第四章 頻域分割達到空間解析度調變 (Spatial Scalability with Frequency domain partitioning)

本章說明本篇論文提出的方法，從基本概念，推展至實作上的重點。

4.1 概念

幾乎所有的多媒體壓縮技術都是對頻率域(Frequency Domain)的係數作壓縮，而 MPEG-4 的編碼流程中，儘管有不同的預測型態(Prediction Type)，不同的量化步階(Quantization Step)，基本層(Base-layer)與加強層(Enhancement-layer)編碼過程也不同，而造成不同的編碼路徑(Coding Path)，但每編碼路徑上共同的運算是 -- 離散餘弦轉換。

而離散餘弦轉換將時間域的資料轉換爲頻率域的係數，每個係數代表原數值在特定空間解析度的分量大小，有些是高頻率分量，有些是低頻率分量。所以，當我們只保留低頻率係數，捨去高頻率係數，本質上，保留的資訊就如同金字塔型影像架構的低解析度影像。

4.1.1 以離散餘弦轉換作爲縮減取樣之濾波

經過進一步查證，我們發現， 離散數學轉換可以作爲縮減取樣的濾波方法，其步驟爲：

離散餘弦轉換 → 保留低頻係數 → 反轉離散數學轉換 → 縮減取樣

因爲濾波的優劣直接影響到低解析影像的品質，所以在採用離散餘弦濾波達到空間解析度可調層次式之前，我們先比較離散餘弦轉換濾波之縮減取樣與一般金字塔型空間解析度可調層次式所採用的縮減取樣的濾波方法。

4.1.1.1 不同縮減取樣方法之比較

由於縮減取樣之濾波方式並無精確的最佳定義，故本節我們比較一些常見的方法，並觀察其差異。我們採用下列濾波方式來比較：

- 平均濾波：
  把 4 個影像點數值平均。
- 離散餘弦濾波：
  以 8x8 的區塊爲單位，做 8x8 離散數學轉換，保留 4x4 低頻，經適當比調整後以 4x4 離散餘弦反轉換。
- 遮罩濾波：
  利用階數不等的濾波遮罩進行濾波。我們採用的是 MATLAB 標準濾波遮罩。
- 無濾波：
  即直接縮減取樣，不做任何濾波。

