

高效能彩色影像漸進傳輸研究與實作

陳文儉¹ 王秋祥¹ 梁世昌²

¹大葉大學資訊工程學系

51591 彰化縣大村鄉學府路 168 號

²光特資訊科技股份有限公司

10359 台北市大同區重慶北路 2 段 197 號 3 樓

摘 要

本論文提出一個彩色影像漸進傳輸原型，利用影像金字塔結構將原影像分割成多重解析度，接著將每一層影像資訊利用無失真壓縮加以編碼。此原型包含兩個部分伺服器與客戶端；在伺服器端，提供小波編碼、色彩轉換、小波係數壓縮儲存與漸進傳輸功能。在客戶端，包含小波解碼、色彩轉換、接收與已傳送的小波係數合成，最後將已傳送影像合成，逐漸改善重建影像的品質。實驗結果顯示各層重建時間大概只需 2~3 秒，本論文所開發的原型，可以投入商業應用中。

關鍵詞：影像漸進傳輸，小波轉換，色彩轉換

Study and Implementation of High Effect Progressive Color Image Transmission

WEN-JAN CHEN¹, QIU-XIANG WANG¹ and SHIH-CHANG LIANG²

¹*Department of Computer Science and Information Engineering, Da-Yeh University*

168 University Rd., Dacun, Changhua 51591, Taiwan, R.O.C.

Quantasoft Technology Incorporation

3F, No. 197, Sec. 2, Chongcing N. Rd., Datong District, Taipei 10359, Taiwan, R.O.C.

ABSTRACT

In this paper, a prototype for the progressive transmission of a color image is presented. The original image is segmented into multiple-resolution images by a pyramid structure; then the information for each layer is coded by lossless compression. The prototype consists of a server and a client. The server provides wavelet transform, color transform, storage, compressed wavelet coefficients and progressive transmission. In the client portion, by inverse wavelet and color transforms, the received wavelet coefficients are used to build successively-improving reconstructions of the image. The experimental results indicated that the required reconstruction time for each level is only 2 to 3 seconds. The prototype developed in this research can be practically implemented in commercial applications

Key Words: progressive image transmission, wavelet transform, color transform

一、緒論

隨著網際網路與無線網路的發展，數位化多媒體的需求也日益增。在有限的頻寬，加上愈來愈多使用者的情況下，如何讓高解析度彩色影像能有效率的壓縮與傳送的技术，是目前相當重要的研究主題。使用漸進傳輸的動機起源於需要透過低頻寬的通道傳輸影像，例如使用無線網路、撥接上網...等低頻寬傳送影像資料，它能夠做到快速辨認影像，在傳送時間有限的情况下，特別適合使用。且使用影像漸進傳輸有可能會讓使用者省下一大筆時間。

傳統的影像傳輸模式皆屬於序列式 (sequential) 的模式，這種傳輸模式是每一次從影像最左上角的像素 (pixel) 開始傳送，一次一列接一列傳送，其缺點是，無法在第一次的傳送裡，提供粗略的影像資訊，有的只是影像第一列的資訊而已，當影像已達到可辨認的範圍時，通常是已傳送了影像大部分資訊。

影像漸進傳輸模式通常提供較好的選擇，此傳輸模式第一次傳送通常會先傳送較低解析度的影像，再漸進地補送資訊與已傳送的影像合成，讓顯示端中的影像逐漸清晰，直到合成的影像能完成原始品質重建，或者使用者覺得影像品質已達到滿足即可停止傳送。這種傳輸模式的優點是，第一次傳送的低解析度的影像通常已能粗略地顯示出影像中輪廓，進而讓使用者選擇是否需要繼續傳送影像資訊，可替使用者減少資訊傳送量與時間。

影像漸進傳輸的編碼方式可使用好幾種方法；Al-Asmari 與 Ahmed [3]、Voukelatos 與 Soraghan [14] 及 Wang 與 Goldberg [15, 16] 都是利用 VQ 編碼的方式，Goldberg 與 Wang [6]、Kim 與 Song [8]、Mongatti 等人 [9] 及 Zeng 等人 [17] 是利用影像金字塔結構將原始影像分割成多重解析度的影像，Chung 與 Tseng [5] 及 Hu 與 Jiang [7] 是利用 quadtree 的特性將影像拆解成好幾階，另外 Qi 與 Tyler [11] 是將影像漸進傳輸應用在 3D 醫學影像上，Chang 等人 [4] 及 Oller 等人 [10] 則是利用小波轉換的特性對其編碼與傳送。

本論文利用影像金字塔結構將原始影像分割成多重解析度，再利用無失真壓縮將每一層的影像資訊加以編碼，以滿足高效率的漸進傳輸要求。本論文所用影像為全彩高解析度都市地區航照影像，經由小波轉換 (wavelet transform)，所產生小波係數 (wavelet coefficients) 執行色彩轉換後，再經由 Shapiro [13] 學者所提出的嵌入式零樹小波 (embedded

zerotree wavelet, EZW) 影像壓縮法，經 EZW 每一回合編碼後的符號，使用字典模式 (dictionary techniques) 無失真壓縮方法來編碼，最後寫出位元串流 (bit stream) 檔；漸進傳輸部分，則是先讀取最上層的檔案，解壓縮後建立重建影像，再由上而下逐層讀入檔案，解壓縮後與已傳送的資訊做合成，再建立逐層重建影像，重建影像會逐漸清晰。

