

# 頻率域影像轉換之分析與比較

董俊良<sup>1</sup> 王偉華<sup>2</sup> 廖琇怡<sup>1</sup> 王贊鑽<sup>1</sup>

<sup>1</sup> 國立勤益技術學院 資訊管理系

<sup>2</sup> 東海大學 工業工程與經營資訊學系

E-mail : ctung@chinyi.ncit.edu.tw

## 摘要

現今，影像的轉換與壓縮的技術愈來愈精進，基於電腦與網際網路傳輸的發達，這方面的技術研究著重於影像的資料大小與符合需求的失真程度。影像壓縮的主要目的在於如何將影像的資料量縮小，且在傳輸時間上縮短。數位化的影像資料可分為空間域 ( spatial domain ) 影像資料與頻率域 ( frequency domain ) 影像資料，空間域的影像藉由頻率轉換方法轉換成頻率域的影像資料，頻率影像轉換方法有很多種，例如：離散餘弦轉換 ( Discrete Cosine Transformation, DCT )、哈達碼轉換 ( Hadamard Transform, HT ) 和離散小波轉換 ( Discrete Wavelet Transform, DWT )，本文之研究在比較此三種頻率影像轉換法在做影像轉換時的轉換速度，讓研究者可以在不同的影像中，選擇較適當的轉換法，來縮短影像的轉換時間，並給予壓縮。本文針對離散小波轉換之轉換法，以 MATLAB 程式撰寫完整的程式碼，讓研究者在將來設計此轉換法的程式時，可以做參考與比較其差異。

關鍵詞：空間域、頻率域、離散餘弦轉換、哈達碼轉換、離散小波轉換

## 一、前言

網路上資訊流通的速度與資料量，隨著網際網路的盛行而日漸的暴增，在現今追求速度快、品質效果佳的網路資料傳輸中，針對於影像資料的處理，便有所謂的影像壓縮技術的產生，在影像資料的儲存方式中，可分為空間域影像資料與頻率域影像資料，而頻率域的影像資料中，因其資料內容會呈現高低頻帶的分佈，且不同頻帶具有不同的能量，故以此特性，我們常針對以頻率轉換法轉換後的重要影像資料做壓縮。

在做影像資料的壓縮前需經過一個影像轉換的步驟，即先將空間域的影像資料藉由頻率域的轉換方法，將其轉換為頻率域的影像資料，再針對於頻率域影像資料中的重要係數做壓縮動作，以求得最佳的壓縮效果。

本文在影像轉換這部分，為了使研究者可以更快地適當抉擇要使用哪一種轉換法，以縮短轉換時間。本文除了針對常見的頻率域轉換法，如：離散小波轉換 [5]、離散餘弦轉換、哈達碼轉換等給予介紹外，並以七張不同的影像 (大小皆為 256×256)，實驗比較此三種轉換法在影像中的轉換速度。本文針對離散小波轉換之轉換法，以 MATLAB 程

式[3]撰寫完整的程式碼，讓研究者可以與其他轉換法參考比較。

本篇文章分為六個章節，第一章為前言，第二章為影像轉換與壓縮，第三章為離散餘弦轉換與哈達碼轉換，第四章為離散小波轉換，第五章為實驗結果，第六章為結論。

## 二、影像轉換與壓縮

在我們平常瀏覽網頁時所看到的圖檔，或使用影像編輯軟體所處理的圖片，以及掃描器所掃描出來的圖檔，這些都是數位影像的資料型態，由於網路資料傳輸的發達，加上數位影像被廣泛的運用，使得數位影像處理技術受到相當的重視。