原圖			
無濾波	平均濾波	離散餘弦濾波	二階濾波

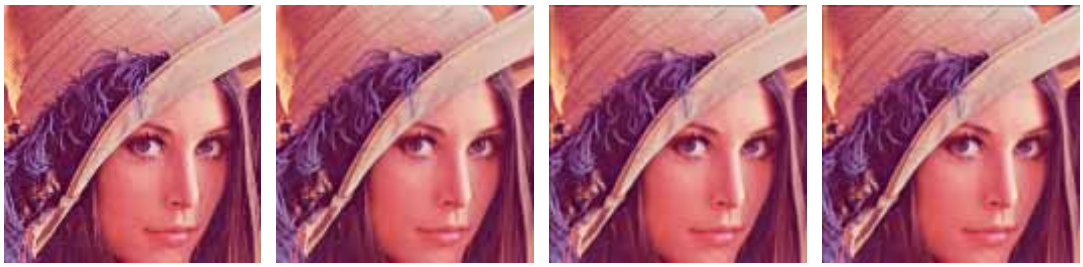


圖 4.1 縮減濾波比較

由液晶顯示器或彩色雷射印表機的精細度來觀察，幾乎無法辨認各種濾波方式的差異，以下我們觀察其局部放大圖來討論其細微的差異。

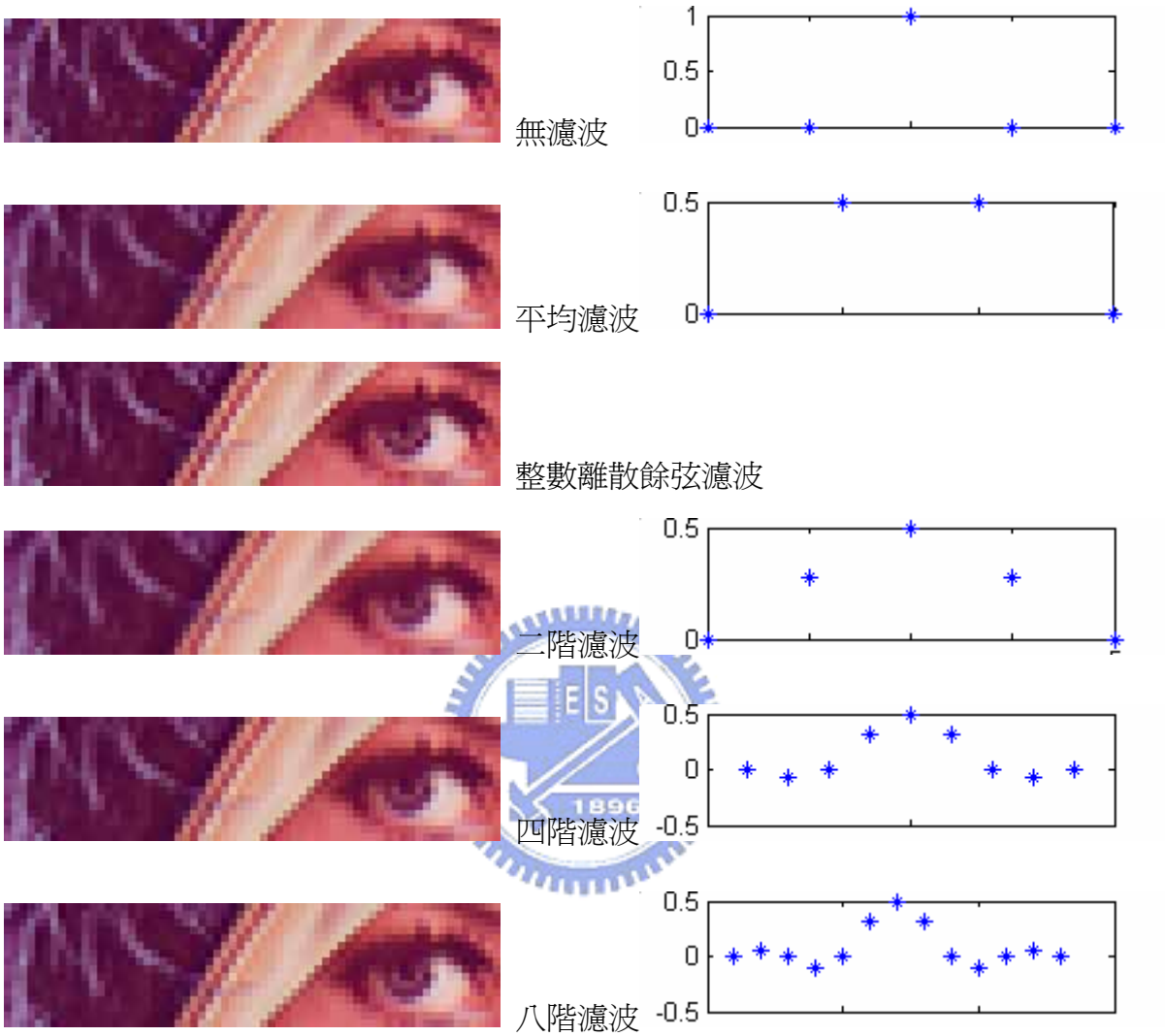


圖 4.2 縮減濾波細節比較

經過局部放大的細部比較之後，我們仍然很難觀察出各種濾波方式的差異，所以我們嘗試以下兩種方式，來佐證離散餘弦濾波是可行的縮減取樣濾波方式。

一. 快速翻頁之比對：

把圖片併在一起無法辨別，於是我們把圖片快速的在顯示器上切換，以突顯出相同部位的細微差異，然而這一部分的模擬結果無法在論文上表現。下面列出主觀性的比較結果：

- 無濾波縮減取樣的影像很銳利，但是有點不自然。
- 平均濾波縮減取樣的結果稍微有點模糊。
- 遮罩濾波縮減取樣的影像比較自然；階數低的時候畫面稍微有些模糊，但八階以

後幾乎沒有差異，都很清晰自然。

- 離散餘弦轉換濾波的影像比遮罩濾波縮減取樣的結果更清晰，卻不會像無濾波縮減取樣有不自然的感覺。而且，完全觀察不到區塊效應(Block Effect)。

二. 理論上的佐證：

雖然數位影像的頻譜難以精確定義，但以比較的觀點來說，無濾波與平均濾波的干擾失真很嚴重，但因爲一般影像高頻部分能量小，所以在上面的比較中不易發現。而遮罩濾波的干擾失真與階數成反相關，階數越高，干擾失真越小。

而離散餘弦濾波主要的缺失在於區塊爲基礎的濾波；在每個區塊內，離散餘弦濾波的干擾失真並不比遮罩濾波嚴重，但是以區塊爲基礎，便已經破壞整張影像其空間性與頻率性的對應關係。換句話說，這就是「離散小波轉換」(Discrete Wavelet Transform)比「區塊基礎之離散餘弦轉換」優越之處；遮罩濾波縮減取樣與離散小波轉換的過程相似，顧及了頻率資訊的區域性；而離散餘弦濾波縮減取樣由於計算精確度與複雜度的考量，必須先把畫面切割爲區塊，進而擾亂了頻率資訊的區域性，如果每個區塊經過轉換與反轉換之後有足夠的差異，由於每個區塊的差異各自不同，我們將會在還原的影像觀察到區塊效應。

但是，觀察了濾波取樣的結果，我們沒有發現人眼可感知的區塊效應。

另一方面，本篇論文的目的是在 MPEG-4 視訊編碼中必須的離散餘弦轉換中，同時做空間資訊的分離，意即，即使是金字塔型空間解析度可調層次式編碼，也會有以區塊爲基礎的離散餘弦轉換運算，而且運算執行的位置與我們的方法相同。所以我們相信採用離散餘弦轉換空間可調層次式編碼，並不會使區塊效應惡化(其區塊效應的程度與原本相同)。

綜合以上兩點，我們認爲利用離散餘弦轉換達到空間資訊的分離，是空間解析度可調層次式編碼可採用的一種方式。

4.1.2 以離散餘弦轉換係數分割達到空間解析度可調層次式

金字塔型空間解析度可調層次式在編碼之前，以縮減取樣，先形成不同解析的的連續影像，然後個別編碼；換句話說，在編碼之前，便已經完成空間資訊的分離。

既然離散餘弦轉換可以把頻率資訊分開，而且是 MPEG 編碼的必要運算，那麼我們便可以利用離散餘弦轉換達到空間資訊的分離，而且在編碼的過程即可做到。以下我們詳述如何配合 MPEG 之編碼解碼過程，以離散餘弦轉換係數的分離達到空間解析度可調層次式。

金字塔型空間解析度可調層次式在編碼前先以整張影像爲基礎形成金字塔影像架構，再分別編碼；我們提出的「離散餘弦轉換空間解析度可調層次式」是在編碼的過程中達成空間解析度可調層次式，而且是以區塊爲基礎。換句話說，我們可調層次式每個區塊的大

小。編碼端以 8x8 的區塊爲單位進行編碼，而解碼端可接收 2x2、4x4 或 8x8 的區塊資訊，分別還原成 2x2、4x4 或 8x8 的區塊，每個區塊的大小被可調層次式，則整個畫面的大小也跟著被可調層次式。

