

# 新的高動態範圍影像資訊隱藏演算法

## A Novel Data Hiding Algorithm for High Dynamic Range Image

王宗銘 鄭友銘 曾彥博 甘惠文 黃永煌 呂培瑛 謝育書

國立中興大學資訊科學系

E-mail: cmwang, s9156048@cs.nchu.edu.tw;

s9156001, s9156020, s9156024, s9156034, s9156035@mail.cs.nchu.edu.tw

摘要

Keywords: high dynamic range image, data hiding, block area, edge area

近年來數位影像上的重要發展在於如何利用數位化的資料真實呈現自然世界的真實色彩，高動態範圍影像即為此一研究中重要的趨勢。資訊隱藏有別於密碼學中對資訊內容的保護，其特性為隱藏資訊本身的存在，使人難以察覺該媒體之中藏有資訊，對於藉由網路傳輸的資訊形成了一種保護色。就我們所知，將資訊隱藏技術應用在高動態範圍影像上，至今仍無相關的論著。本文利用高動態範圍影像與生俱來的差異性對影像進行分區，劃分為區塊與邊界兩個部分，再透過我們提出的新的高動態範圍影像資訊隱藏演算法，將資訊嵌入，產生不易為外人所察覺的視覺效果，達到高資訊嵌入量及低失真之目的。實驗結果顯示：我們的技術首開先例，成功地在高動態範圍影像上提出資訊隱藏演算法，達到肉眼不易察覺與原始影像差異之效果。

**關鍵詞：**高動態範圍影像、資訊隱藏、區塊、邊界。

### Abstract

In recent years, an important development on digital images concentrated on how to present real world color by the information of digitization. High dynamic range (HDR) image is an important trend in this studying. Data hiding, whose characteristic is to hide the existence of information itself, is different from the encryption in cryptography. Data hiding makes people hard to perceive there's something hidden in the information, and produce a kind of protective coloration with information that we transmit through the Internet. To the best of our knowledge, no literatures have ever addressed how to employ a HDR image in the technology of data hiding. In this paper, we utilize the HDR images innate difference between pixels to classify all pixels into two parts of block area and edge area. Then, according to the new algorithm that we proposed, we can embed information in HDR images, which are hard to perceive. We provide a larger embedding capacity and to minimize the distortion of the stego image. Experimental results demonstrate that our algorithm is feasible to produce a data hiding HDR image with difficult to perceive differences between original image and it.

### 1. 簡介

在許多應用領域，資訊隱藏技術已越來越受到重視。軍用通訊系統不斷拓展信量安全技術的使用，不僅僅是使用加密技術來加密消息的內容，還力圖隱藏消息的發送者、接收者，甚至是消息本身的存在。人們往往誤以為，只要針對通訊的內容加密即可保證通訊的安全，事實上，一份加密後的資訊在傳遞時，通常更易引起他人的不法竊取、破壞與解密，因為此文件顯然包含某種重要資訊。是故，通訊安全的保障不能僅依靠加密，還須以隱藏資訊本體為基本考量。有鑒於此，「資訊隱藏」技術因應而生[3, 4, 5, 8, 13]，其優點為不易引起他人注意，藉此降低資料遭竊的風險，即使檔案遭竊，隱藏在內的資訊亦無法輕易取得。

