第五章

基礎數學工具與影像壓縮編碼

課程內容

- 轉換編碼（Transform Coding）簡介
- 離散餘弦轉換(DCT)
- 離散小波轉換(DWT)
- 零樹小波編碼(EZW)
- 階層式集合分割影像壓縮技術(Set Partitioning in Hierarchical Trees , SPIHT)
- 最佳截斷嵌入式碼塊編碼 (Embedded Block Coding with Optimized Truncation ,EBCOT)

5.1 轉換編碼

- 它的做法是將原信號經過一個轉換變成另一種表示法。
- 這個表示法可經由逆轉換恢復成原來的信號，而且它的能量較原信號來得集中，因此比較容易作壓縮。
- 轉換編碼最成功的應用是影像資料壓縮。靜態影像壓縮標準JPEG，就是採用轉換編碼。

- 轉換的目標就是將原信號取樣間之相關性打散，使其訊號的能量重新分佈，而且只集中在少數幾個轉換係數。
- 這麼一來，許多係數在量化後便可以予以忽略而不再作編碼（因爲量化後它們爲零）。
- HVS（Human visual system，人類視覺系統）之對比敏感函數於轉換係數之量化工作上，而達到視覺上無失真的壓縮。

- 影像的轉換編碼一般是將原來*N x N* 的影像先切割成不重疊*n x n* 的方塊，然後在對每個*n x n* 的方塊作單位轉換（unitary transform）。
- 轉換的核心（kernel）可以分解成兩個一維轉換的核心，分別是水平與垂直方向的運算。因此要完成一個二維可分開的的轉換，我們可以分兩個步驟來完成。

- 第一我們將 *n x n* 的方塊中每一個橫列作一維的*n*點轉換，然後所得的結果再爲每一個縱列作一維*n*點轉換。
- 這樣子作出來的結果是跟直接作方塊的二維轉換是相同的。

- 希望轉換具有以下三種特性：

1. 打散方塊內影像內像素間的相關性，也就是說將大部分的能量集中在最少數的基底函數上，使其像素間無關或重複的資料（redundancy）能夠去除。
2. 與影像無關的基底函數：所謂最佳的轉換於影像的統計特性有關，因此最佳轉換通常隨著影像的不同而有異。

找出最佳轉換的基底函數非常耗時，因此，通常會捨棄最佳化的轉換而採用與影像無關的基底函數。

3. 快速完成轉換

於 *n* 點的轉換，所需要的計算量爲 $O(n\log_2 n)$ 的等級。因此*n* x *n* 的二維轉換爲 $O(n^2\log_2 n)$ 的等級。

結論

- 轉換本身並不能達到資料壓縮的目的；它是將原資料打散，且將訊號的大部分能量集中在相當少的轉換項。
- 真正資料壓縮的是轉換後，接著做的量化動作，以及量化後的更進一步編碼。
- 一般壓縮的程度決定於量化的程度；比較粗糙的量化可達到比較高的壓縮比，但是也得到較差的重建品質。
- 一些被採用的影像轉換，包括傅立葉、離散餘弦、小波轉換等。

5.2 離散餘弦轉換（Discrete Cosine Transform）

- DCT 也是離散傅立葉轉換（DFT, Discrete Fourier Transform）的一員。
- DFT 所產生之係數爲複數（包括實部與虛部，或者大小值與相位）
- DCT所產生之係數爲實數值。

- 離散餘弦正轉換（FDCT, Forward Discrete Cosine Transform），就是將空間域數位影像資料轉換成頻率域。
- 反之若將頻率域數位影像資料還原成空間域，則稱爲離散餘弦反轉換（IDCT, Inverse Discrete CosineTransform）。

- 對一個*n x n* 的方塊而言，其二維FDCT 轉換之定義爲：

$$D(i,j)=\frac{1}{\sqrt{2n}}C(i)C(j)\sum_{x=0}^{n-1}\sum_{y=0}^{n-1}f(x,y)\cos\left[\frac{(2x+1)i\pi}{2n}\right]\cos\left[\frac{(2y+1)j\pi}{2n}\right]$$

其中

$$\begin{cases} i=0, & C(i)=1/\sqrt{2} \\ i\neq 0, & C(i)=1 \end{cases}$$

$$\begin{cases} j=0, & C(j)=1/\sqrt{2} \\ j\neq 0, & C(j)=1 \end{cases}$$

- 對一個*n x n* 的方塊而言，其二維IDCT 轉換之定義爲：

$$f(x,y)=\frac{1}{2n}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}C(i)C(j)D(i,j)\cos\left[\frac{(2x+1)i\pi}{2n}\right]\cos\left[\frac{(2y+1)j\pi}{2n}\right]$$

其中

$$\begin{cases} i=0, & C(i)=1/\sqrt{2} \\ i\neq 0, & C(i)=1 \end{cases} \qquad \begin{cases} j=0, & C(j)=1/\sqrt{2} \\ j\neq 0, & C(j)=1 \end{cases}$$

方法一

- 一張灰階影像完整地劃分成 8x8 像素大小的區塊（block）且每一個區塊並不重疊，先將每一個空間域區塊中所有像素值減去128 。
- 減去128 ，主要的目的爲了將原本介於0~255 的灰階像素值，調整成介於-128~127 之値，以利FDCT的處理。
- 以FDCT做轉換，即可得到一個和空間域區塊中像素個數相同的頻率域區塊
- 將此頻率域的資料經由IDCT 轉換後，每個元素值加上128 後，即可還原成原來的數位灰階影像，如圖5.1(a) 所示。

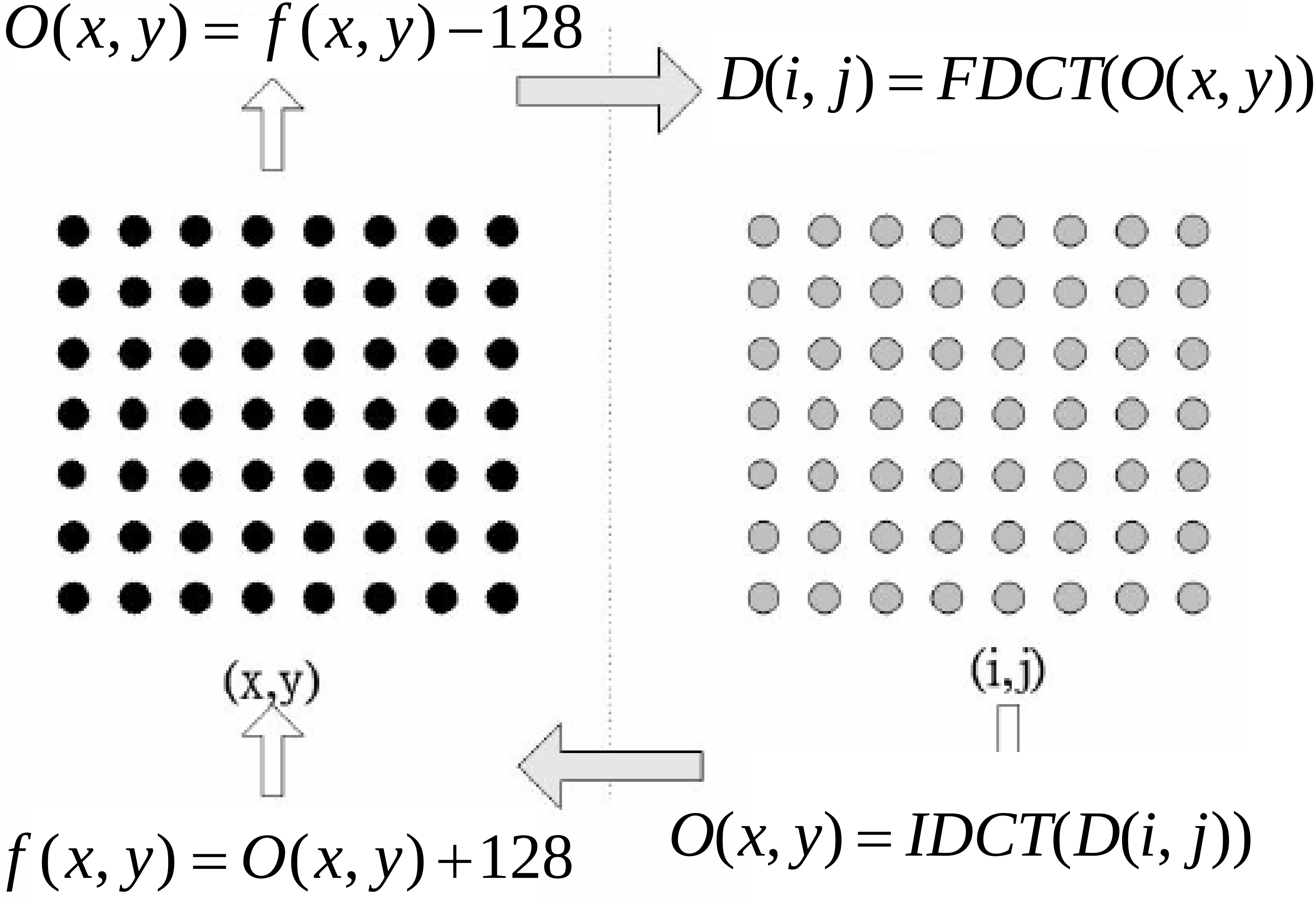


圖5.1 (a)空間域及頻率域影像資料轉換循環示意圖

方法二

- 一張灰階影像完整地劃分成 8x8 像素大小的區塊（block）且每一個區塊並不重疊，先將每一個空間域區塊中所有像素值除以255 。
- 除以255 ，主要的目的爲了將原本介於0~255 的灰階像素值，調整成介於0~1 之值，以利FDCT的處理。
- 以FDCT做轉換，即可得到一個和空間域區塊中像素個數相同的頻率域區塊
- 將此頻率域的資料經由IDCT 轉換後，每個元素值乘128 後，即可還原成原來的數位灰階影像，如圖5.1(b) 所示。

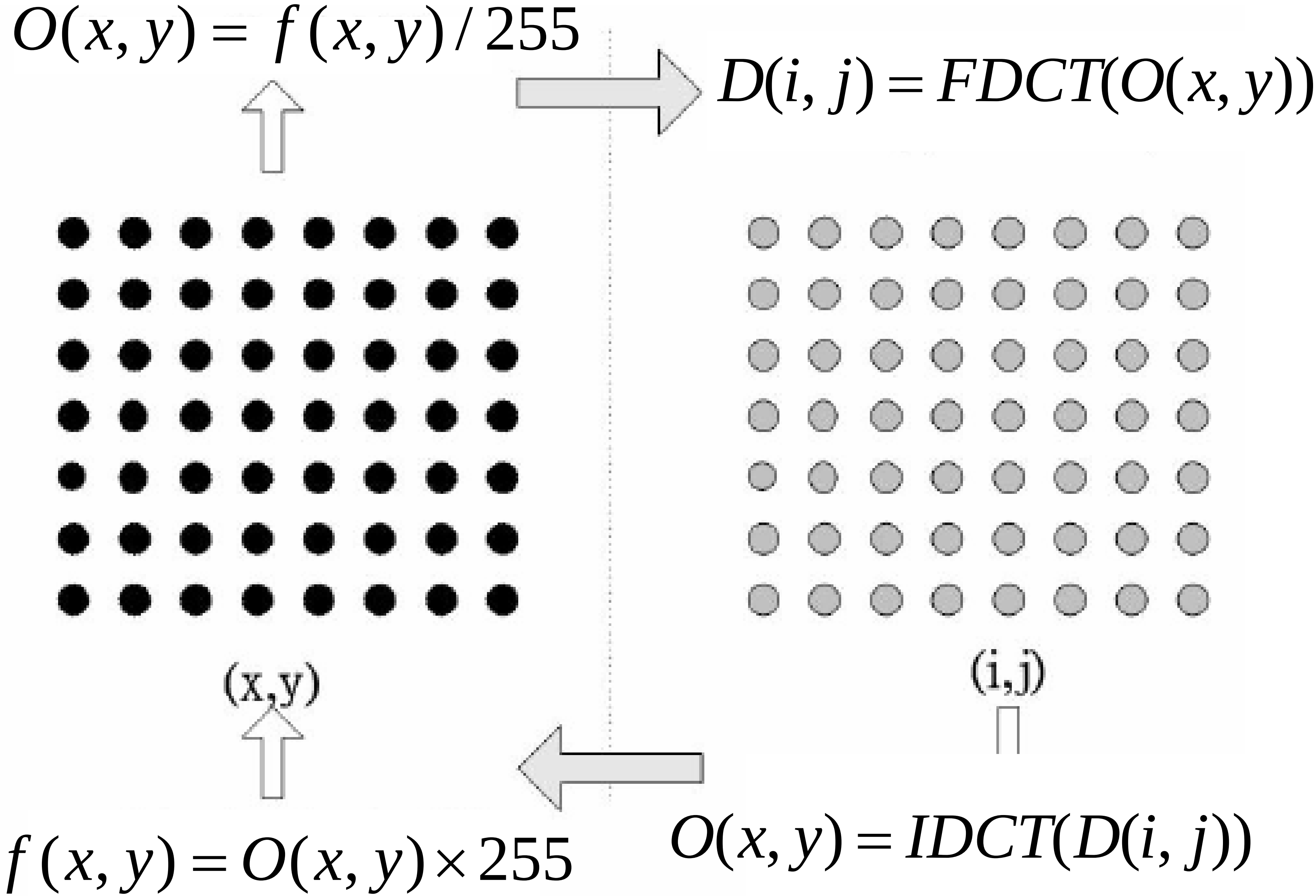
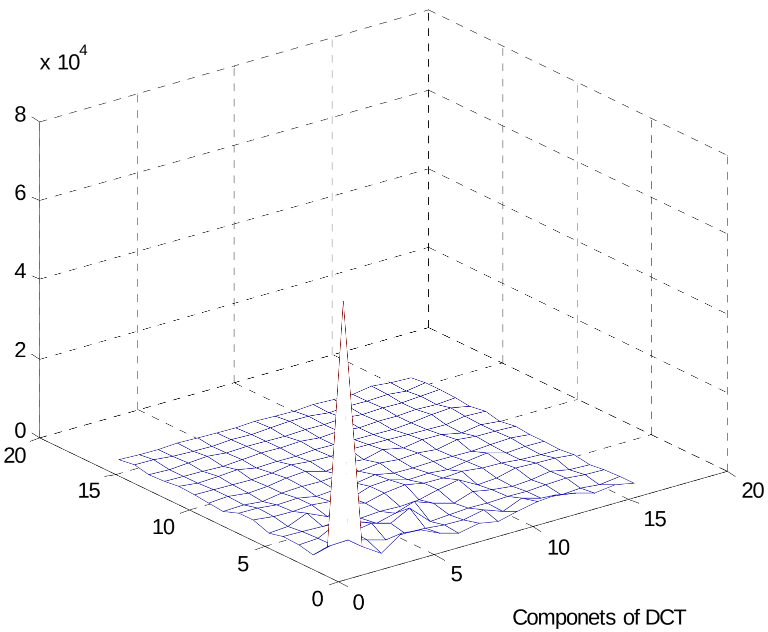