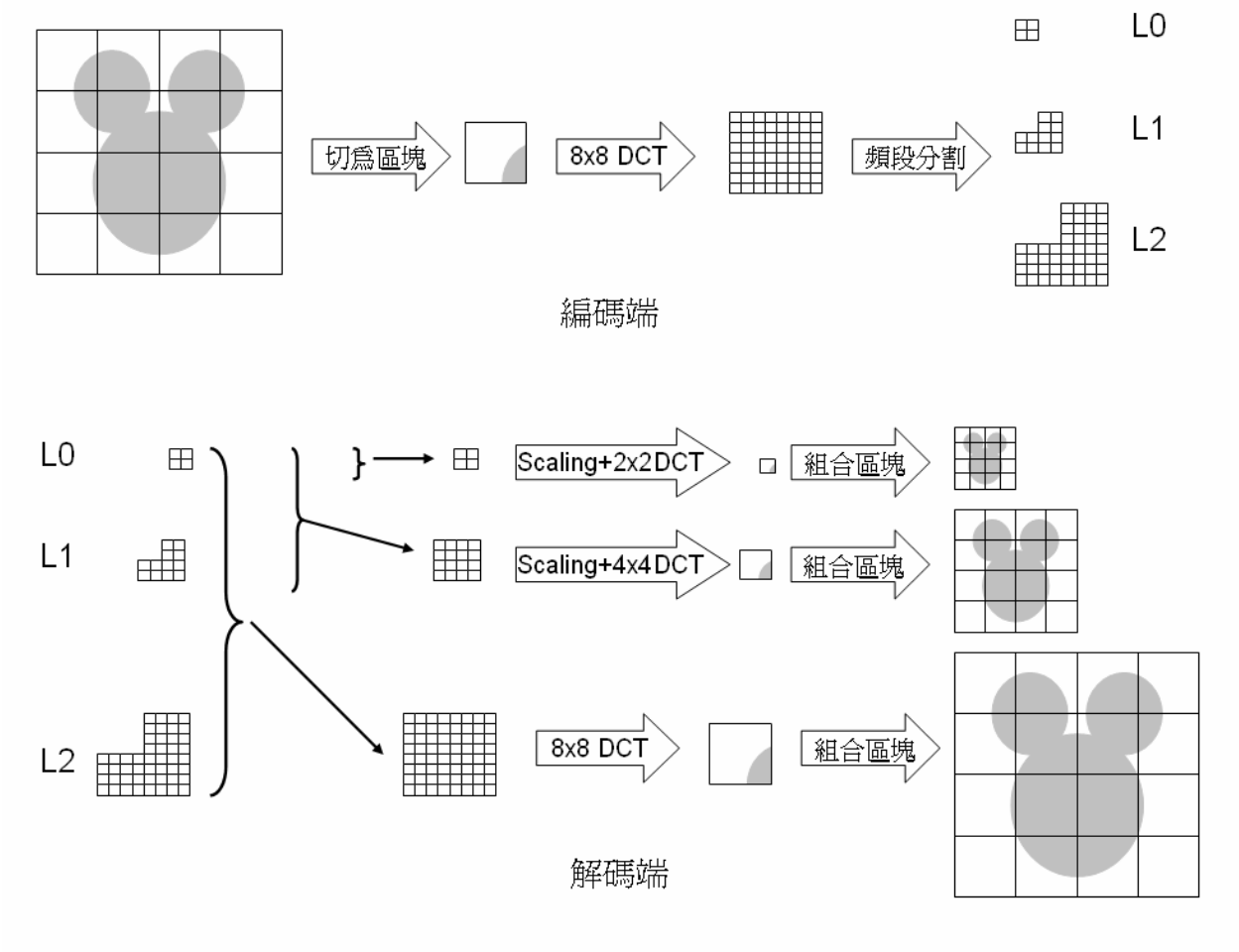


圖 4.3 分割離散餘弦轉換係數達到空間解析度可調層次式

4.2 離散餘弦轉換係數處理

靠著分割離散餘弦轉換係數，我們可以達到空間解析度可調層次式。而分割保留的低頻係數，要做縮小尺寸的離散餘弦反轉換，必須要經過一些處理，細節與原理詳述如下。

4.2.1 離散餘弦轉換係數

離散餘弦轉換是多媒體壓縮常用的頻域轉換，其優點在於集中能量於低頻，便利後續的壓縮動作。

4.2.1.1 離散餘弦轉換

長度 8 的一維離散餘弦轉換(1x8 DCT)的公式如下：

正轉換 $$f(j) = a(j)\sum_{k=0}^{N-1} x(k)\cos[\frac{(2k+1)j\pi}{2}]$$

反轉換 $$y(k) = \sum_{j=0}^{N-1} a(j)f(j)\cos[\frac{(2k+1)j\pi}{2}]$$

$$a(u) = \sqrt{\frac{1}{N}}, u = 0$$

$$a(u) = \sqrt{\frac{2}{N}}, u = 1,2,..N-1$$

上述公式把八個時間點的資料轉換成八個頻率係數。我們用基底變換(Basis transform)的角度來看：每個頻率係數分別是以八個數字與時間域資料相乘並總合，此八個數字在時間軸排開即是此頻率係數代表的基底函數(Basis Function)；對應相乘並求和的動作是求投影量。