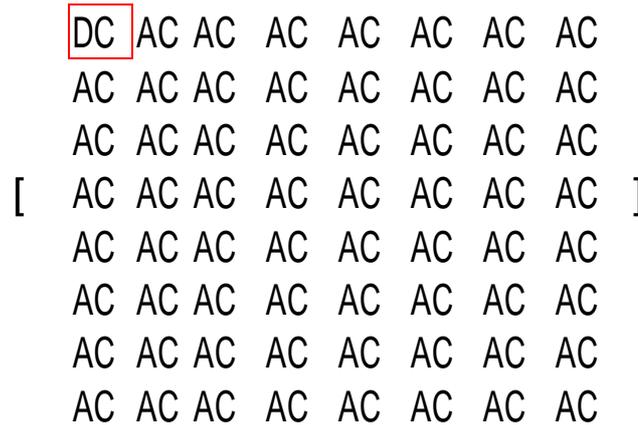
在追求速度快、傳輸品質佳的網路傳輸中，影像資料的傳送品質則藉由現今廣泛在研究的影像轉換與壓縮的技術來加以改善。在影像轉換技術中，常見的空間域轉頻率域的方法為離散餘弦轉換、哈達碼轉換和離散小波轉換，這些轉換法，可以將空間域的影像資料，轉換成頻率域的影像資料，因頻率域的影像資料內容會呈現出高、低頻帶的分佈，而人類的肉眼對於影像中低頻的部分較為敏感，只要在低頻的地方，稍做資料的變更，人類的眼睛必能察覺出來，因此我們通常說影像重要的部分在低頻的地方，利用此特性，我們可以将資料低頻的地方給予壓縮，達到快速傳輸資料的目的。通常在各頻率域轉換法中各有適當的壓縮方法配合運用，例如：離散餘弦轉換、哈達碼轉換則是使用向量量化編碼（Vector Quantization, VQ）壓縮，而離散小波轉換則與 SPIHT（Set Partitioning in Hierarchical Trees）或 EZW（Embedded Zerotree Wavelet）壓縮法配合使用。[1,2]

## 三、離散餘弦轉換與哈達碼轉換

### 3.1 離散餘弦轉換

離散餘弦轉（Discrete Cosine Transformation, DCT）的目的就在於頻率的轉換，在頻率轉換的過程中，影像資料會呈現能量高低分佈的情形。DCT 屬於離散傅利葉轉換(DFT)的一員，且較 DFT 具有轉換係數為實數，能量分佈更集中的優點，因而現在廣泛地應用在許多影像壓縮標準之上。

影像經 DCT 公式運算後，會將影像轉換成頻率的資料型態，在左上角低頻資料量集中形成直流係數(DC)，在剩餘的區域高頻資料量集中形成交流係數(AC)如圖一所示，由於人類視覺不易察覺交流係數的誤差，因此大部分的壓縮法多對此區資料量做壓縮，而人類視覺對直流係數的變動相當敏感，直流係數只要稍有變動一眼就可看出其差異的地方，因此此區資料量通常不做壓縮，只對交流係數做壓縮。[4]



圖一 DC 表示直流係數，AC 表示交流係數

以離散餘弦轉換將影像資料由空間域轉成頻率域，這步驟稱為離散餘弦正轉換 (Forward Discrete Cosine Transformation, FDCT)，反之則稱為離散餘弦反轉換 (Inverse Discrete Cosine Transformation, IDCT)。影像之轉換，可透過 FDCT 與 IDCT 兩種轉換公式做轉換，其公式如下：

$$C(u, v) = a(u)a(v) \sum_{x=1}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y-1)v\pi}{2N}\right] \quad (1)$$

其中  $u, v=0, 1, 2, \dots, N-1$ ， $C(u, v)$  表示轉換後每一係數。

$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} a(u)a(v)C(u, v) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y-1)v\pi}{2N}\right] \quad (2)$$

其中  $x, y, u, v=0, 1, 2, \dots, N$ ， $a$  定義如下：

$$a(u) = \begin{cases} \sqrt{\frac{1}{N}} & u = 0 \\ \frac{1}{\sqrt{N}} & u = 1, 2, \dots, N \end{cases} \quad (3)$$

公式(1)為 FDCT，公式(2)為 IDCT，其中公式(1)與公式(2)的  $(u, v)$  指頻率域之係數位置。 $(x, y)$  指空間域之係數， $N$  表示二維陣列的寬度與長度。[2]

### 3.2 哈達碼轉換

哈達碼轉換 (Hadamard Transform, HT) 為頻率轉換方法中的一種，主要特徵為其核心陣列的元素是由 +1 與 -1 所組成，藉由這個特性，相當適合於實作的積體電路上。