二、相關技術回顧

(一) YIQ 色彩轉換演算法

RGB 就是紅綠藍三原色。若我們將 R、G、B 影像分別以相同的方法做壓縮，通常是沒辦法做到很理想的壓縮，這是由於 R、G、B 影像間存在著很大的關聯性，而分別處理是沒辦法去掉這些累贅的。大部份的彩色影像壓縮法都是使用轉換的觀念，將 R、G、B 轉換成亮度 (luminance) 及彩度 (chrominance)。而本文所採用的是 YIQ 色彩轉換系統，轉換後的訊號大部份能量集中到 Y 平面，其中能量高達約 93%，I 及 Q 平面分別大約佔總能量的 5%、6%。這表示 I 和 Q 兩張彩色平面像素與像素之間的差異極小，有利於壓縮。YIQ 色彩轉換公式如下所示：

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ 0.596 & -0.274 & -0.322 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

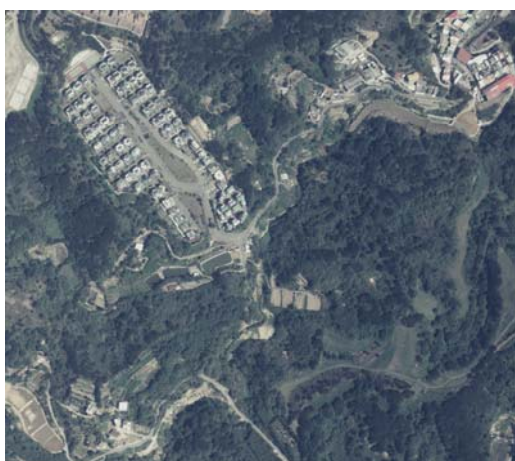
利用反矩陣可將公式 1 推導出 YIQ 逆轉換式子。事實上 YIQ 色彩轉換可使用其它的色彩轉換代替，例如：YCbCr、YUV...等，其目的只要能將能量集中到 Y 平面，讓其餘彩度平面只佔能量的少部分，這會使得壓縮更有效率。

(二) 小波轉換演算法

彩色影像漸進傳輸，利用影像金字塔結構將原始影像分割成多重解析度的影像，一張影像經過小波轉換處理後會產生重要性不同的影像資訊，這些資訊會構成高低不同的頻率，而這些高低不同的頻率也就是不同解析度影像。對人類視覺系統來說，低頻 (low frequency) 是人眼可以接受敏感度較高的部份，低頻像素與像素之間變化小，影像較平滑，所以低頻的值只要稍有改變，人眼立即能察覺之間的差異；對於高頻 (high frequency)，人眼對它的敏感度較低，因為高頻的像素與像素之間差異較大，所以對高頻的值做少許修

改，肉眼是無法清楚察覺出來的。基於這個概念本文利用小波轉換將影像分解成多重解析度的影像，以小波轉換最上層 LL 頻帶 (subband) 做為第一次傳送的影像資訊，因為 LL 頻帶包含影像大部分資訊。完成小波轉換後如圖 1 所示，圖 2 為各頻帶相對應關係。

圖 1(a) 為航照圖的原始影像 1877×1681 大小，圖 1(b) 為圖 1(a) 原始影像經過 4 階小波轉換後的結果，小波轉換各頻帶相對應關係圖如圖 2 所示。低頻部份是對每一次小波轉換後的結果，再切割所得到的，也是影像重要部份再切割得到的最重要部份。最左上角 LL4 頻帶是影像中最低頻帶，觀察 LL4 可以發覺原始影像大致輪廓，LL4 類似縮小後的原始影像，而影像大部分資訊也都集中在此；其餘的 HL4、LH4、HH4、HL3、LH3、HH3...等頻帶，都是影像中的高頻，是經由像素與像素間相減所得到的差值，係數值不大，



(a) 原始影像



(b) 經 4 階小波轉換後

圖 1. 航照圖

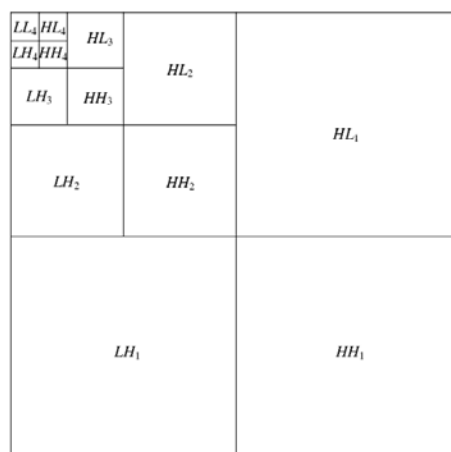


圖 2. 小波轉換各頻帶相對應關係圖

大部分小波壓縮演算法的重點是對高頻做壓縮。由於中頻 (LH、HL) 及高頻 (HH) 是用來加強低頻的，即使是對高頻部分做有失真壓縮，捨去值較小的高頻係數，對重建影像的品質影響不大。

本論文所採用的小波轉換是 Haar 小波轉換，因為 Haar 小波是一種簡單又有效率的小波轉換方法，它只需要經過簡單相加、相減的運算就能獲得小波係數，雖然目前有很多不同的小波如 Daubechies，其效能不錯但因其轉換計算較複雜，關於 Haar 及 Daubechies 小波轉換詳細的介紹可參考戴顯權著作 [2]，而本研究論文之結果希望能應用在商業軟體上，需非常注重計算量，所以採用最簡單的 Haar 小波轉換。本文採用 4 階 (scale) 小波轉換，經由 Haar 小波編碼將產生 5 個區域，其中一個區域為低頻 LL 頻帶，其餘四個區域皆為 HL、LH、HH 頻帶所組成，包括 (HL₁、LH₁、HH₁)、(HL₂、LH₂、HH₂)、(HL₃、LH₃、HH₃) 及 (HL₄、LH₄、HH₄) 四個區域。採用 4 階的原因主要是未來實作在商業軟體上，因實際的航照影像解析度非常大，為使 LL 頻帶的資料量不要太大，而不採用一般論文在 512×512 解析度下的 2 階或 3 階。