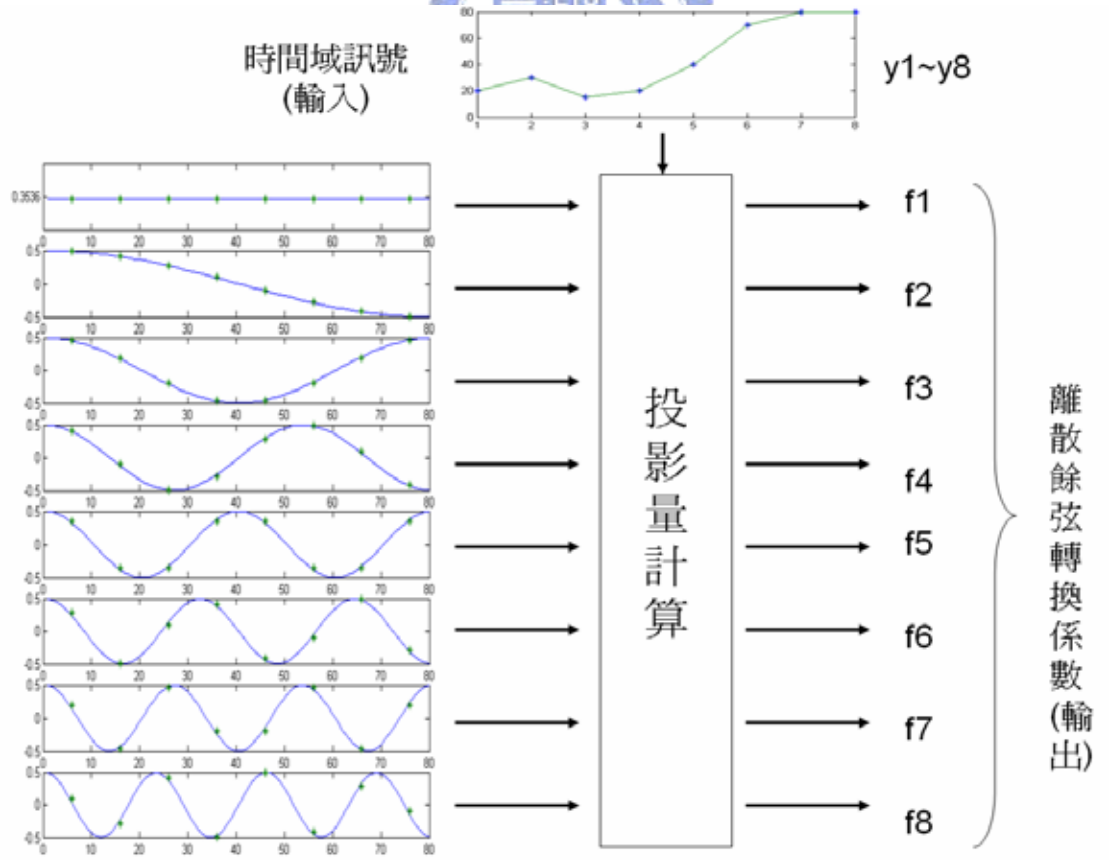


圖 4. 離散餘弦轉換示意圖

從轉換公式中，或是從上圖之中，可得知，每個基底函數所持有的八個數字，正是其對應的特定頻率的餘弦函數的八個取樣點。

4.2.1.2 不同解析度之間的係數對應

比較 1x8 的離散餘弦轉換與 1x4 的離散餘弦轉換：

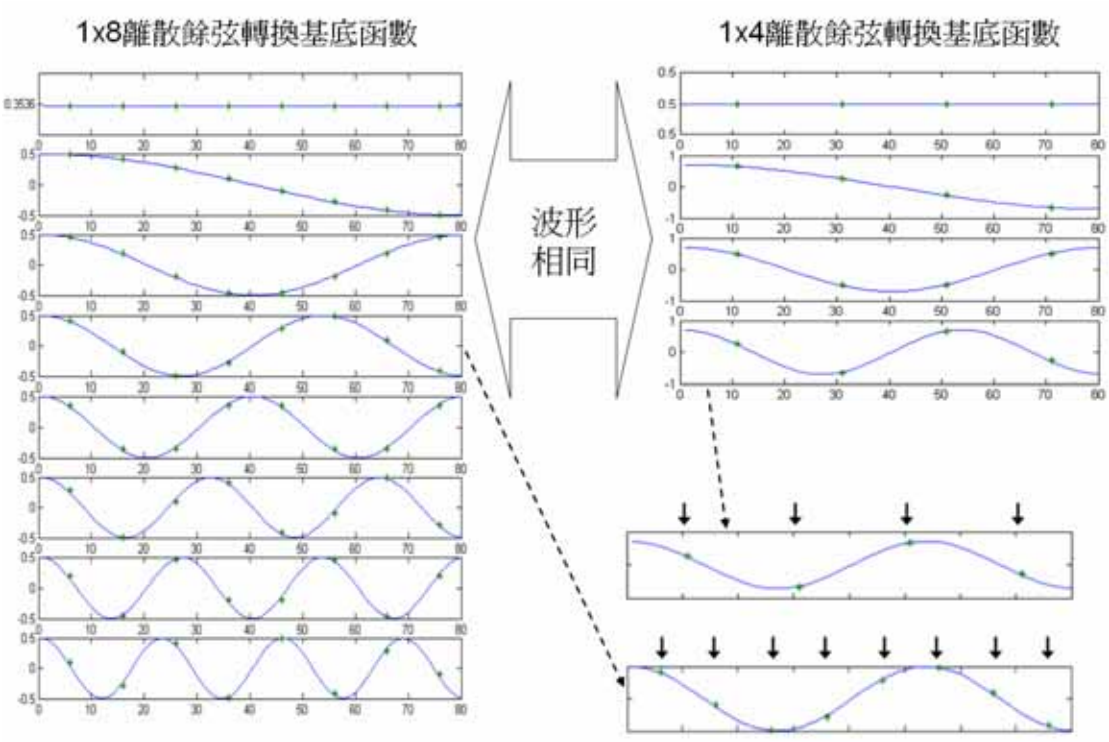


圖 4.5 不同尺寸離散餘弦轉換基底函數之比較

1x8 離散餘弦轉換的前面四個低頻基底函數，與 1x4 離散餘弦轉換的四個基底函數形狀相同(重複週期相同)，抽出一雙對應的基底函數比較，發現其取樣點正好錯開。1x4 離散餘弦轉換的第四個基底函數具有四個取樣點；1x8 離散餘弦轉換的第四個基底具有八個取樣點，可分成四對；前者的每一個取樣點正好安插在後者的每一雙之間。

進一步比較兩者的轉換公式：

1x8： $f(3) = a(3)\sum_{k=0}^{8-1} x(k)\cos[\frac{(2k+1)3\pi}{16}]$,

cos 取樣相位：[3/16 9/16 15/16 21/16 27/16 33/16 39/16 45/16]* π

1x4： $f(3) = a(3)\sum_{k=0}^{4-1} x(k)\cos[\frac{(2k+1)3\pi}{8}]$,

cos 取樣相位：[6/16 18/16 30/16 42/16]* π

對於目前所選定的基底函數，取樣點錯開(3π/16)。這個情形可推廣至所有(四個)對應的基底函數，對於四雙對應的基底函數(k=0、1、2、3)，取樣點相位分別錯開 (kπ/16) 。

4.2.1.3 不同解析度之間的係數轉換

因爲解碼的餘弦轉換尺寸與編碼的餘弦轉換尺寸可能不同，我們必須能夠調整原先的餘弦轉換係數，以配合編碼端的轉換尺寸。(如果尺吋相同則不需調整)

我們以具體的實例分析這個問題：編碼端 8x8 餘弦轉換的係數，取了其中的 4x4 低頻後，在作 4x4 反轉換之前，必須如何調整？ (8x8 轉 2x2 的過程可類推)

開始之前，我們定義一些符號方便推導：

1x8 離散餘弦轉換公式 $f(j)=a(j)\sum_{k=0}^{7}x(k)\cos[\frac{(2k+1)j\pi}{16}]$，

$$a(u)=\sqrt{\frac{1}{N}},u=0$$
$$a(u)=\sqrt{\frac{2}{N}},u=1,2,..N-1$$

另設 $a_0^8=\sqrt{\frac{1}{8}}$ ， $a_1^8=...=a_7^8=\sqrt{\frac{2}{8}}$ ， $b_{jk}^8=\cos[\frac{(2k+1)j\pi}{16}]$ ， $c_{jk}^8=a_j^8*b_{jk}^8$ ；

故 $f(j)=\sum_{k=0}^{N-1}x(k)c_{jk}^8$

同樣的，1x4 離散餘弦轉換公式可寫爲 $f(j)=\sum_{k=0}^{3}x(k)c_{jk}^4$

其中 $c_{jk}^4=a_j^4*b_j^4$ ； $b_{jk}^4=\cos[\frac{(2k+1)j\pi}{8}]$ ； $a_0^4=\sqrt{\frac{1}{8}}$ ， $a_1^4=...=a_3^4=\sqrt{\frac{2}{8}}$ ；

首先，我們先討論一維餘弦轉換的情況，即討論 1x8 餘弦轉換的係數如何調整爲 1x4 反轉換所需的係數：

考慮 1x8 餘弦轉換的前四個係數，與 1x4 餘弦轉換的四個係數的形成

在 1x8 餘弦轉換的前四個係數都有八個取樣比例→ $C_j^8=[\,c_{j0}^8\ c_{j1}^8\ c_{j2}^8\ c_{j3}^8\ c_{j4}^8\ c_{j5}^8\ c_{j6}^8\ c_{j7}^8\,]$