哈達碼轉換是一種非常有效率的影像轉換演算法 [4]，因為它的核心陣列僅包含 +1 或 -1 元素，大小為  $N \times N$ ， $N = 2^n$ ，以  $N = 8$  為例，哈達碼核心陣列可表示為圖二所示。

$$H_8 = \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

圖二 8×8 大小的哈達碼核心陣列

二維哈達碼轉換定義如下：

$$H(x_1, y_1) = \frac{1}{\sqrt{N}} \sum_{x_2=0}^{N-1} \sum_{y_2=0}^{N-1} f(x_2, y_2) (-1)^{\sum_{i=0}^{n-1} [b_i(x_2)q_i(x_1) + b_i(y_2)q_i(y_1)]} \quad (4)$$

其中， $f(x_2, y_2)$  代表  $N \times N$  大小的影像

$H(x_1, y_1)$  為  $f(x_2, y_2)$  的頻率係數

$b_i(D)$  為以二進位代表整數  $D$  時第  $i$  位碼的值

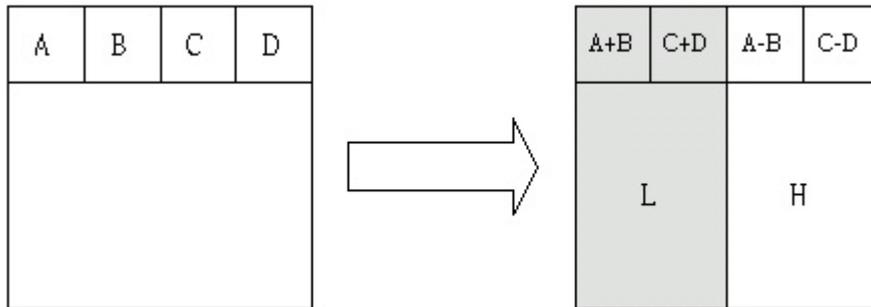
二維哈達碼逆轉換定義如下：

$$f(x_2, y_2) = \frac{1}{\sqrt{N}} \sum_{x_1=0}^{N-1} \sum_{y_1=0}^{N-1} H(x_1, y_1) (-1)^{\sum_{i=0}^{n-1} [b_i(x_2)q_i(x_1) + b_i(y_2)q_i(y_1)]} \quad (5)$$

### 四、離散小波轉換

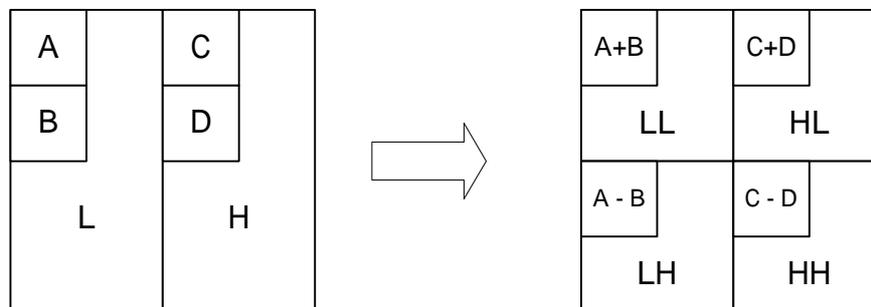
離散小波轉換 ( Discrete Wavelet Transform, DWT ) [6] 是一種將空間域影像 ( spatial domain ) 轉換成頻率域影像 ( frequency domain ) 的技術 [7, 8]。將空間域影像轉換成頻率影像的好處在於轉換後的資料會有其重要性的差異存在，而我們可藉由這種特性，對於較重要部份的影像加以壓縮。離散小波的轉換方式有很多種，而以 Haar 函數的離散小波轉換方式是為較簡單的做法。Haar 函數所使用的離散小波轉換方式是將影像的所有像素分別視為各自獨立的數值，並對這些數值做相加、相減的運算，以求得這張影像的頻率 [2]。而因為相加的數值會越來越大，且人類的視覺對這部份的敏感度較高，所以相加的地方視為低頻的部分，相反的，相減的地方則為高頻的部分。

