原島 博 監修

電子情報通信学会 編

影像壓縮技術
映像情報符号化

酒井善則・吉田俊之 共著

白執善 編譯

Ohmsha

全華科技圖書股份有限公司 印行

(a) L = 8 位元

(b) L = 6 位元

(c) L = 4 位元

(d) L = 2 位元

圖 1.11　L = 2、4、6、8 位元的量子化後影像實例

(a) 4 : 4 : 4

(b) 4 : 2 : 2

(c) 4 : 2 : 0

(d) 僅 Y 方向在水平垂直各 1/2

1.21　YC_bC_r各分量減樣的影響

675840 位元
(a) 原影像 (單色 8bpp)

129984 位元
(b) 解碼影像

圖 2.2　對「花園」第一畫格進行壓縮編碼之例

(a) 4×4 (1.2bpp, 33.9dB)

(b) 4×4 (0.4bpp, 21.2dB)

(c) 8×8 (1.2bpp, 34.8dB)

(d) 8×8 (0.4bpp, 21.5dB)

(e) 16×16 (1.2bpp, 34.2dB)

(f) 16×16 (0.4bpp, 21.5dB)

(g) 32×32 (1.2bpp, 33.2dB)

(h) 32×32 (0.4bpp, 21.1dB)

圖 6.11　使用不同區塊尺寸進行 DCT 後之編碼、解碼結果

發刊詞

這幾十年來，電子技術有著飛躍性的發展。但是，技術上的成熟發展，在一般的使用者的立場看來，就是逐漸滿足「食衣住」的需求。滿足了食衣住的需求之後的下一步便是「創造文化」。以電子通訊爲代表的媒體技術，是不是正在創造往後的通訊文化呢？

「人際通訊工程叢書」大體上來說，就是受此觀念啓發的企劃。目前爲止的通訊技術，基本上都是像電話機般的終端機與終端機之間的訊號傳遞技術。但是如果想得更深入，通訊眞正的的終端不就是人嗎。展望未來，不僅侷限於訊號傳遞技術，而是以人際通訊眞正終端的立場，創造出「服務人類的通訊支援技術」或是更進一步的「人本資訊環境技術」。

電子資訊通訊學會將這種技術領域命名爲「人際通訊工程」以獨立的研究社群蓬勃地發展著。本叢書也是其活動的一環，整理出由不同的的立場所得到的進路。無論何者都還不是系統化的技術領域。這表示本叢書並非集技術之大成者，而具有著可稱爲摸索的性格。希望本叢書能成爲一種契機，讓新技術得以發芽、茁壯。

前言

多媒體這個名詞開始流行已經超過了十年。現在可以說是 IT 的時代，但 IT 具有更勝於多媒體的產業基礎，這個時代應該能持續下去。爲了實現 IT 需仰賴許多技術。除了奠基於電腦的軟硬體技術。光通信技術、封包交換技術造就了高速的網際網路。感測、顯示等技術成就了影像媒體的輸入輸出；VLSI 技術則是讓以電腦、通訊裝置爲代表的許多機器得以面世的基礎。

在這些可見諸於表面上技術的同時，對 IT 時代之實現具有貢獻之訊號處理技術亦悄悄地進步著。訊號處理技術雖然用在聲音、影像的辯識處理等許多領域，但是其中花了數十年的研究，而在 IT 時代開花結果的技術爲編碼技術。編碼技術使得聲音、影像等多媒體資訊數位化，在此同時具有在將品質衰減壓抑在最低限度的基礎下壓縮資料量的功能。在其他方面，還同時可糾錯編碼，可以說由仙農(SHANON)開創的資訊理論最重要的應用即在於多媒體的編碼。

多媒體編碼的歷史總是伴隨著許多的迂迴曲折的經驗。直到發展出用於編碼的LSI技術能讓數位傳送普及之前，編碼技術鮮少受到注意。另一方面，光纖的發展造就了大容量通訊，亦產生了是否需要壓縮編碼技術的說法。但是不管大容量傳輸、大容量記憶體如何發展，資訊量仍不斷增加，因此編碼在今後仍然一直作爲必要且重要的基礎技術而發展。相對於 JPEG、MPEG 的標準化使得編碼技術的開發已經到達盡頭的意見，許多企業開發了聲稱較這些標準爲佳的編碼方式，在大眾媒體中亦時有所聞。

本書的目的在對於多媒體編碼，特別像是 JPEG、MPEG 等著名的影像編碼方法，其基礎理論與實際應用技術作簡單的解說。

第一章介紹關於數位影像所需最低限度的基礎知識。第二章則是透過簡單的編碼演算法概述影像壓縮編碼的觀念，同時說明關於「像素間相關程度」的重要概念。此外，依據「去除像素間相關程度—量子化—熵編碼」的流程闡明一般的影像編碼演算法。第三章與第四章分別簡單介紹熵編碼化觀念與量子化和量子化誤差的統計處理方法。第五章在許多去除像素間相關程度的方法中，詳細地說明 JPEG、MPEG 等常用的編碼方法。首先，說明最佳化轉換的Karhunen-Loeve轉換(KLT)之後，再推導其極限狀態：廣泛用於JPEG或是MPEG的離散餘弦轉換(DCT)。第六章將 DCT 具有的性質與數個例題一起介紹之後，再簡介實際應用於JPEG或是MPEG的DCT編碼方式。

第七章所著眼者則變爲動態影像的壓縮編碼方法。介紹目前使用的 MPEG 等動態推測／動態補償的觀念，並說明其具體的演算法。第八章，雖然討論有些偏理論化，但是對企圖對資料率－失眞理論作些許的介紹。藉此可從與影像編碼不同的角度來切入。最後第九、第十章分別爲在實際應用情形中，JPEG與MPEG的概要說明。

以JPEG或MPEG爲首的影像碼國際標準頗有日新月異之勢，現有者爲JPEG2000、MPEG-4，再來會有MPEG7、MPEG21等新的標準接踵而至。由本書的書名或是本系列的特性觀之，應該要將這些通通納進來，最好能將多媒體影像的所有編碼技術加以網羅。但是由於作者之力有限，到了推導出DCT爲止就花了百來頁的篇幅，連談到作爲JPEG2000基礎的小波轉換的餘力也沒有。僅就影像編碼而言，也頗有「仰之彌高」之感。惟儘可能地自基礎循序漸進，再儘可能地不要使得觀念有跳

得太快的情形下作介紹。因此先將小波轉換或是MPEG-4這些較難懂的篇幅給刪除說不定反而效果較好。

最後，本書記載的影像編碼技術在許多個人電腦、電視中都看得到，作爲在廣播、網際網路等影像處理的基礎技術。本書期待能爲許多讀者們在理解編碼的原理與流程時提供一些幫助。

監修者簡歷

原島　博

昭和 48 年，東京大學大學院博士，同年，同大學講師，昭和50年，同大學副教授，現任同大學大學院情報學環，學際情報學府教授，工學博士。

酒井　善則

昭和49年，東京大學大學院博士，同年任職電電公社電氣通訊研究所，昭和62年，東京工業大學大學副教授，平成 2 年升任教授，現任同大學大學院理工學研究所集積系統專攻教授，工學博士。

吉田　俊之

平成 3 年，東京工業大學大學院博士，同年，同大學助教，平成 6 年，同大學副教授，現任同大學大學院理工學研究所集積系統專攻助教授，工學博士。

譯者序

隨著電子科技的進步，電子產品的功能也愈來愈生活化。其中，首當其衝地映入我們眼簾的當屬圖像與動畫的應用。本書之重點即在介紹這些讓我們眼花撩亂的影像壓縮技術。不過尤為難能可貴的是本書說理相當地清楚，而且內容旁徵博引，如在說明的時候，還引用了著名的漫畫哆啦A夢(第一章、參考文獻[5])。對於欲一窺影像壓縮技術堂奧的讀者來說，這本書具有相當紮實的內容，有助於讀者建立清晰的觀念，同時也提供了相當豐富的參考文獻與網路資源。相信這本書不管對於教師、學生，都能夠當作相當好的學習工具。

感謝全華科技圖書給我這個機會能貢獻所學，也謝謝研發處孫先生與完稿的工作人員在翻譯校對過程中給予譯者的協助。最後也謝謝讀者購買本書，希望這本書能對各位在學習過程中有所助益。

譯者簡介：工學碩士，曾於日本進修兩年，目前從事寫作、翻譯、技術文件審訂工作。

編輯部序

「系統編輯」是我們的編輯方針，我們所提供給您的，絕不只是一本書，而是關於這門學問的所有知識，它們由淺入深，循序漸進。

本書循序漸進，對於多媒體編碼的基礎理論與實際應用技術作詳細的解說。第一章介紹關於數位影像所需最基礎知識，第二章則是透過簡單的編碼演算法概述影像壓縮編碼的觀念，第三、四章分別介紹熵編碼化觀念與量子化誤差的統計處理方法，第五章詳細地說明JPEG、MPEG等常用的編碼方法，第六章簡介實際應用於JPEG或是MPEG的DCT編碼方式，第七章著眼於動態影像壓縮編碼方法，第八章企圖對資料率的失眞理論作些許的介紹，最後第九、第十章則分別在實際應用情形中，對JPEG與MPEG的概要說明。本書能爲許多讀者們在理解編碼的原理與流程時提供一些幫助。

同時，爲了使您能有系統且循序漸進研習相關方面的叢書，我們以流程圖方式，出各有關圖書的閱讀順序，以減少您研習此門學問的摸索時間，並能對這門學問有完整的知識。若您在這方面有任何問題，歡迎來函連繫，我們將竭誠爲您服務。

相關叢書介紹

書號：03329017
書名：數位信號—影像與語音處理(附範例光碟)(修訂版)
編著：林宸生
20K/576 頁/440 元

書號：05490007
書名：語音訊號處理(附語音資料光碟片)
編著：王小川
20K/416 頁/380 元

書號：0380401
書名：數位信號處理的新利器 – TMS320C6X(修訂版)
編著：王逸如.陳信宏
16K/512 頁/520 元

書號：0525301
書名：資料壓縮的原理與應用(修訂版)
編著：鍾國亮
20K/472 頁/420 元

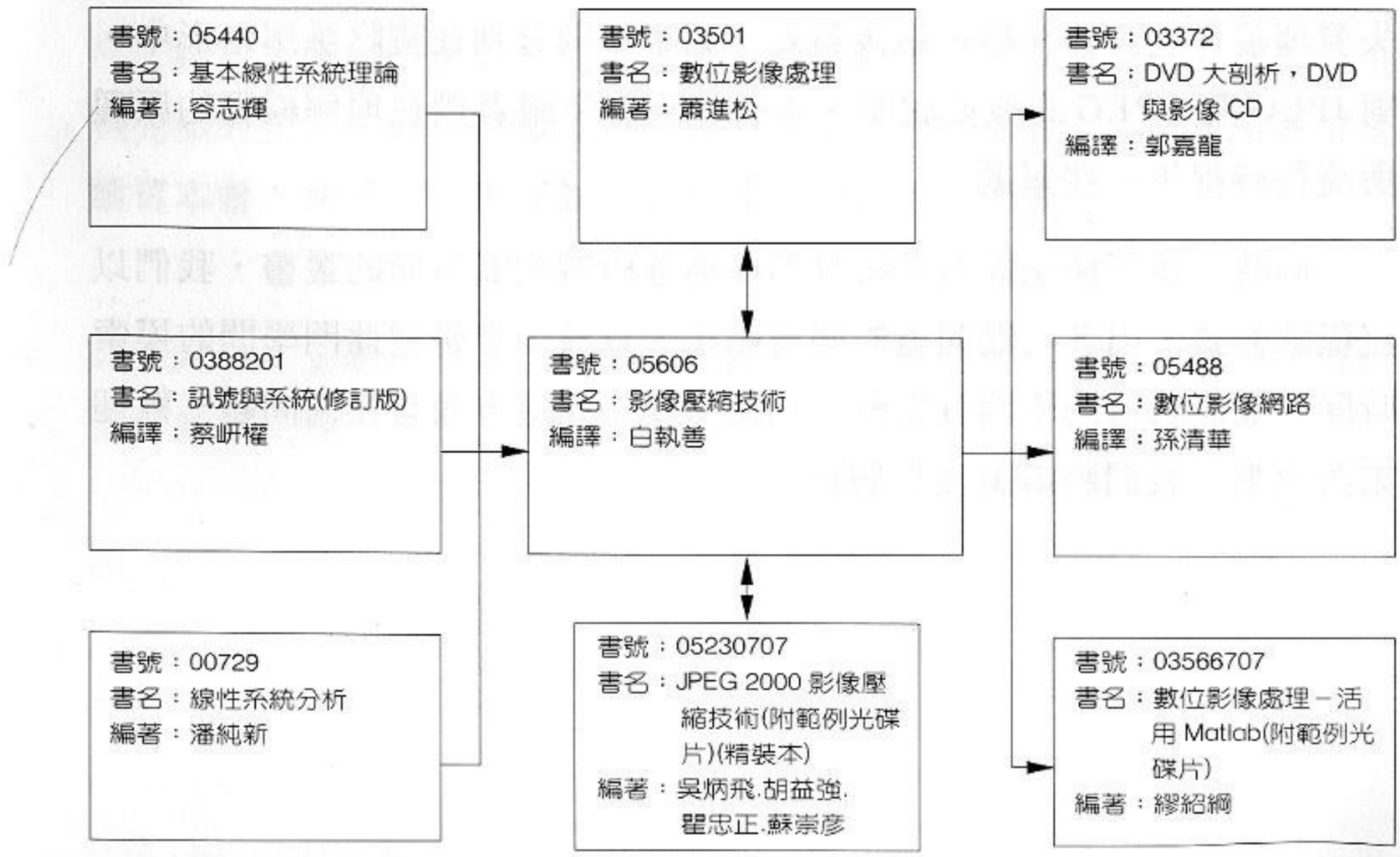
書號：03435017
書名：定點數位訊號處理器應用程式設計與實驗(修訂版)(附光碟片)
編著：胡竹生.陳榮偉.陳界全
20K/624 頁/550 元

書號：05125007
書名：訊號處理 – MATLAB 的應用(附範例光碟片)
編著：羅華強
20K/336 頁/350 元

書號：02574
書名：影像處理與辨認
編譯：楊武智
20K/296 頁/250 元

◎上列書價若有變動，請以最新定價為準。

流程圖



CHWA

目錄 CONTENTS

1 章 數位影像的基礎 1-1

何謂數位影像 1-1
各種數位影像的格式 1-17
關於影像的壓縮編碼與畫質的評估 1-24

2 章 影像壓縮編碼的基礎 2-1

像素值的寫法 2-1
何謂影像壓縮編碼 2-2
為什麼可以壓縮影像 2-5
影像壓縮編碼的原理—透過簡單的編碼演算法加以介紹 2-10
像素間相關程度的定量關係 2-14
影像編碼演算法的一般組成 2-20

3 章 熵編碼 3-1

前言—同時說明用語與基礎知識 3-1
解碼的條件 3-6
資訊來源編碼定理 3-9
Huffman 編碼 3-12
算數編碼 3-17

4 章 量子化—進行統計處理 4-1

前　言 4-1

各種量子化器 4-3

量子化誤差之統計處理 4-5

關於式(4.19) 4-12

結語－量子化與反量子化的實際操作 4-14

5 章 正交轉換與 DCT 5-1

線性轉換與正規化正交轉換 5-2

使用正交轉換之影像編碼演算法 5-7

巴塞瓦性質與其應用 5-9

轉換係數的最佳位元分配[35] 5-11

Karhunen-Loeve 轉換(KLT) 5-16

由 KLT 至 DCT －推導 DCT 5-33

結　語 5-37

6 章 DCT 之性質與其在編碼上的應用 6-1

DCT 的性質 6-1

編碼時的應用－使用二維 DCT 進行影像編碼的實際情形 6-8

結語－補遺 6-15

7 章 動畫影像之壓縮編碼與動態補償 7-1

動畫影像之基礎－關於掃描 7-1

動畫影像編碼之基本方法 7-3
使用動態補償之動畫影像編碼 7-8
實際的演算法 7-19

8 章 資料率－失真理論－影像編碼之理論處理入門8-1

資料率－失真理論之簡介 8-1
資料率－失真函數 8-3
為什麼可以壓縮影像 8-6
靜止影像與動畫影像之壓縮極限 8-10

9 章 多媒體編碼實務(1)JPEG－靜止影像之壓縮編碼 9-1

前　言 9-1
基本編碼演算法 9-2
JPEG 基線方式之位元串組成 9-8
JPEG 基線方式之編碼特性 9-16
結　語 9-18

10 章 多媒體編碼實務(2)MPEG－動畫影像之壓縮編碼 10-1

MPEG 的歷史－常識程度的介紹 10-1
MPEG-1 簡述 10-3
MPEG-1 組成 10-9

MPEG-2 簡述 10-39
結　語 10-49

參考文獻 參-1

索　引 索-1

1 章

數位影像的基礎

對影像編碼技術而言，「數位影像」的相關知識是必要的。因此，本章歸納後續章節所需關於數位影像的必要知識。在此方面有一定程度的讀者可以略過本章直接進入第二章。

何謂數位影像

影像感測的原理

物體表面的反射光通過凸透鏡之後，在適當的位置設置屏幕，在屏幕上會顯現出物體的影像(圖 1.1)，圖 1.2 爲藉由凸透鏡成像之一例。我們的眼睛或是攝影機等許多影像感測系統即奠基於圖 1.1 之原理，以攝影機爲例，即是將屏幕換成可將亮度值轉變爲電子訊號的元件組成。

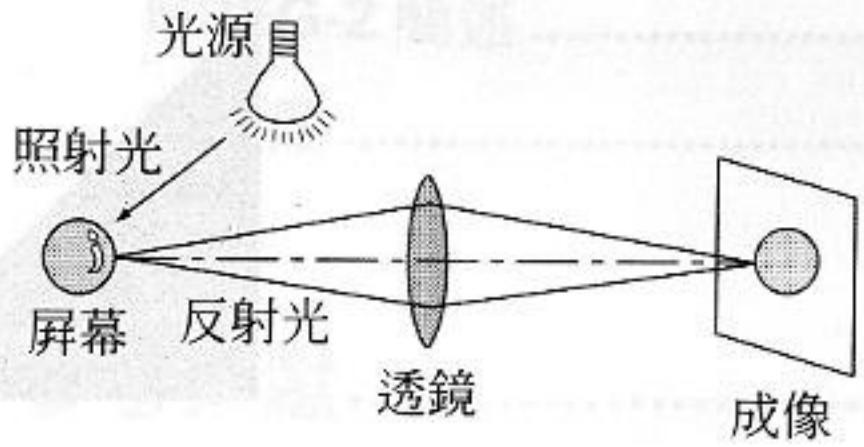


圖 1.1　藉由凸透鏡於屏幕上成像

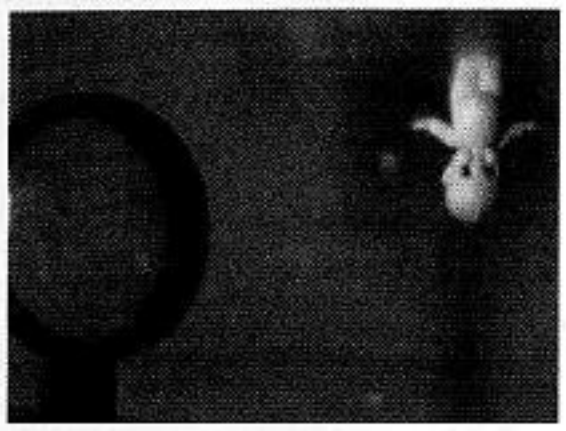


圖 1.2　屏幕上的影像實例

此外，圖 1.2 所示之物體影像具有：

- 在任意兩點間的任何一點，影像的亮度值固定。
- 各點的亮度值爲實數。

的性質。這些在一般的數位電腦上是無法記錄的。也就是說：

① 影像由無限個點所構成，爲了記錄各點的亮度，記憶體大小需爲無限大。

② 爲了記錄(表現)出以實數表示的各點亮度，資料精密度(位元長度)需要無限長。這種影像在空間方向與各點的亮度方向都需取連續值，稱之爲「類比影像」。

爲了在電腦上處理影像，作爲處理對象的類比影像非得被讀入電腦裏頭不可。這時爲了配合上述的①②兩點，需要做下列兩項操作。

- 處理影像，以有限個點的集合加以表現。
- 處理各點的亮度值，以有限精密度(位元長度)加以表現。

前者稱爲**空間取樣**，後者稱爲**量子化**。

空間取樣

首先，先介紹一維的訊號取樣[1]。考慮一維連續訊號(類比原始訊號) $f_a(t)$，將其以傅利葉變換轉換爲頻譜 $F_a(f)$(圖 1.3(a)與 1.3(b))[2]。

$f_a(t)$以時間間隔$T[s]$取樣後的訊號稱爲**取樣值訊號** $f_s(nT)$(如圖 1.3 (c))[3]，在此n爲整數變數。詳細的證明參照訊號處理的教科書，在此僅敘述結論：

> 以時間間隔$T[s]$對類比原始訊號取樣之後，會得到取樣訊號值，其頻譜爲類比原始訊號的頻譜(圖 1.3(b))以$1/T$[Hz]**為間隔反覆疊加**(圖 1.3 (b))。

此爲著名的**取樣定理**，其性質推導如後。

註
1. 對於一維訊號之取樣全然沒有接觸過的讀者，可以參考諸如參考文獻[1]、[2]。
2. 頻譜$F_a(f)$一般而言為複數，圖 1.3(b)與(d)表示者為其絕對值。
3. 在此處，各取樣值 $f_s(nT)$在振幅方向上仍未取量子化，因此仍取實數值，為了與後述的「數位訊號」作區別而使用此用詞。

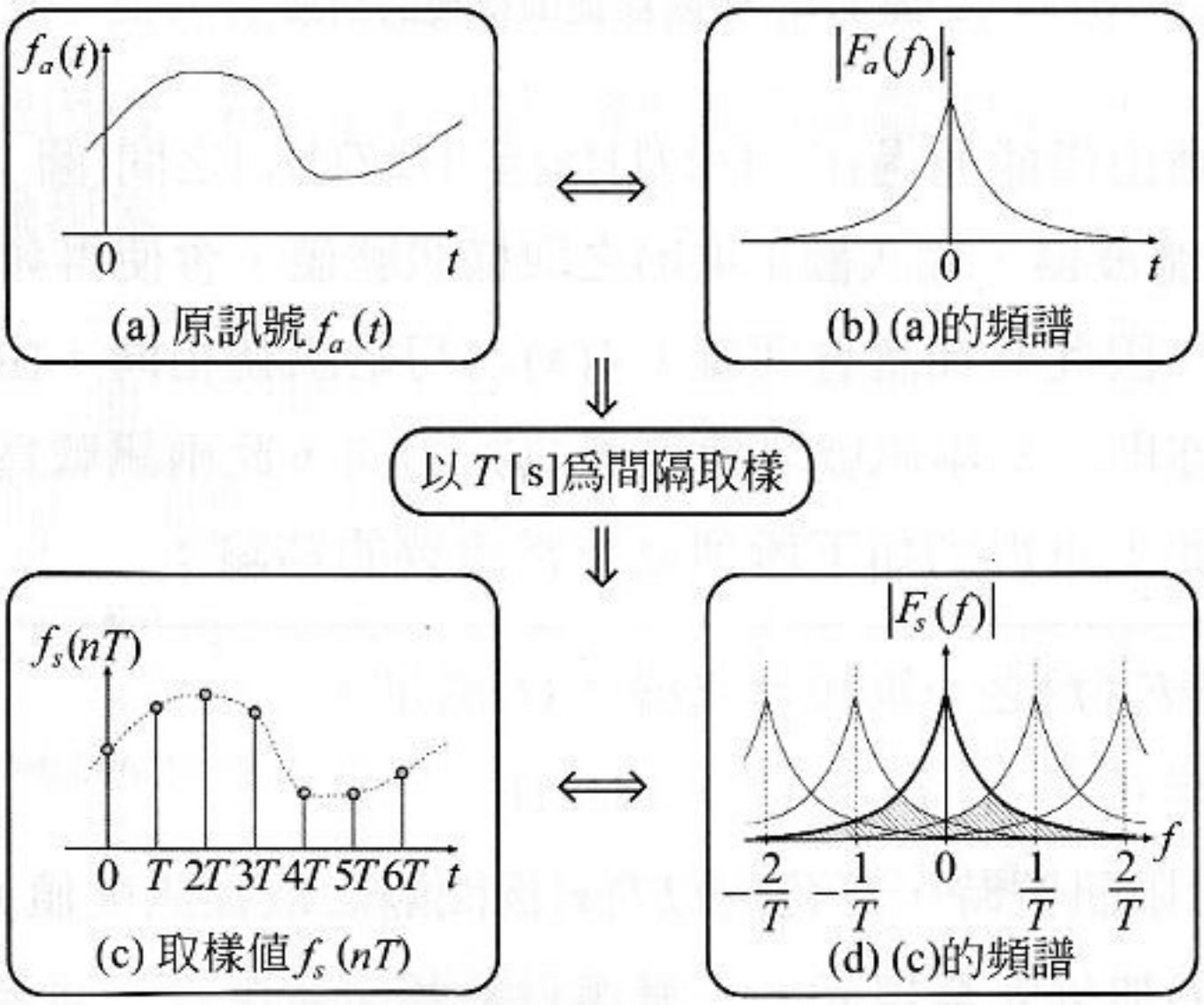


圖 1.3　類比訊號的取樣，取樣造成頻譜的變化

在圖 1.3(d)當中粗線所表示者爲原始訊號的頻譜以 $1/T$[Hz]反覆的結果，斜線所示部份爲頻譜重疊的部份。接著，頻譜$F_a(f)$僅在$-1/2T$[Hz]到 $1/2T$[Hz]的範圍中具有非零的值，考慮類比原始訊號在其外側之值爲零(圖 1.4(a))。具有此種性質的訊號稱爲帶限訊號。將圖 1.4(a)對$T[s]$取樣之後，會得到如圖 1.4(b)中的頻譜，此種情形與圖 1.3(d)不同，頻譜完全不會重疊。亦即，取樣後的頻譜不會受到兩側訊號之干涉，可完整保存頻譜之形狀。

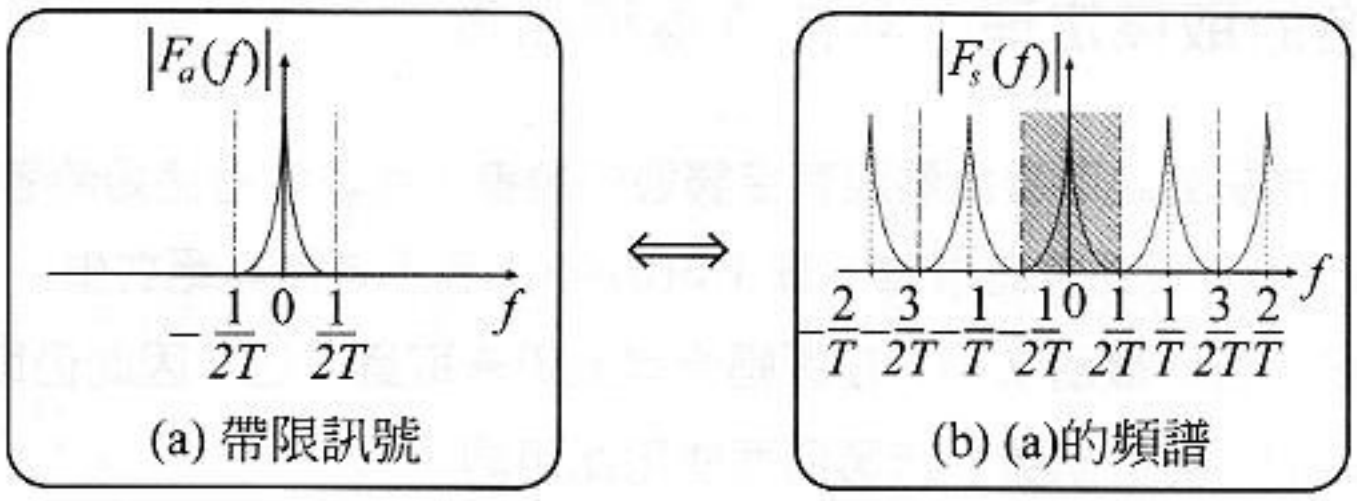


圖 1.4　帶限訊號取樣後的頻譜

因此，藉由僅能通過在$-1/2T$[Hz]至 $1/2T$[Hz]之間(圖 1.4(b)當中的斜線區域)的濾波器，輸入圖 1.4(b)之取樣訊號值，會使得頻率$|f|>1/T$之成分爲零，因此其頻譜會與圖 1.4(a)之初始訊號相同。由傅利葉變換之一致性，亦即「當兩訊號之傅利葉訊號相同，此兩訊號爲同一訊號」之性質。在此，可得到如下所列之非常重要的結論：

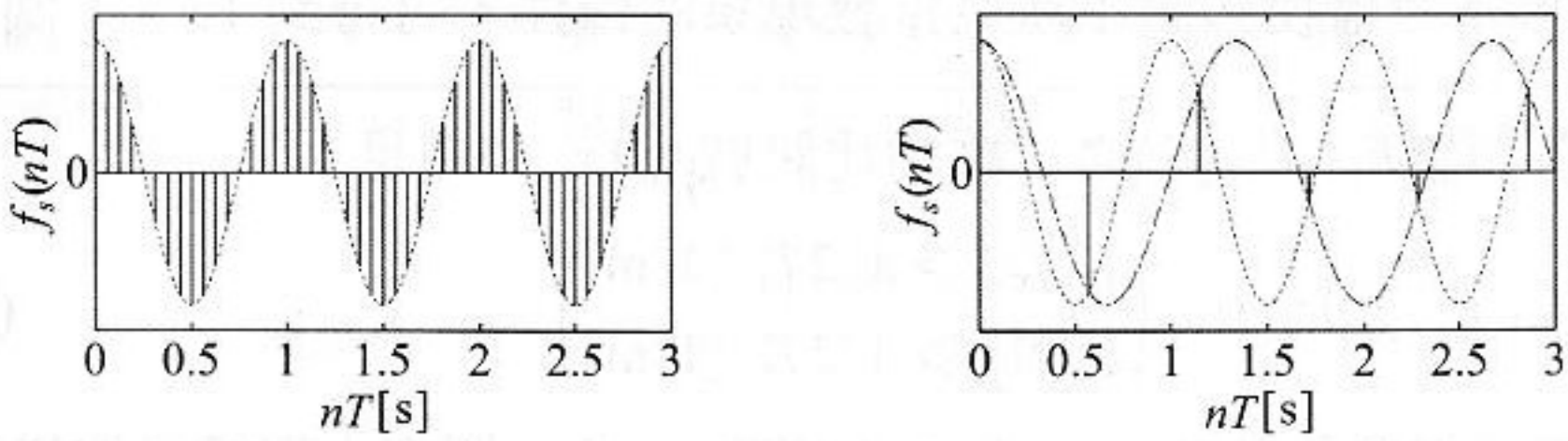
當具有頻譜$F_a(f)$之一維類比訊號 $f_a(t)$滿足：

$$F_a(f)=0 \qquad (|f|>1/2T \ \ [\mathrm{Hz}]) \tag{1.1}$$

條件而爲帶限訊號時，可藉由以$T[s]$爲間隔之取樣訊號值 $f_s(nT)$，將原訊號 $f_a(t)$加以完全復元。

藉此性質可知，當有足夠之頻率限制，可藉由僅記錄所給予訊號在取樣點上之訊號值而解決在第 1-2 頁中的第①項問題。也就是說，在電腦上不易處理的類比訊號，可以用至少是較容易處理的訊號取樣值加以進行**等價轉換**。

在此，以正弦波爲例說明取樣間隔$T[s]$的影響。圖 1.5(a)爲頻率是 1Hz 的正弦波，與$T=0.0625$s 時的取樣訊號值。頻率爲 1Hz 之正弦波爲僅包含$|f|=1$Hz 成份之帶限訊號，可藉由式(1.1)以間隔T小於 0.5s 的取樣訊號值加以復元。如圖 1.5(a)以夠小的間隔進行取樣，可以復元原正弦波。如圖 1.5(a)以夠小的間隔加以取樣的話，看來可以復元原來的正弦波，實際上也是可以。另外，(b)爲以$T=0.571$s 加以取樣的結果，在此情形下，其取樣值確實是一頻率爲 1Hz 之正弦波(虛線)取樣後的結果，但同時亦是一頻率爲 0.75Hz(點鏈線)之取樣結果。亦即，(b)的取樣訊號值，無法區別爲頻率爲 1Hz 之正弦波或是頻率爲 0.75Hz 之正弦波。此處因爲不符合式(1.1)中情況而造成錯亂的情況，一般稱之爲混疊或是重疊現象。



(a) 以T=0.00625s 進行取樣的情形　　(b) 以T=0.571s 進行取樣的情形

圖 1.5　頻率為 1Hz 之一維正弦波的取樣

因此回到原本對影像訊號進行取樣－空間取樣－的目的。在此情況的對象爲，在水平(x)與垂直(y)方向上延伸之類比(二維)影像訊號。類比影像訊號在座標(x , y)上的亮度表示爲 $f_a(x, y)$。

最普遍的空間取樣，取樣點(○點)以正交狀態配置(圖 1.6)，T_x、T_y分別表示在x、y方上的取樣間隔。另外，在影像的情形中，如圖 1.6 所示，有著y軸方向向下之慣例。

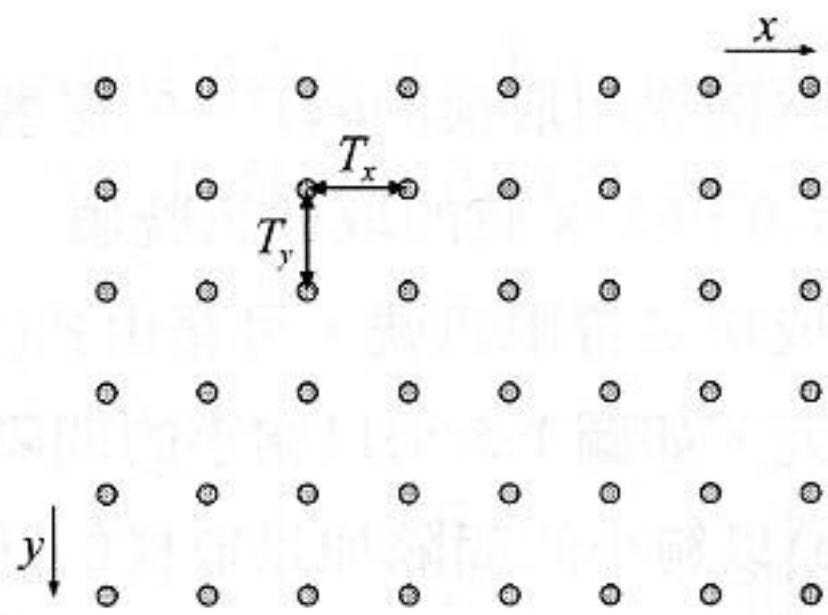


圖 1.6　空間取樣圖形

省略證明，在這種圖案進行取樣後，可以得到由一維的情形加以類推的結果。也就是說，初始類比訊號 $f_a(x,y)$的二維頻譜$F_a(f_x,f_y)$，$f_a(x,y)$與圖 1.6 取樣後之取樣訊號值的頻譜$F_s(f_x,f_y)$，原頻譜$F_a(f_x,f_y)$爲在f_x方向上以$1/T_x$，f_y方向上以$1/T_y$爲間隔的反覆排列(圖 1.7)。因此，在此情形下考慮使頻譜不產生重疊現象的條件，可得到：

當具有頻譜$F_a(f_x,f_y)$之二維類比訊號 $f_a(x,y)$滿足：

$$F_a(f_x,f_y)=0 \quad \left(\begin{cases} |f_x| > 1/2T_x \ [1/\mathrm{m}] \\ |f_y| > 1/2T_y \ [1/\mathrm{m}] \end{cases}\right) \tag{1.2}$$

條件而爲帶限訊號時，可藉由以如圖 1.6 所示圖案上取樣之取樣訊號值$F_a(f_x,f_y)$，完全復元原訊號 $f_a(x,y)$。

的結論。因此，可用有限的取樣點記錄我們眼中所見之影像。

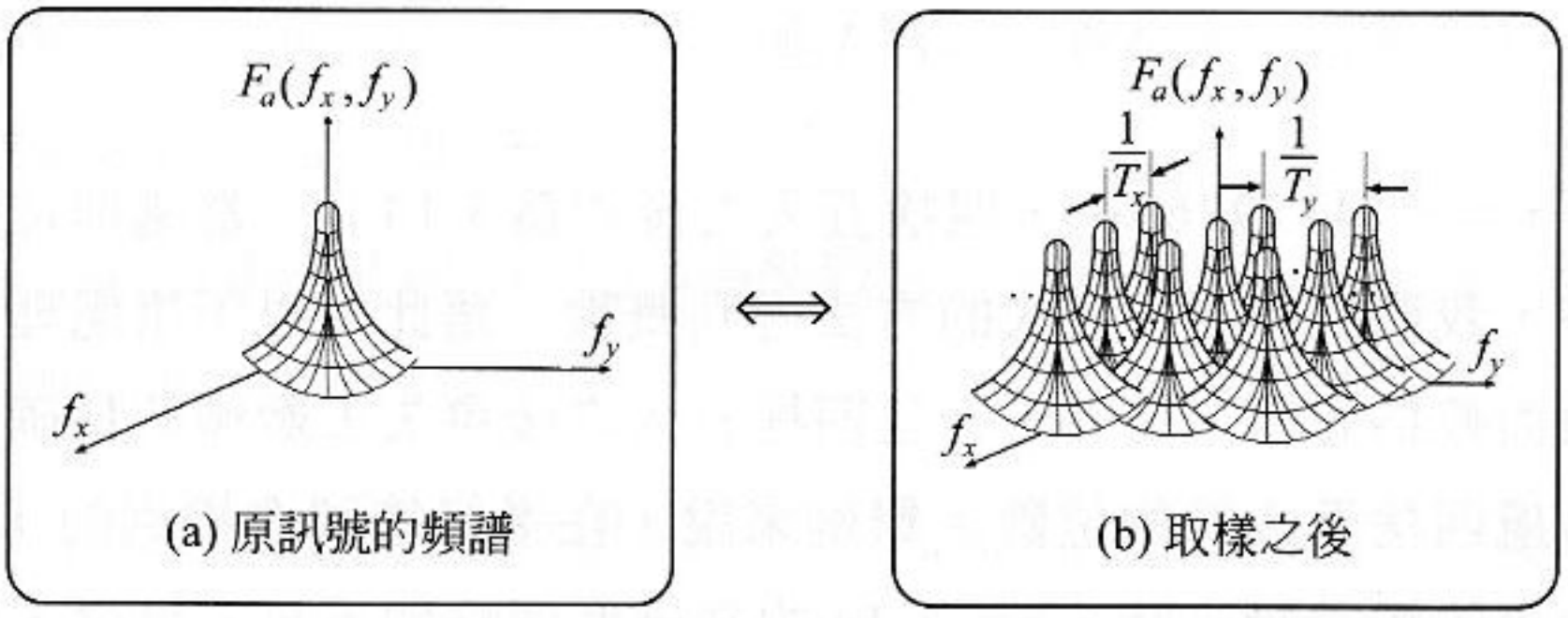


圖 1.7　二維訊號之空間取樣後頻譜變化

常見的數位相機中，係利用CCD(電荷耦合元件)攝影元件進行空間取樣。該元件在圖 1.6 的各點上設有攝影二極體，將入射光量轉變爲電荷量，再利用設有電荷輸送元件之CCD依序(以電壓值)加以讀取。實際上，數位攝影機之組成如圖 1.8 所示，在光線通過透鏡之成像平面上設有 CCD 攝影元件。但是，爲了防止取樣時的交疊現象，需要有滿足式(1.2)的頻率限制。實際上，照相機在透鏡與攝影元件間係設有低通濾鏡(LPF)，以進行頻率限制[4]。

在影像處理領域中，取樣點有畫素、像素(pixel)、畫像單元(pel)等稱呼。圖 1.8 中，自 CCD 輸出各像素的亮度值，在此取一般的實數值時，會產生如第 1-2 頁中的問題②。接著進入之後的量子化操作。

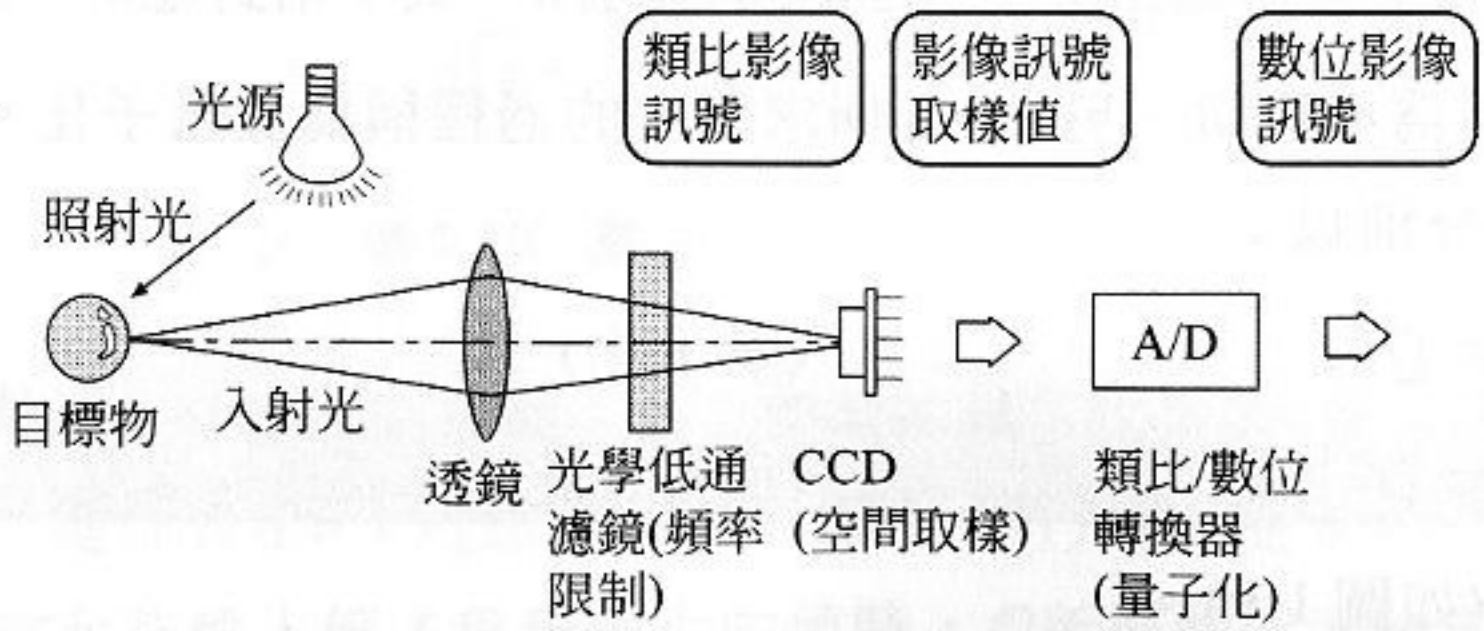


圖 1.8　數位相機的構造

量子化

「$\pi = 3.14159265\cdots$，四捨五入之後約爲 3.14」。當處理位數多的數值時，我們會用四捨五入的方法得到概數。藉此可以不用處理這麼多之位數而較爲省事。另外，一旦四捨五入之後會失去數值的精確度，因此需考慮四捨五入後的位數。舉例來說，在進行後頭會提到的 DCT 之例行程序的程式設計時，$\pi = 3.14$ 的精確度便略顯不足。反之，以地球長度約 6400 公里進行赤道長的概略計算時，π取 3.141592659 就有點無意義了。

量子化處理的目的，在於降低處理對象之數值的精密度，以得到簡潔的表現，與數值四捨五入的目的相近。在此所謂「簡潔的表現」，簡而言之就是以較短的位元長度來表示。

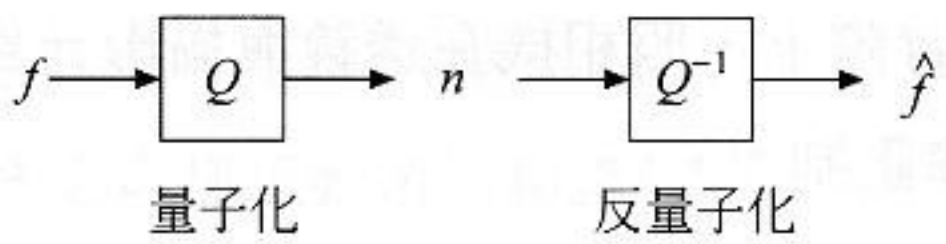


圖 1.9　量子化與反量子化

形式上的**量子化**，爲將輸入值 f 對應至預先定義的離散集合$\mathcal{O}$中的適當的元素$\hat{f}$，再輸出與$\hat{f}$ 相對應的編號n，與 f 相對應的n稱爲量子化指數，f 稱爲重現值。另外，自n求取$\hat{f}$ 的過程稱爲反量子化。量子化、反量子化分別以：

$$n = Q[f] \qquad \text{(量子化)} \tag{1.3}$$

$$\hat{f} = Q^{-1}[n] \qquad \text{(反量子化)} \tag{1.4}$$

加以表示(如圖 1.9)。

簡單的例子如：$\pi \sim 3.1$，實數值 f 爲對小數點第二位取四捨五入，所取重現值 $\hat{f}$ 的值的集合爲：

$$\mathcal{O} = Q\{\Delta \cdot N \mid \Delta = 0.1，n爲整數\} \tag{1.5}$$

量子化操作可表示爲：

$$\hat{f} = Q^{-1}[n] = [f/\Delta + 0.5] \tag{1.6}$$

反量子化操作表示爲：

$$\hat{f} = Q^{-1}[n] = \Delta n \tag{1.7}$$

其中，Δ稱爲量子化的步進幅度，此外，$\lfloor x \rfloor$爲不超過x之最大整數之函數[4]。圖 1.10 爲式(1.6)、(1.7)中，f 與n、$\hat{f}$的關係。

註 4. 舉例來說，$\lfloor 2.5 \rfloor = 2$。$\lfloor -2.5 \rfloor = -3$(不是$-2$!)，在 C 語言中，以 floor( )表示。

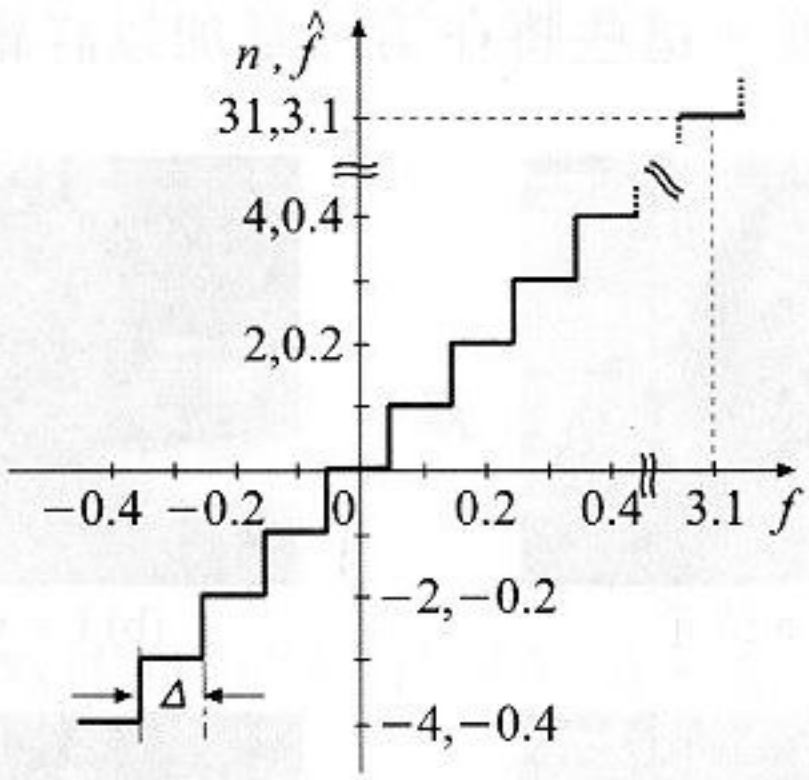


圖 1.10　量子化的例子($\Delta = 0.1$)

進行量子化時，相對於輸入值之整數的量子化指數n係一對一對應，在此，以二進位表示n，可以將連續的(實數值) f 值以近似的二進位數加以表示。當允許輸入值 f 爲無限大的值時，$\mathcal{O}$所需的元素總數也會爲無限大。其以二進位表示時需用到無限長的位元長度。在此，需對輸入值

f 實際的處理範圍(變動範圍)預先加以設限。量子化器的**量子化指數的數目**(有限集合$\mathcal{O}$的元素總數，或稱**量子化等級**)N以二進位表示爲：

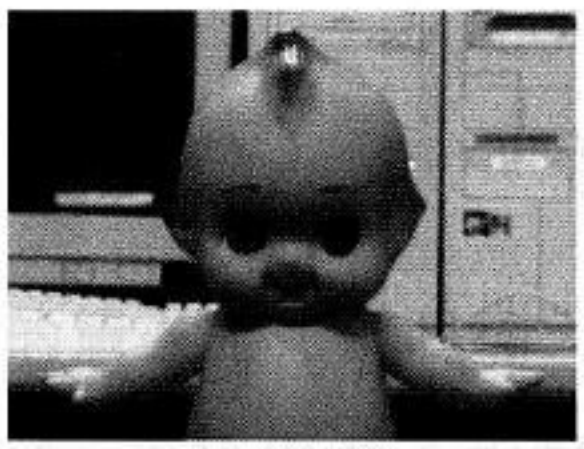
$$L = \lceil \log_2 N \rceil \quad [\text{位元(bit)}] \qquad (N\text{爲 256 時}L\text{爲 8 位元}) \tag{1.8}$$

(其中$\lceil x \rceil$爲比x大之最小整數)。此稱爲L位元的量子化範圍。

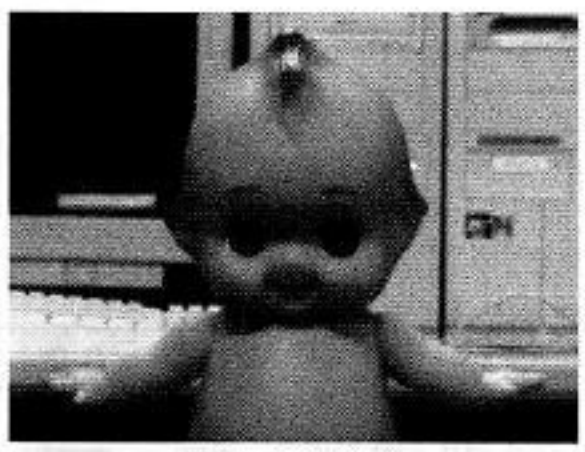
另外，進行量子化之後，輸入值f的精度會有所損失，反量子化後的重現值$\hat{f}$對於f會產生誤差，$\hat{f}$與f的誤差：

$$e = \hat{f} - f \tag{1.9}$$

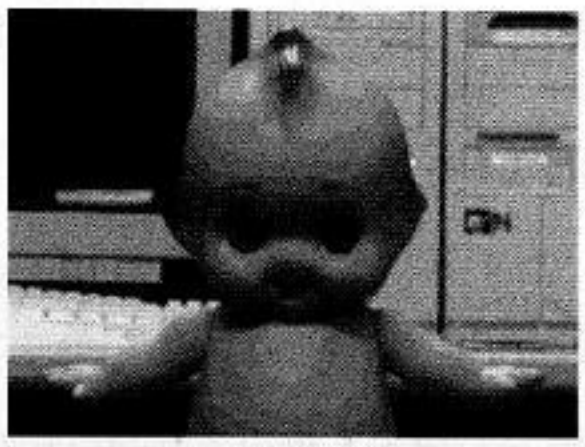
稱爲**量子化誤差**或是**量子化雜訊**。雖然縮小量子化步進幅度可以減少e值，但同時也會增加所需的位元長度。舉例來說，變動範圍保持一定時，Δ減半的話，量子化指數n需增加爲兩倍，式(1.8)中的L需增加一位元。這會使得硬體規模或是需要傳送的資訊量變大，因此需善加考慮量子化範圍的位元長度。這些將在第四章加以詳細討論。



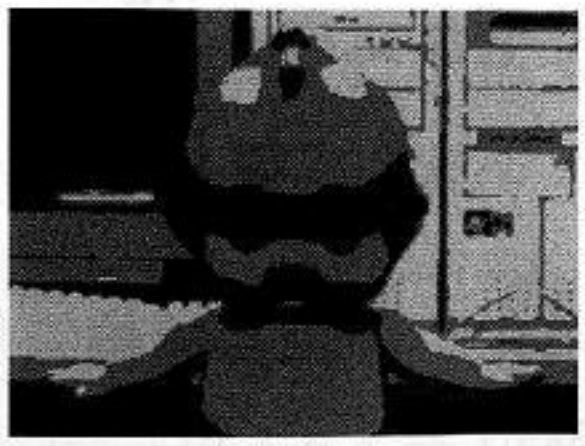
(a) L = 8 位元



(b) L = 6 位元



(c) L = 4 位元



(d) L = 2 位元

圖 1.11　L = 2、4、6、8 位元的量子化後影像實例(參照內頁)

我們的目的在於以有限位元長度表現自 CCD 取得的像素的亮度值(實數值)。這就是說，以既定的動態範圍表示最暗的與實際應用中夠亮的像素，而中間的像素值則以適當的位元長度加以量子化表示，在實際上，則藉由圖 1.8 中的**類比／數位轉換器**加以實施。

在一般影像的情形中，$L = 8$ 亦即$2^8 = 256$ 的亮度色階可說即已足夠[5]。在圖 1.11 之中，爲$L = 2 \sim 8$ 位元長後量子化的單色影像之一例。位元長度在 4 以下的時候，會出現在 8 位元、6 位元時見不到的塊狀現象，(同一圖(c))。此處，在亮度些微變化的部份所造成大幅的步進幅度變化在量子化時所產生的錯亂，稱之爲**偽輪廓**。在量子化時，需採用足夠的位元長度使偽輪廓不被察覺。

註 5. 高精密度影像中，也有採用 10 位元、12 位元的情形。

彩色化

三原色的原理與等色函數

(譯按：等色函數國內譯名未予統一，在攝影界亦稱爲配色函數、對色函數，在此一併指明。)

光通過稜鏡，因爲波長λ的折射率不同，會在屏幕上產生光譜(圖 1.12)。爲了使「光的顏色」與圖 1.12 的光譜一致，需完全地記錄其光譜成份，或是作出具有相同光譜分布的光線，並據以記錄、重現。但是，在實際上並不需要將光譜加以記錄與重現。對此視爲理所當然的讀者可以跳過本節直接進入下一部份。

我們的眼睛，並不是由光線的光譜分布直接感受到「顏色」，而是**感受到自光譜所得到，作為基準的三種積分值**。因此，即使光譜分布不

同，只要該三種(積分)值相等的話，我們會視爲同一種顏色。當然，這事實與自古以來爲人熟知的「三原色原理」，也就是說「適當地混合三種顏色，幾乎可以做出所有的顏色」有著相當深刻的關聯。我們的眼睛感受最強烈的就是紅色、綠色與藍色，基於此事實，混合的三種原色即使用紅色(R)、綠色(G)與藍色(B)。

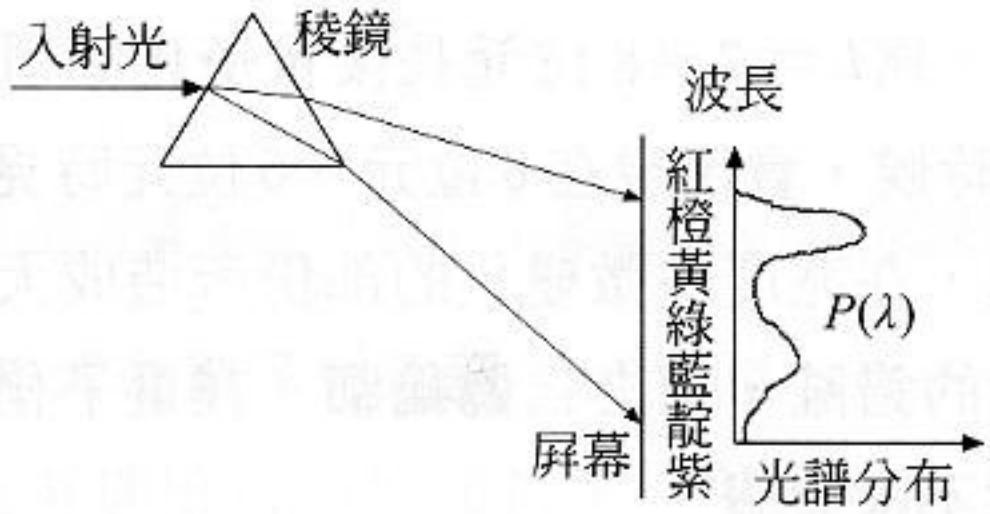


圖 1.12　以稜鏡進行分光

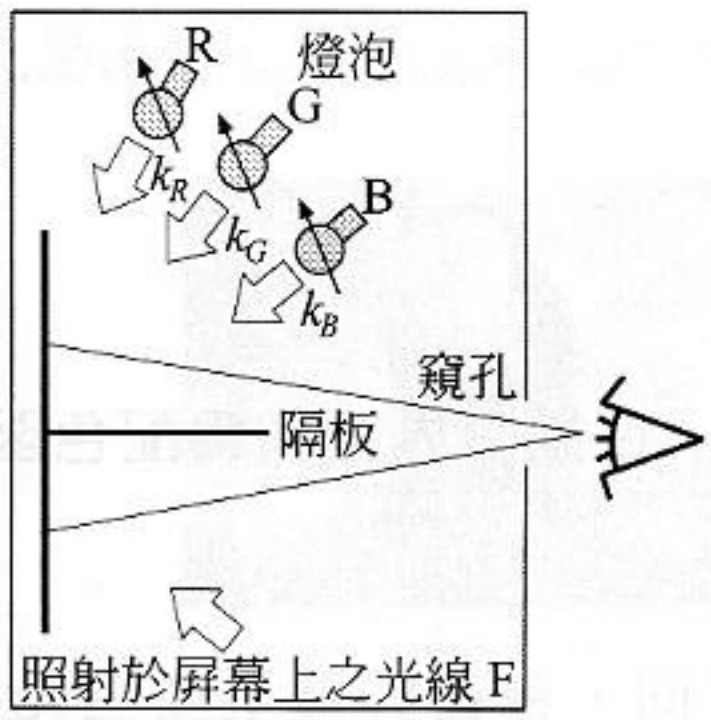


圖 1.13　三原色的等色實驗

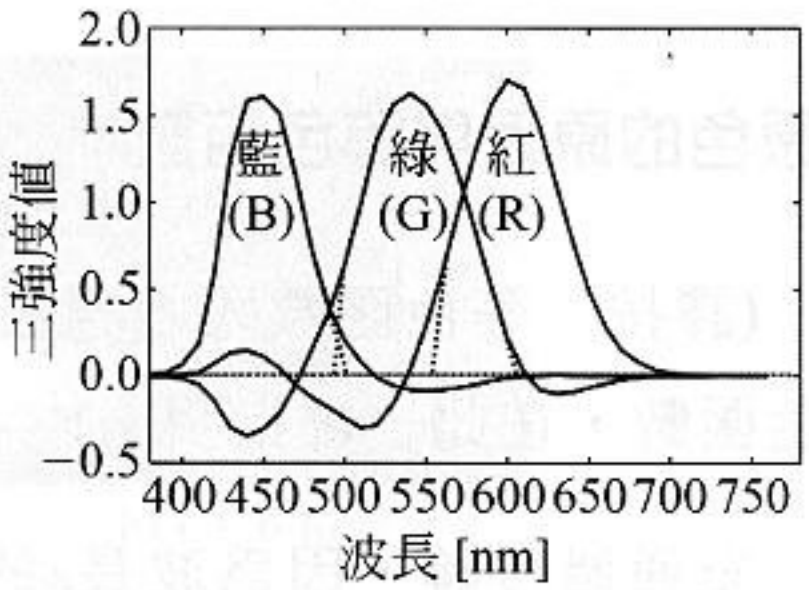


圖 1.14　等色函數之例

如圖 1.13 所示，以隔板分隔的屏幕之一側以具有顏色之光線F加以照射，另外一側以可分別調整亮度之紅、綠、藍光燈泡照射。此時，自各燈泡發射的光線混合之後會產生其他種新的顏色(稱爲加法調色)，調整各燈泡的亮度k_R、k_G、k_B之後，自窺孔觀之，可以看到兩側的顏色完

全相同。在色彩學中，該顏色F稱爲等色，亮度k_R、$\cancel{R_G}$ k_G、k_B稱爲**三原色刺激值**。

此處，對於具有單位能量之單色光進行以下實驗：三原色刺激值可爲波長λ的函數，因此可得$k_R(\lambda)$、$k_G(\lambda)$、$k_B(\lambda)$。這可稱爲等色函數，在使色彩重現時具有非常重要的地位。圖 1.14 爲在實際量測時等色函數之一例[6]。

註 6. 在圖 1.14 中具有負值之部份，負的部份表示「打開之後會變暗的燈泡」，說實話這種東西如果哆啦 A 夢(參考文獻[5])不存在的話大概也找不到實物。但是負的部份是怎麼測定出來的呢？可以參考文獻[6]，其中有相關實例。

關於顏色還有另一個重要的性質。如果具有三原色刺激值k_{R1}、k_{G1}、k_{B1}的光線F_1，與另一三原色刺激值爲k_{R2}、k_{G2}、k_{B2}的光線F_2爲等色。此時將F_1與F_2疊加，所得到三原色刺激值爲$k_{R1}+k_{R2}$、$k_{G1}+k_{G2}$、$k_{B1}+k_{B2}$的光線F_1+F_2亦爲等色。該性質恆成立，與混合顏色光線的數目無關，稱爲「重疊原理」。

因此，可由圖 1.12 的光譜得知，任意的光線F可由其光譜分布(記爲$P(\lambda)$)等同於一系列波長爲λ之單色光之疊加。因此，由「重疊原理」可知，F的三原色刺激值k_R、k_G、k_B可由構成光線F的所有單色光的三原色刺激值($P(\lambda)k_B(\lambda)$等)疊加而求得，如以下形式：

$$k_R=\int P(\lambda)k_R(\lambda)d\lambda \tag{1.10}$$

$$k_G=\int P(\lambda)k_G(\lambda)d\lambda \tag{1.11}$$

$$k_B=\int P(\lambda)k_B(\lambda)d\lambda \tag{1.12}$$

其結果使得在記錄物體色彩時，記錄自物體反射光線由式(1.10)～(1.12)所求得之三原色刺激值即可，反之，在重現該物體顏色時，由所

得到的三原色刺激值如圖 1.13 般地加以混色即可。CRT 或是液晶顯示器即是基於後者之原理；另外前者爲後頭述及之彩色攝影機的原理。

彩色攝影機的原理

吾人日常所見之電視規格中，有美日所用的NTSC標準、英德等國爲主的PAL標準以及俄法等國用的SECAM標準[7]7。這三種方式在影像訊號的傳送、處理方法與作爲基準的三原色等處皆有所不同，舉例來說在圖 1.14 中爲 NTSC 標準中的三原色與相對應的等色函數[8]。

註 7. 分別為 National Television System Committee, Phase Alternation by Line, Sêquentiel Couleurs A Mêmoire 的縮寫。

彩色攝影機，如圖 1.14 依據式(1.10)至(1.12)可以將各像素的RGB值轉換爲電子訊號。實際上，將透鏡、CCD 元件、以及 RGB 用的三種顏色的彩色濾片組合後，其設計就是將入射光線以近似於圖 1.14 的方式加以處理。但是，如果原來所選用的三原色的強度無法以負值顯示的話則無法重現圖 1.14 中負值的部份。因此，實際上在拍攝的階段就先剔除負值的部份，各特徵(原色)的邊緣處僅就所剔除的部份加以補償[8]8。

註 8. 然而這會使得顏色的重現範圍有所限制，但是在實際應用上可以說是相當足夠了。

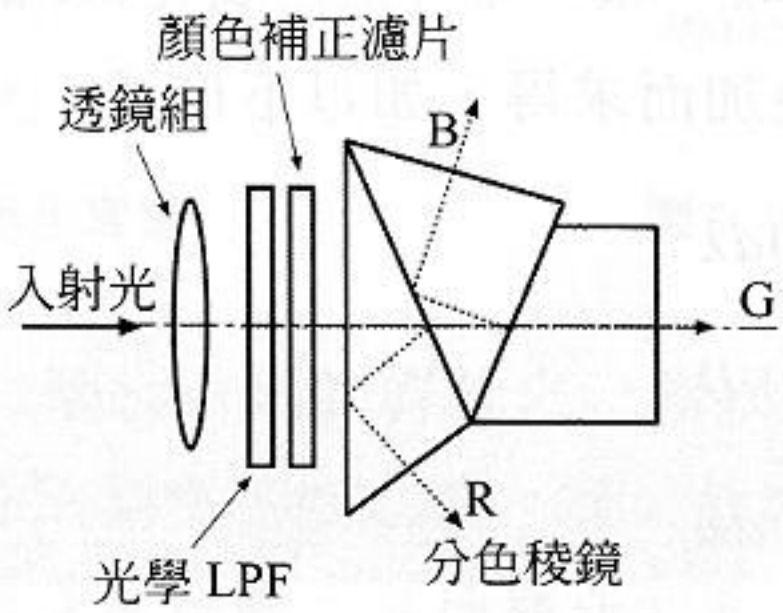


圖 1.15 三板式 CCD 彩色攝影機原理

接著介紹一種實際用於 CCD 攝影機之組成，圖 1.15 係稱爲三板式[4]的組成方式[9]。圖中的**分色稜鏡**(Dichroic Prism)之各反射面依序反射紅(R)、藍(B)光至圖面中的上下方向，同時讓綠色(G)光線通過。而各面分別設有 CCD，自透鏡至 CCD，其濾光片或是反射面之光學特性之設計使得感測特性接近於圖 1.14，而得最所欲得到之RGB訊號。藉由量子化(類比／數位轉換)，可得到最後的數位影像訊號，但是，對於重現顏色而言，RGB值是完全相等的，因此 RGB 各值需以完全相同的條件加以量子化。通常是各別以長度爲八位元加以量子化，一個像素以 24 位元加以表示。此時可表現的色數爲$2^8 \times 2^8 \times 2^8 = 1670$ 萬色。

註 9. 其他亦有以一枚 CCD 與濾光片陣列所構成稱為單版式之組成方式[4]。

在本書中，將數位攝影機基於以上原理的所拍下的影像稱之爲**數位影像**。

修正-γ修正

現有用於彩色影像之攝影與顯示之最常見組合爲CCD與CRT(陰極射線管)，但是將兩者以圖 1.16 中的虛線加以連接之後，也無法得到如圖 1.17 中所期待的影像。這是因爲如圖 1.16 右上方所示，CRT 的輸入輸出關係，亦即輸入電壓與映像管面亮度的關係具有「向下彎曲」的非直線特徵。

CCD攝影機的入射光量I與輸出電壓Y間的關係極爲接近直線，兩者的關係以k_{CCD}爲比例常數，可寫爲：

$$Y = k_{CCD} I \tag{1.13}$$

另外，CRT的輸入電壓U與輸出亮度L兩者間的關係相當近似於下列式子：

$$L = k_{\mathrm{CRT}} U^{\gamma_{\mathrm{CRT}}} \tag{1.14}$$

其中k_{CRT}爲常數。在此γ_{CRT}爲用來表示非直線特徵的常數，一般未經調整的CRT其γ_{CRT}值幾乎無例外的都可取在2.5左右。該特性以式(1.14)中的γ爲名稱爲**γ特性**。

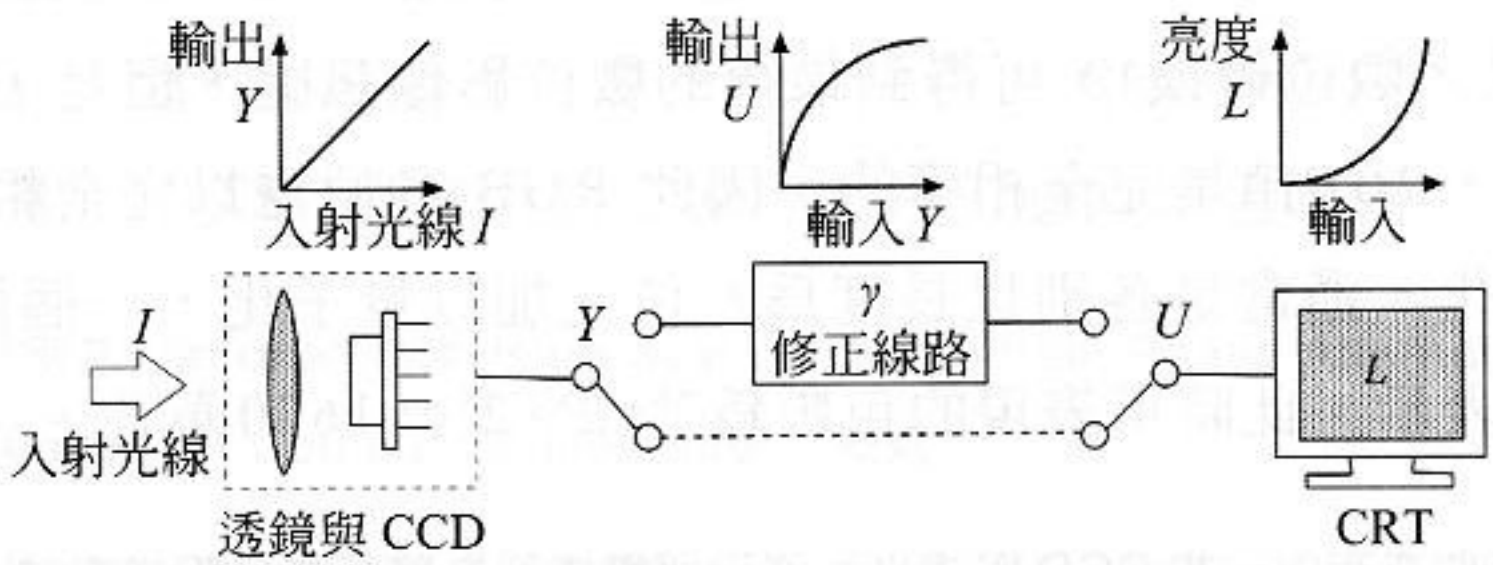


圖 1.16　CCD 攝影裝置與 CRT 顯示裝置的連接

(a) 理想情況

(b) 直接連接的情況

圖 1.17　由 CCD 攝影與 CRT 顯示連接顯示影像之一例

爲了消除γ特性的影響，可如圖 1.16 所示，在攝影機與CRT中間插入具有：

$$U = Y^{\gamma_c} \tag{1.15}$$

爲特徵之γ修正線路。此γ_c的最佳設定應該讓$\gamma_{\mathrm{Total}} = \gamma_c \gamma_{\mathrm{CRT}} = 1$，但是對於人類而言，最佳的$\gamma_c$值與 CRT 周圍的亮度有關，並非如此簡單。

由於 CRT 之γ特性的影響，在早先播送電視訊號時即造成問題。因爲在播送的時候是屬於廣播的一對多形式，在電視臺這邊播送的是進行過γ修正的訊號，以使得收視戶端較爲省事。因爲有此歷史背景，因此在攝影端進行γ修正成爲慣例，舉例來說，遵循 NTSC 標準的攝影機，爲了使自CCD產生之輸出訊號$\gamma_{\text{Total}} = 1.125$，即規定以$\gamma_c = 2$進行修正。另外，$\gamma$修正係對於$R$、$G$、$B$各訊號分別進行，因此依慣例進行$\gamma$修正後的各訊號分別以$R'$、$G'$、$B'$加以表示。

另外一方面，現今CRT也用在電腦的顯示裝置，因此也需了解關電腦上的γ修正。舉例來說，IBMPC 一般不具有進行γ修正的硬體，而Windows或是Unix等作業系統也不會做γ修正，直接以未經調整的CRT特徵值加以輸出。因此，在顯示影像的時候，必須對在檔案中記載之各點的亮度值以0.45的值進行γ修正[10]。此外，Macintosh 中具有用於γ修正的檢索表，其γ_{Total}值預設爲1.8(可加以調整)。因此在電腦界由於所對應機種不同，其γ特徵亦有所不同，這是必須注意的[10]。

註　10. 因此，自攝影機輸出至 IBMPC 的時候(通常是在攝影機輸出時進行γ修正)，直接將類比／數位轉換後的像素值記錄於檔案內即可。

各種數位影像的格式

彩色影像以像素爲單位紀錄其 RGB 值，因此可加以紀錄、保存。在配合壓縮編碼的實際應用上，幾乎不會對各像素的RGB值直接編碼。其理由爲，爲了考慮人類的視覺特徵，可以有較 RGB 更爲有效的影像表現方式。

人類的視覺特性與顏色表現

當我們看到顏色的時候，不只是看到「顏色」，同時也感受到其「明亮程度(亮度)」。舉例來脫，通常的映像管上當R(紅)、G(綠)、B(藍)各原色分別以同一種程度發光的時候，我們的眼睛會認爲明亮度的順序爲G、R、B。這是因爲我們的眼睛對各原色的感度有所差異。

可以將NTSC的三原色寫成以下式子

$$Y = 0.299R + 0.587G + 0.114B \tag{1.16}$$

，只要$[R, G, B$的組合使Y的值相同，在我們的眼中會感受到相同的明亮程度(亮度)。因此式(1.16)中所記載的Y值稱爲亮度值。也就是說，可以用式(1.16)將彩色影像訊號轉換爲單色訊號。

圖 1.18 表示亮度與色彩組成對人類眼睛的感度特徵。「明暗的特徵」爲對於單色濃淡圖案之感度特性，另外，「紅一綠」、「黃一藍」分別代表固定的紅色與綠色、以及黃色與藍色圖案之感度特性，橫軸相當於圖案的細密程度(空間頻率)[11][12]。在同一圖中，可以看出「在高頻區域中，亮度對感度的影響較顏色差異對感度的影響爲大」。因此，在記錄影像時，顏色不需要像亮度需儲存在高頻空間區域的資料，因此可減少使用的資訊量。

考慮三維向量$[R, G, B]^T$之RGB值，可以用一3乘3之方陣F

$$\begin{bmatrix}\alpha\\ \beta\\ \gamma\end{bmatrix} = F\begin{bmatrix}R\\ G\\ B\end{bmatrix}, \quad \begin{bmatrix}R\\ G\\ B\end{bmatrix} = F^{-1}\begin{bmatrix}\alpha\\ \beta\\ \gamma\end{bmatrix} \tag{1.17}$$

將其轉換爲另一個三維向量$[\alpha, \beta, \gamma]^T$。可用$F$的反矩陣將$[\alpha, \beta, \gamma]^T$完全回復爲原來的$[R, G, B]^T$，因此可用$[\alpha, \beta, \gamma]^T$取代$[\alpha, \beta, \gamma]^T$記錄像素值。

在此，$[\alpha,\beta,\gamma]^T$中的一個分量可爲式(1.16)所表示的亮度值Y。此時，爲了完成式(1.17)中的轉換，只要在 RGB 的三維空間中決定出與Y方向不同的兩個軸即可。該兩軸取爲B-Y軸與R-Y軸[11]，分別表示「藍」與「紅」的分量。此外，相對於滿足$R=G=B$時爲無色彩的條件，B-Y與R-Y兩者經常同時爲零[12]，因此可自然地將該兩者考慮用爲表示色彩成分。

註 11. 雖然也可使用G-Y，但是儘可能地用於離Y軸較遠者在數值計算上較為有利，因此利用在式(1.16)中係數較小的B與R[13]。

12. 在式(1.16)中代入$R=G=B$後使得$Y=R=G=B$，之後會較易推導。

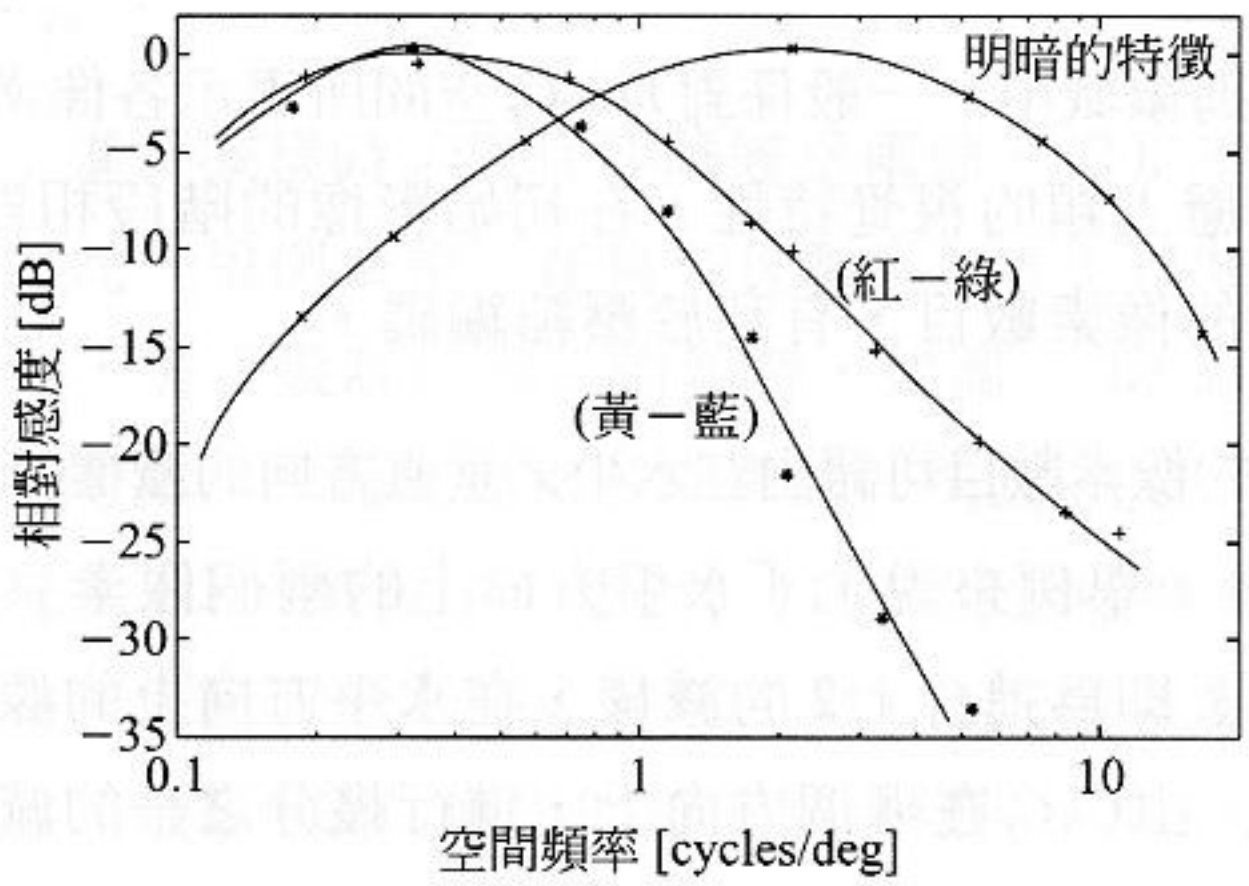


圖 1.18　亮度組成與顏色組成對人類眼睛的感度特徵

先假設將RGB值已經正規化爲在 0～1 之範圍內，可知自式(1.17)所得到的Y、B-Y與R-Y分別落在$0 \le Y \le 1$，$-0.886 \le B\text{-}Y \le 0.886$，$-0.701 \le R\text{-}Y \le 0.701$ 的範圍內，B-Y與R-Y的區間寬度依比例縮放爲-0.5～$+0.5$，分別稱之爲C_b、C_r之色差分量。將式(1.17)中的$[\alpha,\beta,\gamma]^T$取$Y, C_b, C_r]^T$，與其對應的轉換可得到：

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.334 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1.18}$$

[13]該三維空間稱爲YC_bC_r彩色空間。

註 13. 但是，在使用進行γ修正後的$R'G'B'$值的情形中，式(1.18)的$[R, G, B]^T$必需置換為$[R', G', B']^T$。

4:4:4、4:2:2、4:2:0 格式

在影像編碼領域中，一般係對YC_bC_r空間所表示各像素值進行編碼。其理由爲，考慮人類的視覺特性，在初始影像的階段相對於亮度Y可以減少色差C_bC_r的像素數目，有利於壓縮編碼。

色差C_bC_r的像素數目可藉由在水平／垂直方向的**減樣**(downsampling：縮減)加以減少。舉例來說，「水平方向上的兩個像素只取一個像素而捨去另外一個」即爲進行 1/2 的減樣，在水平方向上的像素數目即減少爲原來的 1/2。由C_bC_r在哪個方向上，進行幾分之一的減樣而有各種影像格式的存在，主要有(圖 1.19)：

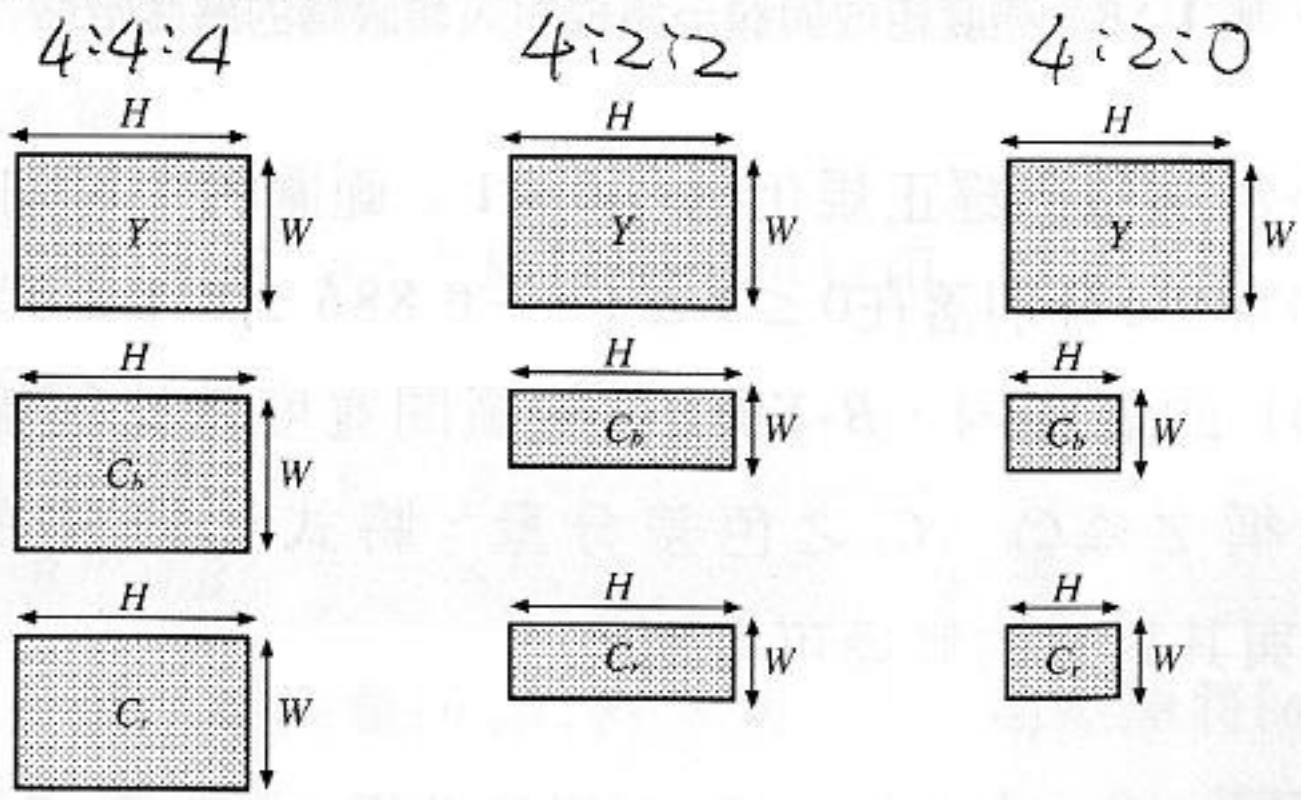


圖 1.19 各種YC_bC_r的格式

(a) 不進行減樣。

(b) 在水平方向進行 1/2 的減樣。

(c) 在水平與垂直方向均進行 1/2 減樣。

這些格式通常以$Y=4$，Y、C_b、C_r間的水平方向的像素數目比例表作爲表示的慣例，圖 1.19 中(a)(b)(c)各圖分別稱爲 4:4:4、4:2:2、4:2:0 格式[14]。

註 14. 但是在(c)的情形當中，由於在垂直方向上進行減樣，在垂直方向的奇數線條上僅對C_b進行 4:2:0，而在垂直方向的奇數線條上僅對C_r進行 4:0:2 以前者的 4:2:0 作為代表。

在C_b與C_r上進行減樣時，取樣間隔變爲兩倍，但是不希望爲此而造成交疊現象。因此，舉例來說，在具有係數爲[1 3 3 1]/8 的濾波器的方向上進行減樣時，需要限制C_b與C_r的頻帶，如圖 1.20(a)所示，這些具有偶數個係數的濾波器的輸出點爲各係數的對稱點(在此例中爲 3 與 3 的中間)[15]，因此在 4:2:2 的情況中，相對於YC_bC_r的初始像素(以○表示)，濾波後的取樣點╳係配置於其中心。在 4:2:0 格式中需要進一步地在垂直方向進行濾波，最後YC_bC_r像素的位置關係如同一圖中(b)所示[16]。