此八個比例常數對應八個影像值 → $X=[\,x_0\ x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\,]$

轉換係數即是此兩者的內積 → $f_j^8=X\bullet C_j^8$

在 1x4 餘弦轉換的每個係數都有四個取樣比例 → $C_j^4=[\,c_{j0}^4\ c_{j1}^4\ c_{j2}^4\ c_{j3}^4\,]$

理想中這四個比例常數對應四個影像值 → $Y=[\,y_0\ y_1\ y_2\ y_3\,]$

轉換係數即是此兩者的內積 → $f_j^4 = Y \bullet C_j^4$

Y 即是 X 縮減取樣的結果，而我們可以將兩者的係數轉換的基礎來自於－

$$s_j * f_j^8 = f_j^4 \rightarrow s_j *(X \bullet C_j^8) = (Y \bullet C_j^4)$$

s_j 爲一切換常數，不同頻率係數(不同的 j)的切換有著不同的數值

接下來用實際的切換過程說明 c 的角色：

原始訊號上的八個取樣點 → $[\, x_0\ x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\,] = X$

經過 1x8 餘弦轉換後的係數 → $[\, f_0^8\ f_1^8\ f_2^8\ f_3^8\ f_4^8\ f_5^8\ f_6^8\ f_7^8\,]$

四個切換常數 → $[\, s_0\ s_1\ s_2\ s_3\,] = S_{8\to4}$

[a0~a3]與[c0~c3]逐項相乘 → $[\, g_0^4\ g_1^4\ g_2^4\ g_3^4\,] = G^4$

再經過 1x4 的反轉換 → $[\, y_0\ y_1\ y_2\ y_3\,] = Y$

而 Y 是 X 的縮減取樣結果；G^4 則是對應的 1x4 餘弦轉換係數；S 則是我們要求出的切換常數。

$$f_j^8 = X \bullet C_j^8 = x_0 * c_{j0}^8 + x_1 * c_{j1}^8 + x_2 * c_{j2}^8 + x_3 * c_{j3}^8 + x_4 * c_{j4}^8 + x_5 * c_{j5}^8 + x_6 * c_{j6}^8 + x_7 * c_{j7}^8$$

$$= a_j^8 *[x_0 * b_{j0}^8 + x_1 * b_{j1}^8 + x_2 * b_{j2}^8 + x_3 * b_{j3}^8 + x_4 * b_{j4}^8 + x_5 * b_{j5}^8 + x_6 * b_{j6}^8 + x_7 * b_{j7}^8]$$

$$f_j^4 = Y \bullet C_j^4 = y_0 * c_{j0}^4 + y_1 * c_{j1}^4 + y_2 * c_{j2}^4 + y_3 * c_{j3}^4$$

$$= a_j^4 *[y_0 * b_{j0}^4 + y_1 * b_{j1}^4 + y_2 * b_{j2}^4 + y_3 * b_{j3}^4]$$

$$c_j = \frac{f_j^4}{f_j^8} = Y \bullet C_j^4 = \frac{a_j^4}{a_j^8} * \frac{y_0 * b_{j0}^4 + y_1 * b_{j1}^4 + y_2 * b_{j2}^4 + y_3 * b_{j3}^4}{x_0 * b_{j0}^8 + x_1 * b_{j1}^8 + x_2 * b_{j2}^8 + x_3 * b_{j3}^8 + x_4 * b_{j4}^8 + x_5 * b_{j5}^8 + x_6 * b_{j6}^8 + x_7 * b_{j7}^8}$$

$$s_{j1} = \frac{a_j^4}{a_j^8} = \sqrt{2}$$

令

$$s_{j2} = \frac{y_0 * b_{j0}^4 + y_1 * b_{j1}^4 + y_2 * b_{j2}^4 + y_3 * b_{j3}^4}{x_0 * b_{j0}^8 + x_1 * b_{j1}^8 + x_2 * b_{j2}^8 + x_3 * b_{j3}^8 + x_4 * b_{j4}^8 + x_5 * b_{j5}^8 + x_6 * b_{j6}^8 + x_7 * b_{j7}^8}$$

考慮 $x_0 \cong y_0 \cong x_1$ ， $x_2 \cong y_1 \cong x_3$ ， $x_4 \cong y_2 \cong x_5$ ， $x_6 \cong y_3 \cong x_7$

且 $$\frac{b_{j0}^4}{b_{j0}^8 + b_{j1}^8} = \frac{b_{j1}^4}{b_{j2}^8 + b_{j3}^8} = \frac{b_{j2}^4}{b_{j4}^8 + b_{j5}^8} = \frac{b_{j3}^4}{b_{j6}^8 + b_{j7}^8} = \frac{1}{2 * \cos(j\pi/16)}$$

故 $s_{j2} = \frac{1}{2 * \cos(k\pi/16)}$ ， $s_j = s_{j1} * s_{j2} = \frac{\sqrt{2}}{2 * \cos(k\pi/16)}$

可得 $S_{8\to4}$=[0.7071 0.7210 0.7654 0.8504]

擴充爲二維切換係數

$S_{8x8\to4x4} = S_{8\to4}{}' * S_{8\to4}$ = [

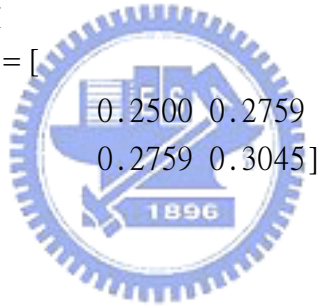
0.5000 0.5098 0.5412 0.6013
0.5098 0.5198 0.5518 0.6131
0.5412 0.5518 0.5858 0.6509
0.6013 0.6131 0.6509 0.7232]

同理 $S_{8\to2}$ ==[0.5000 0.5519]

擴充爲二維切換係數

$S_{8x8\to2x2} = S_{8\to2}{}' * S_{8\to2}$ = [

0.2500 0.2759
0.2759 0.3045]