Haar 函數之離散小波轉換，其正轉換的運算大致上分兩個步驟：(1)水平分割，(2)垂直分割，水平分割的意思是讀取係數的順序是依照水平方向由左至右來取，儲存時也是以水平方向儲存。而垂直分割的意思是讀取係數的順序是垂直方向由上至下來取；儲存時也是垂直方向儲存。[2]首先，以水平分割的步驟而言，假設，現有一張大小為  $4 \times 4$  的灰階影像，現在以水平方向由左至右的順序，取出空間域影像的相鄰兩像素，作相加、相減的運算，然後再將結果分別作儲存。L 區塊為低頻的部分，其餘為高頻的部分，如圖三所示。



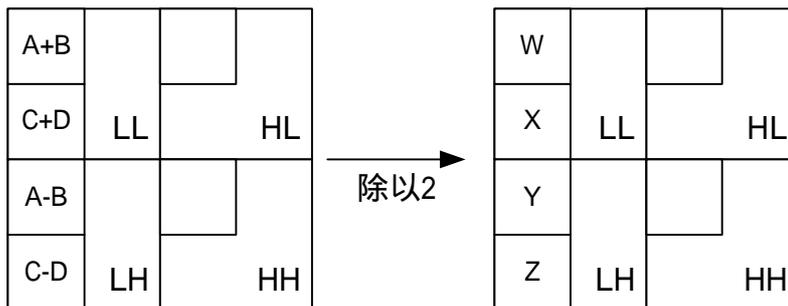
圖三 離散小波轉換之水平分割步驟

將水平分割後的結果，再來做垂直分割的動作。依垂直的方向，由上至下取出係數做相加、相減運算，且將結果做儲存，如圖四所示。



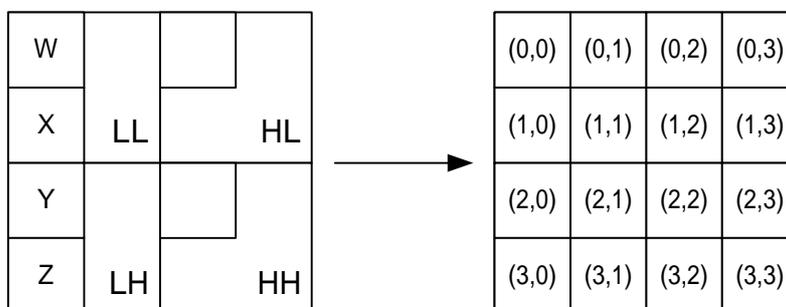
圖四 離散小波轉換之垂直分割步驟

當一張空間影像轉換成頻率影像後，若要再轉換回原來的空間影像，則也有其轉換步驟，以下為離散小波逆轉換的介紹。一開始因為是先由水平分割，再做垂直分割的正轉換步驟，所以當做逆轉換步驟時，先從垂直分割做起。首先是將垂直分割後的結果每一個值除以 2，而得到各個的值，以  $W, X, Y, Z...$  來標示。 $W$  表示  $(A+B)/2$  的值， $X$  表示  $(C+D)/2$ ， $Y$  表示  $(A-B)/2$ ， $Z$  表示  $(C-D)/2$  的值，依序類推，如圖五所示。



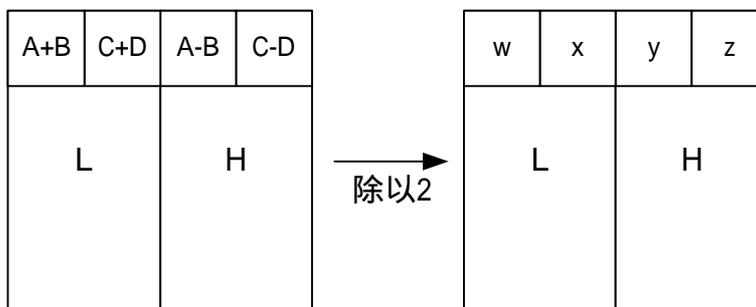
圖五 離散小波逆轉換之垂直逆轉步驟

接著再將  $W+Y$  的值置於  $(0,0)$  的位置，將  $W-Y$  的值置於  $(1,0)$  的位置， $X+Z$  的值置於  $(2,0)$  的位置， $X-Z$  的值置於  $(3,0)$  的位置，如圖六所示。