註 15. 在數位訊號的表現上，為「濾波器的群延遲＝(定位長 1)/2」。

16. 但是，在 MPEG-2 的規格中，C_bC_r的取樣位置與圖 1.20(b)不同，詳情參照文獻[62]。

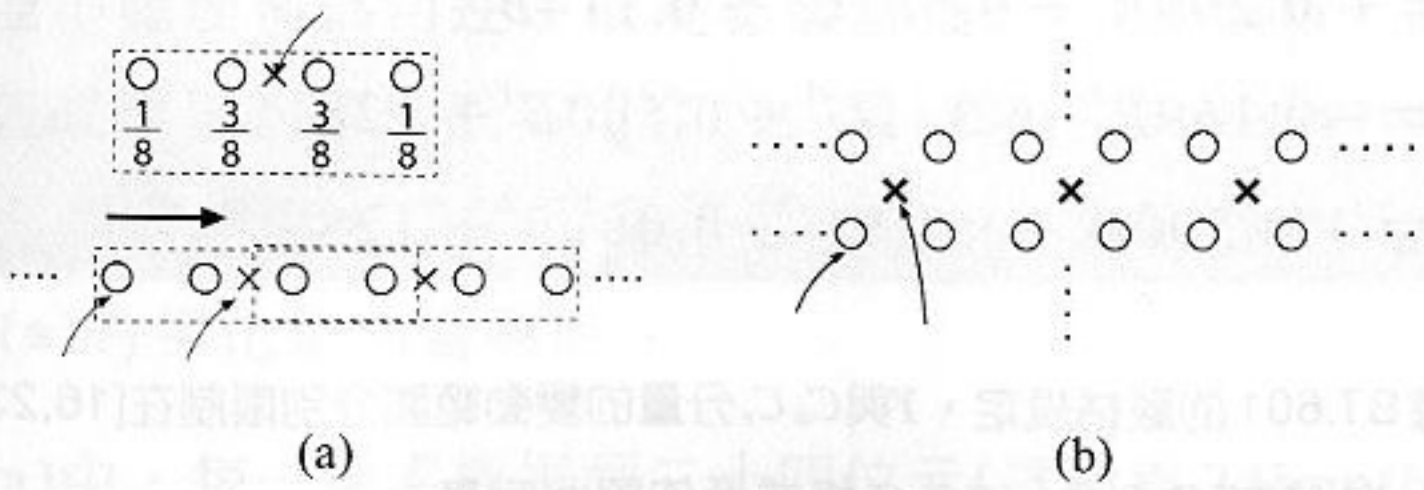


圖 1.20　C_bC_r之濾波與減樣

數位影像的各種格式

影像編碼／解碼器除了要能對應不同尺寸影像的基本功能外，爲保有各種影像機器間的互換性，常用如表 1.1 所示的各種尺寸的影像。此處的BT.601 爲稱作ITU-R BT.601[14]，制定爲電視廣播用的影像格式之國際標準。

表 1.1　各種影像格式的尺寸

格式名稱	尺寸	顏色表現	備考
BT.601	720×480	4:4:4，4:2:2	NTSC 用
	720×576	4:4:4，4:2:2	PAL 用
SIF-525	352×240	4:2:2，4:2:0	NTSC 用
SIF-625	352×288	4:2:2，4:2:0	PAL 用
CIF	352×288	4:2:0	共通格式

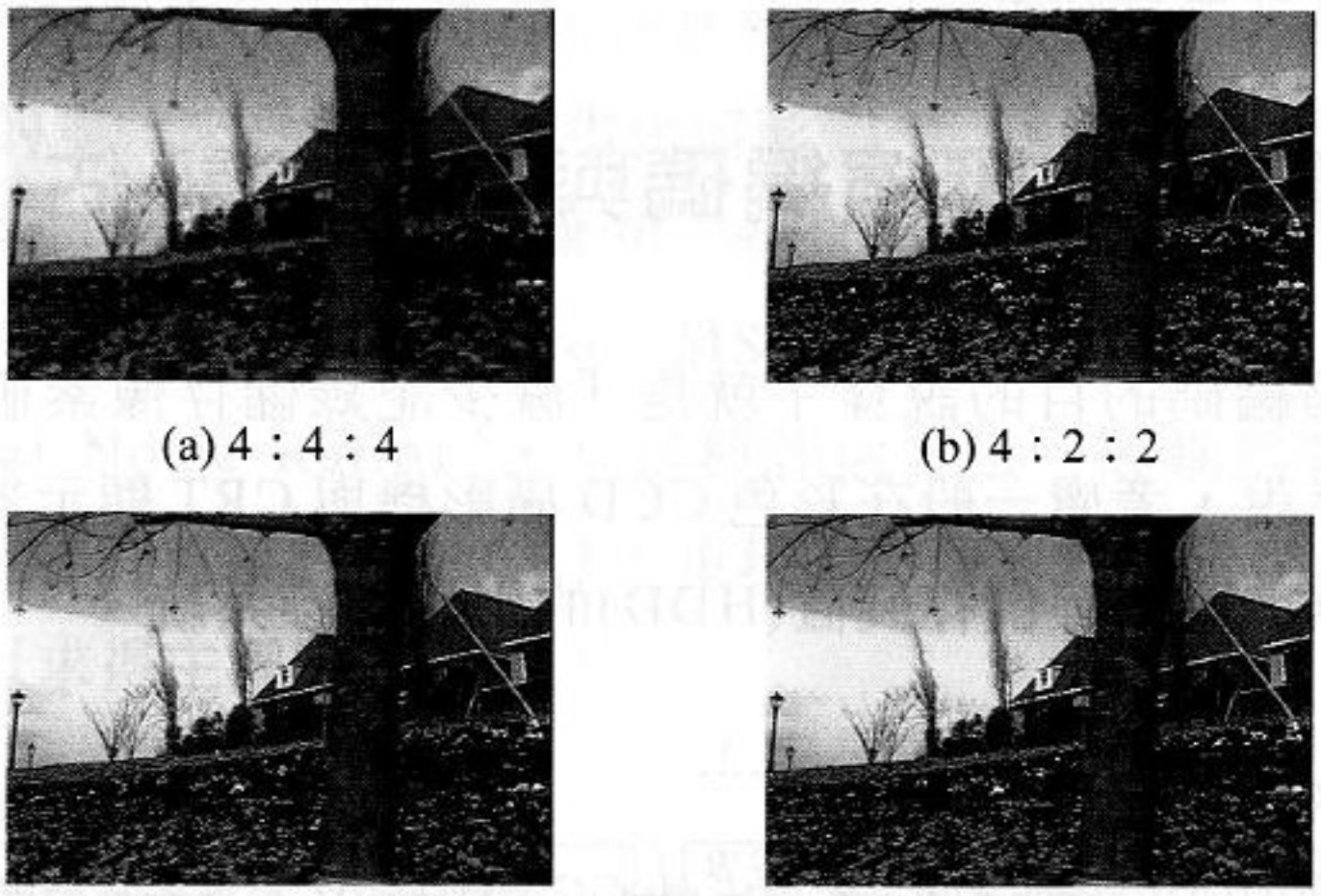
也有 CIF 與 SIF 的邊長小爲二分之一，稱爲 QCIF、QSIF 的格式。

這些格式的共通點爲，YC_bC_r各成分以 8 位元長 256 階加以表現。在γ修正之後攝影機輸出的$R'G'B'$訊號分別以八位元加以表現的情況中，自$R'G'B'$轉換爲YC_bC_r的轉換可以根據式(1.18)而寫成[17]：

$$\begin{cases} Y = +0.299R' + 0.587G' + 0.114B' \\ C_b = -0.169R' - 0.331G' + 0.500B' + 128 \\ C_b = +0.500R' - 0.419G' - 0.081B' + 128 \end{cases} \tag{1.19}$$

註　17. 根據BT.601 的嚴格規定，Y與C_bC_r分量的變動範圍分別限制在[16,235]、[16,240][14]，在 MPEG-2 中，YC_bC_r格式係依照此定義。

在此需注意，爲了使C_bC_r成分落在 0～255 的範圍內，因此加上了 128 的偏移。反過來說自YC_bC_r求取$R'G'B'$時，可以利用式(1.19)解$R'G'B'$。



(a) 4：4：4　(b) 4：2：2

(c) 4：2：0　(d) 僅 Y 方向在水平垂直各 1/2

圖 1.21　YC_bC_r各分量減樣的影響(參照扉頁)

最後可以來看對C_bC_r進行減樣的影響，圖 1.21 爲原影像的$R'G'B'$訊號分別對YC_bC_r作(a)4:4:4、(b)4:2:2、(c)4:2:0 的轉換後，再轉換回$R'G'B'$訊號的結果。(a)4:4:4 因爲不對C_bC_r作減樣，因此與原影像完全一致，但是(b)4:2:2 與(c)4:2:0 因爲減樣而導致資料的流失，因此無法復原爲與原影像完全相同。在影像處理的領域中，如(a)一般可得到相同結果的一連串處理稱爲**可逆**的或是**非破壞性**的(lossless)，如(b)、(c)這樣多少會造成誤差的則稱爲**不可逆**的或是**破壞性**的(lossy)。但是，這種情況下的影像破壞對人類的視覺特徵來說是相當輕微的，要看出(b)與(c)相較於(a)的劣化是相當難的。

而在(a)中，每一像素要用到二十四位元(這稱爲 24bpp(bit per pixel)或是 pel)，但是(b)是 16bpp，(c)更少，爲 12bpp，比(a)少了 50%的資

料量。同時，在 Y的水平垂直方向而得到同一圖(d)18bpp的結果，其畫質明顯地較(b)、(c)來得差，因此可以了解到僅對色差資訊進行刪減的處理手法有其正當性。

關於影像的壓縮編碼與畫質的評估

影像壓縮編碼的目的說穿了就是「減少記錄儲存像素值的檔案大小」。舉例來說，考慮一般在彩色 CCD 攝影機與 CRT 顯示器中間接上編碼／解碼器與硬碟等儲存裝置(HDD)的影像儲存系統。

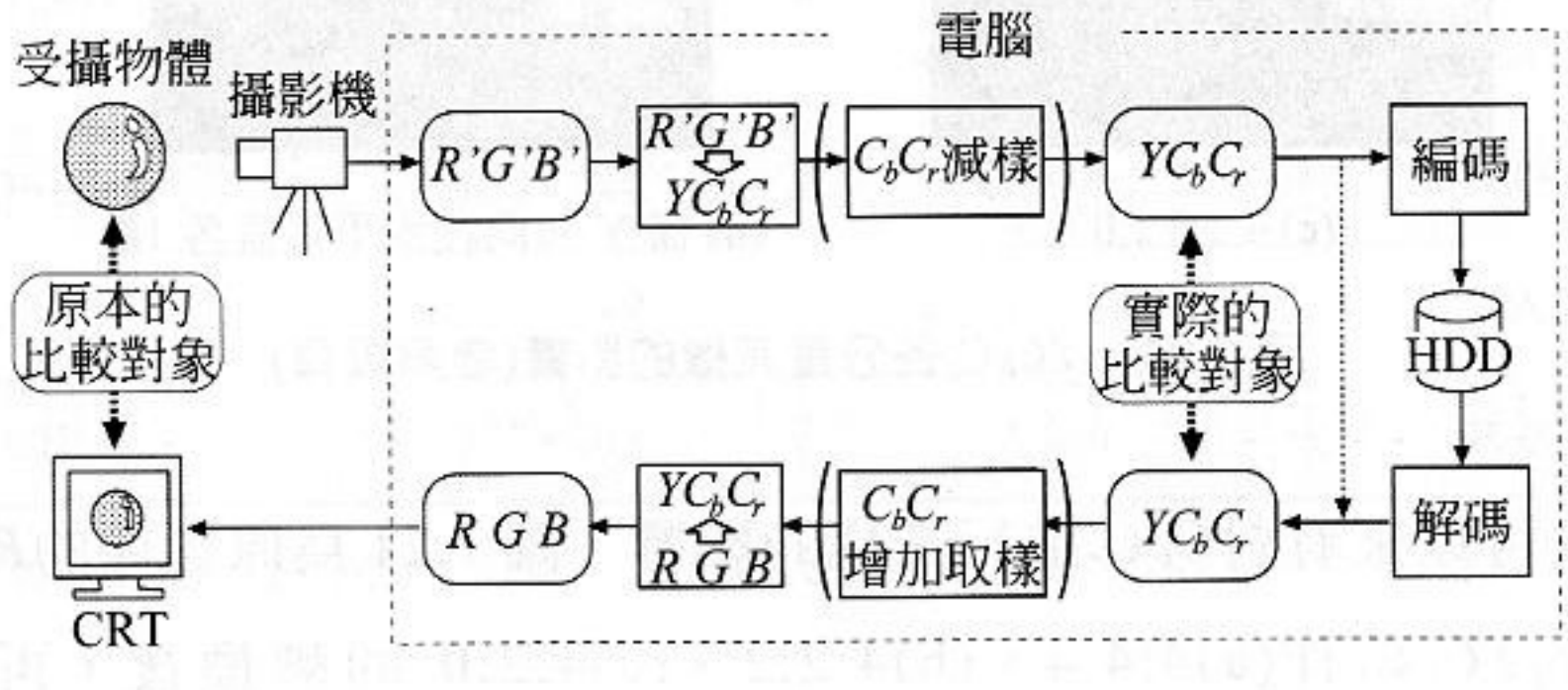


圖 1.22　一般的影像儲存系統

這種系統的目的，是儘可能地以最小的檔案大小表現受攝物體，同時將其儘可能地於CRT上忠實地重現，本來應該是直接比較受攝物體與在CRT上重現的影像以進行編碼／解碼器的性能評估。但是，在兩者之間有著三維物體與二維影像的差異，進一步地摻雜前述的γ修正的誤差或是在C_bC_r減樣造成的影響後，會使得僅就編碼／解碼器造成的畫質劣化評估難以進行。

因此，在圖 1.22 的編碼器至解碼器的連接爲直接連接的情況下，假設攝影機與 CRT 等皆調整至可以相當忠實地表現影像。在此假設條件下，受攝物體與重現影像間的直接比較可以被取代，而用編碼／解碼後的 訊號間的誤差來評估畫質的劣化情形[18]。因爲這種方法最單純，所以在影像編碼的領域中被廣泛的使用。在實際比較上，常用像素值之間的平均平方差(Mean Square Error, MSE)，或是其對數值之訊號雜訊比 SNR(Signal Noise Ratio)[19]，該這種評估方法一般稱爲**客觀評估**。另外，實際上亦有直接評估於CRT上重現的影像，對多數受測者提示爲同一個影像以求得**主觀評估**結果。

註　18. 常常僅比較x訊號間的誤差。

19. 如後頭所述及的，用訊號等級除最大變動範圍所得的 PSNR(Peak SNR)。

實際上，一直以來都有使用 MSE 或是 SNR 的客觀評估無法涵括影像整體畫質的負面意見[15]。此外，在進行主觀評價時的標準也正在訂定[16]。但是：基於以下理由：

- 主觀評估需考慮到各種心理學的因素以及編碼／解碼器所不包括的其他因素。
- 即使使用MSE或是SNR等客觀評估，也可得到相當程度的畫質評估結果。
- 在後頭會提到，MSE 特別地適用於影像壓縮編碼之資訊理論的處理，在理論上也淺顯易懂。

在本書中關於編碼／解碼的畫質評估主要是採用藉由MSE或是SNR值進行的客觀評價。

2章

影像壓縮編碼的基礎

將現有的電視影像以數位影像加以處理時，每秒產生的資料量可計算如下，畫面大小是 720×480；4:2:0 的格式，每秒有三十格，可得到

720×480×12×30 = 124Mbit/s

byte

這相當於每秒 15.5MB，舉例來說，要寫進一片容量為 650MB 的光碟(CD)，大概四十秒左右就用完了。影像壓縮編碼最大的誘因，就是因為影像具有如此龐大的資料量。

本章在詳細地進入影像壓縮編碼方法之前，先介紹影像可以壓縮的基本概念與編碼的原理等觀念。

像素值的寫法

首先，先規定在本書中使用的影像像素值的寫法。對於靜止影像來說，如圖 1.6 取x軸與y軸，對於像素(x, y)(其中x、y爲整數)而言，其像

素值以 $f(x,y)$、$g(x,y)$等加以代表。在不特別限定的情況下，f、g指RGB或是YC_bC_r之一般分量(表示亮度Y的機會較多)。

另外，因爲動畫影像爲靜止影像以一定的格數並列，在此使用相當於第幾格的第三個變數n，以$f(x,y,n)$代表第n格上的像素值。

何謂影像壓縮編碼

所謂的**影像壓縮編碼**，就是將作爲處理對象的目標影像儘可能以最少的資料量，儘可能地用最忠實的方式來解碼[20]。現在考慮一組編碼器(encoder)與解碼器(decoder)。編碼器接收到目標影像I之後，將其轉換爲位元串b再加以輸出[21]。另外一方面，解碼器在接受位元串b之後將其加以解碼並輸出爲解碼後的影像I'(圖 2.1)。如果編碼後的結果，使得位元串b的總位元量較原影像I 的總位元量來得低就是影像的壓縮編碼。

註 20. 這就是說，「影像壓縮編碼」是最正確的說法，而經常將「壓縮」加以省略，說成是「影像編碼」。

21. 這是指所得到編碼後的成果是一串由「0」與「1」構成的二進位資料，或稱位元的串流。

圖 2.1 影像之編碼／解碼器

波形編碼

圖 2.2(a)爲名爲「花園」的影像(SIF影像)的第一格，表現方式爲 8 位元的單色影像，總位元量爲 352×240×8 = 675840 位元。另外(b)爲(a)

加以壓縮編碼[22]後，再以相對應的解碼器解碼之後的結果。編碼後的位元串的總位元量(資料量)是 129984 位元，**壓縮率**是 129984/675840，約等於 0.1923，1/5 左右。

註　22. 使用後述之 JPEG 方式。

在圖 2.2 中，原影像與解碼後影像的差異幾乎無法看得出來，但是**兩者不完全相同**。也就是說，在編碼的過程中，原來的影像有部份資料遺失了，因此**解碼之後沒有辦法完全復原原影像**。如同在第一章中提到的，這種編碼方式稱爲**不可逆編碼**(或是**破壞性編碼**)，與其相反的可逆編碼(非破壞性編碼)，在編碼／解碼之後可以完全復原原影像，但是反之也有無法達到像破壞性編碼那麼高的壓縮比率之限制。

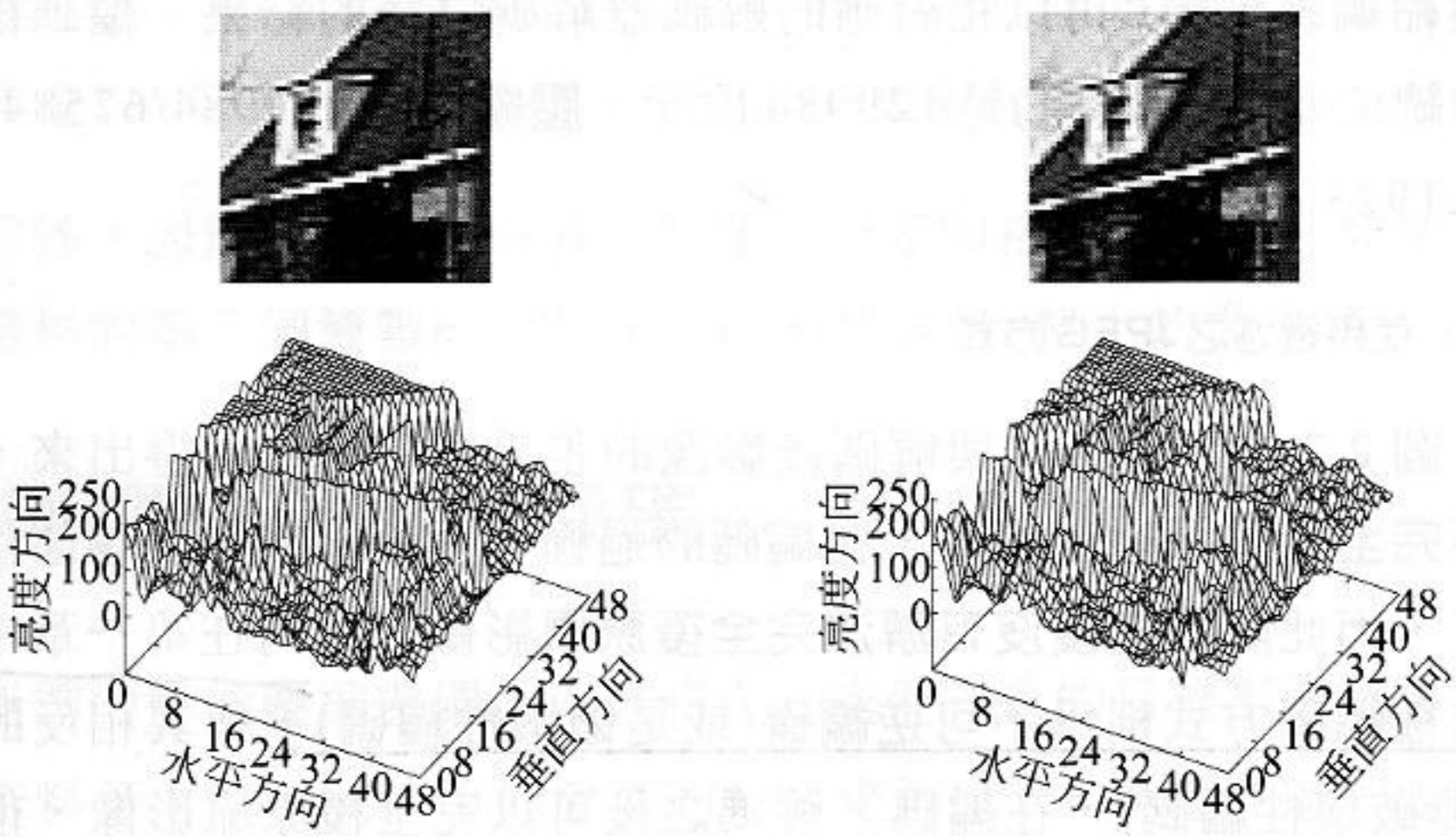
圖 2.3 爲相對於圖 2.2(a)、(b)，切出一 48x48 的區域並加以放大顯示，在 z 軸取亮度值之二維波形後的顯示結果。編碼前後的亮度值在二維波形上觀之，可以知道兩者之間極爲接近。

本書中介紹的影像編碼方法，主要屬於波形編碼之分類。波形編碼是以「目標影像是哪種圖案，就將其呈現出來」的方式，也就是說**不詳細過問其內容為何**，僅捕捉影像的二維波形，並**以最少的資料量呈現該波形**。

675840 位元
(a) 原影像 (單色 8bpp)

129984 位元
(b) 解碼影像

圖 2.2　對「花園」第一畫格進行壓縮編碼之例(參照扉頁)



(a) 原影像　　(b) 解碼影像

圖 2.3　圖 1 之中的正方形區塊的放大與三維顯示

編碼效率評估-資料量與失真的評估方法

因爲在破壞性編碼中，原影像與解碼後的影像間會有失眞產生，爲了評估編碼的效率，需要有一能同時評估資料量與失眞大小的方法。

一般資料量與影像尺寸或是畫格數目相關聯，將影像整體的資料量除以像素之正規化之後可以 bpp(bit/pixel)的單位表示，舉例來說，在圖 2.2(b)當中，爲 1229984 位元/(352×240 像素)約爲 1.54bpp。

另外，在第一章中所提到，在本書中使用的客觀評估值爲 MSE 與 PSNR。在原影像與解碼後影像分別以 $f(x,y)$、$f'(x,y)$表示，影像尺寸爲 $W\times H$的情形下，MSE 可寫爲(每一像素的平均平方誤差)：

$$\mathrm{MSE}=\sqrt{\frac{\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}(f(x,y)-f'(x,y))^2}{WH}} \tag{2.1}$$

但是在相同的MSE值下，8位元256階與10位元1024階的場合中，失真的情況也會大大的不同，因此需要加以正規化。因此，以表現出不同階的影像中所需的最大訊號振幅與 MSE 的比值以分貝(dB)表示之後的PSNR(Peak Signal to Noise Ratio)也經常被採用。特別是[23]，在8位元影像的場合當中可寫爲：

$$\mathrm{PSNR} = 20\log_{10}\frac{255}{\mathrm{MSE}} \tag{2.2}$$

在圖2.2中，MSE＝10.45，PSNR＝27.75dB(分貝)。另外，在不特別限制的情況下，彩色影像的MSE與PSNR值係在影像的亮度值Y分量間加以計算。

註　23. 請注意，PSNR值愈大，失真量愈小。

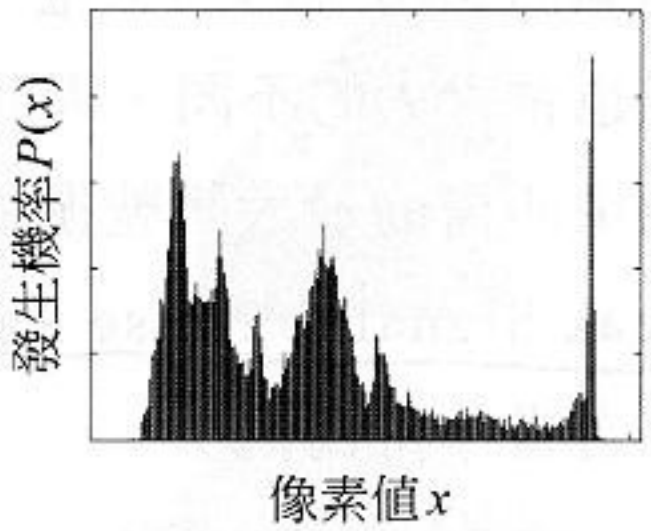
為什麼可以壓縮影像

看到的影像其實是「像素的排列」

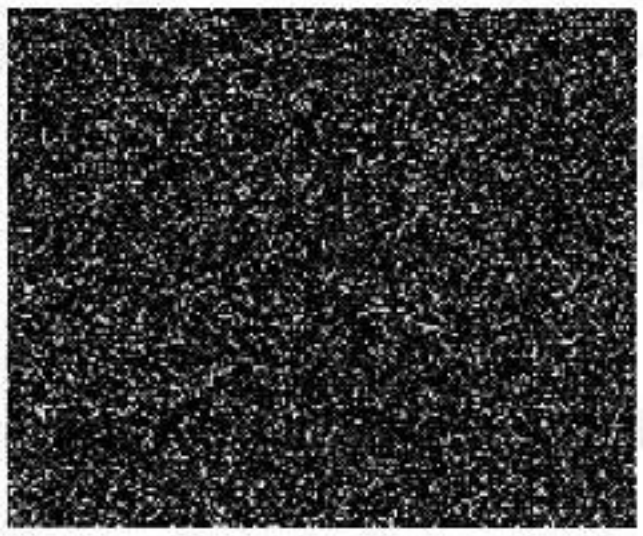
圖2.4(a)爲測試用影像「車用電話」(QCIF176×144像素)，係爲八位元的單色影像。計算影像上各像素值(0～255)的產生的總數，再除以總像素數目 25344 可以求得各像素值x的發生機率$P(x)$(圖 2.4(b))。另外，圖 2.4(c)是與(b)發生機率相同的像素全部以隨機方式產生，以176×144個加以配置而作成的「疑似影像」。

站在「影像是像素的排列」的立場，圖 2.4(c)也不能說不是「影像」，但在實際上(c)對我們來說也不過就是一片雜訊。因此，僅就像素的發生機率實在是無法區分「影像」與「雜訊」。

(a) 影像「車用電話」



(b) (a)當中像素值的發生機率



(c) 發生機率與(b)相同的疑似影像

圖 2.4　測試影像「車用電話」與疑似影像

edge detection

爲了讓「像素的排列」看來像是「影像」，至少需要：

- 存在亮度值激烈改變的「**邊緣**」。
- 整體的影像可以由邊緣分割出數個可以視爲一致的**區域**。

這兩個條件。

特別是後者無法僅由單一的像素值加以判別，需要**檢查與附近的像素關係**。

影像的基本性質

pixel

在此，如圖 2.5 所示在水平或是垂直線上相隔了 k 個像素的兩個像素取像素值 f_1 與 f_2，將影像上所有的像素值組合(f_1，f_2)畫在二維座標上

(此稱爲**分布圖**)。圖 2.6 爲對圖 2.4(a)的「影像」在水平、垂直方向上取 $k=1,2,4,8,16,32$ 之後的結果，圖 2.7 爲對圖 2.4(c)進行處理的結果。

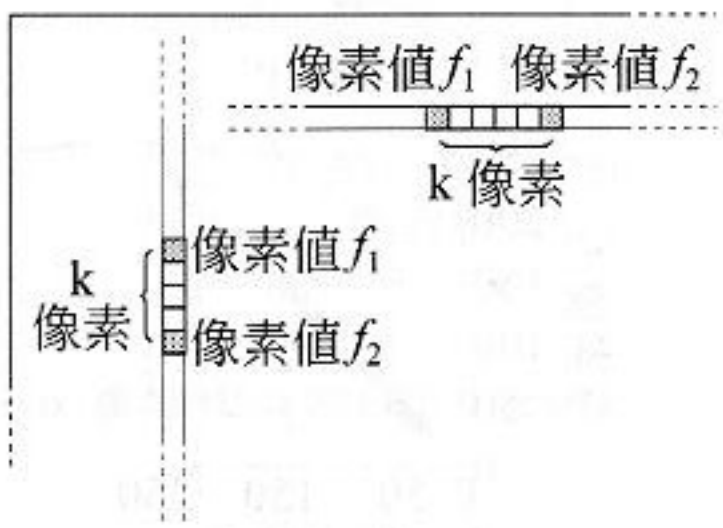


圖 2.5 影像中的像素組合(f_1 , f_2)的取法

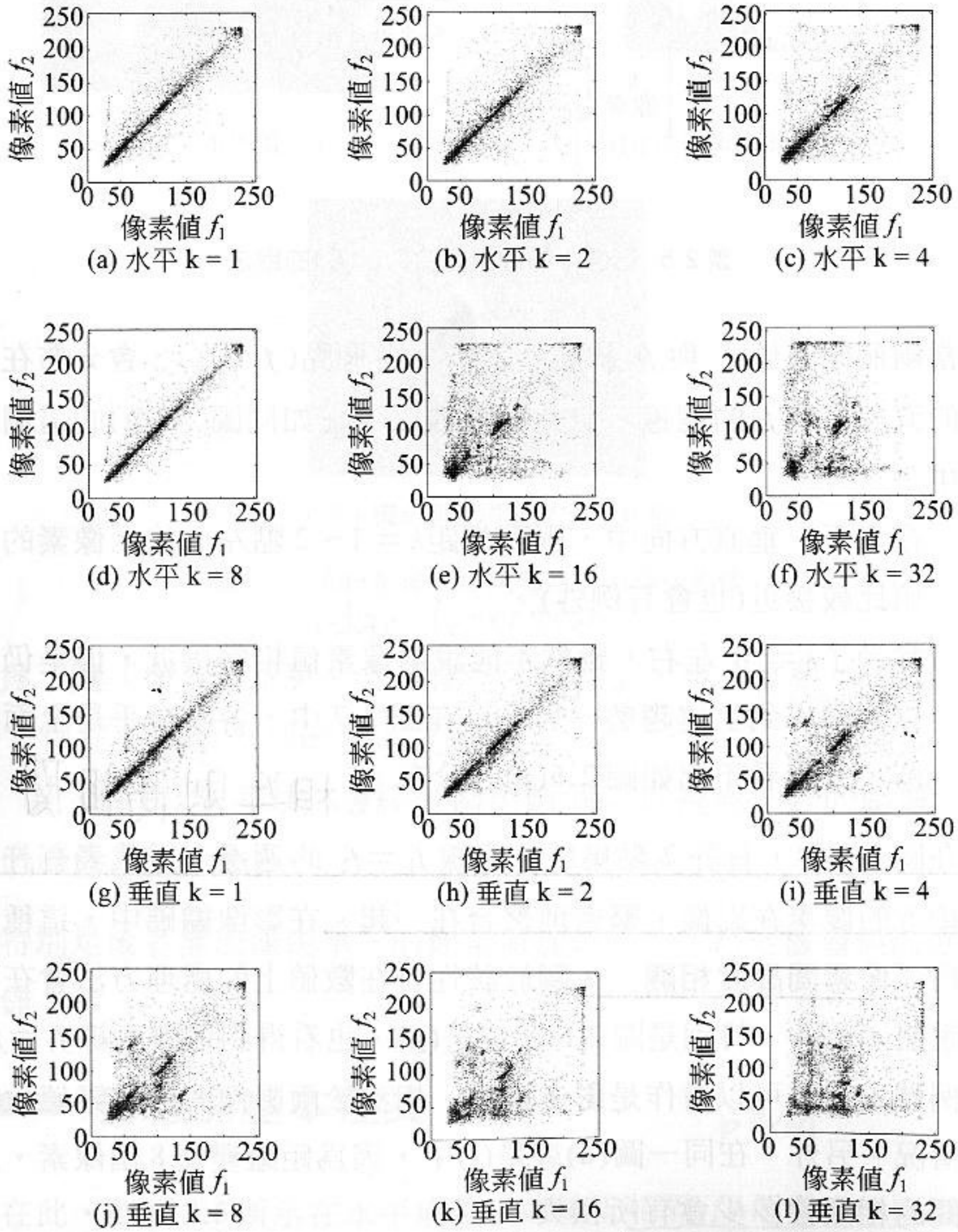
當兩個像素值 f_1 與 f_2 約略相等時，各個點(f_1 , f_2)，會分布在通過原點的直線 $f_1=f_2$ 的週邊。自此觀點觀之，在如同圖 2.4(a)的這個「影像」中：

① 在水平、垂直方向中，兩點相距 $k=1\sim2$ 點左右的兩像素的像素值比較接近(也會有例外)。

② 當到了 $k=8$ 左右，雖然不能說兩像素值相當接近，但是仍然可以觀察得到上述趨勢。另外，在圖 2.7 中，各點幾乎是全面地散佈，完全看不出如圖 2.6(a)的分布。 相近則高相關

在圖 2.6 中，有許多點集中在直線 $f_1=f_2$ 的週邊，這代表有許多像素值接近的像素在影像上緊密地聚合在一起。在影像編碼中，這種性質稱之爲「**像素間高度相關**」，關於該性質在數值上的處理方法會在後頭詳細說明。此外，特別是圖 2.6(a)或是(g)，也看得到遠離直線 $f_1=f_2$ 的點的例外，這些可以看作是**影像邊緣**，相當於兩影像間的像素值大幅改變的情況。另外，在同一圖(d)或是(j)中，因爲距離變爲 8 個像素，所以像素間的相關性多少會有所損失。

結果就是，不單是「像素的排列」是區分出非「雜訊」而是「影像」的理由，臨近像素間高度相關才是。這是「影像」之間廣泛共通的最基本性質。



(a) 水平 k = 1　(b) 水平 k = 2　(c) 水平 k = 4

(d) 水平 k = 8　(e) 水平 k = 16　(f) 水平 k = 32

(g) 垂直 k = 1　(h) 垂直 k = 2　(i) 垂直 k = 4

(j) 垂直 k = 8　(k) 垂直 k = 16　(l) 垂直 k = 32

圖 2.6　圖 2.4(a)裏頭相隔為k的像素值組合(f_1，f_2)之分布圖

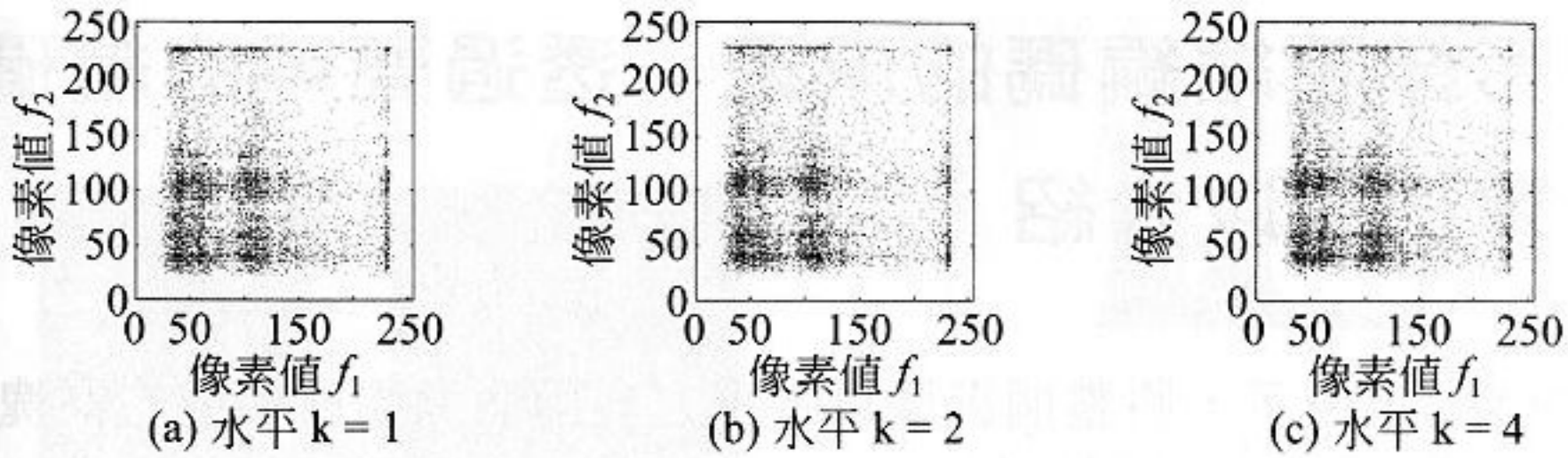


圖 2.7　圖 2.5(a)裏頭相隔為k的像素值組合(f_1 , f_2)之分布圖

為什麼影像可以壓縮—像素間的相關性與壓縮編碼

結論就是：因爲**鄰近的像素間高度相關**的性質，所以**影像**可以**壓縮編碼**。因此，如同圖 2.4(c) 像素間相關性較低的「雜訊」就無法壓縮[24]。

註　24. 正確地說，由於圖 2.4(c)的像素發生機率有所偏移，在套用熵編碼之後，仍可以做些許的壓縮。

如果想詳盡地了解其理由，需要研讀稱爲「資料率—失眞理論」的資訊理論，但是可惜的是在現階段關於數學的基礎仍嫌不足，因此無法充份地討論。接下來進入的則是，在學習實際使用的影像壓縮編碼方法時，所需要數學上的準備工作。

影像壓縮編碼的原理—透過簡單的編碼演算法加以介紹

如圖 2.8 所示，將整個影像以水平方向兩個像素所組成的區塊加以分解，分別將左右兩個像素稱爲 f_1 、f_2 。現在，將 f_1 、f_2 以下列形式轉換爲F_1 、F_2 ：

$$F_1 = f_2 + f_1 \text{ ，} F_2 = f_2 - f_1 \tag{2.3}$$

F_1 、F_2等像素的轉換結果，在本書中稱爲**轉換係數**，或是簡稱爲**係數**。

圖 2.9 爲將圖 2.4(a)與(c)代入式(2.3)之中，所得到左側爲F_1，右側爲$|F_2|$所構成的單色影像之顯示結果[25]。

註 25. 當所得F_2的值為負值的時候，在圖 2.9 中所表示者不為F_2，而為$|F_2|$。另外，如接下來所提到的，因為F_1的最大值為 510，實際上顯示的的像素值為$F_1/2$。

因爲圖 2.4(a)與(c)爲八位元 256 階的影像，因此F_1 、F_2的值分別落在 $0 \le F \le 510$ ；$-255 \le F_2 \le 255$ 的範圍內，如以二進位表示的話，需要用到九位元的長度。亦即，影像在套用式(2.3)的情形時資料尺寸會變大，因此並看不出特別的優點。

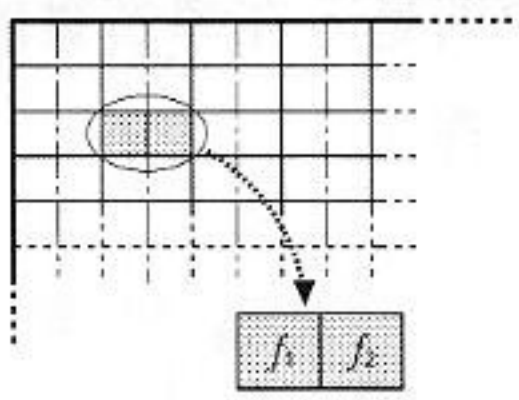
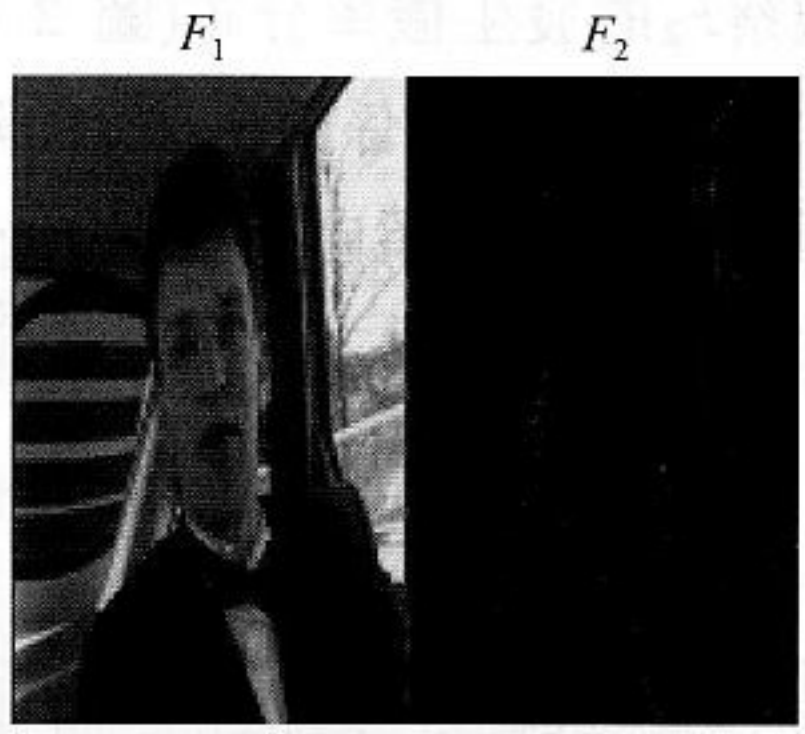
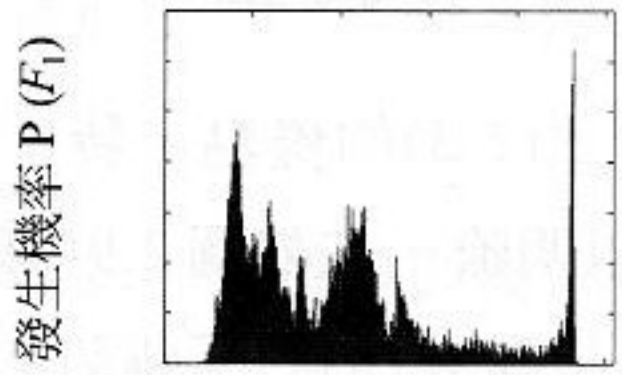


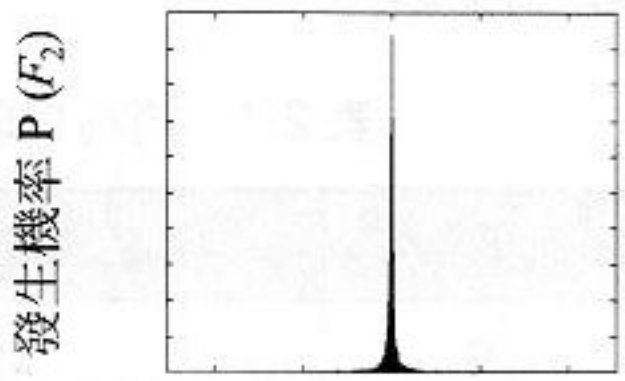
圖 2.8 將影像分解為水平方向兩像素區域的組合



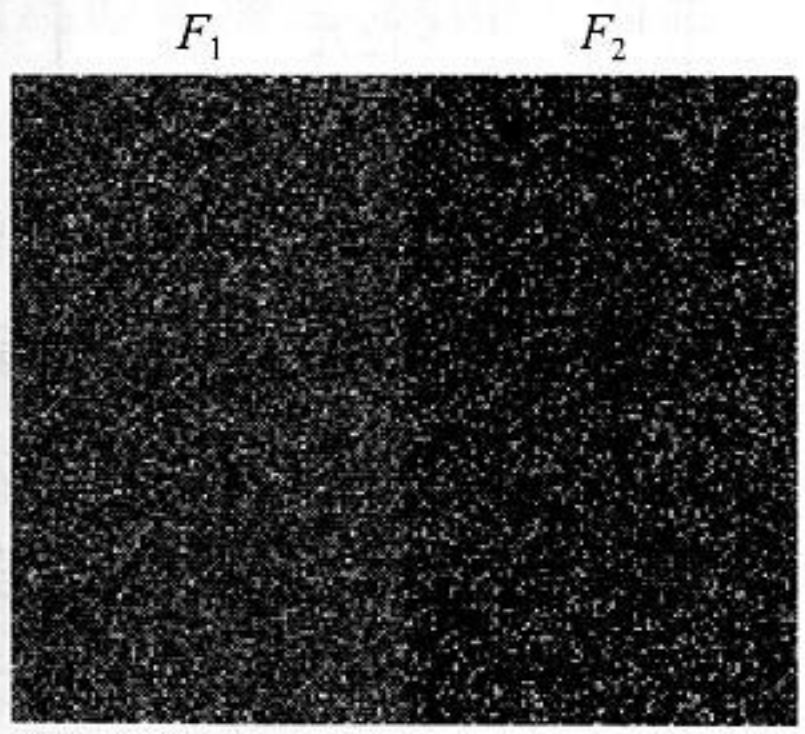
(a) 相對於圖 2-4(a)產生的結果



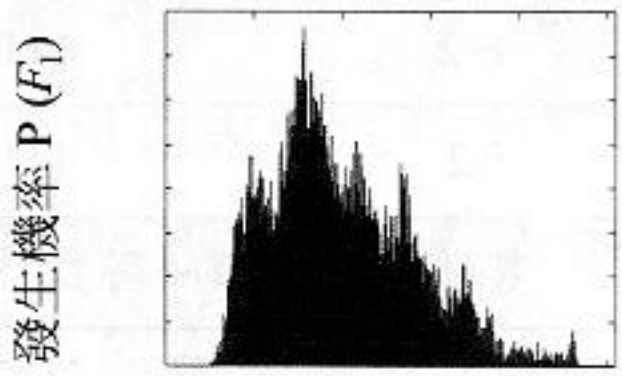
(b) 在(a)當中，F_1的發生機率



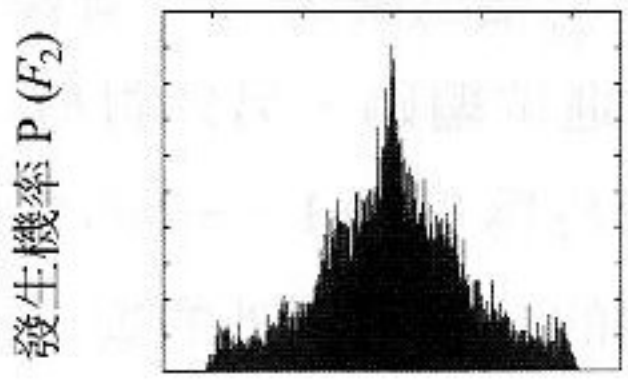
(c) 在(a)當中，F_2的發生機率



(d) 相對於圖 2-4(c)產生的結果



(e) 在(d)當中，F_1的發生機率



(f) 在(d)當中，F_2的發生機率

圖 2.9　圖 2.4(a)與(c)套用式(2.3)的轉換後所得到的結果，以及像素的發生機率分布

可逆編碼

考慮式(2.3)的優點，特別是在觀察F_2的發生機率分布(圖 2.9(c)與(f))就可以明瞭。在如圖 2.9(a)般的「影像」當中，係數F_2在 0 的附近集中產生，但是在(d)的「雜訊」中，就看不到這種傾向。事實上，(c)的發生機率中，在$|F_2| \le 2$範圍內非常的高，當$|F_2|$變大之後，其分布(機率)就急劇的降低。

表 2.1　將F_2加以編碼時所使用的二進位編碼

F_2	二進位編碼	編碼長度$L(F_2)$
0	10	2 位元
+1	1100	4 位元
−1	1101	4 位元
+2	11100	5 位元
−2	11101	5 位元
其他	0 +等長二進位編碼	10 位元

在此，依循「**發生機率較高的情形使用較短的符號長度**」的策略，對F_2進行二進位編碼[26]。 實際上，對F_1在 0～510 的範圍進行位元長度爲 9 的二進位編碼，另外對F_2以表 2.1 的方式進行二進位編碼。在表 2.1 中，對於F_2爲 0、±1、±2 的情形分別賦予位元長度爲 2、4、5 位元的編碼，其他的在表示例外的編碼(**例外(ESC)碼**)「0」之後，再加上該數值的 9 位元等長數值編碼[27]。

註　26. 此為壓縮編碼時所共通的常用方法，當發生機率不一致的時候，藉由賦予等的編碼可以降低整體編碼後的資料量。

27. 藉此，對於在 範圍內的各數值可以賦予一致的編碼，利用該編碼可以完整的將復原。這個事實留待讀者自行確認。

在將圖 2.4(a)以上述方式加以編碼的情形中，計算每一像素的平均編碼量。

首先，將F_1以 9 位元等長的方式加以編碼，整體編碼後的資料量爲 $88\times144\times9=114048$ 位元。另外，F_2整體編碼後的資料量爲其發生次數 $88\times144\times P(F_2)$再乘上編碼長度$L(F_2)$得到在$-255\le F_2\le255$範圍內加總之後的總和，可以用下式加以計算：

$$88\times144\times\sum_{F_2=-255}^{255}L(F_2)P(F_2) \tag{2.4}$$

使用圖 2.9(c)的發生機率$P(F_2)$加以計算之後，得到該值約爲 73939。因此，總編碼後資料量爲 114048 + 73939 = 187787 位元，每一像素約爲 187787/(176×144) = 7.42bpp。只壓縮一點點

因爲原影像爲 8bpp，因此上述方法僅可以說是成功地壓縮了部份資料。爲了易於說明，**本編碼方法為可逆**的，亦即，以表 2.1 解碼F_2之編碼後，可以式(2.3)進行逆轉換而完全復原原像素值 f_1 、f_2 。

另外，對圖 2.4(c)實施此方法，每一像素平均的編碼後資料量爲 9.34bpp。亦即，對「影像」來說是有效的演算法，對於如圖 2.4(c)的雜訊是不具有效作用的。

不可逆編碼

接著說明捨棄可逆性以得到較高壓縮比率的演算法。在此對式(2.3)的轉換係數F_1、F_2進行**量子化**，以降低係數的精度。舉例來說分別對F_1、F_2以量子化步進幅度 4 與 8 加以量子化。這就是說，相較於F_1、F_2以二進位長度爲 9 位元加以表示，相當於在末位分別捨去 2 位元與 3 位

元之後，位元長度分別爲 6 位元、7 位元。在此，F_1以 7 位元的方式等長地加以編碼，F_2以表 2.1 加以編碼(在此情形中，表 2.1 的最下面一行「其他」部份的編碼長度修正爲 7 位元)。因此，最後的每一像素資料量爲 4.82bpp(原影像與復原影像間的 PSNR 爲 43.6dB，畫質相當好)。

另外，這個非可逆編碼方法用於圖 2.4(c)時，編碼後資料量爲 6.45bpp，誤差 PSNR 爲爲 41.0dB，與圖 2.4(a)的結果相較，在編碼後資料量多用了 1.6bpp，但在PSNR值上卻以 3dB程度更爲劣化。因此與前述的可逆編碼的情形一樣，本方法亦無法有效地應用於如圖 2.4(c)的雜訊。

像素間相關程度的定量關係

自協方差函數

在此，先較深入地介紹像素間相關程度的數值計算方法。首先，在討論以整數x爲變數的一維訊號之後，再延伸至如同影像一般的多維訊號。

所謂「像素間相關程度」簡單來說就是「與臨近像素的像素值接近程度」，取其定量關係的計算方法有好幾個，通常使用稱爲自協方差的函數[28]。

註 28. 也有許多文獻採用「自共變異函數」，本書為了避免產生混淆，僅討論自協方差。

首先，說明稱爲**協方差**的量。考慮兩個一維訊號 $f(x)$、$g(x)$。定義爲：

$$C_{fg} = \lim_{x \to \infty} \frac{1}{2X} \sum_{x=-X}^{X} (f(x)-\mu_f)(g(x)-\mu_g)$$
$$\equiv E_x[(f(x)-\mu_f)(g(x)-\mu_g)] \tag{2.5}$$

稱爲 f 與 g 的協方差。其中使用符號 $E_x[\cdot]$，係避免第一式當中又有 lim 又有 Σ 使得式子變得太長的情形。μ_f 與 μ_g 的符號分別是 $\mu_f = E_x[f(x)]$、$\mu_g = E_x[g(x)]$，用來表示像素值在空間中的平均值。式(2.5)係表示平均值周圍波形的乘積 $(f(x)-\mu_f)(g(x)-\mu_g)$ 的平均。

相關

圖 2.10 爲波形 $f(x)$ 與 $g(x)$ 的例子。在同一圖(a)，$f(x)$ 與 $g(x)$ 極爲接近的情形中，式(2.5)的協方差 C_{fg} 爲接近 $f(x)$ 或是 $g(x)$ 之平方差的值。接著，如(b)所示者爲 $f(x)$ 與 $g(x)$ 較爲離開的情形 C_{fg} 的值較(a)的情形中爲小。再來看如同(c)，在 $f(x)$ 與 $g(x)$ 兩者差異相當大的情況中，式(2.5)中各點的 $(f(x)-\mu_f)(g(x)-\mu_g)$ 值因爲正負值相互抵消的關係，使得 $C_{jg} \cong 0$。

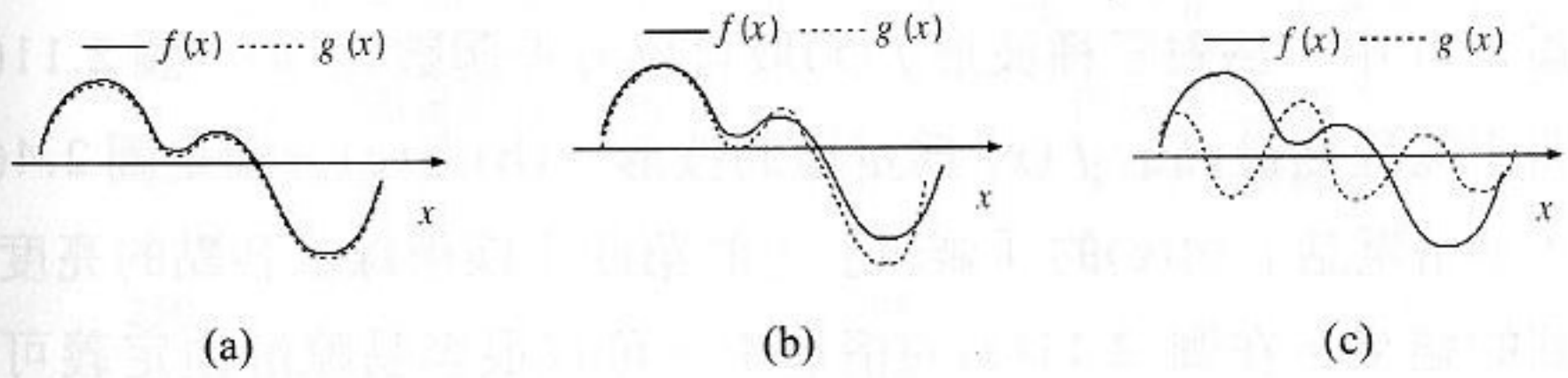


圖 2.10　$f(0)$、$g(x)$

藉由以上討論可以知道之，可藉由協方差**定量計算「兩波形間的接近程度」**。因此，爲了達成本節目的之評估「計算與臨近像素間的像素值接近程度」，將 $f(x)$ 稍微偏移一個位移 u (u 爲整數)的波形 $g(x) = f(x+u)$，代入式(2.5)。此時

- 式(2.5)爲位移 u 的函數。
- 平均值 $E[f(x)]$ 與 $E_x[g(x)] = E_x[f(x+u)]$ 的值相等(設爲 μ)。

考慮此點，可得到

$$C_{ff}(u) = E_x[(f(x)-\mu)(f(x+u)-\mu)] \tag{2.6}$$

稱爲**自協方差函數**[29]。

註 29. 前述的「自我相關函數」相當於式(2.6)中不減除平均值 的結果。在本書中為避免混淆，僅使用「自協方差」。

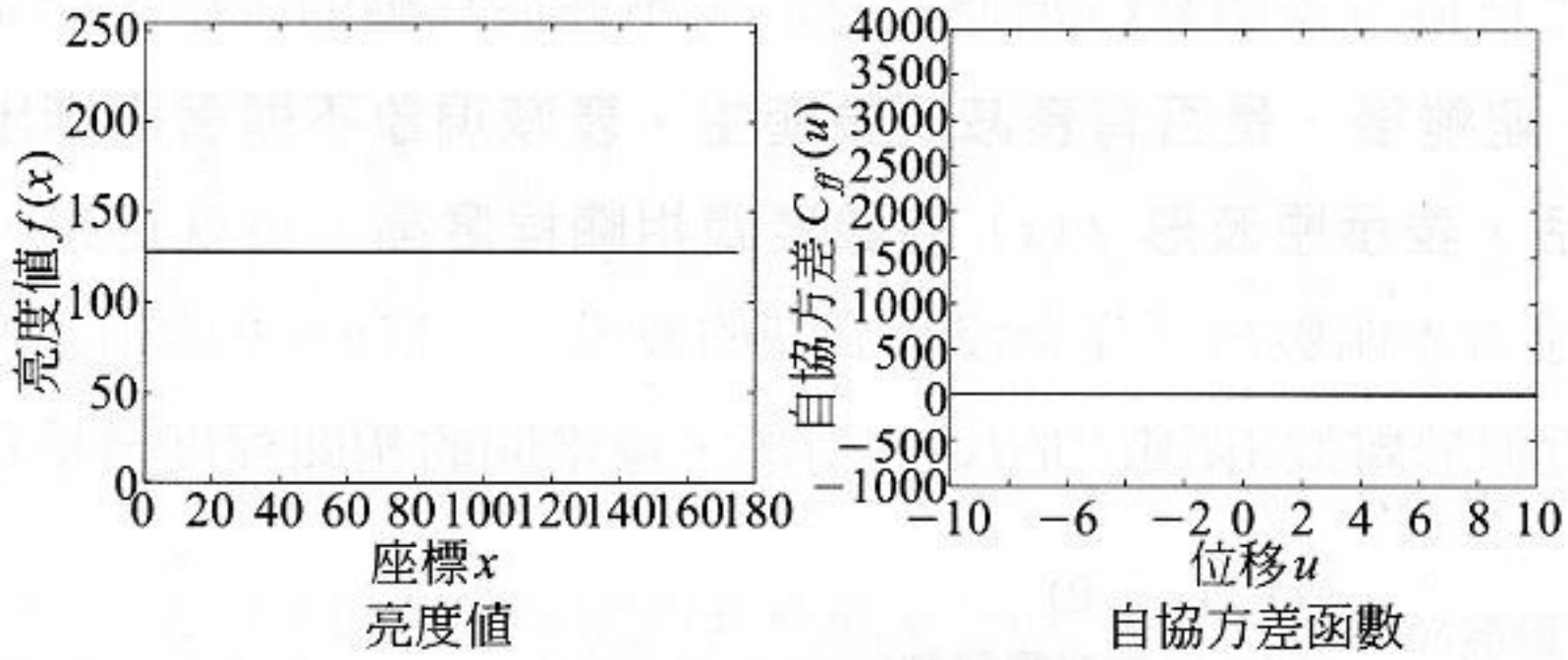
自協方差函數，亦繼承了協方差的性質，對於任意的u而言，可以簡單地證明[19][30]：

$$|C_{ff}(u)| \le C_{ff}(0) = \sigma^2 \tag{2.7}$$

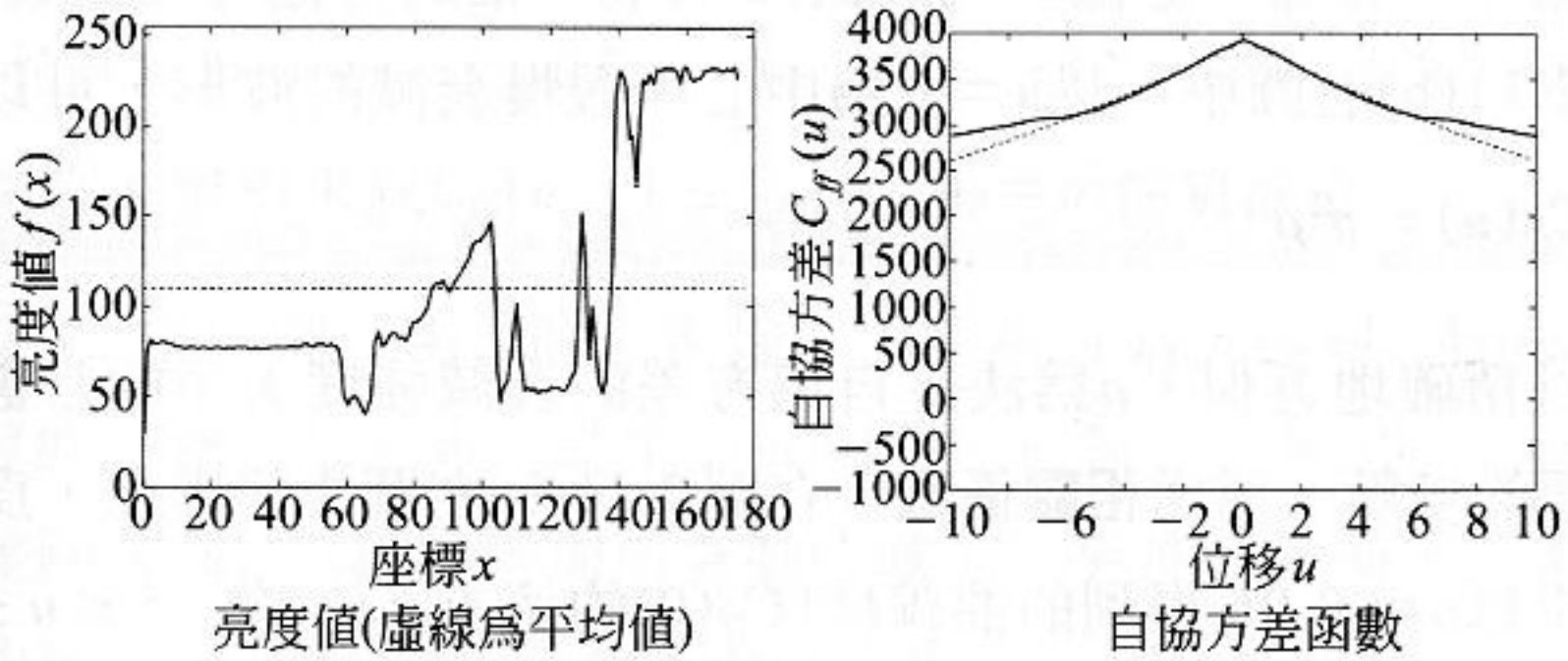
註 30. 在式(2.7)的最右側，$C_{ff}(u)$以$u=0$代入，與平方差的定義是一致的。

自協方差函數之例與其性質

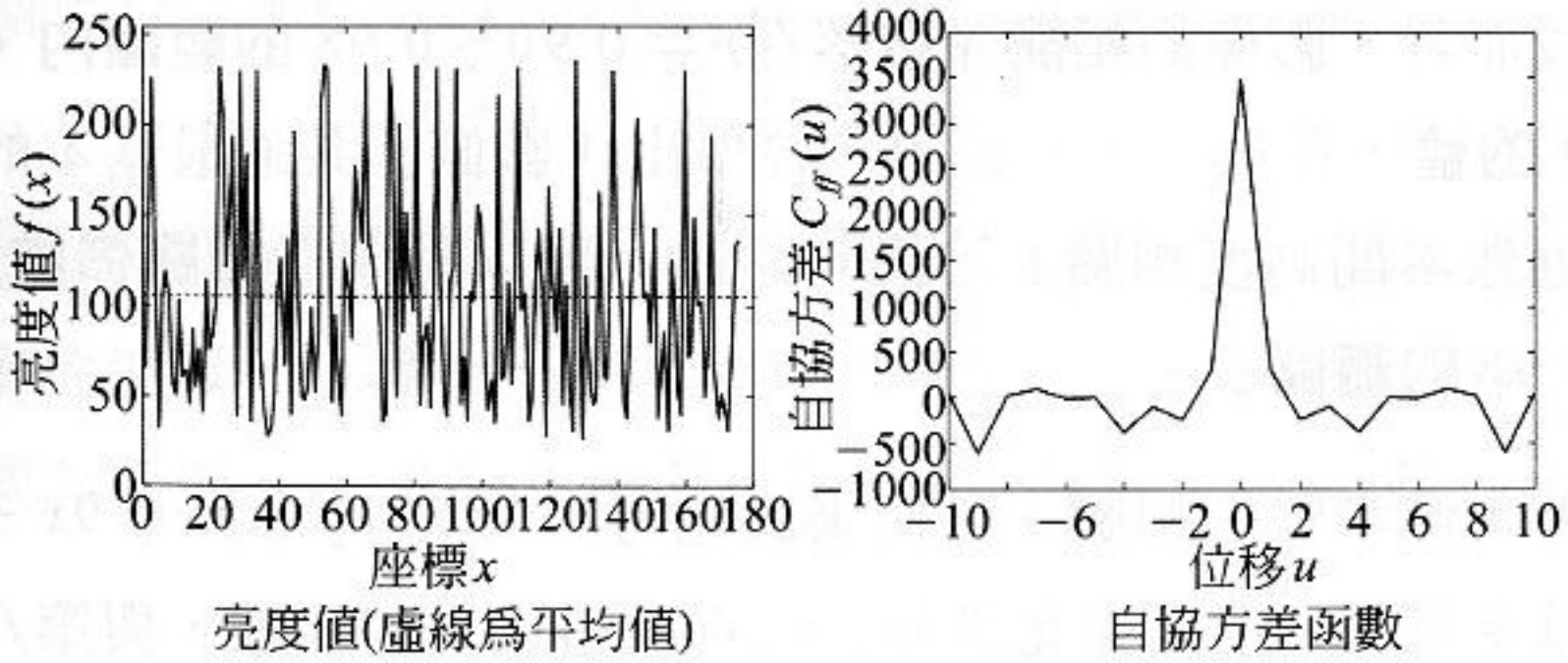
圖 2.11 中，爲對三種波形$f(x)$取自協方差函數$C_{ff}(u)$。圖 2.11(a)爲像素間相關程度最高的$f(x)$爲定值的波形，(b)與(c)分別是圖 2.4(a)的影像「車用電話」與(c)的「雜訊」上的第四十條橫線上各點的亮度變化所得到的結果。在圖 2.11(a)的情形中，可以很容易瞭解由定義可得各點的$C_{ff(u)}=0$。另外，如同一圖(c)中，由於與平均值之間的差距，波形以零爲中心激烈振動，當位移$u \ne 0$，波形在相加之後會因爲**正負相抵消**，而讓**自協方差向零收斂**。因此，除了$u=0$的情形之外，$C_{ff} \cong 0$。另外，在「影像」的情形之中，由於像素間的相關程度較高，在$u=\pm1 \sim \pm5$像素，不會像(c)產生「抵消」的現像，會呈現自$u=0$向左右平緩地減少的函數圖形。



(a) $f(x) = 128$ 固定值的情形



(b) 圖 2-4(a)的「影像」「車用電話」的第四十條橫線



(c) 圖 2-4(c)的「雜訊」的第四十條橫線

圖 2.11　三種波形的實例與其自協方差函數

對於自協方差函數C_{ff}所應注意的，**不在於其值的大小**，而是**其自原點$u=0$ 遠離後，是否有衰減現象產生，衰減現象不顯著而拖出長長的尾巴的話，表示原波形 $f(x)$ 的像素間相關程度高**。這就是說，圖 2.11 (a)的自協方差函數爲「完全沒有衰減的波形」，自$u=0$開始急速衰減，可以說近似於離散Δ函數[31]的(c)的情形，像素間的相關程度幾乎是沒有。

註 31. 函數$\delta(x)=\begin{cases}1 & (x=0)\\ 0 & (x\text{ 為非零整數})\end{cases}$

實際上，各種「影像」代入式(2.6)取一維的自協方差函數， 會得到如圖 2.11(b)右圖中，以$u=0$爲中心，緩慢衰減的波形，可以用：

$$C_{ff}(u)\simeq\sigma^2\rho^{|u|} \tag{2.8}$$

加以精確地近似。ρ爲決定自協方差的衰減強度大小的常數，爲一非常重要的參數，稱爲**相關係數**。在圖 2.11(b)右圖中的虛線，爲式(2.8)中模型中以$\rho=0.96$ 得到的曲線(以$C_{ff}(0)$的值求ρ^2)，在$-5\le u\le 5$ 的範圍內，可以相當精確地近似於自協方差的實際測量值。

一般而言，影像的**相關係數**落在$\rho=0.90\sim0.98$ **的範圍內，為一非常接近 1 的值**，在第二十八頁中，曾舉出「影像」共通最基本的性質爲「與臨近像素間高度相關」，換句話說，可以說成「**相關係數落在**$\rho=0.90\sim0.98$ **的範圍**」。

另外，像素間的相關，舉例來說也可以用$E_x[|f(x)-f9x+u)|]$等函數加以定量化。但是在此強調式(2.6)的自協方差函數，與第八章中討論的**資料率－失真理論有著相當深的關聯**。

擴展至二維訊號

如以上所述的一維自協方差函數可以輕易地擴展至二維或是多維。二維(影像)訊號 $f(x,y)$ (x、y爲整數)，其自協方差函數可寫爲：

$$C_{ff}(u,v)=\lim_{X,Y\to\infty}\frac{1}{4XY}\sum_{y=-Y}^{Y}\sum_{x=-X}^{X}(f(x,y)-\mu)(f(x+u,y+v)-\mu)$$
$$\equiv E_{xy}[(f(x,y)-\mu)(f(x+u,y+u)-\mu)] \tag{2.9}$$

μ爲像素值在空間中的平均值$E_{xy}[f(x,y)]$。式(2.9)繼承了在一維情況中的各種性質，舉例來說$C_{ff}(u,v)\le C_{ff}(0,0)=\sigma^2$性質成立。

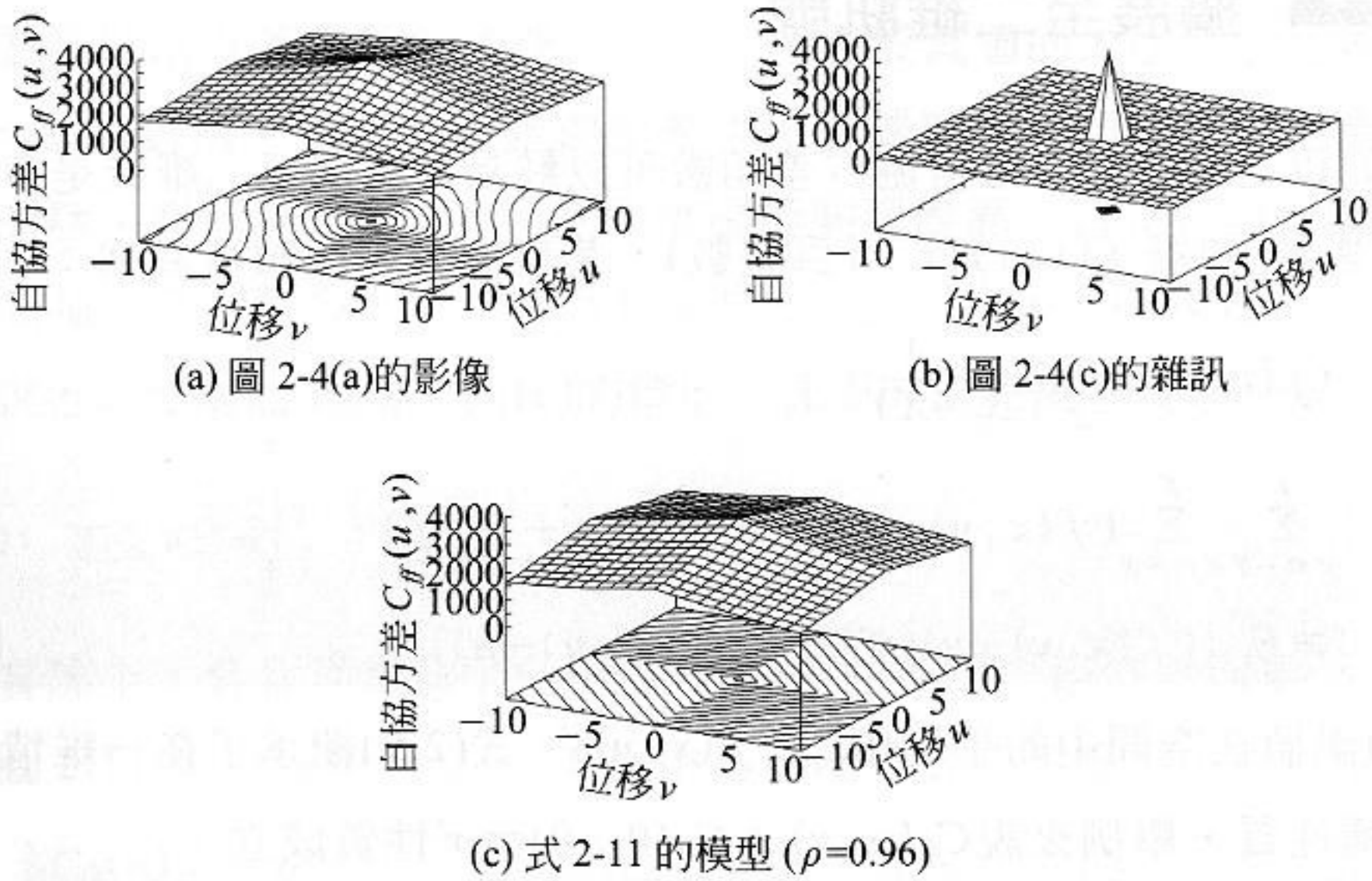
圖 2.12 爲對圖 2.4(a)的「影像」「車用電話」與同一圖(c)的「雜訊」計算的二維自協方差。結果如同預測，相對於「雜訊」的情況中，自協方差隨著$|u|$、$|v|$的增加而急劇的減少，另外一方面，「影像」拖著長長的尾巴，其形狀如同式(2.8)可以用

$$C_{ff}(u,v)=\sigma^2\rho^{\sqrt{u^2+v^2}} \tag{2.10}$$

或是

$$C_{ff}(u,v)=\sigma^2\rho^{|u|+|v|}, \qquad \sigma^2\rho_H{}^{|u|}\rho_V{}^{|v|} \tag{2.11}$$

加以精確地近似。(ρ_H與ρ_V分別爲水平及垂直方向的相關係數)。圖 2.12 (c)爲以式 2.11 中，$\rho=0.96$的繪圖結果，與(a)相較，係良好的近似結果。



(a) 圖 2-4(a)的影像

(b) 圖 2-4(c)的雜訊

(c) 式 2-11 的模型 (ρ=0.96)

圖 2.12　二維自協方差函數的例子

影像編碼演算法的一般組成

不可逆編碼演算法的再檢討

先前基於式(2.3)表示的不可逆編碼演算法係由下列三個步驟所組成：

① 以像素的組合(區塊)分割影像，得到轉換係數。

② 量子化轉換係數。

③ 以量子化後係數之發生機率爲基礎，進行二進位編碼。

在此，再次討論這三個步驟。

關於步驟①

在步驟①當中，以式(2.3)將水平方向的像素組合(f_1 , f_2)轉換爲(F_1 , F_2)。圖 2.13 爲分別以F_1、F_2爲水平、垂直軸，對於圖 2.4(a)影像

的各轉換係數加以繪製。由式(2.3)可以知道，此與圖 2.6(a)旋轉 45 度後的結果是一樣的。換句話說，這兩個圖不過是以兩個不同角度來觀察相同的分布。但是，在圖 2.13 的各點，雖自 $f_1 - f_2$ 坐標系看來高度相關，但是在$F_1 - F_2$坐標系上F_1、F_2之間的相關程度相當地低[32]。

註　32. 因為並不是沿著直線$F_1 = F_2$分布。因此看不到$F_1 \approx F_2$的性質，所以可以得到低度相關的結論。

實際上，計算 $f_1 - f_2$ 之間與$F_1 - F_2$之間的協方差後可以得到$C_{f_1 f_2} = 3123$，$C_{F_1F_2} = 1.88$，$C_{f_1 f_2} \gg C_{F_1F_2}$的結果，因此藉由式(2.3)的轉換，藉由將 f_1、f_2 表示爲F_1、F_2，可以去除像素間的相關程度，並且可藉數值結果確認此事。

綜上所述，①可以稱爲**去除像素間的相關程度(非相關化)**的步驟。

關於步驟②

在步驟②中，將非相關後的係數各別量子化後以達到減少資料位元量的目的。如第九頁所述，一般在量子化／反量子化的過程中會伴隨誤差的產生，進行量子化後，編碼／解碼過程係不可逆的，會造成影響。

關於步驟③

在步驟③中，並不是對於量子化後的各係數進行等長的二進位編碼，而是對發生頻率較大的編碼值分配較短的編碼，藉此削減編碼後的資料量。在影像編碼領域中，這種運算稱爲熵編碼或是可變長度編碼。

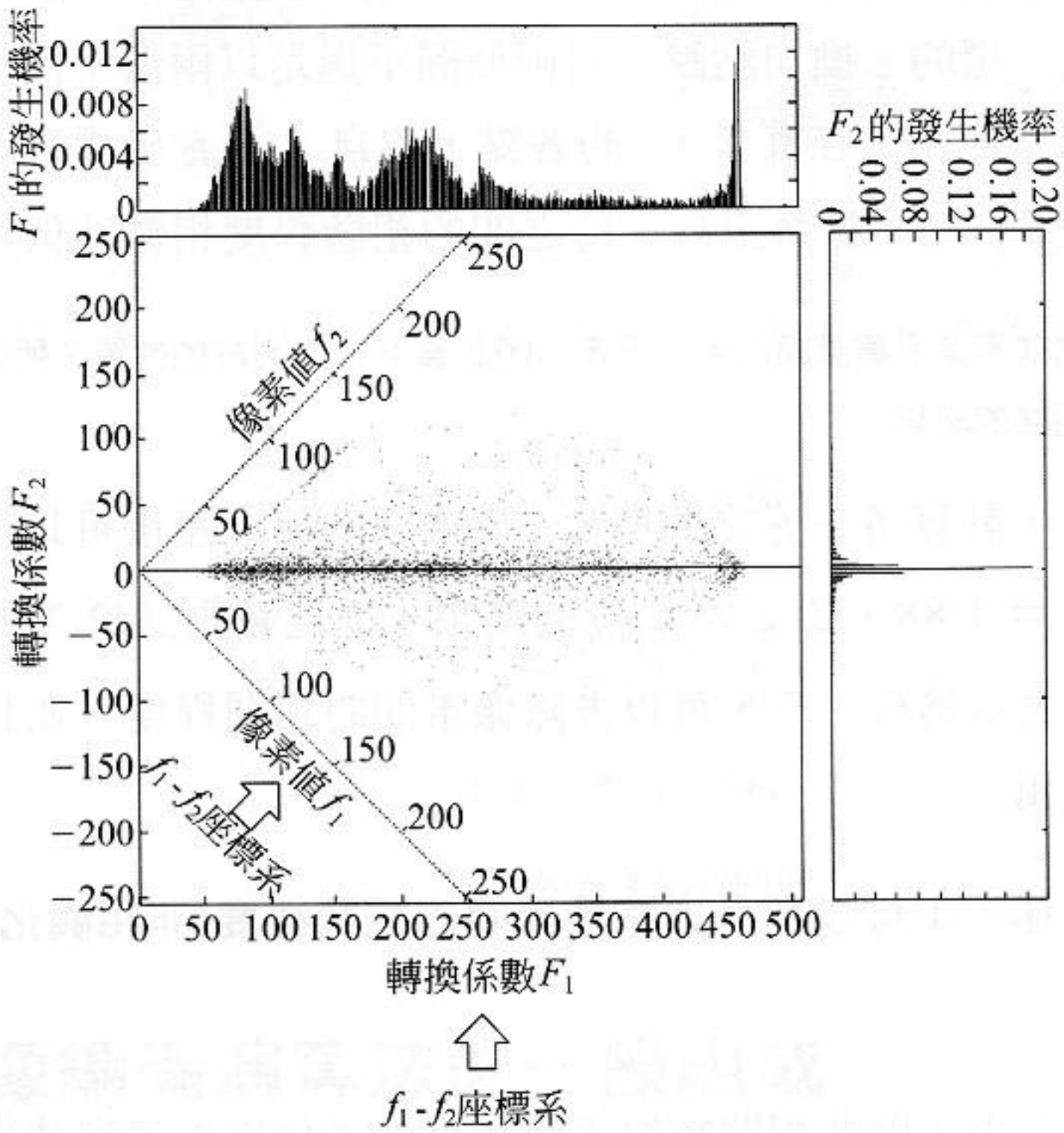


圖 2.13　轉換係數F_1-F_2的分布圖

影像編碼演算法之一般組成

許多現在廣泛使用的影像編碼演算法，如上所述，由**去除像素間相關程度部份**、**量子化部份**與**熵編碼部份**所組成。

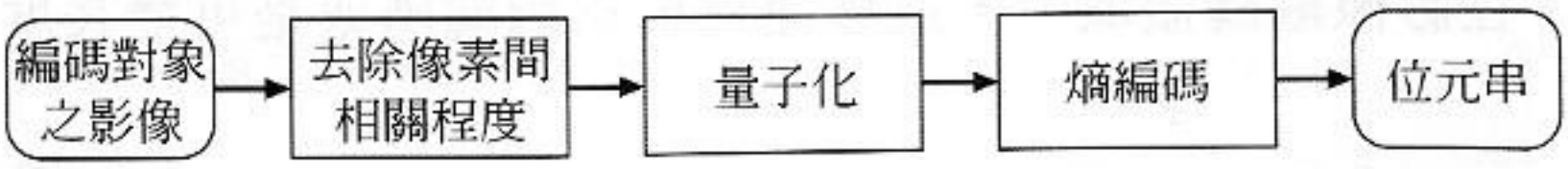


圖 2.14　影像編碼演算法之一般組成

首先，在之前的不可逆編碼演算法的例子中，僅能將 8bpp 的影像壓縮爲 4.82bpp 程度爲止。如何能在不使畫質劣化的條件下得到更高的壓縮率。在此歸納基於迄今爲止的知識所考慮到的的解決方法。

① **加大轉換的區塊範圍**

在式(2.3)中，得到對於由水平方向上兩個像素組成的區塊之轉換係數，另外，由圖 2.12(a)可得知，影像的像素間相關程度即使是離開數個像素之後仍然相當高，因此，**加大轉換的區塊範圍**，可以同時轉換更多的像素，以提高壓縮效率。這種加大轉換區域的範圍，式(2.3)的一般轉換方式，將在第五章中詳細介紹。

② **不僅在水平方向，也在垂直方向進行轉換**

由圖 2.12(a)可以得知，像素間的相關程度在水平與垂直方向幾乎同有相同的傾向，因此不僅在水平方向可以去除其相關程度，在垂直方向上同時去除的話，可以得到更高的壓縮率。這觀念，可運用於對影像進行各種轉換之時，實際上 ，現在的影像編碼演算法中，離散餘弦轉換(Discrete Cosine Transformation :DCT)。就是在水平方向與垂直方向兩個方向上同時進行，以去除相關程度。

③ **反覆進行轉換**

在水平方向代入式(2.3)所得到圖 2.9(a)的F_1部份，仍可以看成是「影像」。亦即F_1部份像素間的相關程度仍然相當高。因此，對於F_1部份再度代入式(2.3)，將進一步地分成F_1與F_2。反覆進行後，F_1部份的相關程度會逐漸降低。這種處理，實際上以**小波轉換**而廣爲週知。但是，本書由於篇幅的關係，不會涉及進一步的討論。

④ **加大量子化的增加幅度**

如此一來雖然確實可以減少總位元量，但是同時會增大解碼影像的失眞。因此，對於轉換係數量子化步進幅度的選取，需要作理論上的討論，在第四章中會對這個問題詳加討論。

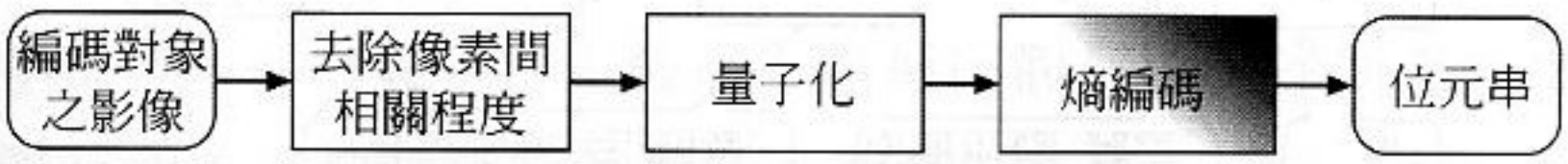
⑤ **提升熵編碼的性能**

之前舉例所使用的表 2.1 中之二進位編碼，係單就發生機率高的係數值以較短位元長度的編碼加以表示之觀念加以製作。更爲系統化的熵編碼的方法，將在接下來的第三章加以討論。

3章

熵編碼

本章將討論第二章所述之「編碼演算法之組成」當中的「熵編碼部份」。



並說明專用於影像編碼的熵編碼之系統化組成方式。

前言—同時說明用語與基礎知識

符號與資訊來源

Information Theory

本章的內容涉及「資訊理論」[23][24]領域，而這方面的討論使用了特有的用語。在本節中，將介紹在理解熵編碼時所必需的最低限度基礎知識與用語，同時對於吾人所要處理的問題舉出實例加以說明。

所謂熵編碼，就是將編碼對象的數值與記號以**(字)碼**表示的處理方式。在資訊理論中，將作爲編碼對象的數值、記號稱爲**符號**；產生符號的來源稱爲**資訊來源**。表示各符號的數碼(語言)中，幾乎沒有例外地先使用由「0」與「1」的組合組成的二進位編碼[33]。

註 33. 亦可稱為二進制；理論上亦有三進位、四進位的組成，但是二進制中的「0」與「1」對應到數位計算機中的兩個狀態，開與關，因此配合數位計算機的相容性，幾乎沒有例外地優先使用二進位制。

考慮具體的實例。圖 3.1 爲一資訊來源S與其(熵)編碼器，假設作爲資訊來源產生的符號爲**1 至 8 的整數**。此處不考慮符號具有的「意義」，僅著眼在「資訊來源產生符號的機制」，或是「產生符號的統計性質」。