4.2.2 三階層空間解析度可調層次式的實作

4.2.2.1 整體架構

本篇論文提出架構在 MPEG-4 上，具有三層空間解析度可調層次式(3-Level Spatial Scalability)的視訊編碼，其整體架構如下圖：

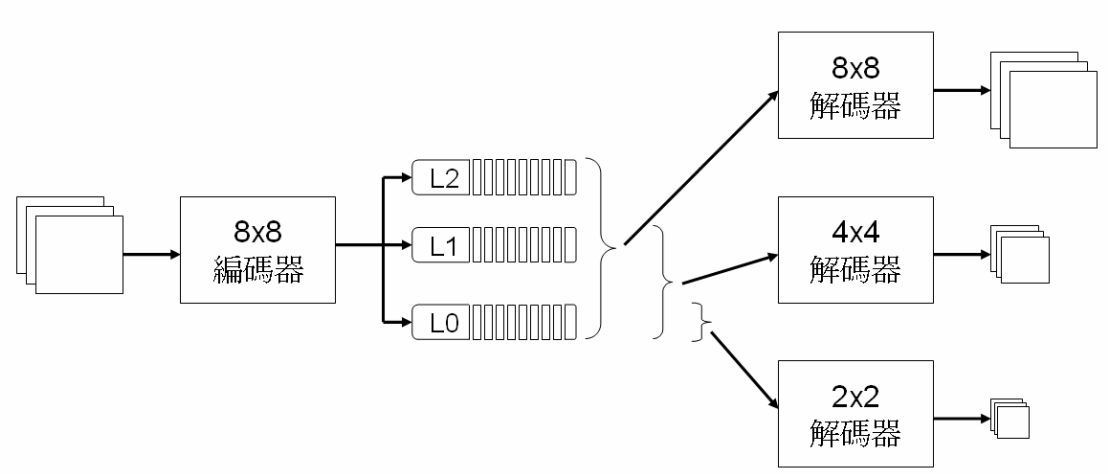


圖 4.6 離散餘弦分割空間解析度可調層次式整體架構

「8x8 解碼器」代表其內的運算是以 8x8 的區塊爲基礎；解碼器有三種，8x8、4x4、2x2 分別代表其基礎區塊的大小，三者分別還原出高解析度、中解析度、低解析度的視訊。

與金字塔型空間解析度可調層次式編碼架構上主要不同是：

1. 我們只需要一個 MPEG-4 編碼器，其中某些運算單元需要修改。而金字塔型空間解析度可調層次式編碼需要三個 MPEG-4 編碼器。

2. 高解析與中解析解碼器中，只需要一個 MPEG-4 解碼器，其中某些運算單元需要修改。而在金字塔型空間解析度可調層次式編碼中，分別需要三個與兩個 MPEG-4 解碼器。

4.2.2.2 編碼器

編碼器的架構與第三章頻率資訊選擇編碼能力的編碼器一樣，能把離散餘弦轉換係數分成三個頻段編碼。細節請參照 3.1.2。

4.2.2.3 解碼器

解碼器的架構變動比較多，而且每一個解析度的解碼器架構稍有不同。所有的變動可以分成兩類：

1. 鋸齒狀掃描與變動長度編碼
2. 因應畫面大小變化所做的調整

第一項完全與編碼器對應；第二項包含數項內容，因爲在編碼器端一個 8x8 的區塊，在解碼器端會以 8x8、4x4、2x2 還原，故在很多步驟都有影響。接下來針對每一個步驟詳細說明：

- 所接收的離散餘弦轉換係數：

  8x8 離散餘弦轉換係數，要以 2x2 或 4x4 做反轉換之前，需要先做比例調整。下圖以 L1 解碼器爲例，說明調整步驟：

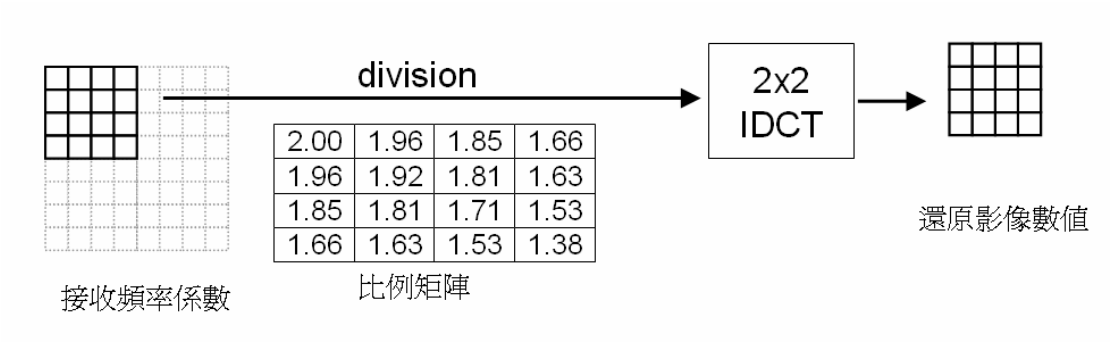


圖 4.7 離散餘弦係數比例調整

- 離散餘弦轉換：

  8x8、4x4、2x2 解碼器分別使用 8x8、4x4、2x2 離散餘弦反轉換。

- 移動向量(Motion Vector)：

在 4x4 與 2x2 解碼器中，每一個影像區塊皆被縮小，故在做移動補償時，移動向量也對應縮小。4x4 解碼器必須把接收到的移動向量除以 2；2x2 解碼器必須除以 4，不滿 1 的無條件捨去。

意即，在 4x4 解碼器與 2x2 解碼器中，移動向量必須調整，我們目前採用的方法是直接做位元切斷(Truncation)。

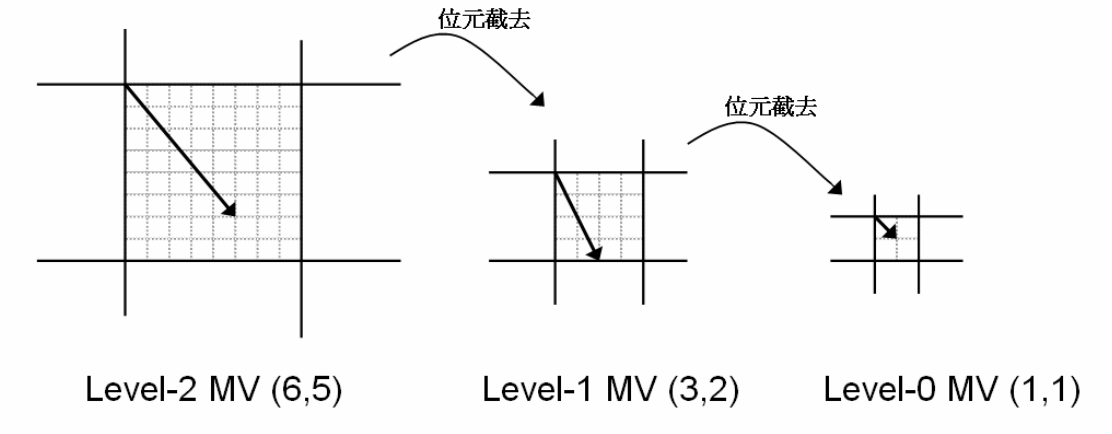


圖 4.8 移動向量調整

4.3 空間可調層次式編碼的延伸議題

我們所提出的頻域分割空間解析度度可調層次式方法達到空間解析度度可調層次式的要求，並且提供了良好的彈性(flexibility)，但同時也帶來了一些不理想的改變，主要是還不能完美的符合視訊編碼的需求，以下我們將逐一提出，並且討論解決的方法。