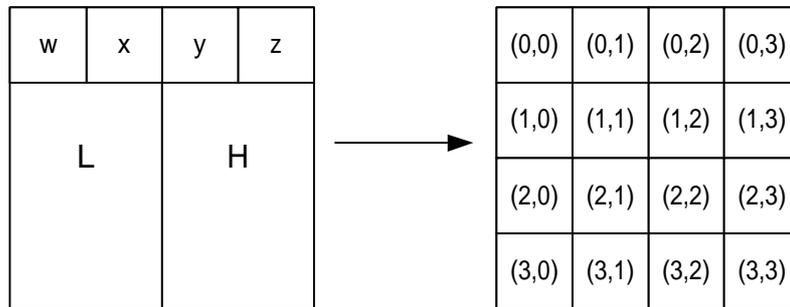
圖六 離散小波逆轉換之垂直逆轉步驟

將圖<六>垂直逆轉換的結果，再做水平逆轉換的動作，因垂直逆轉換的結果為水平分割後的結果，所以將水平分割後的各值皆除以 2，而得到各個值以  $w, x, y, z \dots$  標示。 $w$  表示  $(A+B)/2$  的值， $x$  表示  $(C+D)/2$  的值，如圖七所示。



圖七 離散小波逆轉換之水平逆轉步驟

接著將  $w+y$  的值置於  $(0,0)$  的位置，將  $w-y$  的值置於  $(0,1)$  的位置，將  $x+z$  的值置於  $(0,2)$  的位置，將  $x-z$  的值置於  $(0,3)$  的位置。重複此步驟，即可以得到一張原來的影像，如圖八所示。



圖八 離散小波逆轉換之水平逆轉步驟

### 五、實驗結果

本文所使用的實驗平台為 Pentium 4 1.8GHz 處理器、128MB 記憶體、作業系統為 Windows 2000 Professional、程式撰寫工具為 MATLAB 5.1。本實驗選出七張不同的影像，分別以離散小波轉換、離散餘弦轉換、哈達碼轉換等三種轉換法加以作轉換。一般 SPIHT 壓縮法，通常以離散小波轉換後的影像資料來做資料的壓縮，且使用三階離散小波轉換後的資料，配合此壓縮法，即可獲得的 PSNR 值約為 25 左右的影像還原品質，而在離散餘弦轉換與哈達碼轉換中，通常以  $4 \times 4$ 、 $8 \times 8$  大小之訓練樣本做轉換，並配合 VQ 之壓縮法，便可得到 PSNR 值為 29 左右的影像還原品質，所以在本文中便只做三階離散小波轉換之測試，而離散餘弦轉換與哈達碼轉換，除了用一般以  $4 \times 4$ 、 $8 \times 8$  大小之訓練樣本做測試轉換外，本文還測試以  $2 \times 2$  大小的訓練樣本做轉換，提供研究人員做參考。本實驗所測試的七張灰階影像，其影像大小為  $256 \times 256$ ，如表一所示，並將此三種轉換之正、逆轉換後的時間，以表格列示，如表二、表三、表四所示，使研究人員能清楚比較各轉換時間的差別。

表一 本實驗所使用之圖片

Baboon	Barbala	Boy & Girl	F16	Girl	Lena	Pepper
						

表二 DCT、HWHT 區塊大小為  $2 \times 2$  與三階 DWT 之正、逆轉換時間比較

image	Baboon		Barbala		Boy & Girl		F16		Girl		Lena		Pepper	
轉換方法	$T_1$	$T_2$	$T_1$	$T_2$	$T_1$	$T_2$	$T_1$	$T_2$	$T_1$	$T_2$	$T_1$	$T_2$	$T_1$	$T_2$
DWT	70	80	60	80	60	90	60	80	70	90	50	90	70	70
DCT	1180	1270	1150	1280	1190	1270	1190	1250	1170	1230	1150	1250	1170	1270
HWHT	1180	1230	1170	1270	1170	1230	1170	1250	1190	1250	1190	1250	1160	1280