目前為止，已有許多學者投入此領域研究，但是，卻僅侷限於傳統的低動態範圍影像，如點陣圖。有鑒於高動態範圍影像之趨勢，我們首創高動態範圍影像的資訊隱藏技術。

近年來廣為討論及使用的影像格式：高動態範圍影像，相較於低動態範圍影像，包含更高範圍的亮度，所含的影像資訊更為豐富。低動態範圍影像每個像素為 24 位元，以 RGB 模式而言，R、G 及 B 各佔 8 位元，即其顏色數值範圍介於 0~255，代表亮度僅有 256 個層級，顯然無法紀錄下自然世界的色彩；然而，高動態範圍影像像素範圍可達  $0\sim 2^{127}$ ，其數值遠遠超過 256，有效呈現自然世界的色彩。基於此優勢，高動態範圍影像勢必為未來之趨勢。然而，由於高動態範圍影像與生俱來與低動態範圍影像的差異性，目前已發表於低動態範圍影像之資訊隱藏演算法皆不適宜直接應用於高動態範圍影像，此外，至今仍無相關文獻對此著墨。有鑒於此，我們首創於高動態範圍影像上呈現資訊隱藏技術，展現出更不易令人察覺的影像效果。目前高動態範圍影像可分成許多不同的格式，如：光輝(Radiance) RGBE, PSD/PSB, LogLUV TIFF, PFM 及 OpenEXR 等，本文將針對最常見之光輝 RGBE 格式的高動態範圍影像提出資訊隱藏演算法。

本文首創在高動態範圍影像上提出資訊隱藏演算法。利用高動態範圍影像亮度分佈特性，亦即

RGBE 格式中的指數值 E，將整張影像切割成區塊(Block Area)與邊界(Edge Area)，其中，與相鄰像素之 E 值相同之像素稱為區塊，而與相鄰像素之 E 值不同之像素則稱為邊界。接著，先對區塊做雙邊界(Two-Sided)之區塊相配(Side Match)演算法，分別對區塊的每個像素根據其與鄰近像素之差異程度嵌入適當長度的資料。而邊界則藉其與相鄰像素之 E 值差異，轉換相鄰像素之 RGB 數值，使其落於同一階層之 E 值，再以雙邊界之區塊相配演算法處理。最後，即可取得一張內含資訊且視覺上不易令人察覺有異的高動態範圍影像。

本文結構如下：第二節說明相關工作；第三節敘述我們所提出的技術；第四節說明實驗結果。最後，第五節提出結論與未來工作。

## 2. 相關工作

鑒於目前仍無高動態範圍影像之資訊隱藏演算法，此節我們將以探討傳統低動態範圍影像之資訊隱藏演算法及高動態範圍影像之特性為主。

資訊隱藏是將欲秘密傳送的資訊(Message)隱藏於一張掩護影像(Cover-Image)而產生一張偽裝影像(Stego-Image)的過程，並利用密鑰使得檢測或取出過程僅限於密鑰的擁有者。主要目的是在雙方之間建立一個隱蔽的通訊，令潛在的攻擊者不知道這個通訊的存在，使資訊安全更有保障[1, 11]。

關於資訊隱藏的技術，已有許多學者進行這方面的研究，其中最為人所熟知的技術是：最低有效位元取代法(Least Significant Bits Substitution) [3, 8, 13]，以欲嵌入的資訊直接取代固定長度的最低有效位元。此技術的最大特色是：簡單且快速。

但是，最低有效位元取代法存在某些缺失，在掩護影像中，並非每個像素均可容許改變相同長度的量而不被看出差異，尤其在影像中的平滑區域，過多的隱藏資訊將導致明顯易見的瑕疵。針對此問題，張真誠等學者利用區塊相配演算法嵌入資訊，藉由參考鄰近的像素值決定每個像素最適合的隱藏量[4, 10, 15]。影像中平滑區域像素值的改變量縮小，減少隱藏量；反之，邊界區域允許較多的隱藏量。以此為基礎，張真誠等學者在影像壓縮的過程，利用改良的區塊相配演算法，提出可逆的演算法，使接收者從偽裝影像取出資訊後，亦能夠重建出原始的掩護影像，供後續之利用[5]。