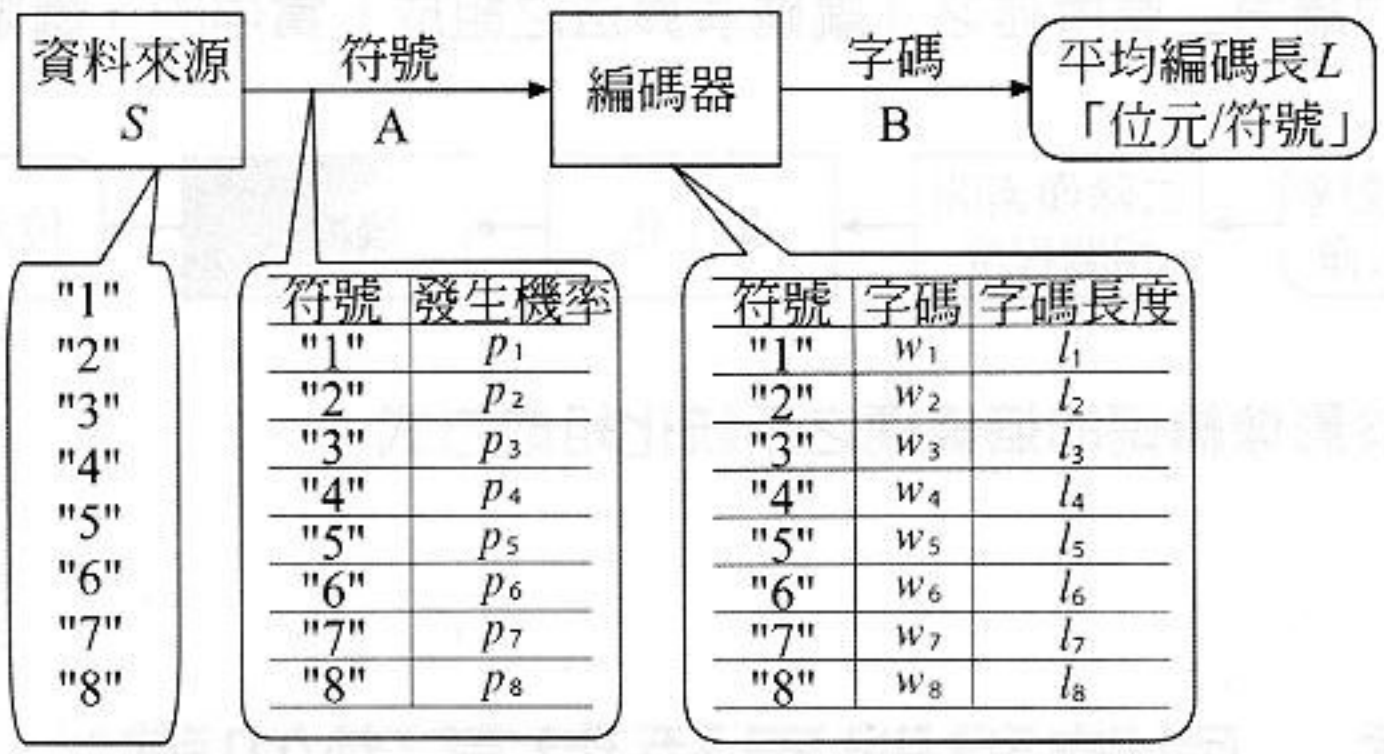


圖 3.1 資訊來源與熵編碼器的模型

即使是同有相同符號集合的資訊來源，符號的統計性質也會因爲其產生機制而不同。舉例來說，「在裏頭有個精靈在擲正八面體的骰子，並輸出朝上那面的數字」的資訊來源S_1。在此情況下，觀察幾次輸出的符號後，舉例來說會得到如式(3.1)的數列。每次擲骰子的實驗爲各自獨立的，在此情況下各符號爲在**統計上是獨立**的，某一次實驗的結果(骰

子朝上那面)不會影響其他次的實驗結果。這種資訊來源稱爲**非記憶資訊來源**。式(3.1)爲使用公正的骰子的一例，但是當精靈使用了不公正的骰子時可能會得到如式(3.2)的結果。這種情況下，雖然符號不是以相等的機率產生，但是在統計上仍獨立，因此仍爲非記憶資訊來源。

$$S_1：1463825738463158672521 74\cdots \tag{3.1}$$

$$S_2：781382758346576762825747\cdots \tag{3.2}$$

$$S_3：787678765423545321123234\cdots \tag{3.3}$$

$$S_4：765856878756141324123432\cdots \tag{3.4}$$

另外，式(3.3)爲記憶性資訊來源所產生符號列的一例，可以很容易看得出有非常高的機率在某些符號與前一個符號間的差在 1 以內。因爲前一個符號值與會對接下來產生的符號值造成影響，因此稱爲「具有記憶」。最後，式(3.4)的符號列中，前半部中 5～8 的發生機率較高，反之在後半部 1～4 的發生機率較高。這種因觀察位置不同而使得符號的統計性質有所差異的資訊來源稱爲**不定資訊來源**。反過來說，在當中，骰子各面的發生機率是固定的，分類爲**固定資訊來源**。

平均編碼長

現在，假設圖 3.1 的資訊來源爲**固定**的，將N 個(N 爲相當大的數)符號加以編碼後，總體編碼資料量爲R (位元)。此時，每一個符號的**平均編碼長**L爲：

$$L = \sum_i p_i l_i \quad (\text{每個符號的位元長}) \tag{3.5}$$

另外可以用各符號的發生機率以異於式(3.5)的形式表示L。因爲假設資訊來源爲固定的，各符號的產生次數夠多之後，藉由對於產生數目

加以正規化，可以將如圖 3.1，求得符號s_i的發生機率p_i $(i = 1,2,\cdots,8)$。在編碼過程中，對於第「i」個符號寫作字碼w_i。現在將w_i的**編碼長度**(「0」與「1」的個數)以l_i表示。此時，可以將平均編碼長度L以編碼長度l_i的期待值表示，寫作：

$$L = \sum_i p_i l_i \quad \text{(每個符號的位元長)} \tag{3.6}$$

減少平均編碼長度L

參照式(3.6)爲了減少平均編碼長度，因此最好依循以下原則：

> 著眼於各符號的發生機率p_i，p_i較大的符號s_i儘可能的以編碼長度l_i較短的字碼表示。

因此，實際的熵編碼器如圖 3.2 所示，一般包含以下兩個部份：

- 諸如推測自資訊來源產生的符號之發生機率的**模式化統計性質**部份[34]。
- 基於所得到的統計性質(發生機率)，以**字碼表達各符號**。

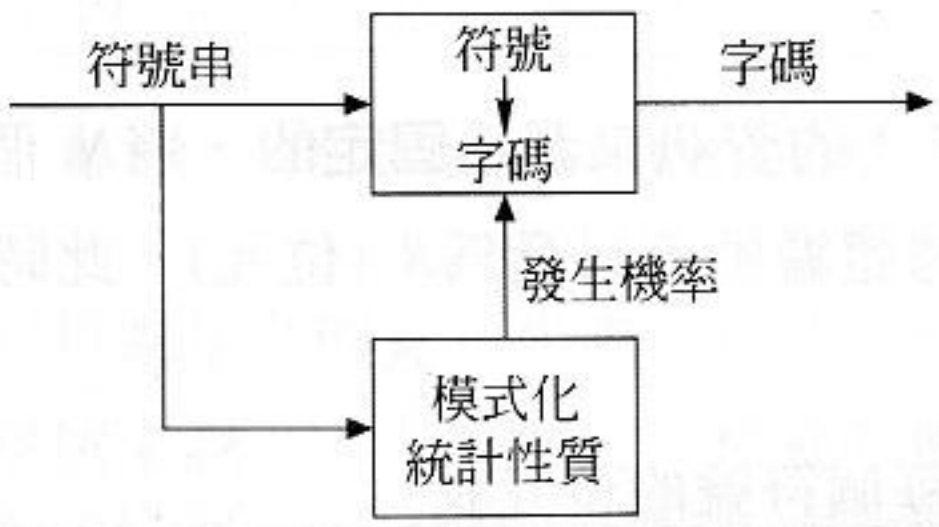


圖 3.2　熵編碼器的一般組成

註　34. 稱為可變長度編碼。

當資訊來源爲**非記憶且為固定**的情形中，符號的發生機率與符號列的位置無關而爲固定，而且先前產生的符號不會對後來發生的符號產生任何影響，因此對於產生的各符號可以沿著上述原則以字碼加以表示，因此可以進行效率良好的編碼。另外，在資訊來源爲記憶性的或是不定的情形中，需要進行更複雜的處理。但是，在本書中所討論的並非爲一般編碼方法，而是**特別用在影像編碼的熵編碼方法**，首先自此觀點來觀察作爲編碼對象的資訊來源或是熵編碼化器的特徵。

應用於影像編碼的假設

參照本章之初，表示影像編碼演算法之一般構成的圖，在圖 3.3 中以虛線圍住的部份相當於與熵編碼器相對的資訊來源。在此情況中，輸入的符號(量子化器輸出的結果)，因爲已經在之前去除相關程度，所以**是互相不相關但不獨立的**。在統計上獨立的機率變數是不相關的，但是反之並不成立(不過，在機率變數依循常態分布的情況中，不相關與獨立是一樣的)[37]。因此，圖 3.3 的符號本來就無法視爲非記憶的獨立資訊來源加以處理。

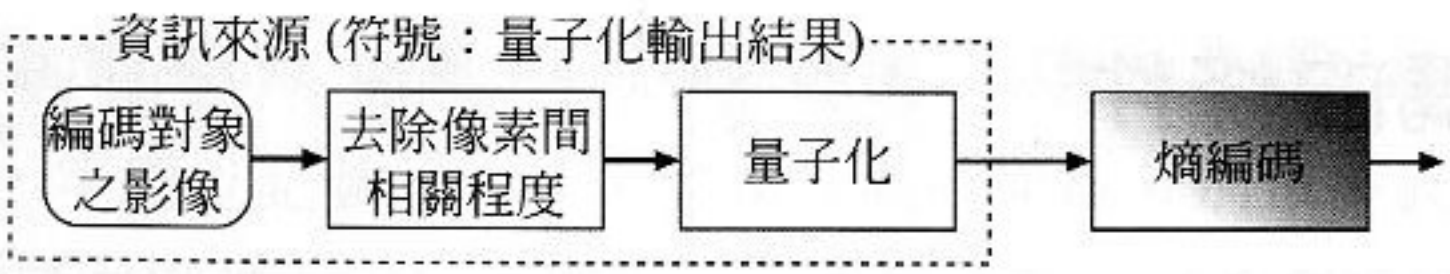


圖 3.3　影像碼中的熵編碼與資訊來源

記憶性的資訊來源，可視爲**馬可夫**(Markov)**資訊來源**加以模式化(亦可稱爲**情境模式化**)，一般基於此進行編碼[27]。但是，這種模型的影像依存度相當高，因此無法對於任意影像得到有效的共通模型。另外，

與符號爲非記憶的獨立編碼的情況相較，處理用記憶體顯著地增加，但是無法對所有影像得到足夠的利益。

另外，在如同圖 3.3 的情形中，量子化器輸出的結果實際上是不固定的，爲了追蹤其統計性質，在熵編碼時需運用**適當的方法**。但是，與先前的情況相同，因此使得計算量增加的同時，常無法得到相對應的改善效果。

因此，在 JPEG 或 MPEG 等標準中，主要是對圖 3.3 中輸出的符號視爲不定、非記憶性加以處理。這就是說，在制定標準的同時積體電路技術與電腦的運算能力較現在要落後得多，因此在即時運算時，減少計算量與記憶體使用量是不可或缺的。另外，一般在圖 3.3 的量子化器輸出結果可以用近似於正規分布加以模式化，時如同先前註釋所述「非相關性＝獨立性」。因此，這個事實使得將符號視爲非記憶性加以處理具有正當性。

本書最終的目的在於使讀者理解 JPEG 或 MPEG 等標準的基本演算法，同時限於篇幅，之後解說的是，對固定非記憶資訊來源的熵編碼方法(實際上爲後述的固定使用 Huffman 編碼的方法)。

解碼的條件

一致解碼性與瞬間解碼性

熵編碼器所需具備的條件，在儘可能地縮減平均編碼長度L之前，必須滿足下列兩個條件：

- 編碼後的符號在解碼後是一致的(**一致解碼性**)。

- 在接收各編碼最後一位元的資料時，即使沒有後續的位元也可以解碼。(瞬間解碼性)。

舉例來說，考慮表 3.1 中相對於符號s_1～s_4的四種編碼，可以輕易地確定編碼 I 可以一致解碼、瞬間解碼。編碼 II 在對於字碼「11」會有符號s_1s_1或是符號s_2的兩種解釋而無法一致解碼。編碼 III 雖然可以一致解碼，但是在剛接收到「1」的時候，無法馬上對於現有的字碼加以解碼。相對來說，編碼 IV 爲可以同時滿足一致解碼與瞬間解碼的編碼。

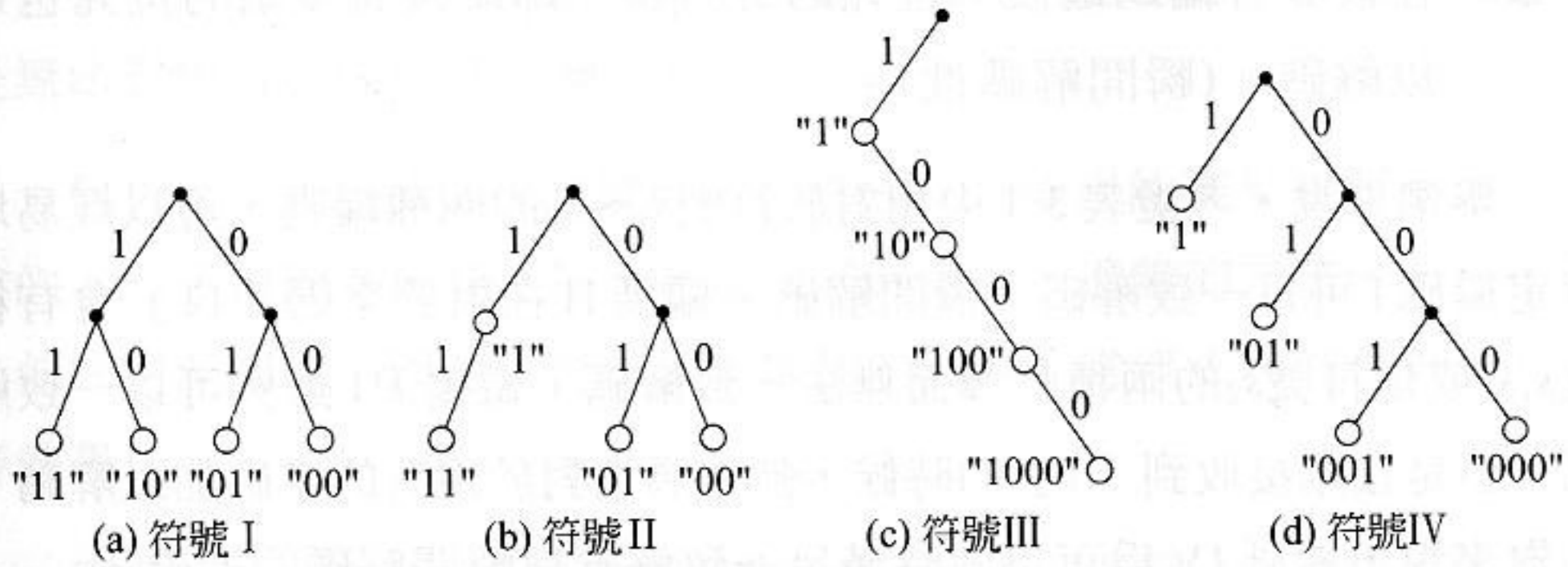
表 3.1　編碼的組成例

	編碼 I	編碼 II	編碼 III	編碼 IV
s_1	11	1	1	1
s_2	10	11	10	01
s_3	01	01	100	001
s_4	00	00	1000	000

編碼的條件

表 3.1 的編碼可以如圖 3.4 以樹狀圖加以表現。此稱爲**編碼樹**[35]。在圖 3.4 中，字碼以記號○表示，到達該點的樹枝分別依序表示「0」與「1」。字碼在編碼 I、IV 中僅以編碼樹的**端點**，另外在編碼 II、III 樹的**中間點**也可以表示。在後者的情形中，一個字碼會與其他字碼初前頭的一部份(稱爲字頭)相同，無法一致或瞬間解碼。

註　35. 在此使用圖學理論的用語，編碼樹的起點稱為「根部」，末端稱為「葉」。



(a) 符號 I　(b) 符號 II　(c) 符號 III　(d) 符號 IV

圖 3.4　編碼樹

如上所述，可得到接下來的定理。

定理 1

全部的字碼位在編碼樹的端點爲同時滿足一致與瞬間解碼的充份必要條件。

另外，關於編碼的編碼長度l_i $(i=1,\cdots N)$，有著以下的定理：

定理 2

同時滿足一致與瞬間解碼的充份必要條件爲，第i個編碼的編碼長度滿足以下式子：

$$\sum_{i=1}^{N} 2^{-l_i} \le 1 \tag{3.7}$$

反之不然

式 3.7 稱爲**Kraft-McMillan**不等式(其直觀說明詳見參考文獻[25])，另外，並不是所有滿足了 Kraft-McMillan 不等式就同時滿足一致與瞬間解碼，需注意以下事項：

- 當編碼長度 滿足 Kraft-McMillan 不等式的情形中，其編碼長**足以構成**滿足一致與瞬間解碼的編碼(**至少有一組滿足一致與瞬間解碼的編碼存在**)。
- 當不滿足的情況中，其編碼長**不足以構成**滿足一致與瞬間解碼的編碼(**不存在**)。

資訊來源編碼定理

考慮由N 種符號組成的**非記憶資訊來源**，$S=\{s_1, s_2, \cdots, s_N\}$，已知各符號的發生機率爲$p_1, p_2, \cdots, p_N$，各別表現編碼的字碼的編碼長爲$l_1, l_2, \cdots, l_N$，此時平均編碼長度$L$如同先前式(3.6)所示，關於此點，如同以下所示之著名的 Shannon 資訊來源編碼定理而爲人所知[36]。

註　36. 在此為簡化討論，僅處理非記憶資訊來源。

定理 3(Shannon 資訊來源編碼定理)

平均編碼長度L的下限爲：

$$H(S)=\sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i} \quad \text{(每一符號的位元長度)} \tag{3.8}$$

(其中，$H(S) \le L$)。另外，存在編碼，使得平均編碼長度L趨近$H(S)$。

在此，$H(S)$稱爲資訊來源的熵，可以解釋爲自資訊來源所得的平均資料量。

爲證明此定理，首先證明此引理 1。

引理 1

符號s_1，s_2，⋯，s_N，可在其平均編碼長度L滿足下式的情況下：

$$H(S)\ leq\ L < H(S) + 1 \tag{3.9}$$

加以編碼。 less equal

引理 1 的證明

首先，對於符號s_i ($i = 1, \cdots, N$)，當編碼長度l_i滿足以下式子時，可編碼。

$$\log_2 \frac{1}{p_i} \le l_i \tag{3.10}$$

將式(3.10)寫成：

$$2^{-l_i} \le p_i \tag{3.11}$$

將等式兩邊由$i = 1$加到N後，得到

$$\sum_{i=1}^{N} 2^{-l_i} \le \sum_{i=1}^{N} p_i = 1 \tag{3.12}$$

因此該編碼滿足 Kraft-McMillan 不等式，由定理 2 得知滿足式(3.10)後，必定存在編碼方式，滿足一致與瞬間解碼。另外，因爲必定存在一整數l_i，滿足下列式子

$$\log_2 \frac{1}{p_i} \le l_i < \log_2 \frac{1}{p_i} + 1 \tag{3.13}$$

由 Kraft-McMillan 不等式，以l_i爲編碼長度可以構成編碼。

因此，在式(3.13)的兩側都乘上p_i，自$i = 1$加到N後，可以得到

$$\sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i} \le \sum_{i=1}^{N} p_i l_i < \sum_{i=1}^{N} p_i \left(\log_2 \frac{1}{p_i} + 1 \right) \tag{3.14}$$

因此可得到式(3.9)。 (證明完畢)

接著，考慮「產生的n個符號編組後再加以編碼」之處理[37]。在此情況中，總共有N^n種的符號存在，該集合以S^n加以表示，S^n爲S的元素s_1，s_2，⋯，s_N以可重複的方式排了n個，將此記爲s_{i1}，s_{i2}，⋯，s_{in}。另外，在將S^n編碼的情況中，平均編碼長度記爲L_n。

註 37. 這種操作稱為資訊來源的擴張。

因爲是非記憶性的，假設原先符號的產生爲相互獨立的，n個編組後的符號s_{i1}，s_{i2}，⋯，s_{in}發生的機率，可記爲p_{i1}，p_{i2}，⋯，p_{in}。因此，對於S^n，計算式子(3.8)的熵，可得到：

$$\begin{aligned}H(S^n) &= \sum_{\substack{(i_1 i_2 \cdots i_n)\\ \text{的所有組合}}} p_{i1}p_{i2}\cdots p_{in}\log_2\frac{1}{p_{i1}p_{i2}\cdots p_{in}}\\ &= \sum_{\substack{(i_1 i_2 \cdots i_n)\\ \text{的所有組合}}} p_{i1}p_{i2}\cdots p_{in}\left(\log_2\frac{1}{p_{i1}}+\cdots+\log_2\frac{1}{p_{in}}\right)\\ &= \sum_{i1} p_{i1}\log_2\frac{1}{p_{i1}}\left(\sum_{\substack{(i_2\cdots i_n)\\ \text{的所有組合}}} p_{i2}\cdots p_{in}\right)+\cdots\\ &\quad+\sum_{in} p_{in}\log_2\frac{1}{p_{in}}\left(\sum_{\substack{(i_1\cdots i_n-1)\\ \text{的所有組合}}} p_{i1}\cdots p_{in-1}\right)=nH(S)\end{aligned} \tag{3.15}$$

可利用這結果證明定理 3。

定理 3 的證明

將引理 1 代入上述的結果，對於以n個編組的符號，滿足下列式子：

$$nH(S)\le L_n < nH(S)+1 \tag{3.16}$$

可構成滿足一致與瞬間解碼的編碼。這種編碼的每一編碼的平均編碼長度$\overline{L}=L_n/n$爲：

$$H(S)\le \overline{L} < H(S)+\frac{1}{n} \tag{3.17}$$

式子右邊的n**趨近於無限大之後，會趨近**$H(S)$，即可證明定理三。

(證明完畢)

定理三所說的就是「資訊來源的熵可以用非常接近平均編碼長的方式加以編碼，反過來說，就**是無法在平均編碼長小於熵的情形下進行(可逆)編碼**」。因此稱爲熵編碼。

Huffman 編碼

Huffman 編碼[26]之組成

熵編碼方法中最著名者爲 **Huffman 編碼**。現在，考慮編碼對象係由N種符號所組成集合$S=\{s_1, s_2, \cdots, s_N\}$，已知各別元素的發生機率爲$p_1, p_2, \cdots, p_N$，此時，Huffman 編碼由下列的順序所構成。而(　　)內所表示的爲圖 3.5 的實例。

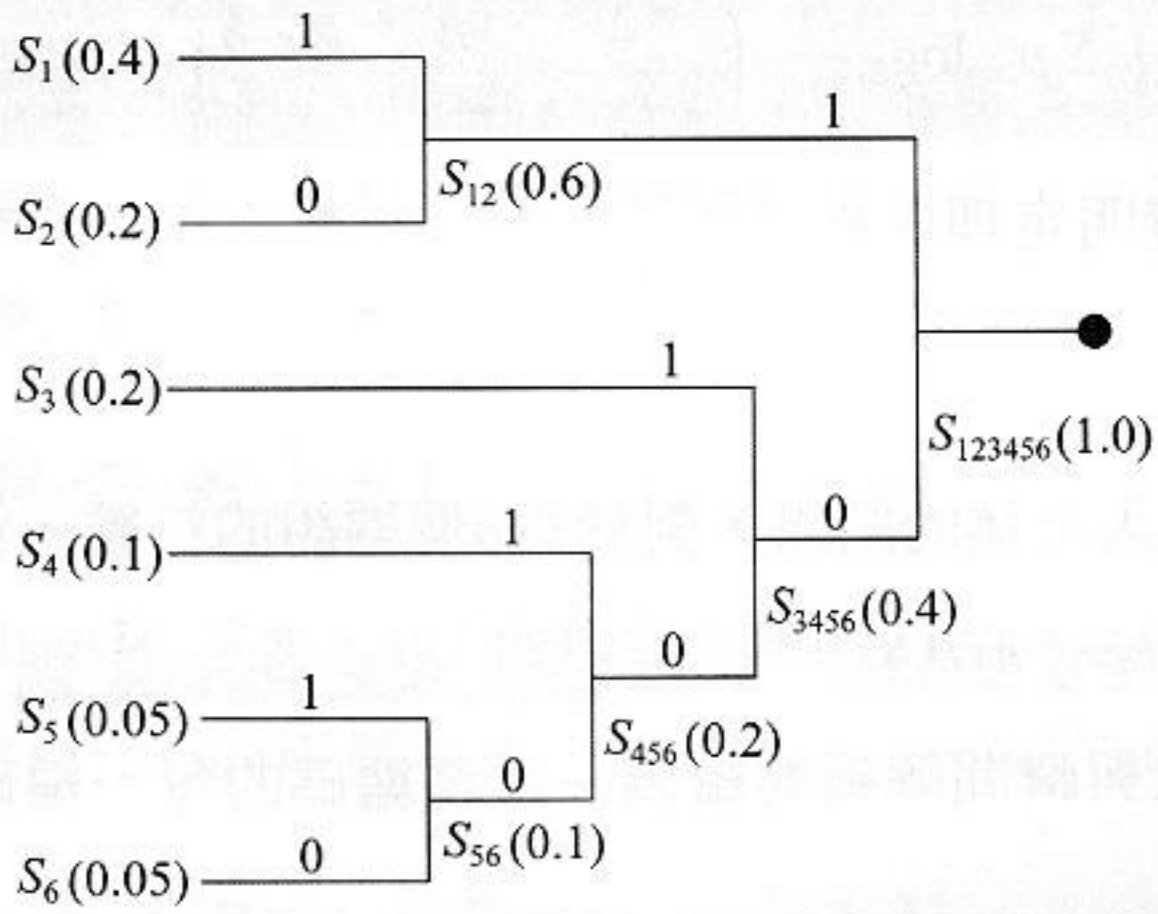


圖 3.5　Huffman 編碼之一例(( )內為符號的發生機率)

① 選擇發生機率較小的兩個符號(s_5 , s_6)，將其合併後成爲一個符號(s_{56})[38]該處理稱爲**符號的縮減**，合併後新符號的發生機率爲兩個符號的相加(0.1)。

② 對於在步驟①所選的兩個符號以編碼 1 與 0 加以表示(何者用 1 都可以)。

註 38. 當同時存在複數個發生機率相同的符號時，選擇適當的即可。

表 3.2　圖 3.5 的 Huffman 編碼

符號	發生機率	Huffman 編碼	編碼長度
s_1	0.4	11	2
s_2	0.2	10	2
s_3	0.2	01	2
s_4	0.1	001	3
s_5	0.05	0001	4
s_6	0.05	0000	4

③ 當全部的符號可合併爲一個符號則進入④，爲否的話則重覆①與②的處理。

④ 最後，可利用所得到圖 3.5 的編碼樹(在此情形下稱爲 **Huffman 樹**)，而得到各符號。

表 3.2 所表示爲藉由圖 3.5 的Huffman樹所得的Huffman編碼(稱爲**Huffman表**)。發生機率較高的編碼長度確實較短，其平均編碼長度$L=2.3$，另外，該資訊來源的熵值經計算後爲$H=2.2$，可以知道 Huffman 編碼具有與熵值接近的平均符號長度。實際上，Huffman編碼是在所有編碼中，平均編碼長度可爲最短的編碼方法[23]。

可變長度編碼的解碼與Huffman編碼的問題點

在本節中討論由包含Huffman編碼的可變長度編碼串解碼原符號的方法與其問題點。

可變長度編碼的解碼方法大致上可分成以下兩類。

- 位元比對法
- 表對照法

前者將輸入的位元串依每一位元加以讀取，對應其「1」、「0」的排列如圖 3.5 所示由編碼樹的根至葉前進，到達葉之後就輸出該符號，而後回到根再進行相同的處理。以這種方法處理雖然較爲單純，但是另外一方面會有在解碼時需要的運算量依其編碼長度而有所不同的問題，另外，因爲需以位元爲單位加以運算，因此效率較差。另一方面，亦有依下列順序進行解碼者。

準備(製作解碼表)

① 調查所有的字碼中最長的編碼長度，稱該編碼長度爲$l_{\max}$。

② 符號s_i的字碼w_i(編碼長度爲l_i)，在$l_{\max}$位元中靠左排列，剩下的$l_{\max}-l_i$位元再賦予 1 與 0 所有的組合。因此對於編碼長度爲l_i的符號s_i，可以得到$2^{l_{\max}-l_i}$個二進位編碼，而作出關於符號s_i的**解碼表**。

③ 對於所有符號反覆進行步驟②，最後會完成具有$2^{l_{\max}}$項的解碼表。

(例) 表 3.3 爲對於表 3.2 的Huffman編碼製作的解碼表。(在此情況中，$l_{\max}$爲 4)

解碼處理

① 準備$l_{\max}$位元長的緩衝區B，自解碼對象的位元串讀取$l_{\max}$位元後，存至B。

表 3.3 圖 3.5 的 Huffman 編碼之解碼表

二進位	s_i	編碼長度	二進位	s_i	編碼長度
1100	s_1	2	0100	s_3	2
1101	s_1	2	0101	s_3	2
1110	s_1	2	0110	s_3	2
1111	s_1	2	0111	s_3	2
1000	s_2	2	0010	s_4	3
1001	s_2	2	0011	s_4	3
1010	s_2	2	0001	s_5	4
1011	s_2	2	0000	s_6	4

② 在解碼表中找出與B的位元形態(值)對應的項之s_k與l_k，將s_k以解碼符號加以輸出。

③ 僅將B平移l_k位元，同時自位元串右方讀取新的l_k位元後寫入B的右側，重覆進行步驟②。

(例) 使用表 3.2 的 Huffman 編碼的符號列$s_1s_4s_3s_2s_6$加以編碼之後的位元串$(1100101100000)_2$以表 3.3 的解碼表加以解碼，如圖 3.6。

使用表對照法可以僅參照解碼表加以解碼，雖然解決了位元比對法的兩個問題，另外，但是另一方面，當最大編碼長度$l_{\max}$舉例來說增加至 20～30 位元後，解碼表所需的記憶體容量就會變得非常大。特別是當 Huffman 編碼中的符號總數N較大，每個的發生機率都非常小的符號都存在的情況中，最大編碼長度$l_{\max}$會顯著地增大。實際上需要有特別的對策。

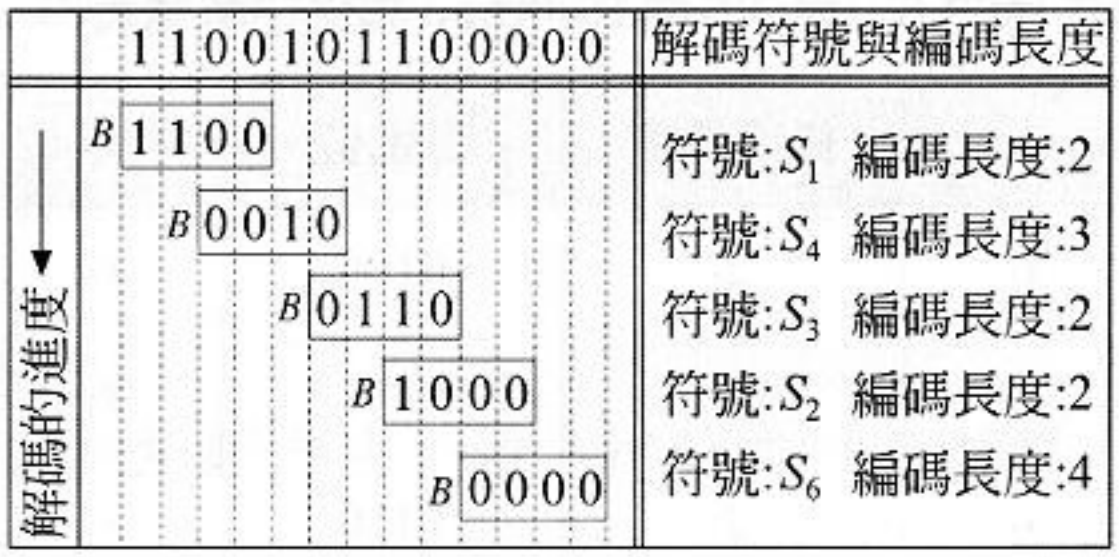


圖 3.6 使用解碼表 3.3 的解碼例

Huffman 編碼中編碼長度的限制

在此，對於Huffman編碼演算法中，限制l_{max}的一種觀念加以介紹。演算法非常簡單的以下列方式表示：

① 將符號集合$\{s_1, s_2, \cdots, s_N\}$分成數群，賦予各群編號。現在，有$M$群，第$m$ ($m=1, 2, \cdots, M$)群包含n_m個原符號。

② 爲了要辨別第m ($m=1, 2, \cdots, M$)群內的原符號s_i，賦予各原符號固定長度的二進位編碼。所需編碼長度爲$\log_2 n_m$(位元)。

③ 將M個群的編號$m=1, 2, \cdots, M$視爲新的符號，對此進行Huffman編碼。

④ 對原符號s_i編碼時，首先輸出對應於s_i所屬的群編號m之Huffman編碼，接著輸出②之固定長度二進位編碼，即爲s_i的編碼。

在符號總數爲N時，將總群數M抑制在最小範圍，藉此限制Huffman編碼的最大編碼長度l_{max}，製作 Huffman 樹，同時也易於解碼。另外，用以辨別各群內的原符號的固定長度編碼所占的比重增加，使得平均編碼長度免於惡化(增長)。這種編碼方法實際運用於 JPEG 或是 MPEG 等標準中。細節於第九章、第十章中介紹。

算數編碼

爲了儘可能地減少 Huffman 編碼中的平均長度L，必須儘可能地對最多的原符號一起進行編碼。但是，如此一來作爲編碼對象的符號數量就會顯著地增加，先前所述的在解碼時的問題便會益發明顯，同時在製作 Huffman 樹所需要的計算量亦無法加以忽視。接下來介紹的算數編碼，就是在 Huffman 編碼中解決此種問題的編碼方式。

算數編碼係依照符號發生機率切割數次數線，最後分割的部份的位置以二進位加以表現，藉此賦予編碼的方法。但是，要詳細理解有必要對其細部加以說明，在此僅說明其基礎的 Elias 編碼，至足以理解算術編碼的原理爲止。

現在，作爲編碼對象的符號列由「0」、「1」所組成，已知「0」、「1」的發生機率分別爲p_0、p_1($p_0>p_1$)。圖 3.7 表示在符號列爲「011000」的情況中，Elias編碼的順序。首先將縱軸依p_0、p_1比例分割，所得區間分別稱爲A_0、A_1。縱軸爲由輸入「0」或「1」選擇p_0、p_1的對應部份。此處在最初輸入值爲「0」的情況中，選擇A_0。接著，再依選擇區間的p_0、p_1比例切出區間A_{00}、A_{01}。當輸入值爲「1」時，就選擇區間A_{01}，之後依相同方式逐次選擇區間，最後如圖 3.7，會選擇區間A_f。

考慮該區間A_f在垂直方向的位置以二進位表示。符號的整個位元長度爲N，其中「0」的個數爲N_0，「1」的個數爲N_1，最後所選擇的區間在垂直方向的長度可表示爲$p_0^{N_0}p_1^{N_1}$(圖 3.7 的例子爲$p_0^4p_1^2$)。該區間內的一點以二進位表示所需的最低精度，約爲：

$$(-N_0 \log_2 p_0 - N_1 \log_2 p_i) \quad [\text{位元}] \tag{3.18}$$

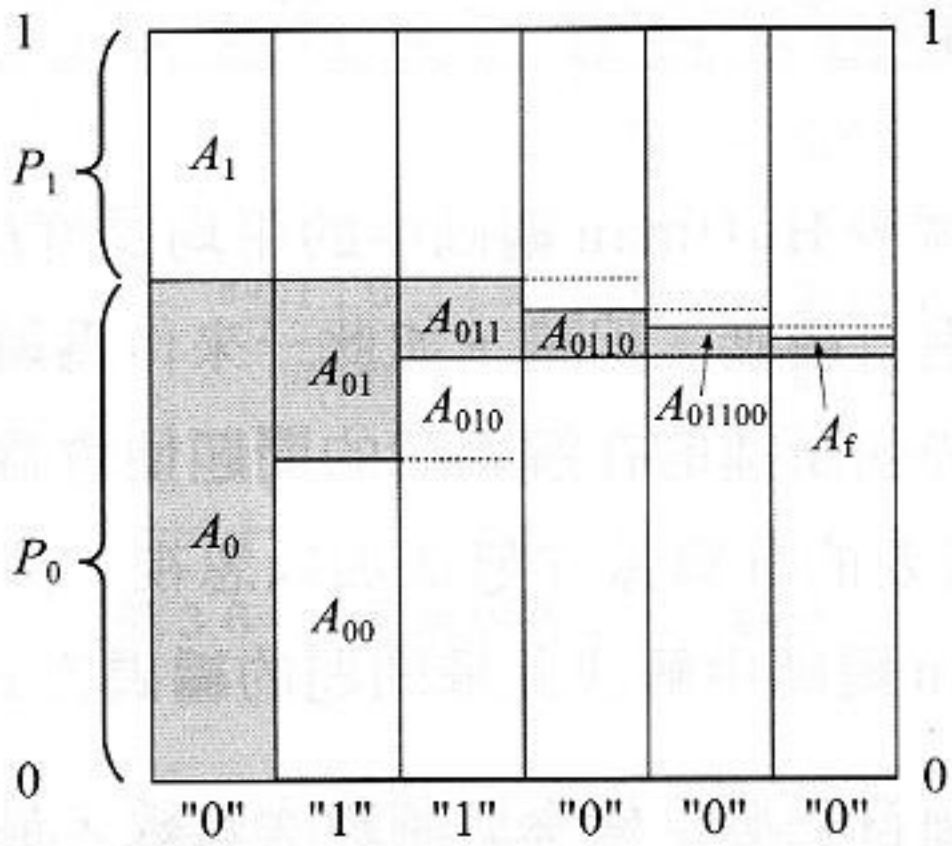


圖 3.7 Elias 符號的例子

編碼器輸入的符號長度假設爲N，每一符號的平均編碼長度L可表示爲：

$$L \simeq \frac{-N_0 \log_2 p_0 - N_1 \log_2 p_1}{N} \quad [\text{位元}/] \tag{3.19}$$

當N相當大時，

$$p_0 \simeq \frac{N_0}{N}, \quad p_1 \simeq \frac{N_1}{N} \tag{3.20}$$

代入式(3.19)後，可得

$$L \simeq p_0 \log_2 \frac{1}{p_0} + p_1 \log_2 \frac{1}{p_1} = H(S) \tag{3.21}$$

亦即，爲可達成 Shannon 理論的極限值之編碼方式。

一般的的符號發生機率爲實數，但是爲了進行 Elias 編碼，需要反覆進行實數的乘法計算。但是，爲此需要無限的計算精度，所以並不實用。爲此，有人提出將p_0、p_1中較小者以2^{-Q}加以近似的實用編碼演算法，稱爲**L-R 算數編碼**，Q稱爲傾斜值，在適切地模式化處理對象之資訊來源時，依據其統計性質可以適當地改變Q值，亦可良好地對於記憶

性的、或是不定的資訊來源進行編碼。這種方法，稱爲標準化 **QM 碼 (QM CODER)**。關於實用的算數編碼或是 QM 碼的詳情，請參考文獻 [27]～[29]。

4章

量子化—進行統計處理

量子化的基本觀念，如先前第一章第七頁所示，另外，在第一章中，對於在影像編碼演算法中的量子化，在下列順序中：

說明對於除去像素間相關程度後的轉換係數進行量子化。

量子化器在其性質上，在輸入輸出間會伴隨量子化誤差，為了分析其影響，需對量子化誤差e進行統計的處理。在本章中加以討論，推導關於e的離散程度與量子化後位元長度的重要關係。

前 言

在第一章中，將量子化定義爲「將輸入值對應於預先定義的離散集合中之適當元素」。現在，令輸入量子化器的輸入值爲x，在離散集合

中的元素爲y_0，y_1，y_2，⋯，y_{L-1} $(y_0<y_1<y_2<\cdots<y_{L-1})$[39]。此處，假設元素的總數$L$(量子化等級)爲有限的[40]。

註 39. 為了說明方便，使用與第一章不同的變數。
40. 該理由如第一章第八、九頁所述。

量子化器對於輸入值x，輸出滿足：

$$\min_n |x-y_n| \tag{4.1}$$

的量子化指標n $(n=Q[x])$。另外，對**反量子化器**輸入n會輸出重現值[41] $y_n(y_n=Q^{-1}[n])$。在此，當$y_n \neq x$，輸入量子化器與反量子化器輸出值之間會產生誤差$e=x-Q^{-1}[Q[x]]$。此稱爲量子化誤差或是量子化雜訊，在本章中，爲了簡化寫法，遵守下列規定。因量子化指標 n 與重現值 y_n爲一對一對應，將量子化器Q視爲不是輸出 n 而是直接輸出 y_n，寫成 $y_n=Q[x]$。如此一來，量子化誤差可簡單地表示爲：

$$e=x-Q[x]=x-y_n \tag{4.2}$$

註 41. 也有量子化代表值、輸出等級等稱呼。

對於對應於某重現值y_n之輸入值x的區間以$[x_n^{\min}, x_n^{\max}]$表示。此時如前述，$\Delta_n=x_n^{\max}-x_n^{\min}$稱爲量子化步進幅度。反過來說，當量子化步進幅度爲$\Delta_n=[x_n^{\max}, x_n^{\min}]$的情形中，令其重現值爲$y_n$，在此區間內量子化誤差的平方平均$E[e^2]$可寫爲：

$$E[e^2]=\int_{x_n^{\min}}^{x_n^{\max}} (x-y_n)^2 p_n(x)dx \tag{4.3}$$

在此，$p_n(x)$爲區間$[x_n^{\min}, x_n^{\max}]$中，x的發生機率。爲了達成儘量減少量子化誤差的目的，求取使$E[e^2]$爲最小的重現值y_n。爲此將式(4.3)對y_n進行微分後取其值爲 0，而可以得到以下結果：

$$\frac{\partial E[e^2]}{\partial y_n} = -\int_{x_n^{\max}}^{x_n^{\max}} 2(x-y_n)p_n(x)dx = 0 \tag{4.4}$$

$$y_n = \int_{x_n^{\max}}^{x_n^{\max}} xp_n(x)dx \tag{4.5}$$

結果爲「對於重現值y_n取對應區間內發生機率分佈函數的重心，會使得量子化誤差e爲最小」。接下來，以滿足式(4.5)取重現值y_n。

各種量子化器

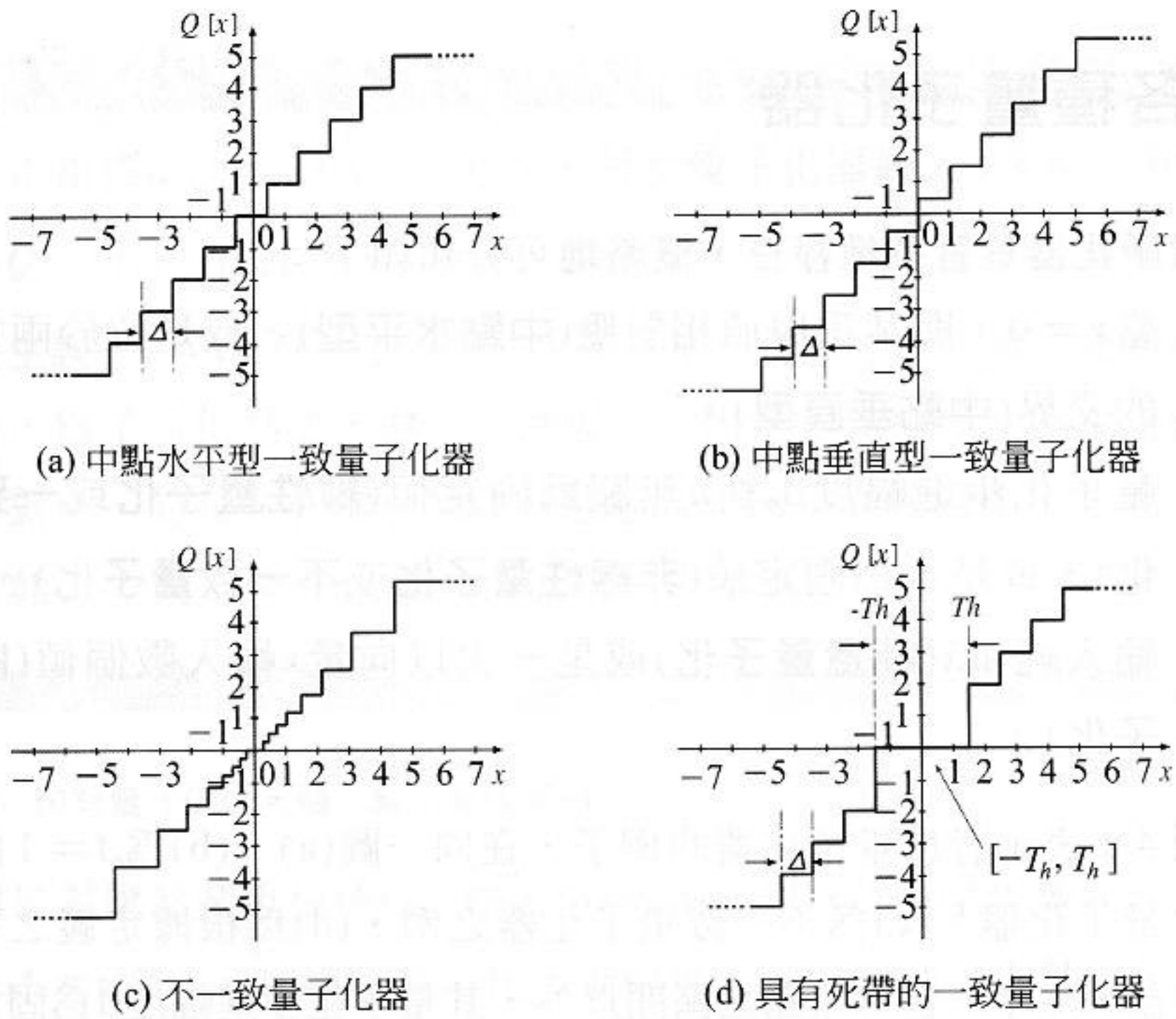
量子化器具有各種存在，概略地可分類如下：

- 當$x=0$，與某重現值相對應(**中點水平型**)，或是位於兩重現值的交界(**中點垂直型**)。
- 量子化步進幅度Δ_n與n無關爲固定值(**線性量子化**或**一致量子化**)，或是不爲固定值(**非線性量子化**或**不一致量子化**)。
- 輸入純量x(**純量量子化**)或是一次以向量x輸入數個值(**向量量子化**)。

圖 4.1 表示者爲量子化器的例子，在同一圖(a)、(b)爲$\Delta=1$的情況之一致量子化器，(c)爲不一致量子化器之例，(d)爲根據定義之不一致量子化器，除了一包含原點之區間以外，其量子化步進幅度Δ爲固定的。這種量子化器，特別稱爲**具有死帶的一致量子化器**，用於 MPEG 中的 DCT 係數。(譯按：死帶亦譯作最小靈敏區)

非線性量子化器需要適當地訂定各重現值，因而設計較複雜，另外，在輸入訊號的統計性質(特別是輸入值x與發生機率密度函數 $p_x(x)$)爲已知的情況中，可以實現量子化誤差之平均平方爲最小的**最佳量子化器**[30]。進行最佳量子化時，量子化指標n的發生機率爲$p_x(x)$，爲扁平的

性質，爲此，在如圖 2.14 一般，即使在後段設有熵編碼器，也無法對n再加以壓縮[30]。也就是說，在進行最佳量子化時，具有毋需在後段設置熵編碼器的優點。但是，當不知道其統計性質時該項優點便不存在，此時「一致量子化器＋熵編碼器」之組成較爲有利。爲此，雖然最佳量子化器在理論上相當有趣，但是鮮少用於實際應用。



(a) 中點水平型一致量子化器　(b) 中點垂直型一致量子化器

(c) 不一致量子化器　(d) 具有死帶的一致量子化器

圖 4.1　量子化器的輸入輸出特徵

另外，雖然藉由向量量子化進行影像編碼的處理方法也廣泛地被使用，但是本書僅討論純量量子化，對此有興趣的讀者請參照文獻[31]。

量子化誤差之統計處理

粒狀誤差與溢位誤差

圖 4.2 爲對圖 4.1(a)的一致量子化器輸入波形$x(t)$的情況中，輸出的波形$Q[x(t)]$。在此，以 表示量子化器所容許的最大輸入範圍[42]。進行量子化時在輸出輸入間會產生由式(4.2)所定義的量子化誤差e，配合輸入$x(t)$之大小，可大致將誤差分爲兩類：

- 位於$|x(t)| \geq x_{\max}$範圍 A、C 內的誤差。
- 位於$|x(t)| \leq x_{\max}$範圍 B 內的誤差。

註　42.

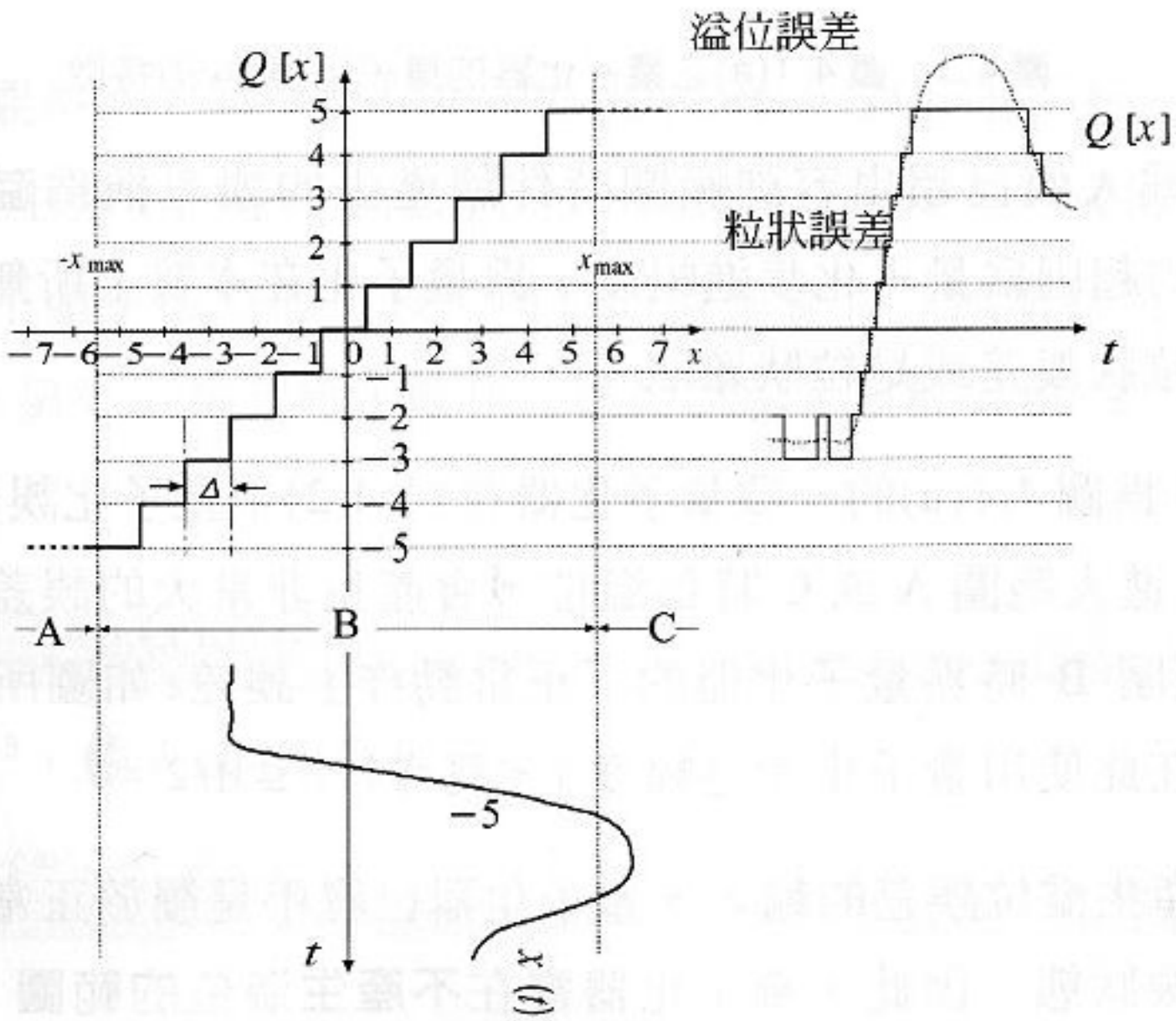


圖 4.2　圖 4.1(a)之量子化器對於波形$x(t)$的響應

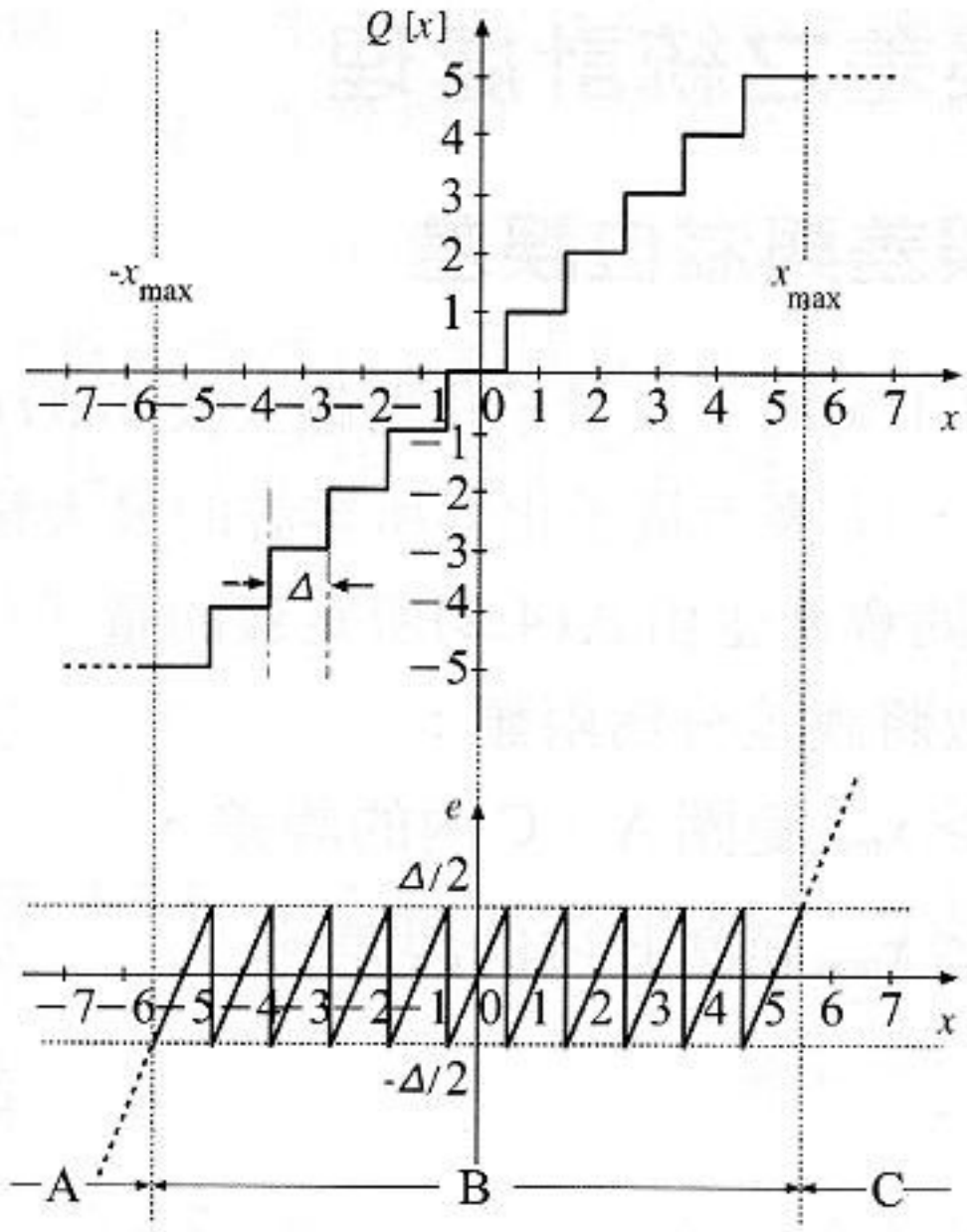


圖 4.3　圖 4.1(a)之量子化器的量子化誤差e的特徵

如前者般輸入值已超出容許範圍時伴隨產生的誤差稱爲**溢位誤差**。另外，後者的起因爲量子化步進幅度，爲量子化在本質上所無法避免的誤差。稱爲粒狀誤差或是粒狀雜訊。

圖 4.3 爲圖 4.1(a)的一致量子化器在式(4.2)中量子化誤差e的計算結果。輸入x進入範圍 A 或 C 時爲溢位，會產生非常大的誤差。另外，輸入x進入範圍 B 時爲量子化器的「正常動作」誤差e如圖所示呈現鋸齒狀。此時在此使用量子化步進幅度Δ，寫作$|e| \le \Delta/2$。

對於產生溢位誤差的輸入，量子化器已經不是屬於正常動作，可說是處於危險狀態。因此，量子化器需在**不產生溢位的範圍 B 內加以使用**，或是輸入的所有值x必須**加以設計，使得溢位不致產生**。以後的討論，係假設滿足這些條件，**完全不會產生溢位**。此時，所產生的量子化誤差僅爲粒狀誤差。

量子化誤差之統計性質

在此，對於量子化誤差，導出以下這個非常重要的關係式。

量子化誤差的離散程度

$=$常數$\times 2^{-(2\times\text{量子化位元長度})}\times$輸入訊號的離散程度

爲此，依以下順序進行討論：

① 計算量子化器的輸入值x的離散程度σ_x^2。

② 計算量子化誤差e的離散程度σ_e^2。

③ 量子化器的容許最大輸入值$x_{\max}$與σ_x^2的關係。

量子化器的輸入值x的離散程度σ_x^2

現在，觀察對於量子化器的許多輸入值x，對其進行統計的結果，會得到x的發生機率的密度函數 $p_x(x)$。如圖 4.4 所示，考慮的 $p_x(x)$各種分佈形狀由不產生溢位的假設，所有的機率會有共通點就是 $p_x(x)=0$ $(|x|>x_{\max})$。另外，在此爲了簡化之後的討論，將輸入值x的平均值μ_x假設爲[43]：

$$\mu_x=\int_{-\infty}^{\infty}xp_x(x)dx=0 \tag{4.6}$$

在此假設下[43]，輸入x的離散程度σ_x^2可寫爲：

$$\sigma_x^2=\int_{-\infty}^{\infty}x^2p_x(x)dx \tag{4.7}$$

註　43. 在$\mu_x\neq 0$的情況中，可以將x代換為μ_x-x，以滿足此條件。

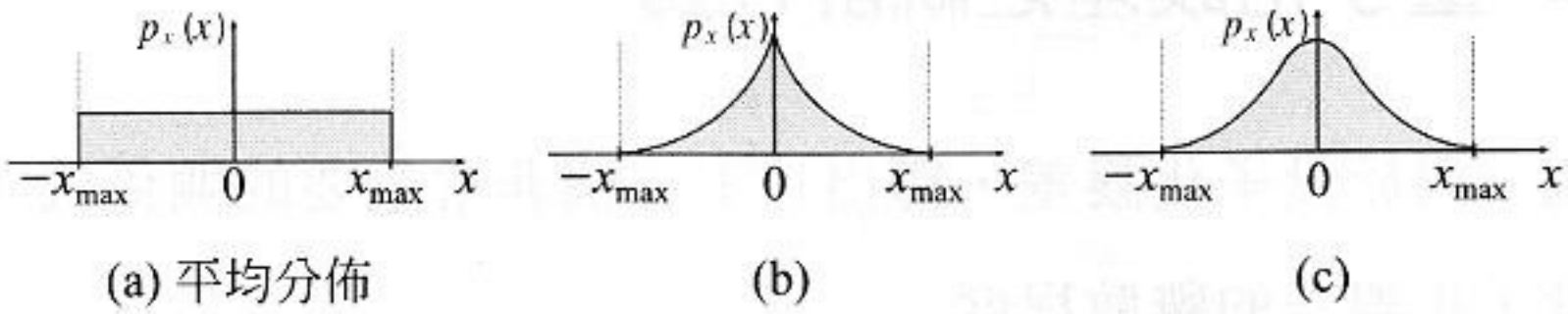


圖 4.4　量子化器的輸入值 x 的發生機率密度函數$p_x(x)$之例

量子化誤差e的離散程度σ_e^2

另外，量子化誤差e的離散程度σ_e^2可用下列式子計算：

$$\sigma_e^2 = E[(e-\mu_e)^2] = \int_{-\infty}^{\infty}(e-\mu_e)^2 p_e(e)de \tag{4.8}$$

其中，μ_e爲e的平均值，$p_e(e)$爲量子化誤差e的發生機率密度函數。

$p_e(e)$爲對於夠多的輸入值x，進行觀察後的結果，但是實際上會有各種分佈。當假設量子化等級L大到某種程度，對多數x計算誤差e的統計值，在之間所產生誤差僅觀察得到特定值是不自然的，較自然的情形爲在此區間內e的發生機率是平均產生的。如此考慮之後，假設$p_e(e)$的分佈爲如下的平均分佈(圖 4.5)：

$$p_e(e) = \begin{cases} 1/\Delta & (-\Delta/2 \le e \le \Delta/2) \\ 0 & (e < -\Delta/2，\Delta/2 < e) \end{cases} \tag{4.9}$$

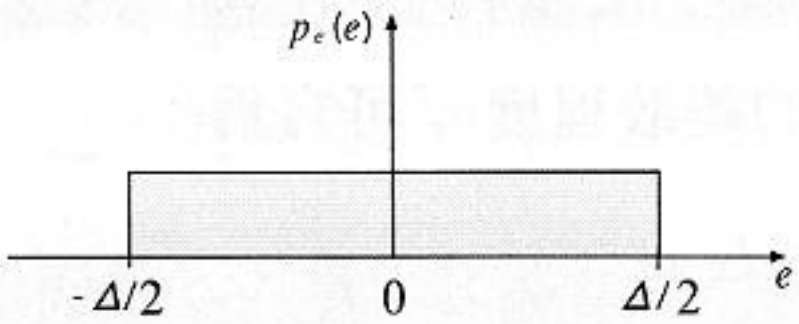


圖 4.5　量子化誤差e的發生機率密度函數$p_e(e)$

實際上，在式(4.9)的假設中，需要限制在滿足以下條件：

- 量子化器的量子化等級L需要大到某種程度。

- 輸入值x的離散程度要大到某個程度。
- 輸入值x需爲非周期性的。

才能確保其效果良好。以簡單的實驗驗證式(4.9)的強健性，現在對於界限爲$x_{max}=1$的量子化器Q輸入兩種輸入波形：

$$x_1(t)=\cos 2\pi t \tag{4.10}$$

$$x_2(t)=\frac{1}{2}(\sin \pi t+\cos 2\sqrt{2}\pi t) \tag{4.11}$$

調查其量子化誤差$e=x_i(t)-Q[x_i(t)]$的發生機率。如圖 4.6(a)、(b)所示，$x_1(t)$爲週期波形，$x_2(t)$爲非週期波形，兩者都在溢位界限內振盪。對此，以量子化器的量子化等級L爲 1024(10 位元長)、256(8 位元長)、64(6 位元長)、16(4 位元長)測量其量子化誤差e的發生機率標示於同一圖的(c)～(j)。另外，縱軸的發生機率，爲了要易於比較，以式(4.9)的$1/\Delta$加以正規化。

除了(g)或(i)當輸入爲週期性且量子化等級較小的情況中，式(4.9)的的假設是足以成立的。特別是，如$x_2(t)$的非週期波，即使在位元長度爲 4 的情形下進行量子化，該假設也具有某種程度的強健性。

接著，以式(4.9)的$p_e(e)$計算e的離散程度。首先，由於式(4.9)的$p_e(e)$對稱於$e=0$，e的平均值爲$\mu_e=0$。將此結果代入式(4.8)可計算：

$$\begin{aligned}\sigma_e^2&=E[e^2]=\int_{-\infty}^{\infty}e^2p_e(e)de\\&=\int_{-\Delta/2}^{\Delta/2}e^2\frac{1}{\Delta}de=\frac{1}{12}\Delta^2\end{aligned} \tag{4.12}$$

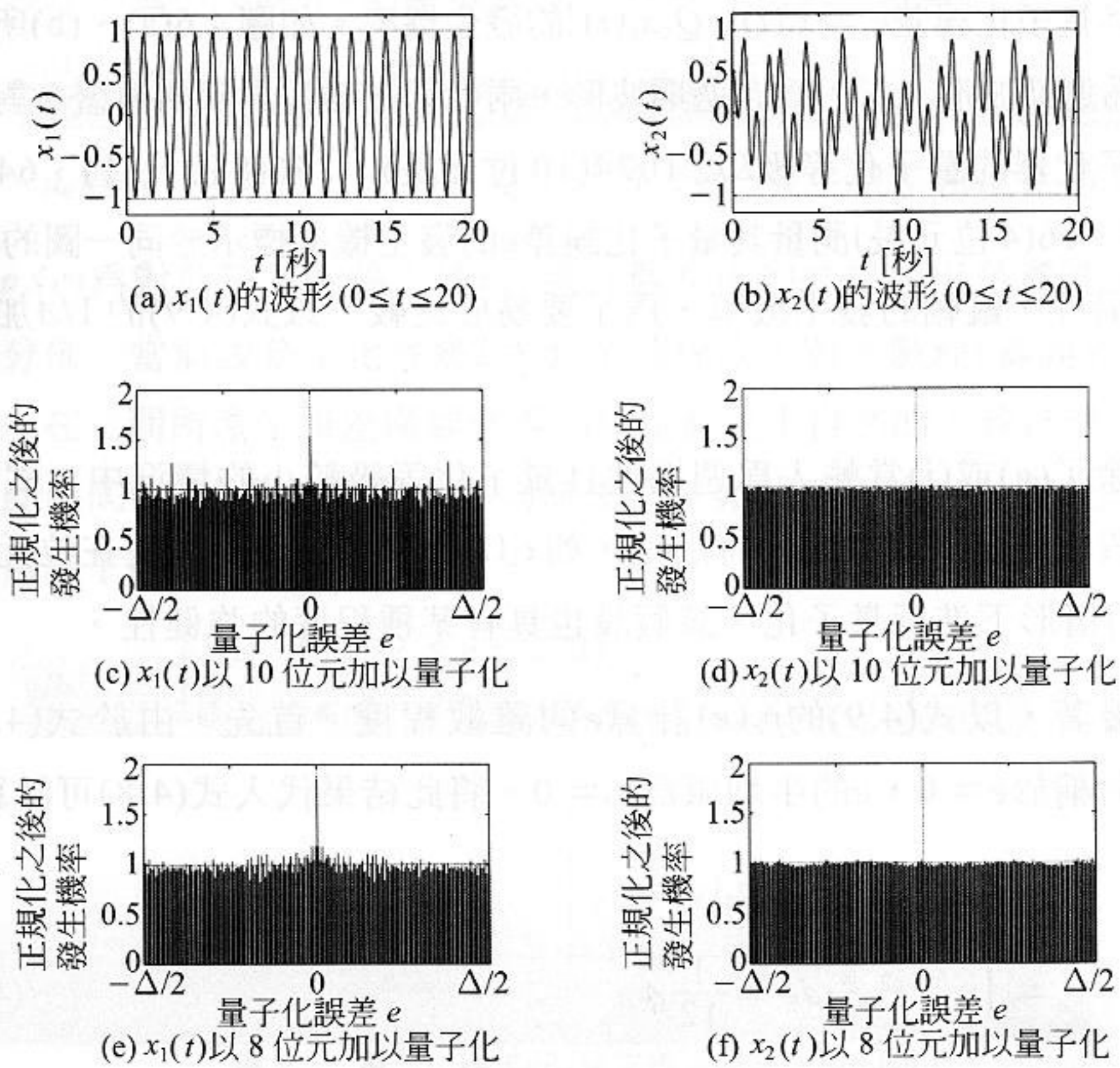
其中，假設量子化器的位元長度爲R(位元)，，量子化等級爲$L=2^R$，參照圖 4.3，可寫作[44]：

$$\Delta = \frac{2x_{\max}}{2^R} = 2x_{\max}2^{-R} \tag{4.13}$$

代入式(4.12)後，可得：

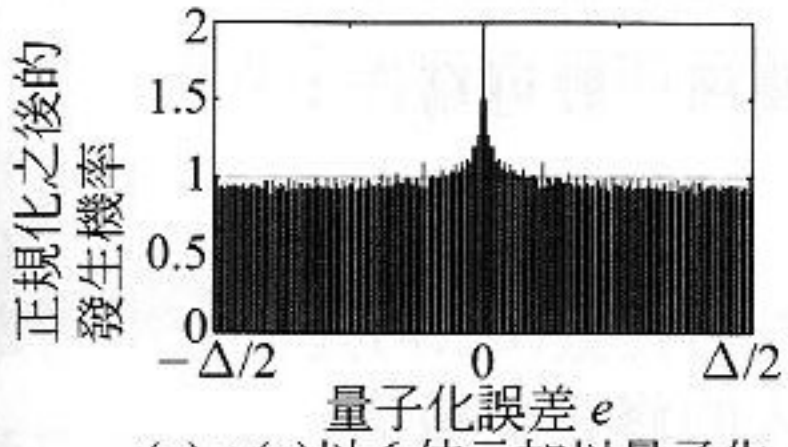
$$\sigma_e^2 = \frac{1}{3}x_{\max}^2 2^{-2R} \tag{4.14}$$

註 44. 在中點水平的情形中，量子化等級的數目需為奇數才能使式(4.13)嚴格地成立，但是只要$R \geq 4$就可以忽略其誤差。

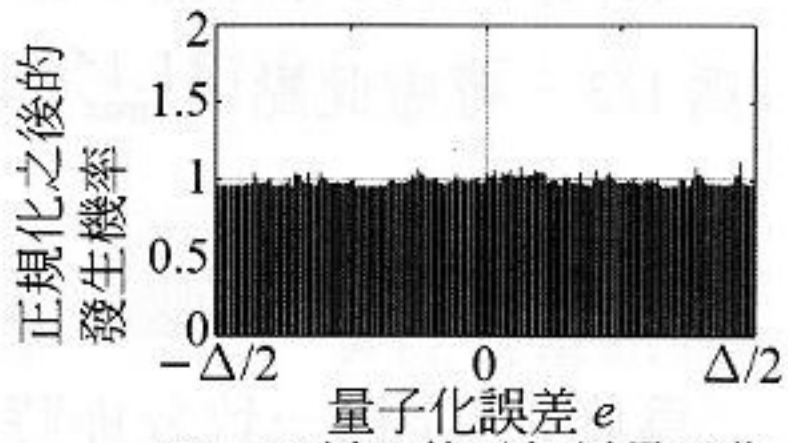


(a) $x_1(t)$的波形 $(0 \leq t \leq 20)$
(b) $x_2(t)$的波形 $(0 \leq t \leq 20)$
(c) $x_1(t)$以 10 位元加以量子化
(d) $x_2(t)$以 10 位元加以量子化
(e) $x_1(t)$以 8 位元加以量子化
(f) $x_2(t)$以 8 位元加以量子化

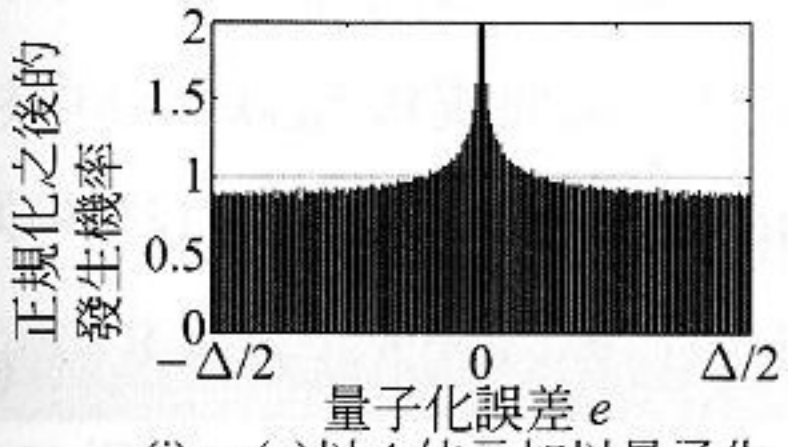
圖 4.6 量子化器的輸入波形與對應的量子化誤差e的發生機率



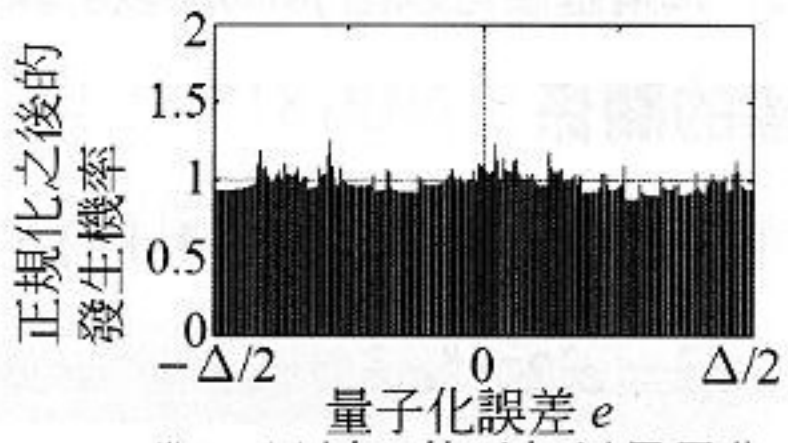
(g) $x_1(t)$以 6 位元加以量子化



(h) $x_2(t)$以 6 位元加以量子化



(i) $x_1(t)$以 4 位元加以量子化



(j) $x_2(t)$以 4 位元加以量子化

圖 4.6　量子化器的輸入波形與對應的量子化誤差e的發生機率(續)

量子化器的容許最大輸入值x_{max}與σ_x^2的關係

最後，將式(4.14)轉換爲簡單的形式，以求取x_{max}與σ_x^2的關係。

現在，先(任意地)假設輸入x的發生機率密度函數具有如圖 4.4(a)的一致分佈，寫成：

$$p_x(x)=\begin{cases}\dfrac{1}{2x_{max}} & (|x|\le x_{max})\\ 0 & (|x|>x_{max})\end{cases} \tag{4.15}$$

，代入式(4.7)後，可寫成：

$$\sigma_x^2=\int_{-x_{max}}^{x_{max}} x^2\frac{1}{2x_{max}}dx=\frac{1}{3}x_{max}^2 \tag{4.16}$$

進一步代入式(4.14)，可得

$$\sigma_e^2=\sigma_x^2 2^{-2R} \tag{4.17}$$

但是，一般的$p_x(x)$的分佈非一致分佈，此影響使得在式(4.16)中，x_{max}^2的係數爲 1/3，考慮此點，x_{max}與σ_x^2的關係一般可寫作：

$$\sigma_x^2 = \frac{1}{3\varepsilon^2} x_{max}^2 \tag{4.18}$$

其中，ε^2爲修正x非爲一致分佈時所導入的修正係數[45]。

註 45. 其精確值可以由 $p_x(x)$加以計算。

與目標的關係

再將式(4.18)代到式(4.14)，可以得到我們想要的關係：

$$\sigma_e^2 = \varepsilon^2 2^{-2R} \sigma_x^2 \tag{4.19}$$

關於式(4.19)

在式(4.19)中，σ_x^2爲量子化器輸入訊號的離散(功率)[46]，左邊的σ_e^2稱爲量子化器產生誤差(雜訊)的功率。一般來訊號功率與雜音功率的比率σ_x^2/σ_e^2稱爲訊號／雜音比(Signal to Noise Ratio：SNR)，通常以其分貝數加以表示：

$$\text{SNR} = 10 \log_{10} \frac{\sigma_x^2}{\sigma_e^2} = 20 \log_{10} \frac{\sigma_x}{\sigma_e} \tag{4.20}$$

將式(4.20)代入式(4.19)後，可得到以下關係：

$$\text{SNR} = 10 \log_{10} \frac{\sigma_x^2}{\sigma_e^2} \simeq 6.02R - 10 \log_{10} \varepsilon^2 \tag{4.21}$$

註 46. 訊號的離散程度可稱作功率或是電能。此為電學的解釋，將訊號x設想為電壓的情形。亦即訊號電壓x的離散程度與電壓的平方σ_x^2因次相同。在電學中係與功率(電能)成比例而加以量化。作者是電學出身的，對於功率或是電能的稱呼感到親切。

右邊第二項爲常數，因此可知量子化位元長增加一位元後，約可改善6dB。第二項表示，該直線的截距，如式(4.18)所示，輸入值x的離散程度爲常數。

舉例來說，式(4.10)及式(4.11)的輸入波形，實際測量訊號功率σ_x^2與雜音功率σ_e^2的結果如圖 4.7 所示。圖中的╳記號爲實際測量點。另外，如圖(4.10)，在量子化器的輸入界限內$[-x_{\max}, x_{\max}]$完全振盪的之正弦波的情形中$\sigma_x^2 = x_{\max}^2/2$(對此請自行確認)，與式(4.18)相較，可得$\varepsilon^2 = 2/3$。代入式(4.21)，可以得到對於正弦波，量子化的位元長度與SNR的關係：

$$\mathrm{SNR} = 6.02R + 1.76 \quad [\mathrm{dB}] \tag{4.22}$$

式(4.22)相當於圖 4.7 的實線，即使在位元長爲 4 位元時也幾乎與實測結果一致。另外，在$x_2(t)$的情形中，由於ε^2不同，所以與正弦波的截距不同，但是仍然可觀察到係以 6dB/位元的斜率向右上方傾斜。

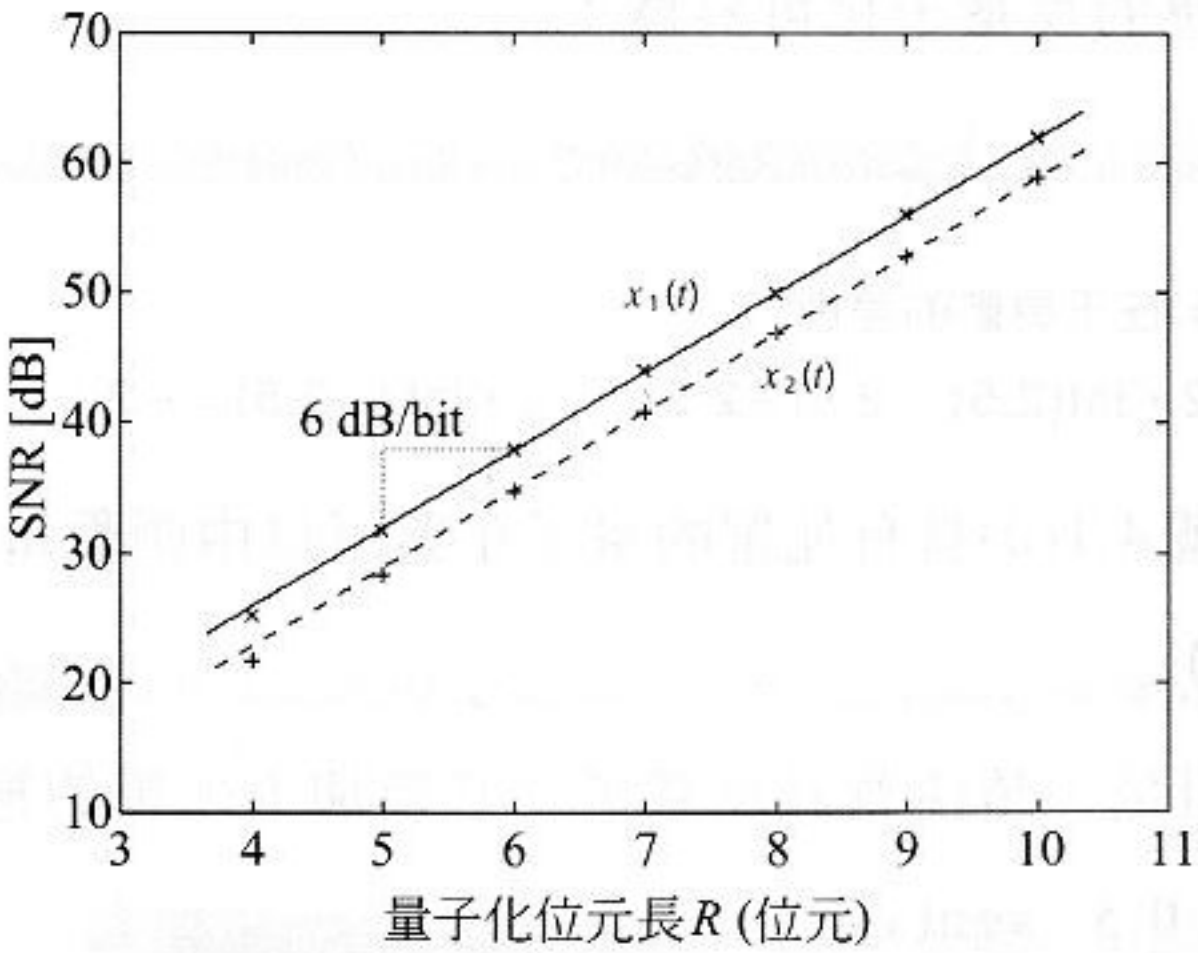


圖 4.7　量子化位元長 R 與 SNR 的關係

結語－量子化與反量子化的實際操作

稍微又開話題，考慮在之後章節實施圖 4.1(a)與(d)中的量子化／反量子化器的方法。

式(1.6)中，量子化步進幅度爲Δ的一致量子化以高斯符號$\lfloor x \rfloor$表示爲：

$$\lfloor x/\Delta + 0.5 \rfloor \tag{4.23}$$

在此使用**捨去小數點之後的函數** int (x)[47]，以下列方式加以實施：

$$\mathrm{int}\,(x/\Delta + 0.5 \cdot \mathrm{sgn}(x)) \tag{4.24}$$

其中，sgn (x)爲符號函數：

$$\mathrm{sgn}(x) = \begin{cases} +1 & (x > 0) \\ 0 & (x = 0) \\ -1 & (x < 0) \end{cases} \tag{4.25}$$

另外，與其對應的反量子化可寫爲：

$$\Delta \cdot x \tag{4.26}$$

註 47. 需注意$\lfloor x \rfloor$在正負數的差別：

$\lfloor 2.5 \rfloor = 2$，$\mathrm{int}(2.5) = 2$，$\lfloor -2.5 \rfloor = -3$，$\mathrm{int}\,(-2.5) = -2$。

另外，在圖 4.1(d)具有死帶的量子化器可以由函數int(x)的特徵以：

$$\mathrm{int}\,(x/\Delta) \tag{4.27}$$

加以實施[48]。但是，請注意其反量子化必須加上死帶的補償項。

$$\Delta \cdot (x + 0.5 \cdot \mathrm{sgn}(x)) \tag{4.28}$$

註 48. 雖然可用$\lfloor x \rfloor$加以表示，但是需使用較 sgn (x)為複雜的函數。

最後，提及以C語言編寫程式。之前所述的高斯符號$\lfloor x \rfloor$對應於C語言的函數 floor( )，牽涉到 int (x)的許多處理可僅用(int)x 加以實施[49]。因此，要將 double x 一致量子化爲 int Qx，舉例來說，可用：

```
Qx=(int) floor(x/delta+0.5);
```

或

```
Qx=(int)(x/delta+(x>0? 0.5:-0.5));
```

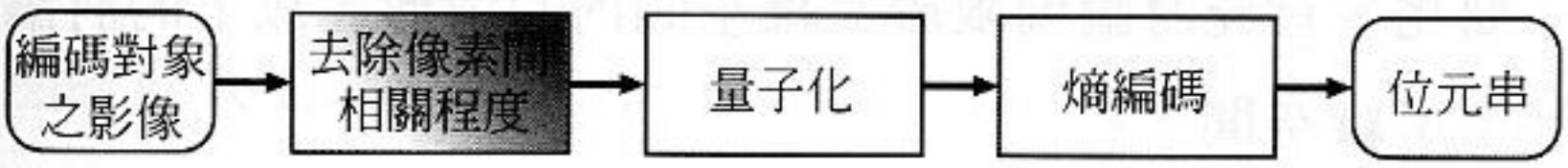
另外，前者在呼叫(call)函數時需用到額外處理的時間(overhead)，所以在實作上後者較爲快速。在具有死帶的情形中，或是在逆量子化的情形中，可用式(4.26)～(4.28)加以編寫程式。

註　49. 前者為 double 型，後者為 int 型的資料指標，因而有所不同。

5 章

正交轉換與 DCT

在圖 2.14 中的去除像素間相關程度部份：



可以說是影像編碼演算法中最核心的部份，迄今也提出過許多的處理方法。

- 預測編碼法。
- 正交轉換法。
- 分頻(subband)編碼／小波轉換法。

等為其代表例[30][31]。在本章中，對其中的「正交轉換法[32]」與其中一種離散餘弦轉換(Discrete Cosine Transform)[33]加以說明。特別是 DCT 係用在以 JPEG 或是 MPEG 為首的許多影像編碼方法之中，是一種非常重要的正交轉換方法。

本章首先求取在對影像編碼而言效率最高的Karhunen-Loeve轉換(KLT)，再推導其極限情形之 DCT。

線性轉換與正規化正交轉換

聽到所謂DCT—離散餘弦轉換—的名稱，或許會讓人聯想到什麼特殊複雜的「轉換處理」，實際上也不過就是在高中或是大學一年級中所學過的**一次轉換(線性轉換)**。並且，所謂「正規正交性」是具有相當方便的性質—正規化正交轉換[50]—的轉換方法。本章首先復習線性轉換與正交轉換[34]。

註 50. 「正規化正交轉換」是最正確的名詞，在影像編碼的領域中，常用較簡短的「正交轉換」稱之。

線性轉換

爲了簡化，首先討論侷限於三維空間內的情形，以下的討論可以簡單地擴展至N維空間[51]。

註 51. 只要增加一個(維度)，就失去了「可以把圖畫出來」在直觀上較易理解的優點。

在三維空間內的任意點$\boldsymbol{x}$以直向量$[x_1, x_2, x_3]^t$加以表示[52]，其中x_1、x_2、x_3稱爲點$\boldsymbol{x}$的座標(圖 5.1)。