$T_1$  : 正轉換時間 (ms) ;  $T_2$  : 逆轉換時間 (ms)

表三 DCT、HWHT 區塊大小為  $4 \times 4$  與三階 DWT 之正、逆轉換時間比較

image	Baboon		Barbala		Boy & Girl		F16		Girl		Lena		Pepper	
轉換方法	$T_1$	$T_2$	$T_1$	$T_2$	$T_1$	$T_2$	$T_1$	$T_2$	$T_1$	$T_2$	$T_1$	$T_2$	$T_1$	$T_2$
DWT	70	80	60	80	60	90	60	80	70	90	50	90	70	70
DCT	390	400	370	410	390	390	390	400	370	410	380	390	390	410
HWHT	410	400	410	400	400	420	390	410	390	390	400	390	390	390

$T_1$  : 正轉換時間 (ms) ;  $T_2$  : 逆轉換時間 (ms)

表四 DCT、HWHT 區塊大小為  $8 \times 8$  與三階 DWT 之正、逆轉換時間比較

image	Baboon		Barbala		Boy & Girl		F16		Girl		Lena		Pepper	
轉換方法	$T_1$	$T_2$	$T_1$	$T_2$	$T_1$	$T_2$	$T_1$	$T_2$	$T_1$	$T_2$	$T_1$	$T_2$	$T_1$	$T_2$
DWT	70	80	60	80	60	90	60	80	70	90	50	90	70	70
DCT	230	160	230	140	230	160	250	150	70	90	50	90	70	70
HWHT	370	160	390	160	370	160	390	160	240	140	240	150	230	160

$T_1$  : 正轉換時間 (ms) ;  $T_2$  : 逆轉換時間 (ms)

本實驗主要是在測試目前常見的頻率域轉換的轉換速度，其實驗結果如表二、表三、表四所示，可以看出離散小波轉換的轉換速度比離散餘弦轉換、哈達碼轉換快，而離散餘弦轉換與哈達碼轉換的轉換速度則差不多，但哈達碼轉換因其核心為+1 與-1 的特性，較容易實作於實體上，使它在此方面有優於離散餘弦轉的趨勢。

若基於不將資料壓縮的情況下，對傳送影像資料而言，離散小波轉換有一個優點，它可以利用 scan order 的方式來傳送資料[2]，因為此方式的影像資料傳送過程，是先將低頻的資料做傳送，而離散小波轉換後的主要影像集中在低頻的部份，可以看出影像的大致輪廓，其餘高頻的部分是加強影像的解析度，倘若資料在傳輸過程中，因不可預測的原因，而發生資料傳輸中斷的情況，基本上還可以看到影像的原始輪廓，而接收端則可以考慮是否要再請傳送端再做一次影像傳輸的動作。

## 六、結論

網路資訊傳輸的發達，在講求傳輸速度快，且傳輸品質佳的情況下，許多專家學者便投入如何讓資料量縮小且傳輸品質佳的研究。在網路上常呈現的資料，除了文字、聲音外，最常見的便是影像圖檔。為了讓這些影像圖檔能夠減少其儲存的空間以利增加傳輸的速度，且又為了能夠讓傳輸品質在最佳的狀態下，對於影像轉換與壓縮這方面的研究領域，愈來愈多人去研討改進的方法。由本篇論文可看出，離散小波轉換具有許多的特性，除了轉換速度快、程式容易撰寫的優點外，更是適合漸進式影像傳送。將影像透過離散小波轉換，並利用 SPIHT 壓縮，可將影像資料在網路系統中快速傳遞。