迄今，雖已有許多學者進行這方面的研究，但均侷限於低動態範圍影像、影音檔案與三維模型等。高動態範圍影像包含豐富的資訊，近年來廣受討論及使用，但至今仍未有學者將資訊隱藏技術應用於此。鑒於資訊隱藏之重要性及高動態範圍影像之視覺效果，發展高動態範圍影像之資訊隱藏技術勢必為未來之趨勢。因其表現方式與影像格式有別於低動態範圍影像，故目前資訊隱藏的技術皆不適用。因此，本文提出新的高動態範圍影像資訊隱藏演算法，達到高資訊嵌入量及低失真之目的。

針對高動態範圍影像，Greg Ward 等學者提出光輝 RGBE 格式，降低像素的儲存空間[14]。將原本每個像素佔 96 位元的浮點數形式(r、g、b 各 32 位元)，以指數與尾數的替換方式，使其降低為僅佔 32 位元之整數格式(R、G、B 各 8 位元及共用 8 位元的指數值 E)；透過反運算，可將 32 位元的資料再度還原回 96 位元的資訊，可呈現的像素值範圍介於  $0 \sim 2^{127}$ 。雖然此舉會造成些微失真，但對高動態範圍影像的高像素值範圍而言，這樣的失真是可容許的，是故此壓縮方式成為高動態範圍影像採用的主流檔案格式之一。因此，我們針對此整數型態的 RGBE 之特性提出適用之資訊隱藏演算法。

高動態範圍影像與低動態範圍影像相較，有著許多的優點，是故許多學者紛紛投入此研究領域。Debevec 等學者提出一個以傳統攝影設備來製造高動態範圍影像的方法[8]，其利用影像為基底的塑模(Image-Based Modeling)、影像合成(Image Compositing)等技術，將多張不同曝光程度的影像合成一張高動態範圍影像。

基於高動態範圍影像的特性，若需要將其顯示在 24 位元的顯示器上，必須將其作色調映射(Tone Mapping)處理。Ashikhmin 學者提出一個高動態範圍影像的色調映射演算法，其方法是沿用人類視覺系統(Human Visual System)的分析，將其亮度的對數-對數空間(Log-Log Space)上的曲線簡化為四個線性區間處理，藉此提升處理效率[2]。

Fattal 等學者提出了一個高效能、且健全的方法來壓縮高動態範圍影像，使其能正常的顯示於傳統的顯示器上[7]。Reinhard 等學者使用延伸的技術處理數位影像，其論文指出傳統的攝影即是將真實世界亮度的高動態範圍影像對應到低動態範圍影像上。因此，其使用攝影的習慣來進行演算，避免使用知覺模型所產生的問題[12]。

Durand 等學者使用降低對比的方法，保存影像的細部特徵，促使高動態範圍影像能正常的顯示[9]。其提出色調映射可略分為兩種方式：廣域(Global)與局部(Local)；於此，Durand 等學者採用局部的手法，將高動態範圍影像分解成數個不同比率的圖層，經處理後再將各個圖層合併，此外，Durand 等學者透過非線性的對稱濾波器(Bilateral Filter)，可免除參數設定，並更快速地得到結果。

關於高動態範圍影像之資訊隱藏，若以最基本的區塊相配演算法來對影像的 RGB 整數部分做資訊隱藏，則會導致影像品質惡化。因為其影像的亮度取決於 E 值，是故在高動態範圍影像中，若兩像素雖有相近的 R、G、B 整數值，但兩者指數值 E 差距很大，則會導致實際上 r、g、b 浮點數值差距很大，而造成決定嵌入量時的錯誤判斷。針對此問題，我們將指數值 E 納入考慮，將高動態範圍影像劃分為區塊與邊界，成功地將資訊嵌入其中。