4.3.1 各層編碼效率

以最高解析度來看，無論是基本層或是加強層，我們的架構編碼效率緊跟 MPEG4；但是以低解析度或中解析度的角度來看，其編碼效率相當不理想。

表 4.1 空間解析度可調層次式各層編碼效率

		高解析度		中解析度		低解析度	
		基本層	加強層	基本層	加強層	基本層	加強層
aki	MPEG4	49,504	13,819,437	14,025	4,585,091	5,159	1,345,677
	SS-1	49,510	14,260,334	49,260	5,401,728	46,732	1,556,443
	increase(%)	0	3	251	18	806	16
fman	MPEG4	280,708	17,759,694	70,429	5,372,678	18,524	1,456,402
	SS-1	280,712	18,367,627	280,227	6,344,904	275,815	1,676,563
	increase(%)	0	3	298	18	1,389	15
mob	MPEG4	386,390	25,187,222	112,701	7,733,961	19,272	1,942,759
	SS-1	386,389	25,959,742	369,130	8,206,993	316,964	1,901,624
	increase(%)	0	3	228	6	1,545	-2
total	MPEG4	716,602	56,766,353	197,155	17,691,730	42,955	4,744,838
	SS-1	716,611	58,587,703	698,617	19,953,625	639,511	5,134,630
	increase(%)	0	3	254	13	1,389	8

參考一般對於空間解析度可調層次式的要求，我們的架構的編碼效率在兩處有著不可接受的低編碼效率，分別是”中解析度基本層”與”高解析度基本層”，編碼效率差了兩倍與十三倍以上。

其原因在於，我們的架構在三個解析度的區塊彼此對應，在高解析度的 8x8 區塊，在中解析度變成 4x4，在低解析度變成 2x2；所以在中解析度與低解析度中，我們相當於用 4x4 和 2x2 的區塊大小進行編碼。過小的區塊會造成編碼效率的低落。

加強層的表現較好，其原因在於我們有針對中低頻段的加強層統計特性設計合適的熵編碼。而基本層仍然採用與高解析度一樣的編碼方式(只是把中高頻段編碼符號留在別的串流)，所以編碼效果非常低落。

進一步分析其資料分布，我們可以更精確的說出編碼效率低落的因素：

- 移動向量資料量大

  中頻段的區塊數目爲 MPEG4 的 4 倍，低頻段的區塊數目爲 MPEG4 的 16 倍，越多區塊，需要越多的移動向量。即使移動向量已經過預測，還是需要很大的資料量。

- 可變長度編碼在中低頻段並無建樹

  可變長度編碼最主要的能力在於把連續的零以很少位元的符號作有效率的編碼，然而中低頻段只負責傳送原本 8x8 區塊的低頻部分，而此部分的零不多，故無法做有效率的編碼。追根究柢，其原因在於過小的區塊上作餘弦轉換，沒能有效的把低頻資訊聚集，低頻資訊聚集不佳，自然不會在高頻段留下連續零。例如：8x8 區塊的 DC 紀錄了 64 個畫素的平均，但是 2x2 區塊的 DC 只紀錄了 4 個畫素的平均。

由於缺陷來自於過小的區塊，但是區塊的對應性是架構的主體，故必須在維持小區塊的情況下改善小區塊的缺點。故可以朝「聯合區塊」的方式尋求解決之道，即聯合多個區塊，一起做移動補償與可變長度編碼。

4.3.2 移動補償造成的錯誤遺留(MC error drift)

由於我們的架構採用「一致的移動補償」(詳見 2.4)，並且以高解析爲主，故在中低解析度的移動補償將會造成錯誤遺留。

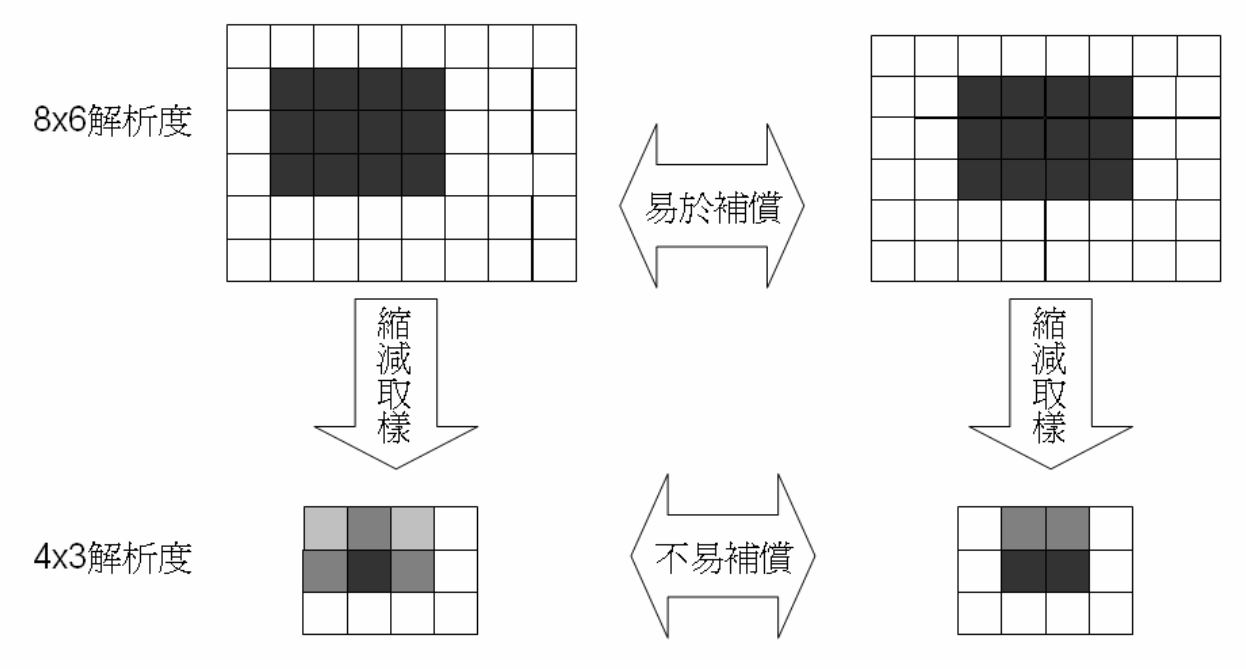


圖 4.9 移動補償之錯誤遺留

上圖用了粗糙但實際的例子說明了縮減取樣後移動補償不匹配的可能。而且不匹配所造成的錯誤如果又被當作下一次移動補償的參考區域，則此錯誤將會留在畫面中，而造成錯誤遺留。

