Colour Information Transmission in the Retina,” in Proceedings of Royal Society, London, Ser. B 220, pp. 89-113, 1983.
- [7] S. Katzenbeisser, F. A.P. Petitcolas, “Information Hiding Techniques for Steganography and Digital Watermarking,” Artech House Book Company, 2000.
- [8] Z. Ni, Y. Q. Shi, N. Ansari, W. Su, Q. Sun, and X. Lin, “Robust Lossless Image Data Hiding,” in Proceedings of the 2004 IEEE International Conference on Multimedia and Expo, pp. 2199-2202, 2004.
- [9] 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, 2001.
- [10] E. Reinhard, A. O. Akuyuz, M. Colbert, C. E. Hughes, and M. O'Connor, “Real-time Color Blending of Rendered and Captured Video”, Interservice / Industry Training, Simulation and Education Conference, 2004.
- [11] D. J. 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.
- [12] C. M. Wang and Y. H. Huang, “A Novel Color Transfer Algorithm for Image Sequences,” Journal of Information Science and Engineering, Vol. 20, No. 6, pp. 1039-1056, 2004.
- [13] T. Welsh, M. Ashikhmin, and K. Mueller, “Transferring Color to Greyscale Images,” in Proceedings of SIGGRAPH 2002, pp. 277-280, 2002.
- [14] G. Wyszecki and W. S. Stiles, Color Science: Concepts and Methods, Quantitative Data and Formulae, second edition, John Wiley and Sons, New York, 1982.
- [15] M. Zhang and N. D. Georganas, “Fast Color Correction Using Principal Regions Mapping in Different Color Spaces,” Real-Time Imaging, Vol. 10, pp. 23-30, 2004.