圖5.1(b) 空間域及頻率域影像資料轉換循環示意圖

- $O(x,y)$ 係指 $f(x, y)$ 位置所放的值減去128 後或除以128後的空間域像素值。
- $D(i, j)$ 是指(i, j)位置上的頻率係數值。
- DCT 係數值不一定是正值，也有可能是負值。

Original picture

FDCT



1.0e+004 *

6.8215	-0.4581	-0.1163	0.5009	0.4299	-0.2204
0.3872	0.4218	-0.2714	-0.1309	-0.1876	0.1455
0.0118	0.1179	-0.3138	0.3709	-0.2686	0.4895
-0.0639	-0.1052	0.3461	-0.2470	0.0861	0.1662
-0.1014	0.0978	0.0859	0.1163	-0.0112	0.0595
-0.0244	-0.0286	0.2532	0.0097	-0.2151	-0.0794

The 6x6 coefficients of FDCT

Original picture

Picture of IDCT with method 1

Picture of IDCT with method 2

Original picture

Picture of IDCT with method 1

Picture of IDCT with method 2

Original picture

Picture of IDCT with method 1

Picture of IDCT with method 2

original picture

compression of ratio 1/4

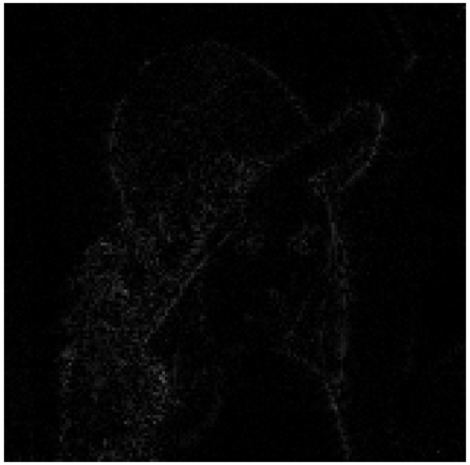
compression of ratio 1/16

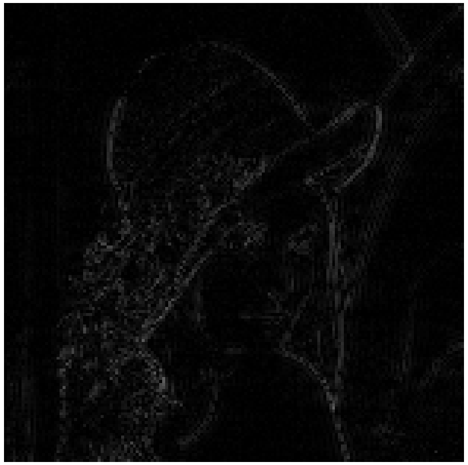
compression of ratio 1/64

MSE =12.2866

original picture

difference of origin and compression of 1/4





difference of origin and compression of 1/16

MSE =53.7164

difference of origin and compression of 1/64

MSE =141.8612

作業4

- 利用MATLAB撰寫程式，將lena的影像(512x512) ，切割大小爲32x32的方塊(不重疊)，使用第一、二方法進行FDCT與IDCT，重建影像。

5.3離散小波轉換（Discrete Wavelet Transform）

- 離散小波轉換與離散餘弦轉換相似，都是一種將空間域影像轉換成頻率域影像的技術。
- 一張影像經過離散小波轉換的處理後會產生重要性不同的資料，這些資料會構成高低不同之頻率。
- 如此一來，我們就可以根據這些資料的不同重要性分別作處理。例如：完整地保存重要性資料，而選擇不重要的地方藏入數位浮水印或其他資料等。

- 有關離散小波轉換方法有很多種，它們提供了一些離散小波的轉換方式來將影像由空間域轉變成頻率域。
- 在這裡，我們介紹較爲簡單的Haar 函數離散小波轉換方式。

- Haar 函數所使用之離散小波轉換使用的方式，是將影像的所有像素分別視爲各自獨立的數值，並對這些數值作相加、相減的運算，以求得這張影像的頻率。
- 其中相加後的值會越來越大，值越大也就表示該值越重要。利用我們的肉眼來看，這個部分會非常的清楚，所以相加的部分就是低頻的部分。
- 相減後的值代表著像素與像素之間的差距，當遇到影像中的物體邊緣時，像素間的差值就會很大，當遇到影像中物體平滑的地方時，則像素間的差值就會很小，所以相減的部分是高頻的部分。

- Haar 函數離散小波轉換的運算大致上有兩個步驟：一爲水平分割，另一爲垂直分割。
- 水平分割是讀取順序是依照水平方向由左至右取出；儲存時也是水平方向儲存。
- 垂直分割是讀取順序是垂直方向由上至下來取；儲存時也是垂直方向儲存。

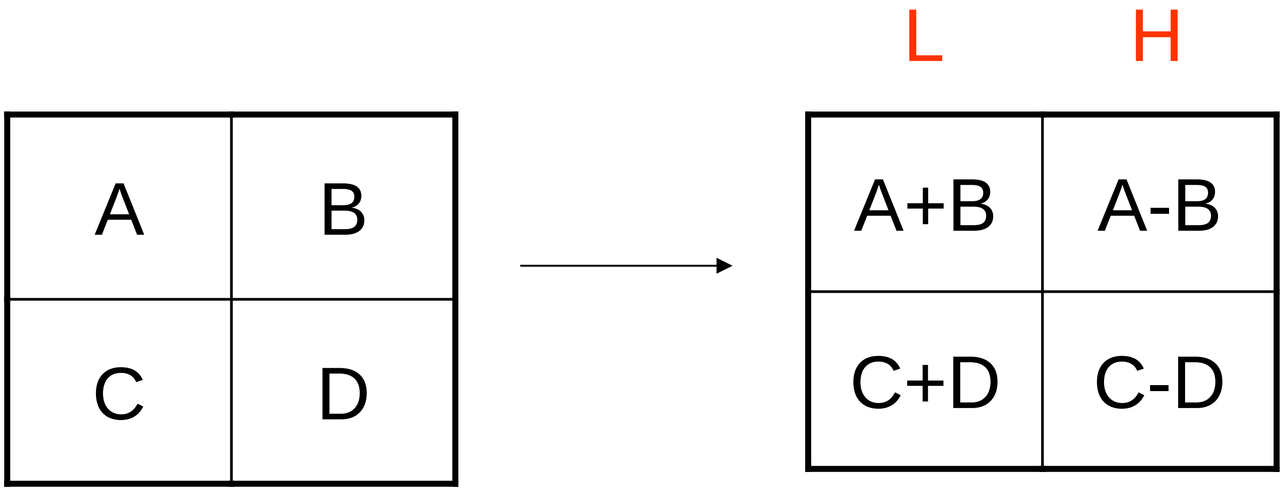


圖5.3 第一次水平分割計算示意圖

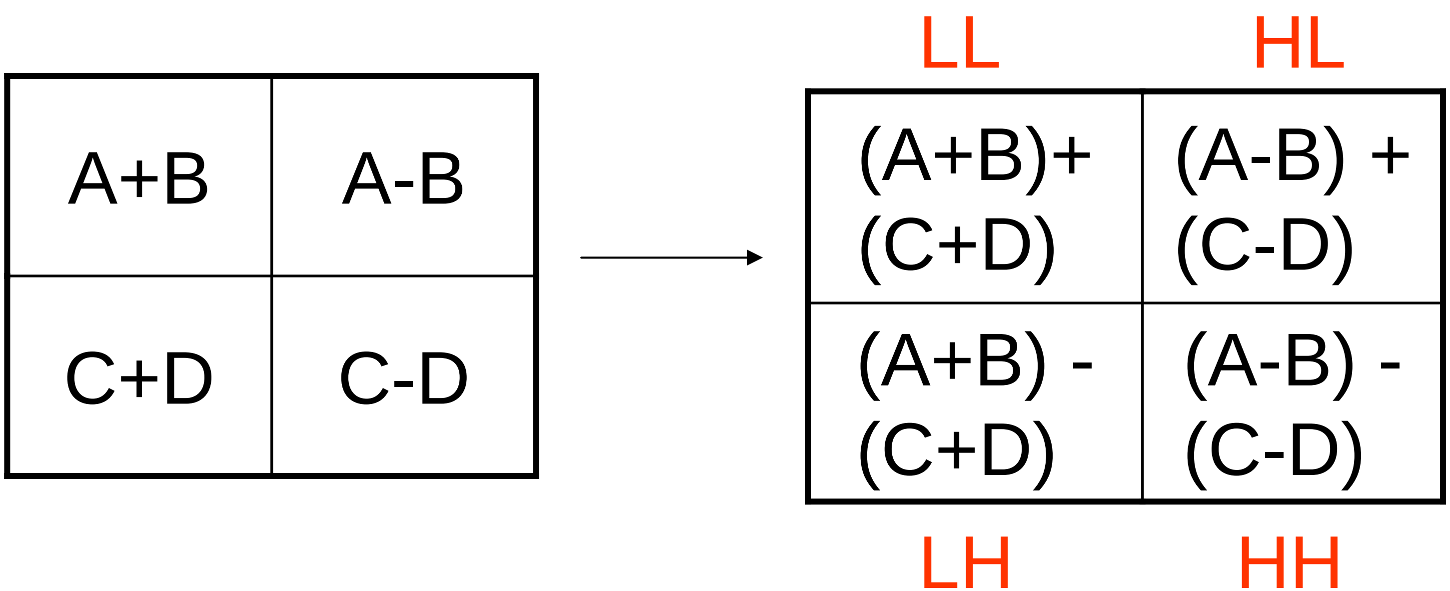


圖5.4 第一次垂直分割示意圖

- 完成了以上第一次水平分割及第一次垂直分割後，這樣算是第一階的離散小波轉換；我們分別得到了不同頻率的區塊LL、LH、HL 及HH；我們稱這些區塊爲頻帶（subband）。
- 因爲LL 這個頻帶是影像中最重要的部分，所以可再對LL 做一階離散小波轉換動作，成爲二階離散小波轉換，當然可一直做下去，直到*n* 階。