本文首創於高動態範圍影像的資訊隱藏，除了可迅速地嵌入大量資訊並確保其正確性之外，亦提升不可視性，使偽裝影像具有絕佳的視覺效果。

### 3. 新的高動態範圍影像資訊隱藏演算法

本文提出一個新的高動態範圍影像資訊隱藏演算法，嵌入和取出資訊的流程如圖 1 所示。

嵌入資訊部分：首先，輸入一張掩護影像，利用 RGBE 中的指數值 E 將影像中每一個像素分類，其中，與其相鄰像素 E 值相同的像素稱之為區塊，反之則為邊界。接著，對每個像素做分析處理，如果像素為區塊，則使用區塊相配演算法進行資訊隱藏；如果像素為邊界，則以本身像素的指數值為依據，將相鄰像素的 RGBE 數值轉換，使其相鄰像素之 E 值皆與本身的 E 值相同，再使用區塊相配演算法進行資訊隱藏。最後，輸出一張已嵌入資訊之偽裝影像。

取出資訊部分：首先，輸入一張偽裝影像，並對此影像進行像素分類，劃分出區塊與邊界。接著，根據區塊與邊界以不同的方式取出資訊。最後，將取出的資訊輸出。

演算法區分為三個步驟：1. 像素分類、2. 區塊處理與 3. 邊界處理。以下將詳細說明此三個步驟。

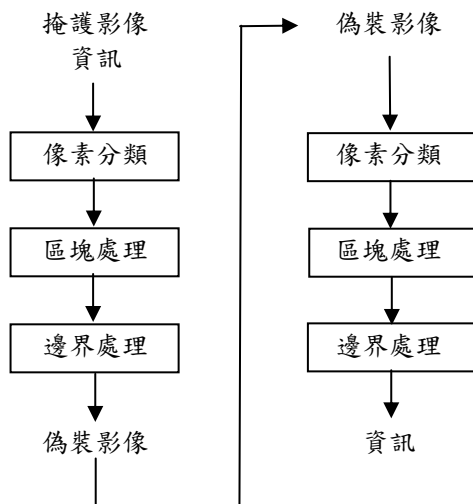


圖 1. 嵌入和取出資訊流程圖。

#### 3.1 像素分類

根據色彩學理論，自然界的亮度分佈皆是循序漸層的，因此，高動態範圍影像相鄰的像素皆有相同或相近的指數值 E。根據此一特性，我們將整張影像切割為區塊與邊界，與相鄰像素之指數值 E 相同的像素稱為區塊，區塊中的亮度相近且顏色較為平滑；與相鄰像素之指數值 E 不同的像素稱為邊界，邊界為亮度變化較大的分界點，故視覺上較醒目且亮度、顏色和周圍像素差距較大。

由於 RGBE 中的指數值 E 於決定最後的亮度值時有偌大的代表性，故區塊相配演算法應用於高動態範圍影像上時，無法直接對邊界處理，因此，我們將針對區塊與邊界提出不同的處理方式。其中，rgb 浮點數型態與 RGBE 整數型態之間的轉換，可藉由圖 2 演算法求得：

$$\begin{aligned}
 v &= \text{fexp}(\max(r, g, b), \&e) * 256 / \max(r, g, b); \\
 R &= r * v; \\
 G &= g * v; \\
 B &= b * v; \\
 E &= e + 128;
 \end{aligned}$$

圖 2. rgb 轉換 RGBE 演算法。

#### 3.2 區塊處理

影像中的區塊部分，我們利用區塊相配演算法嵌入資訊。區塊相配演算法根據參考像素多寡，分成雙邊界(Two-Sided)、三邊界(Three-Sided)及四邊界(Four-Sided)，其中，邊界多寡代表參考的鄰近像素數目[6, 14]。為了避免嵌入的資訊無法取出，參考過的像素不能再次嵌入資訊，是故參考的像素越多，可供嵌入資訊的像素越少，但相對地，色彩改變較少，視覺效果較好。考慮視覺效果和嵌入量，以下我們採用雙邊界的處理方式：以上方和左方的像素作為參考，像素處理的方向如圖 3 所示，灰色部分的像素因無參考的像素，故不嵌入資訊。