(三) EZW 演算法

小波轉換後，低頻部份是對像素做平均運算的結果，再次平均運算所得到的，也是影像重要部份再次平均運算得到的最重要部份；影像中的高頻，是經由像素與像素間相減所得到的差值。所以，在低頻帶中的係數值會高於高頻帶中的係數。由於人類視覺系統對於低頻係數較為敏感而對高頻係數較不敏感，所以有時捨去部份高頻係數，對於視覺品質影

響不大，EZW 演算法就是利用這項特性，將係數切割成好幾階再加以編碼送出。

1933 年由 Shapiro 學者所提出的 EZW 演算法有很重要的貢獻，他是在小波係數上從事影像壓縮器實作的先驅。在 EZW 演算法中，關鍵地方是各頻帶的相對應關係，由圖 2 中表示出 HH_4 、 HH_3 、 HH_2 和 HH_1 頻帶在空間位置上有相對應關係； HL_4 、 HL_3 、 HL_2 和 HL_1 頻帶在空間位置上有相對應關係； LH_4 、 LH_3 、 LH_2 和 LH_1 頻帶也有相對應關係，它們都有父子相依關係。依照這個關係 EZW 演算法可以節省下很多不必要的編碼；以下將透過一個例子來解釋 EZW 演算法。

首先，先定義 EZW 演算法將使用到的符號：

- POS (positive significant)：若該小波係數大於 T ，且為正整數，該係數會被標識為 POS。
- NEG (negative significant)：若該小波係數的絕對值大於 T ，且為負整數，該係數會被標識為 NEG。
- IZ (isolated zero)：若該小波係數的絕對值小於 T ，但其相對應的子孫係數中，有一個以上係數的絕對值大於 T 時，則該係數會被標識為 IZ。
- Z (zero)：若該小波係數小於 T ，且該係數已沒有子孫，則該係數會被標識為 Z，事實上符號 Z 可改用 ZTR 代表。
- ZTR (zerotree root)：若該小波係數的絕對值小於 T ，而且該係數的子孫係數絕對值皆小於 T 時，則該係數會被標識為 ZTR。

EZW 在進行編碼時，是採用 Z 字掃描方式圖 3 為 Z 字掃描方式示意圖。依圖 3 的掃描順序會依次將小波係數分類。詳細的例子可參考蔡文輝等人著作 [1]。在 1996 年由 Said 與 Pearlman [12] 共同提出架構樹集合分割 (set partitioning in hierarchical trees, SPIHT) 來改善 EZW，其壓縮效果比 EZW 來的好，另在 2000 年所製定的影像壓縮標準 JPEG2000，在壓縮效果也比 EZW 好，如戴顯權 [2] 所介紹；但 SPIHT 與 JPEG2000 的複雜度都相對的提高，所以本研究論文採用計算簡單且效果也還不錯的 EZW 來實作，以利在商業軟體上應用。

三、高效能影像漸進傳輸系統實作

高效能影像漸進傳輸，利用影像金字塔結構將原始影像分割成多重解析度的影像，一張影像經過小波轉換處理後會產生重要性不同的影像資訊，這些資訊會構成高低不同的頻

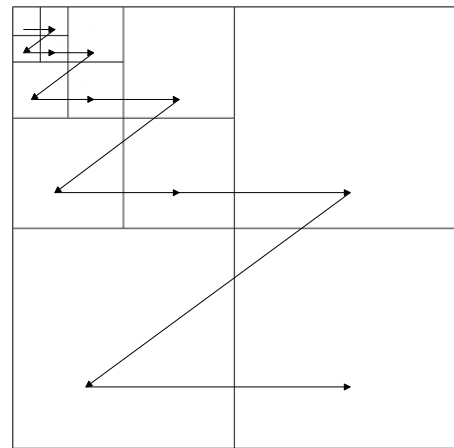


圖 3. Z 字掃描方式

率，而這些高低不同的頻率也就是不同解析度影像。再透過無失真壓縮將每一層影像資訊加以編碼，以滿足高效率的漸進傳輸要求。

(一) 系統架構

本文提出的伺服器系統流程如圖 4 所示。彩色影像會先分解成 R、G、B 三個平面，分別進行 Haar 小波轉換得到 R、G、B 小波係數再經由 YIQ 色彩轉換，將得到的 Y、I、Q 小波係數進行 EZW 編碼及無失真壓縮，這裡所採用的是 Zip 這類的字典模式壓縮，最後分別以 LL 頻帶及 EZW 編碼次序寫出檔案。

客戶端系統流程如圖 5 所示，讀入第 1 個 zip 檔案，此檔案包含 LL 頻帶資訊及相關檔頭，此檔頭資訊記錄了原始影像大小、各階影像大小、此影像有幾階小波轉換、傳送順序...等資訊，後面將會有詳細敘述檔頭資訊。接著，對 zip 進行解壓縮並分析檔頭資訊，且執行 EZW 解碼得到 YIQ 色彩資訊，接著反向 YIQ 色彩轉換，分別得到 R、G、B 小波係數，再對其係數執行反向 Haar 小波轉換，產生一張與原