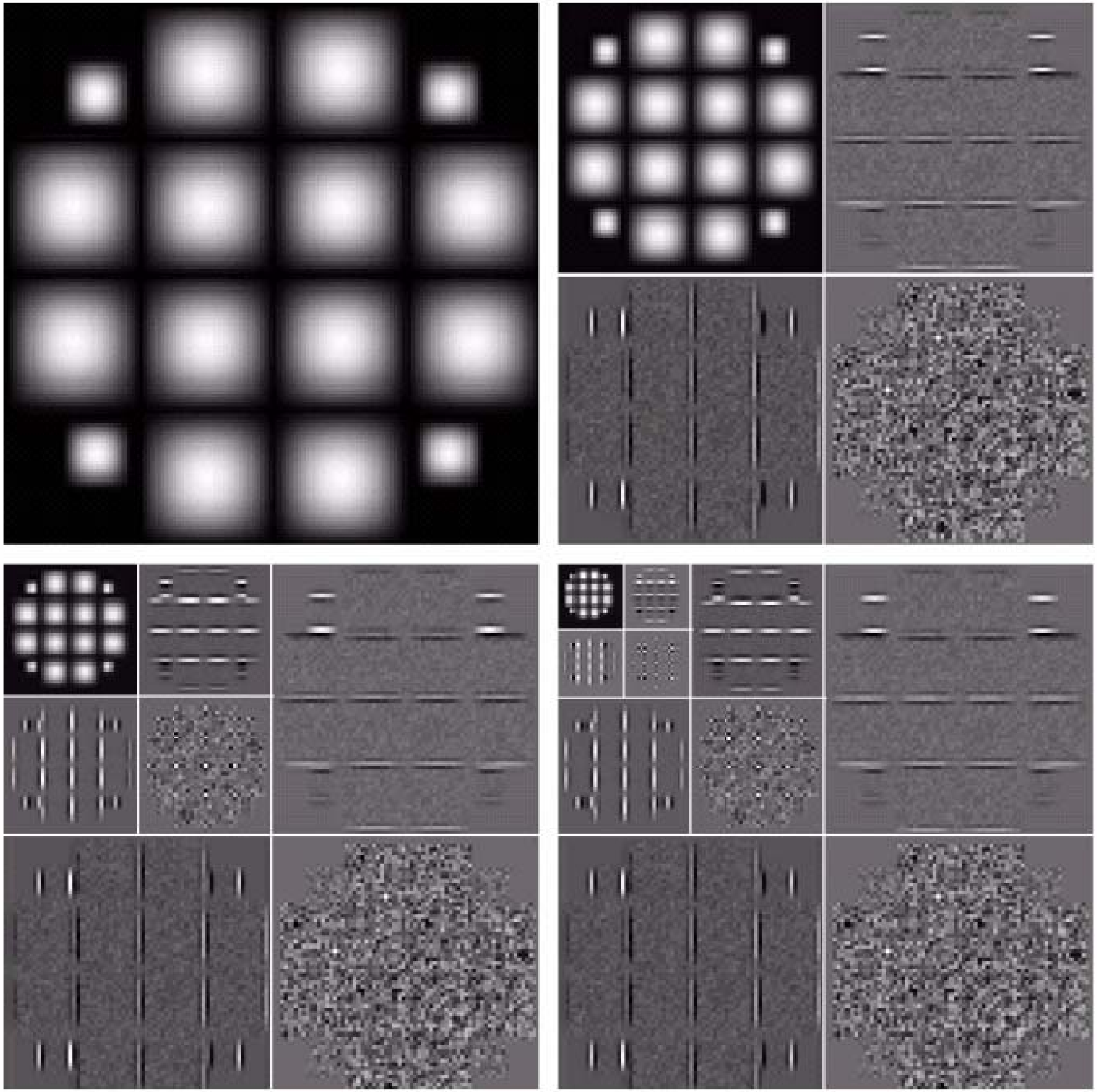
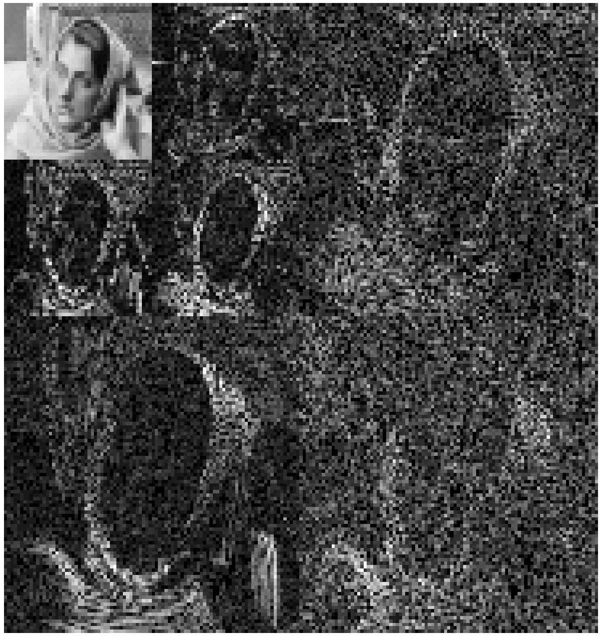


圖5.6 三階Haar 函數離散小波轉換

original picture



first order and second wavelet transforms

- 一般文件大多是一列一列儲存及傳送；但是，利用在影像的傳輸上則不適合，如若傳輸中斷或停止時，對於尚未傳送的部分影像是毫無訊息的。
- 假如我們能把重要的訊息先傳送，然後才傳送較不重要的訊息，如此即使傳輸中斷了，接收者仍可依據先前取得的重要訊息獲得整張影像之完整輪廓，這就是漸進式影像傳遞法的觀念。

- 使用漸進式傳送（progressive transmission）時，影像資料分成幾次傳送；每次接收端都可以得到更接近於原影像的重建訊號。
- 使用漸進式傳送的動機始於我們需要透過低頻寬的傳輸線（相對於資料量），例如電話線，來傳送影像資料。
- 重要的一點：一旦收到的影像資訊已經足以令人滿意或發現不是所要的影像，傳送便可馬上中止。這種由使用者中止所導致的傳送資料之減少我們稱爲有效壓縮（effective compression）。

- 爲了達到漸進式影像傳送的目的，我們以scan order 這種方式來傳送離散小波轉換後的影像。
- scan order 傳送方法：由低頻先傳送，在傳送中頻，最後才是高頻。
- 因爲低頻是影像最重要，中頻及高頻用來加強低頻，所以這種傳送方式會讓傳輸的影像由模糊到清楚。

- Scan order 的做法如下：將離散小波轉換所得到的頻帶以Z 字形的順序一一傳輸。
- 此傳輸的順序如圖5.7 所示，從最重要的低頻（LL3）到中頻（HL3、LH3）一直到最不重要的高頻（HH1）。
- 至於頻帶內的係數掃描方式，也是依Z 字形的順序傳輸係數。

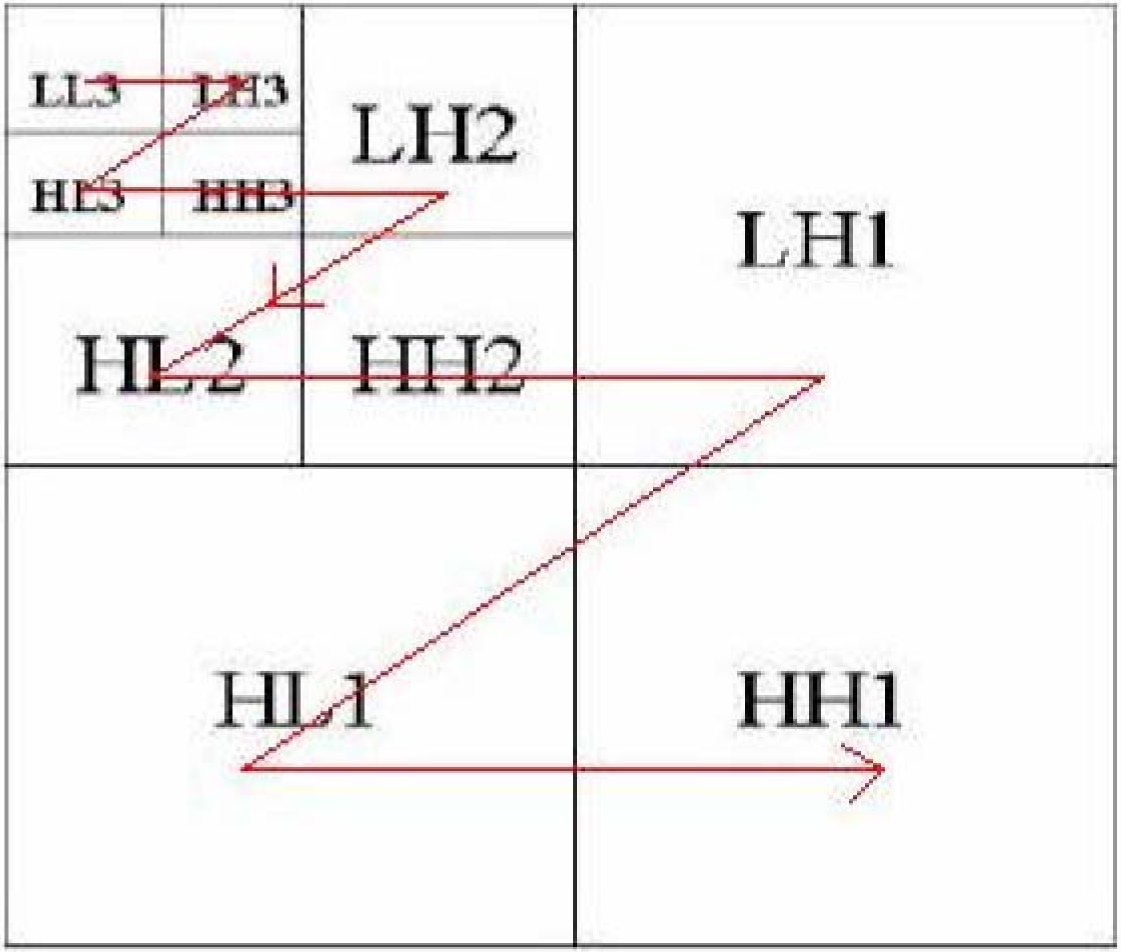
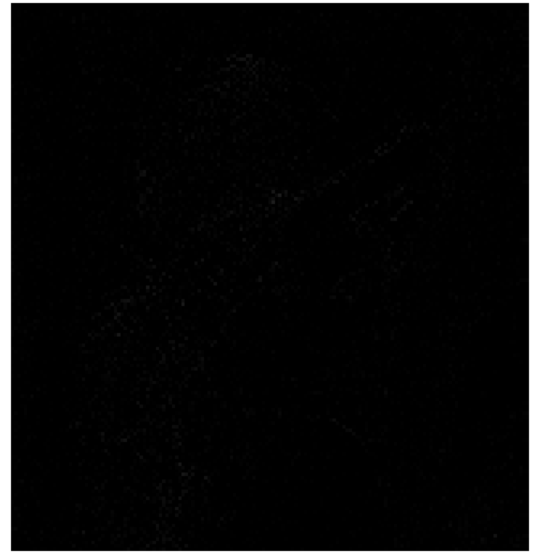
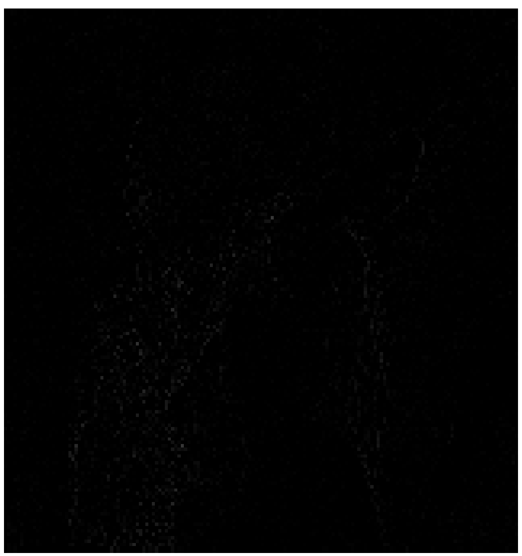


圖5.7 scan order 傳送頻帶順序之示意圖

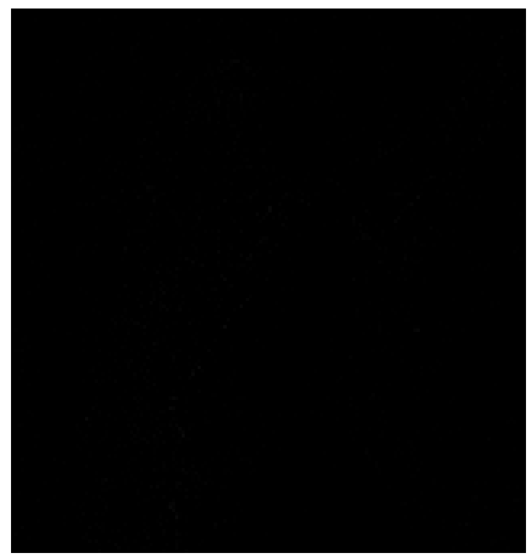
LL1



HL1



LH1



HH1

original picture

HH1 is lost

LH1 and HH1 are lost

HL1, LH1 and HH1 are lost

original picture

HH2 is lost

LH2 and HH2 are lost

HL2, LH2 and HH2 are lost

original picture

compression of ratio 1/4

compression of ratio 1/16

compression of ratio 1/64

original picture

MSE =57.4999

difference of origin and compression of 1/4

difference of origin and compression of 1/16

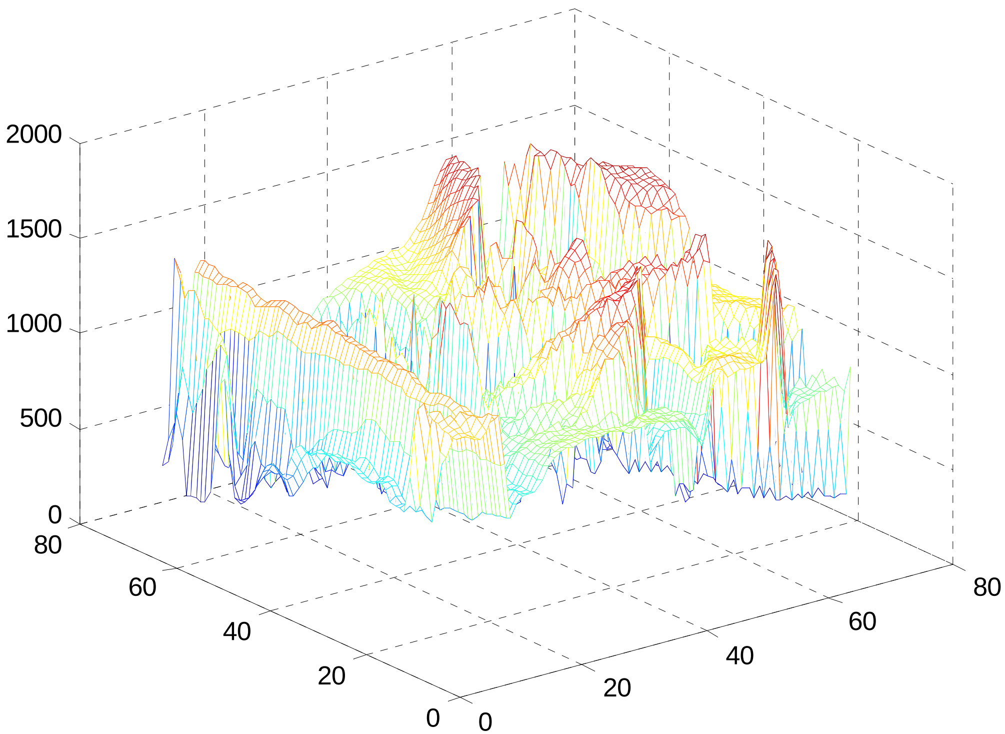
MSE = 899.4207

difference of origin and compression of 1/64

MSE = 8.5614e+003

Original picture

DWT DWT DWT



The absolute values of 64x64 coefficients of LL3

1.0e+003 *

1.3519	1.3410	1.3229	1.3133	1.3459	1.4278
1.3304	1.3323	1.3271	1.3151	1.3714	1.4266
1.3265	1.3378	1.3340	1.3334	1.3969	1.3908
1.3348	1.3510	1.3479	1.3830	1.4008	1.3504
1.3474	1.3614	1.3863	1.4113	1.3893	1.3471
1.3638	1.3749	1.3964	1.3489	1.3586	1.3619