而造成錯誤的原因有二：

- 解析度降低細節消失

  高解析度中明顯的黑白邊界可能在低解析度中變成灰色，也可能不會；如果這一張影像是清晰黑白邊界，但參考影像爲灰色邊界，則低解析度將無法如同高解析度適當的作良好的移動補償。

- 移動向量變得粗糙

  MPEG4 目前採用 1/2 畫素的移動預測及補償，如果某個區塊在高解析度移動向量爲 3.5，則到了低解析度，理論上應該使用 1.75 的移動向量，但 MPEG4 只提供 1/2 畫素的移動補償，我們必須犧牲其移動向量精確度，例如無條件捨去多餘的 0.25 而採用 1.5 作移動補償，如此一來便造成物體移動距離的失真。

針對這兩個問題，可能的改善方式如下：

1. 考慮採用獨立的移動補償，並且配合多迴圈的編碼(multi-loop)。所謂多迴圈的編碼，指每個解析度有獨立的移動補償與補償誤差的進一步編碼。因爲沒有任何一個解析度能夠完全模擬其他解析度的移動補償誤差，故必須有獨立編碼過程，但是先編碼的解析度其資料可供其他解析度作參考或預測。

2. 高解析度中採用 MPEG4 原本的 1/2 畫素移動預測及補償；在中解析度(不需

作預測)採用 1/4 畫素的移動補償；在低解析度採用 1/8 畫素的移動補償。

4.4 前瞻發展

我們提出的方法，可以達到目前空間解析度可調層次式的普遍要求。然而，由於其架構具有良好彈性，我們還可以推廣至更前瞻的功能。

1.4.1 多層空間解析度可調層次式

金字塔架構裡的每一層解析度皆需要幾乎完整的編碼程序，如果要把空間可調層次式增加至多層，相對的會增加其計算量；而且需要不同的縮減取樣遮罩。而我們的架構單純的在離散餘弦轉換係數上作切割，編碼解碼皆很方便，故我們可以輕易的達到多層空間解析度可調層次式，剩下的就是如何對分割過的係數作良好的壓縮。

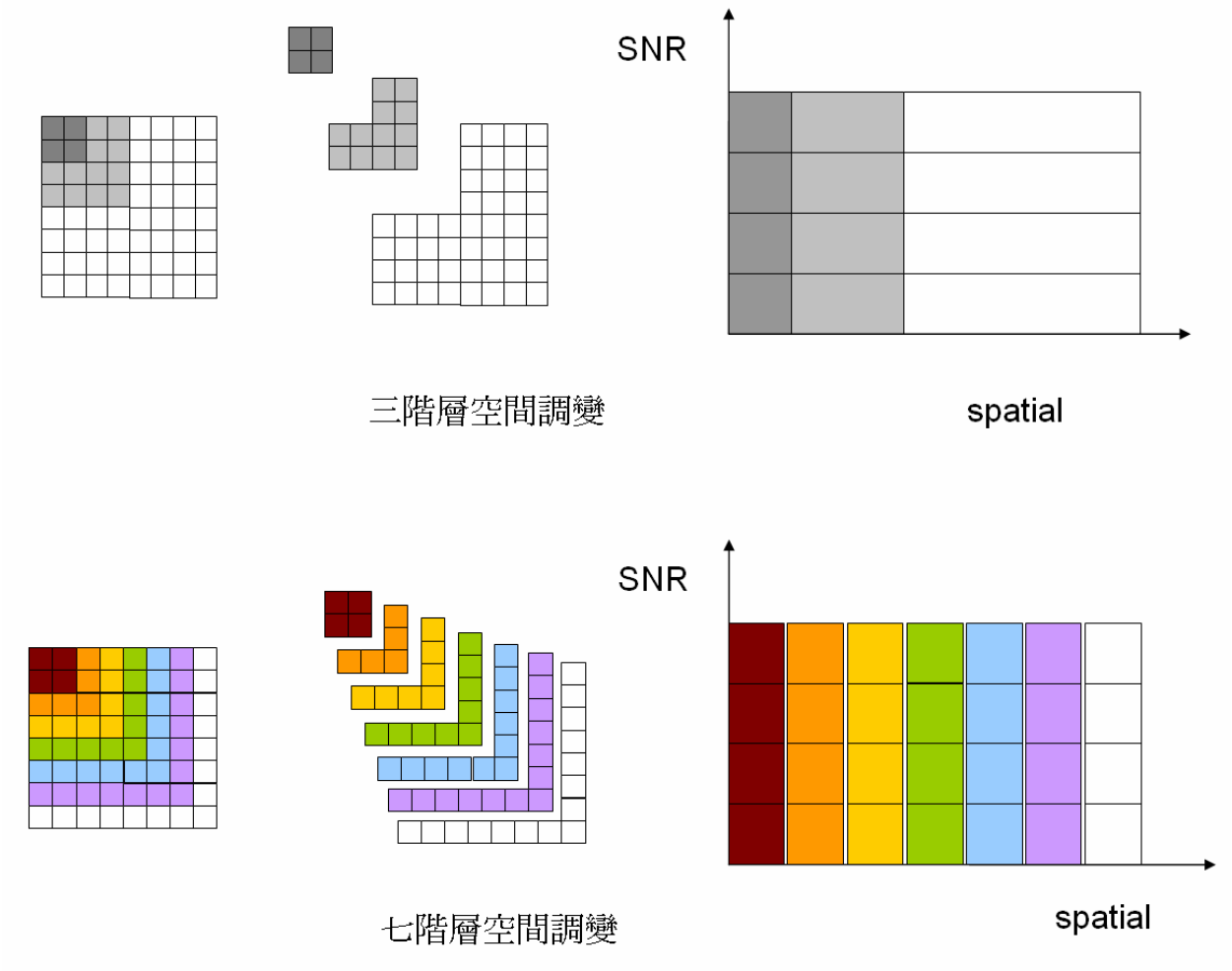


圖 4.10 多階層空間解析度可調層次式