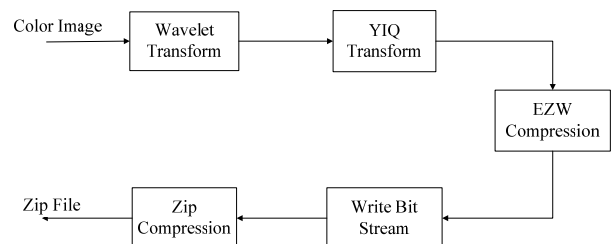


圖 4. 伺服器系統流程圖

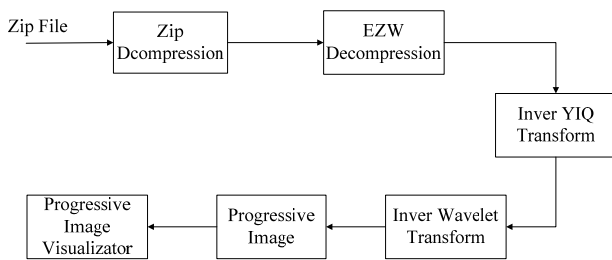
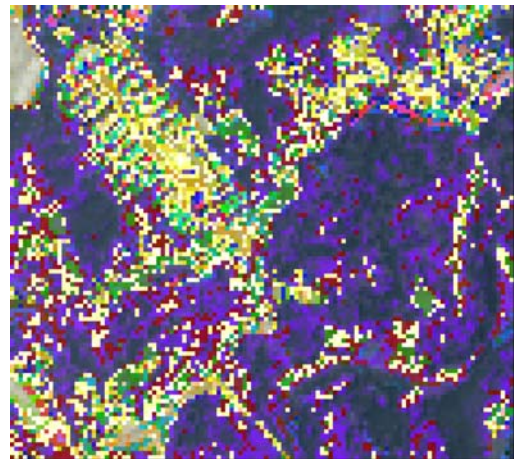


圖 5. 客戶端系統流程圖

始影像大小相同的粗略重建影像並顯示於用戶端，此影像已包含影像最重要的資訊。接著，依序送出其它資訊與已傳送的影像合成，逐漸改善影像品質使其逐漸清晰，直到所有的高頻資訊傳送完畢達到與原始影像相同品質，或者使用者覺得已達到所需要的影像品質即可停止傳送。

從圖 4 可以觀察到 YIQ 色彩轉換程序是被放在 Haar 小波轉換後面，這種做法與一般的彩色影像壓縮先將 RGB 三原色平面經由 YIQ 色彩轉換後接著執行小波編碼的方法不同，因為如果先 RGB 三原色平面執行 YIQ 色彩轉換再執行 Haar 小波轉換，這會造成 I、Q 色彩平面影像類似取樣過度，導致顯示第 1 層影像漸進傳輸時，因色彩資訊不足而造成影像失真，這在逐漸補送剩餘資訊，色彩顯示才逐漸正確。所以，本研究發現，直接對 R、G、B 小波係數進行 YIQ 色彩轉換，在解碼時能產生正確的彩色影像；經研究發現，對 R、G、B 小波係數執行 YIQ 色彩轉換，依然保留能量中集的效果，再經由無失真壓縮有較良好的壓縮倍率。

接下來說明為何要將 YIQ 色彩轉換流程擺在 Haar 小波轉換之後。一般影像壓縮流程 RGB 影像執行 YIQ 色彩轉換後，其中 Y 平面的能量高達約 93%，I 及 Q 平面分別大約佔總能量的 5%、6%；接著對 Y、I、Q 平面執行 Haar 小波轉換，得到小波係數；若不做漸進傳輸，直接把 YIQ 小波係數壓縮送出，至解碼端解碼後的影像會如圖 1(a) 原始影像所示，色彩正確顯示。但漸進傳輸模式中，會先傳送 Y、I、Q 的 LL 頻帶，而 LL 頻帶是反覆經過 Haar 小波轉換後的結果；Haar 小波轉換演算法所得到的 LL 頻帶類似於，對影像一再的做次取樣，所得到的最後結果；圖 6(a) 為對 RGB 影像轉換成 Y、I、Q 平面執行 Haar 小波轉換後的 LL 頻帶重建影像，由圖 6(a) 顯示出 Y、I、Q 經一再的次取樣造成色彩資訊不足，造成重建影像無法提供正確的色彩資訊而顯示錯誤，其 PSNR 只有 20 dB 左右；這時再逐層傳送 EZW



(a) LL 頻帶的重建影像



(b) 加入部份 EZW 資訊後的重建影像

圖 6. I、Q 色彩資訊不足失真影像

部份資訊與已傳送影像合成，重建影像會逐層清晰且色彩顯示愈來愈正確，直到所有資訊傳送完畢得到的重建影像會如圖 6(b)，其影像中的輪廓已經相當清晰，但與原始影像圖 1(a) 相較之下，可發現圖 6(b) 有極大部分色彩顯示是錯誤的。要解決此問題須先對 RGB 做小波係數轉換再執行 YIQ 色彩轉換，而不是直接對 RGB 像素執行 YIQ 色彩轉換再做小波係數；此方法能解決重建影像色彩顯示錯誤問題，且影像大部分資訊量還是集中在 Y 小波係數上，這是因為 Haar 小波的 LL 頻帶相當於一張次取樣後的影像，依然保留了影像中重要的資訊，將這張縮小後的影像轉到 Y、I、Q 平面上依然與原始影像有著同樣的效果。將 Haar 小波替換成 5/3 小波，在重建影像也同樣會有色彩顯示錯誤的問題。

(二) 檔案格式

在這一節裡將詳細說明每一項檔頭項目，共有三種檔頭

格式，每個表格次序為檔頭儲存次序。

表 1 為 LL 頻帶檔頭，檔頭佔總檔案 264 bits，所儲存的資料還包括 Y、I、Q 的 LL 頻帶和影像邊界無法被 Haar 小波處理的像素，並且儲存 EZW 第一次切割的資訊。其中資料位元率 (bit per data, BPD) 用來紀錄每個像素需要用多少 bit 儲存，支配資料 (dominant data, DD) 為支配回合資料的長度，而精鍊資料 (refinement data, RD) 為精鍊時資料的長度。