註 52. 上標^{t}為向量轉列的轉置，將直向量刻意以橫向量的轉置向量表示，放在章節中以節省篇幅。此外，之後在不特別限定的情況下「向量」係以表示直向量。

線性變化即是以適當的轉換矩陣A將向量$\boldsymbol{x}$轉換爲另一向量$\boldsymbol{y}$。可寫作以下形式[53]：

註 53. 除了在本章推導的DCT，廣為所知的離散傅利葉轉換(DFT)等亦可寫作此形式。在DFT中，轉換矩陣A的元素為複數，另外一方面，DCT的各元素為實數。在本書所用到的轉換矩陣，其元素僅限於實數。

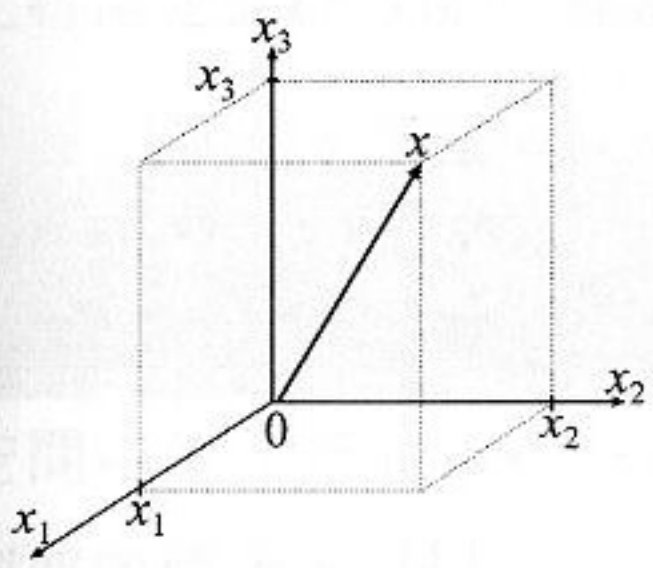


圖 5.1　點x與其座標

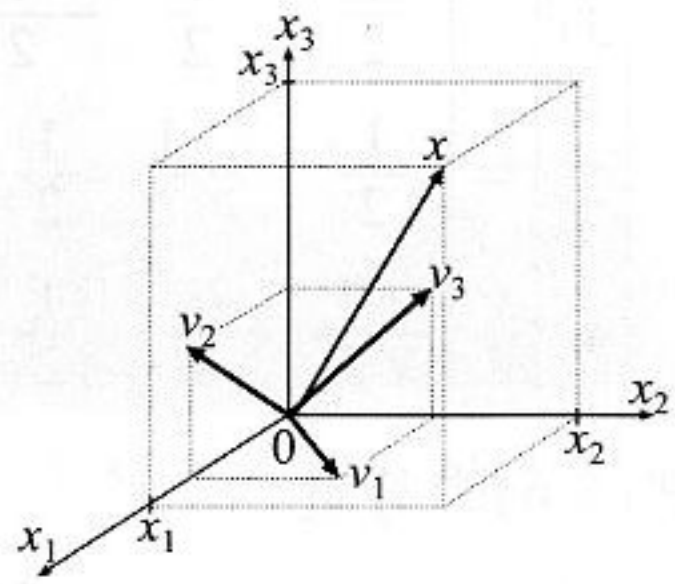


圖 5.2　向量v_1, v_2, v_3

$$y = Ax \tag{5.1}$$

簡單地舉例。考慮現有的三個方向互異(因此，是一次獨立的)的向量：

$$v_1 = [1\ 1\ 0]^t，v_2 = [1\ 0\ 1]^t，v_3 = [0\ 1\ 1]^t \tag{5.2}$$

三維空間的任意點x，可以用三個向量(**基底向量**)的疊加，亦即是線性相加的形式加以表示：

$$x = y_1 v_1 + y_2 v_2 + y_3 v_3 \quad (y_1 \sim y_3 \text{爲適當常數}) \tag{5.3}$$

其中，在實際計算x相對於式(5.3)的常數$y_1 \sim y_3$時。先將基底向量$v_1 \sim v_3$並列，得到如同下列的矩陣V，

$$V = [v_1\ v_2\ v_3] \tag{5.4}$$

式(5.3)可寫作以下矩陣形式：

$$x = Vy \quad \text{其中} \quad y = [y_1\ y_2\ y_3]^t \tag{5.5}$$

因爲$v_1 \sim v_3$爲一次獨立的，在V滿足矩陣值$\det V \neq 0$時具有反矩陣A，寫作$V^{-1} = A$。將式(5.5)的兩側在左邊乘上A之後，成爲：

$$Ax = AVy = y \tag{5.6}$$

可得式(5.1)的線性轉換。實際上計算反矩陣$A = V^{-1}$後，可用：

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{5.7}$$

表示式(5.6)的分量。

考慮基底v_1，v_2，v_3爲三維空間的**新座標軸**，式(5.1)或(5.7)可以解釋爲在該新座標系中的某點$\boldsymbol{x}$的座標$[y_1, y_2, y_3]^t$ [54]。這種「座標轉換」在訊號處理領域中是經常被使用的重要處理過程，實際上在 DCT 或是離散傅利葉轉換中雖然有「…轉換」之名，但是也僅是進行此類座標轉換。

註 54. 其理由可以參見圖 5.2 或是式(5.3)。

爲了求取新座標系內的座標值，必須實際計算式(5.7)中的反矩陣V^{-1}。但是，可以藉由座標軸的選取簡化這項處理。

進行正交轉換

(正交轉換)爲「**選擇構成新座標系的基底$\boldsymbol{v}_1$～$\boldsymbol{v}_3$，使其可以正規化正交**」。**正交**爲**兩向量的內積為零**的性質，**正規化**爲**各基底向量大小為 1** 的性質。因此，在三維空間中的正規正交可以定義爲滿足下式的基底向量$\boldsymbol{v}_i$組合：

$$\begin{cases} \boldsymbol{v}_1 \cdot \boldsymbol{v}_2 = \boldsymbol{v}_2 \cdot \boldsymbol{v}_3 = \boldsymbol{v}_3 \cdot \boldsymbol{v}_1 = 0 \\ \boldsymbol{v}_1 \cdot \boldsymbol{v}_1 = |\boldsymbol{v}_1|^2 = 1 \\ \boldsymbol{v}_2 \cdot \boldsymbol{v}_2 = |\boldsymbol{v}_2|^2 = 1 \\ \boldsymbol{v}_3 \cdot \boldsymbol{v}_3 = |\boldsymbol{v}_3|^2 = 1 \end{cases} \tag{5.8}$$

其中，符號「・」爲向量間的內積。

先將滿足式(5.8)的一組基底向量寫作：

$$\boldsymbol{v}_1=\begin{bmatrix}v_{11}\\v_{21}\\v_{31}\end{bmatrix},\ \boldsymbol{v}_2=\begin{bmatrix}v_{12}\\v_{22}\\v_{32}\end{bmatrix},\ \boldsymbol{v}_3=\begin{bmatrix}v_{13}\\v_{23}\\v_{33}\end{bmatrix} \tag{5.9}$$

三維空間中的任意向量$\boldsymbol{x}=[x_1,x_2,x_3]^t$以式(5.3)的形式表示。對式(5.3)的兩側取內積後，可得：

$$\begin{aligned}\boldsymbol{v}_1\cdot\boldsymbol{x}&=\boldsymbol{v}_1\cdot(y_1\boldsymbol{v}_1+y_2\boldsymbol{v}_2+y_3\boldsymbol{v}_3)\\&=y_1\boldsymbol{v}_1\cdot\boldsymbol{v}_1+y_2\boldsymbol{v}_1\cdot\boldsymbol{v}_2+y_3\boldsymbol{v}_1\cdot\boldsymbol{v}_3=y_1\end{aligned} \tag{5.10}$$

(需注意最後的移項需使用式(5.8)中的性質)。式(5.10)表示「新座標的基底向量 可由基底向量$\boldsymbol{v}_1$與向量$\boldsymbol{x}$的內積加以計算」。y_2、y_3也是一樣，可寫成$y_i=\boldsymbol{v}_i\cdot\boldsymbol{x}\ (i=1,2,3)$，這些以直向量加以整理，可得：

$$\begin{aligned}\begin{bmatrix}y_1\\y_2\\y_3\end{bmatrix}&=\begin{bmatrix}\boldsymbol{v}_1\cdot\boldsymbol{x}\\\boldsymbol{v}_2\cdot\boldsymbol{x}\\\boldsymbol{v}_3\cdot\boldsymbol{x}\end{bmatrix}=\begin{bmatrix}v_{11}x_1+y_{21}x_2+v_{31}x_3\\v_{12}x_1+y_{22}x_2+v_{32}x_3\\v_{13}x_1+y_{23}x_2+v_{33}x_3\end{bmatrix}\\&=\begin{bmatrix}v_{11}&v_{21}&v_{31}\\v_{12}&v_{22}&v_{32}\\v_{13}&v_{23}&v_{33}\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}\end{aligned} \tag{5.11}$$

依式(5.4)定義矩陣V後，注意可將式(5.11)寫作：

$$\boldsymbol{y}=V^t\boldsymbol{x} \tag{5.12}$$

式(5.11)或是式(5.12)爲線性轉換，在形式上雖與式(5.6)相同，但不需計算反矩陣，僅由轉置矩陣V^t便可計算，這點爲兩者最大的不同。

將以上擴展至N維的一般情形，首先，滿足正規正交性的N個基底向量$\boldsymbol{v}_j=\cancel{[x_{1j},x_{2j},\cdots x_{nj}]^t}\ (j=1,2,\cdots,N)$，將其並排，可以定義$N\times N$矩陣：

$[v_{1j}, v_{2j}, \cdots, v_{nj}]^t$

$$V=[\boldsymbol{v}_1\ \boldsymbol{v}_2\ \cdots \boldsymbol{v}_N]=\begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1N} \\ v_{21} & v_{22} & \cdots & v_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ v_{N1} & v_{N2} & \cdots & v_{NN} \end{bmatrix} \tag{5.13}$$

由正規正交性，可以簡單地滿足以下條件[55]：

$$VV^t=V^tV=I \quad (I\ 單位矩陣) \tag{5.14}$$

式(5.12)所表達的是V 與V^t互爲反矩陣，像此類本身的轉置矩陣與反矩陣相同的矩陣稱爲**正交矩陣**。對於求取任意向量$\boldsymbol{x}$在新的座標系$\boldsymbol{v}_1$～$\boldsymbol{v}_N$中的座標 $\boldsymbol{y}$，可參考式(5.12)寫成：

$$\boldsymbol{y}=V^t\boldsymbol{x} \tag{5.15}$$

或是如式(5.11)中以各分量加以表示：

$$y_n=\sum_{k=1}^{N} v_{kn}x_k \quad (n=1,2,\cdots,N) \tag{5.16}$$

反之，可由下式：

$$\boldsymbol{x}=V\boldsymbol{y} \tag{5.17}$$

$$x_n=\sum_{k=1}^{N} v_{nk}y_k \quad (n=1,2,\cdots,N) \tag{5.18}$$

由$\boldsymbol{y}$求取$\boldsymbol{x}$。請注意式(5.16)與式(5.18)中，v的下標順序有所不同(kn與nk)。

註 55. 可試著以三維為例，分別加以計算。

在影像編碼的領域中，如式(5.15)、式(5.16)般，求取任意向量$\boldsymbol{x}$的新座標$\boldsymbol{y}$的處理稱爲**順向轉換**、**正向轉換**，或是簡稱爲**轉換**，反之，求取向量$\boldsymbol{y}$的原座標$\boldsymbol{x}$的處理稱爲**逆轉換**。此外，由轉換所得到的座標y稱爲**轉換係數**或是簡稱爲**係數**。

使用正交轉換之影像編碼演算法

接著，介紹使用正交轉換對編碼影像以得到轉換係數之具體方法。基本的原則爲「將影像分割成適當的區塊(**轉換區塊**)，以區塊爲單位進行適當的正交轉換」。

在此須注意的是，如 2-23 頁的第②項所述，影像的像素間相關程度一般在水平、垂直兩方向上都很高，在實際上，需要對兩方向的相關性都加以去除。因此，圖 5.3 所示，將大小爲$K_1 \times K_2$，的影像以水平方向像素N_1與垂直方向像素N_2的方形區塊加以分割[56]。各方形區塊包含$N_1 \times N_2$像素，因此可用$x_{n2n1}(1 \le n_1 \le N_1$，$1 \le n_2 \le N_2)$表示。$x_{n2n1}$爲二維的配置(矩陣)，如欲代入式(5.15)的形式，需要對其再排列爲一維配置(向量)。通常以轉換區塊內的像素x_{n2n1}依水平掃描順序，以下列方式加以一維化：

$$x_{n2n1} \Rightarrow x_n\big|_{n = N_1 \cdot (n_2 - 1) + n_1} \tag{5.19}$$

將x_n並列，而得到一般化的N維像素向量：

$$\boldsymbol{x} = [x_1\, x_2\, x_3\, \cdots\, x_{N-1}\, x_N]^t \tag{5.20}$$

註 56. 當影像的尺寸不為區塊尺寸的整數倍的情形中，基本上可在影像的右端與下端補上空白像素。之後，假設影像尺寸與經常為區塊尺寸的整數倍。另外，此處的分割，在區塊間不會造成重疊或是間隙。

在此，稍微論及轉換區塊的尺寸。如圖 5.3 所示以$N_1 \times N_2$的轉換區塊分割尺寸爲$K_1 \times K_2$的影像，總共可得到$K_1 K_2 / N_1 N_2$個區塊。另外，因爲一個轉換區塊內有$N = N_1 \times N_2$個像素，在對式(5.20)的向量進行式(5.15)的正交轉換時，需要進行$N^2 = N_1{}^2 \times N_2{}^2$次的乘法運算。接著，計算對影像整體進行正交轉換所需總乘法計算次數N_{mul}爲：

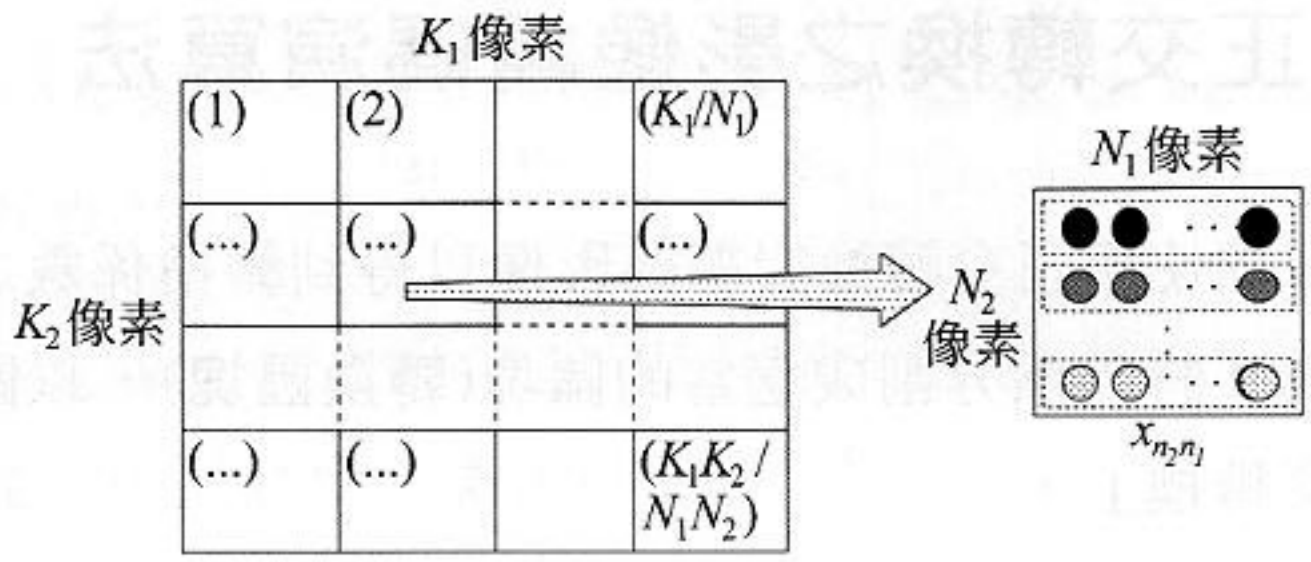


圖 5.3　將影像分割成轉換區塊

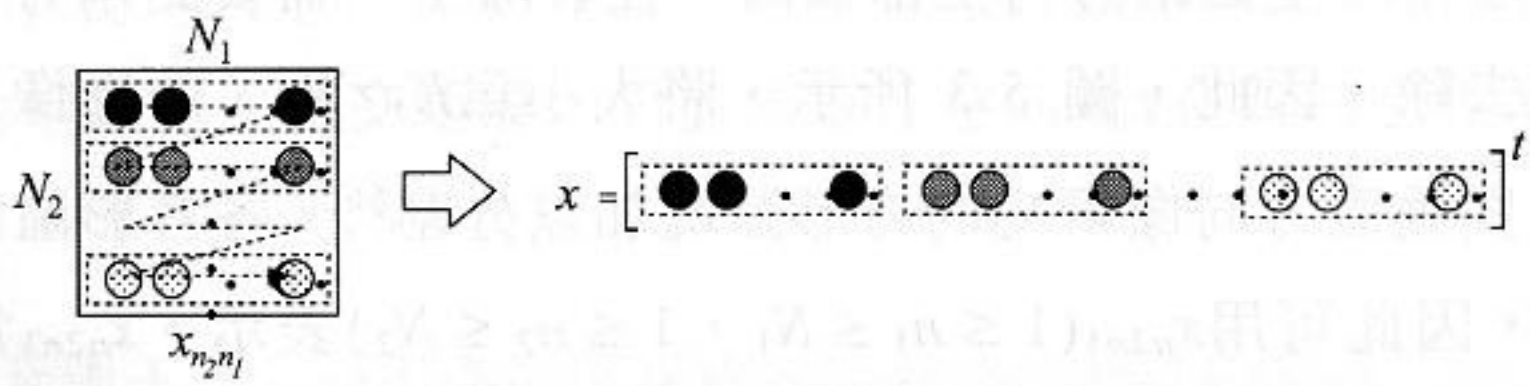


圖 5.4　將轉換區塊內的像素轉為N維向量的組成方法

$$N_{\text{mul}}=\frac{K_1K_2}{N_1N_2}\times N_1{}^2N_2{}^2=K_1K_2N_1N_2 \quad (\text{次}) \tag{5.21}$$

如此一來，雖然由計算量(總乘法次數)觀之，將轉換區塊取小尺寸較爲有利，但另一方面由去除像素間相關的程度觀之，取的太小並不是好方法。在實際的 JPEG 或 MPEG 的演算法中，選轉換區塊大小爲 8×8。關於此問題會在第六章最後再度討論，目前先進行轉換區塊爲任意尺寸的討論。

以上所述之使用分割區塊的正交轉換編碼方法，可依下列順序的演算法加以進行(圖 5.5)：

① 以給定尺寸的轉換區塊分割影像，得到各區塊的像素向量$\boldsymbol{x}=[x_n]$。

② 對各$\boldsymbol{x}$進行正交轉換，得到轉換係數$\boldsymbol{y}=[y_n]$。

③ 轉換係數y_n藉由量子化器$Q_n(n=1,2,\cdots,N)$，以**轉換係數為單位**加以量子化。

④　對量子化後的係數進行熵編碼。

在步驟③中，轉換係數y_n係以各係數爲單位加以量子化，藉由正交轉換而進行的非相關化將y_n轉換爲互不干涉的資料。

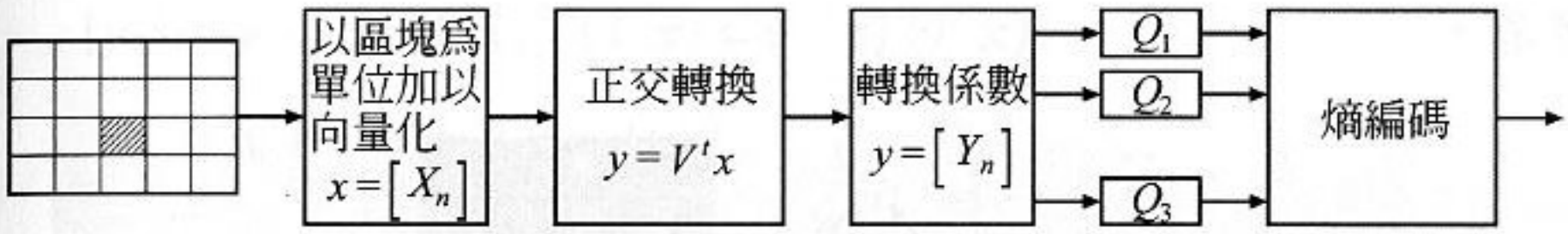


圖 5.5　由正交轉換進行影像編碼演算法之組成

如第四十一頁中的項目①所述，雖可藉「取較大的轉換區塊尺寸」提高壓縮效率，但**在此情形下如何求取轉換矩陣為佳**？爲了準備回答這個問題，在接下來的兩節進行數學上的準備。其一爲正規正交轉換具有的重要性質—巴塞瓦性質，另外一個爲對轉換係數進行量子化時，使影響降爲最小之抑制方法—最佳位元分配理論。

巴塞瓦性質與其應用

最基本的正規正交基底爲自然基底(三座標軸)，在三維空間中，可寫作：

$$\boldsymbol{v}_1=[1\ 0\ 0]^t，\boldsymbol{v}_2=[0\ 1\ 0]^t，\boldsymbol{v}_3=[0\ 0\ 1]^t \tag{5.22}$$

另外，簡單地確認，以下的三個向量亦組成一組正規正交基底：

$$\boldsymbol{v}_1'=\frac{[1\ 1\ 1]^t}{\sqrt{3}}，\boldsymbol{v}_2'=[1\ 0\ -1]^t/\sqrt{2}，\boldsymbol{v}_3'=[1\ -2\ 1]^t/\sqrt{6} \tag{5.23}$$

圖 5.6(a)爲自然基底$\boldsymbol{v}_1-\boldsymbol{v}_2-\boldsymbol{v}_3$與式(5.23)中$\boldsymbol{v}_1'-\boldsymbol{v}_2'-\boldsymbol{v}_3'$的組合的關係表示圖。兩者在「形式」上相同，皆由「三個相互正交，且長度爲 1 的

向量」所組成，相異的僅爲互相的夾角。實際上，假設姆指、食指、中指的長度爲 1，以左手比出正規正交座標系，可以得到圖 5.6(b)～(d)的結果，這些共通點是「手指的形狀」是相同的，但是僅是「角度」有所差異。不管再怎麼變化，其他「手指的形狀」無法組成三維的正規正交座標系。

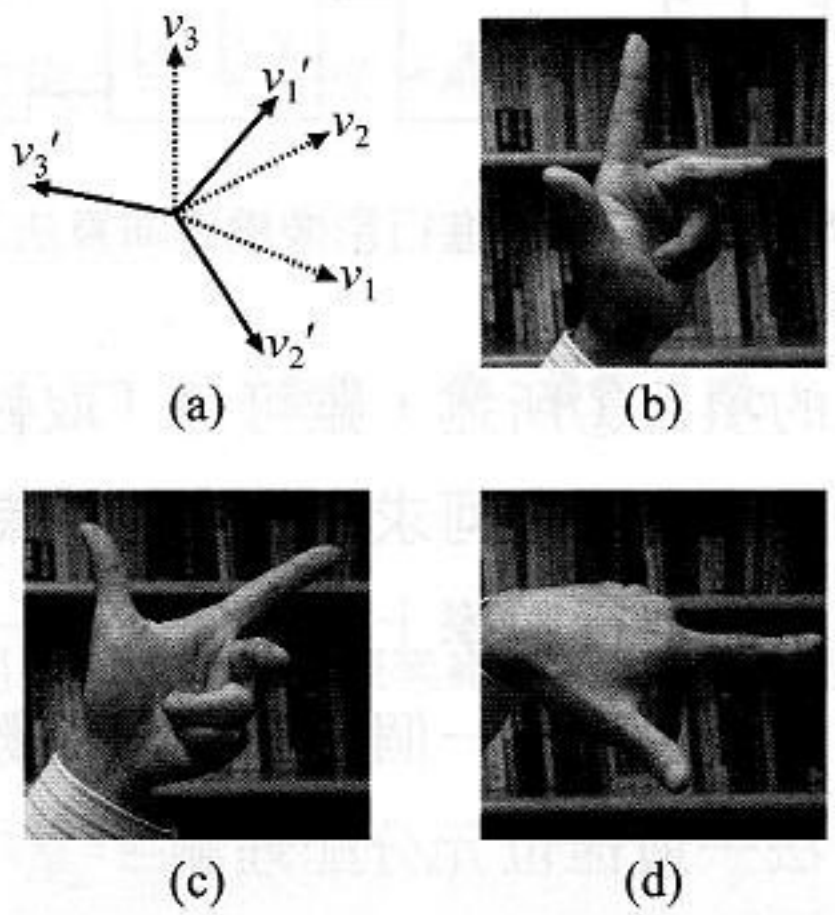


圖 5.6

這個事實，表示「雖然可以組成無數個三維的正規正交座標系，但是其間的差別僅在於軸的角度，**其『形狀』就只有一種而已**」。當然這是個非常重要的性質。此外，此性質不僅限於三維空間，不管在幾維的空間內皆成立。也就是說，在有兩組N維正規正交座標系的情形中，其中一組適當地旋轉後，會與另外一組重合。

如先前所述「某一向量以新的座標系觀之可以賦予新的座標值」。與上述的事實一起考量，可以得到**正交轉換僅為旋轉被轉換對象之向量**的結論。另外，旋轉具有「**向量長度持不變**」之性質[57]，在如式(5.15)～式(5.18)之正交轉換或是逆轉換中，**轉換前後的向量大小保持不變**。

註 57. 但向量長需以歐幾里德距離加以測定。

以上的性質以方程式加以表示後，設正交轉換矩陣爲V，可寫作：

$$|\boldsymbol{y}|=|V^t\boldsymbol{x}|=|\boldsymbol{x}| \tag{5.24}$$

可以下列方式簡單地加以證明：

$$\begin{aligned}|\boldsymbol{y}|^2&=\boldsymbol{y}\cdot\boldsymbol{y}=(V^t\boldsymbol{x})\cdot(V^t\boldsymbol{x})=(V^t\boldsymbol{x})^t(V^t\boldsymbol{x})\\&=\boldsymbol{x}^t\underbrace{VV^t}_{I}\boldsymbol{x}=\boldsymbol{x}^t\boldsymbol{x}=|\boldsymbol{x}|^2\end{aligned} \tag{5.25}$$

式(5.24)爲在訊號處理領域中稱爲**巴塞瓦(Parseval)關係**之重要的關係。

套用式(5.24)後，可以推導在轉換編碼中非常重要的性質。原影像的轉換區塊(向量)$\boldsymbol{x}$進行正交轉換V，可以得到轉換係數$\boldsymbol{y}=V^t\boldsymbol{x}$。$\boldsymbol{y}$爲在次段中以量子化器進行量子化，在解碼端進行反量子化後會得到具有量子化誤差的轉換係數。此稱爲$\hat{\boldsymbol{y}}$，進一步地進行反轉換時像素區塊可依$\hat{\boldsymbol{x}}=V\hat{\boldsymbol{y}}$加以解碼。取$\boldsymbol{x}$與$\hat{\boldsymbol{x}}$的差，可得：

$$\boldsymbol{x}-\hat{\boldsymbol{x}}=V\boldsymbol{y}-V\hat{\boldsymbol{y}}=V(\boldsymbol{y}-\hat{\boldsymbol{y}}) \tag{5.26}$$

由巴塞瓦關係可得：

$$|\boldsymbol{x}-\hat{\boldsymbol{x}}|=|\boldsymbol{y}-\hat{\boldsymbol{y}}| \quad \text{(或是等號兩邊的平方)} \tag{5.27}$$

也就是說，由量子化造成的誤差，如式(5.27)所示，在**解碼後的影像空間中所測得(左邊)與解碼前的轉換係數空間中所測得(右邊)，是完全等價的**。

轉換係數的最佳位元分配[35]

影像編碼的目的也可以說是「以給定的資料量，儘可能地減少原影像與解碼影像之間的誤差」。在此，誤差以平方差(歐幾里德距離)加以

測量，依式(5.27)的性質可知：**為了抑制在在影像空間中的誤差，需抑制在轉換係數空間中量子化前後的誤差**。

再度參考圖 5.5，y_k爲相應的量子化器Q_k進行量子化的結果，爲了減少量子化誤差其量子化等級L_k(或是等價的量子化位元長度r_k)需取較大值。但是，如此一來會使得單就係數整體的資料量$\sum_n r_n$增加，此爲不希望見到的。接著，進行以下考慮。

首先，由量子化產生每個區塊的係數y，(各區塊內的)平均平方差爲d，求取目標影像中的全部區塊d之平均值[58] ($D=E_m[\cdot]$爲整個區塊中的平均值)。接著，對各區塊的係數y賦予平均R (位元)的量子化位元長度[59]，對於y_k分配量子化位元長度r_k，以滿足$R=1/N\sum_{n=1}^{N} r_N$。這種分配方式雖然具有無數的組合方式，但在其中，求取最「好的」r_k組合方式使其中的平均誤差D最小化。如此一來，可藉由量子化使誤差得以抑制在最小。

註 58. D(Distortion)。
59. R(Rate)。

以上的問題加以定量化後可寫作以下形式：

$$\arg\min_{r_1\cdots r_N} D = \arg\min_{r_1\cdots r_N} E_m[d] \tag{5.28}$$

$$\text{Subject to } R=\frac{1}{N}\sum_{n=1}^{N} r_n$$

可以下列方式加以解讀「在$R=1/N\sum_{n=1}^{N} r_N$的條件下，在所有的$r_1,\cdots,r_N$組合中，使得$D=E_m[d]$爲最小」。在這個解下，各量子化的位元長度r_k稱爲**最佳位元分配**。以下以**各量子化器Q_k採用一致量子化器**的情形中，計算最佳位元分配。

現在，由量子化所得的係數y_k造成的量子化誤差e_k，區塊內的平均平方差爲：

$$d=\frac{1}{N}\sum_{n=1}^{N}e_n^2 \tag{5.29}$$

可寫作：

$$D=E_m[d]=E_m\left[\frac{1}{N}\sum_{n=1}^{N}e_n^2\right]=\frac{1}{N}\sum_{n=1}^{N}E_m[e_n^2]$$

$$=\frac{1}{N}\sum_{n=1}^{N}\sigma_{en}^2 \tag{5.30}$$

σ_{ek}^2爲量子化誤差e_k的離散程度[60]。在此，假設各量子化器Q_k採用r_k位元的一致量子化器。利用式(4.19)可將σ_{ek}^2以的y_k離散程度σ_{yk}^2與常數ε_{yk}^2寫作：

$$\sigma_{ek}^2=\varepsilon_{yk}^2\sigma_{yk}^2 2^{-2r_k} \tag{5.31}$$

由以上結果，可將式(5.28)的問題改寫成：

$$\arg\min_{r_1\cdots r_N} D=\arg\min_{r_1\cdots r_N}\frac{1}{N}\sum_{n=1}^{N}\varepsilon_{yn}^2\sigma_{yn}^2 2^{-2r_n} \tag{5.32}$$

$$\text{Subject to } R=\frac{1}{N}\sum_{n=1}^{N}r_n \tag{5.33}$$

求解之後，可以得到最佳的位元分配。

註　60. e_k 為單一區塊內產生的誤差，其平均值可假設為零。

此處有一點必須補充。量子化等級數雖可寫作$L_k=2^{r_k}$，但L_k需爲正整數，式(5.32)除了拘束條件(5.33)還需在$r_k=\log_2$整數，$r_k\geq 0$等條件下進行最小化。此爲關於L_k的整數規劃問題[61]，要以解析方式進行處理是相當困難的。因此，需將這兩個條件給去除，僅在式(5.33)的規範下進行最小化的嘗試。其結果爲，各等級數$L_k=2^{r_k}$爲實數，當L_k大到某個程

度之後附近最接近的整數也不會造成過大的誤差。另外，雖然也可以得到$r_k<0$的解，但是關於此點於之後討論，首先先考慮式(5.32)與式(5.33)的解法。

註 61. 整數值的最佳化問題，在尋找最佳解時需以電腦進行數值計算。

一般而言，對於式(5.32)～(5.33)之具有拘束條件的最小化問題，可利用拉格朗日乘子法(Lagrange multiplier method)[36]轉換成以下沒有拘束條件之最小化問題的形式(λ爲拉格朗日乘子)：

$$\arg\min_{r_1\cdots r_N}\left[\frac{1}{N}\sum_{n=1}^{N}\varepsilon_{yn}{}^2\sigma_{yn}{}^2 2^{-2r_n}-\lambda\left(R-\frac{1}{N}\sum_{n=1}^{N}r_n\right)\right] \tag{5.34}$$

也就是說，式(5.32)～(5.33)的問題之解，可以寫爲使式(5.34)最小化之的組合，式(5.34)對r_n $(n=1,2,\cdots N)$進行偏微分後，取其值爲零，可得到如：

$$\frac{1}{N}(-2\log_e 2)\varepsilon_{yn}{}^2\sigma_{yn}{}^2 2^{-2r_n}+\frac{1}{N}\lambda=0 \quad (n=1,2,\cdots N) \tag{5.35}$$

之N 個條件式。將這些式子整理爲：

$$2^{-2r_n}=\frac{\lambda}{2\log_e 2}\frac{1}{\varepsilon_{yn}{}^2\sigma_{yn}{}^2} \quad (n=1,2,\cdots,N) \tag{5.36}$$

將$(n=1,2,\cdots N)$代入後並相乘，可得：

$$2^{-2(r_1+r_2+\cdots+r_N)}=\left(\frac{\lambda}{2\log_e 2}\right)^N\frac{1}{\prod_{k=1}^{N}\varepsilon_{yk}{}^2\sigma_{yk}{}^2} \tag{5.37}$$

由式(5.33)可知$(r_1+r_2+\cdots+r_N)=NR$，利用此關係，取等號兩邊的N 次方根，可將等式改寫爲：

$$2^{-2R}=\frac{\lambda}{2\log_e 2}\frac{1}{\left(\prod_{k=1}^{N}\varepsilon_{yk}{}^2\sigma_{yk}{}^2\right)^{1/N}} \tag{5.38}$$

將式(5.36)與(5.38)相除，消去λ，可得：

$$\frac{2^{-2R}}{2^{-2r_n}}=\frac{\varepsilon_{yn}{}^2\sigma_{yn}{}^2}{\left(\prod_{k=1}^{N}\varepsilon_{yk}{}^2\sigma_{yk}{}^2\right)^{1/N}}\quad(n=1,2,\cdots,N)\tag{5.39}$$

求解r_n，可以計算最佳的位元分配方式爲：

$$r_n=R+\frac{1}{2}\log_2\frac{\varepsilon_{yn}{}^2\sigma_{yn}{}^2}{\left(\prod_{k=1}^{N}\varepsilon_{yn}{}^2\sigma_{yk}{}^2\right)^{1/N}}\quad(n=1,2,\cdots,N)\tag{5.40}$$

將式(5.40)代入式(5.31)，可以得到在最佳位元分配方式下各量子化Q_K器產生的誤差之離散程度($\varepsilon_{e_k\text{opt}}{}^2$)。實際計算後，可得：

$$\varepsilon_{e_k\text{opt}}{}^2=2^{-2R}\left(\prod_{n=1}^{N}\varepsilon_{yn}{}^2\sigma_{yn}{}^2\right)^{1/N}\tag{5.41}$$

但是有趣的是，**等式右側與k無關而為固定值**，結果可得到結論爲：**在最佳的位元分配方式之下，各量子化器Q_k產生的誤差之離散程度$\varepsilon_{e_k\text{opt}}{}^2$在$k=1,2,\cdots,N$時是相等的**。

此外，將式(5.41)代入式(5.30)，可以得到平均平方差D的最小值($D_{\min}$)爲：

$$D_{\min}=2^{-2R}\left(\prod_{k=1}^{N}\varepsilon_{yk}{}^2\sigma_{yk}{}^2\right)^{1/N}\tag{5.42}$$

另外，在式(5.40)中，當R較大的情況下，可得到各r_n爲正值，如先前所述，取最接近2^{r_k}之整數L_k爲係數y_k之量子化等級，可以得到近似的最佳位元分配。另外，當R 較小時，可能會有某個k值時，會使得$r_k<0$。在此情形下，$r_k<0$ 爲y_k之位元長度爲 0(也就是說不進行編碼，直接捨棄)，剩下的y_k以平均編碼長度R再重行分配，重新計算最佳位元之分配方式。此問題在文獻[31]中有所討論，供有興趣的讀者參考。

Karhunen-Loeve 轉換(KLT)

KLT 之推導

現在，與圖 5.3 相同，對$K_1 \times K_2$的影像以$N_1 \times N_2$之轉換區塊加以分割，對各區塊以水平掃描順序賦予$m=1,2,\cdots$的編號。此時，總區塊數$M \equiv K_1 K_2 / N_1 N_2$，每個區塊內的總像素數目爲$N \equiv N_1 N_2$。各區塊的像素以圖 5.4 或是式(5.19)、(5.20)所示，以水平掃描順序排列爲：

$$\begin{cases} \boldsymbol{x}^{(1)} = [x_1^{(1)} x_2^{(1)} \cdots x_N^{(1)}]^t \\ \quad \vdots \\ \boldsymbol{x}^{(m)} = [x_1^{(m)} x_2^{(m)} \cdots x_N^{(m)}]^t \\ \quad \vdots \\ \boldsymbol{x}^{(M)} = [x_1^{(M)} x_2^{(M)} \cdots x_N^{(M)}]^t \end{cases} \tag{5.43}$$

我們的目的，在於求得對各$\boldsymbol{x}^{(m)}$之平均的**最佳正交轉換**：

$$\boldsymbol{y}^{(m)} = V^t \boldsymbol{x}^{(m)} \quad (m = 1,2,\cdots, M) \tag{5.44}$$

對此，在數個正交轉換當中，要如何判斷滿足什麼要的條件才是**最佳**呢？如之前所提到的，實施正交轉換的目的爲去除像素間的相關程度，在此，可以想成「將各轉換區塊內的N個像素加以**非相關化**的正交轉換」就是最佳轉換。其強健性於之後再加以討論，首先推導滿足此條件之轉換。

首先，需在各區塊內對於轉換前後的相關程度加以定量化。可使用式(2.5)中的協方差：

$$E_x[(f(x)-\mu_f)(g(x)-\mu_g)]$$

此處因爲係以區塊分割影像，因此需要若干的修正。因此，在式(2.5)中①$f(x)$與$g(x)$分別以$\cdot_i^{(m)}$、$\cdot_j^{(m)}$取代($\cdot$相當於x或y)，②將μ_f、μ_g換爲相對應之平均值，③所計算的平均值E_x修正爲對整個區塊內取平均值E_m。如此一來，像素x_i、x_j之間或是轉換係數y_i、y_j之間的相關係數，可以下列方式加以測量($\overline{\cdot}$爲平均值)：

$$C_{x_ix_j} = E_m[(x_i^{(m)} - \overline{x_i^{(m)}})(x_j^{(m)} - \overline{x_j^{(m)}})] \tag{5.45}$$

$$C_{y_iy_j} = E_m[(y_i^{(m)} - \overline{y_i^{(m)}})(y_j^{(m)} - \overline{y_j^{(m)}})] \tag{5.46}$$

此處參照式(5.44)，轉換係數 $\boldsymbol{y}^{(m)}$的第i個元素可寫爲：

$$y_i^{(m)} = \sum_{n=1}^{N} v_{ni} x_n^{(m)} \quad (v_{ni}\text{爲轉換矩陣}V\text{的元素}) \tag{5.47}$$

因此，等式兩側的區塊內平均值爲：

$$\begin{aligned}\overline{y_i^{(m)}} &= E_m[y_i^{(m)}] = E_m\left[\sum_{n=1}^{N} v_{ni} x_n^{(m)}\right] \\ &= \sum_{n=1}^{N} v_{ni} E_m[x_n^{(m)}] = \sum_{n=1}^{N} v_{ni}\, \overline{x_n^{(m)}}\end{aligned} \tag{5.48}$$

如此一來，在計算轉換係數間的相關程度時，必須將式(5.48)求取的平均值代入式(5.46)，而使得計算變得複雜。在此，假設**各像素值**$x_i^{(m)}$**已經先減去平均值**。亦即考慮：

$$x_i^{(m)} \Leftarrow x_i^{(m)} - \overline{x_i^{(m)}} \quad (i = 1, 2, \cdots, N) \tag{5.49}$$

此時，左邊的新$x_i^{(m)}$爲：

$$\overline{x_i^{(m)}} = E_m[x_i^{(m)}] = 0 \quad (i = 1, 2, \cdots, N) \tag{5.50}$$

式(5.45)、(5.46)可以寫作以下的簡單形式[62]：

註　62. 如 $y_i^{(m)}$，可使得式(5.50)與(5.48)更易了解。

$$C_{x_ix_j} = E_m[x_i^{(m)}x_j^{(m)}]，C_{y_iy_j} = E_m[y_i^{(m)}y_j^{(m)}] \tag{5.51}$$

此處，式(5.51)之$C_{x_ix_j}$、$C_{y_iy_j}$以具有ij個元素之矩陣分別加以定義爲：

$$C_{xx} = \begin{bmatrix} E_m[x_1^{(m)}x_1^{(m)}] & \cdots & E_m[x_1^{(m)}x_N^{(m)}] \\ \vdots & \ddots & \vdots \\ E_m[x_N^{(m)}x_1^{(m)}] & \cdots & E_m[x_N^{(m)}x_N^{(m)}] \end{bmatrix} \tag{5.52}$$

$$C_{yy} = \text{上式中的}x\text{置換爲 }y \tag{5.53}$$

稱爲**協方矩陣**。由其構造可以輕易地知道，協方矩陣具有以下性質：

- 爲對稱矩陣。
- ii對角線上元素 或是y_i的離散程度相等。

另外，使用式(5.51)，可知式(5.52)、(5.53)可以向量形式記爲：

$$C_{xx} = E_m[\boldsymbol{x}^{(m)}(\boldsymbol{x}^{(m)})^t] \tag{5.54}$$

$$C_{yy} = E_m[\boldsymbol{y}^{(m)}(\boldsymbol{y}^{(m)})^t] \tag{5.55}$$

此外，$\boldsymbol{y}^{(m)}$與$\boldsymbol{x}^{(m)}$有著式(5.44)的關係，再代入式(5.55)可以推導出以下關係[63]：

$$\begin{aligned} C_{yy} &= E_m[V^t\boldsymbol{x}^{(m)}(V^t\boldsymbol{x}^{(m)})^t] = V^tE_m[\boldsymbol{x}^{(m)}(\boldsymbol{x}^{(m)})^t]V \\ &= V^tC_{xx}V \quad (\text{因}V\text{爲常數矩陣}) \end{aligned} \tag{5.56}$$

註 63. 自式(5.56)的第二式轉換為第三式，係運用V對取平均值之運算E_m為常數。

我們的目的，在於求取使得$\boldsymbol{y}^{(m)}$非相關化之矩陣V。爲了便於理解，非相關的$\boldsymbol{y}^{(m)}$之協方矩陣C_{yy}爲**對角矩陣**。如此一來，V具有爲**協方矩陣C_{xx}之對角化矩陣**的特徵。

由矩陣的特徵值理論[34]，在式(5.56)中，如左邊的C_{yy}爲對角矩陣，V爲C_{xx}之特徵向量加以並排所構成之矩陣。實際上，像C_{xx}這種**對稱矩陣** A，具有以下性質[34]：

① A的特徵值爲實數，包含重根共有N個，爲$\lambda_1, \cdots, \lambda_N$。

② 共可得到N 條對應特徵值的特徵向量，爲$\boldsymbol{v}_1, \cdots \boldsymbol{v}_N$。

③ 所有的特徵向量相互正交。

④ 必定可藉由所得到的特徵向量將A加以對角化，也就是說作成矩陣$V=[\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_N]$之後

$$\mathrm{diag}\,[\lambda_1, \cdots, \lambda_N] = V^t A V$$

可進行以下轉換(diag[⋯]爲對角矩陣)。

在此，求C_{xx}之特徵值λ_i，依其大小加以排列[64]，求取與其對應之特徵向量$\boldsymbol{v}_i$，再以滿足$|\boldsymbol{v}_i| = 1$加以正規化。

特徵值	λ_i	$\geq$	λ_2	$\geq$	$\cdots$	$\geq$	λ_N
	↓		↓				↓
特徵向量	$\boldsymbol{v}_1$		$\boldsymbol{v}_2$				$\boldsymbol{v}_N$

此時，並排所得到的特徵向量(依其對應之特徵值之大小)可以作成轉換矩陣：

$$\boldsymbol{V} = [\boldsymbol{v}_1\ \boldsymbol{v}_2 \cdots \boldsymbol{v}_N] \tag{5.57}$$

依式(5.44)轉換$\boldsymbol{x}^{(m)}$，由式(5.56)可得$C_{yy} = \mathrm{diag}[\lambda_1, \cdots, \lambda_N]$。依上述所作成的轉換矩陣$V$，可由上述的性質③與$|\boldsymbol{v}_i| = 1$進行正規正交轉換。此爲最適合吾人目的之轉換矩陣，一般稱作**Karhunen-Loeve轉換(KLT)**。

註 64. 協方差矩陣之特徵值必定為正。

KLT 之最佳化性質

再次參見圖 5.5。現在取無數個正交轉換中之一組轉換方式V，可得到係數向量 $\boldsymbol{y}$ [65]。在此情況下，各轉換係數y_n之間並非無相關的。爲了

可以實際計算其離散程度σ_{yn}^2，使用式(5.40)，姑且不論是否可以設計出在式(5.42)中滿足最小誤差$D_{\min}$之最佳量子化器Q_k ($n=1,2,\cdots,N$)，但是，正交轉換V 可以不同方式來進行，因此$D_{\min}$的值也會變化，因此可自然地將$D_{\min}$視爲V的函數$D_{\min}(V)$。在先前所求得的KLT，實際上就是在所有的正交轉換中**使得$D_{\min}(V)$為最小的最佳轉換**。

爲了證明，對式(5.42)的常數ε_{yn}^2($n=1,2,\cdots,N$)進行以下假設。ε_{yn}^2爲依照式(4.18)加以定義的常數，由轉換係數y_n之分佈形狀決定，在此，假設各ε_{yn}^2 ($n=1,2,\cdots,N$)近似於固定值σ^2 [66]。在此假設下，式(5.42)可寫作：

$$D_{\min}(V)=2^{-2R}\varepsilon^2\left(\prod_{k=1}^{N}\sigma_{yk}^2\right)^{1/N} \tag{5.58}$$

與V相關的部份寫在(・)裏頭。因此，相對於 KLT 之轉換矩陣V_{KLT}，式(5.58)之$\prod_{k=1}^{N}\sigma_{yk}^2$部份爲最小。因此，可利用接下來的定理。

定理 1

對任意的協方差矩陣，下列不等式成立[37]。

$$\det C_{yy} \le \prod_{k=1}^{N}\sigma_{yk}^2 \tag{5.59}$$

其中，σ_{yk}^2爲C_{yy}之kk對角元素，等號僅在C_{yy}爲對角矩陣時成立。

註 65. 在本節之假設中，已先減去像素之平均值。

66. 此假設在$\boldsymbol{x}$為常態分布時才嚴謹地成立，因為，轉換係數y_n為x_n之疊加(參照式(5.47))，當各x_n係依常態分布時，其疊加y_n之分布也是常態分布。此時，ε_{yn}^2等於對所有常態分佈之修正值ε^2。但是當y_n為常態分布的情形中，各量子化器發生溢位的機率不為零，使得在第四章中導出式(4.19)之假設不成立。因此在此將$\varepsilon_{yn}^2=\varepsilon^2$的假設想成在**近似的情況下**成立。

請注意使用 KLT 時 ~~det~~ C_{yy} 爲對角矩陣，式(5.59)之等號成立。另外，計算式(5.56)兩側的行列式，可得：

$$\begin{aligned}\det C_{yy} &= \det(V^t C_{xx} V) = \det C_{xx} \det(V^t V) \\ &= \det C_{xx}\end{aligned} \tag{5.60}$$

$\det C_{yy}$ 與使用的轉換矩陣 V 無關，可知其係**僅由目標影像決定的常數**。

由以上所述，在將式(5.59)代入式(5.58)的同時，可用式(5.60)對任意的轉換矩陣寫成：

$$\begin{aligned}&\underbrace{D_{\min}(V) = 2^{-2R}\varepsilon^2\left(\prod_{k=1}^{N}\sigma_{yk}^2\right)^{1/N}}_{\text{與 } V \text{ 相關}} \\ &\geq 2^{-2R}\varepsilon^2[\det C_{yy}]^{1/N} = \underbrace{2^{-2R}\varepsilon^2[\det C_{xx}]^{1/N}}_{\text{與 } V \text{ 無關的常數}}\end{aligned} \tag{5.61}$$

因此，當 $V = V_{\mathrm{KLT}}$ 式(5.61)的等號成立，使得 $D_{\min}$ 爲最小值(式(5.61 之最右側))。

最後，使用圖 5.5 之正交轉換的 KLT，可以對其轉換係數以係數爲單位進行最佳量子化，使得解碼誤差爲最小。這表示，KLT 爲所有的正較轉換中之最佳轉換。但是，**雖然可證明 KLT 之最佳性質，但是需注意其範圍僅限於如圖 5.5 之「正交轉換＋量子化」之演算法**[67]。

註　67. 這就是說，除了圖 5.5 以外的結構，還有更有效的演算法之組成方式。但是，此問題直至現在仍不可解。但是在經驗上「圖 5.5 的『正交轉換＋量子化』組合較其他的方法具有更好的效果」。在實際上亦廣泛地加以使用。

實際的 KLT 計算方法

求取 KLT 時需有協方差矩陣之特徵向量，其計算一般來說並不容

易。但是在影像之自協方差函數可如式(2.11)在水平與垂直方向上**可加以分離**的情況中，其計算可大幅地簡化。在這裏，討論此情形。

以$N_1 \times N_2 = 3 \times 2$之小區塊尺寸的情況爲例。$3 \times 2$之轉換區塊爲：

$$X = \begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \end{bmatrix} \tag{5.62}$$

將此依圖 5.4 加以向量化，可表示爲(在此省略區塊編號m)：

$$\boldsymbol{x} = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6]^t \tag{5.63}$$

在KLT的計算中，最初必須以式(5.22)計算協方差C_{xx}，在此，與式(2.11)相同，C_{xx}之ij 元素之$E_m[x_i x_j]$假設可寫爲：

$$E_m[x_i\, x_j] = \rho_{\mathrm{H}}^{[x_i\text{與}x_j\text{的水平距離}]} \cdot \rho_{\mathrm{V}}^{[x_i\text{與}x_j\text{的垂直距離}]} \tag{5.64}$$

之**水平方向與垂直方向分離**之形式($\sigma^2 = 1$)。舉例來說，$E_m[x_1\, x_6]$參考式(5.62)成爲$\rho_{\mathrm{H}}^2 \rho_{\mathrm{V}}$，此時，$C_{xx}$可以下列方式加以計算：

$$C_{xx} = \begin{array}{c} \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array} \\ \begin{bmatrix} 1 & \rho_{\mathrm{H}} & \rho_{\mathrm{H}}^2 & \rho_{\mathrm{V}} & \rho_{\mathrm{V}}\rho_{\mathrm{H}} & \rho_{\mathrm{V}}\rho_{\mathrm{H}}^2 \\ \rho_{\mathrm{H}} & 1 & \rho_{\mathrm{H}} & \rho_{\mathrm{V}}\rho_{\mathrm{H}} & \rho_{\mathrm{V}} & \rho_{\mathrm{V}}\rho_{\mathrm{H}} \\ \rho_{\mathrm{V}}\rho_{\mathrm{H}}^2 & \rho_{\mathrm{H}} & 1 & \rho_{\mathrm{V}}\rho_{\mathrm{H}}^2 & \rho_{\mathrm{V}}\rho_{\mathrm{H}} & \rho_{\mathrm{V}} \\ \rho_{\mathrm{V}} & \rho_{\mathrm{V}}\rho_{\mathrm{H}} & \rho_{\mathrm{V}}\rho_{\mathrm{H}}^2 & 1 & \rho_{\mathrm{H}} & \rho_{\mathrm{H}}^2 \\ \rho_{\mathrm{V}}\rho_{\mathrm{H}} & \rho_{\mathrm{V}} & \rho_{\mathrm{V}}\rho_{\mathrm{H}} & \rho_{\mathrm{H}} & 1 & \rho_{\mathrm{H}} \\ \rho_{\mathrm{V}}\rho_{\mathrm{H}}^2 & \rho_{\mathrm{V}}\rho_{\mathrm{H}} & \rho_{\mathrm{V}} & \rho_{\mathrm{H}}^2 & \rho_{\mathrm{H}} & 1 \end{bmatrix} \end{array} \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{array} \tag{5.65}$$

對此詳細觀察後，可以看出如同

$$C_{xx}=\begin{bmatrix} 1\cdot\begin{bmatrix}1 & \rho_H & \rho_H^2\\ \rho_H & 1 & \rho_H\\ \rho_H^2 & \rho_H & 1\end{bmatrix} & \rho_V\cdot\begin{bmatrix}1 & \rho_H & \rho_H^2\\ \rho_H & 1 & \rho_H\\ \rho_H^2 & \rho_H & 1\end{bmatrix}\\ \rho_V\cdot\begin{bmatrix}1 & \rho_H & \rho_H^2\\ \rho_H & 1 & \rho_H\\ \rho_H^2 & \rho_H & 1\end{bmatrix} & 1\cdot\begin{bmatrix}1 & \rho_H & \rho_H^2\\ \rho_H & 1 & \rho_H\\ \rho_H^2 & \rho_H & 1\end{bmatrix}\end{bmatrix} \tag{5.66}$$

具有矩陣之「網格」結構，這種構造的矩陣，可以用**Kronecker積**簡單地加以表現。

所謂Kronecker積，就是任意矩陣(或是向量)$A=[a_{ij}]$(m行n列)與$B=[b_{ij}]$(p行q列)以下列定義加以計算：

$$A\otimes B=\begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B\\ a_{21}B & a_{22}B & \cdots & a_{2n}B\\ \vdots & \vdots & \ddots & \vdots\\ a_{n1}B & a_{m2}B & \cdots & a_{mn}B\end{bmatrix} \tag{5.67}$$

也就是說，$A\otimes B$係B爲A中間的網格構造，其大小爲pm行，qn列。參照式(5.67)之定義與式(5.66)，可將式(5.65)寫成：

$$C_{xx}=\underbrace{\begin{bmatrix}1 & \rho_V\\ \rho_V & 1\end{bmatrix}}_{C_V}\otimes\underbrace{\begin{bmatrix}1 & \rho_H & \rho_H^2\\ \rho_H & 1 & \rho_H\\ \rho_H^2 & \rho_H & 1\end{bmatrix}}_{C_H} \tag{5.68}$$

在接下來的步驟，計算C_{xx}之特徵值與特徵向量。可利用以下的定理[38]。

定理 2

對於$N_A \times N_A$方陣A與$N_B \times N_B$方陣B，其特徵值與特徵向量可寫爲以下形式。

A：特徵值α_i，特徵向量$\boldsymbol{a}_i$ ($i = 1,2,\cdots, N_A$)

B：特徵值β_j，特徵向量$\boldsymbol{b}_j$ ($j = 1,2,\cdots, N_B$)

此時，$C = A \otimes B$之特徵值爲：

$$\alpha_i \beta_j \ (i = 1,2,\cdots, N_A \ , j = 1,2,\cdots, N_B) \tag{5.69}$$

特徵向量爲：

$$\boldsymbol{\alpha}_i \otimes \boldsymbol{\beta}_j \ (i = 1,2,\cdots, N_A \ , j = 1,2,\cdots, N_B) \tag{5.70}$$

(個數分別爲$N_A N_B$個)

由定理 2，我們需要的式子(5.68)之特徵值與特徵向量可分別以 2×2 之C_V與 3×3 之C_H特徵值與特徵向量加以計算，可大幅降低計算特徵值所需之運算量。

因此，在協方差可依式(5.64)加以分離的情況中，實際上可以**同時降低對轉換區塊進行 KLT 時之運算量**。對以此點再以 3×2 之區塊爲例觀之。首先求取式(5.68)之兩個矩陣C_V與C_H之特徵值與特徵向量(分別爲($\boldsymbol{v}_1 \sim \boldsymbol{v}_2$，$\boldsymbol{h}_1 \sim \boldsymbol{h}_3$))。可依式(5.57)得轉換矩陣：

$$V_V = [\boldsymbol{v}\boldsymbol{v}_2] = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix} \tag{5.71}$$

$$V_H = [\boldsymbol{h}_1 \boldsymbol{h}_2 \boldsymbol{h}_3] = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \tag{5.72}$$

此時$C_{xx} = C_V \otimes C_H$之轉換矩陣代入式(5.70)的性質之後，可寫作：

$$V = V_{\mathrm{V}} \otimes V_{\mathrm{H}} \tag{5.73}$$

進行 KLT 時，計算式(5.63)的向量$\boldsymbol{x}$與$V^t\boldsymbol{x}$是相等的，式(5.73)可寫作：

$$V=\begin{bmatrix} v_{11}h_{11} & v_{11}h_{12} & v_{11}h_{13} & v_{12}h_{11} & v_{12}h_{12} & v_{12}h_{13} \\ v_{11}h_{21} & v_{11}h_{22} & v_{11}h_{23} & v_{12}h_{21} & v_{12}h_{22} & v_{12}h_{23} \\ v_{11}h_{31} & v_{11}h_{32} & v_{11}h_{33} & v_{12}h_{31} & v_{12}h_{32} & v_{12}h_{33} \\ v_{21}h_{11} & v_{21}h_{12} & v_{21}h_{13} & v_{22}h_{11} & v_{22}h_{12} & v_{22}h_{13} \\ v_{21}h_{21} & v_{21}h_{22} & v_{21}h_{23} & v_{22}h_{21} & v_{22}h_{22} & v_{22}h_{23} \\ v_{21}h_{31} & v_{21}h_{32} & v_{21}h_{33} & v_{22}h_{31} & v_{22}h_{32} & v_{22}h_{33} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} \tag{5.74}$$

另外，可否不對$\boldsymbol{x}$進行相同的轉換，而是對轉換區塊之原來形式式(5.62)中矩陣X進行運算？答案是 YES，實際上，可依照：

$$Y = V_{\mathrm{V}}{}^t X V_{\mathrm{H}} \tag{5.75}$$

得到與 $\boldsymbol{y} = V^t\boldsymbol{x}$相同的結果[68]。其中，左邊的矩陣$Y$爲具有相同尺寸矩陣的$X$之轉換係數，爲**轉換係數矩陣**。

註　68. 在實際要代入各分量以確認時，即使是在尺寸為 3×2 的情況下，計算時也需要耐性。

$\boldsymbol{y} = V^t\boldsymbol{x}$與式(5.75)，即使是用在計算同一轉換係數，其計算量也有所不同。實際上，著眼於所需進行的乘法次數。相對於前者的 $6\times6 = 36$ 次，後者之式(5.75)爲 $2\times6 + 3\times6 = 30$ 次，式(5.75)較有效率。

由以上討論，在協方矩陣如下所示之分離性成立時，對任意尺寸$N_1 \times N_2$之轉換區塊成立。

分離性

協方矩陣C_{xx}可如式(5.68)，寫作水平方向之協方差矩陣C_{H}與垂直方向之協方差矩陣C_{V}之 Kronecker 積。

再度歸納現有結果。對任意的轉換區塊$N_1 \times N_2$像素依式(5.62)之二維排列的矩陣X，依式(5.63)加以一維化之後，可得向量$\boldsymbol{x}$。此時，如C_{xx}具有上述之分離性，則以下兩性質成立：

① 對$\boldsymbol{x}$進行KLT時，可以獨立求取C_{H}與C_{V}之特徵值、特徵向量，求取特徵向量時可以分別如式(5.57)以並列的轉換矩陣藉由V_{H}與V_{V}計算。

② 對矩陣進行轉換，可寫作$Y = V_{\mathrm{V}}{}^{t} X V_{\mathrm{H}}$。

如此一來，可降低計算量。

計算以每一區塊爲$N_1 \times N_2 = 8 \times 8$之轉換所需的計算次數，當分離性成立是1024次，不成立是4096次，可以確知分離性在實際應用上是有效的。但是，對於實際影像，分離性並非完全成立。但是，如圖 2.12 (a)、(c)所示，因爲實際影像之自協方差函數可以非常地近似於式(2.11)，上述的分離性也可近似地視爲成立。如此一來的好處是，因爲近似造成的損失可以被抵消。

此外，請注意$Y = V_{\mathrm{V}}{}^{t} X V_{\mathrm{H}}$形式之轉換中，各元素可寫作：

$$y_{vu} = \sum_{j=1}^{N_2} \sum_{i=1}^{N_1} x_{ji} \underbrace{h_{ui} v_{vj}}_{\text{常數}} \quad (1 \le v \le N_2，1 \le u \le N_1) \tag{5.76}$$

KLT 之組成例

舉例來說，以圖 2.4(a)之影像爲對象，進行尺寸爲 8×8 之KLT。此處，假設在之前所示之協方差函數之分離性近似成立，僅求水平方向之V_{H}。

首先，以 8×8 之轉換區塊分割影像，實際測量水平方向之協方差矩陣 C_H 如下：

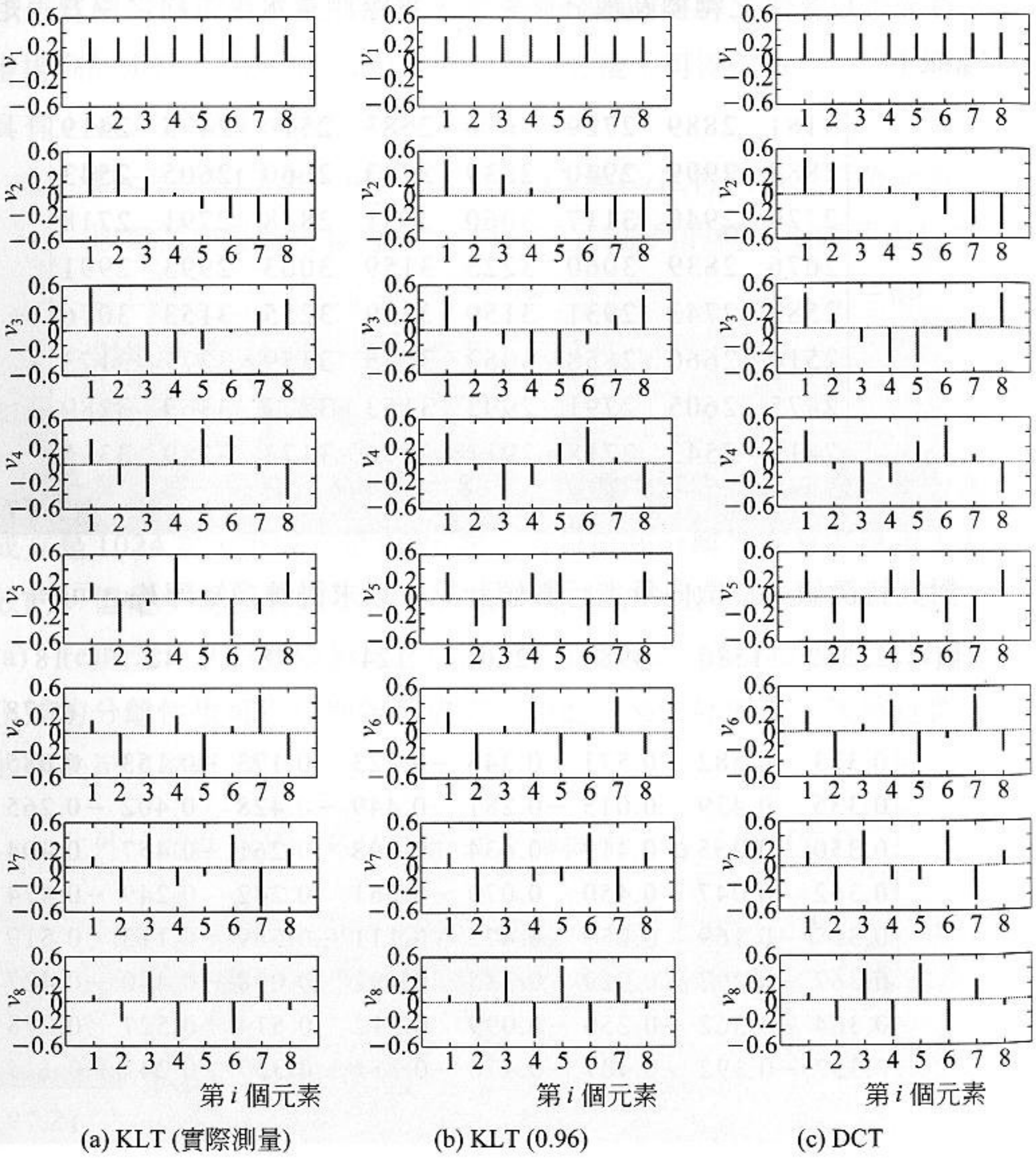
$$C_H=\begin{bmatrix} 3181 & 2889 & 2729 & 2676 & 2583 & 2510 & 2475 & 2419 \\ 2889 & 2999 & 2940 & 2839 & 2743 & 2660 & 2605 & 2543 \\ 2729 & 2940 & 3117 & 3060 & 2931 & 2858 & 2791 & 2718 \\ 2676 & 2839 & 3060 & 3225 & 3159 & 3063 & 2993 & 2991 \\ 2583 & 2743 & 2931 & 3159 & 3309 & 3255 & 3153 & 3076 \\ 2510 & 2660 & 2858 & 3063 & 3255 & 3359 & 3277 & 3173 \\ 2475 & 2605 & 2791 & 2993 & 3153 & 3277 & 3369 & 3289 \\ 2419 & 2543 & 2718 & 2911 & 3076 & 3173 & 3289 & 3385 \end{bmatrix} \tag{5.77}$$

對其特徵值、特徵向量進行數值計算，以求得轉換矩陣 V_H，可得：

$$\text{特徵值}=[23362 \quad 1580 \quad 498 \quad 240 \quad 124 \quad 78 \quad 42 \quad 18] \tag{5.78}$$

$$v_H=\begin{bmatrix} 0.323 & 0.582 & -0.571 & 0.346 & -0.223 & 0.173 & -0.153 & 0.085 \\ 0.335 & 0.439 & 0.015 & -0.281 & 0.449 & -0.428 & 0.402 & -0.265 \\ 0.350 & 0.255 & 0.419 & -0.434 & 0.098 & 0.261 & -0.457 & 0.404 \\ 0.362 & 0.047 & 0.450 & 0.070 & -0.561 & 0.242 & 0.249 & -0.474 \\ 0.367 & -0.169 & 0.259 & 0.472 & -0.111 & -0.509 & 0.115 & 0.519 \\ 0.367 & -0.297 & 0.029 & 0.385 & 0.492 & 0.098 & -0.440 & -0.427 \\ 0.364 & -0.362 & -0.250 & -0.099 & 0.212 & 0.514 & 0.527 & 0.278 \\ 0.357 & -0.392 & -0.407 & -0.476 & -0.354 & -0.357 & -0.247 & -0.114 \end{bmatrix} \tag{5.79}$$

圖 5.7(a)爲式(5.79)之列向量 $\boldsymbol{v}_i$ $(i=1,2,\cdots,8)$ 之圖示。



(a) KLT (實際測量)　(b) KLT (0.96)　(c) DCT

圖 5.7　KLT 與 DVT(a)圖 2.4(a)之水平方向之 KLT(式(5.79))。(b)自協方差模型求得之 KLT(ρ)中，$\rho = 0.96$ 之結果。(c)DCT(式(5.97))

KLT 之性質

任意的正規正交特徵向量$\boldsymbol{v}_i$ ($i=1,2,\cdots,N$)以式(5.57)加以排列，可得N 維之正交轉換矩陣V，其反轉換可寫成：

$$\boldsymbol{x}=V\boldsymbol{y}=\sum_{i=1}^{N} y_i\boldsymbol{v}_i \tag{5.80}$$

現在，右邊的和在中途的第K 項($K<N$)被**打斷**，而被置換爲$i=1\sim K$ 部份的和。對此部份進行反轉換之後所得到的向量，與一般情形中的$\boldsymbol{i}$不同，可表示爲：

$$\hat{\boldsymbol{x}}=\sum_{i=1}^{K} y_i\boldsymbol{v}_i \tag{5.81}$$

進一步地求取$\boldsymbol{x}$與$\hat{\boldsymbol{x}}$的平方誤差，在目標影像內的所有轉換區塊中，平均值爲：

$$e=E_m[|\boldsymbol{x}-\hat{\boldsymbol{x}}|^2] \tag{5.82}$$

先前推導的KLT，實際上**為所有的正交轉換中，對於任意的**K($K=1,2,\cdots,N$)**值而言，使得式(5.82)之誤差**e**為最小之轉換**。其中 KLT 之基底向量$\boldsymbol{v}_i$，如先前所述，依對應之特徵值之遞減而加以排列。在本節當中，詳細地探討其性質。

整理式(5.82)之後，可得[69]：

$$\begin{aligned} e&=E_m[|\boldsymbol{x}-\hat{\boldsymbol{x}}|^2]=E_m\left[\left|\sum_{i=K+1}^{N} y_i\boldsymbol{v}_i\right|^2\right] \\ &=E_m\left[\left(\sum_{i=K+1}^{N} y_i\boldsymbol{v}_i\right)\left(\sum_{i=K+1}^{N} y_i\boldsymbol{v}_i\right)\right]=E_m\left[\sum_{i=K+1}^{N} y_i^2\right] \\ &=\sum_{i=K+1}^{N} E_m[y_i^2] \end{aligned} \tag{5.83}$$

將$y_i = \boldsymbol{v}_i^t \boldsymbol{x}$之寫法代入，可進一步地整理成：

$$e = \sum_{i=K+1}^{N} E_m[(\boldsymbol{v}_i^t \boldsymbol{x})^2] = \sum_{i=K+1}^{N} E_m[(\boldsymbol{v}_i^t \boldsymbol{x}) \cdot (\boldsymbol{v}_i^t \boldsymbol{x})]$$

$$= \sum_{i=K+1}^{N} E_m[\boldsymbol{v}_i^t \boldsymbol{x}\boldsymbol{x}^t \boldsymbol{v}_i] = \sum_{i=K+1}^{N} \boldsymbol{v}_i^t \underbrace{E_m[\boldsymbol{x}\boldsymbol{x}^t]}_{C_{xx}} \boldsymbol{v}_i \tag{5.84}$$

也就是說，使用到第K個爲止的轉換係數之部份復元後的平均平方差，可以用處理對象之影像f之共協方矩陣寫作：

$$e = \boldsymbol{v}_{K+1}{}^t C_{xx} \boldsymbol{v}_{K+1} + \boldsymbol{v}_{K+2}{}^t C_{xx} \boldsymbol{v}_{K+2} + \cdots + \boldsymbol{v}_N{}^t C_{xx} \boldsymbol{v}_N \tag{5.85}$$

註 69. 在此處的化簡用到$\boldsymbol{v}_i$之正交性。

首先，先以圖例觀察協方差矩陣具有的性質與其特徵值的關係。

協方差矩陣與其特徵值、特徵向量

現在，特別將$K = N-1$代入式(5.85)中，可得：

$$e = \boldsymbol{v}_N{}^t C_{xx} \boldsymbol{v}_N \tag{5.86}$$

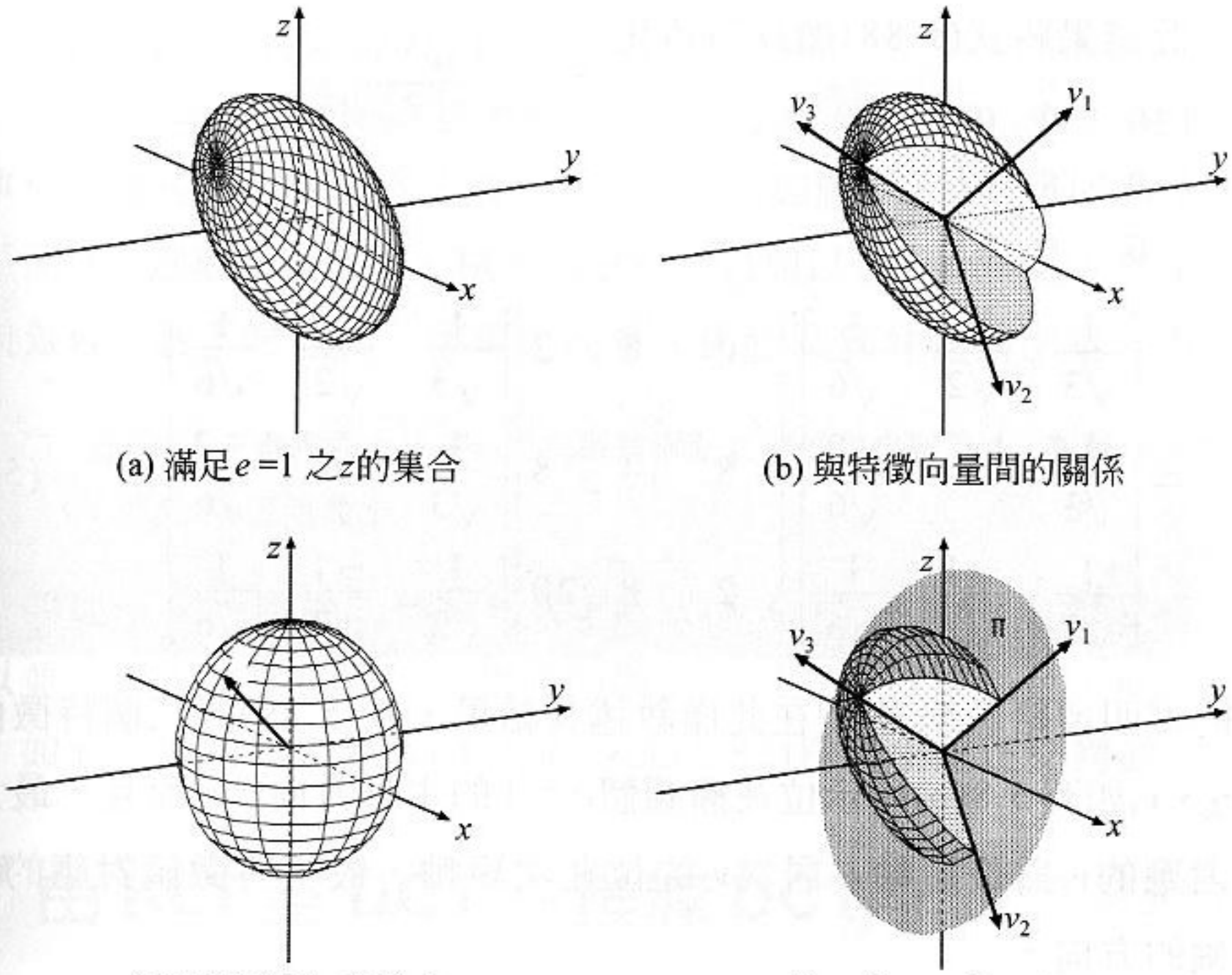
不管以V進行何種正交轉換，其誤差e**絕對不會為負值**，因此用N維的向量$\boldsymbol{z}$作成$\boldsymbol{z}^t C_{xx} \boldsymbol{z}$，只要$\boldsymbol{z} \neq 0$，可得到以下結論：

$$e = \boldsymbol{z}^t C_{xx} \boldsymbol{z} > 0 \tag{5.87}$$

具有這種性質的矩陣稱爲**正定值矩陣**[34][37]。

在式(5.87)中，滿足$e = 1$的N維向量$\boldsymbol{z}$的集合(等向量平面)，構成N維的超橢圓體[39]。在此以三維中的例子加以討論。舉例來說，考慮正定值矩陣：

$$C_{xx} = \begin{bmatrix} 20 & 8 & 2 \\ 8 & 14 & 8 \\ 2 & 8 & 20 \end{bmatrix} \tag{5.88}$$



(a) 滿足$e=1$ 之z的集合　(b) 與特徵向量間的關係

(c) 單位向量v的集合　(d) 第二向量v_2的選擇方法

圖 5.8　式(5.88)的圖形性質

在此情形下滿足$e=1$的集合以實際的圖加以表示。爲如圖 5.8(a)之「傾斜的」橢圓體。

首先，計算式(5.88)之特徵值與特徵向量：

$$
\begin{array}{lccc}
\text{特徵值} & \lambda_1 = 30 & \lambda_2 = 18 & \lambda_3 = 6 \\
 & \downarrow & \downarrow & \downarrow \\
\text{特徵向量} & \boldsymbol{v}_1 & \boldsymbol{v}_2 & \boldsymbol{v}_3 \\
 & \begin{bmatrix} \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \end{bmatrix} & \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{-1}{\sqrt{2}} \end{bmatrix} & \begin{bmatrix} \frac{1}{\sqrt{6}} \\ \frac{-2}{\sqrt{6}} \\ \frac{1}{\sqrt{6}} \end{bmatrix}
\end{array}
\tag{5.89}
$$

再利用此結果將式(5.88)加以對角化。

$$\begin{bmatrix} 30 & 0 & 0 \\ 0 & 18 & 0 \\ 0 & 0 & 6 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & 0 & \frac{-2}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \end{bmatrix}^t \begin{bmatrix} 20 & 8 & 2 \\ 8 & 14 & 8 \\ 2 & 8 & 20 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & 0 & \frac{-2}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \end{bmatrix} \tag{5.90}$$

詳細的證明留給數學書[39]在此僅敘述其結果，式(5.89)的三個特徵向量$\boldsymbol{v}_1$、$\boldsymbol{v}_2$、$\boldsymbol{v}_3$如圖 5.8(b)所示位在橢圓體$e=1$的主軸方向[70]。而且，最大特徵值對應的$\boldsymbol{v}_1$爲最短軸，居次$\boldsymbol{v}_2$的位在次短軸，最小特徵值對應的$\boldsymbol{v}_3$對應長軸的方向。

註 70. 可以簡單地看得出來為橢圓體切一部份下來。

一般的N 維正定值矩陣，e爲定值時其等高面爲超橢圓體，雖然無法加以繪出，但其特徵向量仍指向對應之超橢圓體的主軸方向。

KLT 為對於任意K值使式(5.85)為最小之轉換方式之理由

利用特徵向量具有以上性質，以組成之方法說明其理由。爲此，再度利用在$N=3$時之式(5.88)的矩陣。

首先，求取使式(5.86)之e爲最小之單位向量$\boldsymbol{v}_N=\boldsymbol{v}_3$。全部的單位向量$\boldsymbol{v}$的集合對應至具有單位半徑的球面，式(5.86)相當於對圖 5.8(c)所示的單位球面上之向量進行e的評估。由圖 5.8(b)可以簡單地了解，使其爲最小時之單位向量$\boldsymbol{v}_3$，爲對C_{xx}有最小特徵值之特徵向量$\boldsymbol{v}_3$ [71]。藉此可決定$\boldsymbol{v}_3$。如此一來，在式(5.85)中取$K=1$，選擇使

$$e = \boldsymbol{v}_2{}^t C_{xx} \boldsymbol{v}_2 + \underbrace{\boldsymbol{v}_3{}^t C_{xx} \boldsymbol{v}_3}_{\text{由 } \boldsymbol{v}_3 \text{加以最小化的常數}} \tag{5.91}$$

的單位向量$\boldsymbol{v}_2$。其中，爲了滿足正交性，$\boldsymbol{v}_2$如圖 5.8(d)所示需在與$\boldsymbol{v}_3$垂直之平面Π上選取。$\boldsymbol{v}_3$也是一樣。將式(5.91)加以最小化的$\boldsymbol{v}_2$，爲第二個特徵向量$\boldsymbol{v}_2$。剩下的最後一個單位向量，由正交條件自動決定，即爲$\boldsymbol{v}_1$。

註　71. 在$e = 1$之等高面上，在$\boldsymbol{v}_3$方向取離原點最遠的點之值為 1。在單位球面上，在$\boldsymbol{v}_3$方向的e值為最小。

如此一來，所得的轉換矩陣V爲式(5.88)之共協方矩陣之特徵向量加以並列，相當於KLT。由相同的觀念，以上可以擴張至任意維度的空間，如此一來可以得到KLT爲使得式(5.82)爲**最小之正交轉換**。

由 KLT 至 DCT－推導 DCT

如前一節所討論的，KLT雖可以使得平均平方差最小化而爲最佳的正交轉換，另外一方面，由其推導過程可以知道，KLT與其轉換對象之影像**具有相依賴的關係**，此爲相當麻煩的問題。因此，KLT會產生以下兩個缺點：

- 必須對所給的每個影像各別進行(KLT)，使得計算量增大。
- 爲了可解碼，編碼端使用的KLT轉換矩陣需要傳送至解碼端，編碼的額外處理時間相當大。

因此，對於所有影像，如果可以推導出**保持 KLT 的最佳性質之共通的正交轉換**，在編碼端與解碼端的共通點爲：

- 不需要對個別影像的轉換矩陣再重行計算。
- 不需要將轉換矩陣傳送至解碼端，使得編碼時之額外處理時間得以減少。

在本節中，依此嘗試KLT之「共通化」，推導出其結果之DCT。

自協方差矩陣之模式化與其 KLT

KLT 係由式(5.52)之C_{xx}加以推導。其基本性質可以**全歸納為式(5.52)**。在此，再次討論C_{xx}的性質。

在前一節中所使用式(5.64)的假設，協方差矩陣除了具有分離性之外，事實上還可以進行協方差矩陣的模式化。實際上，在式(5.64)之假設下，任意尺寸之轉換區塊中的水平或是垂直方向之協方差矩陣可寫作：

$$C_{xx\text{model}}=\begin{bmatrix} \rho & \rho^2 & \cdots & \rho^{N-1} \\ \rho^2 & \rho & \cdots & \rho^{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ \rho^{N-1} & \rho^{N-2} & \cdots & \rho \end{bmatrix} \tag{5.92}$$

在此，由式(5.92)之協方差矩陣模型構成KLT。所得到的KLT，僅由式(5.92)中的相關係數ρ所決定，寫作 KLT (ρ)。式(5.92)中具有特殊的形狀使得其第ij 元素之值僅與$|i-j|$相關[72]。與該協方差矩陣相對的KLT，實際上可以取其閉合形式[40]，其基底元素v_{ij}爲：

$$v_{ij}=\sqrt{\frac{2}{N+\lambda_i}}\sin\left\{\frac{(2i-N-1)\omega_j}{2}+\frac{j\pi}{2}\right\} \tag{5.93}$$

此處，λ_j爲式(5.92)之特徵值，可以下列方式求得：

$$\lambda_j=\frac{1-\rho^2}{1-2\rho\cos\omega_j+\rho^2} \tag{5.94}$$

其中，ω_j爲下式的正根：

註 72. 這種矩陣特別稱為 Toeplitz 矩陣。

$$\tan(N\omega_j) = -\frac{(1-\rho^2)\sin\omega_j}{\cos\omega_j - 2\rho + \rho^2\cos\omega_j} \tag{5.95}$$

舉例來說，當$\rho = 0.96$時KLT(0.96)如圖 5.7(b)所示，另外一方面，由圖 2.12(a)與(c)可以將圖 2.4(a)的影像以$\rho \cong 0.96$加以近似，可以將圖 5.7(b)視爲圖 2.4(a)之影像由自協方差的近似模型加以推導之 KLT。事實上，基於自協方差之實際測量結果圖 5.7(a)與基於模型之同一圖(b)，可以確認其形狀是接近的。

KLT 之極限的 DCT[41]

由協方差矩陣之模型(5.92)求得之 KLT (ρ)，雖然僅由ρ加以決定，但其對影像的依賴性仍然存在。另外，如第三十七頁所述，當影像的相關係數在 0.90～0.98 程度之非常接近 1 的值時，可讓$\rho \to 1$以取 KLT (ρ)之極限狀態，可由此處得到靈感，藉此：

- 完全消除對影像之依賴性，得到所有影像通用的正交轉換[73]。
- 失去 KLT 之最佳性質，但是當影像之相關係數原本即非常接近 1 時，取$\rho \to 1$之極限，也可以保持近似的最佳性質(準最佳性質)。
- 幾乎繼承了 KLT 可以有效地去除像素間的相關程度之性質。

註 73. 必須是在像素間的相關程度較高的制約條件下。

$\tan(N\omega_j) = 0$

在此，式(5.93)～(5.95)中，使$\rho \to 1$，此時式(5.95)爲~~$(N\omega) = 0$~~，可得解爲$\omega_j = \pi(j-1)/N$ ($j = 1, 2, \cdots, N$)。代入式(5.94)再使$\rho \to 1$，當 $j = 1$ 的情形中，λ_1會發散至無限大。因此，在式(5.92)中，以$\rho \to 1$直接對矩陣求取特徵值與特徵向量。實際上，在式(5.92)中，當時，所有的元素皆爲 1，其固有值經計算爲：

$$\lambda_j = \begin{cases} N & (j=1) \\ 0 & (j \neq 1) \end{cases} \tag{5.96}$$

將式(5.96)與$\omega_j = \pi(j-1)/N$($j=$ 1,2,⋯,N)一同代入式(5.93)，$\mathrm{KLT}|_{\rho \to 1}$經計算爲($i=$ 1,2,⋯, N)：

$$v_{ij} = \begin{cases} \sqrt{\dfrac{1}{N}} \cos \dfrac{(2i-1)(j-1)\pi}{2N} & (j=1) \\ \sqrt{\dfrac{2}{N}} \cos \dfrac{(2i-1)(j-1)\pi}{2N} & (j \neq 1) \end{cases} \tag{5.97}$$

式(5.97)之所有基底元素係由離弦(cosine)表示，與其對應之轉換爲離散餘弦轉換(Discrete Cosine Transform；DCT)。在$N=$ 8 的情形中，其基底如圖 5.7(c)所示。可以看得出來圖 5.7 的三種基底相當地接近。

DCT 的二維化－化為標準形式

在之前所推導的一維DCT。可以擴張至使用式(5.76)之二維DCT，在此，轉換區塊的尺寸爲$N \times N$之平方。另外，DCT之極限爲，在水平、垂直方向之轉換矩陣之基底要素可以式(5.97)求得。