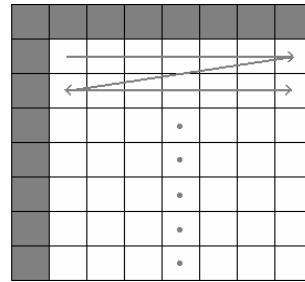


圖 3. 雙邊界區塊相配演算法之像素處理方向圖。

由於 RGBE 中的 R、G 及 B 皆以相同方式處理，為簡化說明，以下皆以 R 為例。假設欲處理的像素 R 值為  $R_x$ ，其上方像素的 R 值為  $R_u$ ，左方像素的 R 值為  $R_l$ ，則差異值  $d$  可由方程式 1 求得。

$$d = (R_u + R_l) / 2 - R_x \quad (1)$$

當  $|d| \leq 1$ ，以最低有效位元取代法嵌入資訊，亦即改變該像素之最後一位元。若  $|d| > 1$ ，則以方程式 2 求得最適合的嵌入量  $n$ 。

$$n = \log_2 |d|, \text{ if } |d| > 1 \quad (2)$$

求得最適嵌入量  $n$  後，再利用嵌入資訊值  $b$ ，藉由方程式 3 算出新的差異值  $d'$ 。

$$d' = \begin{cases} 2^n + b, & \text{if } d > 1 \\ -(2^n + b), & \text{if } d < -1 \end{cases} \quad (3)$$

最後，由方程式 4 取得嵌入資訊後的  $R_x'$  值，其中， $R_x'$  的範圍為 0~255。

$$R_x' = (R_u + R_l) / 2 - d' \quad (4)$$

為了確保嵌入後之數值不會落於 0~255 之外的範圍，在實際嵌入資訊前需再檢查方程式 5 及方程式 6，若任一方程式成立則代表嵌入後之數值將可能落於 0~255 之外的範圍，故不予嵌入資訊。

$$d > 1 \cap (R_u + R_l) / 2 < 2^{n+1} - 1 \quad (5)$$

$$d < 1 \cap (R_u + R_l) / 2 + 2^{n+1} > 256 \quad (6)$$

取出資訊部分與嵌入方式類似，假設欲處理的像素 R 值為  $R_x^*$ ，上方像素的 R 值為  $R_u^*$ ，左方像素的 R 值為  $R_l^*$ 。則可藉由方程式 7 求得差異值  $d^*$ 。

$$d^* = (R_u^* + R_l^*) / 2 - R_x^* \quad (7)$$

當  $|d^*| \leq 1$ ，則取出該像素之最後一位元。若  $|d^*| > 1$ ，則使用方程式 8 求得最適嵌入量  $n$ ，亦即該像素的原始嵌入量大小。

$$n = \log_2 |d^*|, \text{ if } |d^*| > 1 \quad (8)$$

最後，利用方程式 9 算出嵌入資訊值  $b$ ，亦即原始嵌入的資訊值。

$$b = \begin{cases} d^* - 2^n, & \text{if } d^* > 1 \\ -d^* - 2^n, & \text{if } d^* < 1 \end{cases} \quad (9)$$

同樣地，如果方程式 10 或方程式 11 成立，亦即表示之前未嵌入資訊，故不需取出。

$$d^* > 1 \cap (R_u^* + R_l^*) / 2 < 2^{n+1} - 1 \quad (10)$$

$$d^* < 1 \cap (R_u^* + R_l^*) / 2 + 2^{n+1} > 256 \quad (11)$$

### 3.2 邊界處理

影像中屬於邊界部分的像素，因為其相鄰像素的指數值  $E$  不同，故不能直接使用區塊相配演算法嵌入資訊。在嵌入資訊之前，必須先根據本身的指數值  $E$  將相鄰像素的指數值  $E$  轉換成與自己相同的數值。當指數值  $E$  改變後，可藉由方程式 12 求得新的 RGBE 數值。其中， $R'$  為轉換後的 R 值， $E'$  亦為轉換後的  $E$  值。值得注意的是，當轉換後的 RGB 數值超過 255 時，為確保後續的嵌入動作順利，需強迫其降為 255。