表 2 為每次切割 EZW 編碼後所需要紀錄的檔頭資訊，檔頭佔總檔案 168 bits。

表 3 為執行 EZW 編碼後剩餘的差值，所需要紀錄的資訊，檔頭佔總檔案 32 bits，差值檔案是用在最後解碼時；最後解碼時不一定需要差值檔案做補正，可視使用者的需求所接收的影像是否已達到可接受的視覺品質。

表 1. LL 頻帶檔頭

項目	大小 (bits)	描述
Order	8	紀錄解碼順序。
BPD	8	紀錄每個像素需要用多少 bit 儲存。
LL Height	12	紀錄 LL 頻帶高的大小。
LL Width	12	紀錄 LL 頻帶寬的大小。
Height	12	紀錄原始影像高的大小。
Width	12	紀錄原始影像寬的大小。
Remain Size	16	紀錄影像邊界無法被 Haar 小波處理的總長度。
Scale	8	紀錄影像共有幾階小波轉換。
Y Plane Threshold	8	紀錄 Y 平面上 EZW 第一次切割的門檻值。
I Plane Threshold	8	紀錄 I 平面上 EZW 第一次切割的門檻值。
Q Plane Threshold	8	紀錄 Q 平面上 EZW 第一次切割的門檻值。
Times	8	紀錄 EZW 共經過幾次切割。
Y Plane DD Length	24	紀錄 Y 平面 DD 長度。
Y Plane RD Length	24	紀錄 Y 平面 RD 長度。
I Plane DD Length	24	紀錄 I 平面 DD 長度。
I Plane RD Length	24	紀錄 I 平面 RD 長度。
Q Plane DD Length	24	紀錄 Q 平面 DD 長度。
Q Plane RD Length	24	紀錄 Q 平面 RD 長度。

表 2. 每次 EZW 編碼後檔頭

項目	大小 (bit)	描述
Order	8	紀錄解碼順序。
BPD	8	紀錄每個像素需要用多少 bit 儲存。
Times	8	紀錄此檔案屬於第幾切割的 EZW 編碼。
Y Plane DD Length	24	紀錄 Y 平面 DD 長度。
Y Plane RD Length	24	紀錄 Y 平面 RD 長度。
I Plane DD Length	24	紀錄 I 平面 DD 長度。
I Plane RD Length	24	紀錄 I 平面 RD 長度。
Q Plane DD Length	24	紀錄 Q 平面 DD 長度。
Q Plane RD Length	24	紀錄 Q 平面 RD 長度。

表 3. 差值檔頭

項目	大小 (bit)	描述
Order	8	紀錄解碼順序。
Y Difference BPD	8	紀錄 Y 平面剩餘的差值每個像素需要用多少 bit 儲存。
I Difference BPD	8	紀錄 I 平面剩餘的差值每個像素需要用多少 bit 儲存。
Q Difference BPD	8	紀錄 Q 平面剩餘的差值每個像素需要用多少 bit 儲存。

四、實驗結果與分析

在本章中，實際的實作本文所提出的系統架構，並顯示及分析結果影像。本文所使用測試影像如圖 1(a) 為光特資訊科技股份有限公司所提供的 1877×1681 全彩航照影像，扣除檔頭，實際檔案大小為 1877×1681×3= 9,465,711 bytes。測試影像為 1877×1681，不是 2 的乘幂，所以在實作時，我們將水平及垂直分割時剩下的像素點放在檔頭不做壓縮，傳給接收端。實作平台使用 Processor: Intel Core 2 Duo Mobile T7300 @ 2000 MHz, Physical Memory: 4096 MB (2×2048 DDR2-DRAM), Hard Disk: 160 GB (5400 RPM)。本文所開發的成果軟體，將立即投入商業應用，因此，程式設計將採用目前普遍使用的 Java 程式語言，Java 除了在網路上有非常豐富的資源，物件導向 (object-oriented) 的特性可使系統有彈性易於維護，跨平台的特性可執行在不同作業系統。開發工具則採用 Eclipse3 整合開發環境 (integrated development environment, IDE)，Eclipse 是當今主流的 Java 開發工具，能解決 Java 開發細節，以提升程式開發效率。下列將會檢視數據：各層影像的總儲存量、重建原影像所花費的總時間、重建後影像的失真程度。各層影像的總儲存量

應該要愈小愈好，儲存量小表示未來經由網路傳輸的資料量較小；重建原影像所花費的總時間也是愈少愈好，時間少表示前端電腦所耗用的硬體資源較少；重建後影像的失真程度要愈低愈好，失真低表示顯示的品質較接近原影像。

伺服器實作步驟如下：

Step 1. *doColorImageWaveletTrans* (*original_image*, *scale*): *original_image* 為原始影像，*scale* 為共執行幾階小波轉換；首先直接將原始影像拆成 R、G、B 平面，分別執行小波轉換。

Step 2. *doWtDataToYIQ* (*wtdatas*): *wtdatas* 物件包含 R、G、B 小波轉換後的係數；此步驟是將 *wtdatas* 經由 YIQ 色彩轉換。

Step 3. *doEZWCompression* (*yiQ_wtdatas*, *times*): *yiQ_wtdata* 為 R、G、B 小波係數經 YIQ 色彩轉換後的物件，*times* 為執行 EZW 切割次數；此步驟將 *yiQ_wtdatas* 經過 *times* 次數的 EZW 切割，本實驗 *times* = 4。

Step 4. *toBitstream* (*ezwdatas*): *ezwdatas* 物件為經過 EZW 編碼後之結果；此步驟將 EZW 編碼後的資訊加入檔頭資訊，一次一個 byte 為單位寫出。