The 6x6 coefficients of LL3 (a) LL3(1:6, 1:6)

作業5

- 針對 所附之資訊，請依照DWT中Haar方式，執行兩次DWT。
- 請詳列LL2, HL2, LH2, HH2, LL1, HL1, LH1, HH1等結果。

26	6	13	10
21	7	6	4
4	5	4	3
2	3	2	0

5.4 零樹小波編碼 (EZW, Embedded Zerotree Wavelet)

- 影像經過離散小波轉換得到數個頻帶，中頻及高頻的部份是用來加強低頻。
- 離散小波轉換後的係數中，低頻、中頻及高頻之間的關係可用四元樹表示，這四元樹我們稱之爲 quadtree。
- 在這四元樹中，每一節點都有四個子樹(subtree)。
- 以最低頻的係數爲根(root)(最低頻的係數不只一個，那麼根的個數也會不只一個)，此外，quadtree 以中頻及高頻的係數爲子節點。每一個根延伸出四個子節點，這些子節點的作用就是用來加強父節點的係數，如圖5.8(a) ~ (c)所示。

- 原則: $A^{(k)}$ 的係數 ⟶ $A^{(k-1)}$的4個相關係數
- 例如: LH4的係數 ⟶ LH3的4個相關係數
  LH3的係數 ⟶ LH2的4個相關係數
  LH2的係數 ⟶ LH1的4個相關係數

- 以LH4 中的X 係數爲例，X 對應到LH3 的四個係數，所以X 爲LH2 中四個係數的父節點，而LH3 的四個係數是X 的子節點。
- 每一個子節點也可再延伸出子節點，其子節點的作用也是用來加強它的父節點。
- 例如圖5.8(b)中，LH3裡的Y 係數，Y 是X 的子節點，而Y 又對應到LH2 內的四個係數，所以LH2 內的四個係數又是Y 的子節點。
- 因此，若把Y 係數和Y 的子節點構成了X係數的子樹。以X 係數爲根的子樹就是一個quadtree，如圖5.8(b)~ (c)

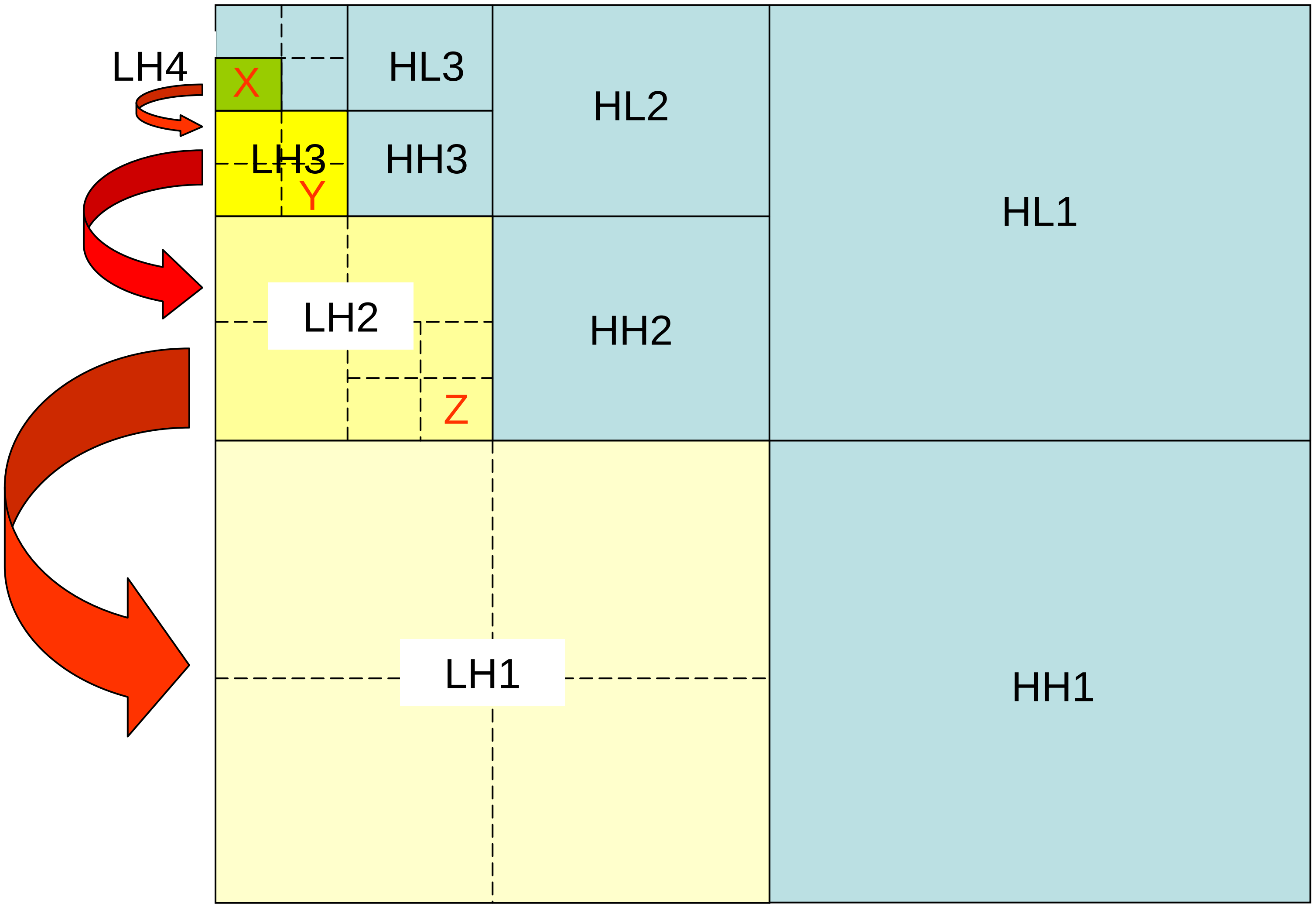


圖5.8 (a)

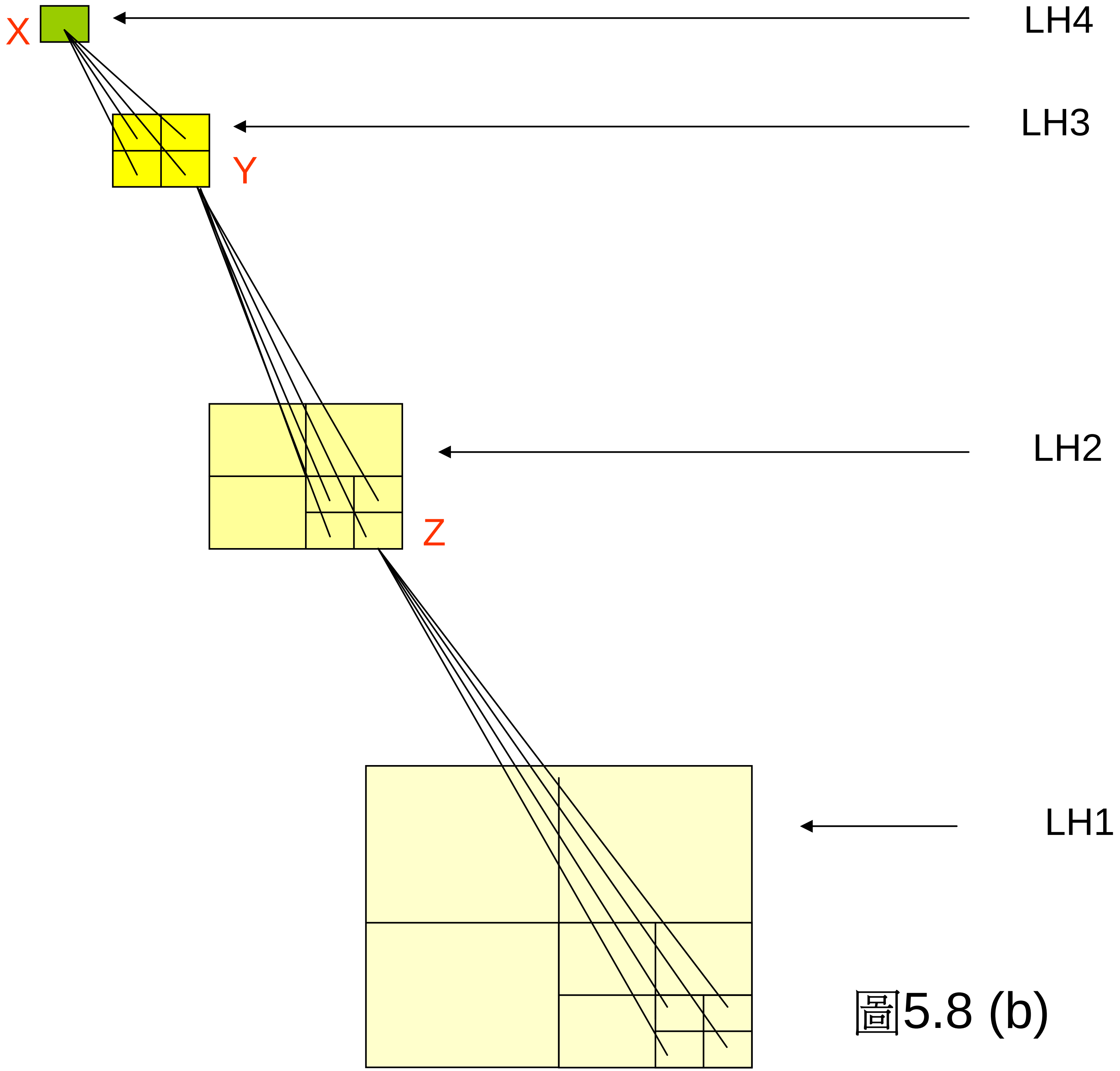


圖5.8 (b)

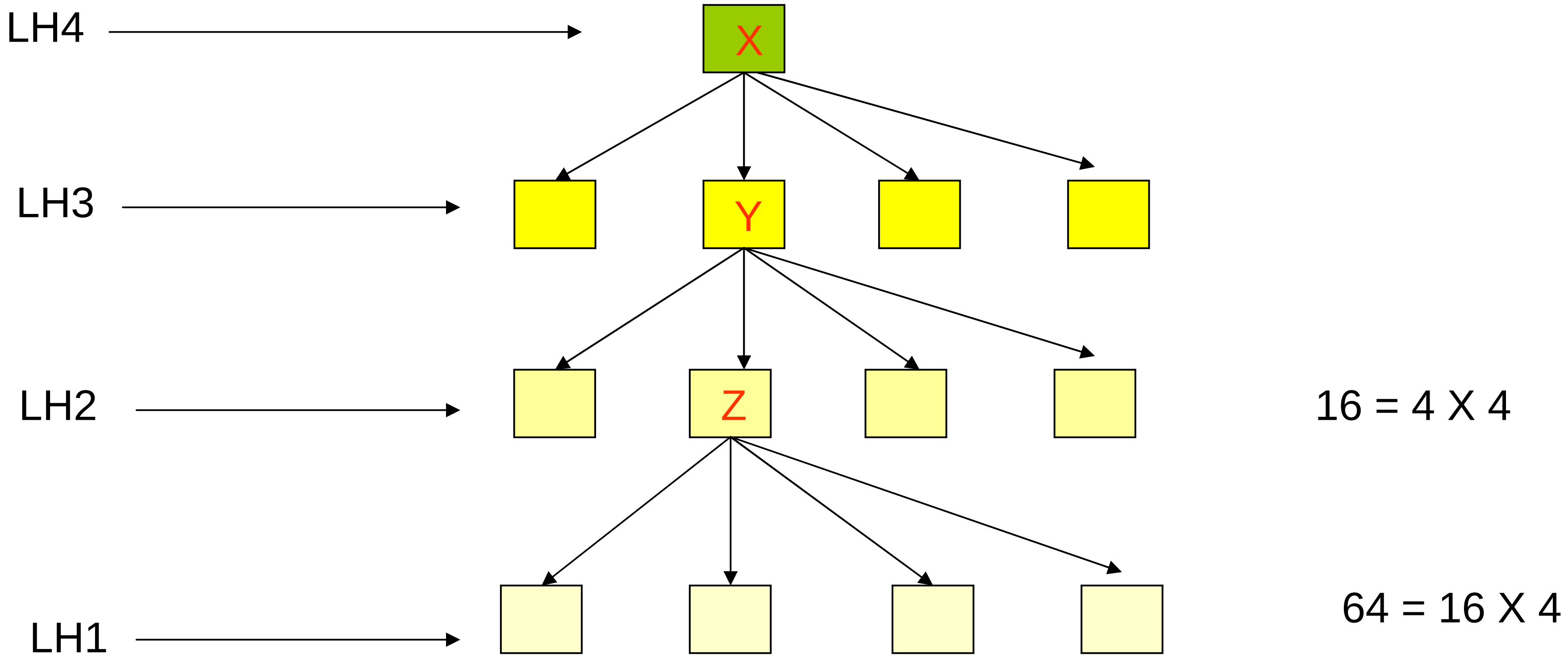
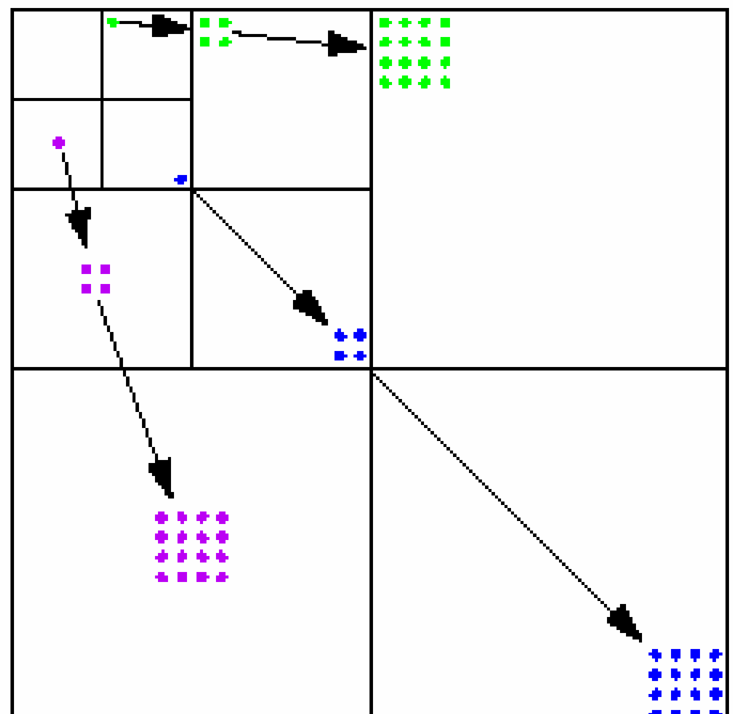