現在，使用：

$$C(n) = \begin{cases} 1/\sqrt{2} & (n=0) \\ 1 & (n \neq 0) \end{cases} \tag{5.98}$$

將式(5.97)因 j 值不同而造成的差別加以統一，再代入式(5.76)中，可以寫作：

$$y_{vu} = \sum_{j=1}^{N} \sum_{i=1}^{N} x_{ji} \frac{2C(u)C(v)}{N} \cos\left(\frac{(2j-1)(v-1)\pi}{2N}\right) \cdot \cos\left(\frac{(2i-1)(u-1)\pi}{2N}\right) \quad (1 \leq u，v \leq N) \tag{5.99}$$

另外，在JPEG或是MPEG的說明書，以DCT進行轉換編碼之處理文件中，以下情形較多：

- 像素以 $f(i,j)$，其轉換係數以$F(u,v)$表示。
- 其指數不爲 $1\sim N$，而爲 $0\sim(N-1)$。

因此，本書亦依照此慣例，標準形式之二維 DCT 與反 DCT 寫成以下形式[74]：

$$F(u,v)=\frac{2C(u)C(v)}{N}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}f(i,j)\cdot\cos\frac{(2i+1)u\pi}{2N}\cos\frac{2j+1)v\pi}{2N} \tag{5.100}$$

$$f(i,j)=\frac{2}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}C(u)C(v)F(u,v)\cdot\cos\frac{(2i+1)u\pi}{2N}\cos\frac{(2j+1)v\pi}{2N} \tag{5.101}$$

$$0\le i,j,u,v\le N-1,\quad C(n)=\begin{cases}1/\sqrt{2} & (n=0)\\ 1 & (n\ne 0)\end{cases}$$

在實際的應用上，該式中的$N=8$時可用在 8×8 點之二維 DCT，爲其一般例。

註　74. 對總和符號為常數之因數，移項至總和的前方。

結　語

DCT與反DCT寫作式(5.100)～(5.101)之形式，看起來雖然會給人如本章最初所說的，好像很複雜的轉換處理之印象，實際上要再度強調「DCT爲正交轉換」。如此一來，如第八十一頁所述，DCT僅爲$f(i,j)$

在N維空間中進行「**旋轉**」。在四維以上的場合，特別是八維中很可惜地無法以圖示加以表示…

6章

DCT 之性質與其在編碼上的應用

在前一章所推導的DCT具有準最佳的正交編碼的地位，將此適用於圖 5.5 中，可以實現高效率的編碼。在本章中，首先討論幾個 DCT 的性質之後，再概略觀察其應用於在 JPEG 或是 MPEG 中編碼處理之基本組成。

DCT 的性質

二維 DCT 之基底元素

將二維 DCT 轉換之基底元素改寫爲：

$$w_{ij}^{uv}=\frac{2C(u)C(v)}{N}\cdot\cos\frac{(2i+1)u\pi}{2N}\cdot\cos\frac{(2j+1)v\pi}{2N} \tag{6.1}$$

另外，之後的 DCT 的區塊尺寸限定在：$N\times N=8\times 8$[75]。舉例來說，在$u=4$、$v=6$ 的情形中基底要素$\omega_{ij}^{46}(0\le i，i\le 7)$如圖 6.1 所示。同一圖

(a)係ω_{ij}^{46}爲矩陣$W^{46}=[\omega_{ji}^{46}]$，(b)表示其亮度[76]。圖 6.2 爲完全一樣的$W^{uv}=[\omega_{ji}^{uv}](0 \le u，v \le 7)$之所有基底加以並排以表示其亮度之結果。

註 75. DCT 之區塊大小留待本章最後再行討論。
76. 以 0 表示第 128 階。

在圖 6.2 中，與$u=v=0$對應的左上方基底爲 DC 基底，另外，該對應轉換係數稱爲DC係數，其他的 63 個統稱爲AC基底與AC係數[77]。AC 基底隨著u、v值的增大縱向橫向條紋的頻率會升高。另外，式(6.1)中，代入$u=v=0$後，DC$u_{ij}^{00}(0 \le i，j \le 7)$基底的元素 ，全部會成爲同一數值 $1/N$ (此現象如同『DC』之簡稱)[78]。

註 77. DC 與 AC 分別為電學中表示直流與交流的簡稱。
78. 另外，在圖 5.7 最上端的位置可以看出來，在 KLT 最初的基底亦幾乎為 DC 的形狀，但是在這個情形下，所有元素的值並不完全相同。

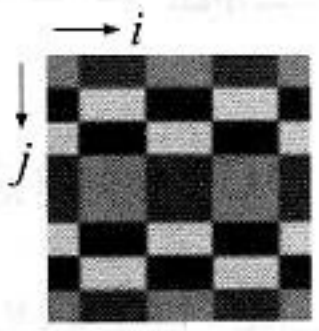
在此，以下列方式定義對於$N \times N$之方陣$A=[a_{ij}]$與$B=[b_{ij}]$進行之運算⊙：

$$A \odot B = \sum_{i,j=0}^{N-1} a_{ij} b_{ij} \tag{6.2}$$

此爲二方陣之所有對應元素間的乘積和，可以視爲**向量內積之矩陣版本**[79]。利用這種記法，式(6.1)之基底間的正規正交性可以寫爲：

$$\begin{array}{c} \longrightarrow i \\ \downarrow j \end{array} \begin{bmatrix} +0.07 & -0.07 & -0.07 & +0.07 & +0.07 & -0.07 & -0.07 & +0.07 \\ -0.16 & +0.16 & +0.16 & -0.16 & -0.16 & +0.16 & +0.16 & -0.16 \\ +0.16 & -0.16 & -0.16 & +0.16 & +0.16 & -0.16 & -0.16 & +0.16 \\ -0.07 & +0.07 & +0.07 & -0.07 & -0.07 & +0.07 & +0.07 & -0.07 \\ -0.07 & +0.07 & +0.07 & -0.07 & -0.07 & +0.07 & +0.07 & -0.07 \\ +0.16 & -0.16 & -0.16 & +0.16 & +0.16 & -0.16 & -0.16 & +0.16 \\ -0.16 & +0.16 & +0.16 & -0.16 & -0.16 & +0.16 & +0.16 & -0.16 \\ +0.07 & -0.07 & -0.07 & +0.07 & +0.07 & -0.07 & -0.07 & +0.07 \end{bmatrix}$$

(a) 以數值表示



(b) 以亮度表示

圖 6.1 $u=4$，$v=6$ 的情形中，二維 DCT 的基底$W^{uv}=[\omega_{ji}^{uv}](0 \le u，v \le 7)$

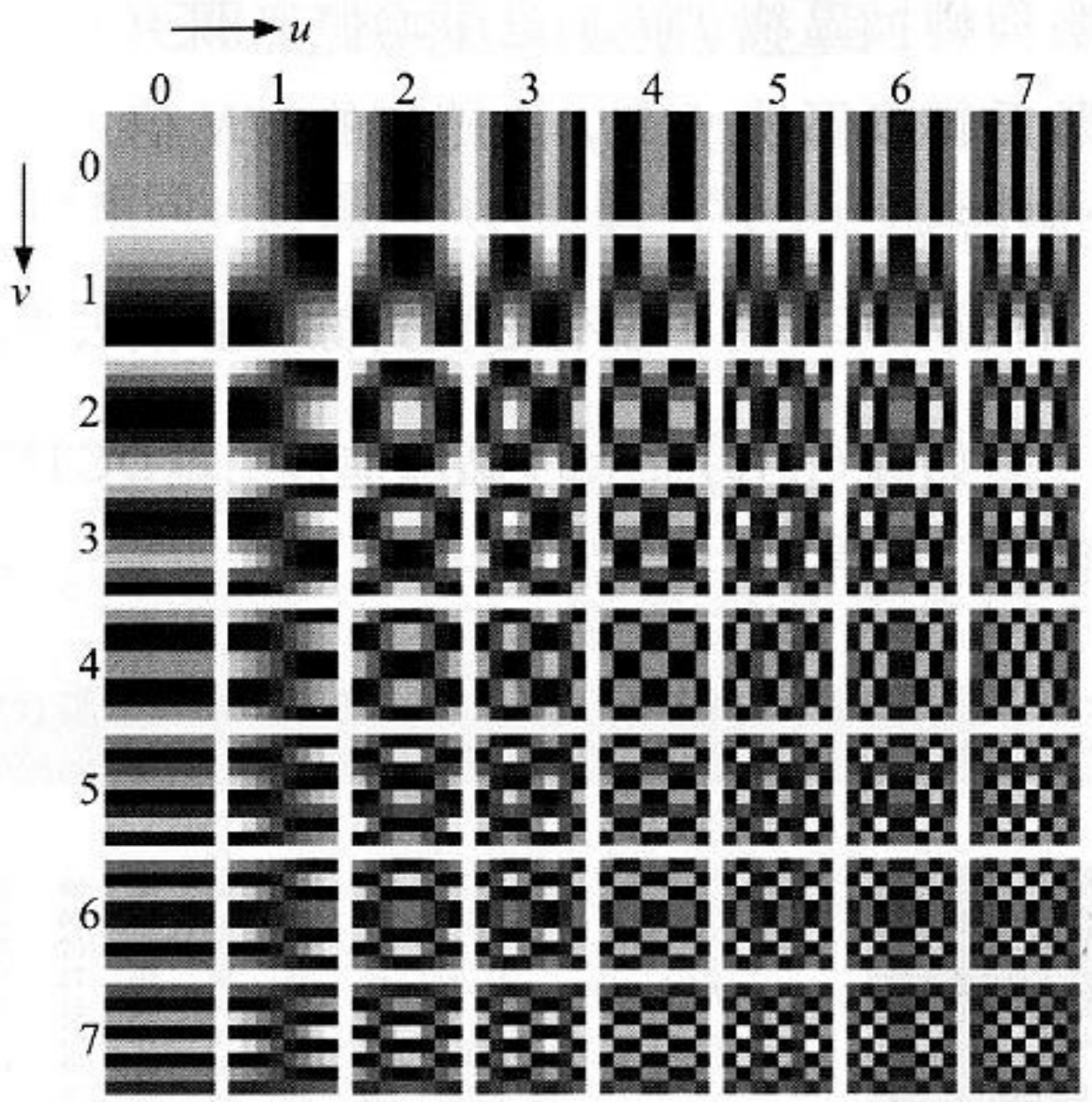


圖 6.2　二維 DCT 的所有基底

$$W^{uv} \odot W^{u'v'} = \begin{cases} 1 & (u = u',\ v = v') \\ 0 & (\text{其他}) \end{cases} \tag{6.3}$$

如先前所述，DC 基底全爲相同值 $1/N$，DC 基底與任意的 AC 基底可以適用式(6.3)而爲：

$$0 = W^{00} \odot W^{uv} = \sum_{i,j=0}^{N-1} w_{ij}^{00} w_{ij}^{uv} = \frac{1}{N} \sum_{i,j=0}^{N-1} w_{ij}^{uv} \tag{6.4}$$

由此可知**對任意的 AC 基底所有元素之平均值**爲 0。

註　79. 可以用矩陣的軌跡寫成：$A \odot B = \mathrm{tr}(AB^t)$，不需要特別定義新的運算⊙，以與向量內積相似的記號加以表記即可，在本書中依此導入該計算方法。

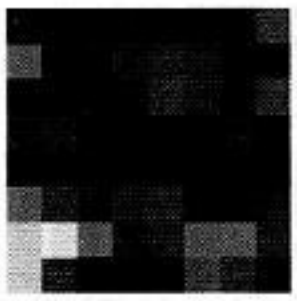
如第 5-17 頁所述，KLT 爲對像素值減去平均值之後所導出，包含平均值之原像素值 $f(i,j)$進行KLT也無法去除其相關性。另外，在DCT的情形中，由式(6.4)可知：

- AC 係數與轉換區塊 $f(i,j)$之平均值無關。
- DC 係數爲轉換區塊 $f(i,j)$之總和的 $1/N$ 倍，亦即轉換區塊平均值之N 倍。

因此，結論是當使用 DCT 時，**沒有**必要對像素值減去平均值。

舉例來說，對如圖 6.3(a)所示的 區塊進行二維DCT[80]。在同一圖(c)中參照圖 6.2 的基底可得：

$$f(i,j)=\sum_{u=0}^{7}\sum_{v=0}^{7}F(u,v)w_{ij}^{uv} \tag{6.5}$$



(a) 處理對象之區塊

$$f(i,j)=\begin{bmatrix} 48 & 39 & 40 & 68 & 60 & 38 & 50 & 131 \\ 149 & 82 & 79 & 101 & 113 & 106 & 27 & 62 \\ 58 & 63 & 77 & 69 & 124 & 107 & 74 & 125 \\ 80 & 97 & 74 & 54 & 59 & 71 & 91 & 66 \\ 18 & 34 & 33 & 46 & 64 & 61 & 32 & 37 \\ 149 & 108 & 80 & 106 & 116 & 61 & 73 & 92 \\ 211 & 233 & 159 & 88 & 107 & 158 & 161 & 109 \\ 212 & 104 & 40 & 44 & 71 & 136 & 113 & 66 \end{bmatrix}$$

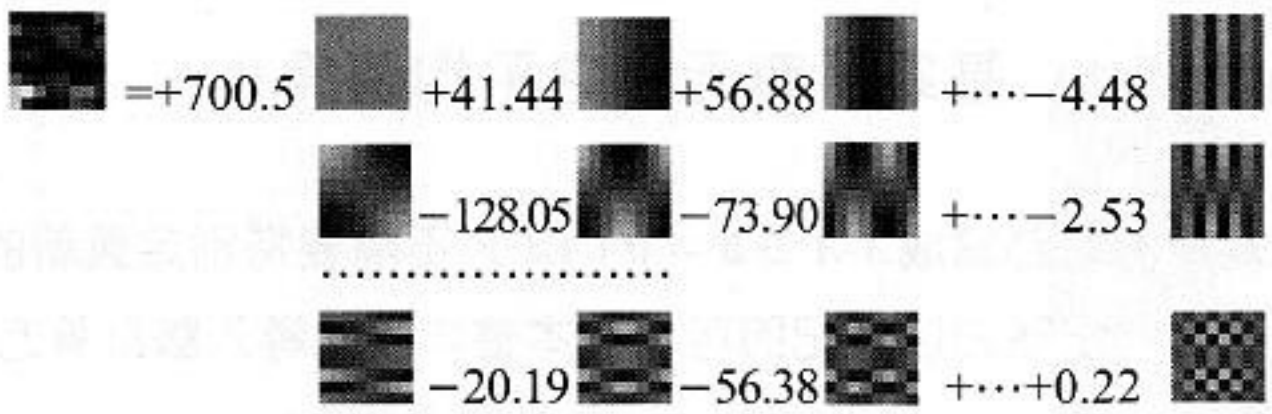
(b) (a)之亮度值

$$\begin{bmatrix} 700.50 & 41.44 & 56.88 & 70.64 & 25.25 & -26.49 & 11.89 & -4.48 \\ -128.05 & -73.90 & -67.99 & -75.39 & 61.17 & -25.38 & 23.55 & -2.53 \\ 87.34 & 28.06 & 63.91 & 42.95 & 16.48 & 16.06 & 16.39 & -13.64 \\ -39.34 & 8.13 & -15.61 & -57.53 & 31.97 & -3.43 & -20.21 & -1.66 \\ -156.25 & -51.13 & 14.90 & -3.25 & -7.50 & 21.49 & -7.80 & -9.58 \\ 93.47 & -10.39 & 47.00 & -49.29 & -57.52 & -9.78 & -28.01 & 10.11 \\ -52.41 & -63.91 & -2.61 & -20.42 & 56.77 & -2.78 & -2.91 & 11.73 \\ -20.19 & -56.38 & -20.14 & -18.60 & -26.23 & -27.34 & 8.66 & 0.22 \end{bmatrix}$$

(c) (a)之 DCT 係數

圖 6.3　DCT 之計算例 1

或是由圖例：



再度確認。

註　80. 圖 6.3～6.5(a)都是由圖 2.2(a)所擷取出來的。

二維 DCT 之轉換係數的性質

與圖 6.3 完全相同，對**圖 6.4(a)**與**圖 6.5(a)**中各區塊進行二維DCT，並相互比較。

(a) 處理對象之區塊

237	238	237	237	236	235	235	236
236	237	237	236	236	236	236	237
236	237	237	237	236	238	237	236
237	238	237	236	237	236	235	236
237	238	237	236	237	236	235	236
237	237	238	237	237	237	237	236
238	237	237	237	237	236	236	236
237	237	237	236	237	236	236	235
238	238	238	236	237	238	237	236

(b) (a)之亮度值

1893.38	3.41	−0.30	−0.11	−1.13	−0.89	0.26	0.35
−1.35	−0.44	0.23	−1.70	0.35	−0.43	−0.34	−1.10
−0.24	0.52	0.64	−0.16	−0.16	−0.17	−0.09	−0.03
−0.64	1.18	0.02	−0.72	1.58	0.39	−0.56	0.45
0.87	1.10	0.16	−0.09	−0.12	−0.58	0.07	0.14
−1.28	1.27	−0.27	0.14	0.58	−0.63	−0.58	−0.10
1.05	0.06	−0.09	0.44	−0.07	0.69	0.11	−0.79
0.45	−0.85	−0.86	0.21	−1.06	1.08	0.03	−0.21

(c) (a)之 DCT 係數

圖 6.4　DCT 之計算例 2

(a) 處理對象之區塊

167	173	156	132	163	173	170	170
167	174	153	125	159	170	167	169
168	174	152	133	168	172	166	168
169	172	147	135	168	171	167	167
170	173	147	136	167	170	170	171
169	172	140	136	171	171	169	167
170	174	144	142	173	171	168	169
173	175	137	135	170	171	170	170

(b) (a)之亮度值

1299.50	−21.87	60.31	58.40	−20.25	−52.35	−11.18	13.72
−3.14	2.58	3.12	−9.67	−13.93	−0.01	11.79	5.45
0.75	−0.49	3.30	−0.23	−2.54	−0.51	0.47	−0.43
1.12	−2.13	−1.29	−1.65	−0.41	2.50	−0.30	0.07
1.75	−1.30	2.27	0.30	0.50	2.02	−0.40	−0.52
4.01	0.84	−7.56	−0.39	2.76	0.88	−0.20	−0.65
0.50	−1.17	−1.53	3.30	1.05	−0.72	−1.30	0.03
5.02	0.56	−1.72	−3.46	0.52	0.42	0.97	0.18

(c) (a)之 DCT 係數

圖 6.5　DCT 之計算例 3

首先，這三個例的共通點爲，得到的DC係數爲非常大的值，另外，一般的 AC 係數之絕對值與 DC 係數相較則小得多。特別是如圖 6.4~5(a)之較單調區塊的情形中，整體的 AC 係數都是非常小的值。另外，在如

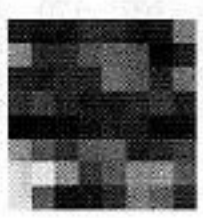
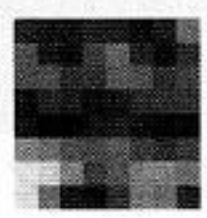
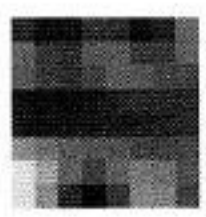
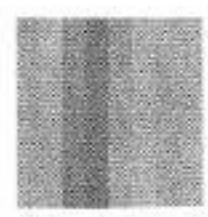
圖 6.3(a)包含較複雜的圖案之區塊中，AC 係數是較大的。另外，在如圖 6.5(a)之鋸齒狀的圖案中，與鋸齒方向正交的方向之 AC 係數具有較大的值。這些觀測結果，可以視爲以二維DCT與給定轉換區塊與圖 6.2 的基底圖案的「內積⊙」加以考慮。

KLT 爲在部份反轉換(5.81)時產生的誤差是最小的轉換方法，而DCT爲其近似版本，因此DCT也具有相同的性質。也就是說，具有「部份的反轉換可以解碼與原影像十分接近的區塊」之性質。實際上，使用圖 6.3～圖 6.5(c)的係數之部份反轉換：

$$f(i,j)=\sum_{u=0}^{K}\sum_{v=0}^{K}F(u,v)w_{ij}^{uv} \quad (1 \le K \le 7) \tag{6.6}$$

加以計算之結果爲表 6.1，可以確認 DCT 亦具有相同的性質。

表 6.1　對圖 6.3～圖 6.5 進行式(6.6)之部份反轉換的結果

	$K=7$ (原影像)	$K=6$	$K=5$	$K=4$
圖 6.3				
圖 6.4				
圖 6.5				
比值$(K+1)^2/64$	1.0	0.77	0.56	0.39

(a) 原影像

(b) DCT $K = 6$
($e = 76.5$, PSNR = 29.30dB)

(c) KLT $K = 6$
($e = 75.5$, PSNR = 29.35dB)

(d) DCT $K = 5$
($e = 178.7$, PSNR = 25.61dB)

(e) KLT $K = 5$
($e = 176.0$, PSNR = 25.68dB)

(f) DCT $K = 4$
($e = 315.8$, PSNR = 23.14dB)

(g) KLT $K = 4$
($e = 309.3$, PSNR = 23.23dB)

圖 6.6　DCT 與 KLT 之部份解碼結果之比較

另外，圖 6.6 爲對 2.2(a)之影像進行二維 KLT 與二維 DCT，在式(6.6)$K=4$、5、6的部份解碼的影像。另外，與原影像的平均平方差e與PSNR(dB)之值。對於DCT之平均平方差，與具有最小值之KLT相較僅多了幾個百分比，這表示DCT與KLT可以說是相當接近的。

編碼時的應用－使用二維 DCT 進行影像編碼的實際情形

JPEG 或是 MPEG 等代表性的編碼方法，基本上可用 DCT 以圖 5.5 的演算法加以實施。在本節中，以JPEG或MPEG爲基礎，簡介其編碼方式。

彩色影像之處理

彩色影像，如第十九頁所述，輸入的原RGB訊號一般轉換爲亮度Y與色差C_b、C_r三個分量。藉由該轉換，三個分量在統計上幾乎爲獨立的，因此可對各分量分別編碼。也就是說，可以對Y、C_b、C_r**獨立地進行DCT、量子化、熵編碼**。另外，在 4:2:2、4:2:0 格式對色差訊號C_b、C_r進行減樣時也是一樣，一般對於縮減後C_b、C_r的與Y進行相同尺寸的DCT。如此一來，舉例來說圖 1.19(c)的 4:2:0 格式之情形中，對於原影像之 16×16 區塊，對應於 4 個Y之DCT區塊，與C_b、C_r之各一個DCT區塊，整體構成一個基本區塊(圖 6.7)。

如上所述，彩色影像之編碼，基本上可以等同於三種單色影像的編碼，在本節以對亮度Y進行編碼爲中心加以闡明。

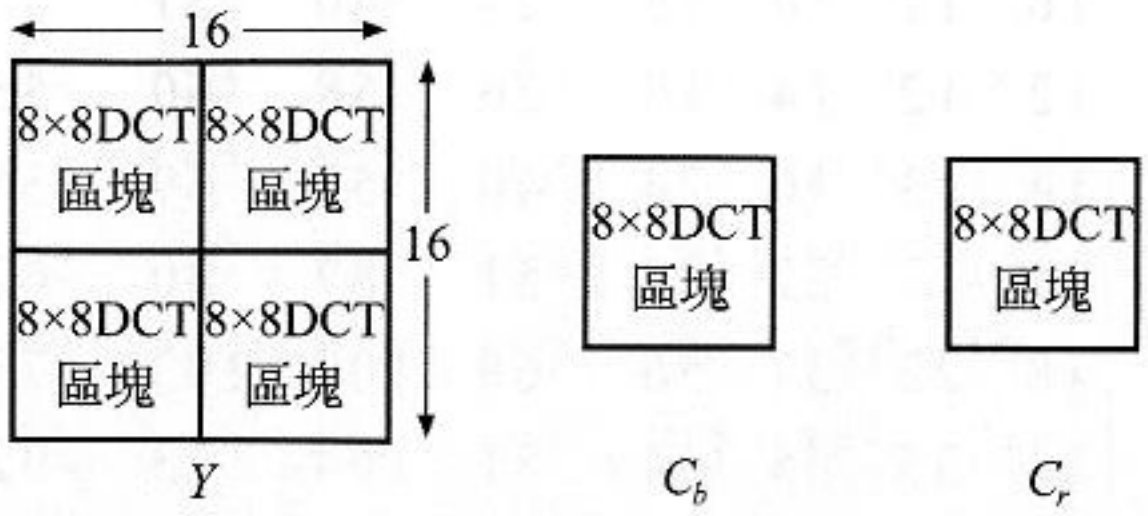


圖 6.7　4:2:0 格式中之 DCT 區塊

DCT 係數之量子化

在前一節中進行的僅對使用係數 0～K進行之部份解碼，雖然等同於在$(K+1)$～8 的各係數之位元長度爲 0 進行量子化，但是由最佳量子化的觀點來看，這種處理仍屬不適當。也就是說，對於DCT係數來說，要使量子化誤差爲最小，必需對處理對象之影像**個別**計算式(5.40)中所提供的最佳位元分配，但是，在這種情形當中：

- 需以影像爲單位進行 64 個量子化器的位元分配，其計算量相當可觀。
- 爲了可以在解碼端進行反量子化，所計算的 64 個位元分配或是量子化的步進幅度需傳送至解碼端，因此會增加資料量。

剛好會產生與 KLT 之缺點相同的問題。因此，在實際應用上，相對於DCT係數$F(u,v)$之步進幅度係常使用一固定的**量子化表**$Q(u,v)$(量子化矩陣、Q 表)舉例來說，使用如同下列的表格：

$$Q(u,v)=\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \tag{6.7}$$

如此一來，DCT 係數$F(u,v)$可藉：

$$F'(u,v)=\left\lfloor \frac{F(u,v)}{Q(u,v)}+0.5 \right\rfloor \quad (0 \le u，v \le 7) \tag{6.8}$$

進行一致量子化。在此，因爲選擇較小的(較大的)量子化表元素會減少(增加)量子化誤差，所以會提升(降低)解碼的影像的畫質。也就是說，畫質與資料量係由量子化表所控制。在實際應用上，常使用一參數q，如式(6.7)將整個表中的所有元素通通乘上q倍加以控制(如此一來，當q增大(減少)時，畫質會變差(變好))，另外，由進行$F'(u,v)$反量子化時，也可以利用量子化表進行以下運算：

$$\overline{F}(u,v)=F'(u,v)\cdot Q(u,v) \tag{6.9}$$

舉例來說，對圖 6.5(a)之轉換區塊$f(i,j)$進行DCT後，由式(6.7)～(6.9)進行量子化，反量子化，其結果爲：

$$f(i,j)=\begin{bmatrix} 167 & 173 & 156 & 132 & 163 & 173 & 170 & 170 \\ 167 & 174 & 153 & 125 & 159 & 170 & 167 & 167 \\ 168 & 174 & 152 & 133 & 168 & 172 & 166 & 168 \\ 169 & 172 & 147 & 135 & 168 & 171 & 167 & 167 \\ 170 & 173 & 147 & 136 & 167 & 170 & 170 & 171 \\ 169 & 172 & 140 & 136 & 171 & 171 & 169 & 167 \\ 170 & 174 & 144 & 142 & 173 & 171 & 168 & 196 \\ 173 & 175 & 137 & 135 & 170 & 171 & 170 & 170 \end{bmatrix} \tag{6.10}$$

$$F(u,v)$$

$$=\begin{bmatrix} 1299.50 & -21.87 & 60.31 & 58.40 & -20.25 & -52.35 & -11.18 & 13.72 \\ -3.14 & 2.58 & 3.12 & -9.67 & -13.93 & -0.01 & 11.79 & 5.45 \\ 0.75 & -0.49 & 3.30 & -0.23 & -2.54 & -0.51 & 0.47 & -0.43 \\ 1.12 & -2.13 & -1.29 & -1.65 & -0.41 & 2.50 & -0.30 & 0.07 \\ 1.75 & -1.30 & 2.27 & 0.30 & 0.50 & 2.02 & -0.40 & -0.52 \\ 4.01 & 0.84 & -7.56 & -0.39 & 2.76 & 0.88 & -0.20 & -0.65 \\ 0.50 & -1.17 & -1.53 & 3.30 & 1.50 & -0.72 & -1.30 & 0.03 \\ 5.02 & 0.56 & -1.72 & -3.46 & 0.52 & 0.42 & 0.97 & 0.18 \end{bmatrix} \tag{6.11}$$

$$F'(u,v)=\begin{bmatrix} 81 & -2 & 6 & 4 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6.12}$$

$$\overline{F}(u,v)=\begin{bmatrix} 1296 & -22 & 60 & 64 & -24 & -40 & 0 & 0 \\ 0 & 0 & 0 & -19 & -26 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6.13}$$

$$\overline{f}(i,j)=\begin{bmatrix} 162 & 176 & 155 & 134 & 155 & 176 & 171 & 167 \\ 163 & 175 & 154 & 135 & 156 & 176 & 171 & 167 \\ 166 & 174 & 151 & 135 & 158 & 176 & 170 & 167 \\ 169 & 172 & 148 & 136 & 161 & 176 & 168 & 167 \\ 172 & 169 & 144 & 137 & 164 & 176 & 167 & 167 \\ 175 & 167 & 141 & 137 & 166 & 176 & 166 & 167 \\ 178 & 166 & 139 & 138 & 168 & 176 & 165 & 168 \\ 179 & 165 & 137 & 138 & 169 & 176 & 164 & 168 \end{bmatrix} \tag{6.14}$$

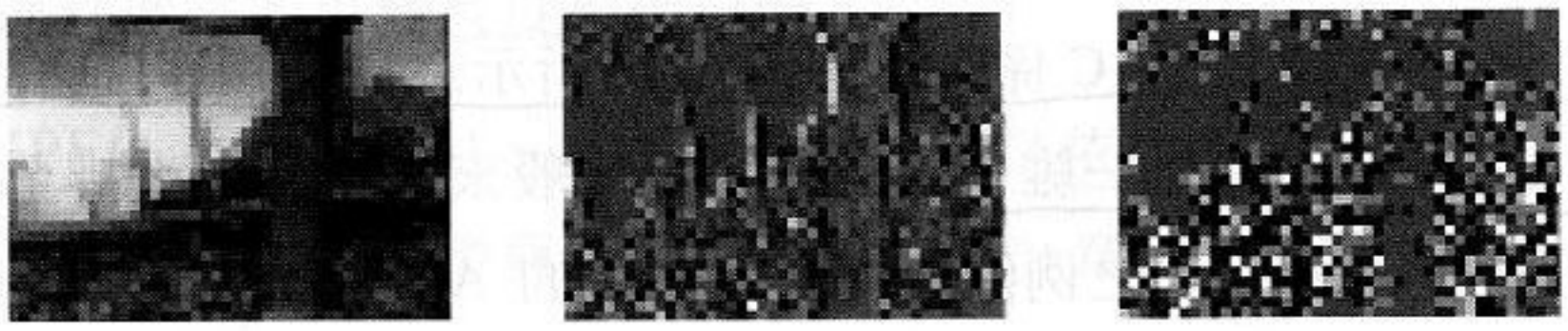
藉由量子化，幾乎所有的DCT係數會變爲0，而得到與原訊號相當接近的解碼結果。

量子化後之 DCT 係數的熵編碼

熵編碼通常分別處理DC係數與AC係數。接著闡明其理由。圖6.8爲以式(6.7)對圖2.2(a)之DCT係數作量子化圖2.2(a)之DCT係數，各影像所表示者爲各區塊量子化後之DCT係數$F'(0,0)$、$F'(1,0)$、$F'(3,2)$[81]。如$F'(1,0)$、$F'(3,2)$之AC係數，其DCT係數之區塊間相關性較小，但$F'(0,0)$爲例外。圖6.8(a)可視爲原影像之縮圖，表示$F'(0,0)$之區塊間相關程度較大。所以：

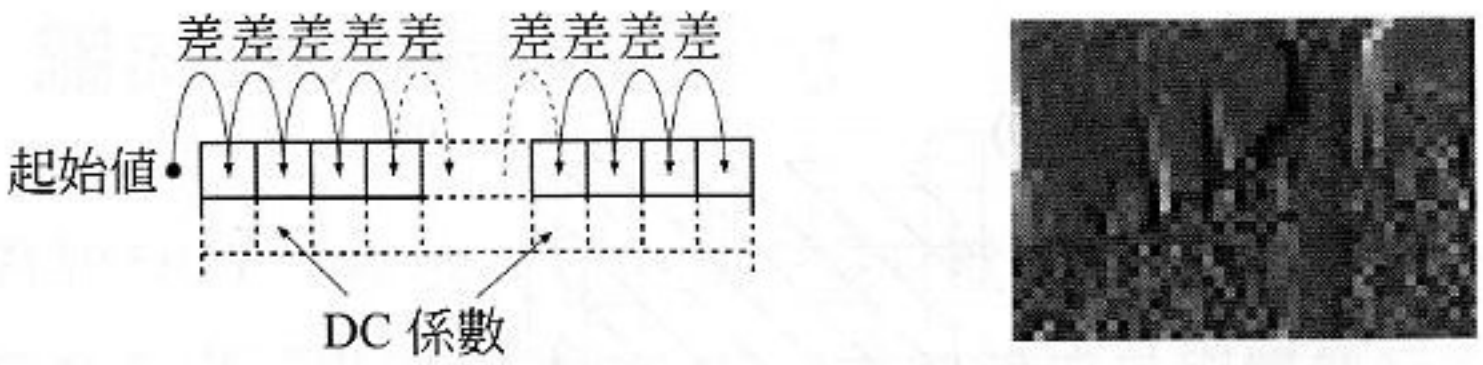
註　81. 因此，一像素可對應於一區塊。另外，因為$F'(1,0)$、$F'(3,2)$為負值所以需要加上 128 之偏移。

- DC 係數之區塊間相關程度較大，得先去除其相關程度後再進行熵編碼。
- AC 係數在區塊間可視作幾乎沒有相關，因此可以區塊爲單位進行熵編碼。



(a) $F'(0,0)$　(b) $F'(1,0)$　(c) $F'(3,2)$

圖 6.8　量子化後之 DCT 係數形成的「影像」



(a) DC 係數之差分編碼、　(b) 對 6-8(a)進行的結果

圖 6.9　DC 係數之差分編碼及其例

DC 係數如圖 6.9(a)所示，取相鄰兩區塊間之差分以去除其相關程度[82]。

註　82. 此操作稱為「差分編碼」或是「預測編碼」。差分編碼為最基本之非破壞性編碼方法。此處，關於初始化的方法(賦予初始值以計算與左上端的區塊的差分)或是迄右端之處理(中斷差分處理，重新開始／接著對次列最左方進行差分)之詳情以應用情形加以說明。

圖 6.9(b)爲對於圖 6.8(a)之區塊間取差分之結果，看得出來藉此可等同於$F'(1,0)$等 AC 係數去除相關程度。DC 係數之熵編碼，係對於該差分值進行適當的可變長度編碼分配。另外，解碼 DC 係數時，係利用差分值自左端加回去[83]。

註 83. 對於數值串 545768，以初始值為 0 計算差分值後，為 5 (−1) 12 (−1) 2。反之可得原來的數值串。

另外，對於 63 個 AC 係數，如圖 6.10 所示之掃描順序(稱爲**斜向掃描** Zig Zag Scan)加以一維化，非零係數一般來說具有由大排至小的順序。另外，以式(6.12)之例觀之，其值爲零的 AC 係數相當多，一維化之後，會如同：

…非零係數 0…0 非零係數 0…非零係數

「零值的排列 0…0」與「非零係數」交錯出現。

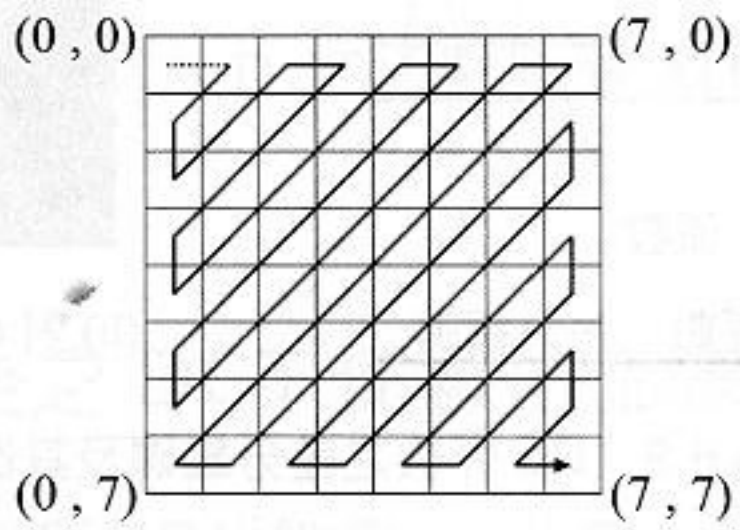


圖 6.10 DCT 係數之斜向掃瞄

舉例來說，對式(6.12)取斜向掃描之後，成爲：

$$-2,\underbrace{0,0,0}_{3},6,4,\underbrace{0,0,0,0,0,0}_{6},-1,-1,-1,-1,\ \underbrace{0,\cdots,0}_{\text{剩下的全爲 0}} \tag{6.15}$$

一般稱「零值之排列」爲**延伸(run)**，其長度稱爲**延伸長度(run length)**。在對 AC 係數進行熵編碼時，在斜向掃描之後：

- 以適當的可變長度編碼表示方式(L：F)表示延伸長度L與中斷該延伸之非零係數F。
- 使用區塊結尾編碼表示在最後的非零係數後頭之一連串的零值。

舉例來說式(6.15)可以下式表示：　特殊之編碼方式

$$(0:-2),(3:6),(0:4),(6:-1),(0:-1),(0:-1),(0:-1),\text{EOB} \tag{6.16}$$

對此可進行適當的可變長度編碼分配。

在 JPEG 或 MPEG 中，基本上也不過就是依序進行上述量子化或熵編碼方法。其細節或特徵與其間的不同在第九章與第十章加以說明。

結語－補遺

關於 DCT 區塊之尺寸

迄今對於 DCT 之區塊尺寸，幾乎很有默契地假設爲$N=8$，但是實際上需要綜合考慮了壓縮編碼的效率與計算量或是硬體執行的規模後再加以決定。

如後所述在N 爲 2 的次方的情形中，才能利用 DCT 之快速運算方法，所以$N=4$、8、16 等情況下對於處理器之運算能力或是硬體的規模加以考慮。

另外，如同在第 2-23 頁中的第一項所說的，一般在N 值較大的情形下可以去除較多像素的相關程度，雖然以編碼效率的觀點來看較爲有利。但是，實際上在像素間相關程度較高的情形下也只以附近的±10 個

像素爲範圍，尺寸在此之上的區塊也不會有更好的效果。實際上，將DCT的N值以4、8、16、32加以改變，以圖6.11表示以第九章所述之JPEG 與相同的演算法進行編碼／解碼的結果。同一圖左邊的資料量爲1.2bpp，右邊爲0.4bpp的情況。同時表示對於原影像取PSNR(dB)值[84]。

註 84. 假設量子化表以最佳量子化方式訂定，熵編碼依照符號之發生機率所求得之修正後Huffman編碼。在此僅表示Y分量之資料量，不包含量子化表與Huffman表之資料量。

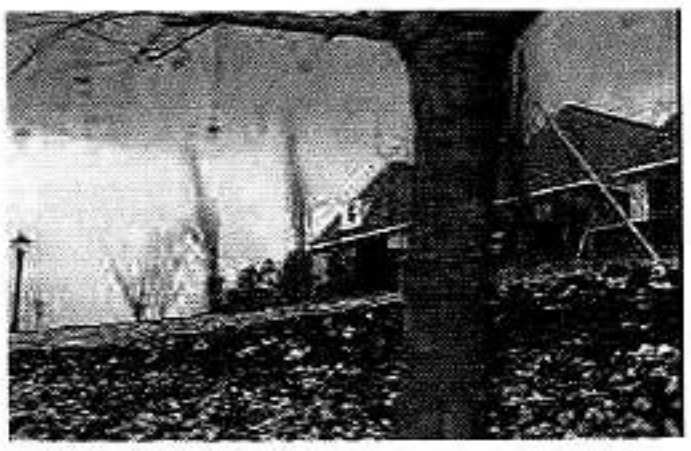
(a) 4×4 (1.2bpp, 33.9dB)

(b) 4×4 (0.4bpp, 21.2dB)

(c) 8×8 (1.2bpp, 34.8dB)

(d) 8×8 (0.4bpp, 21.5dB)

(e) 16×16 (1.2bpp, 34.2dB)



(f) 16×16 (0.4bpp, 21.5dB)

圖 6.11　使用不同區塊尺寸進行 DCT 後之編碼、解碼結果(參照扉頁)

(g) 32×32 (1.2bpp, 33.2dB)

(h) 32×32 (0.4bpp, 21.1dB)

圖 6.11　使用不同區塊尺寸進行 DCT 後之編碼、解碼結果(續)

在取高壓縮率的情形中，一般會有塊狀的雜訊[85]，特別在N 爲 16 以上會較爲顯著。另外，即使在N相對較小的情形中，在邊緣附近會有「雪花」狀的雜訊[86]，在N 值較大的時候，此現象更爲明顯。

註　85. 稱為區塊雜訊(馬賽克現象)。

86. 因為像是蚊群飛舞，可稱為蚊狀雜訊。

綜上所述，可以確認$N=8$ 之 8×8 點DCT在實用上爲最佳的尺寸。

量子化後的 DCT 係數之反轉換

依照圖 6.5 之 DCT 係數量子化，即使是進行反 DCT 也無法復元原影像，因爲包含轉換區塊的誤差。在不進行量子化的情形中，對於DCT而言，最佳的反轉換，也就是說使得平均平方差e爲最小的轉換，並不是反 DCT。也就是說，對 DCT 進行量子化的情形中，這是否成立？吾人對於這個問題並不特別考慮，而假設默認爲 YES。

對於此問題，在文獻[35]中進行理論上的推導，而得到了以下結論：「輸入影像的像素發生機率如果係依照常態分佈，當各量子化之解碼值滿足式(4.5)時，順向轉換之反矩陣(對 DCT 而言爲反 DCT)爲使平均平方差爲最小的轉換。」

「輸入像素依照常態分佈」之假設一般來說是不成立的，因此嚴格來說對於 DCT 之最佳反轉換並非爲反 DCT。但是這種最佳反轉換在實際運用上，以計算量或是資料量的觀點來看並不適當[87]。這就是說，對於DCT而言，其反轉換使用反DCT仍屬妥當。

註 87. 在編碼端預先求得的最佳反轉換，必須將編碼串的寫入法等傳送至解碼端。

快速 DCT

參照式(5.76)或式(5.100)，8×8 點DCT，可以如圖 6.12 所示以一維 8 點的DCT在縱向上進行 8 次，對其結果再於橫向上進行 8 次加以實施。這種情形下的總乘法次數爲 64×8×2 次，與前一章的結果一致。

另外，利用DC基底元素(6.1)的性質，可以削減所需的乘法計算次數，所得到的結果稱爲快速DCT。其實施有各種方法，在此介紹其中最基本的一種方法[42]。

一維的2^k點(k爲正整數)DCT 以下列方式表示：

$$\boldsymbol{X} = \sqrt{2/N}A\boldsymbol{x} \tag{6.17}$$

A爲以下形式之$N \times N$矩陣：

$$A = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \left[\cos\left(\frac{2(j+1)i\pi}{2N}\right)\right]_{ij} \end{bmatrix} \tag{6.18}$$

省略其細節，A可以分解成數個疏矩陣A_i的積[88]，式(6.18)可以表示爲以下形式：

$$\boldsymbol{X} = \sqrt{2/N}\,A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot \boldsymbol{x} \tag{6.19}$$

因此，對$\boldsymbol{x}$依序進行各轉換A_i，最後可實施一維 DCT。此處，因A_i爲疏矩陣，結果可削減整體的計算量。基於此觀念的快速 DCT 計算方法，一般稱爲「疏矩陣分解法」，而廣爲利用。

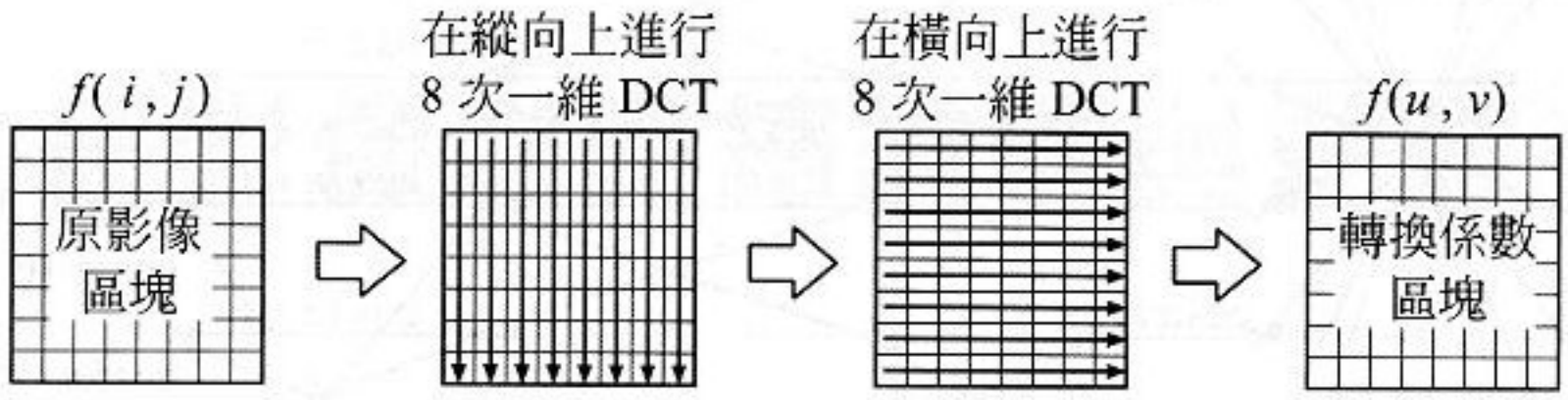


圖 6.12　由一維 DCT 進行二維 DCT

註　88. 大部份的元素為 0 之矩陣。

圖 6.13 爲以此觀念爲基礎之一維 8 點 DCT 之快速計算法。在此情形下，計算 8 個 DCT 計算所需之乘法次數，依圖 6.14 所示之步驟爲 13 次，爲直接計算之 1/4 以下[89]。依照圖 6.12 中的一維DCT之基本運算法亦可擴張至二維，結果在 8×8 區塊的轉換可以用 208 次之乘法計算完成。本節因受限於篇幅，無法涉及細節，對於矩陣A之分解或其他DCT方法有興趣的讀者可參照文獻[41]。

註　89. 式(6.19)中尺寸係數$\sqrt{2/N}$可包含圖中的乘法係數。

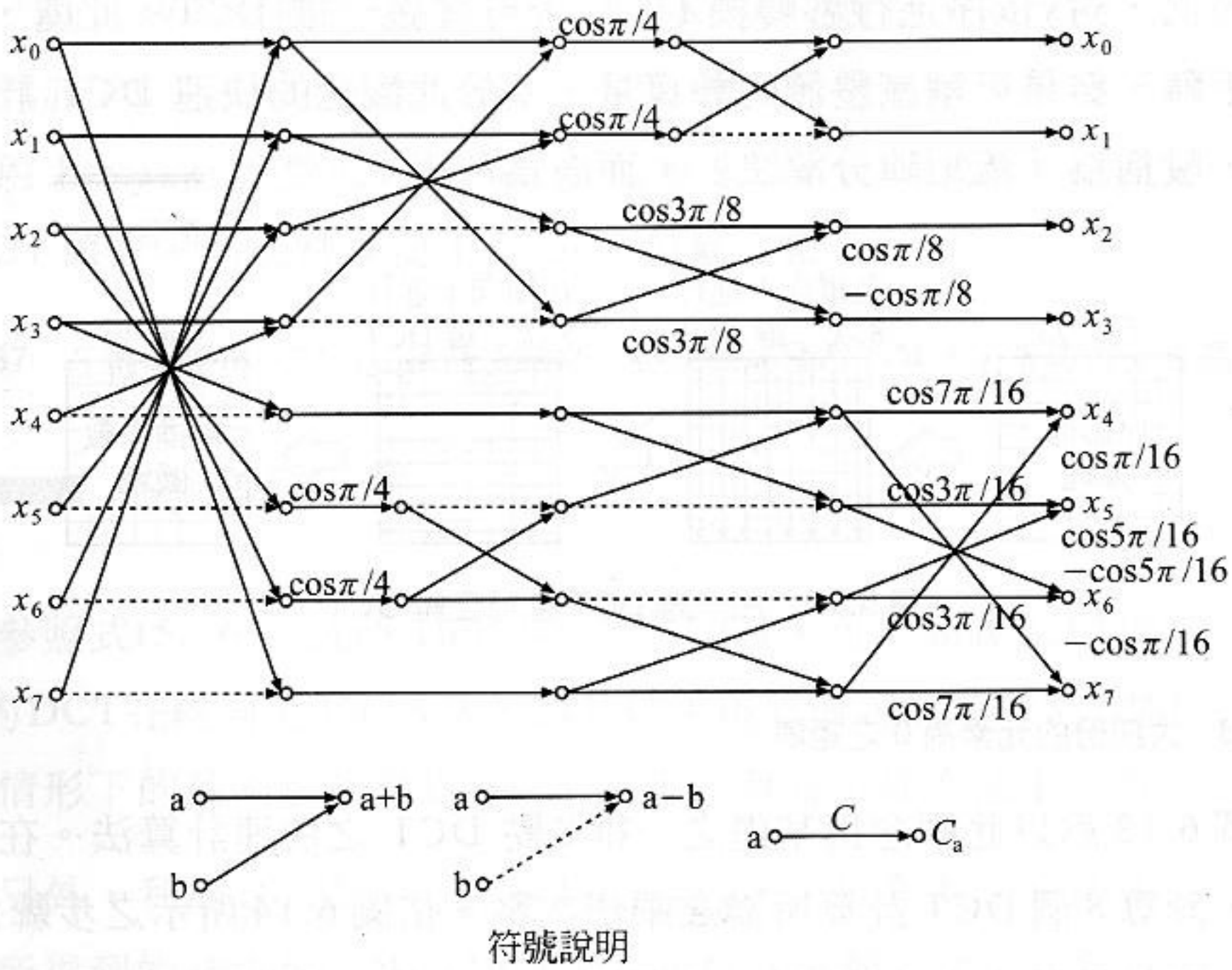


圖 6.13　一維 8 點之 DCT 快速運算法

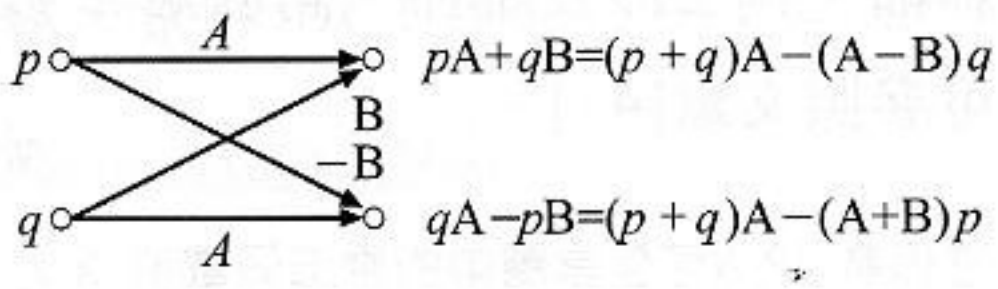


圖 6.14　蝴蝶(交叉)部份之計算方法

7 章

動畫影像之壓縮編碼與動態補償

我們到前一章為止所學的都只是靜止影像之壓縮編碼之基本觀念。在本章中，將其擴張至動畫，並闡明動畫影像壓縮編碼方法之基礎。

動畫影像之基礎－關於掃描

動畫影像，爲對固定時間間隔取樣的靜止影像之集合。構成動畫影像之單幅影像稱爲畫格，每秒鐘之畫格數稱爲畫格速率(frame/s)。

如圖 7.1 所示，因爲動畫影像在本質上爲水平、垂直、時間(畫格)方向上的三維訊號，可表示爲 $f(i,j,n)$或 $f_n(i,j)$。動畫影像的傳送，或是在電腦上的處理，一般係藉「掃描」將三維訊號轉換爲一維訊號。圖 7.2 爲稱爲漸次掃描(Progressive Scanning，或稱循序掃描)之最基本的掃描方式，對於畫格中的各水平線由左至右，再由上至下依序進行掃

描，當到達畫面的右下方之後再對下一張畫格重複進行相同的處理。

國家电視系統会議

另外一方面，在現行之NTSC等電視廣播送方式中，係使用稱爲交錯掃描(Interlaced Scanning)之方式。也就是說，如圖 7.3(a)所示，將每個畫格分成由奇數掃描線與偶數掃描線組成的兩個領域，各自依序進行掃描[90]。圖 7.3(c)爲一般之 NTSC 所拍攝的動畫影像之一幅畫格，移動中的物體會如圖所示爲兩影像之重疊。其原因如圖 7.3(b)所示，單一畫格爲時間差爲 1/60 秒的兩領域所組成。

註 90. 使用交錯式掃描會使得影格速率成為兩倍，在制訂NTSC規格時，係考量藉此可減少殘像。但是，在交錯掃描時在時間與垂直方向的平面上係形成五個孔狀的取樣圖形，如果不施行相對應的頻帶限制，將無法得到足夠的效果[43]。因此實際上，當頻帶限制不足時會產生交疊現象，但是NTSC現在已經是相當普及的規格，所以時至今日仍然沿用。

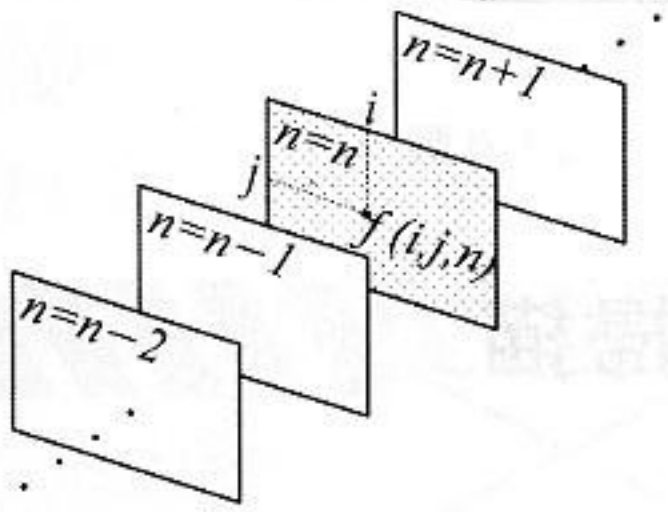


圖 7.1 動畫影像

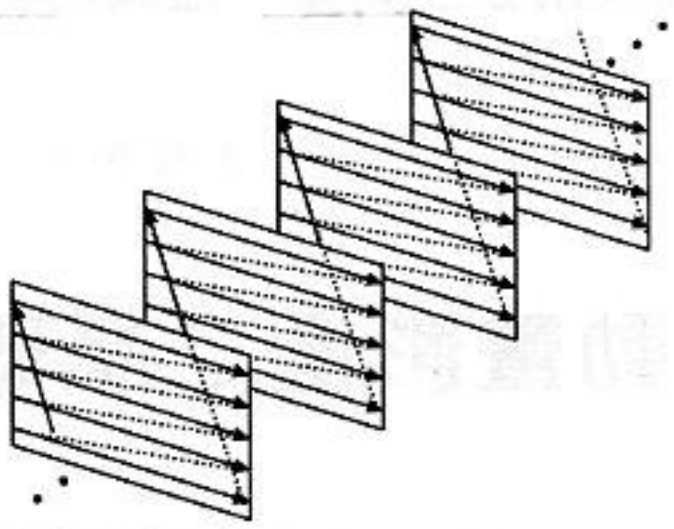
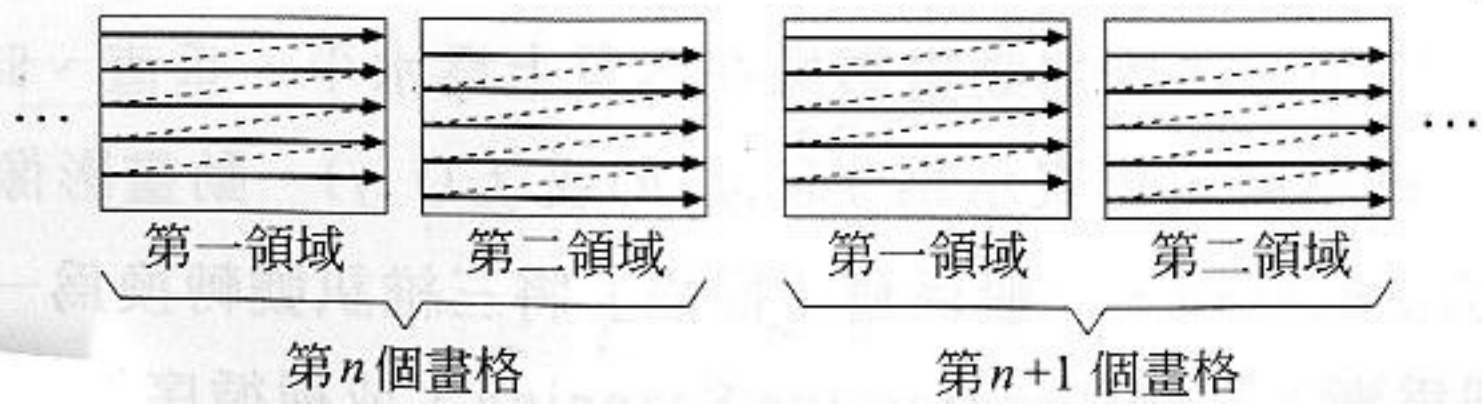


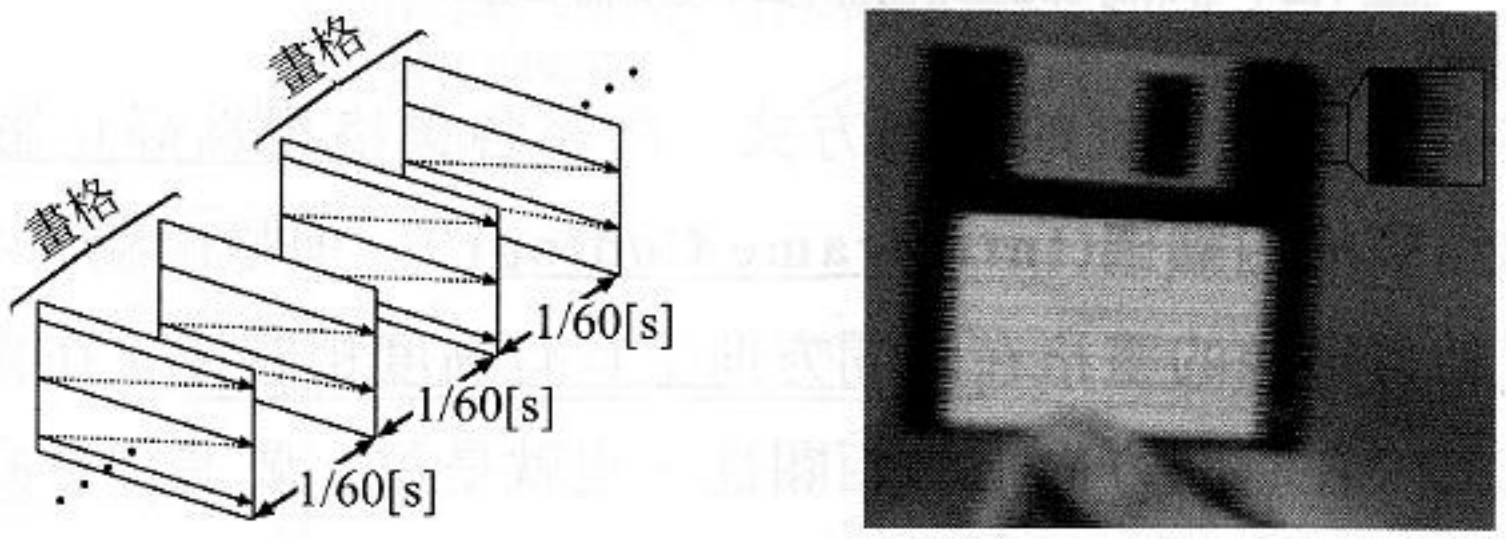
圖 7.2 漸次掃描



(a)

圖 7.3 交錯(interlaced)掃描

反而使要拍攝的影像多?



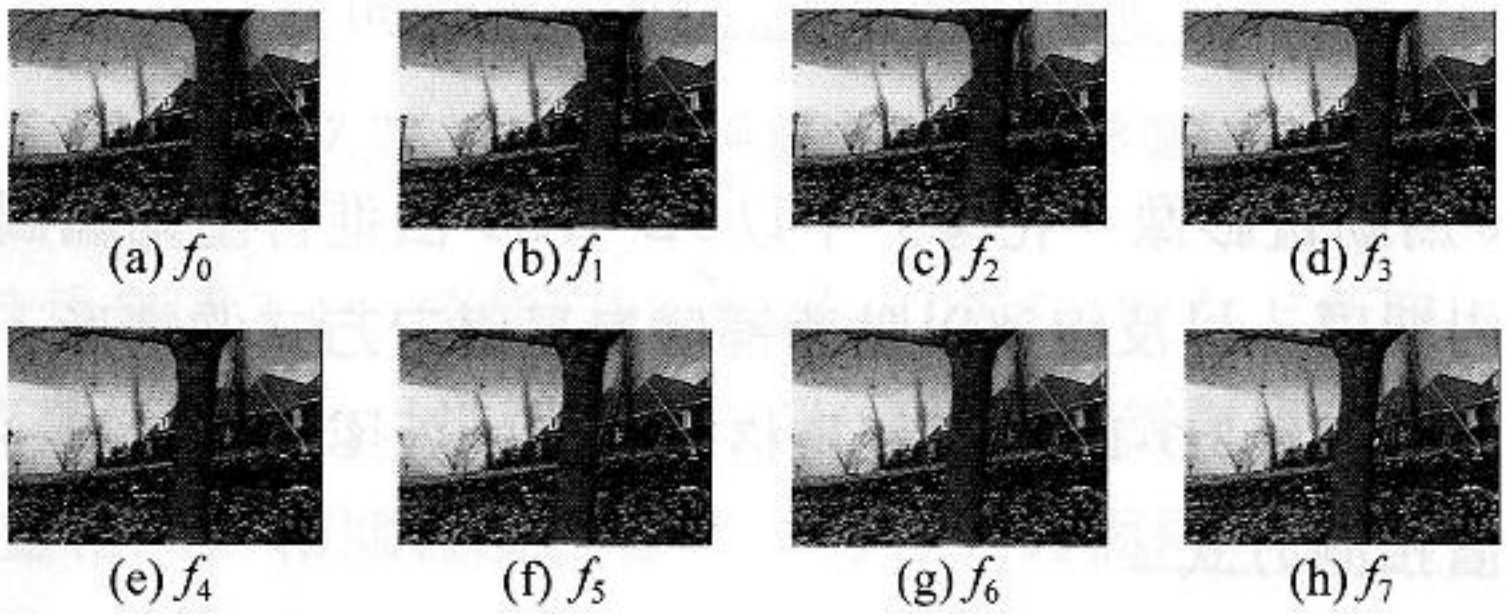
(b)　　(c)

圖 7.3　交錯(interlaced)掃描(續)

在本章中，基本上以次掃描爲對象，討論其編碼方式。雖然亦適用於交錯掃描，但是適用於交錯掃描之特殊的編碼方式，將在第十章中 MPEG-2 規格的章節中說明。

動畫影像編碼之基本方法

圖 7.4 爲動畫影像「花園」之前八個畫格。



(a) f_0　(b) f_1　(c) f_2　(d) f_3

(e) f_4　(f) f_5　(g) f_6　(h) f_7

圖 7.4　「花園」的八個畫格

畫格內編碼到畫格間編碼

動畫影像之最單純的編碼方式，爲將各畫格視爲靜止影像加以編碼，此稱爲**畫格內編碼(Intraframe Coding)**[91]。但是由圖 7.4 可以明顯地看出，動畫影像的畫格在時間方向上具有高度相關性，在畫格內的處理不可能去除在時間方向上的相關性。也就是說，要進行更高效率的壓縮必須去除與時間**有關的相關性**，爲此，牽涉到複數畫格之處理—**畫格間編碼(Intraframe Coding)**—是不可或缺的。

註 91. 。

依时間变化少
时間相近一同编碼

使用三維正交轉換進行畫格間編碼

先前在靜止影像之轉換編碼中，爲了達到去除水平垂直兩方向的相關程度，運用了二維的正交轉換。如單純地擴張此觀念，將縱軸—橫軸—時間軸的三維(3D)正交轉換運用在動畫影像上，可否不僅在空間方向上，且同時去除在時間方向上的相關性[44]？在此以 之 3D 區塊爲單位，試著由3D-DCT、量子化、熵編碼構成之演算法進行動畫影像之畫格間編碼。

圖 7.6 爲動畫影像「花園」中以 3D-DCT 法進行壓縮編碼之效率，使用畫格內編碼，以及後述**以動態補償為基礎之方式**的情形下之效率亦同時表示於圖內。3D-DCT法較畫格內編碼的效率爲高，另一方面，不及動態補償預測方式。

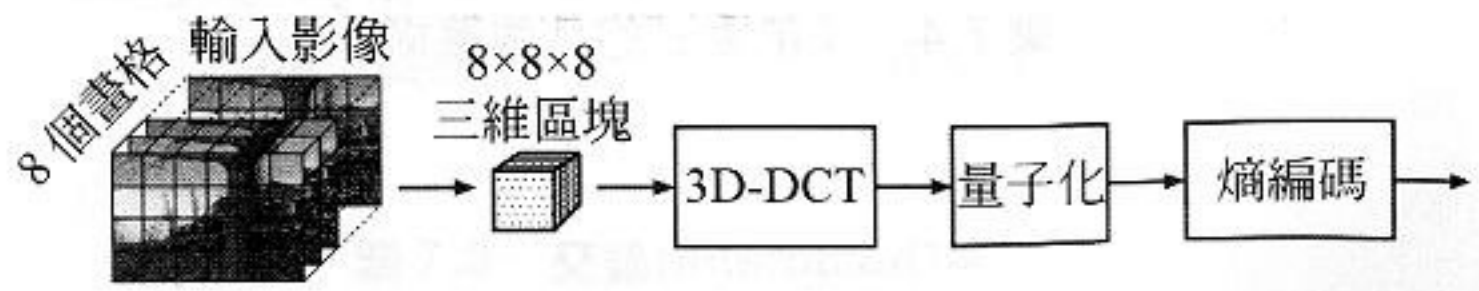


圖 7.5 由 3D-DCT 進行動畫影像之畫格間編碼

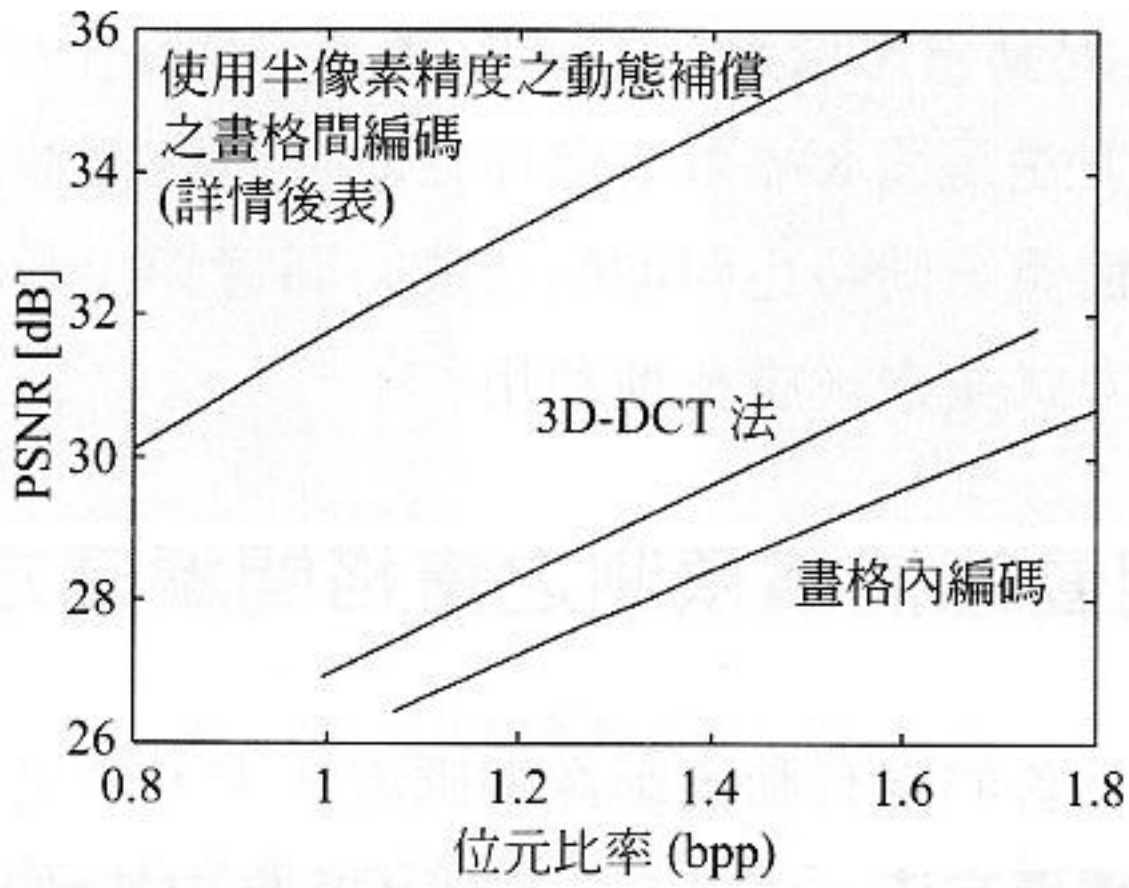


圖 7.6　3D-DCT 法之壓縮效率(處理對象之影像為「花園」)

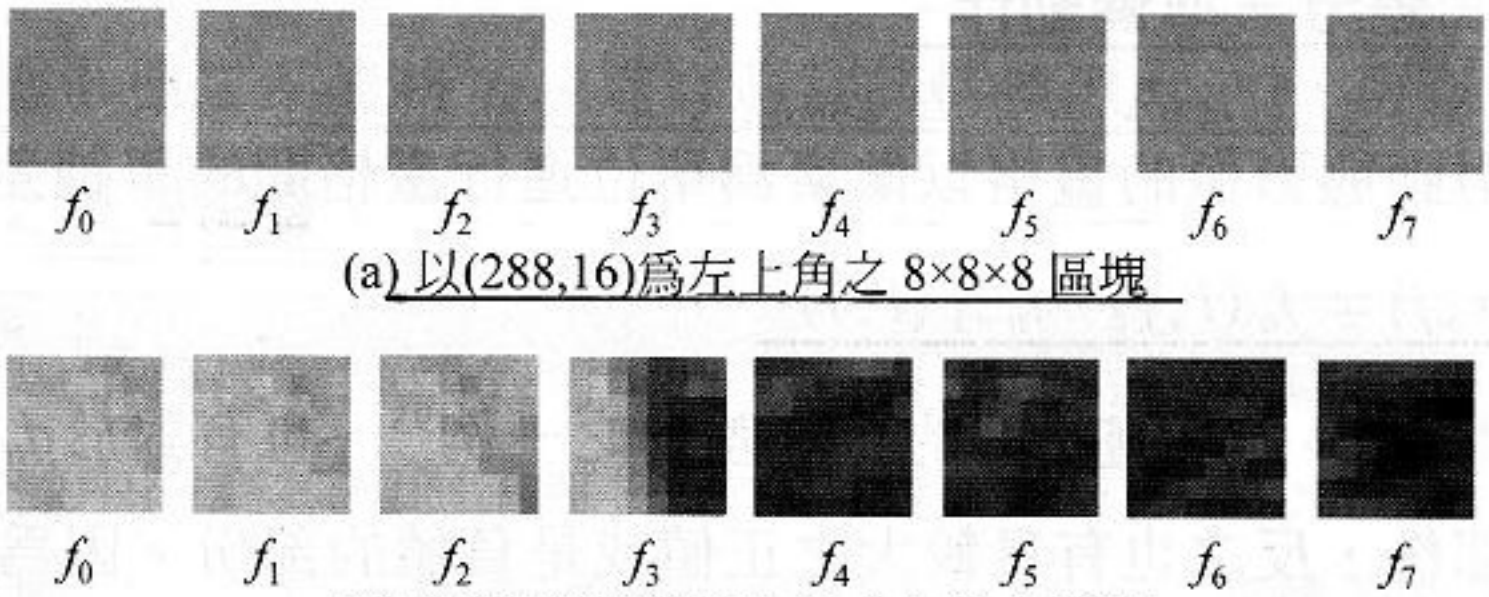


圖 7.7　自圖 7.4 之各畫格切出的 8×8 區塊

圖 7.7 爲自圖 7.4 以座標(288,16)與(192,56)爲左上角取出 8×8×8 之 3D 區塊依畫格順序加以排列。同一圖(a)的區塊都對應到「藍天」的部份，因此畫格間的相關程度相當高，另外一方面同一圖(b)受到「向左移動的樹幹」的影響，因此在區塊內的圖案會隨著時間而有所變化。因而像是(b)這種區塊在使用 3D-DCT 時無法得到滿意的壓縮效果[92]。

註　92. 但只要是在影像中的移動較小的情形中，3D-DCT 的方式可以得到接近甚至凌駕 MPEG 的性能[45]。

以3D-DCT 對動畫影像進行編碼的嘗試，是在1970年代後期才開始討論的。這種方法是將8幅畫格儲存在記憶體後再進行3D-DCT，除了需要大量的記憶體，同時也明顯的造成時間遲滯(latency)的問題。爲此，3D-DCT的方式並未被廣泛地利用。

使用動態補償預測之畫格間編碼方法

在以MPEG爲首的現有動畫影像編碼方法中，係使用第三種實行動態預測之畫格間編碼方法。這種方式以第115頁中註釋82所述之差分編碼爲基礎，並應用於畫格方向上。

關於畫格間差分－準備動作

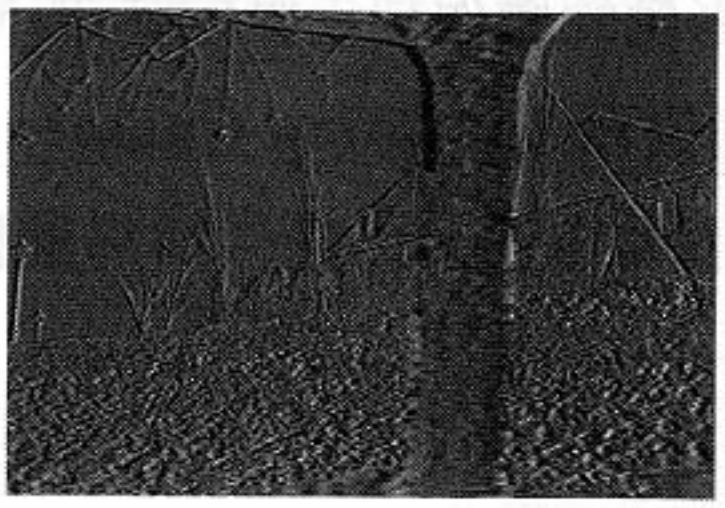
考慮對動態影像的畫格以像素爲單位進行畫格間的單純差分：

$$d_n(i,j)=f_n(i,j)-f_{n-1}(i,j) \tag{7.1}$$

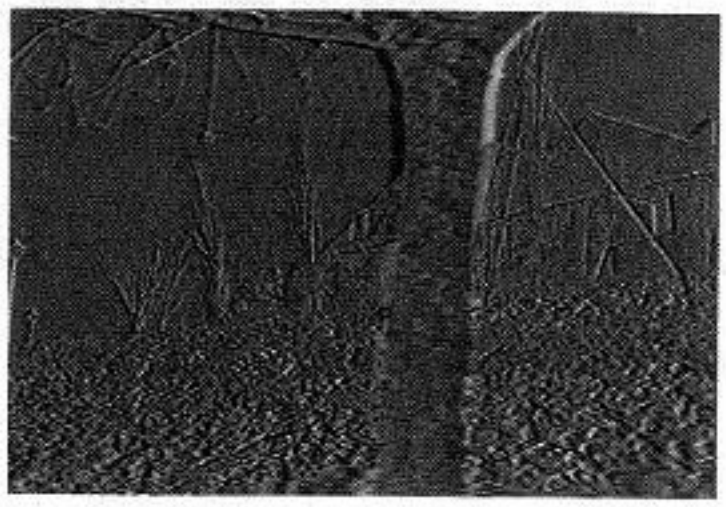
圖7.8爲對圖7.4進行畫格間的單純差分之一例[93]。也有對於$d_n(i,j)$之值接近零的部份，反之也有具較大之正值或是負值的部份。因爲動畫影像之各部份都「在動」，舉例來說即使受拍攝的對象相同，其差分值d_n也不見得就會接近零。此外，實際被拍攝的對象如不爲剛體，也會隨著時間產生變形。其結果可表示爲：**前一個畫格中的同一個像素即使是進行差分編碼取預測值也無法完全地去除其相關程度。**

註 93. 為了得到有正有負的差分值，差分值為零係表示亮度值128的偏移。

差分效果還是有限



(a) $d_1 = f_1 - f_0$



(b) $d_4 = f_4 - f_3$

圖 7.8　畫格間單純差分例(0　128)

畫格差分之改善－導入動態補償

因此，爲了改善動態差分的去除相關程度效率，假設：

> 對於動態影像之某畫格 f_{n-1} 進行適當的轉換$F(\boldsymbol{p}_n, \cdot)$，可以得到下一個畫格之**預測影像** f_n'。

除了包含場景轉換等極端的情形，可以自然地得到這個假設。實際上，在動態影像中的對象物體，雖然會伴隨著平移、放大、縮小、旋轉、變形等種種變化，但是可以視這些平移、放大縮小、旋轉等爲**幾何轉換**的組合。因此，這些幾何轉換可以用整合了預設的參數(平移量、放大率、旋轉角度等參數)之向量$\boldsymbol{p}_n$表示，適當地加以選擇，可以得到以下假設：

$$f_n' = F(\boldsymbol{p}_n, f_{n-1}) \simeq f_n \tag{7.2}$$

在此，符號$F(\boldsymbol{p}_n, f_{n-1})$表示對畫格 f_{n-1} 進行由參數$\boldsymbol{p}_n$決定之幾何轉換。此時，依照式：

$$d_n = f_n - f_n' = f_n - F(\boldsymbol{p}_n, f_{n-1}) \tag{7.3}$$

在「可以準確預測」的假設下，$d_n \approx 0$ 成立，因此可以去除畫格間的相關性。

一般來說，這種自畫格得到x之預測影像的操作稱爲**動態補償(Motion Compensation：MC)**，或是稱作**動態補償預測(Motion Compensation Predition)**，另外，自第$(n-1)$與第n個畫格取得$\boldsymbol{p}_n$之操作稱爲**動態推測(Motion Estimation：ME)**。

使用動態補償之動畫影像編碼

基本演算法

圖 7.9 爲使用動態補償之動畫影像編碼／解碼演算法之基本組成。當編碼端完成了至第$(n-1)$個畫格之編碼的傳送，在開始傳送第 n 個畫格之前，另一側的解碼端爲剛解碼完畢第$(n-1)$個畫格f_{n-1}''的狀態。另外，在編碼端，對於f_n之動態參數$\boldsymbol{p}_n$，假設已經(以後頭所述之方法)完成推測。

在編碼端，對於畫格以接下來的步驟進行編碼。首先，使用動態參數$\boldsymbol{p}_n$求取f_n之預測影像：

$$f_n' = F(\boldsymbol{p}_n, f_{n-1}) \tag{7.4}$$

之後，依照式(7.3)計算差分影像d_n。得到編碼後的編碼串D_n之後，與$\boldsymbol{p}_n$一同傳至解碼端。

另外在解碼端，在得到將D_n解碼後之差分影像d_n'的同時，對既有之解碼影像f_{n-1}''以$\boldsymbol{p}_n$進行 MC：

$$f_n''' = F(\boldsymbol{p}_n, f_{n-1}'') \tag{7.5}$$

可得到預測影像 f_n'''，最後將dn'與 f_n'''疊加，可以解碼第n個畫格的影像 f_n''。

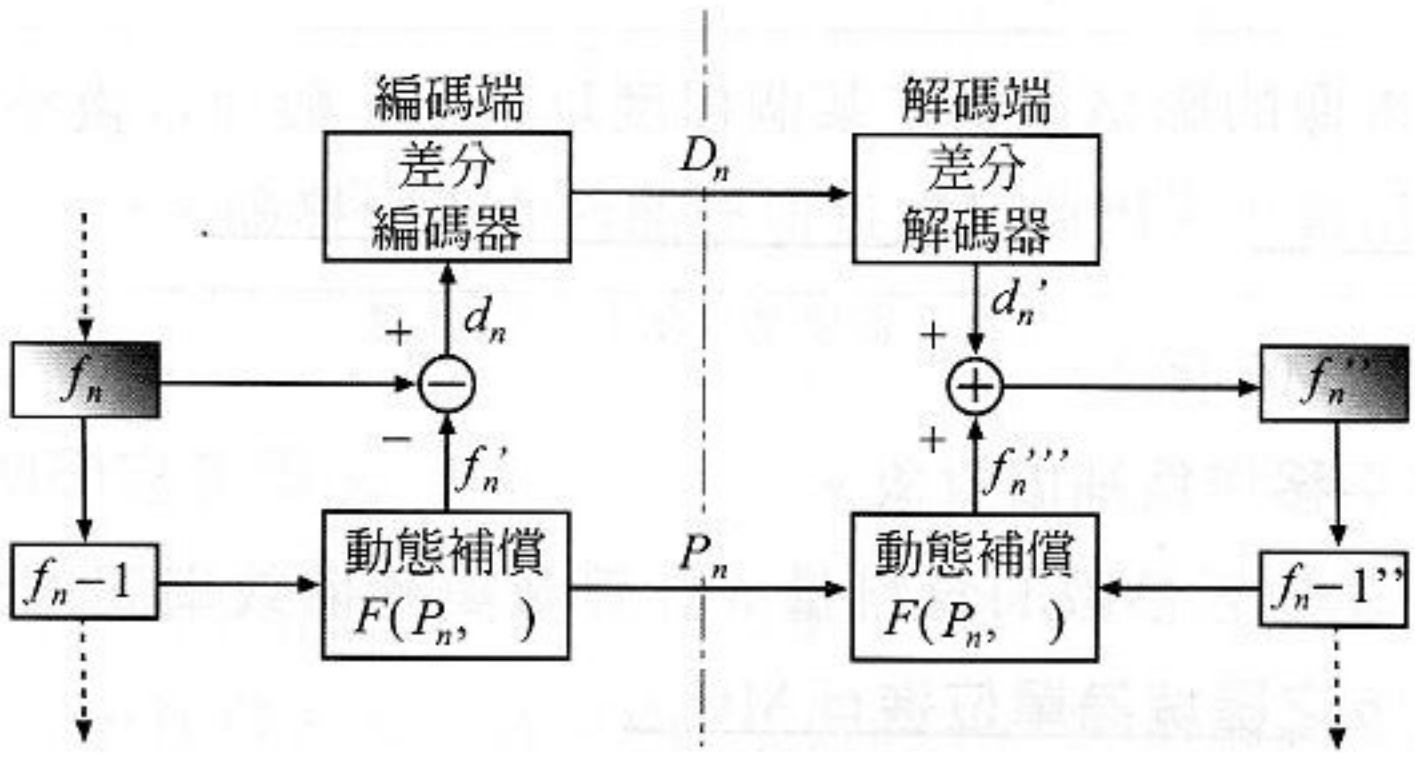


圖 7.9　使用動態補償之動畫影像編碼／解碼演算法之基本組成

對以上的操作以$n+1 \rightarrow n$反覆操作，可以對動畫之各畫格進行編碼／解碼。

動態補償預測方式與其參數

在使用MC時，一般考慮：

$$[D_n\text{之資料量}] < \left[\begin{array}{l}\text{對 } f_n \text{ 進行畫格內編碼}\\ \text{情形下的資料量}\end{array}\right] \tag{7.6}$$

但是，對第n個畫格進行解碼時，除了D_n之外還需要動態參數$\boldsymbol{p}_n$，因此以資料量的觀點，需滿足：　Pn可否共通?

$$[D_n\text{之資料量}] + [\boldsymbol{p}_n\text{之資料量}] < \left[\begin{array}{l}\text{對 } f_n \text{ 進行畫格內編碼}\\ \text{情形下的資料量}\end{array}\right] \tag{7.7}$$

MC方式才爲有效。另外，動畫影像之即時編碼需要達到每秒30畫格的處理，因此ME或是MC所需的運算量必須符合現實情況。另外一方面，

爲了要得到高精密度的預測影像，雖然希望可以有多樣的動態或是變形的補償，不過依照前述的制約條件，無法使用複雜而參數多的 ME／MC。因此，在實際上，基於以下觀念：

> 因爲動畫影像的影格速率在某個程度以上，一般而言畫格間的移動或是變形相當小，因此可以近似於部份的平行移動。

現行的 ME／MC 係：

1. 僅將**平移**作爲補償對象。
2. 爲了在動態參數的資料量或計算量與補償效率間得到妥協，以 16×16 **之區塊為單位**進行 MC。

如此一來，動態參數，對於 16×16 區塊的平移量僅以一個向量表示，此向量稱爲**移動向量(Motion Vector：MV)**，作爲處理單位之x 區塊稱爲**宏塊(Macroblock：MB，或稱巨集區塊)**。