$$R' = R \times 2^{E-E'} \quad (12)$$

當上方像素與左方像素皆透過此運算得到與自己相同的指數值  $E$  之新 RGB 值，需再透過方程式 1~6 將資訊嵌入；並藉由方程式 7~11 取出資訊。

## 4. 執行結果與分析

本節說明並分析實驗結果。我們以 C 語言作

為演算法開發程式語言將上述的演算法予以實現，並以 Intel Pentium IV 2.4GHz CPU 加上 1GB 記憶體作為測試平台。

以下將以實際的圖例介紹透過我們提出的演算法處理之後的影像，以便驗證提出的演算法可以有效於高動態範圍影像中嵌入重要資訊。

由於目前尚無 32 位元顯示器，如要觀察高動態範圍影像，必須要透過色調映射技術。圖 4 是以線性色調映射呈現黃昏；圖 5 則是以非線性色調映射呈現黃昏。如圖所示，由於線性色調映射只能侷限於單一範圍下之色調呈現，故無法清楚看見影像中的顏色色調；反之，非線性色調映射呈現的影像則無論在亮度較高或較低的部分皆可正確顯示，其顏色色調豐富、圖案、輪廓均清晰可見，為便於視覺比較，後續的圖例將以非線性色調映射顯示。

首先，以黃昏的高動態範圍影像作為嵌入資訊的掩護影像為例。其解析度為 720×480，圖 5 為未嵌入資訊前之影像，圖 6 為該圖的邊界分佈圖，其中，白色的部分為區塊，共有 29,376 個像素，約佔全圖的 8%；黑色的部分為邊界，共有 316,624 個像素，約佔全圖的 92%。為驗證嵌入資料之隨機性，本文之範例皆以隨機產生的 0 與 1 為嵌入資訊，並於 RGBE 格式下的 R、G 及 B 值中皆嵌入資訊，此外，取出的資訊保證與嵌入之資訊完全一樣。圖 7 為已嵌入資訊後的偽裝影像，總嵌入資訊量為 1,565,053 個位元，平均每個像素約嵌入 4.53 個位元。從視覺化之影像可發現，掩護影像與偽裝影像差異甚小，單憑肉眼無法輕易察覺出之間的差異性，無論邊界或平滑的區域均擁有相當良好的視覺效果。此外，亦可由表 1 中發現，其 PSNR 數值極高，代表嵌入前後的影像擁有極高的相似性。

由此可知，區塊相配演算法在高動態範圍影像中較雜亂的區域嵌入較多的資訊，而於較平滑的區域嵌入較少的資訊，有效的利用像素的可嵌入量，並保持良好的視覺效果。換言之，若掩護影像擁有越多的雜亂區域，則可嵌入越多的資訊。以下，將分別以建築及庭院二影像代表較平滑及較雜亂之影像範例，說明其可嵌入量及視覺效果之差異性。

首先，圖 8 建築影像為雜亂區域較少的範例，其解析度為 660×433，圖 9 為其邊界分佈圖，其中，約有 87% 為區塊；13% 為邊界。圖 10 則為已嵌入資訊後之偽裝影像，總嵌入的資訊量為 1,552,057 個位元，平均每個像素約嵌入 5.43 個位元。

另外，圖 11 庭院影像為雜亂區域較多的範例，其解析度為 660×433，圖 12 為其邊界分佈圖，其中，約有 52% 為區塊；48% 為邊界，邊界像素約佔整張影像的一半。圖 13 為已嵌入資訊的偽裝影像，總嵌入的資訊量為 2,795,522 個位元，平均每個像素約嵌入 9.78 個位元。