圖5.8 (c)



- 由於每個quadtree 的根都是由低頻係數所組成的，是影像中最重要的資訊。
- quadtree中的其他節點都是用來加強其上層節點的。所以，利用quadtree 這種結構，由根到樹葉的方式來傳輸影像時，影像會從模糊漸漸到清晰。

Zerotree

- 在說明EZW 之前，我們必須先介紹什麼是zerotree 的觀念。我們將所得到的離散小波轉換結果，以座標圖來表示，如圖5.9(a)是離散小波轉換後的結果。
- 將圖5.9(a)中的係數以座標圖表示，會產生類似圖5.9(b)的三維空間座標圖；其中x 軸及y 軸代表離散小波轉換後結果表的水平位置及垂直位置。
- z 軸表示轉換後係數值的絕對值大小，由於低頻係數通常較大，所以箭頭的高度比較高。

	0	1	2	3	4	5	6	7
0	63	-34	49	10	7	13	-12	7
1	-31	23	14	-13	3	4	6	-1
2	15	14	3	-12	5	-7	3	9
3	-9	-7	-14	8	4	-2	3	2
4	-5	9	-1	47	4	6	-2	2
5	3	0	-3	2	3	-2	0	4
6	2	-3	6	-4	3	6	3	6
7	5	11	5	6	0	3	-4	4

圖5.9（a）是離散小波轉換後的結果

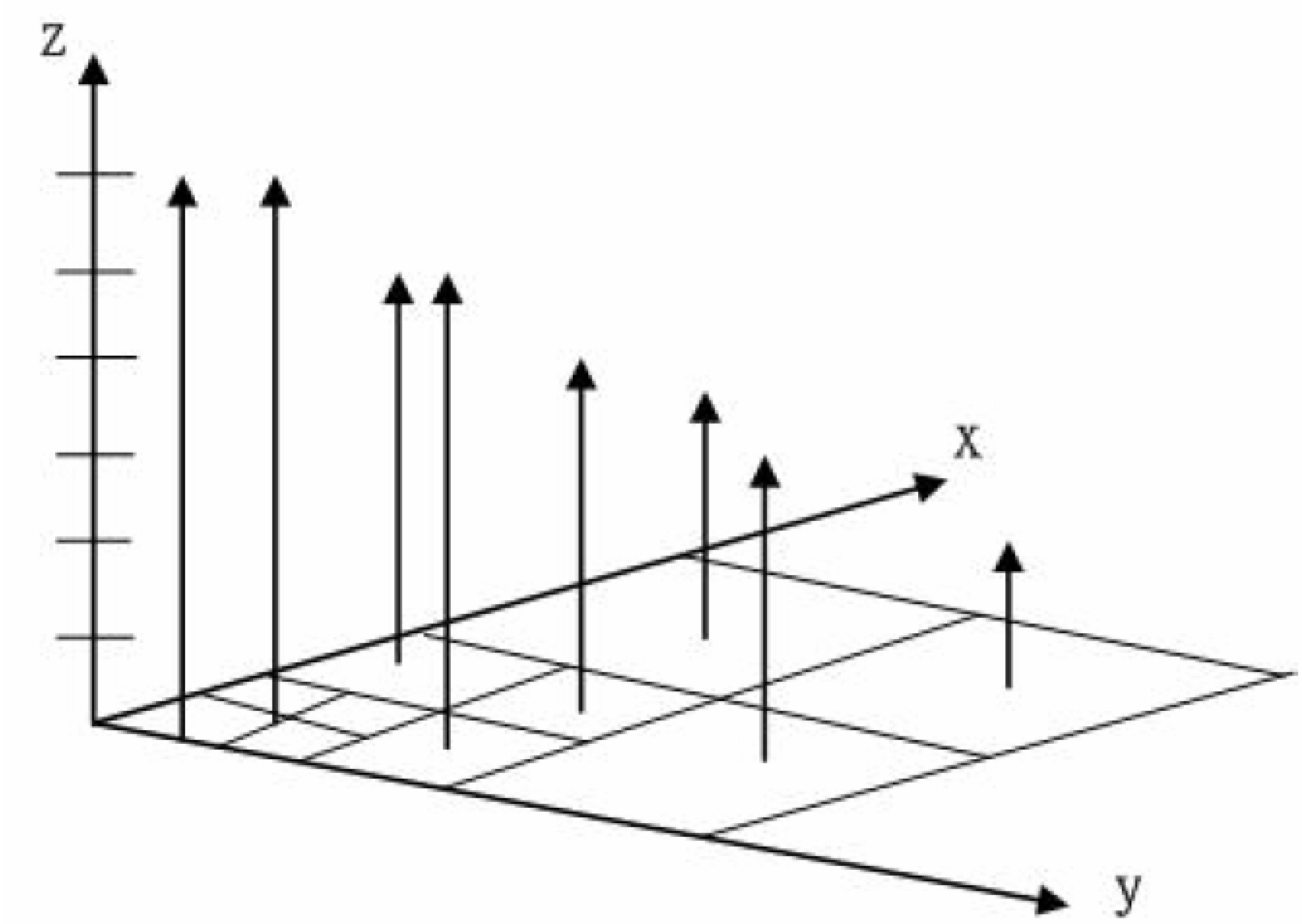


圖5.9（b）的三維空間座標圖

- 由於EZW所設定門檻值由大至小遞減，所以先前儲存的係數值比較大，後面儲存的係數值比較小。
- 係數值大的是重要資訊，係數值小的則是不重要的部份，因此，在傳輸過程中，前面儲存的係數值會比後面重要，所以先傳遞。
- 以上就是所謂zerotree 的做法。利用zerotree 的觀念，就可說明EZW 影像壓縮法的設計。

- 利用門檻值(threshold)將z軸切割成數個層次，在此以符號T_k 來表示第k 個門檻值。
- 只要在門檻值以上的係數，就視其爲重要係數；而門檻以下的係數，就視爲不重要的係數。
- 如圖5.9所示，設定第0次切割的門檻值爲T_0，只要是數值的絕對值大於T_0 的係數，都視爲重要的係數，然後針對重要的係數做第0次的處理及儲存。

- 然後再設定下一個切割的門檻值爲T_1，接著處理及儲存絕對值大於T_1 的重要係數。
- 以此類推。將每次切割後所儲存的結果視爲一個個的區段，然後將這些區段串聯起來；在傳輸影像的過程中，只要一個區段、一個區段的分別傳送即可。

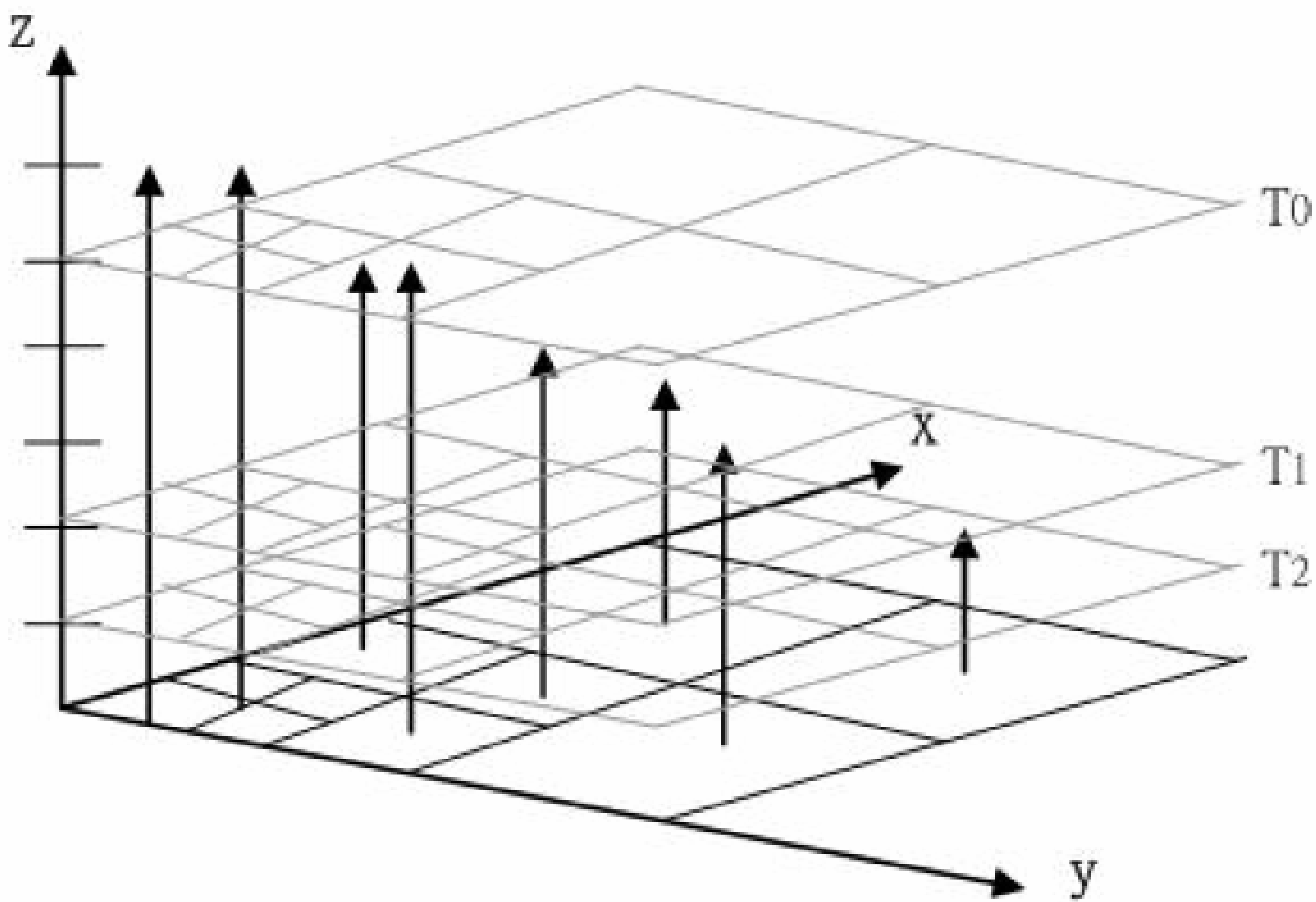


圖5.10 zerotree切割示意圖

EZW 的基本觀念及符號說明

- EZW 這個方法是由Shapiro 於1993 年發表，它是一種對離散小波轉換後係數編碼的方法，其目的是爲了對影像作有效的壓縮。離散小波轉換的兩個主要重點：
- 重點一：當影像作離散小波轉換後，高頻部份的係數會小於低頻部份的係數；這是因爲低頻係數是不斷將像素數值相加的結果，而高頻係數則是不斷將像素數值相減的結果。

- 重點二：離散小波轉換後，係數越大越重要；因爲係數越大的成分也就是影像低頻的部份，由這個部份可以得到模糊的影像；係數小的部份就是高頻的部份，它的作用是加強低頻部份，可以使影像更加的清晰。

專有名詞與符號

- N：以C_{ij} 來表示離散小波轉換後的係數，

$$N = \left\lfloor \log_2(\max(C_{ij})) \right\rfloor$$

（ $Y = \lfloor X \rfloor$ 表示等於或小於X的最大整數)