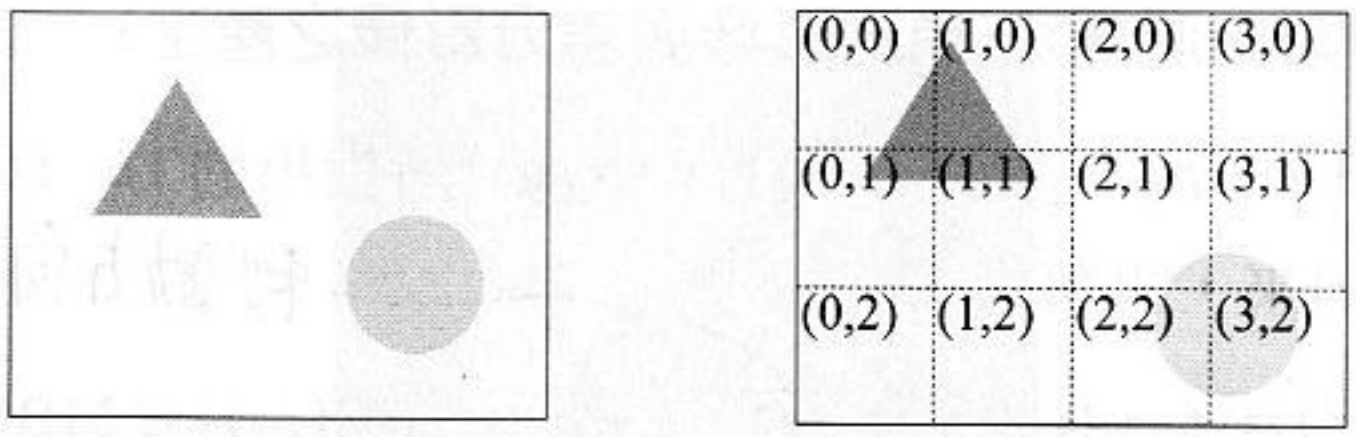
動態推測、動態補償之基本演算法

依照前項的方式，ME 爲以 **MB 為單位計算 MV 之處理過程**，另外一方面，**MC 相當於使用移動向量**$[v_x v_y]^t$依照

$$\begin{bmatrix} i \\ j \end{bmatrix}^t = \begin{bmatrix} i \\ j \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} \in \mathrm{MB} \tag{7.8}$$

對相對應區塊進行平移之處理過程。

因此，ME／MC 可依下列演算法加以實現。在此，以圖 7.10 之兩幅畫格爲基礎加以說明。



(a) 前一張畫格 f_{n-1}　　(b) 現在的畫格 f_n

圖 7.10　連續兩幅動畫的畫格

動態推測(ME)之演算法

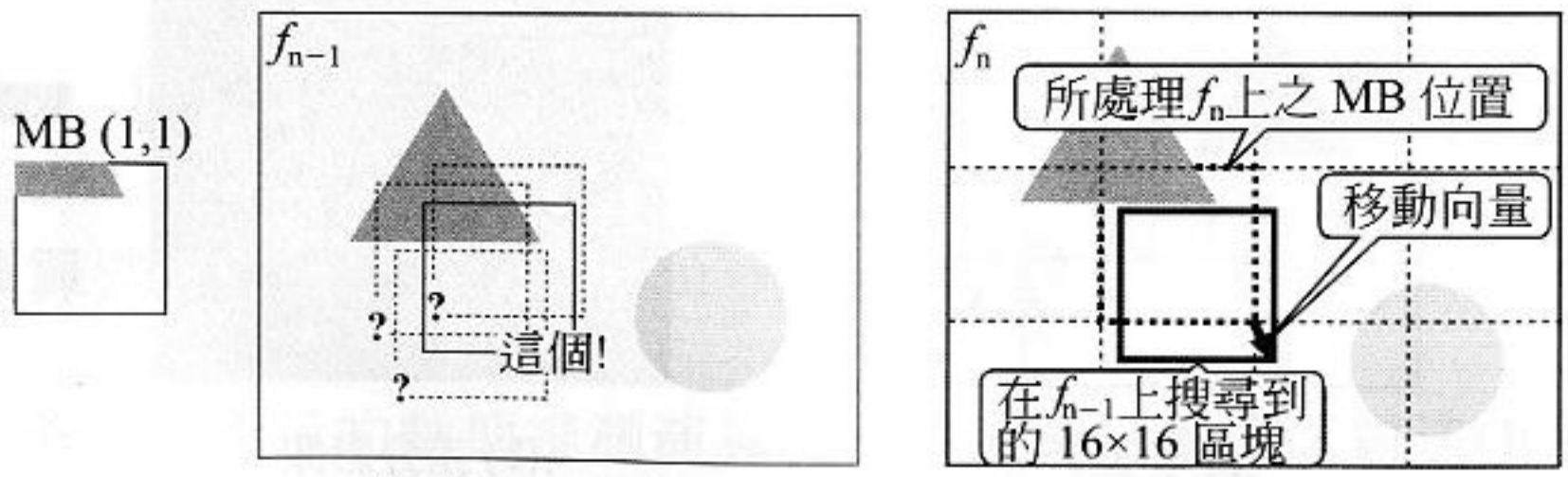
① 將 f_n 分割成不重疊之 16×16MB，賦予各 MB 位址(p, q)(圖 7.10(b))。

② 在 f_{n-1} 上找尋與 f_n 之各 MB **最接近**的 16×16 區塊。舉例來說，對 MB(1,1)，可以找到圖 7.11(a)中實線區塊爲最接近的區塊。此時，將該 MB 之 MV 定義爲(圖 7.11(b))：

自「在所處理畫格 f_n上的 MB 位置」至「在 f_{n-1}上找到的 16×16 區塊位置」間的向量。

以下，將 MB (p, q)的 MV 以$[v_x(p, q), v_y(p, q)]^t$表示。

③ 對 f_n 上所有的 MB 進行上述處理，求得 MV。圖 7.13(a)爲對圖 7.10(b)所得到的所有 MV。



(a) 搜尋區塊　　(b) 所得到的 MV

圖 7.11　搜尋最接近 MB (p, q)之區塊與 MV 之定義

動態補償(MC)之演算法(預測影像與差分影像之產生)

① 在MB(p,q)中，將$[v_x(p,q), v_y(p,q)]'$所指出的16×16區塊自f_{n-1}切割出來，再藉

移動方向

$$f_n'(i,j)=f_{n-1}(i+v_x(p,q),\ j+v_y(p,q))(i,j)\in \mathrm{MB}(p,q) \quad (7.9)$$

貼到預測影像上f_n'所對應之位置(p,q)。

② 對所有MB進行①所述之步驟，得到動態補償影像f_n'。

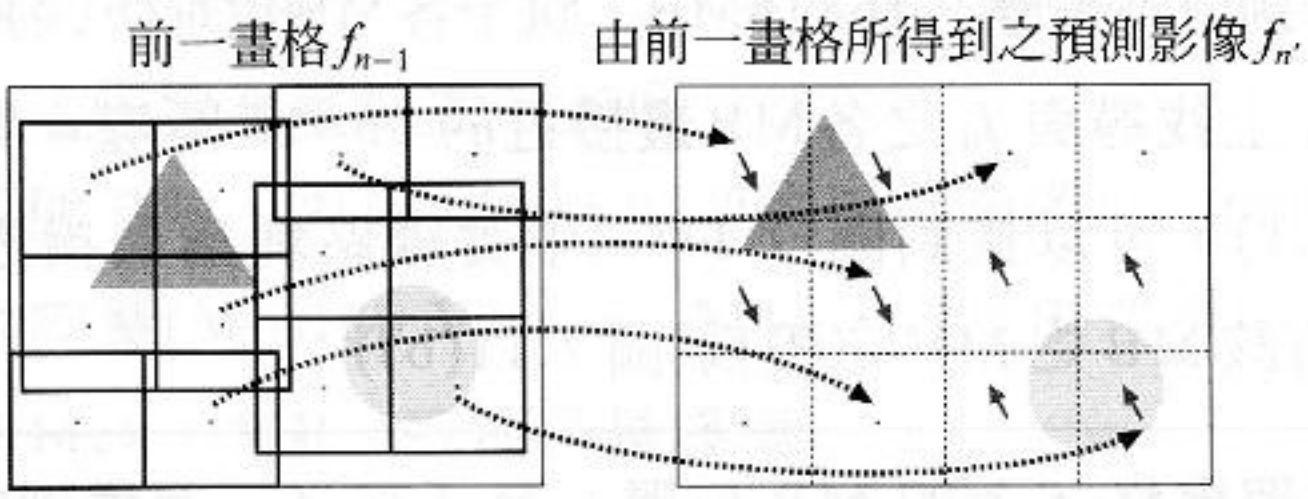
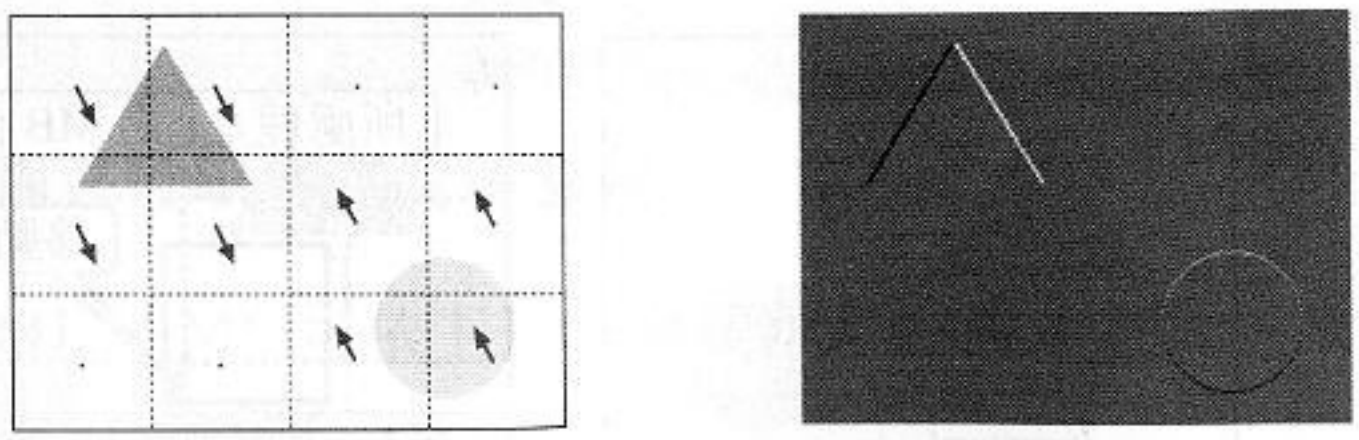


圖7.12 動態補償預測影像f_n'之產生

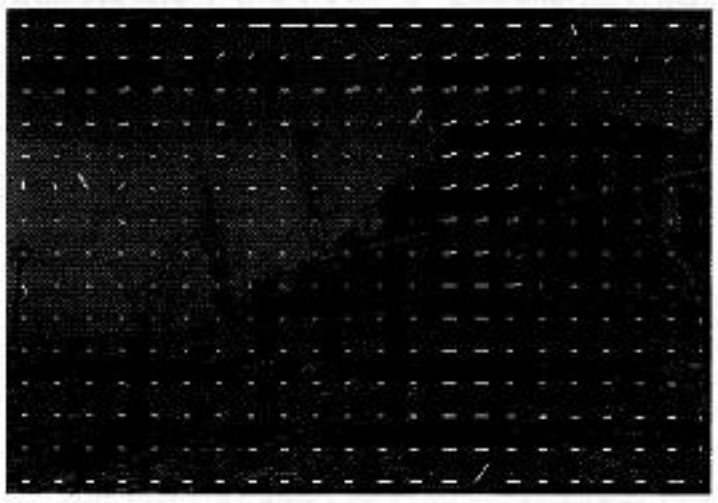
③ 以式(7.3)求得現有的畫格f_n與動態補償影像f_n'兩者間的差分影像。圖7.13(b)爲基於以上步驟所得到圖7.10(b)之差分影像。



(a) 全MV　　(b) 差分影像

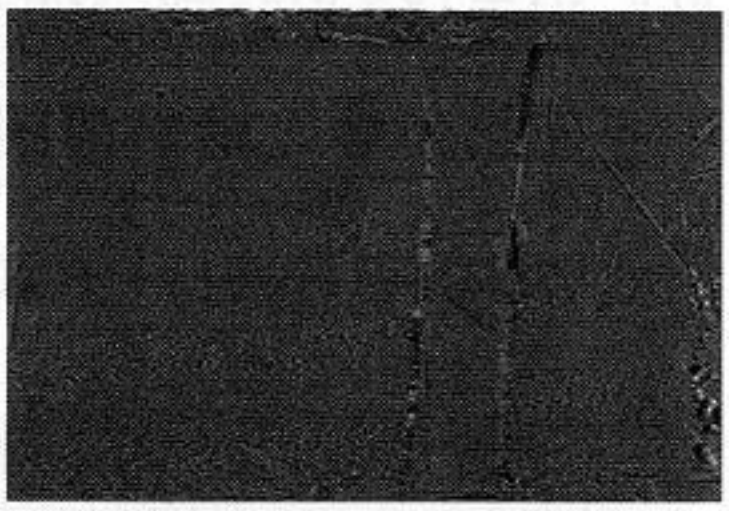
圖7.13 對圖7.10所得到的MV與差分影像

一句話：对移動的預測



(a) MV(爲了易於看見降低亮度)

(b) 預測影像



(c) 原影像與(b)之差分影像

圖 7.14　對圖 7.10 進行 ME ／ MC 之結果

通常，進行處理之現有影像 f_{n-1} 稱爲**目標畫格(Target Frame)**；用來搜尋對應區塊之畫格 f_1 稱爲**參考畫格(Reference Frame)**。圖 7.14 爲以圖 7.4(a)之 f_0 爲參考畫格，(b)之 f_1 爲目標畫格進行 ME ／ MC 之結果。比較圖 7.14(c)之差分影像與圖 7.8(a)之單純差分，可以看到有更多亮度值集中在零附近，因此可確認動態補償的效果。

區塊比對法

基本演算法

在前一節所示的動態推測演算法一般通稱爲**區塊比對法(BM 法)**，在 MPEG 等動畫影像編碼方法中被廣泛使用。

現在以 $f(i,j)$表示目標畫格；$g(i,j)$表示參考畫格。在使用 BM 法時，首先必須決定兩個區塊間的**接近程度**，或是**比對誤差**E之測量方法。通常E以下列方式測定：

$$E=\sum_{(i,j)} \ln \mathrm{MB}\, L(f(i,j),g(i,j)) \tag{7.10}$$

其中，$L(x,y)$爲表示兩值x與y之間距離的函數。一般使用：

$$L(x,y)=|x-y| \quad 或 \quad (x-y)^2 \tag{7.11}$$

在本書中，使用前者之**絕對值誤差**。

使用式(7.10)與(7.11)，BM 法之演算法可表爲：

$$\mathrm{MV}=\arg \min_{(v_x,v_y)\in S} \underbrace{\sum_{(i,j)\in \mathrm{MB}} |f(i,j)-g((i+v_x,j+v_y)y)|}_{E} \tag{7.12}$$

S爲**移動向量之搜尋範圍**，arg min的部份，表示「使得E爲最小之(v_x,v_y)搜尋操作」。實際上，依序對於範圍S內**所有可能移動向量**(v_x,v_y)計算比對誤差E，可得到最小值之(v_x,v_y)爲所對應之宏塊的 MV。

另外，由 BM 法所得到的 MV，實際上並不是「表示畫格間的移動」，應解釋爲「可去除最多畫格間**多餘資料(冗長性)**的向量」爲宜。

比對誤差曲面與推算移動向量之難易度

式(7.12)之比對誤差E，在S內對於各點會形成一曲面(**比對誤差曲面**)。因此，BM 法可解釋爲搜尋此曲面內最低點。圖 7.15 爲數個誤差曲面之例。同一圖(c)爲典型的曲面，可以容易得到使最小值存在點之 MV，另外一方面同一圖(a)中，E幾乎爲定值，不存在明確的最小值，因此 MV 可以取任一可能的點。同一圖(b)、(d)爲對應的圖案無法在參考畫格之S範圍內得到較小誤差的例[94]。

註 94. 在圖 7.15(b)的圖中，沒有$v_y>0$的領域，因此將目標區塊內$v_x>0$部份自參考畫格內切掉。參照第 10-11～10-12 頁。

對實際影像使用BM法時，幾乎所有的誤差曲面皆具有如圖 7.15(c)之具有清楚的最低點的特性。另外，在同一圖(a)的情形中MV幾乎是**不確定的**，至於在同一圖(b)與(d)中無法求得有效的 MV。接著再討論無法進行 ME 的情形。

移動量和誤差

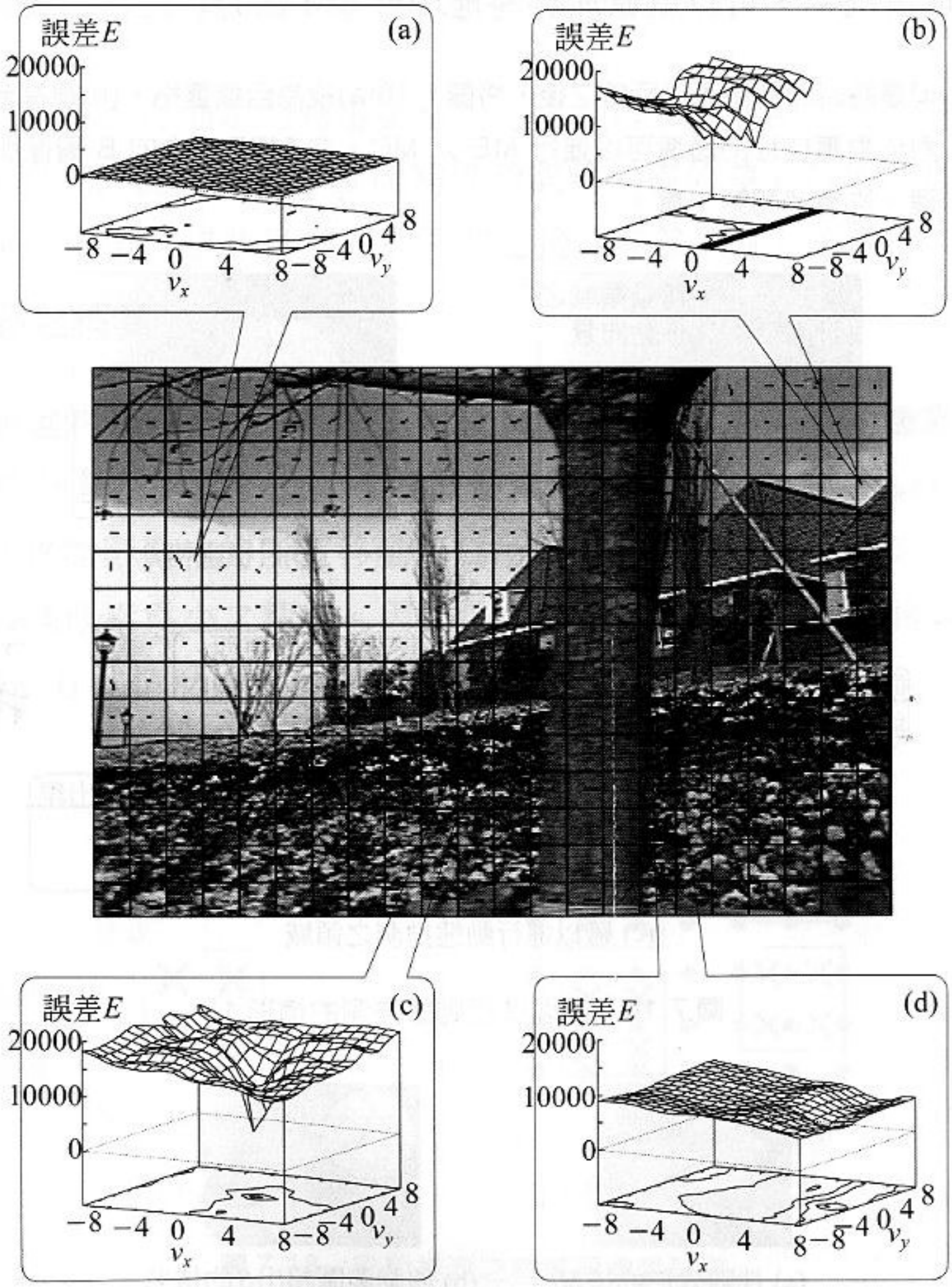
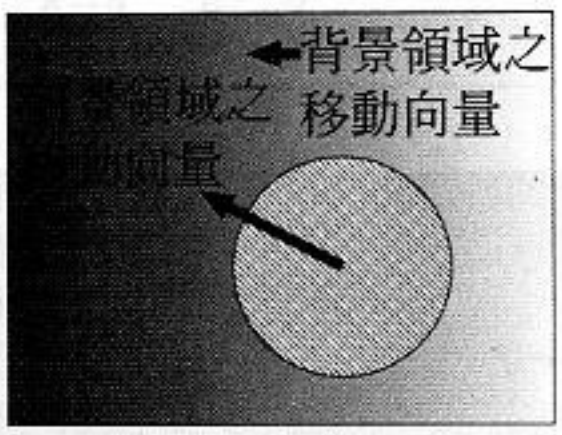


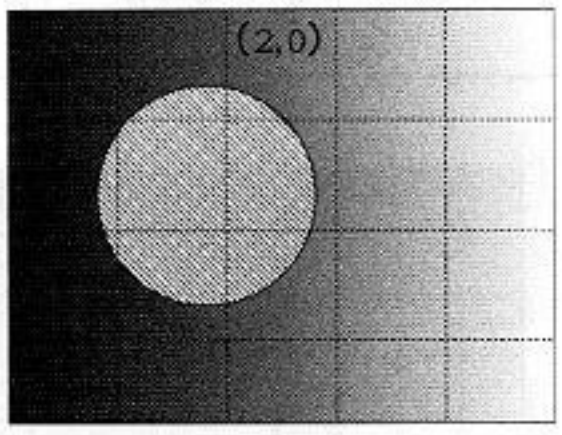
圖 7.15　在 BM 法中比對誤差曲面之例

圖 7.16(a)、(b)爲連續兩畫格的情形。原則上無法對(c)之「同時包含移動物體與背景的區塊」、或是「有新東西進入影像內」進行 ME。另外，「在移動的同時同時出現新的背景領域」一般稱爲**未涵蓋領域 (Uncovered Region)**，這種情形也無法求得MV[95]。另外，在圖 7.17 所示的各種情況中，可以就原理容易地理解 BM 法的功能無法發揮。

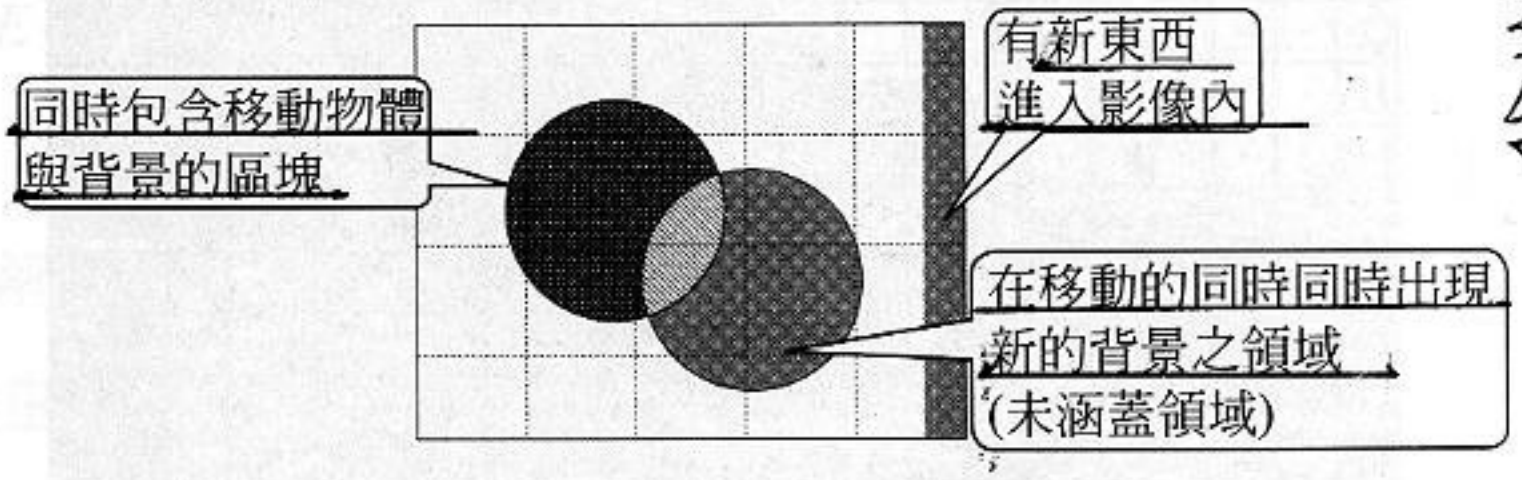
註 95. 但是將時間關係予以反轉之後，將圖 7.16(a)視為目標畫格、(b)視為參考畫格，對於對應位址的區塊可以進行 ME ／ MC。在 MPEG 中的 B 圖像即應用此原理，詳情參閱第十章。



(a) 參考畫格 f_{n-1}



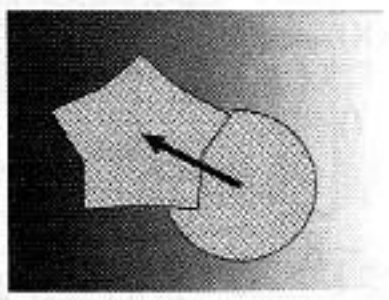
(b) 目標畫格 f_n



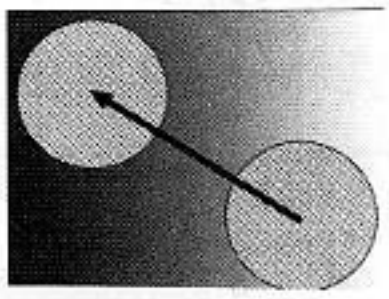
3種
特殊情形

(c) 難以進行動態推測之領域

圖 7.16　難以進行動態推測的情形 1



(a) 伴隨變形的情況



(b) 移動範圍超出 S 的情形

圖 7.17　難以進行動態推測的情形 2

(c) 伴隨旋轉或是縮放的情形　(d) 伴隨亮度變化的情形

7種情形

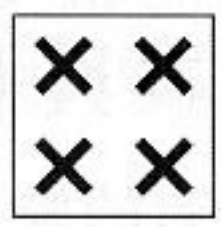
圖 7.17　難以進行動態推測的情形 2(續)

由上所述，在通常的動畫畫格中，可以自然地想到經常有難以進行 ME 的區塊，因此在進行以動態補償預測爲基礎的差分符號的情形中，需要有某些對策，這些具體對策將在之後的章節中加以介紹。

動態推測之精度

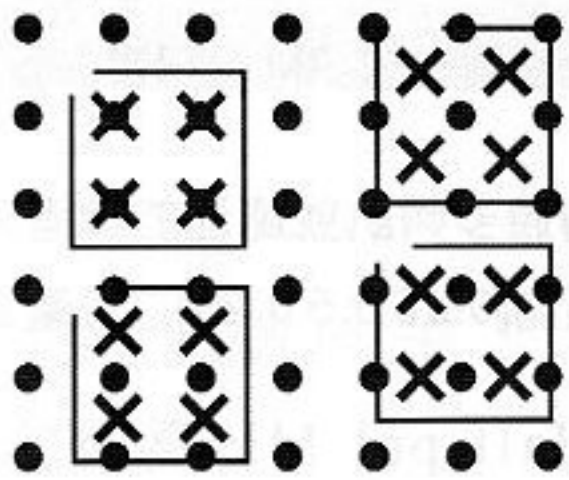
在前述的 BM 法中，係預先假設 MV 的候補點在 S 內的**像素**中，因此 MV 的分量爲**整數**。此稱爲**整數像素精度動態推測(Full-pel ME)**。另外，在實際影像中的所有移動可爲所有的實數值，因此原本就希望可以有較一個像素爲小之精度。因此，考慮**半像素精度之動態推測(Half-pel ME)**。在此情形中，MV 候補點的分量可取整數或是(整數＋0.5)。

第8種情形



✕ ---目標區塊上之影像

(a) 2×2 之目標影像



● ---參考畫格上之像素

(b) 半像素精度之動態推測

圖 7.18　動態推測之精度與像素配置

在此，爲了簡化，以圖 7.18(a)所示之尺寸爲 2×2 區塊爲比對之目標區塊。在同一圖(b)，爲了易於了解，當v_x與v_y爲整數的時候((A)的情形)，比對區塊內的像素╳與參考畫格上的像素●重合。另外一方面，在v_x與v_y至少有一方爲整數＋ 0.5 的情形中((B)-(D))，**像素╳會落在●之間**。因此，爲了計算比對誤差，須計算參考畫格上像素的中間點—對應╳的點—之內插值。

這種像素之內插法有不同的方法。實際上圖 7.19 所示之**雙重一次內插法**被廣泛地使用。在此方法中，首先計算內插點(x,y)之小數部份$h=x-\lfloor x\rfloor$、$v=y-\lfloor y\rfloor$，如圖 7.19 所示使用圍住內插點(x,y)的四個像素值$f(i,j)\sim f(i+1,j+1)$，可以下列方式求得內插像素值$f(x,y)$：

$$f(x,y)=(1-h)(1-v)\cdot f(i,j)+h(1-v)\cdot f(i+1,j)+(1+h)v\cdot f(i,j+1)+hv\cdot f(i+1,j+1) \quad (7.13)$$

特別是如同圖 7.18(b)的情形(C)，可用下列方式求得四個像素中央之像素值[96]：

$$f(x,y)=\frac{f(i,j)+f(i+1,j)+f(i,j+1)+f(i+1,j+1)}{4} \quad (7.14)$$

註 96. 對於精通多頻訊號處理的讀者來說，以「進行兩倍的增加取樣後，多相處理衝擊響應為[0.25 0.5 0.25]之分離式二維FIR濾波器」加以說明說不定更容易理解。

實際的 Half-pel ME 爲對於搜尋範圍S內所有的候補點在必要時以式(7.13)進行像素值的內插，同時測量比對誤差E，以搜尋可得到最小值之向量[97]。

註 97. 將此方法加以擴張，舉例來說可以進行精度為 1/4 像素等 ME。

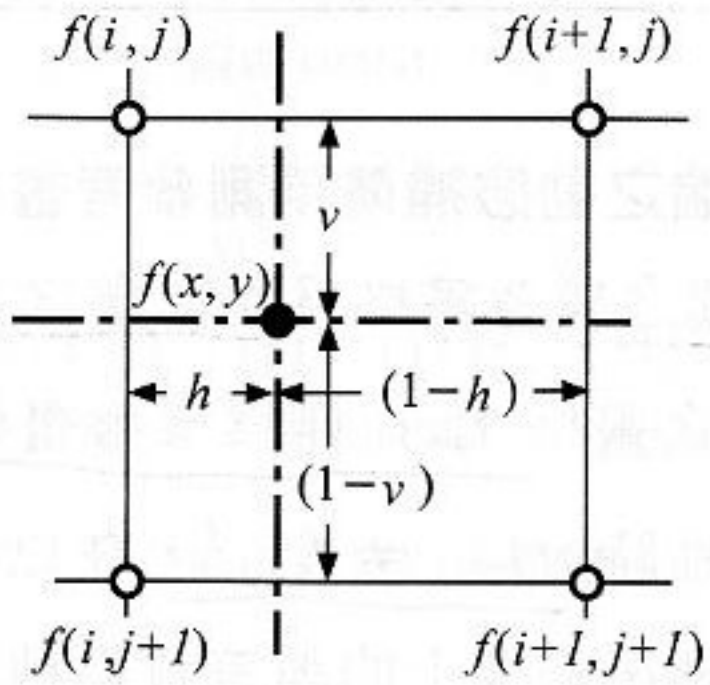


圖 7.19　雙重一次像素內插

圖 7.20 爲對圖 7.4(a)、(b)進行 Half-pel ME／ME 所求得之差分影像。與進行 Full-pel ME 之圖 7.14(c)相較，可以更加減低殘留的預測誤差。

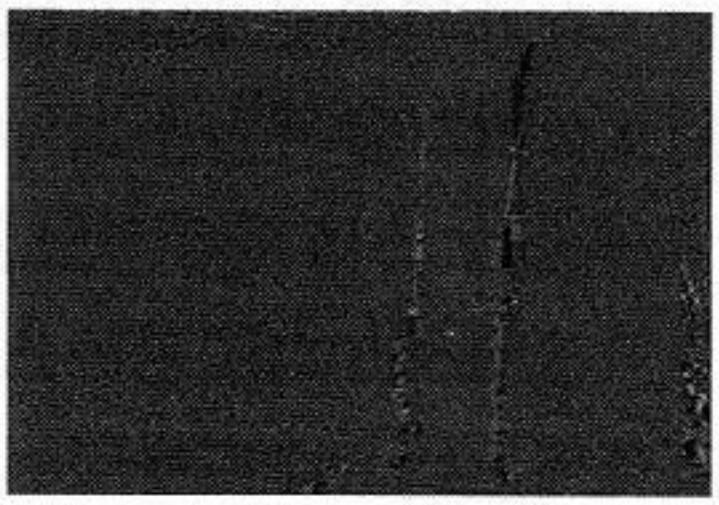


圖 7.20　對圖 7.4 進行 Half-pel ME／ME 之結果

實際的演算法

以動態補償為基礎之實際編碼演算法

在圖 7.9 所表示之基本編碼／解碼演算法中，仍然有接下來三個未解決的課題。

① 在開始編碼之後，最初的畫格沒有參考畫格，因此無法進行畫格間編碼。

② **在編碼端與解碼端之動態補償預測參考畫格不同**，也就是說，在圖 7.9 中，編碼端之參考畫格爲原影像之畫格 f_{n-1}，另外，解碼端係以爲解碼後之畫格 f_{n-1}''作爲參考畫格。一般當編碼／解碼過程不爲非破壞性時，$f_{n-1} \neq f_{n-1}'$，在以$n \leftarrow n+1$進行編碼／解碼的過程中，解碼後影像上的誤差會累積，這種現象稱爲漂移。

③ 在目標畫格上存在有無法進行 ME／MC 之 MB。

要解決①非常地容易，在最初畫格僅進行畫格內(內部)編碼即可。雖然不可免地會造成效率的降低，但是因爲畫格內編碼僅於最初的第一個畫格進行，隨著編碼的進行整體的編碼效率會趨近於畫格間編碼。

另外，雖然有些超前，在此稍微提及刷新的概念，在對畫格間編碼後資料進行解碼時，其前提係對該參考畫格進行解碼，當某畫格在解碼時發生錯誤，會發生什麼情況？不僅是參考該畫格的其他畫格無法解碼，而之後的所有畫格也無法解碼。爲了避免此問題，在實際上以**適當的間隔插入畫格內編碼之畫格**，以去除畫格間的依賴關係，依此作爲抑制誤差擴大的對策。這種處理稱爲**刷新**(圖 7.21)。

對於課題②，當編碼後的畫格在編碼端進行解碼的結果，與在解碼端所得到的是完全相同的影像即可解決。爲此，在編碼器內部設有解碼器，將解碼後影像儲存在畫格記憶體中即可，爲了此目的所設置之解碼器稱爲**區域解碼器(local decoder)**。

至於課題③之對策爲，對預測殘留誤差較大之 MB 進行畫格內(intraframe)編碼。在此情況中即使在進行畫格間編碼的畫格中，仍存在有僅進行畫格內編碼之區塊，需有編碼模式的切換。

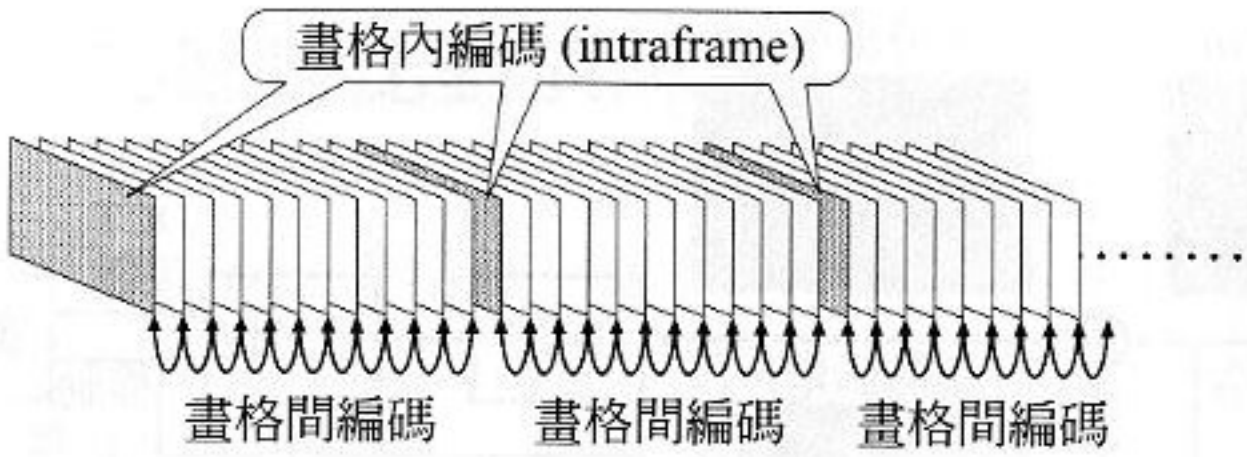


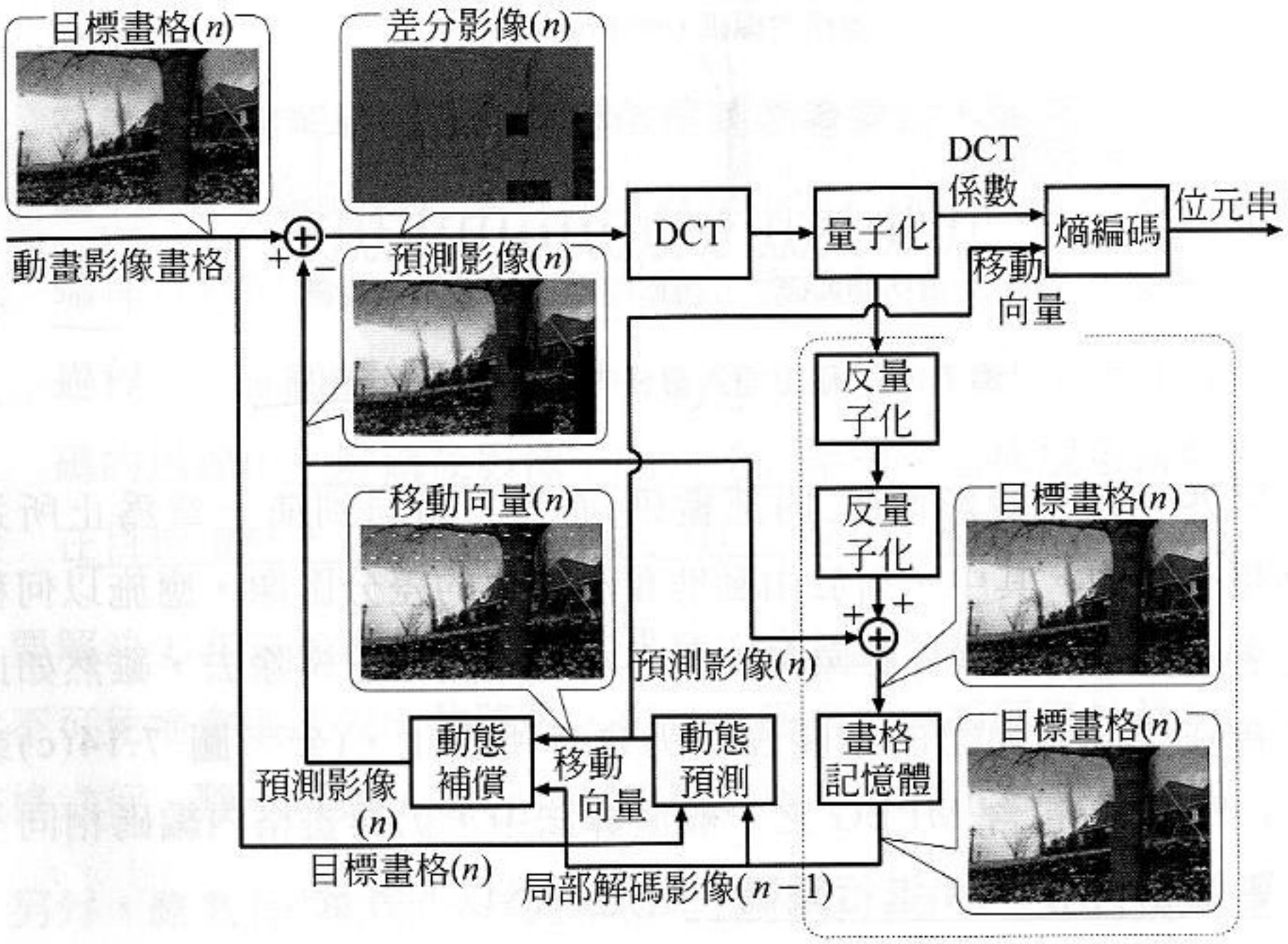
圖 7.21　藉由插入畫格內編碼畫格進行刷新

另外，對一般影像(之內部編碼)而言，如同到前一章爲止所述，DCT是有效的。其中，對於由動態預測產生的差分影像，應施以何種轉換呢？這就是說，差分影像在時間方向的相關性已經除去，雖然如此一來仍會在空間方向上的相關程度仍會有所變化，(參考圖 7.14(c)或圖 7.20)。但是，包含 MPEG 之一般演算法中，仍與畫格內編碼相同，**對差分影像亦利用 DCT 進行編碼**。這是因爲以下考量：

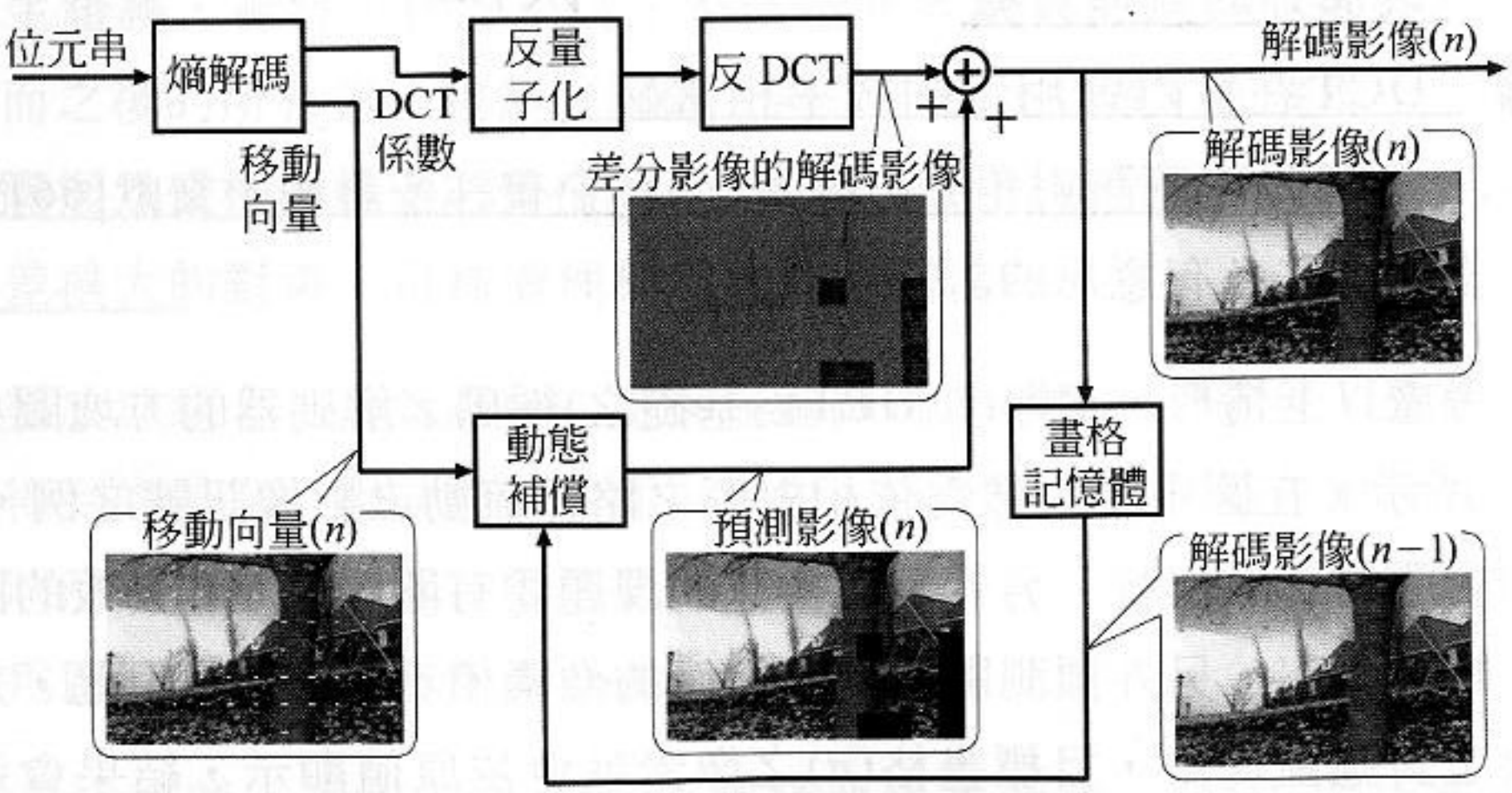
- 差分影像與畫格內編碼影像適用不同的轉換，在實施之時會造成額外的處理負擔。
- DCT 在實際運用上的效率相當好。

(實際上對於是否應該對差分影像進行 DCT 有許多討論，文獻[66]對於其正當性有許多有意思的討論。)

考慮以上情形，實際(以DCT爲基礎之)編碼／解碼器的方塊圖如圖 7.22 所示，在圖中的影像爲依相對應之路徑流動之影像訊號之例，(n)爲相對應的影像編號。另外，與上述的課題③有關，在 ME 失效的區塊無法賦予 MV，另外預測影像中，對應的像素值爲零(全暗)因此，差分影像之對應區塊中，目標畫格(n)之像素就會依原值顯示，結果會適用畫格內編碼。另外，在解碼時，需要各MB之畫格內／格間的資訊(編碼模式)，不用說這些包含在位元串裏頭。



(a)



(b)

圖 7.22　以 MC 與 DCT 為基礎之編碼／解碼器的方塊圖

實際的區塊比對演算法

設想對尺寸爲$W \times H$之畫格進行區塊尺寸爲 16×16 之ME之情況。搜尋範圍S爲水平垂直方向皆爲$\pm s$像素，在S裏頭進行精度爲整數之**全面搜尋**，其加減運算之總數可表爲：

$$2 \times 16^2 \times (2s+1)^2 \times \left(\frac{W}{16}\right) \times \left(\frac{H}{16}\right) \tag{7.15}$$

舉例來說 352×240 的 CIF 影像對每一畫格進行$s = 16$ 之 ME，約需要一億八千四百萬次的加減法運算。對於每秒三十畫格之動畫影像而言，每秒需要其三十倍之運算，在導入後述之 B 畫格後 MV 的搜尋範圍S必須取更大值。另外，Half-pel MP 使用後，這樣也會造成運算量的大幅增加。因此，爲了即時地以BM法進行動態推測，降低計算量的措施是不可或缺的。

降低計算量之**快速搜尋演算法**，雖在過去已有相當多的研究報告，但是基本上皆基於以下想法：

① 省略S內之候補點，降低比對的次數。

② 導入比對誤差曲面之方法，判斷無法得到在某個數值以上的改善之後，會進行中斷搜尋等處理。

③ 在移動物體之移動爲等速移動之假設之下，預測參考畫格中的移動物體的位置，在那附近進行詳細的搜索。

④ 將影像的對比程度調低，以減少表示像素值之位元長度。

其中最爲廣泛利用的是①的處理方式，圖 7.23 爲稱爲**三段搜尋法**[46]之代表性的的方法。搜尋範圍爲垂直水平方向皆爲$\pm s$像素的情況中，演算法可歸納如下。

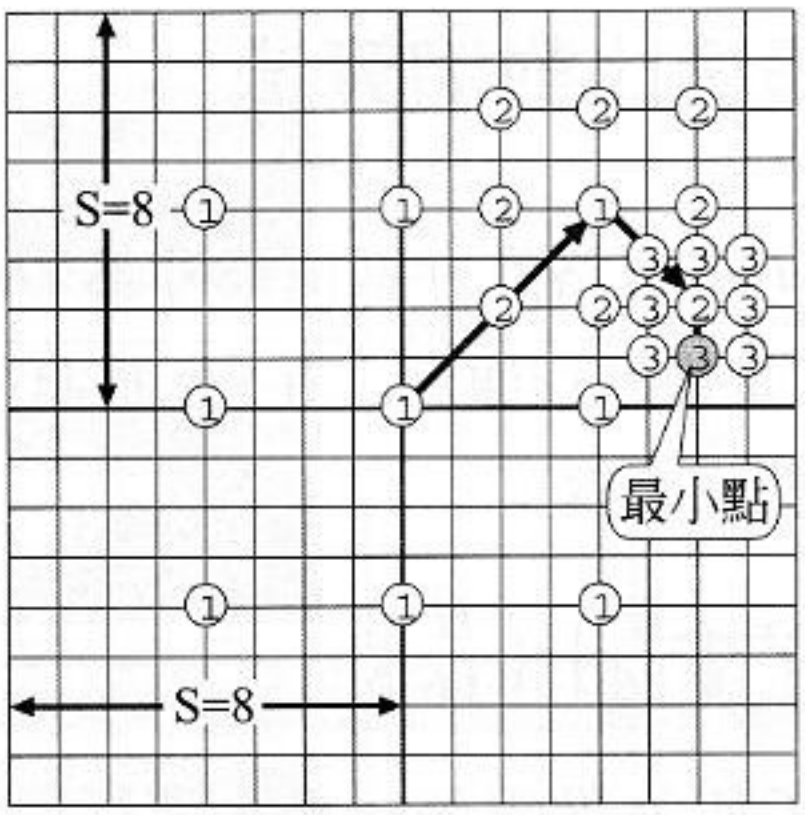


圖 7.23

步驟一：以候補點(0,0)爲中心，測量(±s/2，±s/2)共計九個點(圖 7.23 中①的九個點)之比對誤差，移至最小的點(此處(4,4)爲最小點)。

步驟二：以步驟一之最小點爲中心，接著以求取(±s/2，±s/2)八個點之比對誤差，同樣地移至最小點(此時爲(6,2))。

步驟三：同樣地，以步驟二之最小點爲中心進行搜尋，以其中的最小點(在此情況下爲(6,1))取相對應之 MB 與 MV。 分區

其他的高速搜尋演算法或是各方法之性能比較等在文獻[47]等之中有所討論。

8章

資料率－失真理論－影像編碼之理論處理入門

對於「為什麼影像可以壓縮」的疑問，在第二章中給了「因為像素間具有高度相關」的解答。但是，關於其正當性，僅給了自協方差函數或是 KLT 組成等定性的理由。在本章中，使用稱為「資料率－失真理論」之觀念，對於上述的解答給予明確的正當性。藉此在了解像素間相關程度與可壓縮性間關係的同時，評估靜止影像或是動畫影像的壓縮極限。

但是很可惜的，由於頁數限制，本章中僅能就系統的理論進行入門程度的淺釋，但是這種處理仍可使吾人對於影像編碼有更深刻的洞察。

另外，沒有必要處理這些理論的讀者，可以跳過本章進入第九章。

資料率－失真理論之簡介

考慮如圖 8.1 之一組編碼／解碼器。在評估其性能時，我們使用了每像素之位元量 bpp 與 PSNR。前者所對應爲編碼之**資料率**R，後者爲以平均平方差計算的**失真**D。這兩者以圖 8.2 之圖形表示，可以達到更

小的資料率R與更小的失眞D(大的PSNR)的編碼／解碼器可稱爲「高性能」。

現在，考慮對目標影像以數個編碼器或是編碼演算法加以編碼的情形。如圖 8.2 之虛線，將 PSNR 固定在某數值的條件下，「最好的」編碼演算法是否爲資料率R爲無限小者？以工程學的角度來看其答案爲「否」。不管是誰，在失眞(PSNR)在固定狀態下，自然的影像無法都壓縮到無限小。當然不僅限於影像，包含聲音等所有訊號以波形編碼時，這都是共通的事實，當**失眞D保持在某個數值以下的狀態，其資料率R無法低於某個臨界值R^***。其臨界值R^*係與編碼對象(影像)有關，而爲與編碼／解碼演算法無關的量，也可想成是「只有用神才能設計出的究極編碼／解碼器進行編碼的情形下之臨界值」。

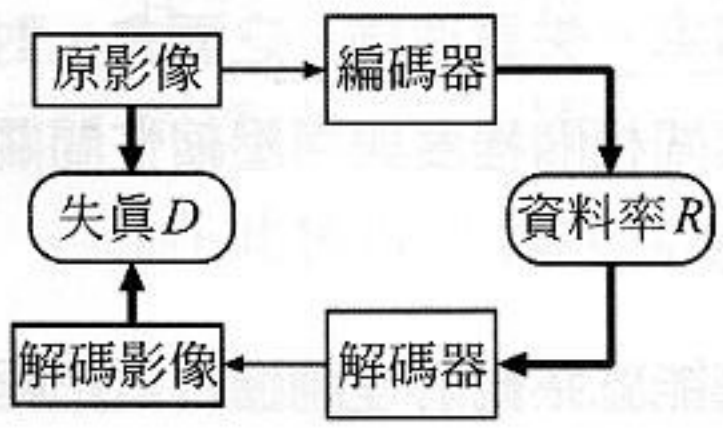


圖 8.1　一組編碼／解碼器

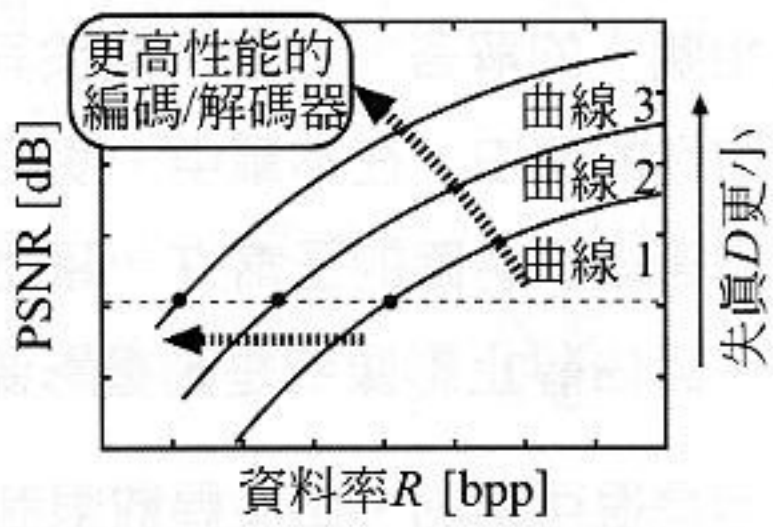


圖 8.2　「好的」編碼／解碼器

資料率－失真理論係賦予此種編碼極限之理論，可以得到的極限資料率與失眞間之關係稱爲**資料率－失真函數**。非破壞性編碼之編碼資料量的下限稱爲**熵**，可用理論加以推得(第四章)，資料率－失眞函數可想作破懷性編碼中與熵對應的量。

資料率－失真函數

在此，僅表示結果。對於某個一維訊號 $f(x)$之資料率－失眞函數可以寫作以下形式。

$$D(\theta)=\frac{1}{2\pi}\int_{-\pi}^{\pi}\min[\theta\,,S_{ff}(\omega)]d\omega \tag{8.1}$$

不明白

$$R(\theta)=\frac{1}{4\pi}\int_{-\pi}^{\pi}\max\left[0\,,\log\frac{S_{ff}(\omega)}{\theta}\right]d\omega \tag{8.2}$$

在此，與導出 KLT 之假設相同，可以先減去 $f(x)$之平均值。$S_{ff}(\omega)$爲 $f(x)$之**自協方差函數**$C_{ff}(u)$ [98]的傅利葉轉換，稱爲**電能頻譜密度函數 (Power Spectral Density Function：PSD)**另外，θ爲正的參數。另外 $\min[x\,,y]$與$\max[x\,,y]$爲表示x與y中較大者與較小者之函數，log 爲對數函數，與熵相同，其底爲 2。

註　98. 因為減去$f(x)$之平均值，$C_{ff}(u)$亦可稱為「自相關函數」，為避免混淆，本章中採與第二章相同之「自協方差函數」。

由式(8.1)、(8.2)所得之$D(\theta)$、$R(\theta)$(單位爲位元)，相當於將訊號 $f(x)$壓縮編碼時的極限，可將資料率－失眞理論描述爲：

> 在將失眞D限制在$D \le D(\theta)$的情況中，其資料率R位元無法小於$R(\theta)$，也就是說$R \ge R(\theta)$爲必要的。

式(8.1)與式(8.2)之推導方法可以參照文獻[48]、[17]、[18]等。

不明

資料率－失真函數之圖解

式(8.1)與式(8.2)中的被積分函數，電能頻譜密度函數$S_{ff}(\omega)$與參數θ之大小關係可以寫作：

$$\min[\theta, S_{ff}(\omega)] = \begin{cases} \theta & (\theta < S_{ff}(\omega)) \\ S_{ff}(\omega) & (\theta \geq S_{ff}(\omega)) \end{cases} \tag{8.3}$$

$$\max\left[0, \log\frac{S_{ff}(\omega)}{\theta}\right] = \begin{cases} \log\dfrac{S_{ff}(\omega)}{\theta} & (\theta < S_{ff}(\omega)) \\ 0 & (\theta \geq S_{ff}(\omega)) \end{cases} \tag{8.4}$$

圖 8.3 所表示者爲自一維訊號 $f(x)$ 求得自協方差函數 $C_{ff}(u)$，進一步以傅利葉轉換得到電能密度函數 $S_{ff}(\omega)$ 之例。首先，請確認在式(8.1)～(8.4)中，將圖中虛線的值**設為參數**θ的情形，相當於：

- 由 $D(\theta)$ 所表示式(8.1)的積分，爲 θ 下方網點部份的面積。
- 由 $R(\theta)$ 所表示式(8.2)的積分，爲 θ 上方斜線部份取 log 之後的面積。

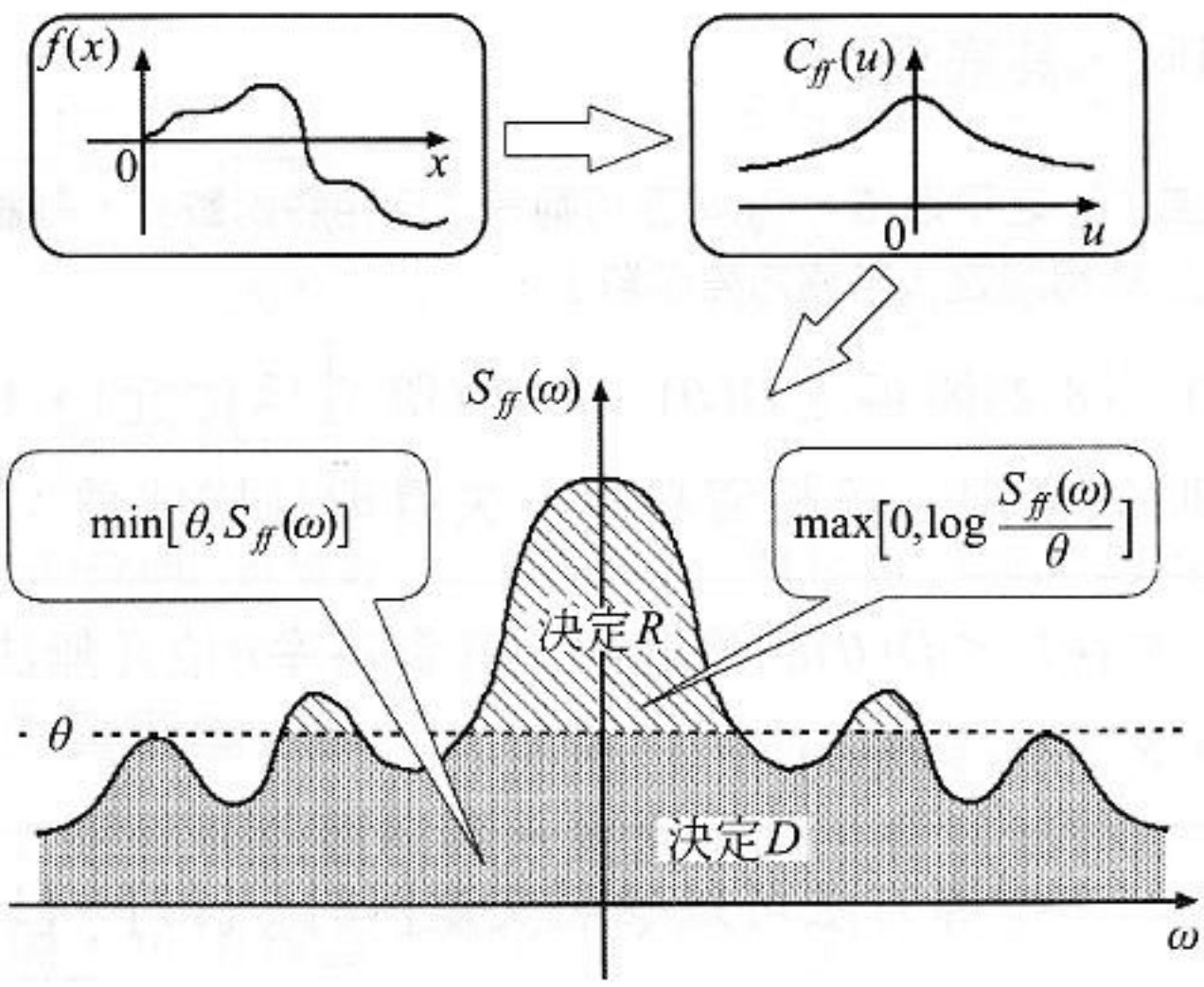


圖 8.3　資料率－失真函數之圖解 1

接著，考慮由圖 8.4(a)與(b)所示之兩個電能頻譜密度函數。當 θ 值以 $\cdots, \theta_1, \cdots, \theta_2, \cdots, \theta_3, \cdots$,變化的同時計算 $D(\theta)$、$R(\theta)$ 之後以失眞爲軸，

可得到(c)中圖形的一條曲線(所謂**資料率－失真曲線**)。另外，要注意的一點是(c)的縱軸取$-10\log_{10}D(\theta)$以分貝表示(因此在圖形上方表示失真較小)。

現在於圖 8.4(a)與(b)中，設定$\theta=\theta_i(i=1,2,3)$評估式(8.2)，假設兩者$R(\theta)$的值相等，此時，各「θ_i以下的面積」明顯得看得出來(a)較大，因此其失真$D(\theta)$較(b)為大。對各θ值進行評估，可求取如同一圖(c)之兩者的資料率－失真曲線。也就是說，資料率相等$\gamma(\theta)$的情形中，對應的失真量為(a)較大，這表示**像(a)這種訊號比(b)更難編碼**(或是說較難有效地編碼)。

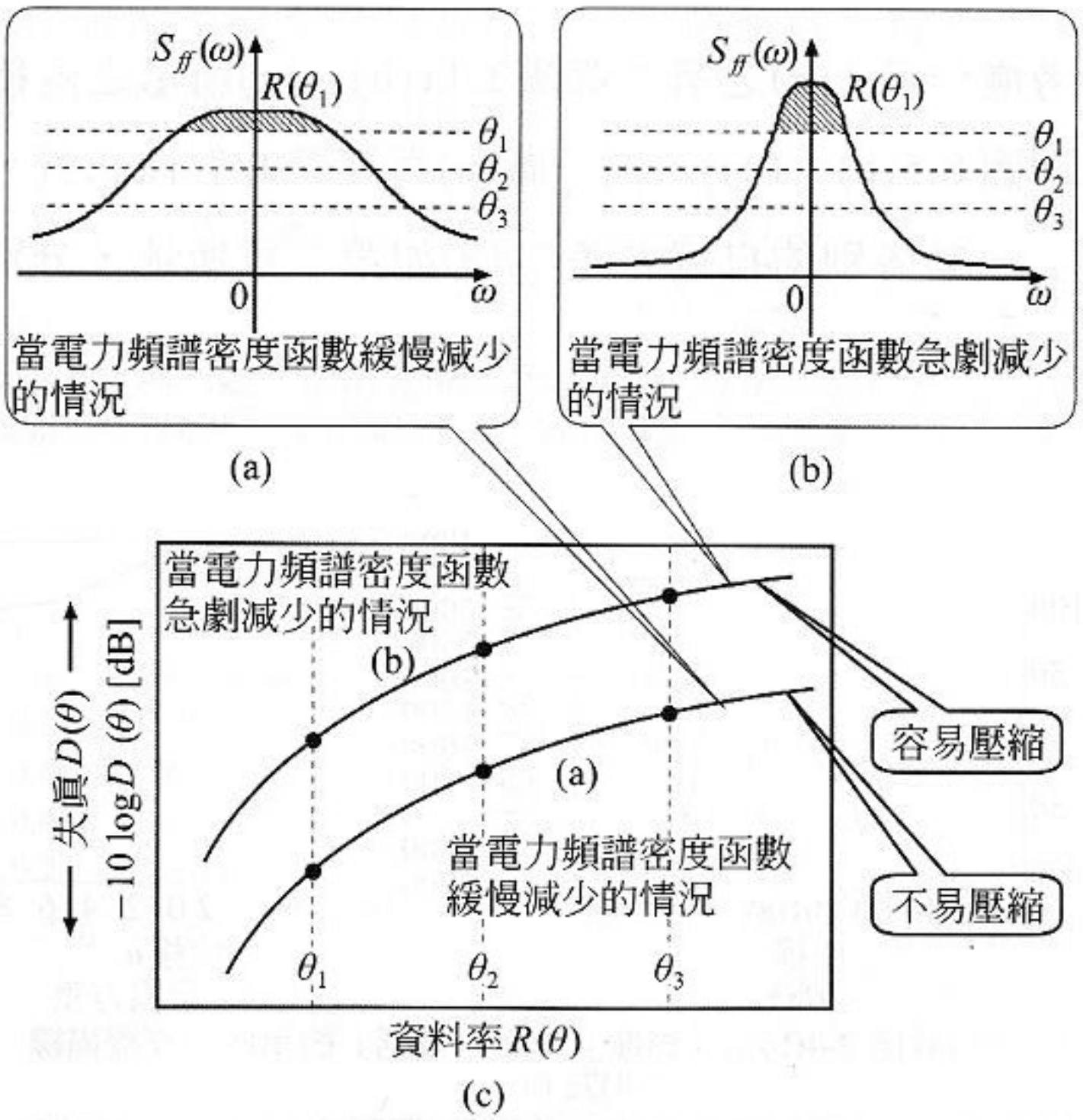


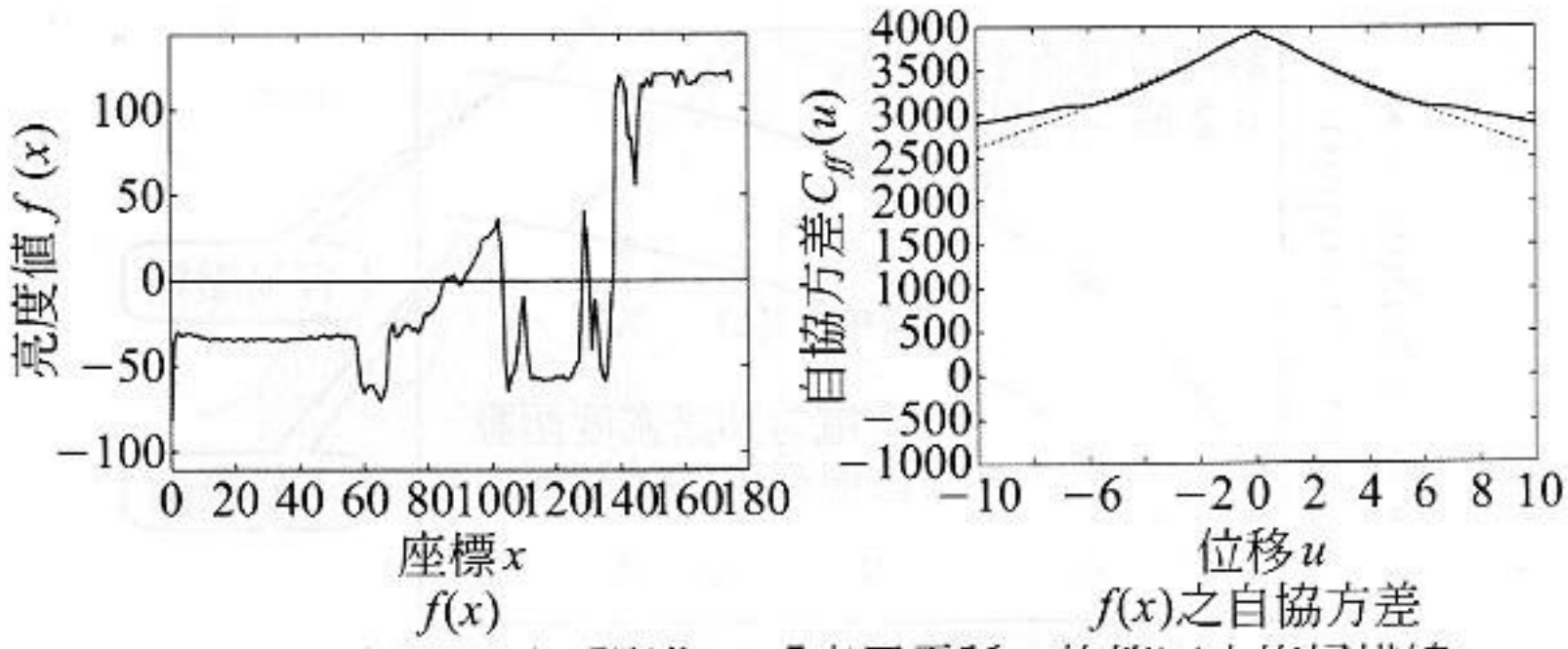
圖 8.4　資料率－失真函數之圖解 2

依據資料率－失眞理論，結論爲**給定訊號之編碼難易度，由其電能頻譜密度函數決定**。也就是說，**電能頻譜密度函數急劇地衰減的訊號較易編碼**。

為什麼可以壓縮影像

在做好準備之後。對於「因爲影像像素間高度相關所以可壓縮」這個答案，以資料率－編碼理論加以考慮正當性。另外，爲了簡化，討論一維訊號時所得的結果可以照樣地擴張至二維訊號。

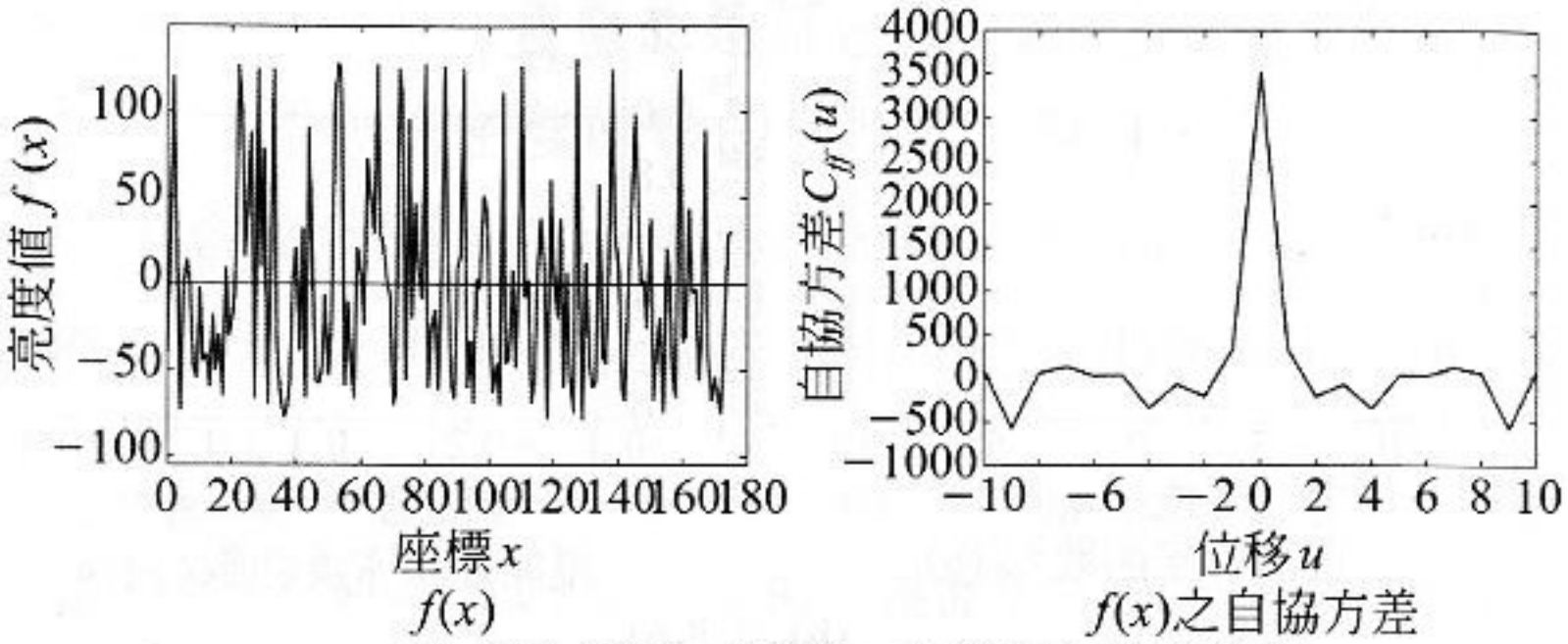
現在，考慮作爲$f(x)$之第二章圖 2.11(b)、(c)所示之兩種波形(減去其平均值，即圖 8.5(a)、(b))。同一圖(a)爲實際的影像訊號，另外(b)相當於「雜音」，其各別的自協方差$C_{ff}(u)$如第二章所述，分別可以$\sigma^2\sigma^{|u|}$及$\sigma^2\delta(u)$加以模式化。



$f(x)$　　$f(x)$之自協方差

(a) 圖 2-4(b)之「影像」「車用電話」的第四十條掃描線

圖 8.5　$f(x)$之例與其自協方差函數(與圖 2.11(b)、(c)相同)



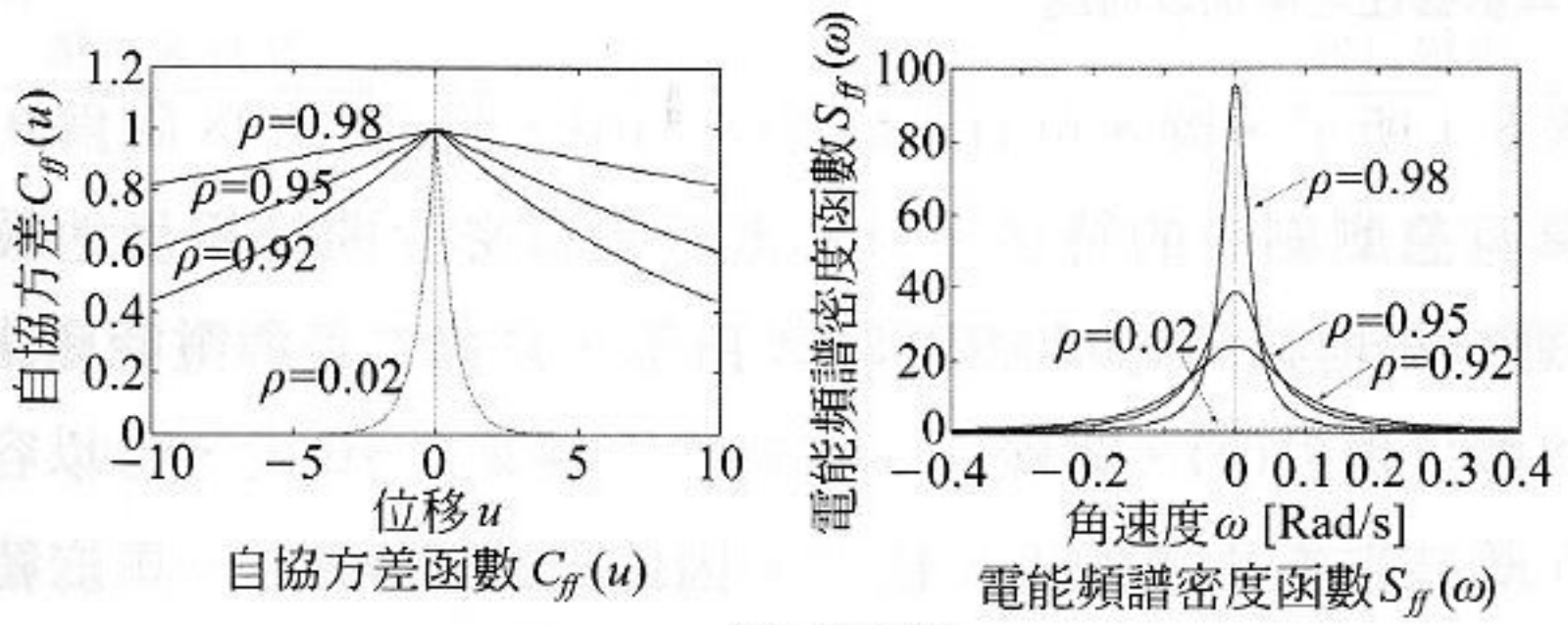
(b) 圖 2-4(c)之「雜訊」的第四十條掃描線

圖 8.5　$f(x)$之例與其自協方差函數(與圖 2.11(b)、(c)相同)(續)

對上述兩種波型模型計算資料率－失眞函數。首先，由兩者之自協方差模型求得電能頻譜密度函數：

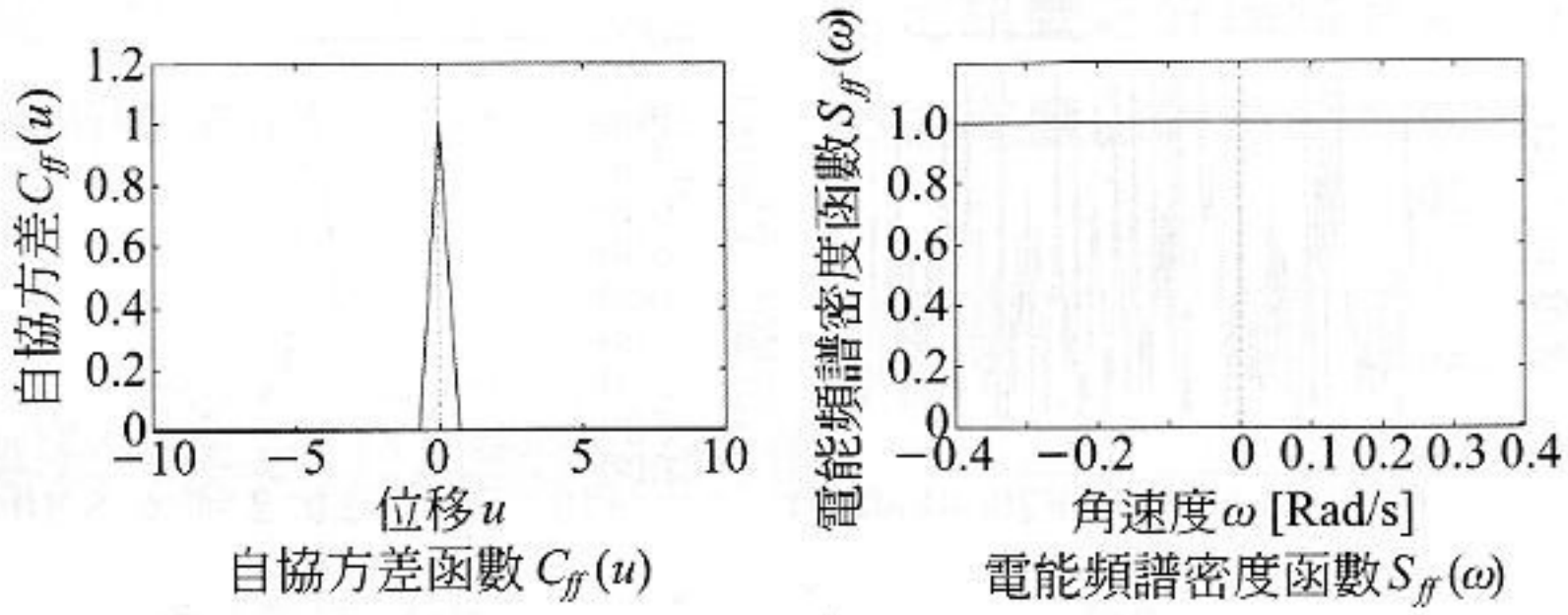
$$F[\sigma^2\rho^{|u|}]=\sum_{u=-\infty}^{\infty}\sigma^2\rho^{|u|}e^{-j\omega u}=\sigma^2\frac{1-\rho^2}{1-2\rho\cos\omega+\rho^2} \tag{8.5}$$

$$F[\sigma^2\delta(u)]=\sum_{u=-\infty}^{\infty}\sigma^2\delta(u)e^{-j\omega u}=\sigma^2 \tag{8.6}$$



(a) 式(8-5)

圖 8.6　式(8.5)與(8.6)之自協方差函數$C_{ff}(u)$與電能頻論密度函數$S_{ff}(\omega)$



(b) 式(8-6)

圖 8.6　式(8.5)與(8.6)之自協方差函數$C_{ff}(u)$與電能頻論密度函數$S_{ff}(\omega)$(續)

表 8.1　ρ對自協方差函數$C_{ff}(u)$與電能頻論密度函數$S_{ff}(\omega)$的影響

	自協方差函數	電能頻論密度函數
$\rho = 0.98$	緩慢地減少	急劇地減少(對應圖 8.4(b))
$\rho = 0.92$	急劇地減少	緩慢地減少(對應圖 8.4(a))

之後爲了簡化，將訊號 $f(x)$之離散程度σ^2正規化爲 1 [99]。式(8.5)與式(8.6)之自協方差函數(左邊)與電能頻譜密度函數(右邊)之圖示爲圖 8.6。

註　99. 其影響在之後加以討論。

如表 8.1 所示，圖 8.6(a)或是式(8.5)中，將ρ由 0.98 降爲 0.92，自協方差具有急劇劇少的特徵，另外電能頻譜密度則是相反地緩慢地減少。將ρ進一步地減到 0.02，如圖 8.6 所示，自協方差與電能頻譜密度函數的形狀都會趨近(b)。實際上，當式(8.5)中取$\rho \to 0$ 後，可以容易地確認式 8.6 等式左右兩邊的近似值[100]。因此，式(8.6)在$\rho \to 0$ 之特殊情況下，可由式(8.5)可以涵括。

註　100.請自行嘗試。

因此在圖 8.6(a)中注意電能頻譜密度。可以得知，當ρ爲 0.98 之接近 1 的值時，電能頻譜密度函數的形狀接近圖 8.4(b)，反之，$\rho = 0.92$ 其值降低時，會接近圖 8.4(a)。因此，如表 8.2 的圖式成立，藉資料率－失眞函數，可以說明「像素間高度相關時易於壓縮」的性質[101]。

註　101.M生的疑問：「但是老師，舉例來說圖 8.6 之電能頻譜密度函數之圖形以$\theta = 20$ 加以分割，比較$\rho = 0.98$ 與$\rho = 0.92$，確實可以知道$D(\theta)$在$\rho = 0.98$ 時較小，但是求$R(\theta)$，在$\rho = 0.98$ 的圖形在$\omega = 0$ 時就往上延伸，其資料率$R(\theta)$不會變得很大嗎？」

Y師的回答：「是這樣的嗎？所以在 8-4 頁就強調『$R(\theta)$為θ上方之斜線部份之面積取 log 值』，因為取了 log 值，因此會被壓縮使得數值不會太大。」

表 8.2　像素間相關與壓縮難易度的關係之推導圖式

像素間相關	高	低
	⇩	⇩
相關係數值	接近 1	小
	⇩	⇩
電能頻譜密度函數	急劇減少(圖 8.4(b))	緩慢減少(圖 8.4(a))
	⇩	⇩
資料率－失眞曲線	位於上方	位於下方
	⇩	⇩
壓縮難易度	易於壓縮	難以壓縮

另外，由資料率－失眞理論，可以進一步地推導「**在所有的波形 $f(x)$中，沒有相關性之雜訊—正確地說是高斯雜訊—是最難壓縮的**」之性質[48]。換言之，當失眞在一定值以下，對所有波形進行編碼的情況中，**高斯雜訊需要最高的資料率**。這代表，可以想成**高斯雜訊「無法壓縮」**。另外，在影像的情形中，可說近似$\rho \approx 1$，因此像素間高度相關，因此較易壓縮。因此，該性質說明了「爲什麼可以壓縮影像」，「因爲像素間高度相關」。

靜止影像與動畫影像之壓縮極限

如之前所述，資料率－失眞函數提到了對於在某個容許失眞值D下壓縮波形所能達成之最小資料率R。在本節中，利用此性質求得靜止影像或是動畫影像之壓縮極限。藉由對於數個相關係數ρ的值計算其實際的資料率與失眞關係，可以定量地求得ρ與壓縮難易度的關係，也可具體地印證前一節中的討論。

靜止影像之壓縮極限

到前一節爲止之討論可以簡單地擴張，如影像之二維訊號$f(x,y)$，式(8.1)、(8.2)之資料率－失眞函數可以下列形式加以二維化：

$$D(\theta)=\frac{1}{4\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\min\left[\theta,S_{ff}(\omega_1,\omega_2)\right]d\omega_1 d\omega_2 \tag{8.7}$$

$$R(\theta)=\frac{1}{8\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\max\left[0,\log\frac{S_{ff}(\omega_1,\omega_2)}{\theta}\right]d\omega_1 d\omega_2 \tag{8.8}$$

另外，影像之自協方差函數$C_{ff}(u,v)$可以式(2.10)或是式(2.11)加以模式化，再以**二維傅利葉轉換**求得電能頻譜密度函數$S_{ff}(\omega_1,\omega_2)$。將此代入式(8.7)、(8.8)，可以求得對「影像」之資料率－失眞函數。

利用以上步驟求取靜止影像之壓縮極限，具體的順序如以下所示。

① 將影像之自協方差函數$C_{ff}(u,v)$以式(2.11)之$C_{ff}(u,v)=\rho^{|u|+|v|}$加以模式化(將離散程度$\sigma^2$正規化爲 1)。

② 計算二維傅利葉轉換後之電能頻譜密度函數$S_{ff}(\omega_1,\omega_2)$，使用式(8.5)可得：

$$F[\rho^{|u|+|v|}] = S_{ff}(\omega_1, \omega_2)$$
$$= \frac{1-\rho^2}{1-2\rho\cos\omega_1+\rho^2} \cdot \frac{1-\rho^2}{1-2\rho\cos\omega_2+\rho^2} \tag{8.9}$$

③　代入式(8.7)、(8.8)，依照數值積分公式加以積分。

④　改變參數$\theta > 1$，重複進行步驟③。

圖 8.7 爲二維的自相關函數(2.11)與電能頻譜密度函數的形狀，所繪出爲$\rho = 0.92$與 0.98 的情況(各自對應到圖 8.4(a)與(b)中的二維情況)。圖 8.8 爲求得的靜止影像之壓縮極限。