圖 8 建築影像與圖 11 庭院影像相比，由於庭院影像中的邊界部分較多，而邊界部分通常亦為影像中較雜亂的區域，因此其平均嵌入量也較多。然而，無論影像屬於較雜亂或較平滑的影像，皆可達

到極佳的視覺效果，此外，亦可由表 1 中發現，其 PSNR 數值極高，代表嵌入前後的影像擁有極高的相似性。由此可知，本文提出的演算法能夠有效地對平滑與雜亂程度不同的區域選擇適合的資訊嵌入量，在保持良好視覺效果的前提下，達到較多的嵌入量，詳見表 1 之數據比較表。

## 5. 結論與未來工作

本文首創之新的高動態範圍影像資訊隱藏演算法成功地嵌入資訊於擁有更不可視之特性的高動態範圍影像，產生不易為外人所察覺的視覺效果。除了保留區塊相配演算法重視不可視的特性外，也可針對影像的特性變動適合的嵌入量，達到高嵌入量及低可視之目的，此外，由於演算法的複雜度不高，故亦能達到高效率的嵌入效能。

有鑒於高動態範圍影像勢必為未來之趨勢，本文首創於高動態範圍影像上提出資訊隱藏演算法。根據高動態範圍影像與生俱來的特性，將其每一個像素依指數值  $E$  劃分為區塊與邊界，並對不同的區域進行不同的嵌入處理：區塊部分以區塊相配演算法進行資訊隱藏，針對該像素與鄰近像素之差異性動態決定其最適嵌入量；邊界部分則須先透過轉換的方式達到共同的指數基準值，再輔以區塊相配演算法進行處理，動態決定其最適嵌入量，創造出高動態範圍影像獨特的資訊隱藏演算法。實驗結果顯示，我們的演算法能有效率地嵌入資訊於高動態範圍影像，達到高嵌入量及低可視性的目的。

未來工作方面，我們建議朝向幾個方向繼續研究：第一、增加區塊嵌入量：若區塊部分可參考人類視覺系統及色調映射技術改變其嵌入規則，則可於保留偽裝影像低可視性之基礎下，再提升其嵌入量。第二、增加邊界嵌入量：由於受限於 8 位元整數值範圍為 0~255 的限制，在邊界處理部分，目前轉換後超出 255 的數值皆須強迫降為 255，此乃限制了其真正的差異性，若可於轉換中保留其差異性，而於最後嵌入階段才檢驗其正確性，則可更進一步增加嵌入量。第三、增加廣泛性：嘗試是否能將我們的演算法，延伸至其餘高動態範圍影像的格式上。目前為止，我們提出的演算法專注於高動態範圍影像主流格式之一的光輝 RGBE 格式。

## 致謝

本研究承蒙國科會之經費補助(94-2213-E-005-022)，謹此致謝。

## 參考文獻

- [1] R. Anderson and F. Peticolas, "On The Limits of Steganography," *IEEE Journal on Selected Areas in Communications*, Vol.16, No. 4, pp. 474-481, 1998.
- [2] M. Ashikhmin, "A Tone Mapping Algorithm for High Contrast Images," *In Proceedings of the*