## 參考文獻

- [1] 張真誠、黃國峰、陳同孝，*電子影像技術*，松崗電腦圖畫資料股份有限公司，2000
- [2] 陳同孝、張真誠、黃國峰，*數位影像處理技術*，松崗電腦圖畫資料股份有限公司，2001
- [3] 張智星，*MATLAB 程式設計與應用*，清蔚科技股份有限公司出版事業部，2000
- [4] 董俊良等人，"植基於頻率轉換之影像壓縮整合平台"，*勤益學報*，第十九期，第173-183頁，2001
- [5] N.Ahamd, E.Oja, and K.R.Rao, "Discrete Cosine Transform," *IEEE Trans.comput*, Vol.C-23, pp.90-93, 1974
- [6] M.Antonono, M.Barlaud, P.Mathieu, and I. Daubechies, 1992 "Image Coding UsingWavelet Transform," *IEEE Trans.Image Processing*, Vol.1, pp.205-220
- [7] R.A.Devore, B.Jawerth, and B.J.Lucier, 1992, "Image Compression Through Wavelet Transform Coding," *IEEE Trans.Informat.Theory*, Vol.38, pp.719-146
- [8] J.Hadamard, "Resolution d'une Question Relative aux Determinants," *Bull.Sci.Math.*, Ser.2, 17, Part 1, pp.240-246, 1893

## 附錄—離散餘弦轉換法之程式說明

### 正轉換

```

%掃描 Main figure 所有物件
fig=1;ipsysobject;
%重新讀取選項設定值
load options;
%取得選取之影像名稱
choice=get(MfPopupMenu1,'Value'); %取得 popmenu1 中選項值 1.2.3....
file_name=image_name{choice}; %由 image_name 陣列中取得影像名稱

%若影像名稱對映之影像檔存在則載入此影像
if ~isempty(which([file_name,'.raw']))
    image_col_size=256; %設定影像之長寬
    image_row_size=256;
    image=readdraw(file_name,image_col_size,image_row_size); %依照影像之長寬載入影像
end

run_time=cputime; %設定執行前時間
result=image; %設定存放分割結果之矩陣大小
level=3; %設定要做幾階轉換
for i=1:level,
    s=size(result)/2;
    result=[result(:,1:2:end)+result(:,2:2:end),result(:,1:2:end)-result(:,2:2:end)]; %水平分割
    result=[result(1:2:end,:)+result(2:2:end,:);result(1:2:end,:)-result(2:2:end,:)]; %垂直分割
    image(1:s(1,1)*2,1:s(1,2)*2)=result; %將轉換後之距陣放回
    result=result(1:s(1,1),1:s(1,2)); %取得最低頻之頻帶
end
time_dwt=(cputime-run_time)*1000 %正轉之時間(單位：ms)
source=image; %source 為 spiht 壓縮之資料來源

```

## 逆轉換

```

image=source;          %source 為 spiht 解壓縮後之結果
image_size=size(image);
level=3;
for i=1:level,
    %垂直分割還原
    s=image_size/2^(level-i+1);    %取得 LLn 矩陣之大小(n 為階數)
    idwt=image(1:s(1,1)*2,1:s(1,2)*2)/2; %取得第 n 階矩陣值/2 之矩陣
    temp=idwt;    %設定暫存矩陣大小
    temp(1:2:end,:)=idwt(1:s(1,1),:)+idwt(s(1,1)+1:end,:); %相加放奇數列
    temp(2:2:end,:)=idwt(1:s(1,1),:)-idwt(s(1,1)+1:end,:); %相減放偶數列
    %水平分割還原
    temp=temp/2; %取得垂直分割還原後矩陣值/2 之矩陣
    idwt(:,1:2:end)=temp(:,1:s(1,1))+temp(:,s(1,1)+1:end); %相加放奇數行
    idwt(:,2:2:end)=temp(:,1:s(1,1))-temp(:,s(1,1)+1:end); %相減放偶數行

    image(1:s(1,1)*2,1:s(1,2)*2)=idwt;    %將第 n 階還原後放回原矩陣
end
run_time2=(cputime-run_time)/60 %經過 dwt 正反轉與 spiht 壓縮、解壓縮所費之時間
MSE=sum(sum((original-image).^2))/256.^2;
PSNR=10*log10((255^2)/MSE)

%image 為反轉後之結果

```