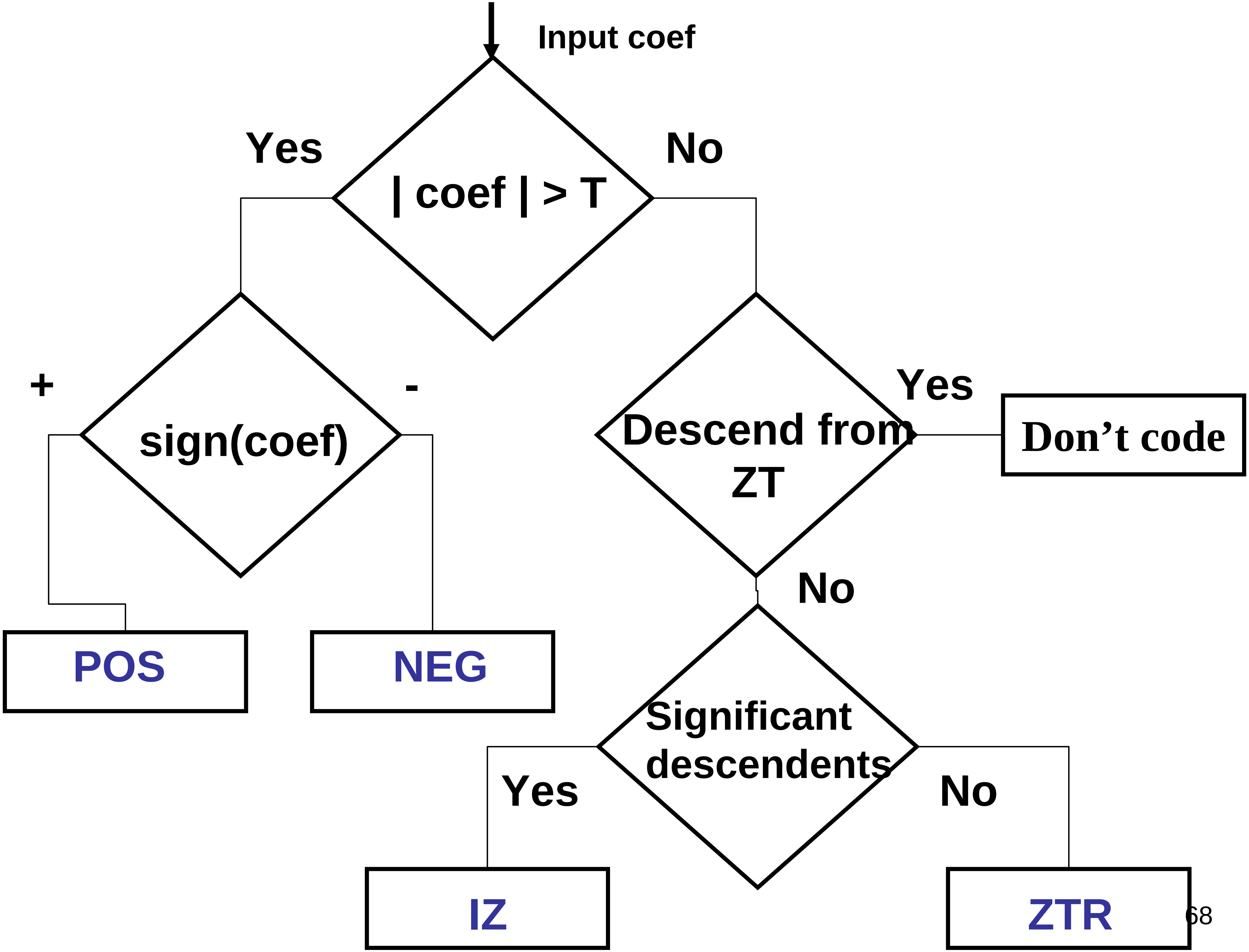
- T_k：爲前面所述zerotree 切割的門檻值；若是係數的絕對值大於門檻值，其爲重要係數；否則就視爲不重要係數。T_k 的下標k 值是由0 開始遞增的整數，而且$T_0>T_1>T_2>\cdots$，逐次遞減。

- POS：若係數C_{ij}爲正(C_{ij}>0)且大於目前的門檻值T_k，則將它標示爲POS，表示C_{ij}是一個大於零的重要係數。
- NEG：若係數C_{ij}爲負的(C_{ij}<0)且它的絕對值大於目前的門檻值T_k，則將它標示爲NEG，表示C_{ij}是一個小於零的重要係數。
- ZTR：若是C_{ij}的絕對值小於目前的門檻值T_k，且C_{ij}的子樹中都沒有大於目前門檻值T_k之係數，則將它標示爲ZTR，表示C_{ij}非重要係數，而且其子樹中也無重要係數存在。

- IZ：若C_{ij}的絕對值小於目前的門檻值T_k，但是C_{ij}的子樹中有大於目前門檻值T_k之係數，則將C_{ij}標示爲IZ，表示C_{ij}係數是不重要的，但其子樹中有重要的係數。

- 以圖5.11 爲例；$C_{11} = 23$，而C_{11}的子樹爲灰色部份，門檻值T_k=32，因爲23<32 且C_{11} 的子樹中沒有大於T_k的係數，所以C_{11}會被標示成ZTR。

- 以圖5.12 爲例；C_{10} = -31， C_{10} 的子樹爲黃色部份。假設目前的門檻值T_0= 32，因爲 |-31| < 32 所以C_{10} 是不重要的係數，但C_{10} 的子樹中有一個C_{43} 係數大於T_0，所以C_{10}被標示爲IZ。


Input coef
| coef | > T
Yes
No
sign(coef)
+
-
POS
NEG
Descend from ZT
Yes
Don’t code
No
Significant descendents
Yes
No
IZ
ZTR

63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	4	3	6	3	6
5	11	5	6	0	3	-4	4

圖5.11 ZTR範例說明圖

63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	4	3	6	3	6
5	11	5	6	0	3	-4	4

圖5.12 IZ範例說明圖

EZW 影像壓縮編碼程序

- **步驟一**：設定門檻值T_0或T_k

  先前的公式求出N，然後令$T_0=2^N$，以圖5.12 爲例，離散小轉換後的結果，其中

  $\max(C_{ij}) = C_{00} = 63$, $N = 5, T_0=2^5=32$
- $T_k = T_{k-1}/2$， $1 \leq K$

- **步驟二**：重建數值的計算

  EZW 第0次切割的重建數值R_0定義爲離散小波轉換結果中最大的係數值($\max(C_{ij})$)與第0次切割的門檻值T_0 相加後在除以2 的數值

$$\mathrm{R}_0 = (\max(C_{ij}) + \mathrm{T}_0)/2$$

- 下一次切割的重建係數R_k 必須是前一次重建係數除以2。

$$R_k = R_{k-1} / 2$$

- 在解壓縮過程中，如果係數C_{ij}的值大於目前的門檻值T_k，則C_{ij}的重建數值（解壓縮數值）爲R_k，否則C_{ij} 的重建數值爲0
- 將重建係數R_k 送至接收端，進行解壓縮。

- **步驟三**：針對某一次zerotree 切割，建立該次切割後的重要係數表：頻帶，係數值，符號。

  頻帶：存放係數C_{ij}是屬於那一個頻帶

  係數值：係數值C_{ij}的值

  符號：係數值C_{ij} 所對應的POS、NEG、ZTR、IZ等

- 完成了這個步驟後，重要係數表中「符號」欄位的內容就是這一次zerotree切割所要的結果。
- 接著EZW 會輸出「符號」欄位的內容，並將這個「符號」欄位輸出的資料串列稱之爲D_k 串列；此k 與T_k 的下標k 意義相同，都是由0、1、2、3…逐漸遞增的編號。

- **步驟四**：建構值的第一次精煉。

  完成了上述的步驟後，EZW 得到了目前門檻值T_k這一層所對應的重要係數。但是，這些重要係數在解壓縮時的建構值太不精確了。因此，EZW對前述所搜尋到的重要係數做建構值精煉的處理，其目的在解壓縮時，還原的係數值可以更接近壓縮前的係數值。

- EZW 建立了一個表格，暫存每一個重要係數的精煉結果，這個表格稱爲精煉表。其包含下面三個欄位：係數值、符號、精煉數值

  係數值：重要係數C_{ij}的值

  符號：填0 或1

  精煉數值：重要係數C_{ij}的精煉結果值

- 建構精煉數值的步驟如下：

1)EZW 依序從第三步驟產生的重要係數表中取出，「符號」欄位是POS 或NEG 的重要係數。

2)如果重要係數C_{ij} 的絕對值比目前的重建數值R_k 還大，將在C_{ij} 精煉表中的「符號」欄位放入1，否則爲0。

3)重要係數C_{ij} 的絕對值比目前的重建數值R_k 還大，精煉數值爲

$$R_k^{(1)} = (\max(C_{ij}) + R_k)/2 \quad (\text{當 } k = 0)$$

$$R_k^{(1)} = (T_{k-1} + R_k)/2 \quad (\text{當 } k > 0)$$

重要係數C_{ij} 的絕對值小於目前的重建數值R_k ，精煉數值爲

$$R_k^{(0)} = (T_k + R_k)/2$$

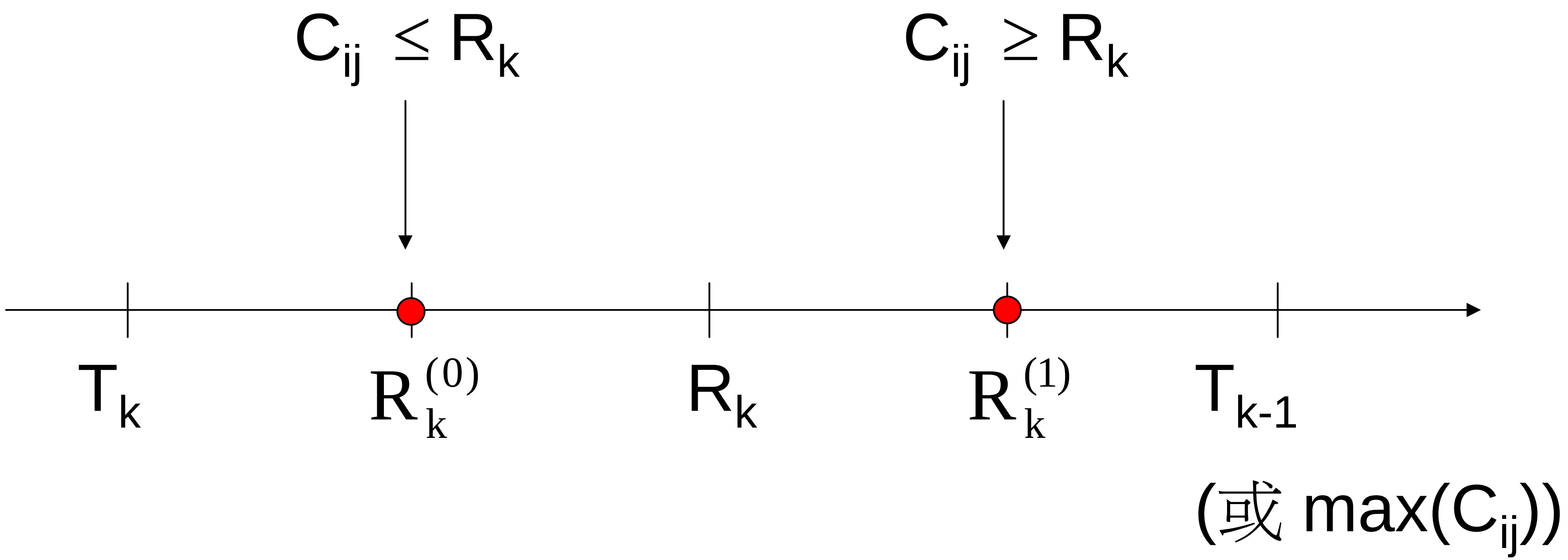


圖5.13重建值的第一次精煉示意圖

- 最後，EZW 輸出這個步驟所產生表格中的「符號」欄位內容，EZW 將輸出的資料串列稱之爲S_k串列。

- 也就是說，將離散小波轉換結果中絕對值大於目前門檻值T_k 的重要係數處理告一段落。

- **步驟五**：先前一次切割的重要係數建構值，再做精煉。

  重複步驟四的動作，讓先前一次重要係數的精煉數值更接近原本的係數值。在處理完上述動作後，並輸出精煉表中符號欄位，形成一組A_k 串列。

- 假設重要係數 C_{ij}=49，C_{ij} 第一次的精煉數值爲56。因爲49<56，當再次精煉時，EZW 會將49的「符號」欄位放入0；並取重建值R 與精煉數值R_1 的中間值(即(56+48)/2=52)作爲C_{ij} 第二次建構的精煉數值；