- 13<sup>th</sup> Eurographics Workshop on Rendering*, pp. 145-156, 2002.
- [3] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for Data Hiding," *IBM Systems Journal*, Vol. 35, No. 3&4, pp. 313-336, 1996.
- [4] C.-C. Chang and H.-W. Tseng, "A Steganographic Method for Digital Images Using Side Match," *Pattern Recognition Letters*, Vol. 25, No. 12, pp. 1431-1437, 2004.
- [5] C.-C. Chang, W.-L. Tai, and M.-H. Lin, "A Reversible Data Hiding Scheme with Modified Side Match Vector Quantization," *In Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05)*, pp. 947-952, 2005.
- [6] P. Debevec and J. Malik, "Recovering High Dynamic Range Radiance Maps from Photographs," *In Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 369-378, 1997.
- [7] R. Fattal, D. Lischinski, and M. Werman, "Gradient Domain High Dynamic Range Compression," *In Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, ACM SIGGRAPH, pp.249-256, 2002.
- [8] N. F. Johnson and S. Jajodia, "Exploring Steganography: Seeing the Unseen," *IEEE Computer*, Vol. 31, No. 2, pp. 26-34, 1998.
- [9] S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High Dynamic Range Video," *ACM Transactions on Graphics*, Vol.22, No.3, pp. 319-325, 2000.
- [10] T. Kim, "Side Match and Overlap Match Vector Quantizers for Images," *IEEE Transactions on Image Processing*, Vol. 1, No. 2, pp. 170-185, 1992.
- [11] F.A.P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn, "Information Hiding- A Survey," *In Proceedings of the IEEE*, Vol. 87, No. 7, pp. 1062-1078, 1999.
- [12] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, "Photographic Tone Reproduction for Digital Images," *In Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, ACM SIGGRAPH, pp. 267-276, 2002.
- [13] R. Z. Wang, C. F. Lin, and J. C. Lin, "Image Hiding by Optimal LSB Substitution and Genetic Algorithm," *Pattern Recognition* Vol. 34, pp. 671-683, 2001.
- [14] G. Ward, "Real Pixels," *Graphics Gems II*, Ed. by J. Arvo, Academic Press, pp. 80-83, 1992.
- [15] H. C. Wei, P. C. Tsai, and J. S. Wang, "Three-Sided Side Match Finite State Vector Quantization," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 10, No. 1, pp. 51-58, 2000.



圖 4. 線性色調映射呈現的影像無法同時顯示高亮度及低亮度的部分。



圖 5. 非線性色調映射呈現的影像可同時顯示高亮度及低亮度的部分。



圖 6. 圖 5 之邊界分佈圖。

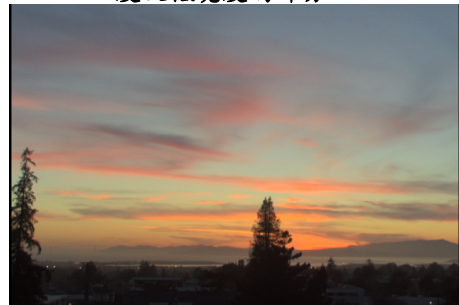


圖 7. 圖 5 嵌入隨機資訊後所得之偽裝影像。



圖 8. 較少雜亂區域的掩護影像。

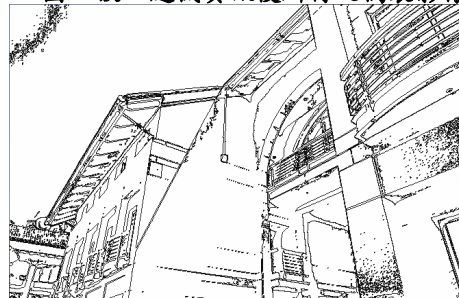


圖 9. 圖 8 之邊界分佈圖。



圖 10. 圖 8 嵌入資訊後所得之偽裝影像。



圖 11. 較多雜亂區域的掩護影像。



圖 12. 圖 11 之邊界分佈圖。



圖 13. 圖 11 嵌入資訊後所得之偽裝影像。

表 1. 黃昏、建築及庭院影像之比較表。

	解析度	邊界比例 (%)	區塊比例 (%)	邊界平均嵌入量 (位元/像素)	區塊平均嵌入量 (位元/像素)	總嵌入量 (位元)	全圖平均嵌入量 (位元/像素)	執行時間 (秒)	PSNR
黃昏	720×480	8.50	91.50	8.14	4.21	1,565,053	4.53	1.468	49.433
建築	660×433	12.90	87.10	9.95	4.72	1,552,057	5.43	1.250	49.215
庭院	660×433	47.83	52.17	11.81	7.99	2,795,522	9.78	1.953	43.934