Step 5. *writeBitstream* (*bitstreams*, *output_file_name*): *bitstreams* 為 EZW 加入檔頭後以串流方式寫出的物件，*output_file_name* 為出檔案的名稱；最後步驟將 *bitstreams* 這個物件以 ZIP 壓縮後寫出檔案。

客戶端實作步驟如下：

Step 1. *readFirstBitstream* (*input_file_name*): *input_file_name* 為帶有 LL 頻帶及 EZW 第一切割的編碼資訊檔案；首先第 1 步驟會解壓縮 ZIP 壓縮檔並且取出 *bitstreams* 資料。

Step 2. *doEZWDecompression* (*read.getWtEzwDdatas* (), *times*): *read.getWtEzwDdatas* () 為 EZW 的編碼資訊，*times* 為此資料屬於第幾次 EZW，第 2 步驟將會分析第 1 步驟所讀取得到的 *bitstreams* 資料，並且執行 EZW 解碼。

Step 3. *doYIQToWtdatas* (*yiQ_wtdatas*): *yiQ_wtdatas* 為第 2 步驟經過 EZW 解碼後產生的 YIQ 小波係數；此步驟將會執行逆 YIQ 轉換。

Step 4. *writeCurrentResultJPG* (*wtdatas*, *output_file_name*): *wtdatas* 為第 3 步驟產生的 RGB 小波係數；此步驟首先對 *wtdatas* 執行逆小波轉換，到此步驟已完成第一

個檔案的讀取及分析。

Step 5. *readNextBitstream* (*read*, *input_file_name*, *output_file_name*): *read* 物件為已接收檔案，此步驟接受到檔案後，將會執行如步驟 1 到 4，差別在於所接受到的檔案與第 1 步驟所接受到檔案不同，所以分析檔頭方式也不相同，最後所接收檔案將會與已接收檔案結合。

Step 6. *readDifferenceBitstream* (*read*, *input_file_name*, *output_file_name*): 最後這一步驟是接收差值檔，並且與已接收檔案結合，最後結果是無失真的。

本實作所使用的小波轉換階數為 4 (*scale* = 4)，EZW 切割次數為 6 (*times* = 6) 次。實作上將 EZW 每次切割的資訊送出，如圖 7 所示為 EZW 演算法各層重建影像。另外圖 8 為不含 EZW 演算法各層重建影像。表 4 為使用 EZW 演算法各層重建比較，表 5 為使用不含 EZW 演算法各層重建比較。從圖 7、8 及表 4、5 可以看出 EZW 演算法在各層重建影像不管在視覺效果或 PSNR 比不含 EZW 演算法都有較好的結果。

從表 4 及表 5 可以發現 2、3、4 層所傳送的資訊量並不是很多，且 PSNR 變化不大，重建效率也較不理想；從圖 7 及圖 8(b)、(c)、(d) 之間的變化可發現，2、3、4 層傳送的資訊為影像中 edge 較明顯部份（高頻係數值較大的部分），由於這部份對人眼的視覺較為不敏感，所以試著將表 4 及表 5 的 2、3 層資訊合併成一個檔案，4、5 層資訊合併成一個檔案，產生 5 層變化，如圖 9 及圖 10 所示 5 層變化結果影像。表 6 及表 7 則為 5 層變化的結果。

同樣地，可以看出 5 層變化的 EZW 演算法在各層重建影像不管在視覺效果或 PSNR 仍然比不含 EZW 演算法都有較好的結果。而合併 2、3 及 4、5 層，可節省重建總時間，且使得各層重建影像變化較為明顯；比較表 6 及表 7 發現，第 5 層資訊量使用 EZW 演算法能有效的將資訊壓縮；而不使用 EZW 演算法，則使得第 5 層明顯得比其它層的資訊量來得多，會造成傳送過程中到第 5 層時傳送時間比其它層來得緩慢，同時也造成重建時間比其它層來得長。如果藉著修改門檻值調整各層容量大小，會因為前幾次重要係數過多，反而增加後面各層累贅，有可能會使得總容量比原始影像來得大。



(a) 第 1 層重建影像



(b) 第 2 層重建影像



(c) 第 3 層重建影像



(d) 第 4 層重建影像



(e) 第 5 層重建影像



(f) 第 6 層重建影像



(g) 第 7 層重建影像

圖 7. 航照影像，EZW 演算法 7 層重建影像



圖 8. 航照影像，小波轉換 7 層重建影像

表 4. 航照影像，各層重建比較 (EZW 演算法)

各層級	容量大小 (byte)	重建時間 (ms)	PSNR (RGB 平均)
1	35,735	2,437	35.859
2	4,356	2,453	36.041
3	83,335	2,484	36.939
4	490,842	2,985	38.394
5	1,085,757	3,203	40.329
6	1,661,384	3,734	41.807
7	2,642,135	3,719	無限大
Total	6,003,544	21,015	-

註：原始影像大小為 9,465,711 bytes。

表 5. 航照影像，各層重建比較 (不含 EZW 演算法)

各層級	容量大小 (byte)	重建時間 (ms)	PSNR (RGB 平均)
1	12,866	2,437	34.492
2	5,054	2,281	35.063
3	18,983	2,250	35.835
4	71,746	2,297	36.898
5	277,343	2,360	38.345
6	1,090,213	2,750	40.303
7	4,275,019	4,375	無限大
Total	5,751,224	18,750	-