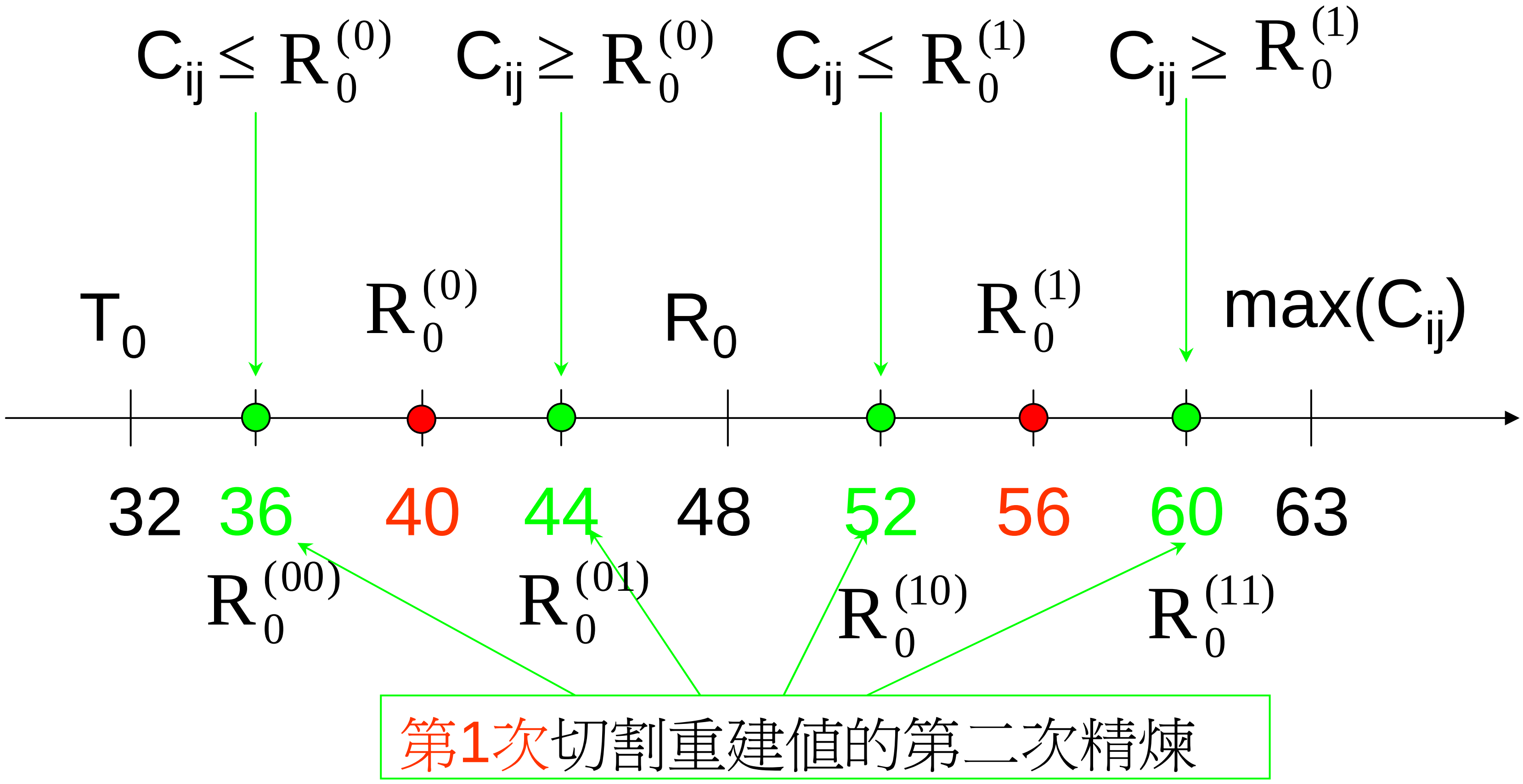
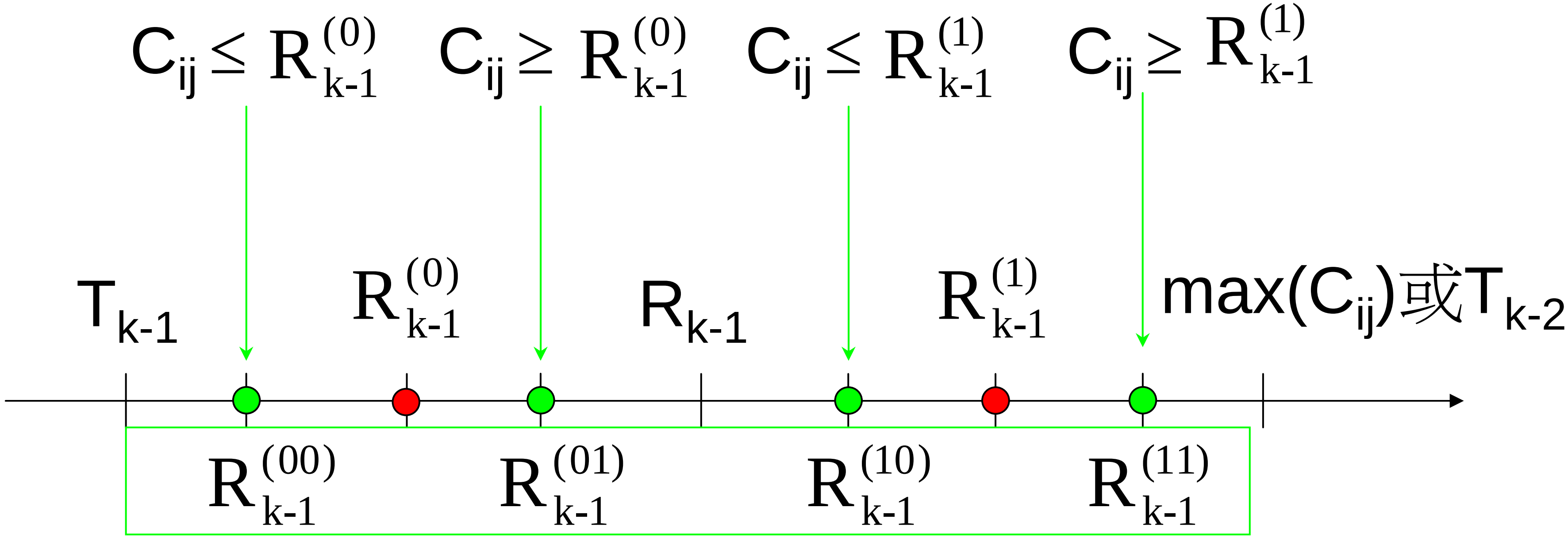


圖5.14(a)第1次切割重建值的第二次精煉說明圖



$$R^{(00)}_{k-1} = (T_{k-1} + R^{(0)}_{k-1})/2 \qquad R^{(10)}_{k-1} = (R_{k-1} + R^{(1)}_{k-1})/2$$

$$R^{(01)}_{k-1} = (R_{k-1} + R^{(0)}_{k-1})/2$$

$$R^{(11)}_{k-1} = (T_{k-2}\ or\ \max(C_{ij}) + R^{(1)}_{k-1})/2$$

圖5.14(b)前一次切割的第二次精煉說明圖(通式)

- **步驟六**：重新設定重要之係數值。

  將步驟三所搜尋到的重要係數一一從離散小波轉換的結果中取出，扣除此係數在步驟四及步驟五設定的精煉值，然後存回離散小波轉換結果中的原來位置。

- 最後，將步驟三的D_k、步驟四的S_k 與步驟五的A_k串列輸出後，EZW 就完成了這一次的切割動作。
- 當第0次切割時，除上述外，還要輸出$\max(C_{ij})$ 。

EZW 影像解壓縮程序

- max(C_{ij})爲解壓縮重要依據
- 在利用已穫取的的資料組(D_0, S_0, A_0) , (D_1, S_1, A_1) , (D_2, S_2, A_2)……等 ，並經由相對應於上述六個壓縮編碼的程序步驟，重建原影像。

第0次切割

EZW 影像壓縮編碼過程範例

步驟一:

$$N = \lfloor \log_2(63) \rfloor = 5$$

$T_0 = 2^5 = 32$

步驟二:

$R_0 = (63+32)/2 = 48$

63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	4	3	6	3	6
5	11	5	6	0	3	-4	4

第0次切割

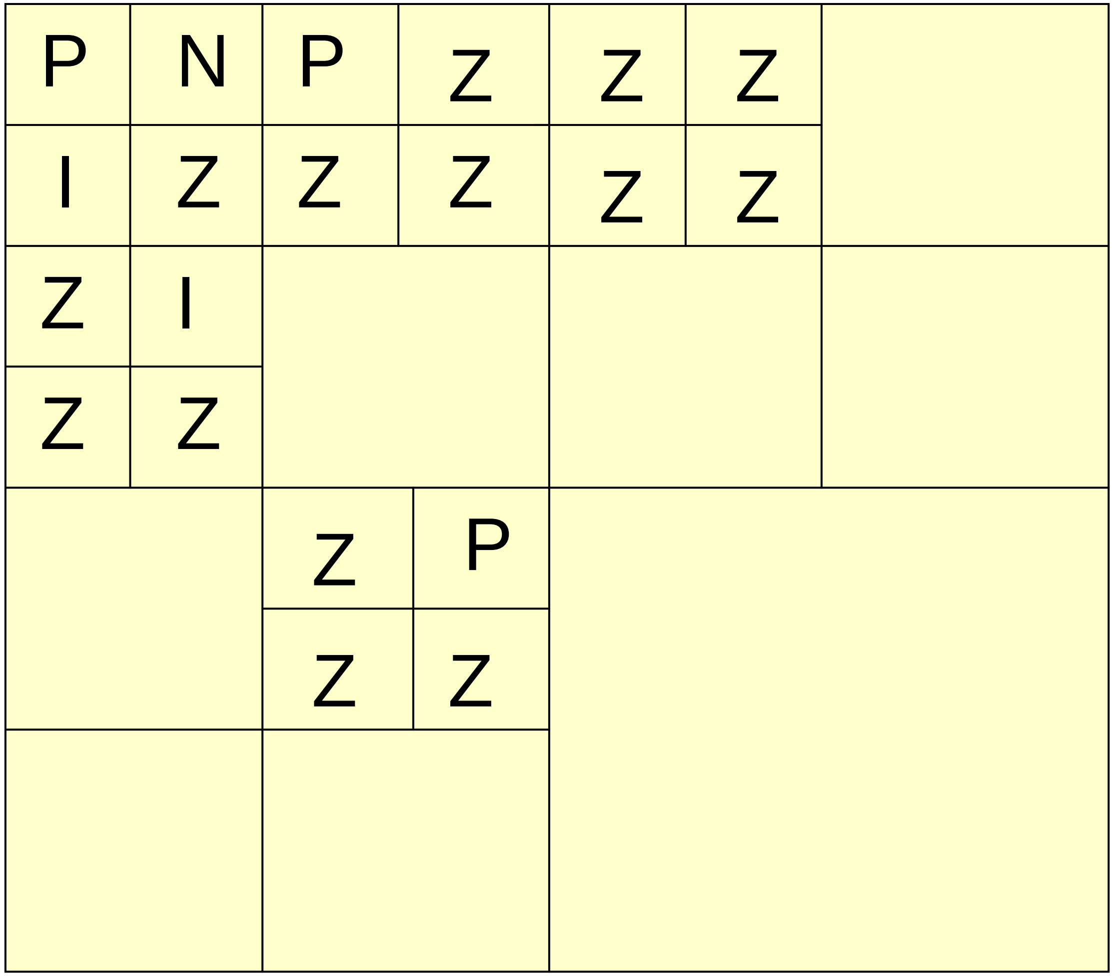
步驟三:

63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	4	3	6	3	6
5	11	5	6	0	3	-4	4

D_0 = {P N I Z, P Z Z Z,
Z I Z Z, Z Z Z Z,
Z P Z Z}

頻帶	係數值	符號	重建數值
LL3	63	POS	48
HL3	-34	NEG	-48
LH3	-31	IZ	0
HH3	23	ZTR	0
HL2	49	POS	48
HL2	10	ZTR	0
HL2	14	ZTR	0
HL2	-13	ZTR	0
LH2	15	ZTR	0
LH2	14	IZ	0
LH2	-9	ZTR	0
LH2	-7	ZTR	0
HL1	7	ZTR	0
HL1	13	ZTR	0
HL1	3	ZTR	0
HL1	4	ZTR	0
LH1	-1	ZTR	0
LH1	47	POS	48
LH1	-3	ZTR	0
LH1	2	ZTR	0

第0次切割



P → POS

I → IZ

N → NEG

Z → ZTR

步驟四: 第一次精煉數值

第0次切割

$$R_0^{(1)} = (\max(C_{ij}) + R_0)/2$$
$$= (63 + 48)/2 = 56$$
$$R_0^{(0)} = (R_0 + T_0)/2$$
$$= (48 + 32)/2 = 40$$

63	-34	49	10	7	13	-12	7
-31	23	14	13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	4	3	6	3	6
5	11	5	6	0	3	-4	4

$S_0 = \{1010\}$

重要係數	符號	第一次精煉數值
63	1	56
-34	0	-40
49	1	56
47	0	40

- **步驟五:**

  由於目前是EZW第0次切割的處理，在此之前並無重要係數的產生，故不需要執行此步驟。

  $A_0 = \varnothing$ {空集合}

- 步驟六:

63-56 = 7

-34-(-40)=6

49 - 56 = -7

47- 40 = 7

7	6	-7	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	7	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	4	3	6	3	6
5	11	5	6	0	3	-4	4

步驟一:

$T_1 = T_0/2 = 16$

步驟二:

$R_1 = R_0/2 = 24$

7	6	-7	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	7	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	4	3	6	3	6
5	11	5	6	0	3	-4	4