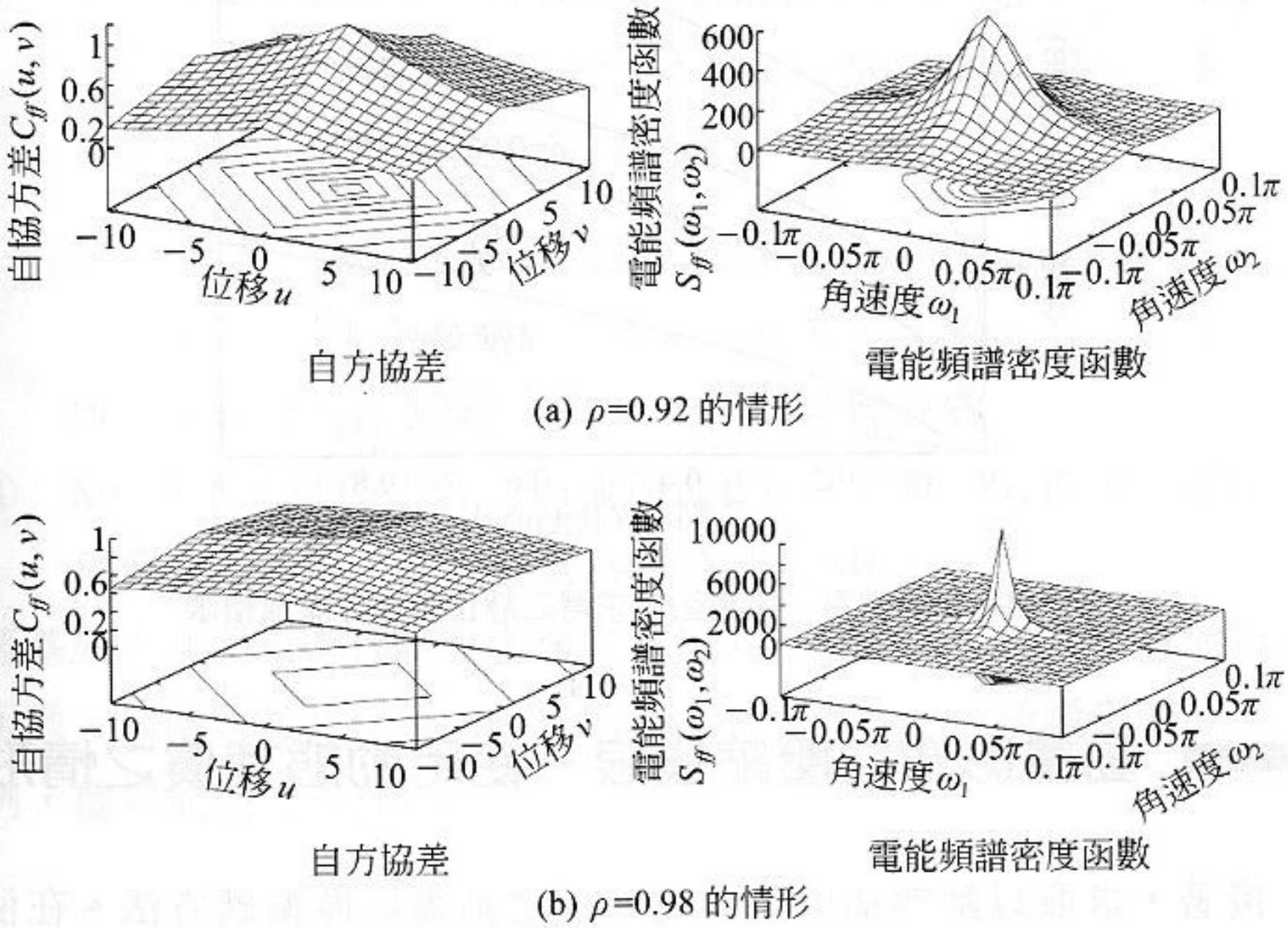


圖 8.7　自協方差與電能頻譜密度函數關係之實際形狀

當ρ值接近 1 時 0bpp 附近會急劇地升高，到了 1bpp 之後，斜率爲 6dB/bit上升[102]。另外，爲了顯示比較結果，高斯雜訊通常是以 6dB/bit 傾斜[103]。兩者的差忠實地表現了「壓縮的難易程度」。另外，在求實際

影像之資料率－失真函數時，除了ρ之外，還需將分散程度ρ^2的測量值代入式(8.7)、(8.8)。另外，縱軸係表示PSNR，注意需要先給定偏移值$10\log_{10}255^2$。

註 102.實際上，與編碼器無關，這個較高的值為6.02dB/bit係其特徵，附帶一提，這個性質可在檢查自己編寫的影像壓縮程式之動作時使用。

103.其數值係將高斯雜訊之電能頻譜密度函數$S_{ff}(\omega_1, \omega_2) = 1$代入式(8.7)與式(8.8)而得到的，在理論上，其順序是相反的。

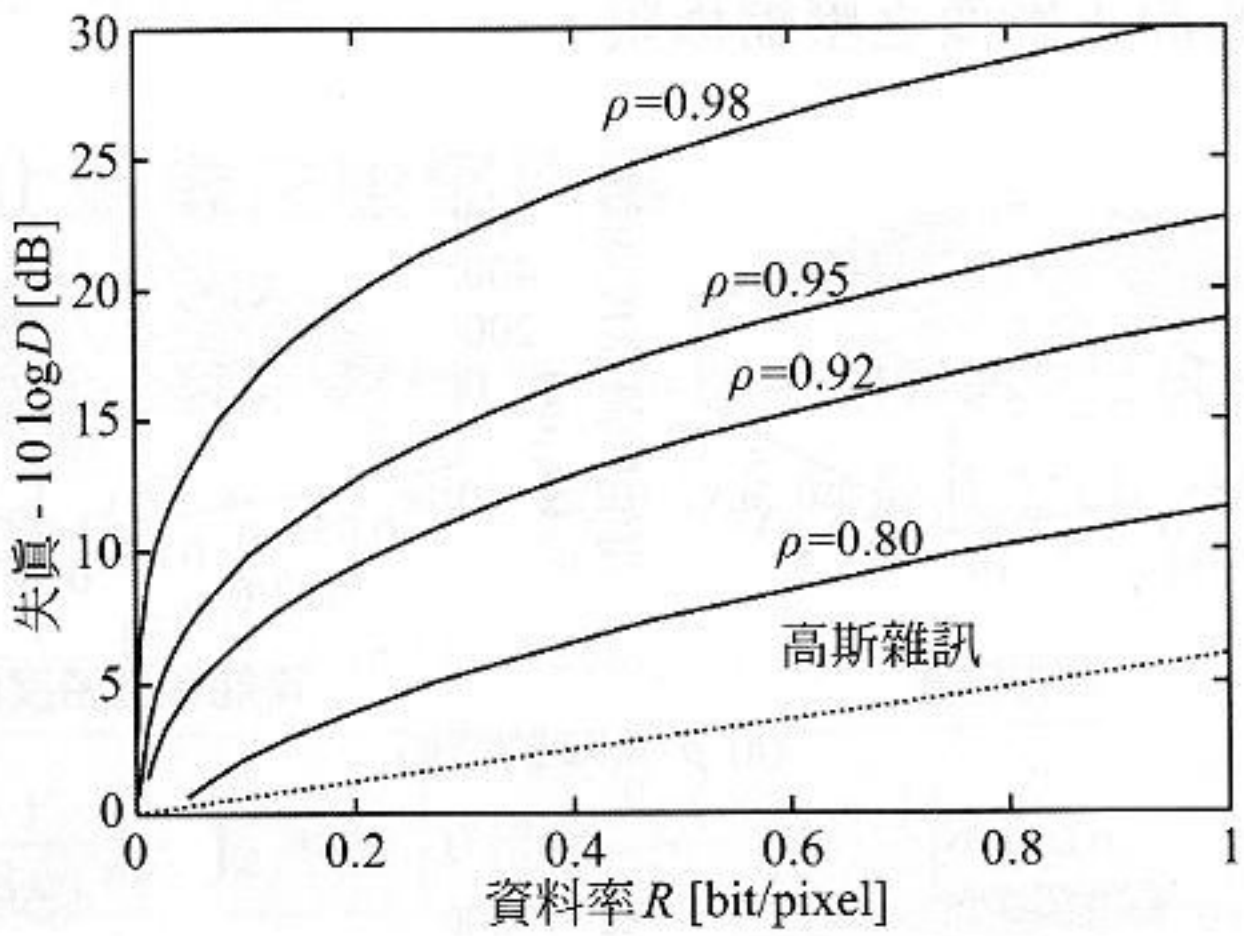


圖 8.8 由資料率－失真函數求得之靜止影像之壓縮極限

動畫影像之壓縮極限－使用動態補償之情形

接著，求取以動態補償預測爲基礎之動畫影像編碼方法。在使用MC的情形中，實際的編碼對象並非爲各動畫影像之畫格，而是該畫格與其預測影像之誤差—無法由預測推估之部份。因此，爲了以資料率－失真理論計算此壓縮極限，需先求得與預測誤差影像之自協方差函數近似的函數。

一般來說，動畫影像畫格內，各部份的移動量是不一致的，另外由於變形或是未涵蓋領域的影響，存在有使MC無法發揮功能的部份，要得到預測誤差影像之自協方差函數的近似結果是非常困難的。因此，爲了簡化問題，需做下列假設。

① 將各畫格之自協方差函數可用式(2.11)加以近似(在此也將離散程度σ^2正規化爲 1)。

② 在畫格間存在之移動量可**全部近似為平行移動**。其他的**旋轉或放大縮小、物體的變形等均不存在**。另外，沒有未涵蓋領域產生。

③ 在畫格間各部位之平行移動量(實際值)爲(v_x^{real} , v_y^{real})，在如第一三五頁說明之k像素精度($k=0.5,1$)進行ME而得到移動向量(v_x , v_y)的情形中，其推測誤差爲：

$$e_x = v_x^{\text{real}} - v_x \text{，} e_y = v_y^{\text{real}} - v_y \tag{8.10}$$

，係在$[-k/2,k/2]$範圍內的實數。在此，推測誤差e_x與e_y如圖 8.9 所示，假設爲在範圍$[-k/2,k/2]$內的**平均分佈**。

④ MC原本係以尺寸爲 之區塊加以實施，在此假設使用「區塊尺寸爲最小之極限，即以**像素**爲單位進行 MC」。

接著基於上述假設進行誤差影像之自協方差函數，以求得其電能頻譜密度函數。其處理相當複雜，以下所述之計算方法爲解析編碼演算法的典型例，儘可能地加以詳述。

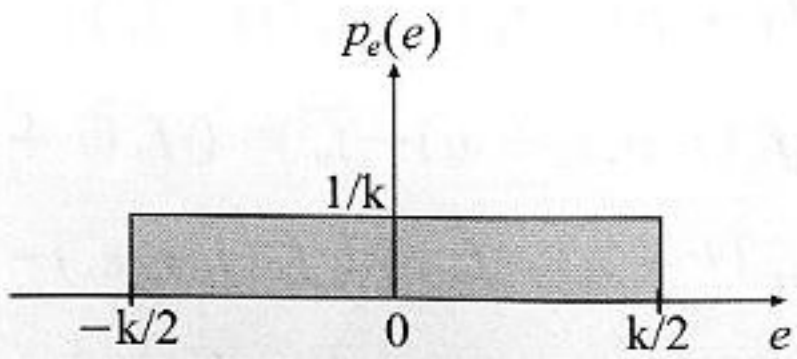


圖 8.9　推測誤差e_x與e_y之分佈

首先，對第n幅畫格$f_n(i,j)$求得預測影像$f_n'(i,j)$，使用假設②與④之推測誤差e_x、e_y寫作：

$$f_n'(i,j)=f_n(i+e_i,j+e_y) \tag{8.11}$$

因此，可得預測影像之差分影像$d_n(i,j)$爲：

$$\begin{aligned} d_n(i,j)&=f_n(i,j)-f_n'(i,j)\\ &=f_n(i,j)-f_n(i+e_x,j+e_y) \end{aligned} \tag{8.12}$$

另外，此處係以影像爲對象，因此所出現的訊號與函數全都是二維的，但是爲了簡化討論，首先將處理對象限定在一維訊號加以計算後，在求壓縮極限時，再將所得結果予以二維化再加以利用。

其中，依照定義求(一維化的)$d_n(i)$之自協方差$C_{dd}(u)$。首先，假設f_n與其預設影像f_n'之空間平均值相等。

$$E_i[f_n(i)]=E_i[f_n'(i)]=\overline{f_n} \tag{8.13}$$

藉此，差分影像$d_n(i)$之空間平均值$\overline{d_n}$爲0。此時，$d_n(i)$之自協方差可寫作：

$$\begin{aligned} C_{dd}(u)&=E_i[d_n(i+u)d_n(i)]\\ &=E_i[\{(f_n(i+u)-\overline{f_n})-(f_n(i+e_n+u)-\overline{f_n})\}\\ &\quad\cdot\{(f_n(i)-\overline{f_n})-(f_n(i+e_x)-\overline{f_n})\}] \end{aligned} \tag{8.14}$$

展開之後，可得：

$$\begin{aligned} C_{dd}(u)&=E_i[(f_n(i+u)-\overline{f_n})\cdot(f_n(i)-\overline{f_n})]\\ &\quad+E_i[(f_n(i+e_x+u)-\overline{f_n})\cdot(f_n(i+e_x)-\overline{f_n})]\\ &\quad-E_i[(f_n(i+u)-\overline{f_n})\cdot(f_n(i+e_x)-\overline{f_n})]\\ &\quad-E_i[(f_n(i+e_x+u)-\overline{f_n})\cdot(f_n(i)-\overline{f_n})] \end{aligned} \tag{8.15}$$

由假設①，可寫作：

$$C_{ff}(u)=E_i[(f_n(i+u)-\overline{f_n})(f_n(i)-\overline{f_n})]=\rho^{|u|} \tag{8.16}$$

式(8.15)可簡化爲：

$$C_{dd}(u)=2\rho^{|u|}-\rho^{|u+e_x|}-\rho^{|u-e_x|} \tag{8.17}$$

但是，此結果僅爲對 1 個推測誤差e_x取自協方差之例。實際上，由假設③可知e_x具有圖 8.9 之分佈，爲了要求平均壓縮極限，需要e_x對取式(8.17)之期望值。也就是說

$$\begin{aligned}C_{dd}(u)&=\int_{-\infty}^{\infty}(2\rho^{|u|}-\rho^{|u+e_x|}-\rho^{|u-e_x|})p_{e_x}(e_x)de_x\\&=2\rho^{|u|}-\int_{-k/2}^{k/2}\frac{1}{k}(\rho^{|u+e_x|}+\rho^{|u-e_x|})de_x\end{aligned} \tag{8.18}$$

爲最後的自協方差函數。

最後，對式(8.18)進行傅利葉轉換求得電能頻譜密度函數$\overline{S_{dd}(\omega_1,\omega_2)}$[104]。依照傅利葉轉換之定義進行實際計算後，得到

$$\begin{aligned}S_{dd}(\omega)&=\int_{-\infty}^{\infty}\overline{C_{dd}(u)}e^{-j\omega u}du=2\int_{-\infty}^{\infty}e^{-j\omega u}du\\&-\int_{-\infty}^{\infty}\int_{-k/2}^{k/2}\frac{1}{k}(\rho^{|u+e_x|}+\rho^{|u-e_x|})de_xe^{-j\omega u}du\end{aligned} \tag{8.19}$$

此處，等式右邊的第一項依照式(8.5)可寫爲$2S_{dd}(\omega)$。另外，右邊第二項利用傅利葉轉換公式，

$$f(u)\leftrightarrow F(\omega)\qquad f(u+X)\leftrightarrow e^{-j\omega X}F(\omega) \tag{8.20}$$

變更積分順序可變形爲：

$$\begin{aligned}\text{第 2 項}&=\frac{1}{k}\int_{-k/2}^{k/2}\int_{-\infty}^{\infty}(\rho^{|u+e_x|}+\rho^{|u-e_x|})e^{-j\omega u}dude_x\\&=S_{dd}(\omega)\cdot\frac{1}{k}\int_{-k/2}^{k/2}(e^{-j\omega e_x}+e^{j\omega e_x})de_x\\&=2S_{dd}(\omega)\frac{\sin(k\omega/2)}{k\omega/2}\end{aligned} \tag{8.21}$$

同時併入第一項，依照先前的規定加以二維化之後可得：

$$\overline{S_{dd}(\omega_1, \omega_2)} = 2\left(1 - \frac{\sin(k\omega_1/2)}{k\omega_1/2}\frac{\sin(k\omega_2/2)}{k\omega_2/2}\right)S_{dd}(\omega_1, \omega_2) \tag{8.22}$$

在此，$S_{dd}(\omega_1, \omega_2)$如式(8.9)所示。

註 104. $\overline{\cdot}$ 為「自協方差之電能頻譜密度函數之期待值」。

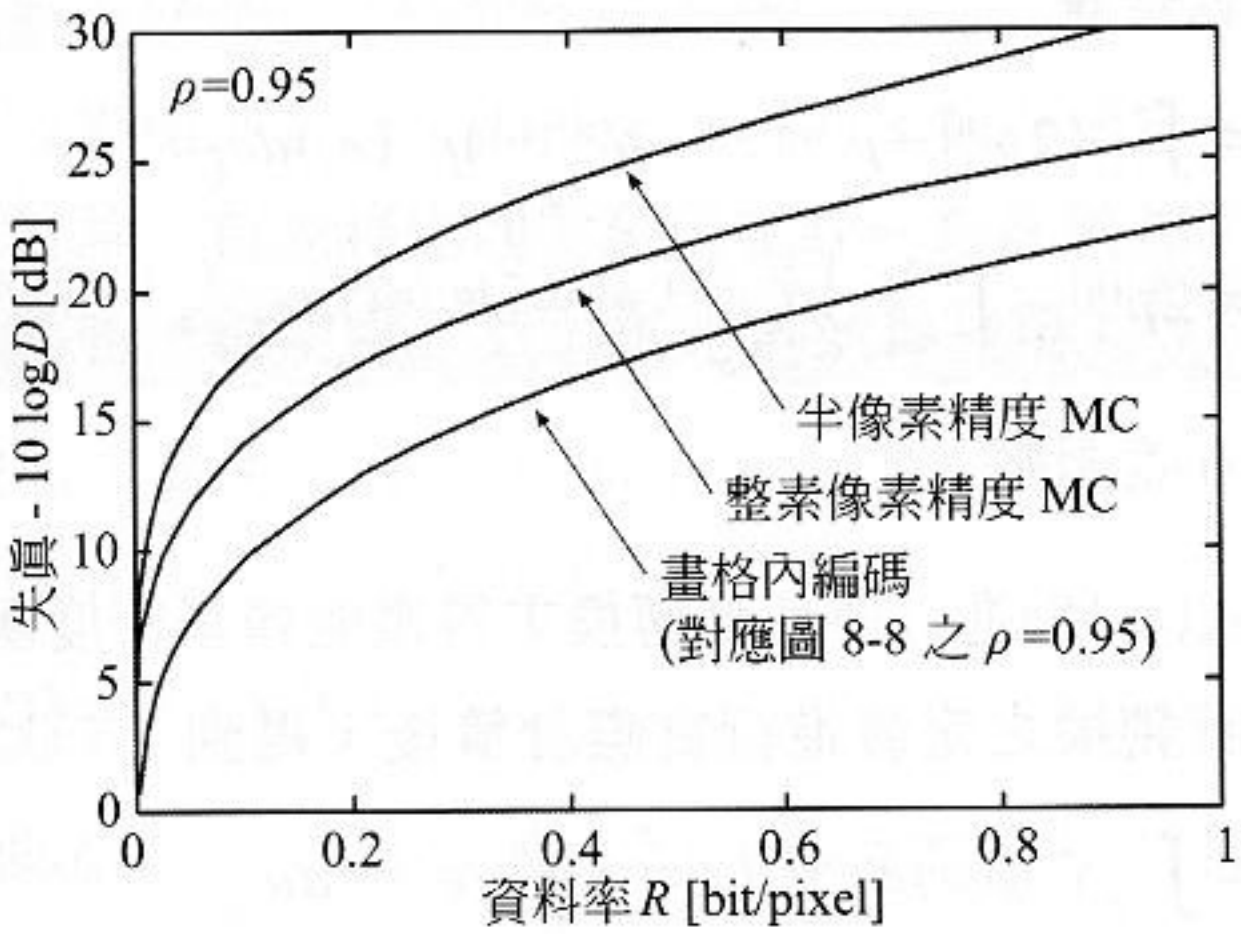


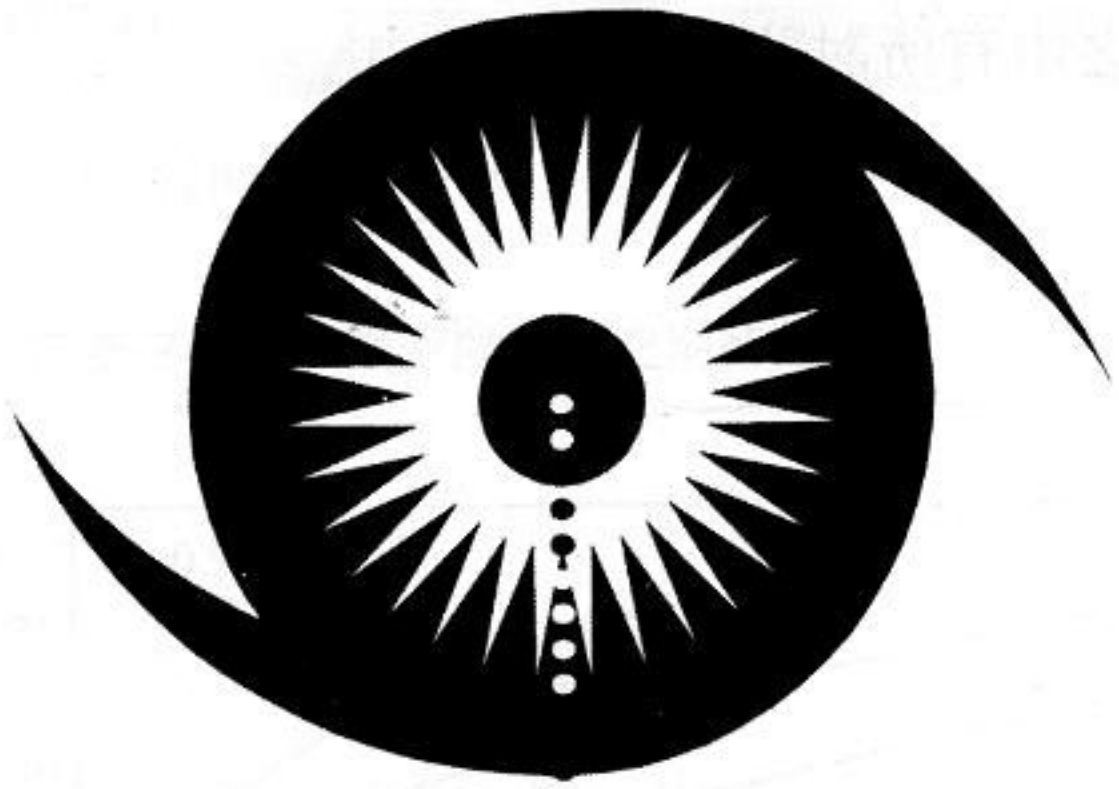
圖 8.10 由資料率－失真函數所求得，以動態補償為基礎之壓縮編碼極限

代入式(8.22)之資料率－失真函數，進行與前一節相同之數值積分，可得到圖 8.10 之結果。藉此，比較提升MC之精度與畫格內編碼相較，可以確認編碼效率有大幅改善[105]。

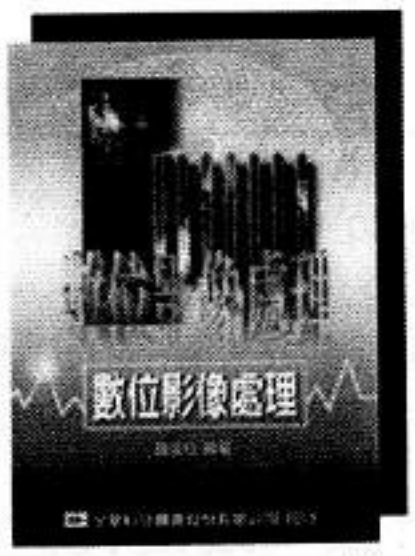
註 105.實際上在未滿 1 像素之 MC 會伴隨著內插時處理，由於此影響，在四分之一像素精度程度時，效率是最好的。

在本節爲了要儘量簡化討論，在使用第 8-13 頁之假設①～④理想化問題的同時，自式(8.14)起一連的計算也導入種種假設。因此，所得到的圖 8.10 僅是說明在導入MC之後的改善效果，實際上在並不是配合

現實的編碼條件。在文獻[49]、[50]等之中，表示更詳細的解析方法與解析結果。有興趣的預可可以參考。另外關於資料率－失眞之應用，在文獻[51]、[52]之中有所討論可供參考。



深入剖析影像處理



數位影像處理

編號03501／蕭進松編著／20 K／280頁／300元

本書從影像概論出發，讓讀者先有一基本觀念，接著介紹各種影像轉換方法，再來是影像的增強及復原，利用不同濾波器得到不一樣的效果，最後爲影像分割及壓縮，以二值及連續壓縮技術來討論影像壓縮方法及標準，讀完本書後，對於影像分類處理、編碼、強化及復原各方法，有清楚的概念。適用於大專院校之電子、電機及光電工程科系之學生，及對影像處理有興趣之資訊處理工程人員作爲研究參考用。

9 章

多媒體編碼實務(1)JPEG－靜止影像之壓縮編碼

JPEG為靜止影像編碼之國際標準，如同衆所周知的，廣泛地用於數位相機或是網際網路，正式名稱為「ISO/IEC 10918-1 Digital compression and coding of continuous-tone still image[53]」，制訂的社群名稱 Joint Photographic Experts Group 為其簡稱。

先前已經學習了好幾章的影像編碼之基礎，關於多媒體影像編碼的實際應用情形，在本章簡單地介紹 JPEG 格式。

前 言

更正確地說來，JPEG爲對於彩色多階影像之編碼方法。在JPEG所規定的各種編碼方式如圖 9.1 所示。吾人通常將其中稱爲**基線系統**的破壞性方式，稱作一般的「JPEG」。以JPEG爲標準之系統，至少需遵照

基線系統進行位元串之編碼／解碼。延伸系統爲基線系統之各性能加以擴張。另外，也訂定有稱爲空間(spatial)方式之非破壞性編碼方式，相對於DCT方式，稱爲**獨立函數**。

- JPEG
  - DCT 方式 (破壞性)
    - 基線系統
    - 延伸系統
  - 空間方式 (非破壞性) — 獨立函數 不明白

圖 9.1　JPEG之各種編碼方式

本章以基線方式爲中心，簡介JPEG編碼演算法與位元串之組成方法。

基本編碼演算法

JPEG 編碼器之基本組成

JPEG 爲了可對所有的顏色空間的影像進行編碼，對表現顏色的顏色空間完全不加以規定。在編碼端對輸入的「數個顏色分量」僅單純地進行編碼，完全不過問是哪個顏色的分量。因此，目標影像之YC_bC_r等顏色空間之轉換操作，原本即屬JPEG編碼器**以外**之處理範疇[106]。

註　106.進一步地說，包含在位元串中作為控制資訊的僅為組成**顏色分量數**，且對於顏色空間之記載方法並沒有規定。因此，在編碼／解碼間，其顏色空間是共有的，在後述之JFIF格式中，需先將顏色空間規定為YC_bC_r。

圖9.2爲處理各資料分量之 JPEG 編碼器之基本組成。輸入的各資料分量，每個像素減去128之偏移值(基線的情況)之後，以8×8爲單位進行 DCT。再送至量子化器。在量子化器中，利用量子化表$Q(u,v)$，將DCT係數$F(u,v)$以下列方式量子化：

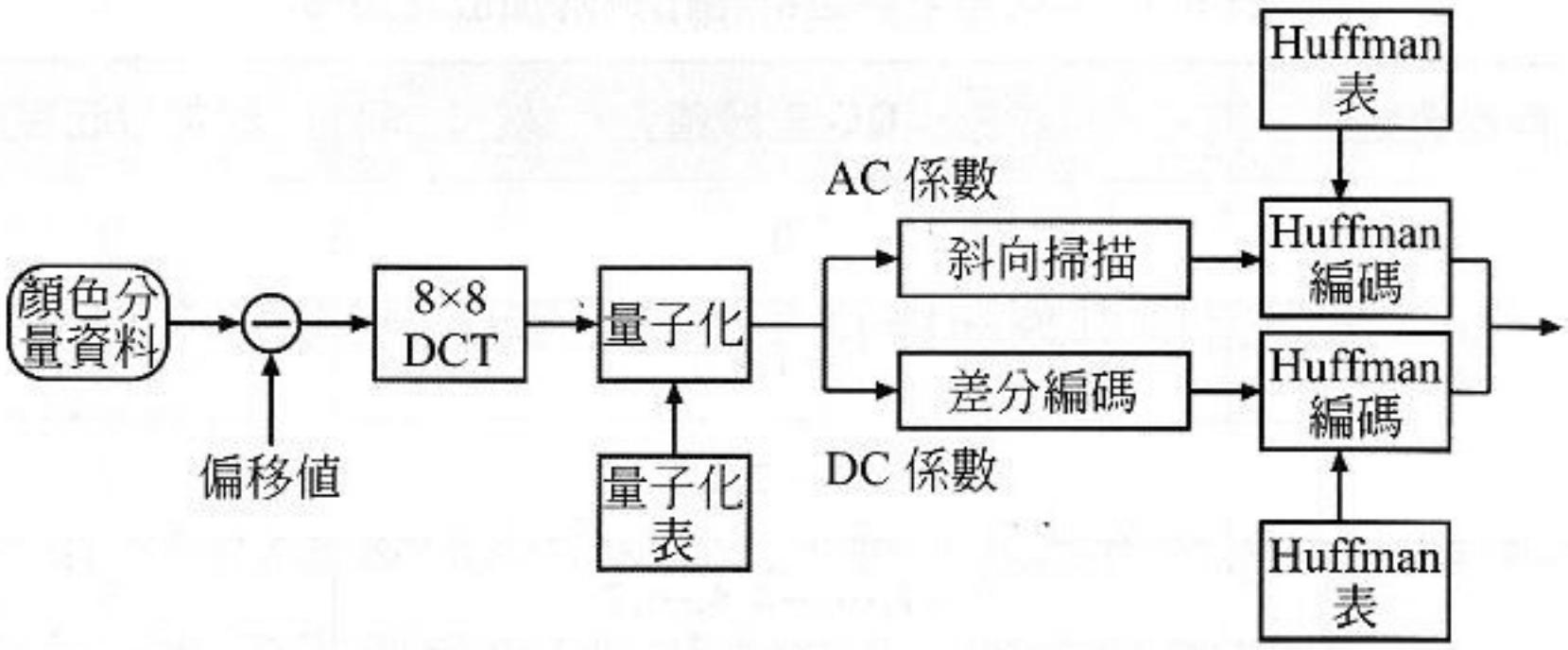


圖 9.2　JPEG 編碼器之基本組成

$$\lfloor F(u,v)/Q(u,v)+0.5 \rfloor \tag{9.1}$$

量子化之後的DCT係數，如 6-13 頁所述，分成AC係數與DC係數進行熵編碼。DC 係數爲了除區塊間相關程度，取之前編碼後的**同一顏色分量**之DC係數的**差分**，對該DC差分值進行可變長度編碼[107]。AC係數，由圖 6.10 之斜向掃描加以一維化，表示爲**延伸長度**與**切斷延伸之非零係數**的組合之後，再進行熵編碼。在基線系統中，以Huffman編碼作爲熵編碼方式。

註　107.在一開始先歸零，到了畫面的右端就進到下一行左端，連續進行。

DCT 係數之 Huffman 編碼

對DC差分值直接進行Huffman編碼會使得編碼長度顯著地增長，爲了避免此現象，使用在 3-16 頁中所述之**群組化**方法。

具體來說，DC差分值。以表 9.1 所示加以群組化之後，Huffman編碼其群組編號(0～11)。但是，編號i的群組中共包含2^i個DC差分值，需有用來識別的資訊(稱爲附加位元)。因此，DC 差分值 X 之實際編碼，可依下列方式進行。

表 9.1　DC 差分值之群組化與附加位元長度

群組編號	DC 差分值	附加位元長度
0	0	0
1	$-1,1$	1
2	$-3,-2,2,3$	2
3	$-7,\cdots,-4,4,\cdots,7$	3
4	$-15,\cdots,-8,8,\cdots,15$	4
5	$-31,\cdots,-16,16,\cdots,31$	5
6	$-63,\cdots,-32,32,\cdots,63$	6
⋮	⋮	⋮
11	$-2047,\cdots,-1024,1024,\cdots,2047$	11

① 由表 9.1 求得X所屬之群組編號，再進行 Huffman 編碼。
② 設第i群中最左端的值爲第零個，再求X爲第幾個，以固定i位數的二進位將此表示爲附加位元。
③ 在群組編號之Huffman編碼後頭加上附加位元，即爲X之編碼。

例 1

編碼 DC 差分值 17。由表 9.1，群組編號爲 5，附加位元爲$(10001)_2$，前者之Huffman編碼爲$(110)_2$，因此可得$(11010001)_2$。

另外，對於 AC 係數，以式(6.16)將其表示爲延伸長度 L 與將中斷延伸的非零係數F之組合$(L:F)$，在有必要時[108]，再使用EOB，以下列方式進行編碼。

註　108.在 JPEG 中，第 63 個 DCT 係數非零的情況中，不另加 EOB。

① 延伸長度以二進位表示爲$(RRRR)_2$(L超過 15 的情況如後所述)。

② 藉表 9.2 求得F所屬之群組，其群組編號以四位元之二進位表示爲$(SSSS)_2$。

③ 將兩者加以連接$(RRRRSSSS)_2$，將其視爲符號進行 Huffman 編碼後，爲了識別爲第F群組內的值，與處理 DC 係數時相同，附加指定的位元長度。

表 9.2　AC 係數 F 之群組化與附加位元長度

群組編號 (SSSS)	F值	附加位元長度
1	$-1,1$	1
2	$-3,-2,2,3$	2
3	$-7,\cdots,-4,4,\cdots,7$	3
4	$-15,\cdots,-8,8,\cdots,15$	4
5	$-31,\cdots,-16,16,\cdots,31$	5
6	$-63,\cdots,32,32,\cdots,63$	6
⋮	⋮	⋮
10	$-1023,\cdots,-512,512,\cdots,1023$	10

另外，當延伸長度超過 15 時，將延伸長度 16 重覆以ZRL碼取代，再回歸到延伸長度爲 15 以下的處理情形。

例 2

對 $(L:F)=(3:6)$ 進行編碼。$L=3$，$(RRRR)_2=(0011)_2$。由表 9.2 可知，$F=6$ 所屬之群組爲 3，$(SSSS)_2=(0011)_2$。另外，$F=6$ 之附加位元爲 $(110)_2$。因此，$(RRRRSSSS)_2$ 之 Huffman編碼爲 $(111111110101)_2$，最後的編碼爲將兩者合併之$(111111110101110)_2$。

在基線系統中，可以如表 9.2 所示將F值分爲 10 個群組，與其各別對應之延伸長度 0～15 可以得到 16 種數值。如此一來，將其當作Huffman編碼的對象，這些組合加上EOB與ZRL合計共有 162 個符號[109]。另外，DC 係數與 AC 係數之 Huffman 編碼之最大編碼長度限制爲 16。

註 109.EOB 之分配為 $(RRRRSSSS)_2=(00000000)_2$，同樣地，ZRL 之分配為 $(11110000)_2$。

量子化表與 Huffman 表

在JPEG中，並**沒有**定義**預設**的量子化表與Huffman表，基本上在位元串中需包含必須的對照表，否則必須由應用程式規定並由編碼／解碼端共用。另外，爲了改善編碼效率而使用依顏色分量而不同之對照表，在一個位元串中包含複數對照表的情形是常見的。

量子化表記載於位元列中，包含下列兩項：

- 對照表之識別編號(爲了標示各顏色分量)。
- 依斜向掃描順序排列的六十四個對照表值(在基線中各值以固定 8 位元的整數加以呈現)。

關於 Huffman 表，並沒有關於 Huffman 樹本身之記載，而是記載使其一致之記載。具體來說，給定下列項目：

- 對照表之識別編號。
- 編碼長度i位元($i=1,2,\cdots,16$)之 Huffman 編碼的總數(以固定長度 8 位元表示)。
- 在二進位編碼並排之符號值(在 DC 係數之情況爲$(SSSS)_2$，在 AC 係數之情況爲$(RRRRSSSS)_2$)。

此時，規定在解碼端出現之符號以「0」優先進行分配，可藉上述資訊製作一致的 Huffman 樹。

在JPEG中利用YC_bC_r時，式(9.2)與(9.3)係爲建議值而廣泛使用(前者用在Y，後者用在C_bC_r)。也就是如第 6-9 頁所述，理想狀況爲利用式(5.40)對每個影像進行最佳量子化表，但計算量會增大，而且各別製作，效果會變差，使得這種做法幾乎不具優勢。因此，利用式(9.2)、(9.3)之對照表，在控制資料量或是畫質之時，多用其整數倍(整數倍之後的對照表值，係在位元串中依照前述量子化表之記載方式給定)。這種情況下，Huffman 對照表也是一樣，廣泛地使用依照 JPEG 建議值所給定之對照表。

$$Q_Y(u,v)=\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \tag{9.2}$$

$$Q_C(u,v)=\begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix} \tag{9.3}$$

JPEG 基線方式之位元串組成

JPEG位元串，係依照圖 9.2 輸出之二進位編碼依照給定格式加以排列，基本上由三種資訊組成：

- 影像尺寸與顏色分量之規格(標頭資訊)。
- 量子化表或 Huffman 表。
- 依水平掃描順序排列之編碼資料。

JPEG 位元串之階層構造

JPEG位元串，如圖 9.3 所示，由影像、畫格、掃描三階的階層所組成，分別以下列資料單位進行考慮。

- 影像：一幅影像的資料。
- 畫格：在分階編碼[55]中的一組階層資料(在基線方式中，畫格素爲 1)[110]。
- 掃描：在影像內所有的區塊之 DCT 係數之集合(後述)。

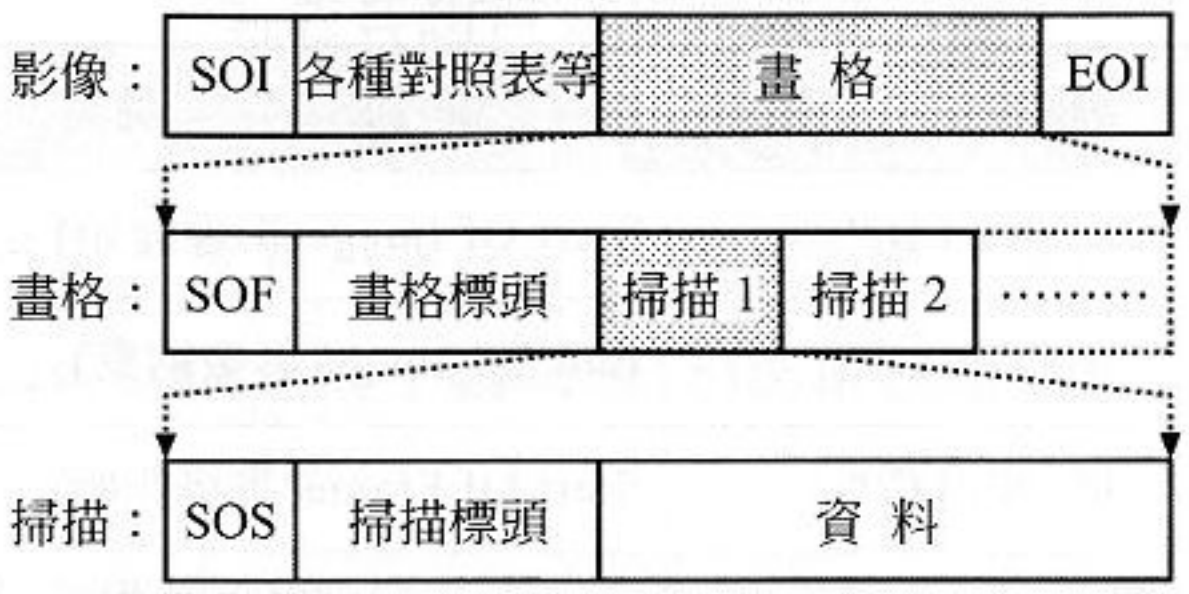


圖 9.3　JPEG 位元串之階層構造

(譯按：分階編碼(Hierarchical encode)亦有譯作階層編碼，爲避免與階層(layer：在繪圖軟體中亦有譯作圖層者)混淆，在本書中各譯作分階編碼、階層，請讀者諒察)

註　110.在本書中，限於篇幅，對於分階編碼不予討論。

在圖 9.3 中，SOI、EOI 等爲用來表示各種資料之起始點的二位元組的編碼，稱爲**標記碼**(marker code)。代表性的標記碼如表 9.3 所示。每種標記碼之起始位元組都是十六進位的 X'FF'[111]，插入位元串時需遵守下列規定。

註　111.在本書中，對於十六進位數值以 X'十六進位值'加以表現。

① 標記碼必須插入於位元組之間(8 位元的整數倍)的位置(此稱爲位元組配置)。於可變長度編碼之後安置標記碼時，配合需要插入適當數目之補碼'1'，以符合位元組配置。

② 在可變長度編碼內，當於位元組之間產生編碼X'FF'$(11111111)_2$的情形中，在其之後需配置無效碼X'00'[112]。在解碼端，當X'00'接著 X'FF'而來時，將 X'00'排除後繼續進行解碼。

註　112.係為了不讓後續的位元將其誤判為標記碼而加以處理。

表 9.3 代表性的標記碼

標記碼	編碼(十六進位)	說明
SOI	X'FFD8'	Start Of Image(影像開始)
EOI	X'FFD9'	End Of Image(影像結束)
SOF	X'FFC0'	Start Of Frame(畫格開始)
SOS	X'FFDA'	Start Of Scan(掃描開始)
DHT	X'FFC4'	Define Huffman Table(定義 Huffman 表)
DQT	X'FFDB'	Define Quantization Table(定義量子化表)

掃描之組成

在 JPEG 的基線方式中，所謂**掃描**相當於「整體影像之各區塊之DCT係數的組合資料」。但是，通常存在複數個顏色分量，此外，必須對應像是4：2：2或是4：2：0，不同分量之間像素數目有所不同的格式。

JPEG中，**分量編號**i之顏色分量的像素數$x_i \times y_i$，以縱向橫向之取樣密度比分別以H_i、V_i加以表示：

$$x_i = \lceil x \times H_i / H_{\max} \rceil \quad 及 \quad y_i = \lceil y \times V_i / V_{\max} \rceil \tag{9.4}$$

其中，$x \times y$爲原影像之像素數目，$H_{\max} = \max_i H_i$，$V_{\max} = \max_i V_i$。舉例來說，如圖 9.4 所示，考慮尺寸爲 48×32 之YC_bC_r格式(Y、C_b、C_r之分量編號分別爲$i = 1$、2、3)。此時各分量之取樣密度比爲：

$$H_1 = 2，V_1 = 2，H_2 = 1，V_2 = 1，H_3 = 1，V_3 = 1 \tag{9.5}$$

儲存在位元串中之資料，爲x、y與各H_i、V_i，藉此保持各顏色分量之像素爲一致。

而在 JPEG 中掃描對於各顏色分量之處理方法，有**非交錯式**與**交錯式**(interleave)兩種方式可供選擇。前者如圖 9.5(a)，將Y、C_b、C_r之DCT係數各有一組掃描。後者為常用的方式，如同一圖(b)所示，以 MCU (Minimum Coded Unit：最小編碼單位)為單位組成具有三種顏色分量之「網格」，只有一組掃描的方式。在此，所謂的「MCU」，係指「解碼時為了決定像素值所需的最小DCT區塊之集合」，舉例來說，在 4：2：0 格式的情形中，如圖 6.7 所示需要Y四個、C_b、C_r各一個共計六個 DCT 區塊[113]。

註　113.一般來說，在各分量由$H_i \times V_i$個 DCT 區塊所構成，全部共有$\Sigma_i H_i V_i$個區塊。

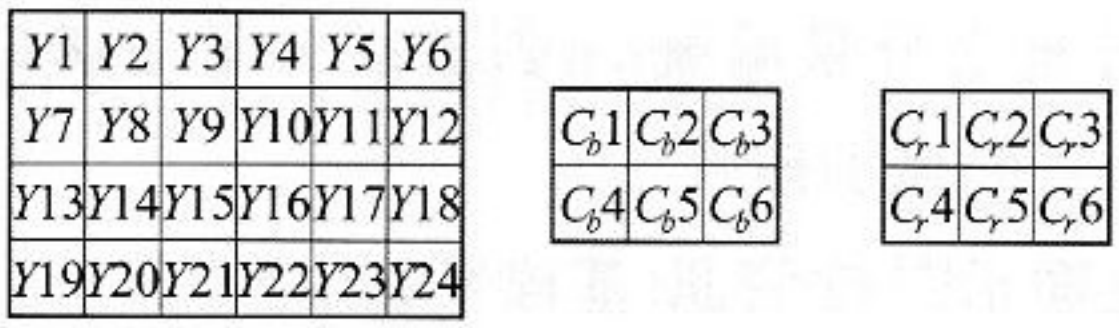


圖 9.4　尺寸 48×32，YC_bC_r 4：2：0 格式影像之例

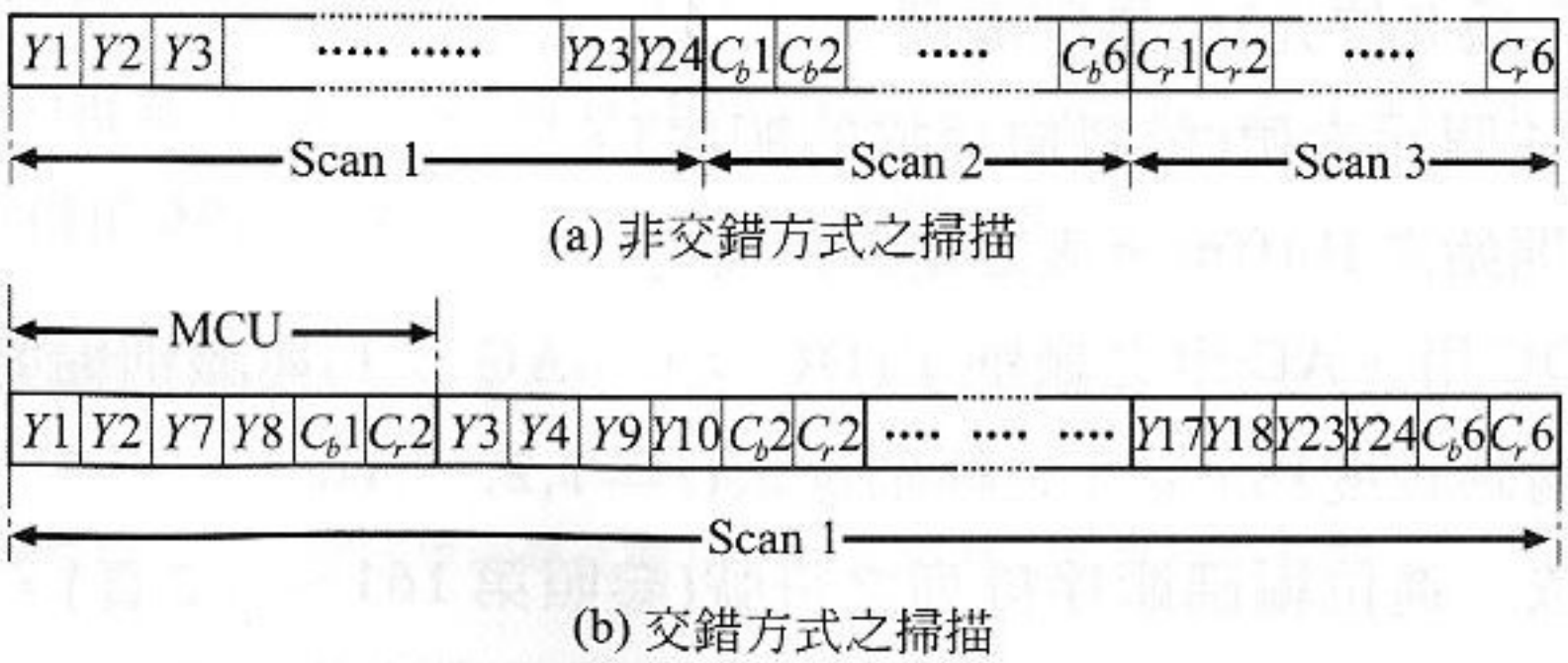


圖 9.5　非交錯方式與交錯方式

實際的資料格式

在表 9.3 中之 SOF 與 SOS 之後，係配置圖 9.3 所示之**畫格標頭**、**掃描標頭**分別爲：

畫格標頭

- 像素之位元精度(基線的情形中，爲 8 位元)。
- 影像之掃描線數目與每條掃描線像素數目(x、y)。
- 顏色分量數，以及各顏色分量之顏色分量編號i、H_i、V_i，以及該分量所使用量子化表之識別編號。

掃描標頭

- 在掃描中所含之顏色分量數目。
- 各顏色分量之分量編號i，對於該分量之 DC 與 AC 所使用 Huffman 表的識別編號。
- 關於漸進顯示[53]之資訊(基線系統中無)。

另外，以DQT開始之量子化表的定義部份中，需有：

- 各元素之精度(位元長度)(0：8 位元、1：16 位元，基線的情形爲 8 位元)與識別編號。
- 64 個元素值(依斜向掃描的順序)。

以DHT開始之Huffman表之定義部需有：

- DC 用、AC 用之識別子(DC = 0、AC = 1)與識別編號。
- 編碼長度爲 i 位元之編碼總數(i = 1,2,⋯,16)。
- 依二進位編碼順序排列之符號(參照第 161～162 頁)。

舉例來說，以YC_bC_r爲 4：2：0 格式、交錯方式之位元串爲例，可以依下列的標記／資料順序寫入[114]。

- SOI。
- 亮度Y與色差$C_b C_r$用的兩組量子化表。
- SOF 與畫格標頭。
- Y的 DC 與 AC；$C_b C_r$的 DC 與 AC，共計四個 Huffman 表。
- SOS 與掃描標頭。
- 掃描資料(四個Y與各一個$C_b C_r$之DCT區塊組成MCU，將此以水平掃描順序排列之資料)。
- EOI。

註　114.在實際的應用中，考慮 JPEG 應用程式之互換性，常使用稱為 JPEG File Interchange Format(JFIF)之格式，在該情形中，在 SOI 之後即插入標記碼 X'FFE0'(參照圖 9.6)。

JPEG 位元串之具體例

爲了觀察 JPEG 位元串之實例，將圖 6.5(a)之 8×8 區塊想成「單色影像」，再對其進行 JPEG 編碼。因爲目標影像爲單色的，因此顏色分量僅有亮度Y，量子化表、Huffman表都使用Y之建議值[115]。目標影像的亮度值$f(i,j)$與量子化後之DCT係數$F'(u,v)$，可由式(6.10)、式(6.12)及式(6.15)推知。Y用的建議 Huffman 表之一部份如表 9.4 所示圖 9.6 爲由以上所得的 JPEG 位元列以 16 進位展開的結果[116]。

註　115.在以下的例中，用的量子化表與 Huffman 表不由位元串提供。

116.JPEG 位元串在 IBMPC 上展開後，會得到如同「D8EF E0FF」之位元正向或是反向排列。要注意的是此與 INTEL 處理器之位元排列有關。

首先，在標記碼 SOI(X'FFD8')之後，如同註解[114]所述安置 JFIF 格式之標記碼X'FFE0'與標頭資訊 117。接著，定義量子化表。圖 9.6 中以

「64 個量子化表數值」所表示的 64 個數值(各一個位元組)以十進位表示後，與式(9.2)之量子化表(依照斜向掃描順序排列)是一致的。

註　117.此處不涉及 JFIF 格式之內容。請情請參閱文獻[54]。

表 9.4　*Y*用的 DC 與 ACHuffman 表

DC 的 Huffman 表

符號(群組編號)	二進位編碼
0	11
1	010
2	011
3	100
4	101
5	110
6	1110
7	11110
8	111110
9	1111110
10	11111110
11	111111110

AC 的 Huffman 表

符號		二進位編碼
RRRR	SSSS	
0	1	00
0	2	01
0	3	100
…	…	…
0	10	1111111110000011
1	1	1100
…	…	…
1	10	1111111110001000
…	…	…
3	3	111111110101
…	…	…
6	1	11110110
…	…	…
15	10	1111111111111110
EOB		1010
ZRL		11111111001

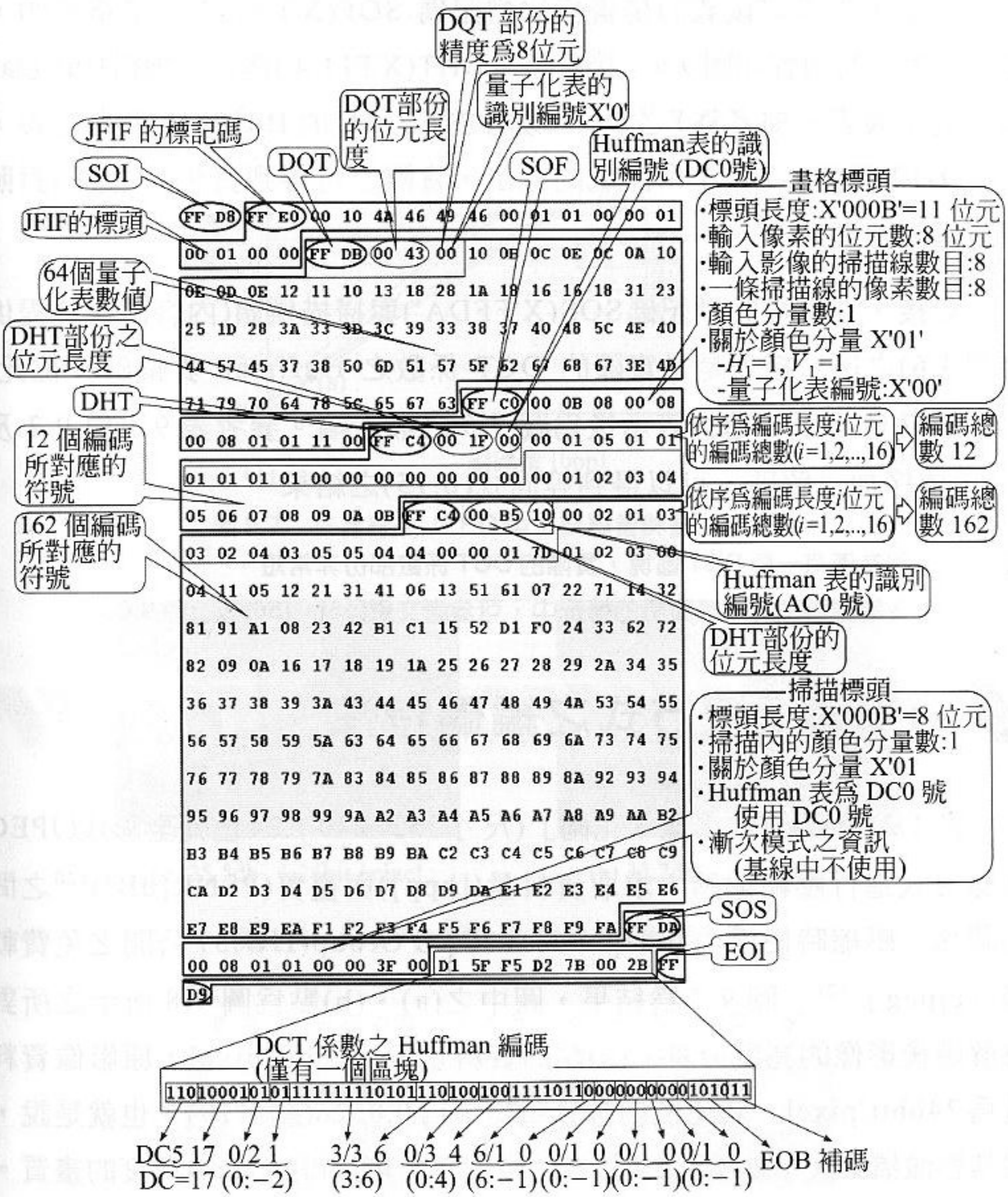


圖 9.6　基線系統位元串之實例

在量子化表之後放的是畫格的標記碼 SOF(X'FFC0')與畫格標頭，標頭所表示的內容如圖 9.6。接著是以DHT(X'FFC4')爲首之兩組Huffman編碼的定義表。前者爲Y之 DC，後者爲Y之 AC 的 Huffman 表之定義，參考 9-12 頁與 9-7 頁中所記載的方法，實際上可得到表 9.4 左邊的對照表可確認並理解之。

最後，在掃描的標記碼SOS(X'FFDA')與掃描標頭(內容同樣地提供於圖 9.6)之後，接著是是實際的 DCT 係數之 Huffman 表編碼(7 位元)[118]。該 7 位元以二進位表示後爲圖 9.6 之最下端。參考表 9.1 與 9.2 及 9.4，依序加以解碼，可以得到等同式(6.15)之結果[119]。

註 118.因為僅有一個 DCT 區塊，實際的 DCT 係數部份非常短。

119.在需要標頭之細節資訊的情況中，可參考文獻[55]、[56]分析圖 9.6。

JPEG 基線方式之編碼特性

圖 1.21(a)扉頁的影像「花園」(尺寸 352×240，24 位元全彩)以JPEG基線方式進行壓縮編碼。求取資料量([bpp])與畫質(PSNR[dB])[120]之間的關係。壓縮時使用由 Independent JPEG Group(IJG)57 公開之免費軟體「cjpeg」[121]。圖 9.7 爲結果，圖中之(a)、(b)點爲圖 9.8 所示之所對應解碼後影像的亮度分量。(a)點的資料量爲 1.26bit/pixel，原影像資料量爲 246bit/pixel，其比例相當於壓縮率爲 0.0526(約 1/19)。也就是說，當其影像檔案大小壓縮至 5%程度的情況下，可以得到 26dB 程度的畫質。

註 120.原影像與解碼後影像之Y分量間的 PSNR。

121.可於本章最後提到的網站下載該 JPEG 壓縮軟體。

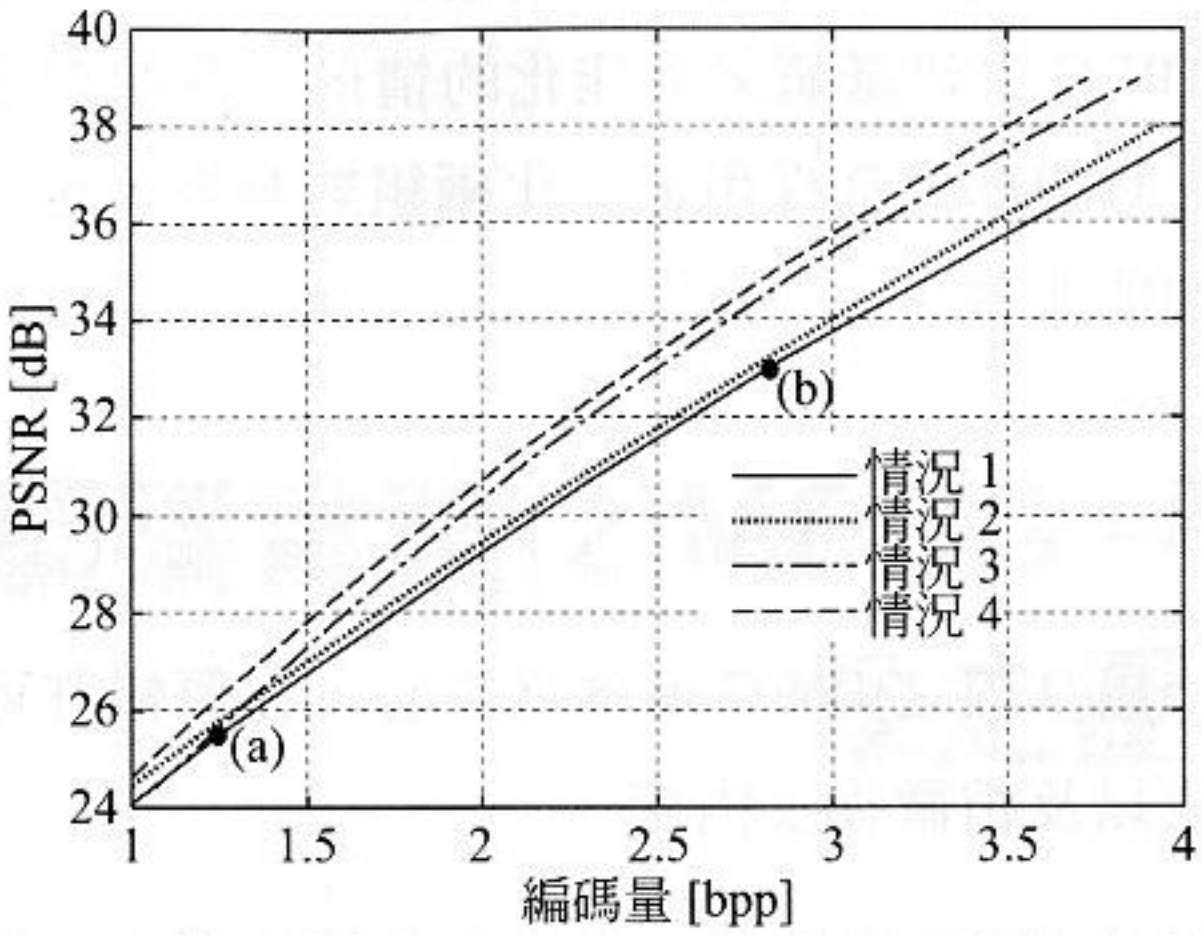


圖 9.7　在基線之 JPEG 中，資料量與畫質的關係

(a) 25.5db , 1.26bit / pixel

(b) 33.2dB , 2.85 bit / pixel

圖 9.8　圖 9.7(a)與(b)點所對應之解碼後影像

表 9.5　不同的量子化表與 Huffman 表之效率比較

		Huffman 表	
		建議值	最佳化
量子化表	建議值	情況 1	情況 2
	最佳化	情況 3	情況 4

另外，在圖 9.7 中的四條曲線，如表 9.5 所示，各爲量子化表／Huffman 表爲 JPEG 之建議值／最佳化的情形，共計對應到四種情形。在同一圖中，可知道即使都採用最佳化兩組對照表，也不會較採用建議值的情形有特別的改善。

結　語

在本章中，以基線之 JPEG 編碼爲中心，在篇幅許可的情形下說明了位元串之組成以及稍微提及內容。

不同讀者對於 JPEG 的標準、規格有各種的看法。舉例來說，也有人是只要了解到基本的編碼演算法就夠了，也有人是以實際設計編碼／解碼器或是進行程式編寫爲最終的目的。對前者來說，理解到第六章爲止與本章的內容就已十分夠用。另外，目的爲後者的讀者則需要位元串之組成等更詳細的資訊。這些在 JPEG 規格本身中有所提及或是可參考文獻[53]或是文獻[29]、[55]、[56]等。

因，在網際網路上也有各種有用的資訊。在本章結束之前，要介紹一個有用的網站。就是關於 JPEG 之 FAQ(常見問題)的網站：

http：//www.faqs.org/faqs/jpeg-faq/

在這裏提供了許多關於 JPEG 的各種疑問質問的回答，或是在對演算法進行程式編寫時的提示。特別是在回答「在哪可讓我找到 JPEG 軟體」的問題時，介紹了一個 JPEG 之免費 C 語言程式碼：

ftp：//ftp.uu.net/graphics/jpeg/jpegsrc.v6b.tar.gz

在「想試試 JPEG 之壓縮編碼」或是「想見識實際的原始碼」時，上述的網站可以提供相當大的幫助，建議一定要去點看看。

10章

多媒體編碼實務(2)MPEG－動畫影像之壓縮編碼

MPEG 為動畫影像編碼之國際標準，與 JPEG 相同，由制訂的委員會 Moving Picture Experts Group 之名稱為其簡稱。在本章中，在數個 MPEG 標準中，說明了最基本的MPEG-1之編碼方式後，再論及其發展形之MPEG-2。

MPEG的歷史－常識程度的介紹

對於動畫影像的壓縮編碼方式加以國際標準化或是規格化的嘗試，最初是在通信領域中以視訊會議爲目標，由國際電信電話諮詢委員會 CCITT(現在的 ITU-T)建議採行 H.261 規格。其係併用了第七章所述的動態補償與 DCT 之整合編碼方式。在此前後，以將一小時左右的的影像以VHS程度畫質儲存於CD-ROM當中爲目的，1998 年國際標準化機構ISO與國際電氣標準化會議IEC之共同機關內設置了由專家組成的研討會Moving Picture Experts Group，開始進行演算法的討論。在 1993

年提出了標準化後的方法「ISO/IEC 11172-2 "Coding of moving pictures and associated audio for digital strange media at up to about 1.5 Mbit/s"」58。這就是簡稱MPEG-1而爲人熟知的規格[122]。MPEG-1與H.261一樣，是基於MC與DCT之整合編碼方式，其基本原理與h.261的演算法高度相關。

註 122.可由正式名稱知道，MPEG1 不單是以動畫影像為壓縮的對象，實際上還規定了聲音訊號之編碼方法或系統組成。

較MPEG-1進一步在高畫質領域中的的應用，舉例來說現行的電視訊號以專業畫質進行編碼等應用，在1990年開始了 MPEG-2方式之標準化。途中，確定了作爲HDTV訊號之編碼之MPEG-3也可以MPEG-2加以實施，最後以包含在HDTV的應用之形式，在1996年提出了標準化方式「ISO/IEC 13818 "Generic coding of moving pictures and associated audio information"」，MPEG-2之處理對象，舉例來說，現行的電視播放爲4～15Mbps程度之編碼的應用，或是HDTV訊號之數十Mbps編碼的應用。

另外，對於行動終端間之動畫傳送等應用，則需要較MPEG-1預定之1.5Mbps更低資料率／高壓縮比之編碼方式，關於此之檢討，在1993年開始了MPEG-4的討論活動。其後，其對象改爲網際網路或是個人電腦上的多媒體資料，最後，同時具有高性能／高壓縮性之編碼方式在1999年以 MPEG-4加以標準化。該方式的特徵爲，將影像以物件爲單位分割後進行編碼，採用**物件導向編碼**爲其特點。

? 型狀?

另外，方向與以上標準化之著眼點不同，以易於進行多媒體資訊之檢索與取得爲目的之儲存方式的MPEG-7，更進一步地配合流通服務或是智慧財產權的保護之MPEG-21也正在發展中。

MPEG-1 簡述

關於 H.261 －準備知識

H.261 方式爲如前述之併用MC與DCT之整合編碼方法，其演算法幾乎與第七章之圖 7.22 相同。簡而言之可想成是「對動畫影像之畫格以 ME ／ MC 作成預測影像，該預測誤差以類似 JPEG 的演算法加以壓縮」。因爲H.261 當初是預定用在通訊領域的規格，因此盡可能地要求減少編碼器之輸入輸出間之延遲。

關於 MPEG-1 －導入雙向動態補償預測

另外，MPEG-1 之目的爲在 CD 等儲存媒體上記錄影像，因此不需要較低的延遲，因此可在貯存數幅輸入畫格後再進行編碼。在 MPEG 中，利用此性質，導入在時間的過去與未來兩個方向上的**雙向動態預測**，以達到提高編碼效率之目的。

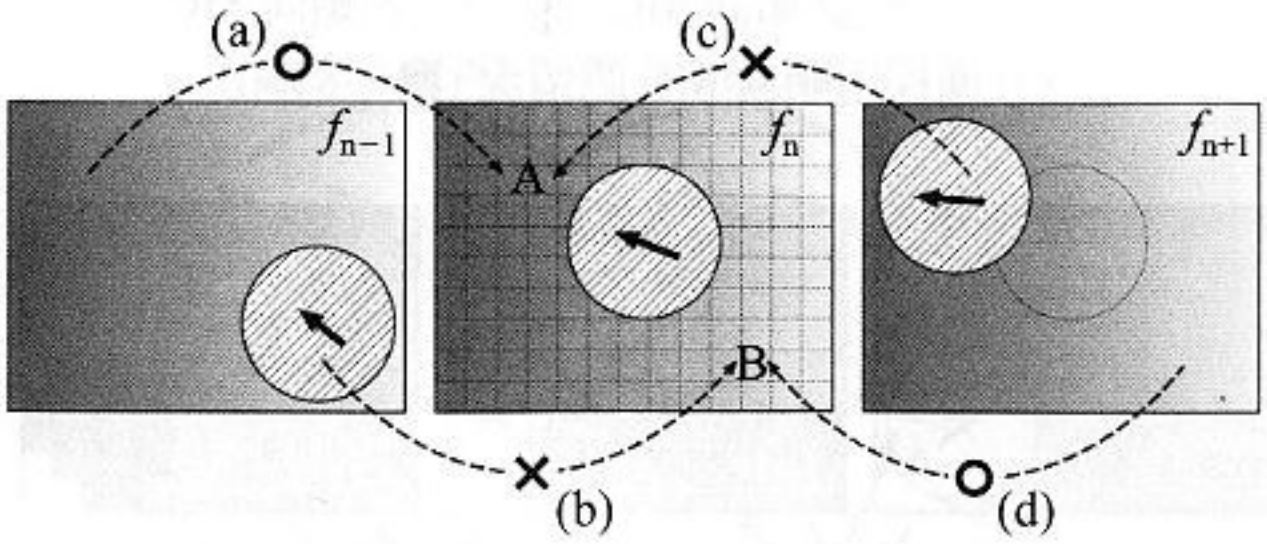


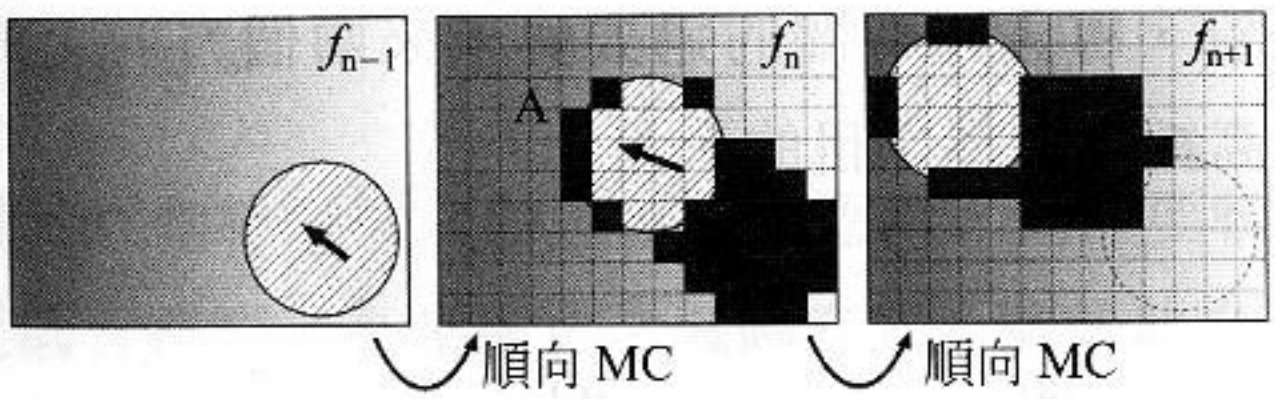
圖 10.1　連續的三個畫格與動態補償

現在，在連續的三幅畫格 f_{n-1} 、f_n 與 f_{n+1} 中，以 f_n 爲對象進行 ME／MC。在第七章說明的 ME／MC 以之前的 f_{n-1} 爲參考畫格搜尋對應的區塊。此稱爲**順向** ME／MC。當目標畫格上的 A 點周圍的區塊在參考畫格上存在對應的區塊，可進行 ME，但在 B 點附近相當於**未涵蓋區域**，因此無法進行 ME。

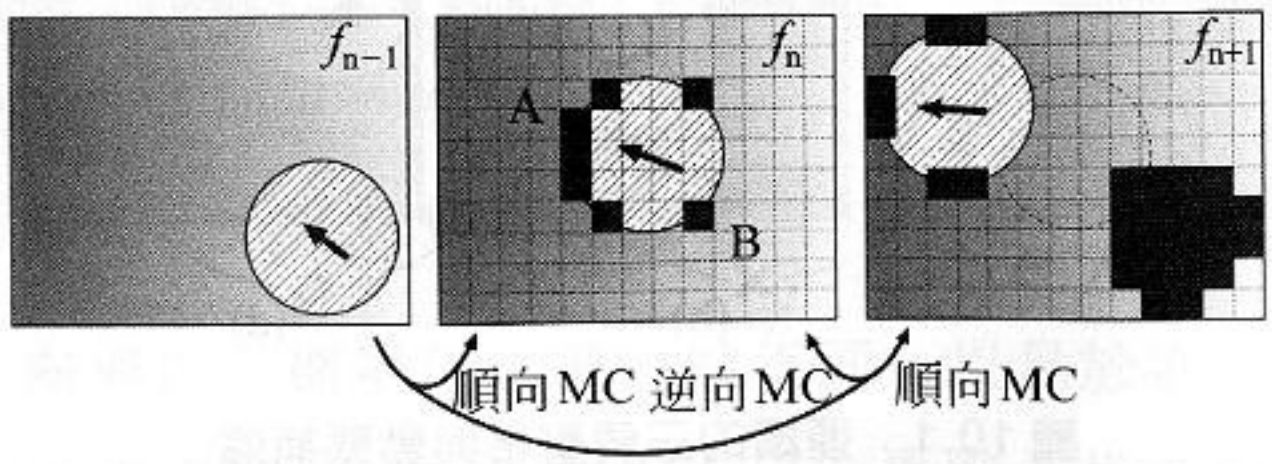
因此，在有些方法中，假設：

> f_n 同時對**過去**的畫格 f_{n-1} 與**未來**的畫格 進行編碼之後，在解碼端可將該兩個畫格當作參考畫格。

此時，對於目標畫格 f_n 以 f_{n+1} 爲參考畫格進行**逆向ME／MC**，這次在A點附近的ME是失敗的，但是在B點附近的ME是成功的。如此一來，對目標畫格同時進行順向與逆向兩種模式的操作，可以減少在 f_n 上 ME 失敗的畫格數。



(a) 僅採用順向 MC 的情況 (■=58 個)



(b) 採用雙向 MC 的情況 (■=33 個)

圖 10.2　順向與雙向 MC(■為進行 ME 時失敗的區塊)

圖 10.2 將 f_n 與 f_{n+1} 上失敗的區塊以■表示[123]，如同一圖(a)所示，對 f_n 、f_{n+1} 進行順向ME，在兩畫格上的失敗區塊數達到 58 個。另外，依下列順序—**雙向 ME／MC**—進行。

① 將 f_{n-1} 視爲參考格對 f_{n+1} 進行順向 ME／MC。

② 接著將這兩個畫格視爲參考畫格，對 f_n 進行順向／逆向任一方向之 ME。

失敗畫格數減少至 33 個(同一圖(b))。

註　123.在區塊內有 50%的像素找不到對應點視為 ME 失敗。

對 ME／MC 失敗的區塊進行內部編碼，與畫格間編碼相較效率要來得差，由雙向 MC 可以減少內部編碼的區塊數，大幅改善編碼效率[124]。另外，進一步地擴展此觀念，可藉「自 f_m 至 f_{m+M} 進行順向 ME，其間的$(M-1)$幅畫格進行雙向 MC」可進一步地改善效率(圖 10.3)。但是，當 M 過大時，因爲畫格間的時間差會使得 ME／MC 所需的計算量增大，或是使得進行內部編碼的區塊數增加，因此 $M=3$ 就相當實用。

註　124.在相同的資料率下，雙向MC與僅使用順向的情形相較，在PSNR可得到 1～2dB 的改善。

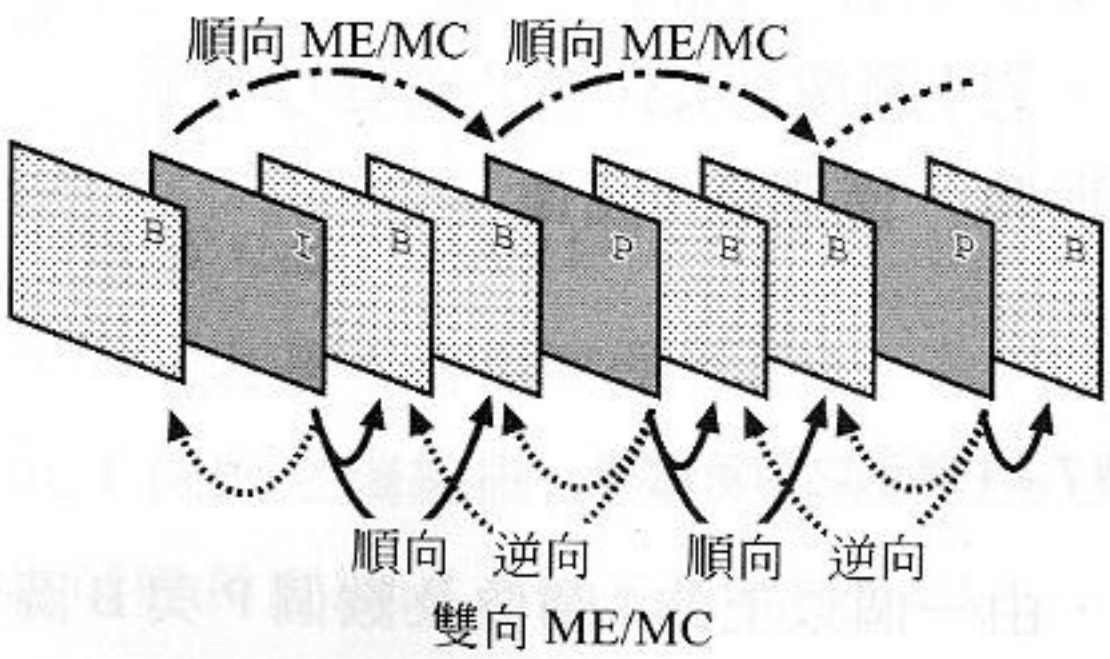


圖 10.3　$M=3$ 的情況下之雙向 ME／MC

在 MPEG 中，配合 MC 的模式，各畫格稱爲[125]：

- **I圖像**(Intra-coded picture)：整體進行內部(intra)編碼的畫格。
- **P 圖像**(Predictive-coded picture)：僅進行順向 ME ／ MC 之畫格。
- **B圖像**(Bidirectionally predictive-coded picture)進行雙向ME ／MC 之畫格。

以$M=3$進行雙向 MC，各圖像之配置如圖 10.3 所示。

註 125.實際上除了這些，還存在著稱為 D 圖像的畫格，但是除了特殊的用途之外是不使用的，在本書中加以省略。

資料之階層構造

利用 CD 等媒體記錄動畫之系統中，必須要有**隨機選取**或是**快轉等特殊播放功能**。因此，在使用基於MC之預測編碼的情形中，對某個畫格進行解碼時，需要有參考畫格，這種連鎖現象到其刷新畫格爲止都是持續的，對於一部份畫格之播放顯示的特殊播放，會造成需要對全部的畫格進行解碼而使得效率變差。因此在MPEG中，需以適當的間格插入可以單獨進行解碼之I圖像，依照：

- 在快轉時，對I圖像或是I與P圖進行解碼。
- 在隨機選取時，僅選取I圖像。

進行特殊播放[126]。

註 126.藉此進行圖 7.21 所示之刷新處理。

在 MPEG 中，由一個以上的 I 圖像及數個 P 與 B 圖像組成的圖像集合稱爲圖像群組(**GOP**)，可將動畫影像視爲GOP的集合。圖 10.4 爲GOP

之一例。GOP 的組成可由使用者自由選擇，雖然其組成可依每個 GOP 而有所不同，但是一般常規定在GOP內的總圖像數N與I/P之畫格間隔M這兩個參數。圖 10.4 爲$N=12$，$M=3$ 之例。

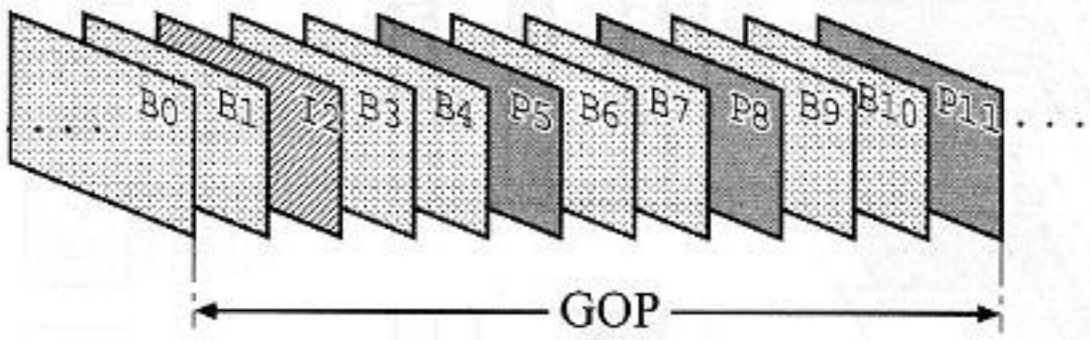


圖 10.4　GOP 之例

實際MPEG的位元串之組成，包含GOP全部共有六個**階層(LAYER)**(圖 10.5)。各層基本上以其下一層的資料集合爲定義，但在序列層至切片層的最前端具有控制資訊集合之**標頭**。各標頭遵守下列規定：

- 開頭爲表 10.1 所示的 4 位元組之起始碼。
- 起始碼必需位在位元組的間隔(8 位元的整數倍)[127]。

因此，在解碼時產生錯誤的情況中，可以當做迴歸標記。

註　127.這個如同在第九章說明過的**位元組配置**。

從圖 10.5 的最下層往最上層看，結果可知MPEG之位元串由「DCT係數之編碼」、「移動向量(MV)」以及「各種控制資訊」所組成，因此要了解MPEG的細節，必須理解：

① MC + DCT 方式之基本編碼演算法。

② MV 或是 DCT 係數之熵編碼(可變長度編碼)方法。

③ 控制資訊或可變長度編碼之位元串構成方法。

在下一節中，對於在 MPEG-1 中之上述三點進行說明。

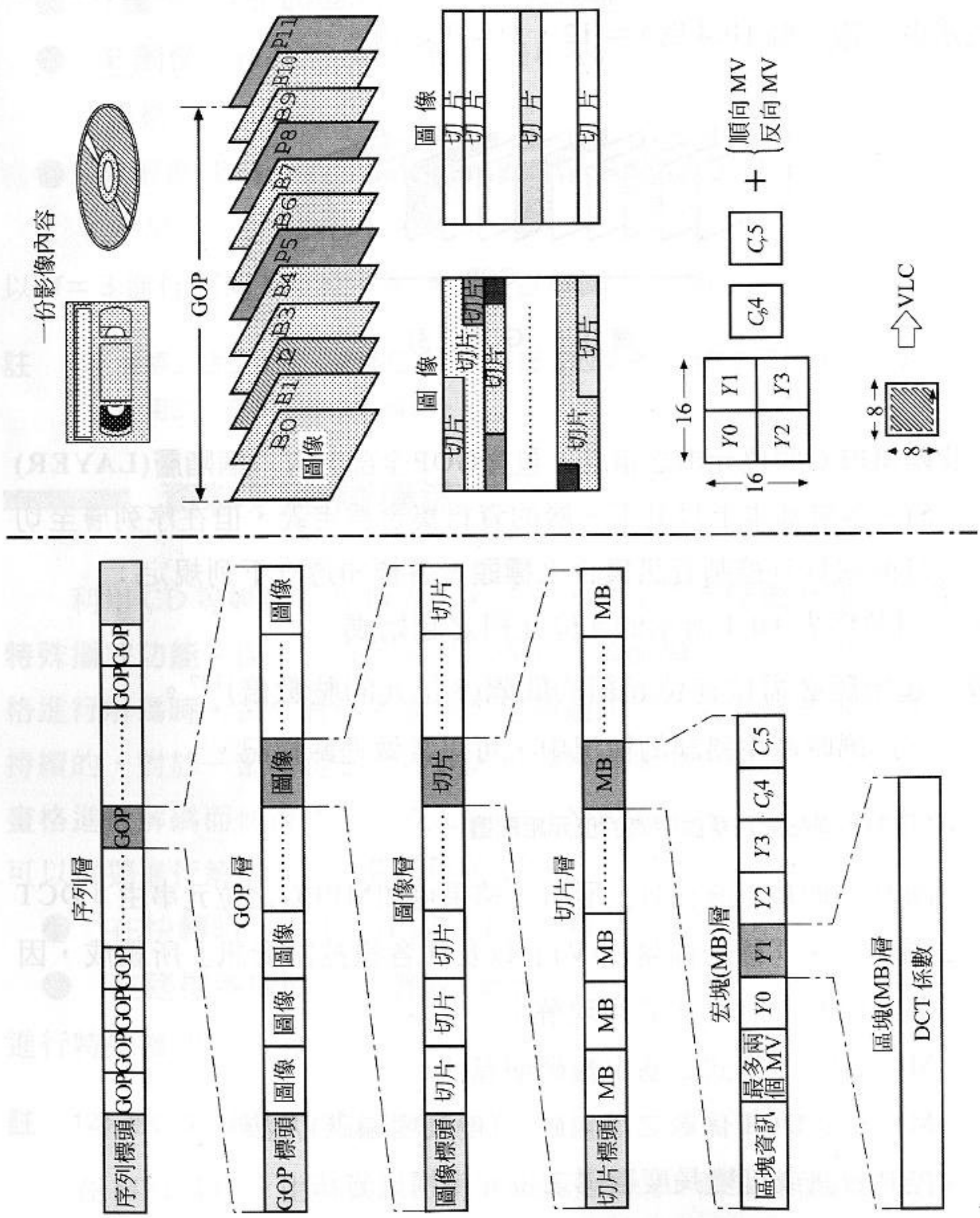


圖 10.5 MPEG 位元串之階層構造

表 10.1　MPEG 之主要起始碼

起始碼	十六進位表示
序列標頭碼	X'000001B3'
序列結束碼	X'000001B7'
GOP 起始碼	X'000001B8'
圖像起始碼	X'00000100'
切片起始碼	X'00000101'～X'000001AF'