註：原始影像大小為 9,465,711bytes。

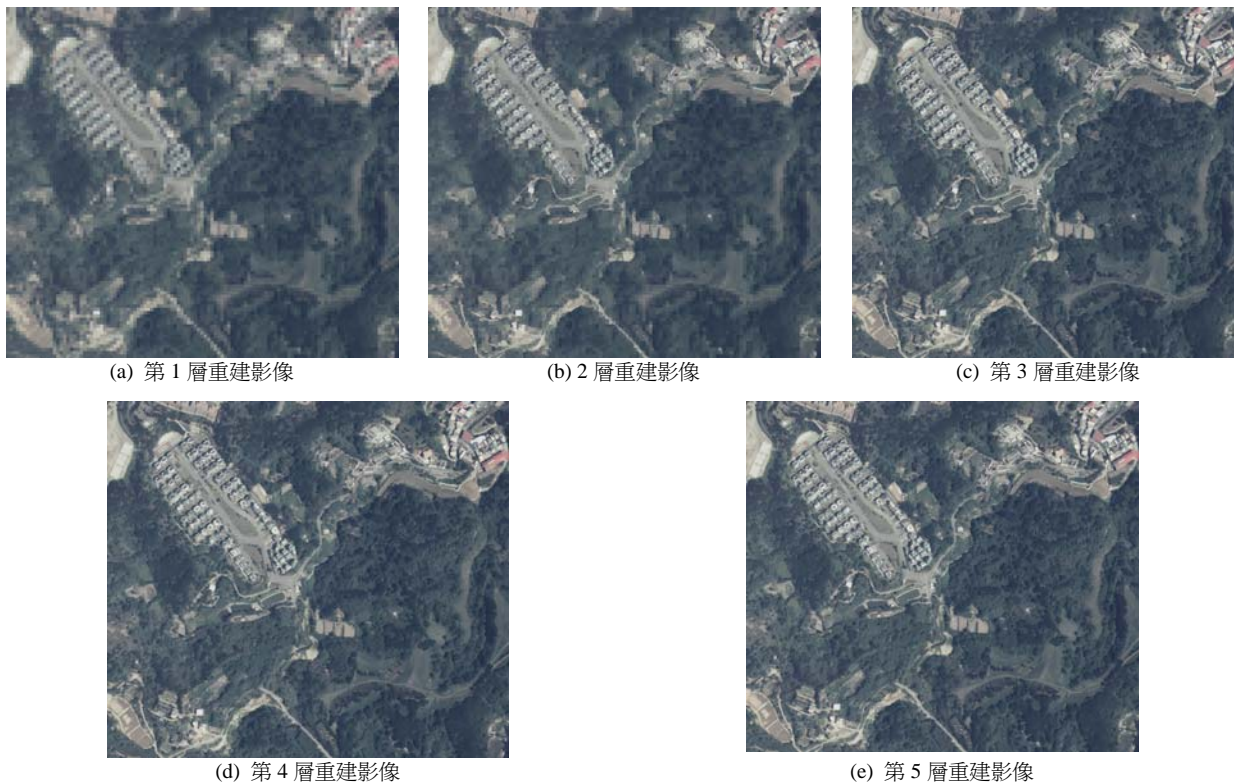


圖 9. 航照影像，EZW 演算法 5 層重建影像

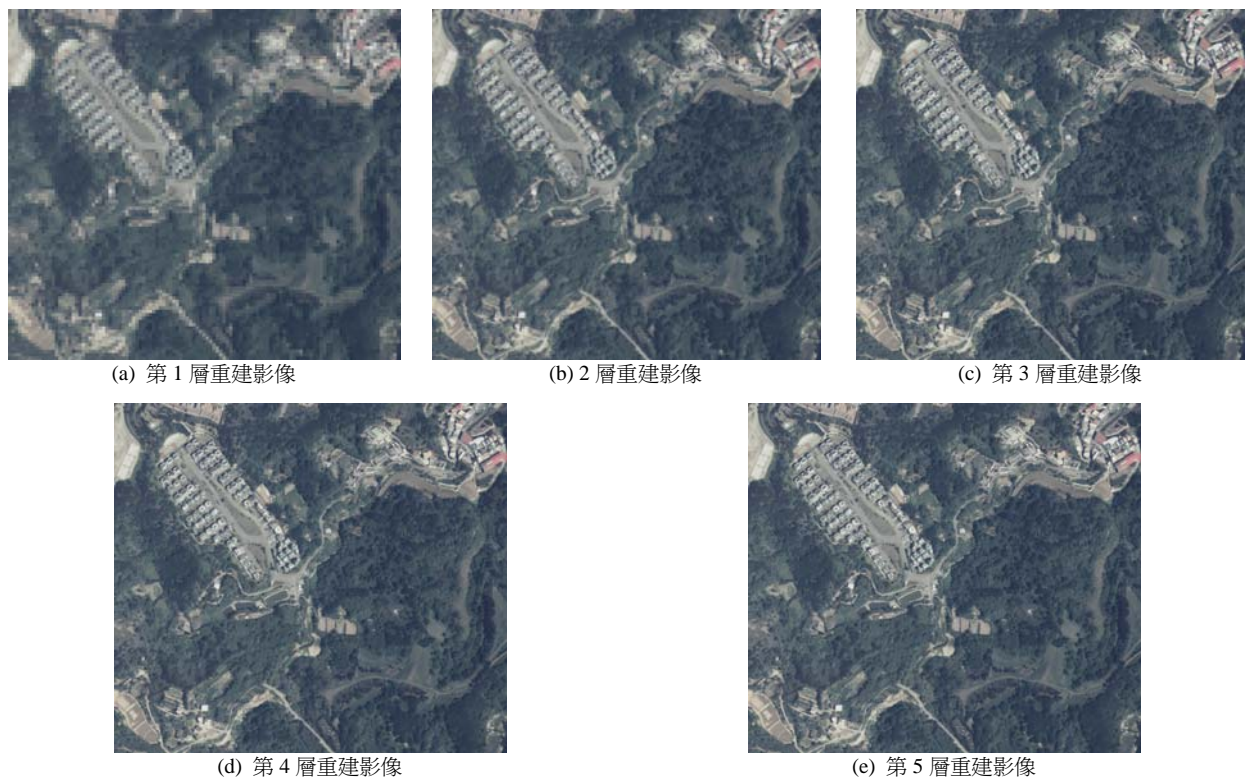


圖 10. 航照影像，EZW 演算法 5 層重建影像

表 6. 航照影像，合併各層重建比較 (EZW 演算法)