第1次切割

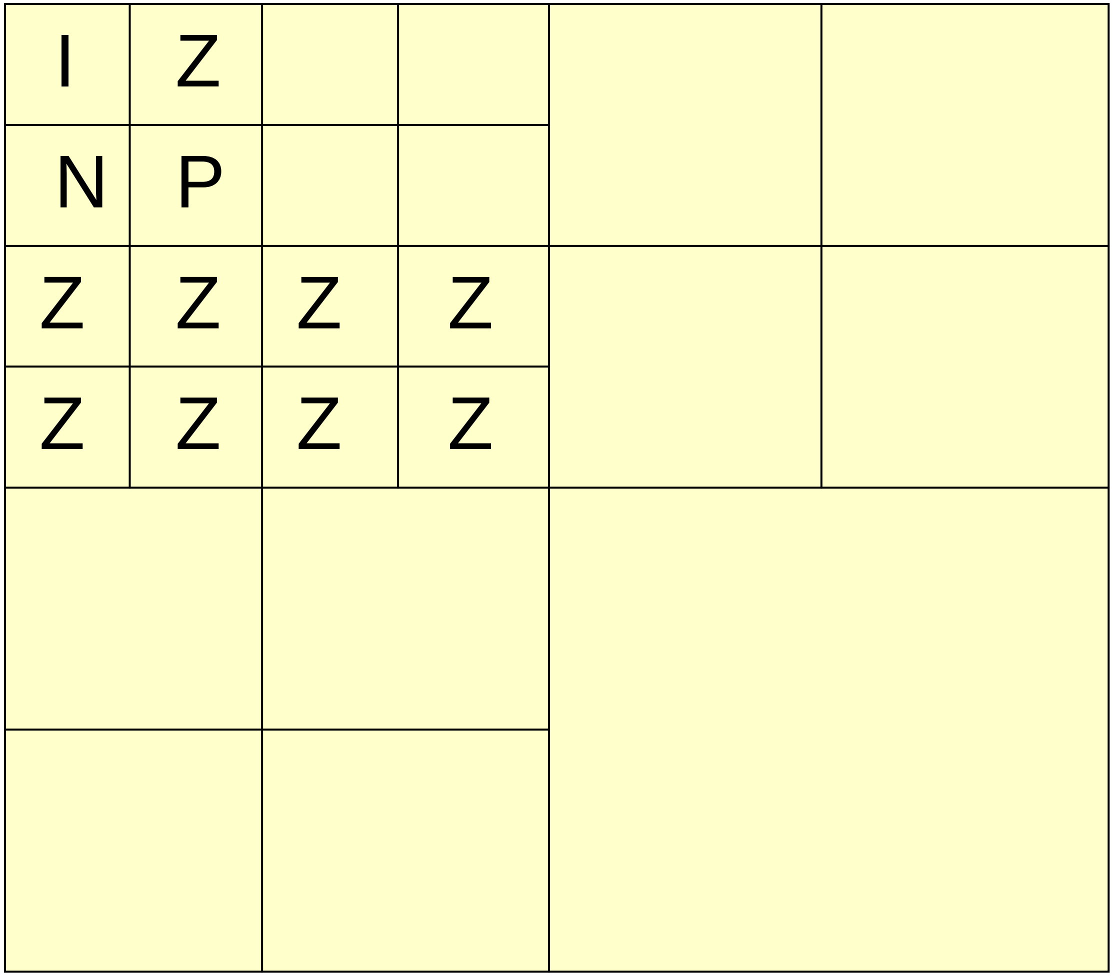
步驟三:

7	6	-7	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	7	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	4	3	6	3	6
5	11	5	6	0	3	-4	4

頻帶	係數值	符號	重建數值
LL3	7	IZ	0
HL3	6	ZTR	0
LH3	-31	NEG	-24
HH3	23	POS	24
LH2	15	ZTR	0
LH2	14	ZTR	0
LH2	-9	ZTR	0
LH2	-7	ZTR	0
HH2	3	ZTR	0
HH2	-12	ZTR	0
HH2	-14	ZTR	0
HH2	8	ZTR	0

$$D_1 = \{I\ Z\ N\ P,\ Z\ Z\ Z\ Z,\ Z\ Z\ Z\ Z\}$$

第1次切割



P → POS

I → IZ

N → NEG

Z → ZTR

第1次切割

步驟四: 第一次精煉數值

7	6	-7	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	7	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	4	3	6	3	6
5	11	5	6	0	3	-4	4

$$R_1^{(1)} = (T_0 + R_1)/2 = (32+24)/2 = 28$$

$$R_1^{(0)} = (R_1 + T_1)/2 = (24+16)/2 = 20$$

重要係數	符號	第一次精煉數值
-31	1	-28
23	0	20

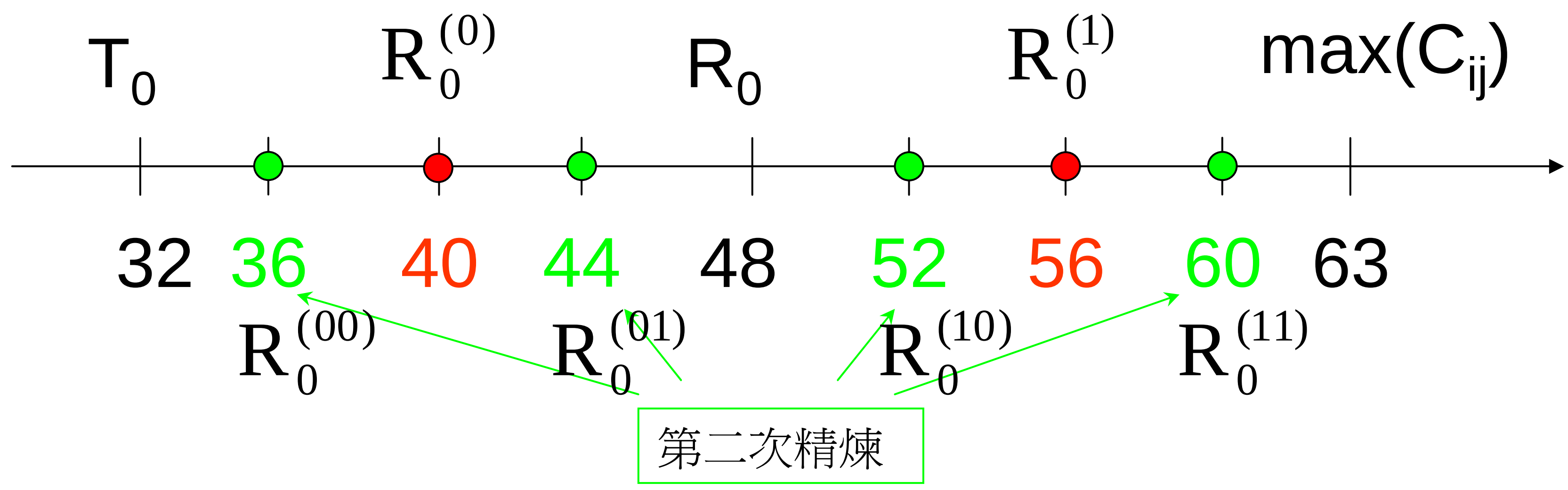
$S_1 = \{10\}$

第1次切割

- **步驟五:**

$A_1 = \{1001\}$

T_0 重要係數	符號	第一次 精煉數值	第二次 精煉數值
63	1	56	60
-34	0	-40	-36
49	0	56	52
47	1	40	44



- 步驟六:

63-60 = 3

-34-(-36) = 2

-31-(-28)= -3

49 - 52 = -3

23 - 20= 3

47- 44 = 3

3	2	-3	10	7	13	-12	7
-3	3	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	3	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	4	3	6	3	6
5	11	5	6	0	3	-4	4

- 以下爲切割兩次後，所輸出之結果 max(C_{ij})

D_0 = {P N I Z, P Z Z Z, Z I Z Z, Z Z Z Z, Z P Z Z}

S_0 = {1010}

A_0 = ∅ {空集合}

D_1 = {I Z N P, Z Z Z Z, Z Z Z Z}

S_1 = {10}

A_1 = {1001}

第0次重建

EZW 影像解壓縮過程範例

步驟一:由已獲取的max(C_{ij}) = 63 ，可計算出

- $N = \lfloor \log_2(63) \rfloor = 5$
- $T_0 = 2^5 = 32$

步驟二:

- $R_0 = (63+32)/2 = 48$
- 清除接收端相關影像矩陣資料

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

第0次重建

- **步驟三：**依據EZW中D_0，將R_0或0填入原影像矩陣中

D_0 ={ P N I Z,

P Z Z Z,

Z I Z Z,

Z Z Z Z,

Z P Z Z }

48	-48	48	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	48	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

第0次重建

- **步驟四**：依據EZW中S_0與R_0關係，將 $R_0^{(1)}$, $R_0^{(0)}$ 取代原R_0

$$R_0^{(1)} = (\max(C_{ij}) + R_0)/2$$
$$= (63 + 48)/2 = 56$$

$$R_0^{(0)} = (R_0 + T_0)/2$$
$$= (48 + 32)/2 = 40$$

S_0 = {1010}

56	-40	56	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	40	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

- **步驟五**：因爲$A_0 = \varnothing$，所以原影像矩陣不做變動。

步驟一:

$T_1 = T_0/2 = 32/2 = 16$

步驟二:

$R_1 = R_0/2 = 48/2 = 24$

第1次重建

- **步驟三:**依據EZW中D_1，將R_1或0填入原影像矩陣中

D_1 = {I Z N P,

Z Z Z Z,

Z Z Z Z }

56	-40	56	0	0	0	0	0
-24	24	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	40	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

第1次重建

- 步驟四：依據EZW中S_1與R_1關係，將 $R_1^{(1)}, R_1^{(0)}$ 取代原R_1

$$R_1^{(1)} = (T_0 + R_1)/2 = (32+24)/2 = 28$$

$$R_1^{(0)} = (R_1 + T_1)/2 = (24+16)/2 = 20$$

$$S_1 = \{10\}$$

56	-40	56	0	0	0	0	0
-28	20	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	40	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

第1次重建

- **步驟五**：依據EZW中A_1與 $R_0^{(0)}$、$R_0^{(1)}$ 關係，將以 $R_0^{(00)} R_0^{(01)}\ R_0^{(10)}\ R_0^{(11)}$ 取代原 $R_0^{(1)}, R_0^{(0)}$

$$R_0^{(00)} = 36$$

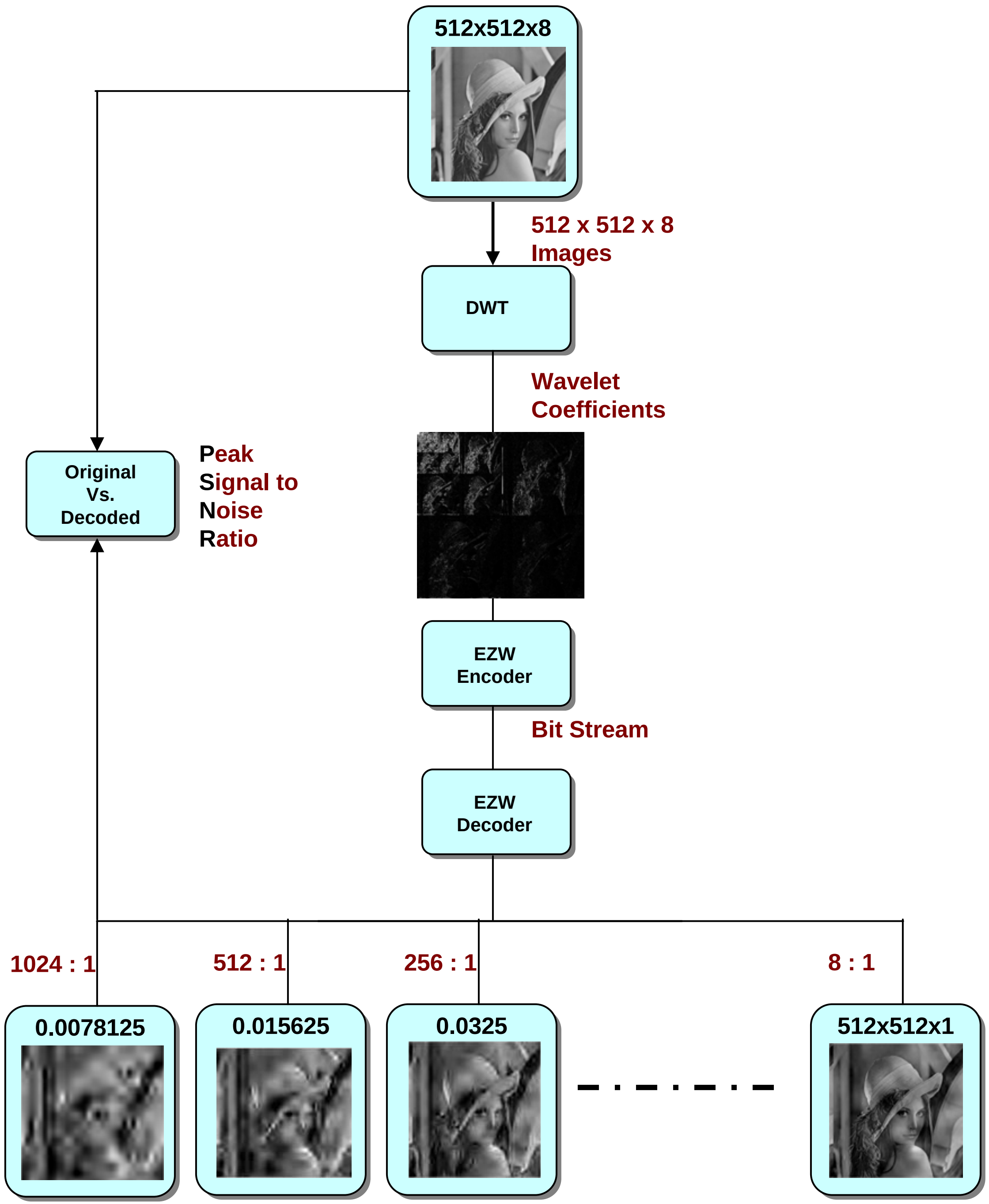
$$R_0^{(01)} = 44$$

$$R_0^{(10)} = 52$$

$$R_0^{(11)} = 60$$

$$A_1 = \{1001\}$$

60	-36	52	0	0	0	0	0
-28	20	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	44	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0


512x512x8
512 x 512 x 8
Images
DWT
Wavelet
Coefficients
Original
Vs.
Decoded
Peak
Signal to
Noise
Ratio
EZW
Encoder
Bit Stream
EZW
Decoder
1024 : 1
512 : 1
256 : 1
8 : 1
0.0078125
0.015625
0.0325
512x512x1


1024:1
512:1
256:1
128:1

64:1

32:1

16:1

8:1

Results – Original Vs. Decoded using 45504 bytes

Vs.

6:1

作業6

- 針對 所附之資訊，請依照EZW壓縮方式，執行兩次的切割與重建。
- 請詳列每步驟相關的結果如(D_0, S_0, A_0)及(D_1, S_1, A_1) 等。

26	6	13	10
-7	7	6	4
4	-4	4	-3
2	-2	-2	0