各層級	容量大小 (byte)	重建時間 (ms)	PSNR (RGB 平均)
1	35,735	2,610	35.859
2	87,339	2,797	36.939
3	1,584,608	3,688	40.329
4	1,661,379	3,563	41.807
5	2,642,153	3,578	無限大
Total	6,011,196	16,236	-

註：原始影像大小為 9,465,711 bytes。

表 7. 航照影像，合併各層重建比較 (不含 EZW 演算法)

各層級	容量大小 (byte)	重建時間 (ms)	PSNR (RGB 平均)
1	35,402	2,219	35.858
2	71,746	2,203	36.935
3	277,343	2,344	38.415
4	1,090,213	2,705	40.474
5	4,275,019	4,218	無限大
Total	5,749,723	13,689	-

註：原始影像大小為 9,465,711 bytes。

五、結論

本文實現一個影像漸進傳輸伺服器及客戶端系統，所開發的成果軟體，將立即投入商業應用，因此本文程式設計採用目前較普遍使用的 Java 程式語言，Java 除了在網際網路上有非常豐富的資源，物件導向的特性使系統易於維護。

從研究發現解決色彩顯示偏差的問題須先對 RGB 做小波係數轉換再執行 YIQ 色彩轉換，而不是直接對 RGB 像素執行 YIQ 色彩轉換再做小波係數；另外 EZW 演算法在各層重建影像不管在視覺效果或 PSNR 仍然比不含 EZW 演算法都有較好的結果；而 EZW 每次需要對重要係數執行再次精煉，且每個重要係數用 1 位元去紀錄精煉值，這會使得切割次數愈多次累贅愈多，如果能改善每次精煉方式將會使得累贅減少，所得到的各層壓縮比將會提高。最後所剩下來的差值，僅使用字典模式壓縮效果不佳，影響漸進傳輸總傳送量；未來會針對差值做更有效率的壓縮，以提升漸進傳輸的效率及壓縮倍率。

誌謝

本研究承蒙行政院國家科學委員會提供專題研究經費

(計畫編號：NSC 97-2622-E-212 -005 -CC3)，及光特資訊科技股份有限公司提供測試航照影像，並在軟體技術上提供支援及配合，謹此致謝。

參考文獻

1. 蔡文輝、林家禎、張真誠 (民 96)，資料壓縮技術與應用 (初版)，全華圖書股份有限公司，台北。
2. 戴顯權 (民 91)，資料壓縮 (第二版)，紳藍出版社，高雄。
3. Al-Asmari, A. Kh. and A. S. Ahmed (1998) A low bit rate hybrid coding scheme for progressive image transmission. *IEEE Transactions on Consumer Electronics*, 44(1), 226-234.
4. Chang, C. C., Y. C. Li and C. H. Lin (2008) A novel method for progressive image transmission using blocked wavelets. *AEUE-International Journal of Electronics and Communications*, 62(2), 159-162.
5. Chung, K. L. and S. Y. Tseng (2001) New progressive image transmission based on quadtree and shading approach with resolution control. *Pattern Recognition Letters*, 22(14), 1545-1555.
6. Goldberg, M. and L. Wang (1991) Comparative performance of pyramid data structures for progressive image transmission. *IEEE Transactions on Communications*, 39(4), 540-548.
7. Hu, Y. C. and J. H. Jiang (2005) Low-complexity progressive image transmission scheme based on quadtree segmentation. *Real-Time Imaging*, 11(1), 59-70.
8. Kim, J. H. and W. J. Song (1996) Pyramid-structured progressive image transmission using quantisation error delivery in transform domains. *IEE Proceedings I Communications, Speech and Vision*, 143(2), 132-136.
9. Mongatti, G., L. Alparone, G. Benelli, S. Baronti, F. Lotti, and A. Casini (1992) Progressive image transmission by content driven Laplacian pyramid encoding. *IEE Proceedings I Communications, Speech and Vision*, 139(5), 495-500.
10. Oller, J. V., R. J. Villanueva and S. Díez (2007) CASANDRA: A prototype implementation of a system of network progressive transmission of medical digital images. *Computer Methods and Programs in Biomedicine*, 85(2), 152-164.
11. Qi, X. and J. M. Tyler (2005) A progressive transmission capable diagnostically lossless compression scheme for 3D medical image sets. *Information Sciences*, 175(3), 217-243.
12. Said, A. and W. A. Perlman (1996) A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technologies*, 6(3), 243-250.
13. Shapiro, T. M. (1993) Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal processing*, 41(12), 3445-3462.
14. Voukelatos, S. P. and J. J. Soraghan (1998) A multiresolution adaptive VQ based still image codec with application to progressive image transmission. *Signal Processing: Image Communication*, 13(2), 135-143.
15. Wang, L. and M. Goldberg (1988) Lossless progressive image transmission by residual error vector quantization. *IEE Proceedings F Communications, Radar and Signal Processing*, 135(5), 421-430.
16. Wang, L. and M. Goldberg (1989) Progressive image transmission using vector quantization on images in pyramid form. *IEEE Transactions on communications*, 37(12), 1339-1349.
17. Zeng, B., M. S. Fu and C. C. Chuang (2001) New interleaved hierarchical interpolation with median-based interpolators for progressive image transmission. *Signal Processing*, 81(2), 431-438.

收件：98.09.07 修正：98.12.22 接受：99.03.02