MPEG-1 組成

雖然「在MPEG-1 規格書中，應該規定一對編碼／解碼器之演算法與其實施方法」是個極其自然的想法，但是實際標準化使是列出讓位元串可被解碼之**最低標準**。因此，舉例來說並不具體規定 ME 之實施方法與GOP構造之決定方法，而由設計者(使用者)來決定。接著說明關於此點的想法。

處理對象之影像訊號的格式

在第二章的開頭，計算了以 4：2：0 格式記錄電視廣播規格之 BT.601 動畫情況下之資料量。結果超過了 124Mbps，要降到MPEG-1 的目標資料率 1.55Mbps 需要達到 80：1 的壓縮比，如此一來將無法維持畫質。因此，MPEG 係以 BT.601 規格縮小一半之 SIF(參照表 1.1)爲影像之處理對象。由 BT.601 要求得 SIF 格式時，將第一區域(圖 7.3)水平線上的像素進行 1/2 之減樣即可。藉此可得到漸次掃描的每秒 30 畫格之動畫影像[128]。

註 128.720/2 = 360，將 SIF 之水平像素僅減少 8 像素，仍可將像素之尺寸保持在 16 的倍數。

原 SIF 之資料率約爲 30Mps，因此所要求的壓縮比約爲 20：1，降低至可實現的範圍。

基本編碼演算法

極端的說，MPEG-1 爲「將 4：2：0 格式之影像訊號，進行半像素(或是全像素)精度之順向或是雙向MC與DCT轉換之演算法」，其標準編碼器如圖 10.6 所示[129]。

註 129.所謂的「標準」代表符合 MPEG 之最低要求。舉例來說，在圖 10.6 中雖僅對 Y分量進行動態推測，也可對於顏色分量(姑且不論效果與效率)進行動態推測。這就是「標準」的意思。

動態推測／補償(ME ／ MC)部份之組成

在 MPEG 之 ME ／ MC，係以各畫格一致以 切割之**宏塊**(MB)爲單位，進行時精度爲一像素或是半像素。在 I 圖像不進行 MC，在此不加以考慮。關於 P 圖像，在一七七頁中之說明雖爲「僅進行順向 MC」，但是實際上可以在「**順向MC**」與「**內部編碼**」兩種模式中，選擇效率較佳的一種[130]。另外，在 B 圖像中，可在在「**順向 MC**」、「**反向 MC**」、「**內插 MC**」及「**內部編碼**」四種模式中選擇一種。

註 130.在編碼端之模式選擇基準由設計者自行裁量(B 圖像亦同)。

因爲 MPEG-1 使用 4：2：0 格式 MB 如圖 10.7 所示，由一個尺寸爲 16×16 的Y分量，各一個 8×8 之C_b、C_r分量所組成，MV 以各 MB 爲單位僅提供Y分量，由YC_bC_r三分量共同使用。關於C_bC_r，配合Y分量之減樣比，MV 取 1/2 倍[131]。

註　131.正確地說，各分量之絕對值取二進位之後，向右取一位元，小數點第二位捨去後再進行解碼，在此操作下，MV = −.35 時為−1.5。

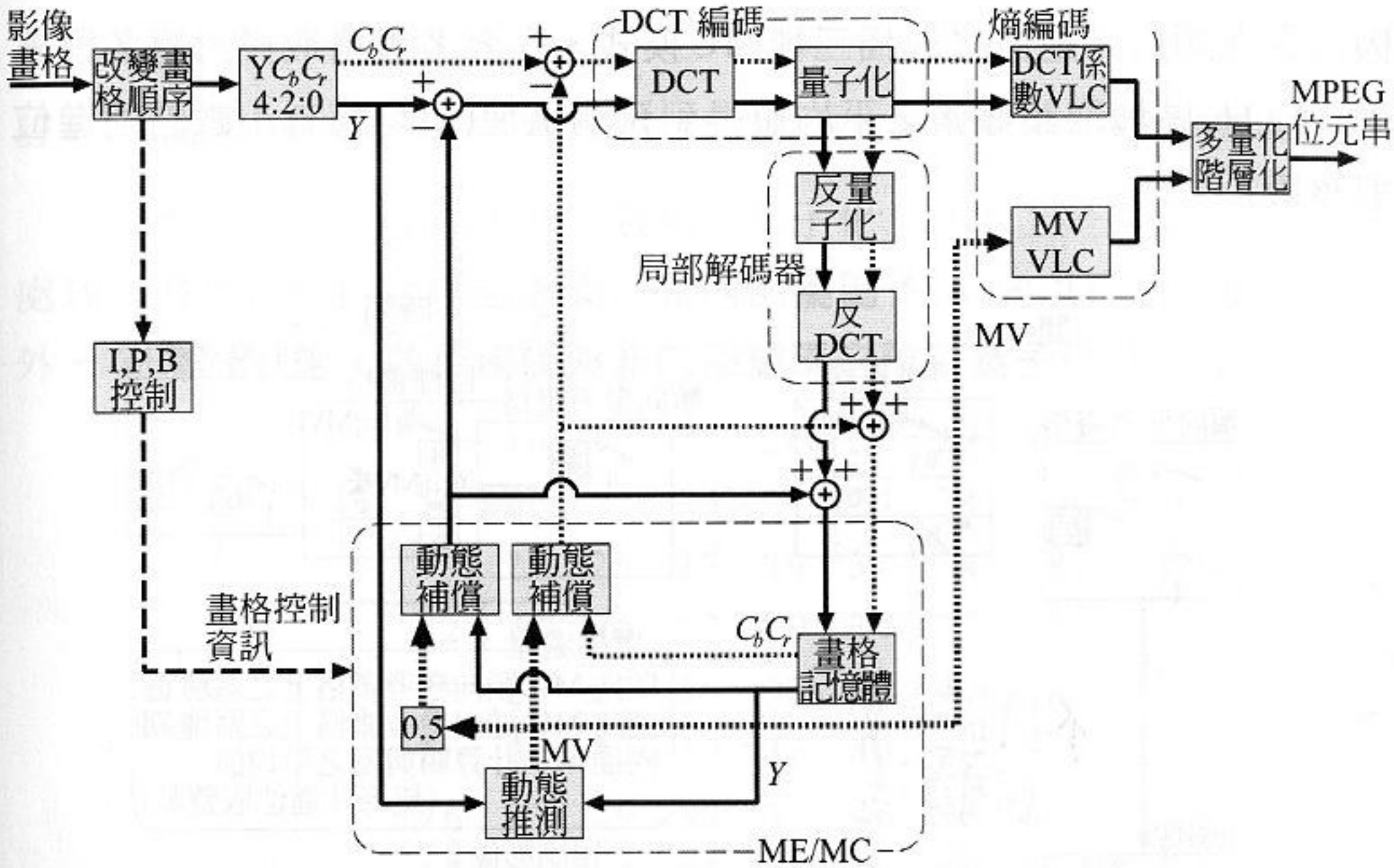


圖 10.6　MPEG 編碼器之標準構成

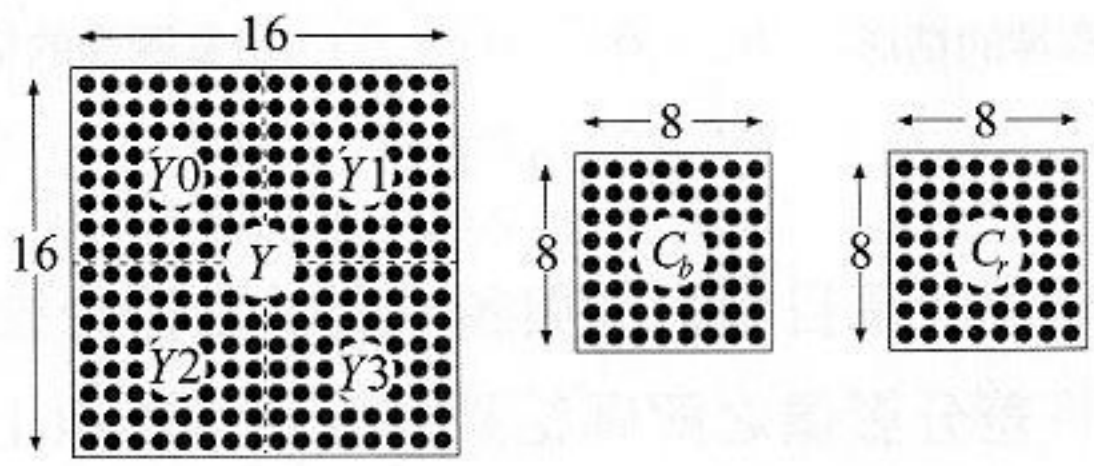


圖 10.7　MPEG 之宏塊的組成

MC之具體的演算法如圖 10.8 所示，首先，P、B圖像之共通點爲，禁止參考參考畫格以外的像素。(同一圖(a)之區塊 b)。另外，當 MV 包

含 0.5 的零頭時，需要進行(7.13)、(7.14)所示之內插(將內插的結果以**四捨五入**加以整數化)。對於 P 圖像(a)依照各 MB 之順序切出 MV 指出之 16×16 區塊，貼在預測影像之對應位置得到預測影像。對於B圖像(b)因爲存在順向／逆向／內插三種MC模式，在參考畫格取得對應各模式的 16×16 區塊取該像素之平均值得到預測區塊(平均值有小數點則**進位**取整數)。

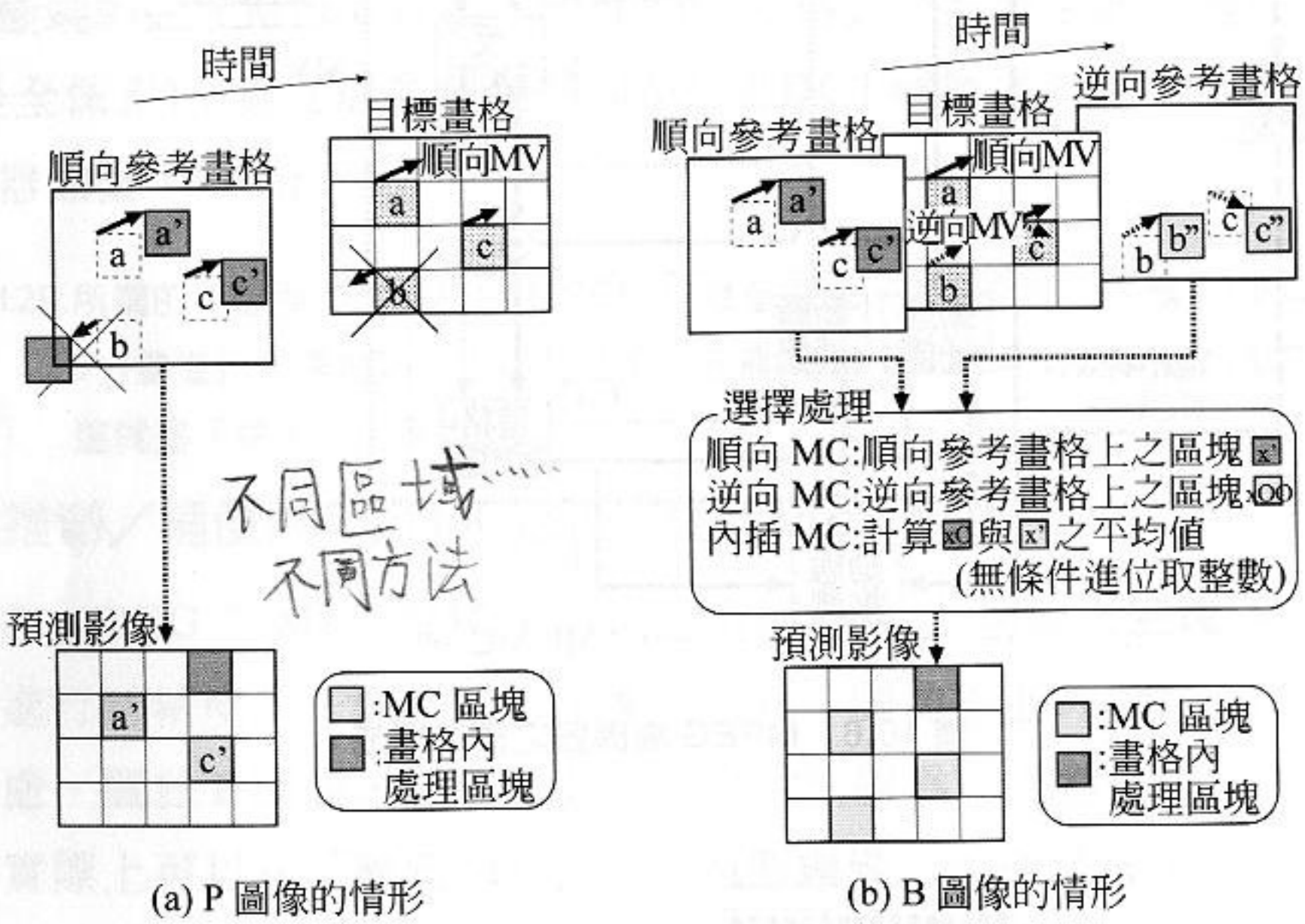


圖 10.8 MC

之後，在編碼端計算目標影像與預測影像之差分後進行DCT編碼。另外在解碼端，將差分影像之解碼結果加到預測影像上以重現原影像。

DCT 編碼端之組成

作爲編碼對象之 MB，可大致區分爲**內部編碼 MB** 與**非內部編碼 MB**。前者爲I圖像內的所有MB與P／B圖像之內部編碼區塊，後者爲P／B圖像之**差分區塊**。

這兩種 MB，首先皆進行 8×8 的 DCT。MB 的尺寸爲 16×16，因此 Y分量可如圖 10.7 分割成 $Y0$～$Y3$的四個 8×8 區塊，再分別進行DCT。對於 $C_b C_r$ 分量則進行 8×8DCT。如此一來，一個MB(最多[132])包含 6 個DCT 區塊。

註　132.在非內部編碼MB中，如同後述，以CBP編碼將不進行DCT的區塊加以排除。

接著，對所求得的DCT係數加以量子化。基本上是與JPEG相同的處理，但其特徵爲對**內部編碼／非內部編碼 MB 施以不同的處理**。另外，在MPEG中，**內部編碼與非內部編碼之預設量子化表**，分別爲：

$$Q_{\text{Intra}}(u,v)=\begin{bmatrix} 8 & 16 & 19 & 22 & 26 & 27 & 29 & 34 \\ 16 & 16 & 22 & 24 & 27 & 29 & 34 & 37 \\ 19 & 22 & 26 & 27 & 29 & 34 & 34 & 38 \\ 22 & 22 & 26 & 27 & 29 & 34 & 37 & 40 \\ 22 & 26 & 27 & 29 & 32 & 35 & 40 & 48 \\ 26 & 27 & 29 & 32 & 35 & 40 & 48 & 58 \\ 26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\ 27 & 29 & 35 & 38 & 46 & 56 & 69 & 83 \end{bmatrix} \tag{10.1}$$

$$Q_{\text{Nonintra}}(u,v)=\begin{bmatrix} 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \end{bmatrix} \tag{10.2}$$

而對 $YC_b C_r$ 是共通的[133]。

註　133.在JPEG中，可以依亮度／顏色分量之不同而定義不同的量子化表，但在MPEG中，兩者是使用相同的對照表。另外，使用者也可以使用自製的量子化表(參照後頭所述的序列標頭)。

內部編碼 MB 之 DCT 係數$F_{\text{intra}}(u,v)$，可用決定量子化表倍率之參數q_s(稱爲**量子化尺度**)依照：

$$F'_{\text{Intra}}(0,0)=\text{int}\left(\frac{F_{\text{Intra}}(0,0)}{8}+0.5\cdot\text{sgn}(F_{\text{Intra}}(0,0))\right) \tag{10.3}$$

$$F'_{\text{Intra}}(u,v)=\text{int}\left(\frac{8\times F_{\text{Intra}}(u,v)}{q_s\times Q_{\text{Intra}}(u,v)}+0.5\cdot\text{sgn}(F_{\text{Intra}}(u,v))\right) \tag{10.4}$$

一致地量子化(參照式(4.24))。在解碼的時候亦需要量子化尺度q_s，如後所述記載於位元串中。另外，如式(10.3)所示，內部編碼 MB 之 DC 係數**通常以數值 8 為單位加以量子化**。另外，非內部編碼MB之DCT係數$F_{\text{Nonintra}}(u,v)$之 DC、AC 係數，則是以：

$$F'_{\text{Nontnra}}(u,v)=\text{int}\left(\frac{8\times F_{\text{Intra}}(u,v)}{q_s\times Q_{\text{Intra}}(u,v)}\right) \tag{10.5}$$

進行具有死帶之一致量子化(參照式(4.27))[134]。另外，在反量子化時，內部編碼／非內部編碼分別依照式(4.26)與(4.28)進行對應的操作。

註 134.此與 H.261 具有高度的共通性，另外，內部編碼 MB 之一致量子化與 JPEG 具有高度共通性。

當反量子化時所得到的 DCT 係數$F''(u,v)$爲**偶數**時，在 MPEG-1 中，以

$$F''(u,v)\Leftarrow F''(u,v)-\text{sgn}(F''(u,v)) \tag{10.6}$$

予以強制地**奇數化**。此乃爲了降低編碼端與解碼端之間的**比對錯誤**而插入之處理，請詳情可參照文獻[60]。

DCT 係數之熵編碼部份之組成

該處理與JPEG幾乎相同，但是考慮內部編碼／非內部編碼之DCT係數的特徵，對兩者進行不同的處理。但是以固定的長度安排可變長度編碼這點與 JPEG 不同。

◆內部編碼 MB 的情形

與 JPEG 相同，對 DC 與 AC 係數各別的處理。爲了除去 DC 係數區塊間的相關，使用**差分編碼**。因爲一個MB包含四個Y區塊與各一個C_b、C_r區塊，依照圖 10.9 所示順序求取之前區塊之 DC 係數，對此進行編碼。實際上，依照接下來的順序。

對 DC 係數進行編碼之演算法

① 在表 10.2 中，求取DC差分值X所屬之群組編號i，配合Y與C_bC_r賦予表中之可變長度編碼(VLC)。

② 在群組i內最左端之 DC 差分值爲第 0 個，再算X爲第幾個，將其以對應附加位元長度之二進位表示。

③ 在配合群組編號所對應的VLC之後加上附加位元，即得到X的編碼。

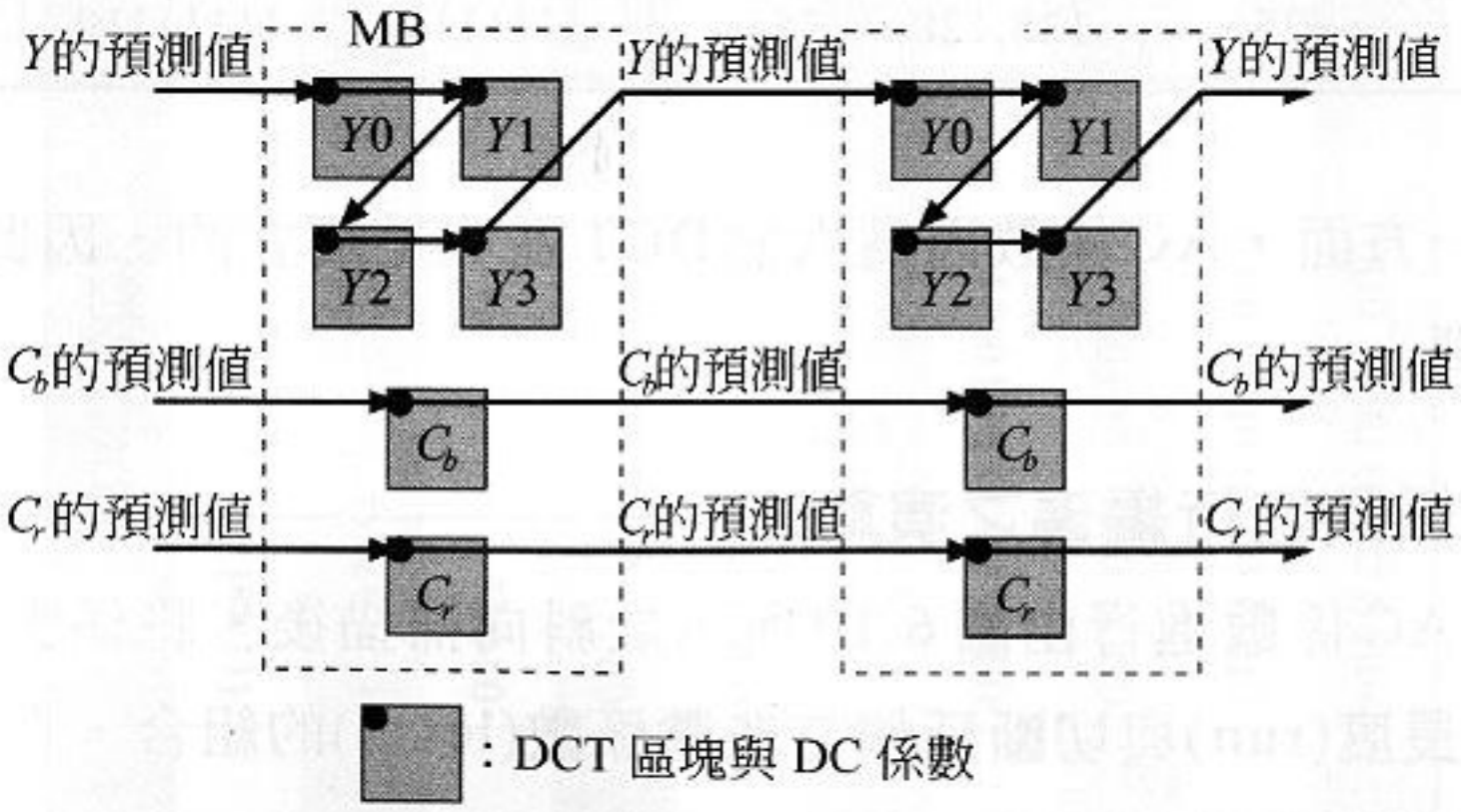


圖 10.9　內部編碼 MB 之 DC 係數與差分編碼

但是，爲了配合解碼誤差，在一定的間隔之後需對預測值加以歸零(刷新)。爲此，在I圖像中在切片的起始以將預測值歸零至128，對於P／B圖像內之內部編碼MB，除了切片的開頭之外、在**略過MB的情形**(後述)以及**非內部編碼MB**的情形中，分別將預測值歸零爲128。

表10.2　DC差分值之群組化與附加位元長度

群組編號	DC差分值	vlc(二進位)		附加位元長度
		Y	C_bC_r	
1	0	100	00	0
1	−1,1	00	01	1
2	−3,−2,2,3	00	10	2
3	−7,⋯,−4,4,⋯,7	101	110	3
⋯	⋯	⋯	⋯	⋯
6	−63,⋯,−32,32,⋯,63	11110	111110	6
7	−127,⋯,−64,64,⋯,127	111110	1111110	7
8	−255,⋯,−128,128,⋯,255	1111110	11111110	8

另外一方面，AC係數因爲六個DCT區塊是獨立的，因此以下列順序加以編碼。

對AC係數進行編碼之演算法

① 對AC係數進行由圖6.10所示之斜向掃描後，將係數表示爲**延伸長度**(run)與**切斷延伸之非零係數**(level)的組合，在最後加上區塊結束碼(EOB)。

表 10.3　DCT 係數之 VLC

run/1v1	VLC(二進位)	run/1v1	VLC(二進位)	run/1v1	VLC(二進位)
0/1	1(first)	1/1	011s	10/1	0010 0111s
0/1	11s(next)	1/2	0000 110s	10/2	0000 0000 1000 0s
0/2	0100s	1/3	0010 0101s	11/1	0010 0011s
0/3	0010 1s	…	…	11/2	0000 0000 0001 1010s
0/4	0000 110 s	2/5	0000 0000 1010s	…	…
…	…	…	…	29/1	0000 0000 0001 1101s
0/37	0000 0000 0010 011s	8/1	0000 111s	30/1	0000 0000 0001 1101 s
0/38	0000 0000 0010 010s	8/2	0000 0001 0001s	31/1	0000 0000 0001 1111 s
0/39	0000 0000 0010 001s	9/1	0000 101s	EOB	10
0/40	0000 0000 0010 000s	9/2	0000 0000 1000 1s	ESC	0000 01

註：s 爲表示 level 之編碼，當 level 爲正的情形爲 0，負的時候爲 1。

② 對於各個 run ／ level、EOB 以表 10.3 之 VLC 加以分配(內部編碼 MB 以對 0/1 對應的 VLC 再加上 next 加以記錄爲$(11s)_2$)，再進行熵編碼。

③ 在表 10.3 找不到相對應的組合時，在輸出ESC碼之後，依照表 10.4 將 run 以 6 位元的固定長度，level 以 8 位元或是 16 位元的固定長度分別加以二進位編碼。

另外，與 JPEG 不同，當 MPEG 的第 63 個係數不爲 0 時也要附加 EOB。另外，在表 10.3 中，與 0/1 相對應的 VLC 有 first 與 next 兩種，此外，在first(正值)的情形中，與EOB是重複的，這點相當有意思。關於此點在之後加以說明。

◆非內部編碼 MB 的情形

非內部編碼 MB 之 DC 係數表示「預測誤差在區塊內的平均」，因此可將區塊間相關係數視爲相當低。因此，可將 DC 與 AC 加以合併，將 64 個 DCT 係數均以之前所述之**AC 係數之編碼演算法**加以編碼。

另外，在此情形中，MB 爲差分區塊，如圖 10.9 所示，在 6 個區塊之中會有幾個(無意)在量子化之後 DCT 係數會通通變爲 0，因此，使用後述的編碼區塊圖形(Coded Block Pattern：CBP)，指定 MB 內的六個區塊之**有意義**(非零)區塊，對該區塊進行編碼。如此一來，在非內部編碼 MB 中，不會有某些 DCT 係數在 VLC 開始之後隨即就是 EOB[135]。

註 135.但是，在 EOB 之後該區塊全為 0，本來即不屬 CBP。

基於以上的理由，在非內部編碼MC當中，使用表 10.3 之VLC時，在編碼開始之後的**第一個(first)**vlc將$(10)_2$以$(1s)_2$之形式(發生機率較高)表示成 0/1 的組合，從**下一個**編碼**(next)**開始將$(10)_2$讓給 EOB，而 0/1

使用$(11s)_2$。另外，內部編碼mb中，最初的AC係數會成爲EOB，無法使用這種「技巧」。表 10.3 之 0/1 **經常使用**表示爲 next 之$(11s)_2$。

表 10.4　使用 ESCAPE 碼之固定長度編碼

run	固定長度編碼
0	0000 00
1	0000 01
2	0000 10
3	0000 11
4	0001 00
5	0001 01
6	0001 10
…	…
58	1110 10
59	1110 11
60	1111 00
61	1111 01
62	1111 10
63	1111 11

level	固定長度編碼
−255	1000 0000 0000 0001
…	…
−299	1000 0000 0111 1111
127	1000 0001
…	…
−2	1111 1110
−1	1111 1111
+ 1	0000 0001
+ 2	0000 0010
…	…
+ 127	0111 1111
+ 128	0000 0000 1000 0000
…	…
+ 255	0000 0000 1111 1111

移動向量之編碼部份之組成

各MB中，P圖像最多有一個，B圖像最多有兩個MV，各自由水平與垂直方向之純量所組成。在MPEG中，MV之純量部份在進行編碼時

規定使用共通的演算法，舉例來說，B圖像中，最多四次將所有的分量加以編碼。另外，MV之水平與垂直分量，因爲分別與其臨近部份**高度相關**[136]，因此使用差分編碼。

註 136.舉例來說參照圖7.14(a)之MV場。

在 MPEG 中，可以圖像爲單位選擇整數像素精度(Full-pel)與半像數精度(Half-pel)之 MC[137]。半像素精度中，MC 會產生 0.5 的零頭，爲了進行與整素像素精度相同的處理，在半像素精度的情形中對各分量乘上兩倍之後，以**整數進行編碼**。

註 137.為了表示此精度，在圖像標頭中採用1位元的編碼(FPFV、FPBV)(參照後述之圖像標頭)。

具體來說首先對編碼對象之分量(以MV表示)與之前進行編碼後相同分量的**預測值**之間的差pMV，求取差分

$$dMV = MV - pMV \tag{10.7}$$

接著將dMV如表 10.5 所示以 VLC 編碼。

但是，在 I／P 圖像間有複數個 B 圖像存在在距離數幅畫格之間仍需要ME／MC，一般的MV值會變得相當大。在這種情形中，如表 10.5 所示，在[−16,16]範圍間的 VLC 會不夠。因此，在 MPEG 利用所謂的**f 碼**，擴大表 10.5 的範圍後進行編碼。

f 碼爲取 1～7 的值之三位元之固定長度符號[138]，以圖像爲單位。f 碼(其值爲 f_{code})依照：

$$f = 2^{f_{\text{code}}-1} \tag{10.8}$$

註 138.為了避免誤判為起始碼，不使用0值。

表 10.5　MV 之 VLC

dMV	VLC
0	1
1	01
2	001s
3	0001s
…	…
15	0000 0011 01s
16	0000 0011 00s

註：s爲表示正負號，正：0，負：1。

表 10.6　f 碼與放大率 f 等的關係

f_{code}	f	編碼可能之 dMV 的範圍[low,high]	r之位元數
1	1	[−16,15]	0
2	2	[−32,31]	1
3	4	[−64,63]	2
4	8	[−128,127]	3
5	16	[−256,255]	4
6	32	[−512,511]	5
7	64	[−1024,1023]	6

對於表 10.5 之分量訂定放大率。f 碼與放大率 f，以及可進行編碼之dMV範圍[low,high]如表 10.6 所示。具體來說，使用放大率 f 將dMV以

$$dMV = dMV_p \times f - \mathrm{sgn}(dMV_p) \times r \quad (r > 0) \tag{10.9}$$

分解爲主值dMV_p與剩餘r。此分解可用圖 10.10 加以說明(圖的垂直方向爲dMV，水平方向爲 f 表示放大率，在圖中僅表示$dMV > 0$的區域)。與某個放大率對應的垂直軸上，將dMV以 f 個由□圍起之群組加以分解。個別對應至一個 $f = 1$(表 10.5)上之dMV。進行式(10.9)之分解。主值dMV_p表示dMV所屬之群組編號，剩餘r表示在群組內之位置。舉例來說，$f_{\mathrm{code}} = 3$($f = 4$)中，$dMV = 10$，可分解爲$dMV_p = 3$，$r = 2$。爲了要將群組內的 $f = 2^{f_{\mathrm{code}}-1}$個元素進行一致的指定，$r$需要爲($f_{\mathrm{code}} - 1$)(位元)之固定編碼。因此，$dMV_p$以表 10.5 之 VLC 進行編碼。另外$r$的值爲以

固定的($f_{\text{code}}-1$)(位元)爲 **1 之補數表現**(r以二進位表示之後，每個位元都會加以翻轉)，連繫兩者爲dMV之編碼。舉例來說，$f_{\text{code}}=3$，$dMV=10$，對應於$dMV_p=3$之 VLC 在表 10.5 中可得$(00010)_2$，另外，$r=2$，爲 $f_{\text{code}}-1=2$ 位元，1 的補數表現爲$(01)_2$，將兩者合併之後，可編碼爲$(0001001)_2$。另外，$dMV=0$ 爲例外，在此情形中不需要附加剩餘r(只要有dMV_p即可一致地決定)。

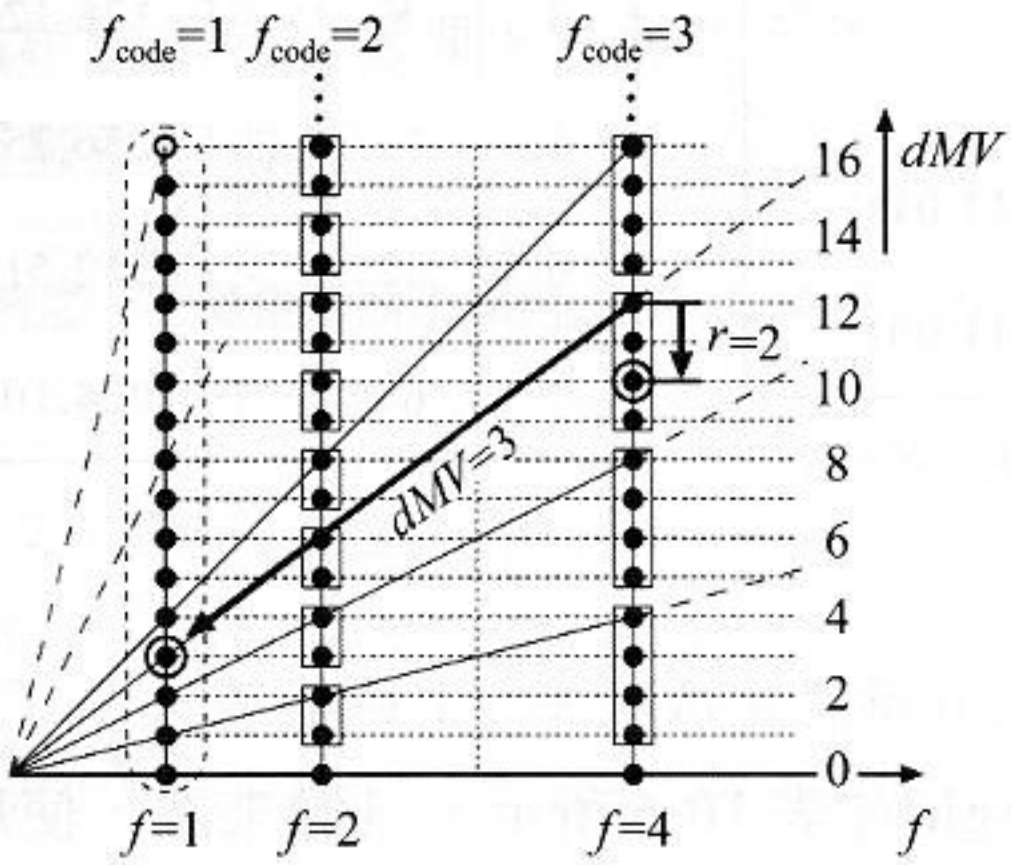


圖 10.10　利用 f 碼擴大編碼範圍

一般以$[-X,X]$爲範圍進行某個值之差分編碼後，其差分值會分布於$[-2X,2X]$，因此與直接編碼相較，編碼的符號數會變爲兩倍，因此效率較差。在 MV 的編碼中以$32\times f$進行**模數運算**以迴避此問題。

在決定f_{code}之後可依表 10.6 進行編碼的dMV範圍[low,high]也隨之決定。在編碼時，依照式(10.7)計算之差分dMV超出對應範圍[low,high]的情形中，依照模數運算：

$$dMV \Leftarrow \begin{cases} dMV+(32\times f) & (dMV<\text{low}) \\ dMV-(32\times f) & (dMV>\text{high}) \end{cases} \tag{10.10}$$

將dMV強制地納入[low,high]之後，再進行編碼。雖然這種處理無法解碼出正確的MV，但是實際上因爲[low,high]係明確加以定義的，因此在解碼時進行相同的模數運算時可以得到正確的值。舉例來說 $f_{\text{code}} = 3(f = 4)$，對於$pMV = 60$，$MV = -62$進行編碼的情形。此時，在式(10.7)中$dMV = -62 - 60 = -122$，但代入式(10.10)進行模數運算後，可得$dMV \Leftarrow -122 + 32 \times 4 = 6$，對該值進行編碼。在解碼端得到$dMV = 6$之解碼結果，試著依$MV = pMV + dMV$加以重現，得到$MV = 60 + 6 = 66$。該值已超過了編碼的上限high，因此依藉由進行模數運算$MV \Leftarrow 66 - 32 \times 4 = -62$，可以重現正確的值。也就是說，在解碼之時，進行下列運算即可[139]：

$$MV \Leftarrow \begin{cases} (MV + pMV) + (32 \times f) & (MV + pMV < \text{low}) \\ (MV + pMV) - (32 \times f) & (MV + pMV > \text{high}) \end{cases} \tag{10.11}$$

註　139.為了要使 MV 可依此模數運算一致地解碼，在表 10.6 中上限 high 為較下限之絕對值|low|小 1 的數值。

最後，歸納MV之差分編碼歸零之時機，在以下情形將預測值pMV**歸零到 0**。

P 圖像的情形

- 切片的開頭。
- 內部編碼 MB 的情形。
- 略過 MB 的情形。
- 具有 MV 之非內部編碼 MB 的情形。

B 圖像的的情形

- 切片的開頭。
- 內部編碼的情形。

移動,差分

各階層之組成

MB 層

在 MB 中，一定要給定宏塊位址增加量(Macro Block Address Increment：MBAI)以及 MB 種類(MBT)這兩個資訊，作爲解碼時的控制資訊。

在靜止之背景等部份中，MV＝(0,0)因此在取其差分之後誤差幾乎爲 0。這種 MB 不需要對 MV、DCT 係數一起進行編碼，可此可加以**省略**。MB 之省略資訊，由 MBAI 加以記載。

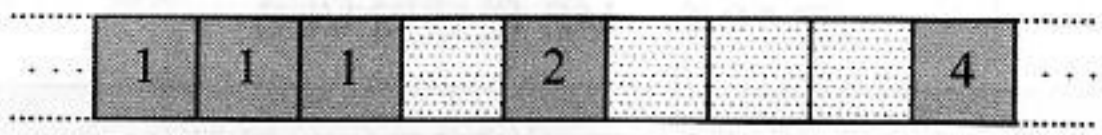
在圖像中MB中依照水平掃描順序賦予位址(左上之位址爲 0，參照圖 10.12)MBAI，基本上可表爲：

MBAI＝(編碼目標之MB位址)－(前一個編碼的MB位址)(10.12)

如此一來如圖 10.11 所示，當 MB 不被省略而連續進行編碼的情況中，MBAI＝1，在有數個 MB 遭省略的情形中，MBAI＝(省略的 MB 數＋1)[140]。但是，在切片的**開頭**與**結尾**，MB 的省略是**禁止**的[141]。另外，MBAI 與切片構造有關，因此在切片的**開頭**需**另外處理**(後述)。

註 140.對對於位址為 0 之 MB(沒有前一個編碼後的 MB)，式(10.12)之第二項以(－1)代入(參照圖 10.12(a))。

141.在 I 圖像內的 MB 無法省略。



1：編碼 MB 與其 MBAI　　：省略 MB

圖 10.11　MB 之省略與 MBAI 之例

MBAI 以表 10.7 之 VLC 加以記載，可利用之 VLC 最大值爲 33，超過的情形利用表中之 MB_ESC。MB_ESC 相當於 MBAI ＝ 33，當其爲超過 33 之值時以數個 MB_ESC 與一個 MBAI 之 VLC 的組合加以記載。另外，MB_STUFFING 爲防止下限溢位之特殊的無效碼。

表 10.7　MBAI 之 VLC

MBAI	VLC(二進位)
1	1
2	001
3	010
4	0011
5	0010
…	…
16	0000 0101 11
17	0000 0101 10
…	…
32	0000 0011 001
33	0000 0011 000
MB_ESC	0000 0001 000
MB_STUFFING	0000 0001 111

表 10.8　MB 種類與 VLC

內部編碼	CBP	反向 MV	順向 MV	量子化尺度	VLC(二進位)
I 圖像					
1	0	0	0	0	1
1	0	0	0	1	01
P 圖像					
0	0	0	1	0	001
0	1	0	0	0	01
0	1	0	0	1	0001
0	1	0	1	0	1
0	1	0	1	1	0010
1	0	0	0	0	0001 1
1	0	0	0	1	0001 01
B 圖像					
0	0	0	1	0	0010
0	0	1	0	0	010
0	0	1	1	0	10
0	1	0	1	0	0011
0	1	0	1	1	0000 11
0	1	1	0	0	011
0	1	1	0	1	0000 10
0	1	1	1	0	11
0	1	1	1	1	0001 0
1	0	0	0	0	0001 1
1	0	0	0	1	0000 01

另外，MB 種類爲表示 MB 當中的編碼資料種類之控制碼，依照圖像不同而賦予如表 10.8 之 VLC。在表 10.8 中，在內部編碼項目爲 1 時表示在該 MB 中進行內部編碼，另外自 CBP 至量子化尺寸的項目中爲「1」則表示該項資料存在(在表中不存在之資料組合是不允許的)。另

外，在表 10.8 中的CBP，如之前所述，用來表示非內部編碼MB中有意義的DCT區塊，使用表 10.9 之 VLC。

表 10.9　CBP 之 VLC

$Y0$	$Y1$	$Y2$	$Y3$	C_b	C_r	VLC
1	1	1	1	1	1	0011 00
1	1	1	1	1	0	0100 0
1	1	1	1	0	1	0101 1
1	1	1	1	0	0	111
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	0	0	0	0	1	1010
⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	0	0	0	1	1	0011 01
0	0	0	0	1	0	0100 1
0	0	0	0	0	1	0101 1

註：在$Y0$～C_r之項目中「1」表示對應的 DCT 區塊存在。

切片層

在MB內以VLC加以記載，另外也沒有起始碼，在解碼時一旦發生錯誤，之後的MB的解碼就無法進行。切片爲其開頭的配置使其可以修復錯誤之階層，由圖像在垂直方向上切割出的區域所組成，舉例來說如圖 10.12 所示，(a)爲圖像整體爲一個切片，(b)爲帶狀的分割，或是(c)爲任意分割，只要每個圖像相同可任意選擇。

爲了利用切片修復解碼的錯誤，需要各切片之起始位置(垂直水平位置)。切片之起始垂直位置以切片起始碼之最後一位元組表示(參照表10.1)。，水平位置如前述由MBAI加以記載，。也就是說，在賦予MBAI之式(10.12)中，除了**切片開頭**爲例外，第二項爲「上一行最右端之MB位址」。舉例來說，在圖10.12(c)中的切片3，切片起始碼爲X'00000103'，因此垂直位置爲 3，另外，上一行最右端之 MB 位址爲 9(參照同一圖(a))，因此由MBAI＝3可知水平位置。

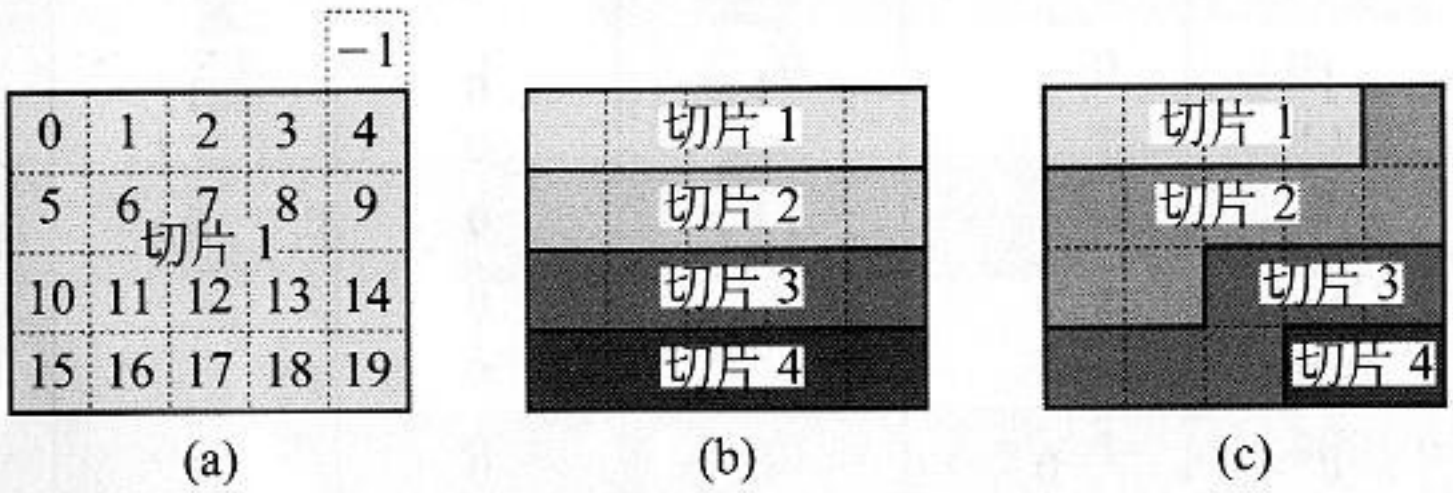


圖 10.12　切片之分割例((a)之數值表示 MB 位址)

其他階層

圖像層與切片層依照上述之順序加以記載，構成單一圖像之編碼資料。圖像編碼種類(Picture Coding Type：PCT)由表10.10中所示之固定三位元之編碼加以記載。另外，各圖像，在 GOP 中依序被賦予固定的編號(時間參考編號：TR)，在接下來所述之更動顯示順序時會被用到。

表 10.10　圖像種類編碼

圖像	PCT 編碼(二進位)
I 圖像	001
P 圖像	010
B 圖像	011
D 圖像	100

GOP 層爲構成 GOP 之所有圖像之集合。但是，寫入 GOP 的圖像順序與實際顯示的順序是不同的。舉例來說，在圖 10.4 之 GOP 組成中，在內部編碼 I2 之後要解碼 B 圖像 B3 與 B4 時需要參考影像 P5。因此，在 MPEG 中依照**參考影像出現先後**進行編碼(此稱爲**編碼順序**)，在圖 10.4 之 GOP 中依照：

顯示順序：	B0	B1	I2	B3	B4	P5	B6	B7	P8	B9	B10	P11
編碼順序：	I2	B0	B1	P5	B3	B4	P8	B6	B7	P11	B9	B10

進行圖像之編碼之後，再將其當作 GOP 資料寫入位元串中。

位元串之組成

在最後說明各階層之位元列的組成。如圖 10.5 所示，除了 MB 層與區塊層的各階層，具有作爲據以識別標頭之**起始碼**，在此寫入各種控制資訊。在標頭之後寫入下一階層之資訊。另外，爲了對各起始碼進行位元組配置，使用補碼「0」。

以下，對於各階層之位元串組成 以方塊圖加以說明，圖例如圖 10.13 所示。

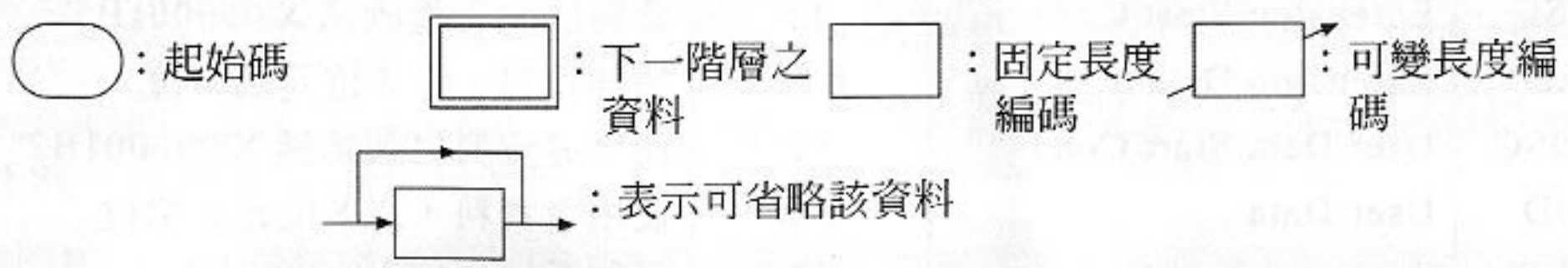
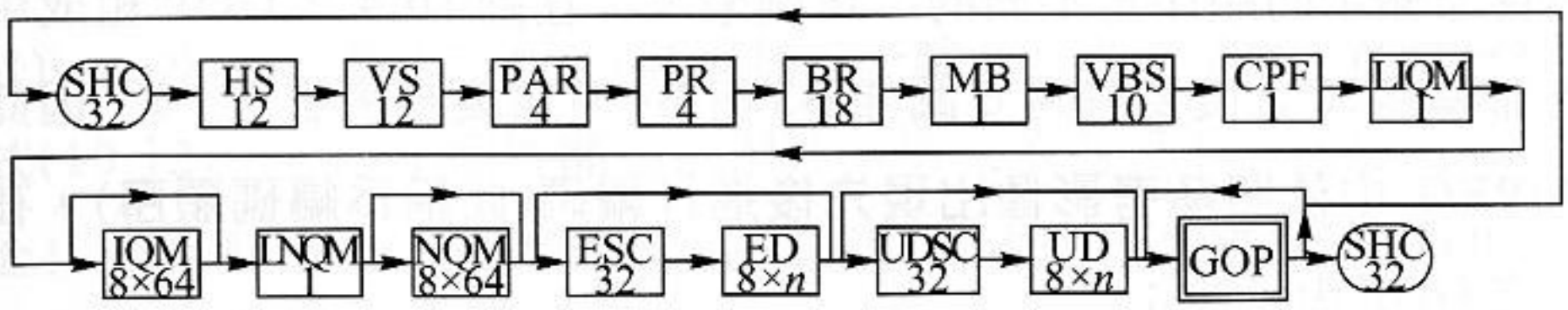


圖 10.13　各階層之方塊圖圖例

序列層



符號	名稱	位元長度	說明
SHC	Sequence Header Code	32	序列層起始碼 X'000001B3'
HS	Horizontal Size	12	水平像素數
VS	Vertical Size	12	垂直像素數
PAR	Pel Aspect Ratio	4	像素間隔之長寬比(表 10.11)
PR	Picture Rate	4	(表 10.12)
BR	Bit Rate	18	以 400 位元爲單位之位元流量，當所有的位元爲「1」時表示爲可變位元流量(VBR)
MB	Marker Bit	1	「1」
VBS	VBV Buffer Size	10	虛擬緩衝區之容量(＝8×2048×VBS)
CPF	Cons. Param. Flag	1	各參數位於限制區域內時爲「1」
LIQM	Load Intra Q Matrix	1	當賦予特殊之內部編碼量子化表時爲「1」
IQM	Intra Q Matrix	8×64	量子化表之值，64 個係數，各 8 位元
LNQM	Load Non Intra Q Matrix	1	當賦予特殊之非內部編碼量子化表時爲「1」
NQM	Non Intra Q Matrix	×64	量子化表之值，64 個係數，各 8 位元
ESC	Extension Start Code	32	資料展開之起始碼 X'000001B5'
ED	Extension Data	8×*n*	展開資料，以 8 位元爲單位
UDSC	User Data Start Code	32	使用者資料之起始碼 X'000001B2'
UD	User Data	8×*n*	使用者資料，以 8 位元爲單位
SEC	Sequence End Code	32	序列層結束碼 X'000001B7'

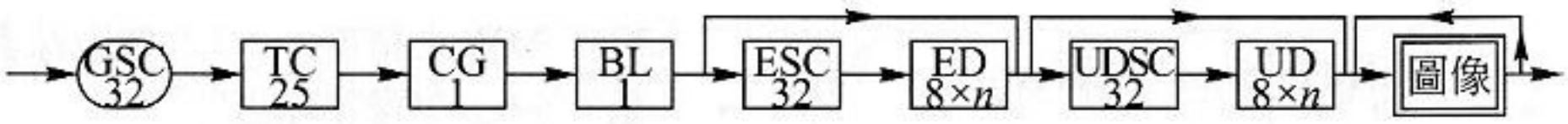
表 10.11　長寬比(PAR)

PAR	長寬比 (H/W)	影像訊號
0001	1.000	電腦
0010	0.6735	
0011	0.7031	16：9，625 條掃描線
…	…	…
1100	1.0950	BT.601，525 條掃描線
1101	1.1575	
1110	1.2015	

表 10.12　圖像速率

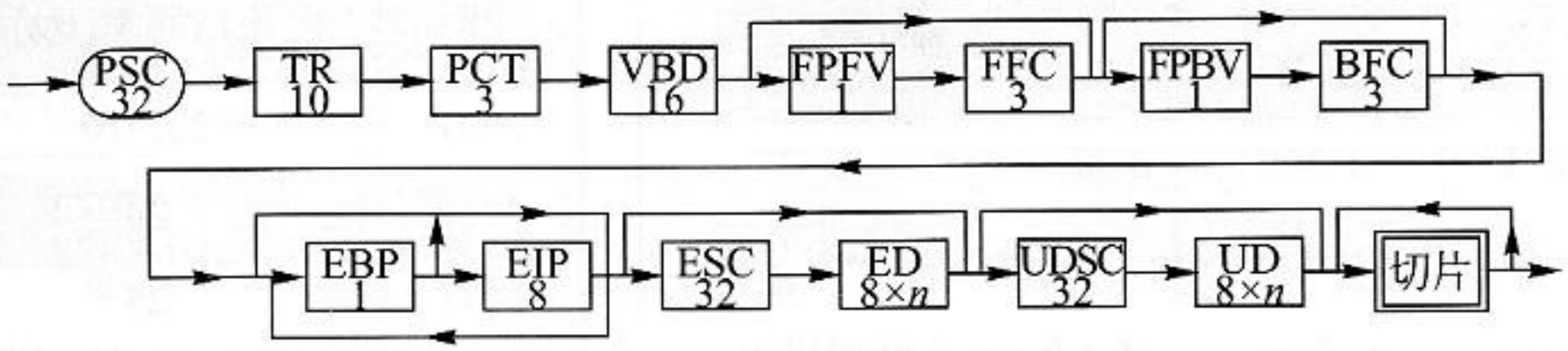
PR	畫格速率[Hz]
0001	23.976
0010	24
0011	25
0100	29.97
0101	30
0110	50
0111	59.94
1000	60

GOP 層



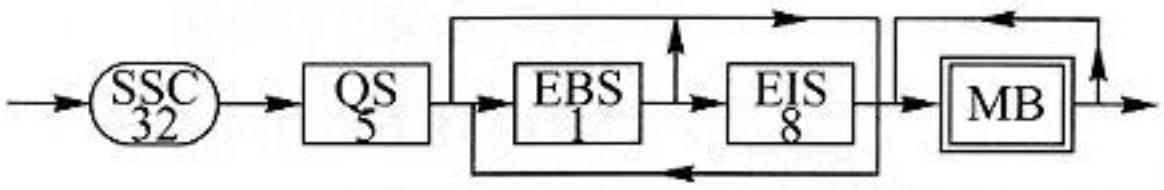
符號	名稱	位元長度	說明
GSC	Group Start Code	32	GOP 層起始碼 X'000001B8'
TC	Time Code	25	自序列之開頭的經過時間
CG	Cloesed GOP	1	當不解碼其他 GOP 也可解碼該 GOP 時爲「1」
BL	Broken Link	1	解碼之前的 B 圖像所要用到之畫格由於編譯錯誤，所以無法參考時爲「1」
ESC	參照序列層，以下同		
ED	參照序列層，以下同		
UDSC	參照序列層，以下同		
UD	參照序列層，以下同		

圖像層



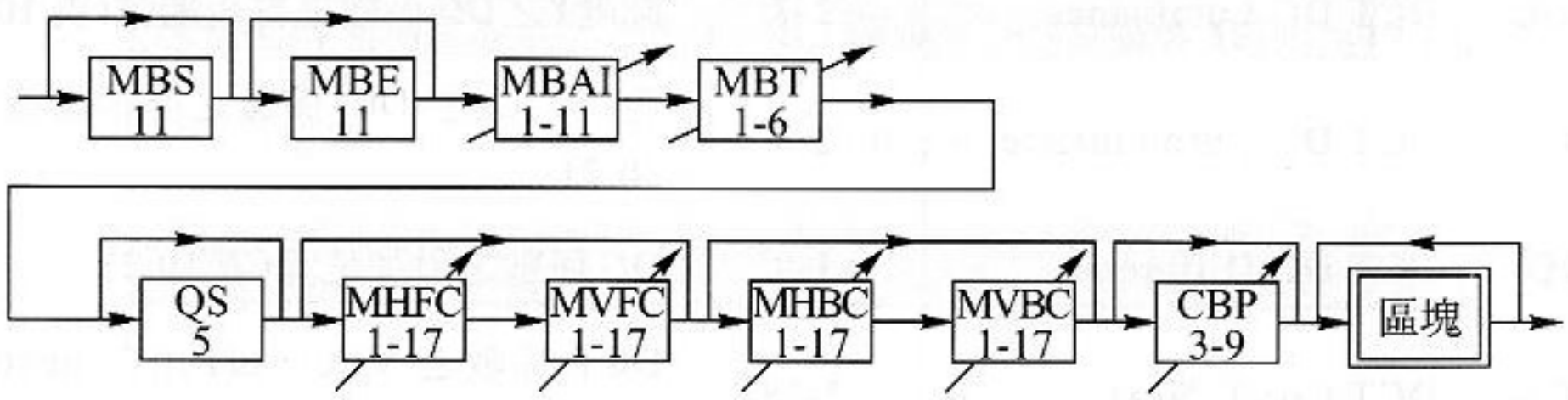
符號	名稱	位元長度	說明
PSC	Picture Start Code	32	圖像層起始碼 X'00000100'
TR	Temporal Reference	10	賦予圖像之參考編號
PCT	Picture Coding Type	3	I：$(001)_1$，P：$(010)_1$，B：$(011)_1$，D：$(100)_1$
VBD	VBV Delay	16	VBV 緩衝區之初始狀態
FPFV	Full Pel Forw. Vector	1	順向 MV 之精度。「1」整數像素精度，「0」半像素精度
FFC	Forw. f Code	3	順向 MV 之 f 碼
FPBV	Full Pel Back. Vector	1	反向 MV 之精度。「1」整數像素精度，「0」半像素精度
BFC	Back. f Code	3	反向 MV 之 f 碼
EBP	Extra Bit Picture	1	當有 EIP 時爲「1」，無則爲「0」
EIP	Extra Information Picture	8	其他的圖像資訊，以 8 位元爲單位

切片層



符號	名稱	位元長度	說明
SSC	Slice Start Code	32	序列層起始碼 X'00000101'～ X'000001AF'
QS	Quantizer Scale	5	量子化尺寸，以固定的五位元二進位表示
EBS	Extra Bit Slice	1	當有 EIS 時爲「1」，無則爲「0」
EIS	Extra Information Slice	8	其他的切片資訊，以 8 位元爲單位

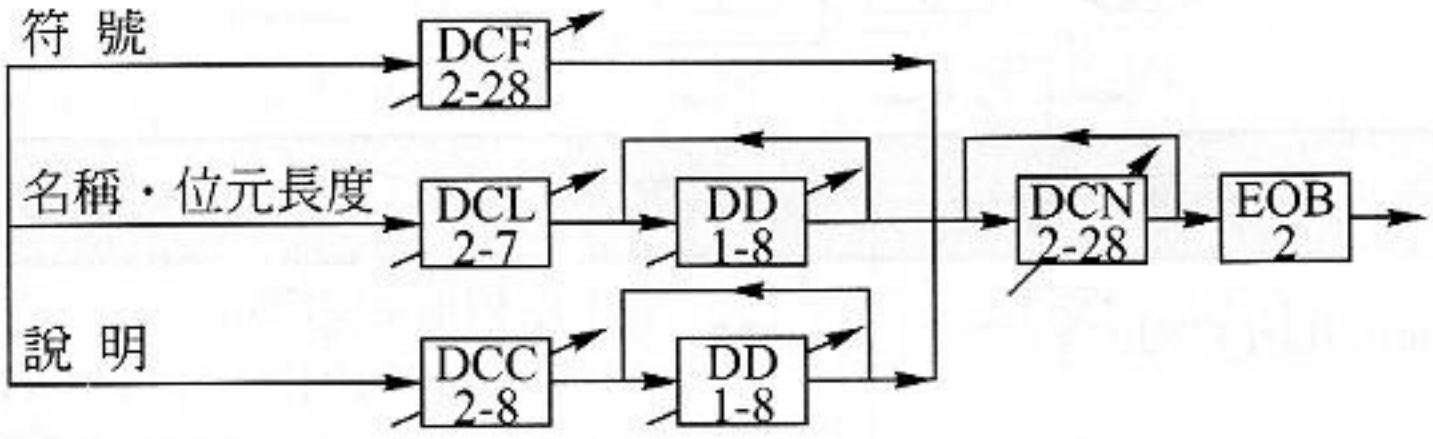
MB 層



符號	名稱	位元長度	說明
MBS	MB Stuffing	11	防止下限溢位之無效碼
MBE	MB Eacpae	11	MB 位址加上 33
MBAI	MB Address Increment	1-11	編碼 MB 之相對位置
MBT	MB Type	1-6	MB 的模式與表示編碼資料(註)
QS	MB Quantizer Scale	5	量子化尺寸，以固定的五位元二進位表示
MHFC	Motion Horiz. Forw. vector	1-17	順向MV水平分量之差分值，包含剩餘r
MVFC	Motion Verti. Forw. Vector	1-17	順向MV垂直分量之差分值，包含剩餘r
MHBC	Motion Horiz. Back. Vector	1-17	反向MV水平分量之差分值，包含剩餘r
MVBC	Motion Verti. Back. Vetor	1-17	反向MV垂直分量之差分值，包含剩餘r
CBP	COded Block Pattern	3-9	六個 DCT 區塊中之非零區塊

註：以下的 QS、MHFC、MVFC、MHBC、MVBC、CBP 的存在與否，由 MBT 加以控制(參照表 10.8)。

區塊層



符號	名稱	位元長度	說明
DCF	DCT Coeff. First	2-28	DCT 係數之 VLC，0/1 用在 first(表 10.3)
DCL	DCT DC Luminance	2-7	輝度Y之DC係數之群組編號(表 10.2)
DCC	DCT DC Chrominance	2-8	色差C_bC_r之 DC 係數之群組編號(表 10.2)
DD	DCT DC Difference	1-8	DC 係數之附加位元(表 10.2)
DCN	DCT Coeff. Next	2-28	DCT 係數之 VLC，0/1 用在 next(表 10.3)
EOB	End of Block	2	DCT 區塊之結尾

MPEG 位元串之具體例

以兩幅小畫格之編碼後之位元串作爲依MPEG-1 進行編碼的實例。

(a) 畫格 1

(b) 畫格 2

圖 10.14　編碼的畫格

圖 10.14 爲作爲編碼對象之兩幅畫格(尺寸 ，單色影像)都具有兩個 MB。在畫格 1 中，左側的 MB(MB1)僅區塊 Y0 有亮度變化，其他的區塊的亮度全爲 0。另外，畫格 2 僅爲畫格 1 在水平方向上向左平移三 3 個像素之後，再稍微增加整體的亮度(因此對畫格 2 進行ME時左側的MB1 之 MV 爲(3,0)，右側的 MB2 之 MV 爲(0,0))。

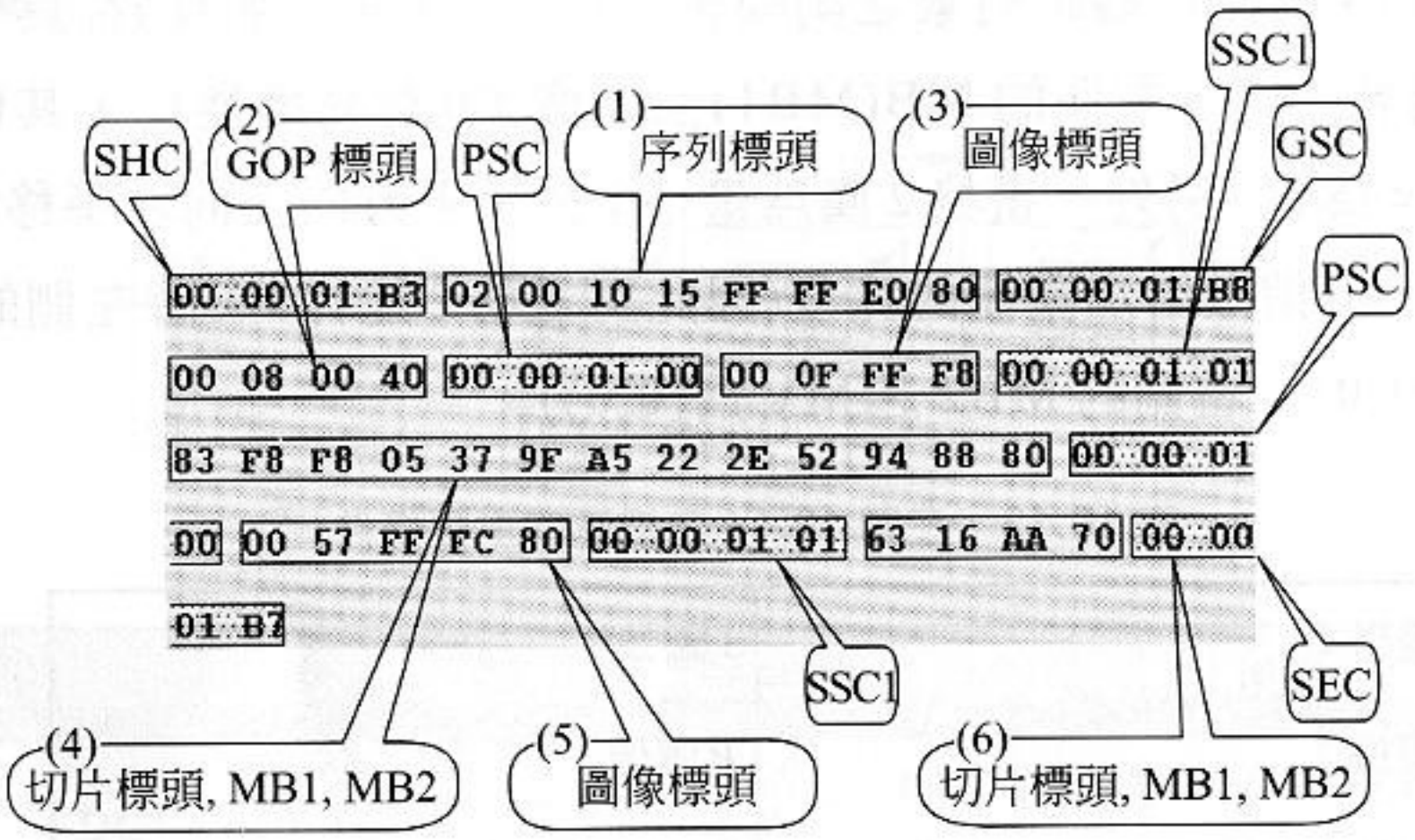
表 10.13　在圖 10.14 之編碼的主要條件

畫格 1	I 圖像
畫格 2	P 圖像
切片之組成	影像整體(MB1 與 MB2)
量子化增加幅度q_s	I 圖像：16，P 圖像 12(固定)
MC 的精度	Full-pel
f 碼	1

編碼使用的主要條件以表 10.13 表示，在實施編碼演算法時，在兩畫格中只有 Y0 之 DCT 區塊爲有意義的，分別爲：

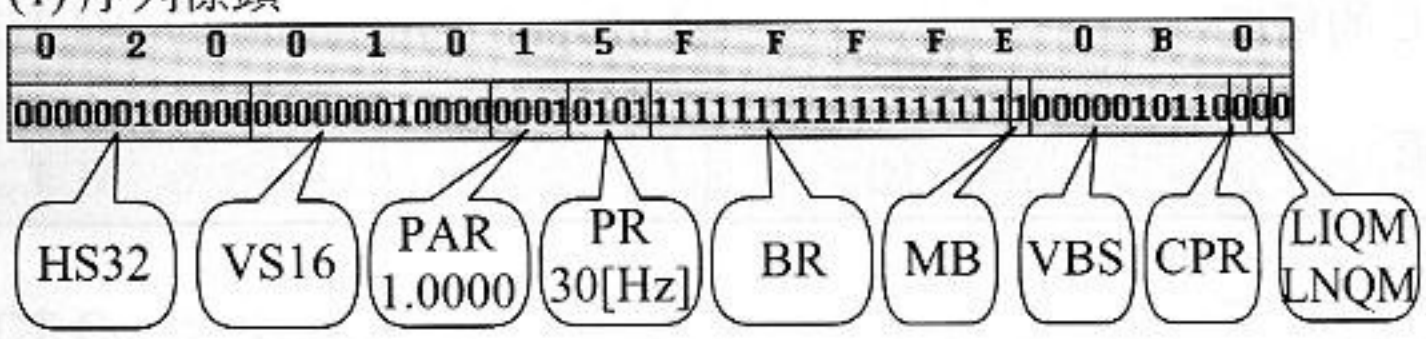
$$\underbrace{\begin{bmatrix} 32 & 0 & & \\ 0 & 0 & & \\ -5 & 0 & & \\ & & & 0 \end{bmatrix}}_{\text{畫格 1 之區塊 } Y0} \underbrace{\begin{bmatrix} 0 & 1 & & \\ 0 & 0 & & \\ 0 & 0 & & \\ & & & 0 \end{bmatrix}}_{\text{畫格 1 之區塊 } Y0}$$

與因爲目標影像爲單色的，C_b、C_r之 DC 係數僅爲值 128，這些在差分編碼後全會變爲 0。

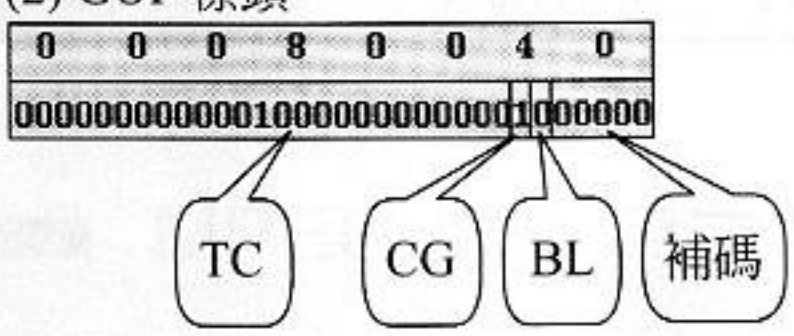


以下爲(1)~(6)各部份之細節

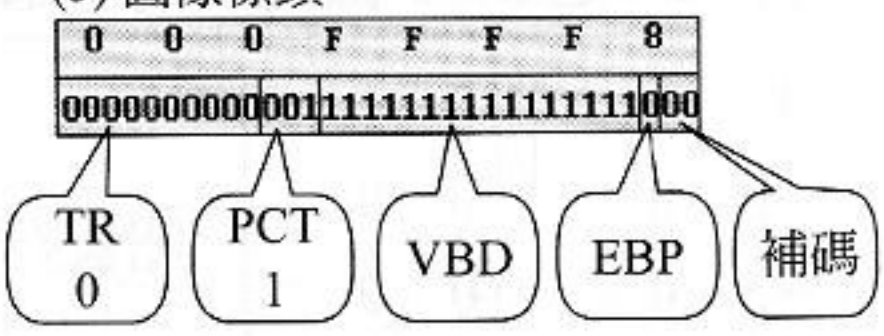
(1) 序列標頭



(2) GOP 標頭



(3) 圖像標頭



接次頁

圖 10.15　圖 10.14 以 MPEG-1 編碼後所得的位元串

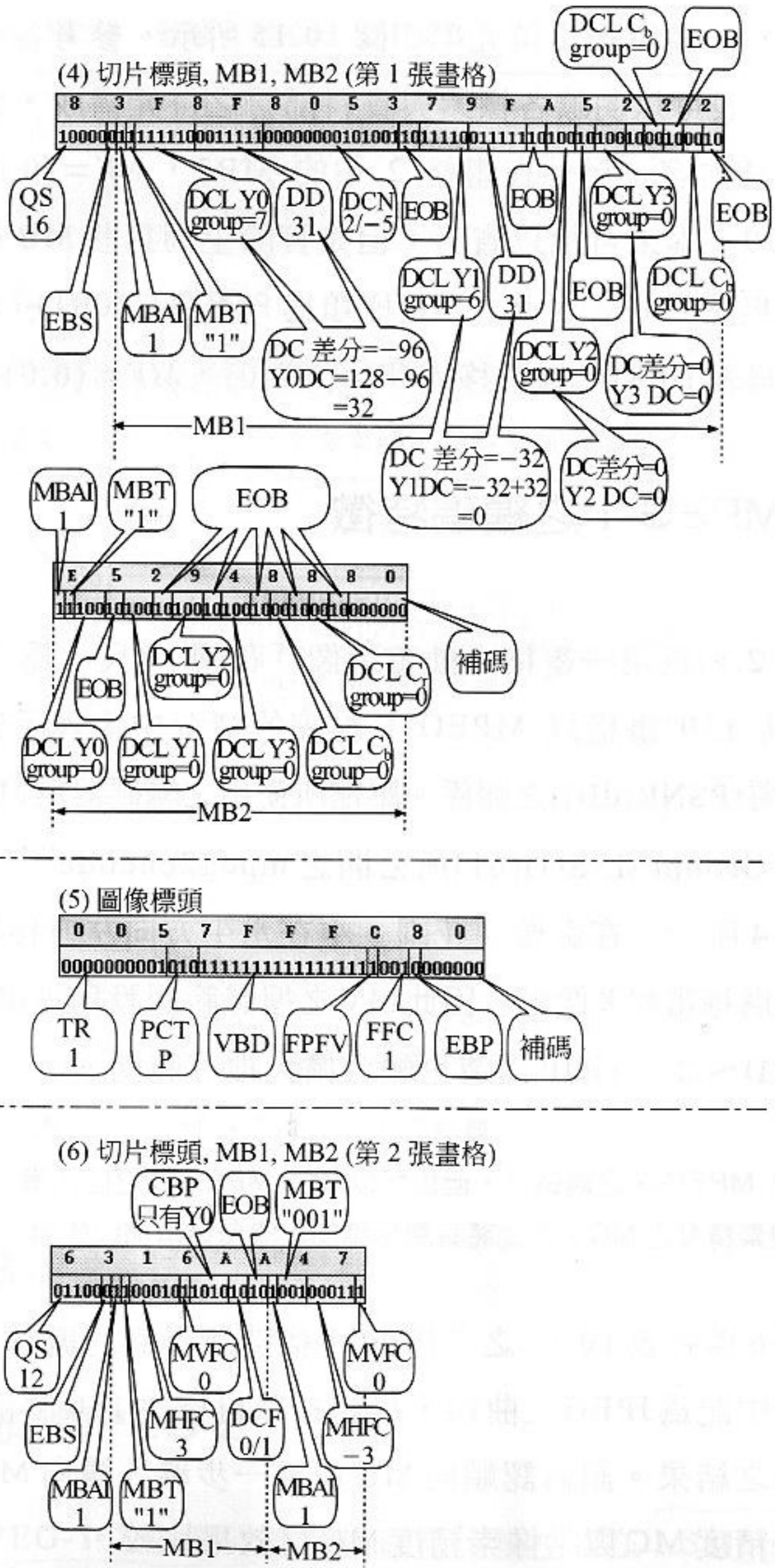


圖 10.15　圖 10.14 以 MPEG-1 編碼後所得的位元串(續)

用MPEG-1編碼後之位元串如圖10.15所示。參考各階層之位元串組成或 VLC 表可以明瞭各標頭與資料部份之位元構成，因此可更加清楚地理解本意內容。另外在畫格 2 中的 MB2，$MV=(0,0)$因此所有的DCT係數爲0，本來可加以省略，但是實際上因爲該MB相當於切片的端點，因此無法省略。因此，MB種類爲P圖像之$(001)_2$(模式僅包含順向MV，參照表10.8)，以平移$dMV=(-3,0)$，$MV=(0,0)$進行編碼。

MPEG-1之編碼特徵

以圖 2.2(a)爲第一畫格之動畫影像「花園」(尺寸爲 352×240，24位元全彩)共 150 畫格以 MPEG-1 壓縮的情況中，表示其編碼資料量([bpp])與畫質(PSNR[dB])之關係。壓縮所使用之編碼器爲MPEG Software Simulation Group(MSSG)[61]所公開之mpeg2encode [142]，編碼的各條件如表10.14所示。在影像「花園」中在水平方向左方移動之樹幹的速度最快，約爲每畫格8像素，因此MV之搜尋範圍設爲每畫格±10像素，表10.14之(1)～(3)之GOP構造也配合擴大搜尋範圍。

註 142.原為 MPEG-2之編碼器，但也可以產生 MPEG-1之位元串。但是無法選擇整數像素精度之MC，因此將其原始碼加以改良後再加以使用。

圖10.16爲將表10.14之三種GOP構造所得到的編碼資料量與畫質的關係，圖中記爲JPEG之曲線，爲將編碼目標之150個畫格以JPEG進行內部編碼之結果。請確認順向MC與進一步導入雙向MC之效果，以及整數像素精度MC與半像素精度MC之效果比較。

表 10.14　對圖 2.2(a)進行 MPEG-1 編碼之各項條件

GOP 構造	(1) IBBPBBPBBPBB ($N=12$，$M=3$) (2)，(3) IPPPPPPPPPPP ($N=12$，$M=1$)
切片組成	影像整體
量子化表	預設值
位元流量控制	固定位元流量控制
MC 之精度	(1)、(2)半像素精度(3)整數像素精度

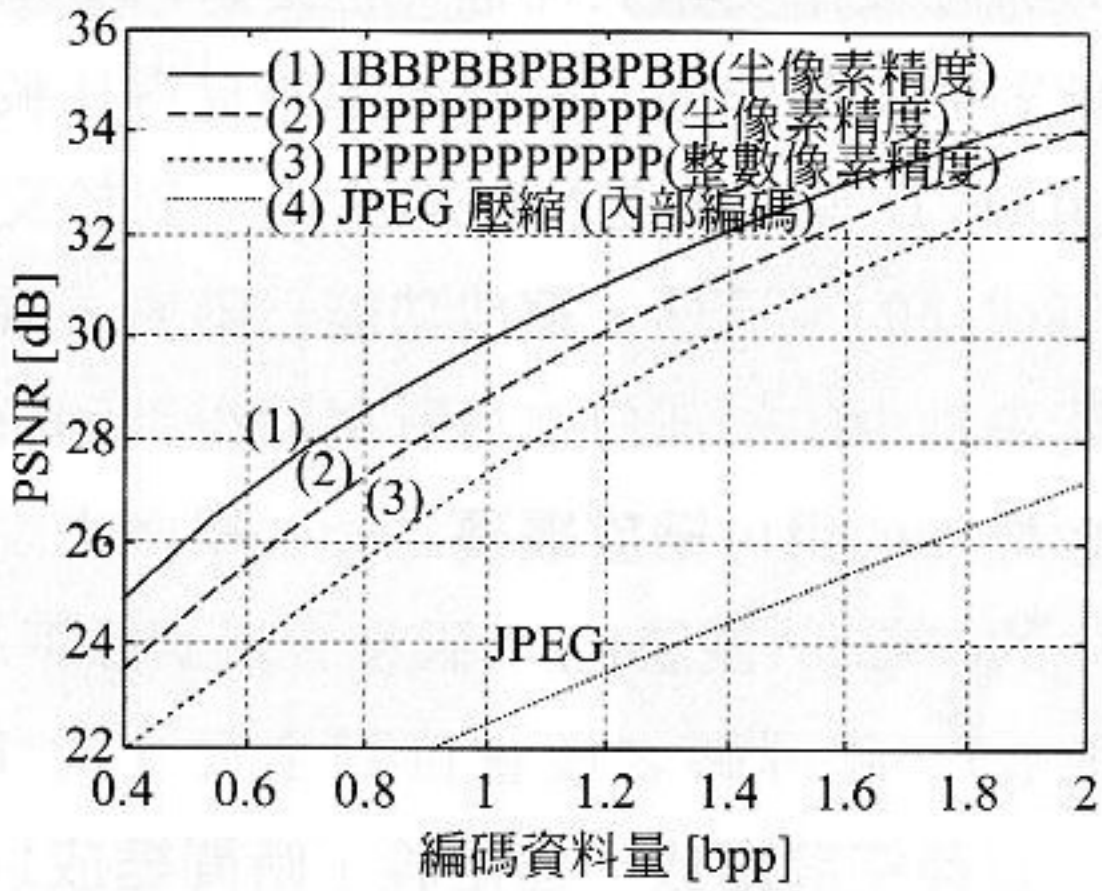


圖 10.16　MPEG-1 之編碼資料量與畫質的關係

MPEG-2 簡述

關於 MPEG-2

相對於MPEG-1 之主要目標爲將資料流量控制在 1.5Mbps，以利儲存於媒體中，MPEG-2 爲可應用於以數位衛星廣播、數位影像光碟(DVD)

等爲代表之廣播、通訊儲存媒體等廣泛領域的標準。因此，使用的視訊格式包括了由 SIF 至 HDTV 之各種不同形式目標，資料流量也達數十 Mbps，較 MPEG-1 高得多。但是，MPEG-2 也是與 MPEG-1 相同，其基礎係基於 MC 與 DCT 的整合編碼方式，實際上的演算法與 MPEG-1 **幾乎相同**。

MPEG-1 與 MPEG-2 最大的差別在於：

① 對應**交錯式掃描**。

② 導入**縮放功能**。

在播放者，一直以來都使用交錯式掃描，但是MPEG基本上以漸次掃描爲處理對象，因此對於交錯影像的編碼效率不見得比較高。因此，在涵蓋播放者或是使用 DVD 等媒體之 MPEG-2 中，對於交錯掃描影像進行有效的編碼是不可或缺的。另外，**縮放功能**爲階層式編碼方法之一種，表示可以僅對位元串當中的一部份進行解碼而得到解析度較原影像低或是低品質的解碼影像。使用可**縮放編碼**之後，舉例來說，應用程式可以做到：位元串當中的一部份(**基層**)在解碼後會得到標準的TV尺寸影像，剩下的資料(擴張層)一起解碼之後會得到全尺寸的 HDTV 影像。在 MPEG-2 中，除了這種**空間縮放**，還定義了**時間縮放**以及 **SNR 縮放**，或是其組合。

在本章的後半，以交錯式影像之處理方法爲中心，簡介MPEG-2 之編碼演算法。

交錯影像與動態補償預測

如圖 7.3 所示，在交錯影像中，一幅**畫格**由於取樣時間不同的兩塊**區域**所組成。因此，觀察在動畫領域爲畫格係的圖 10.17(a)、(b)所示爲

兩張影像重合，與漸次掃描相比，上下的掃描線(line)間的相關性顯著地降低。因此，即使對如圖 10.17(a)、(b)之畫格單純地以 16×16MB 加以切割後使用 MC，特別是非等速移動的情形中無法期待可得到足夠的效果。

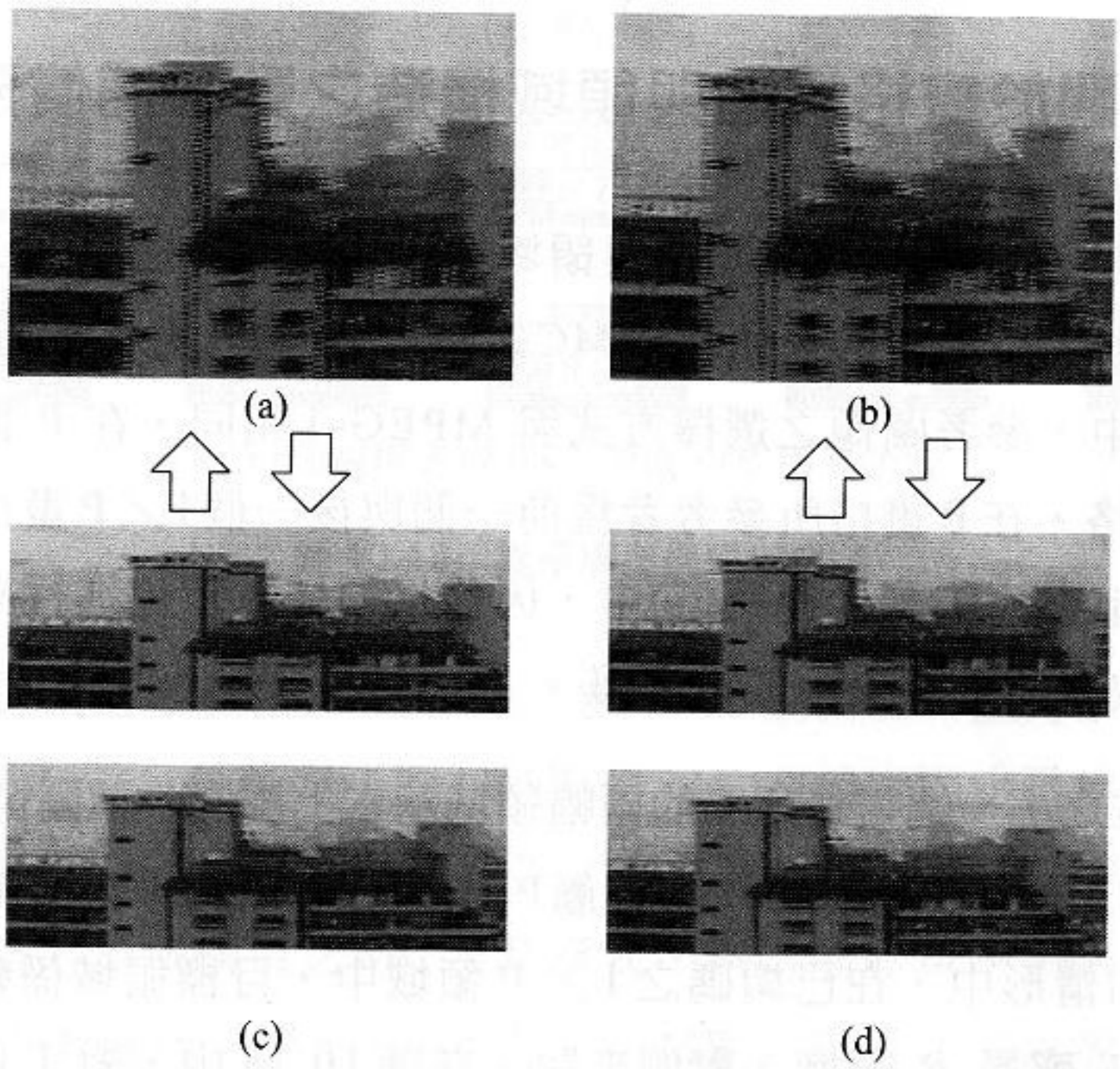


圖 10.17

另外，觀察將單一的畫格如圖 10.17(c)、(d)分解成兩個區域[143]，各自(除了在垂直方向壓扁)與漸次掃描影像無異。因此，相較於 MPEG 中，**將各畫格視為一幅圖像進行編碼**，在MPEG-2 中，爲了要改善交錯影像之編碼率追加了**將各領域視為獨立圖像進行編碼**之模式。也就是說，在MPEG-2 中，關於**圖像之單位選擇**，可選擇合併兩個領域所構成畫格爲IPB圖像之**畫格構造**，或是以各領域構成IPB圖像之**領域構造**[144]。

註 143.在 MPEG-2 中，包含第一條掃描線的領域稱為頂(T)領域。包含第二條水平線之領域稱為底(B)領域。為了減少記錄的長度，此處各自稱為T領域與B領域。

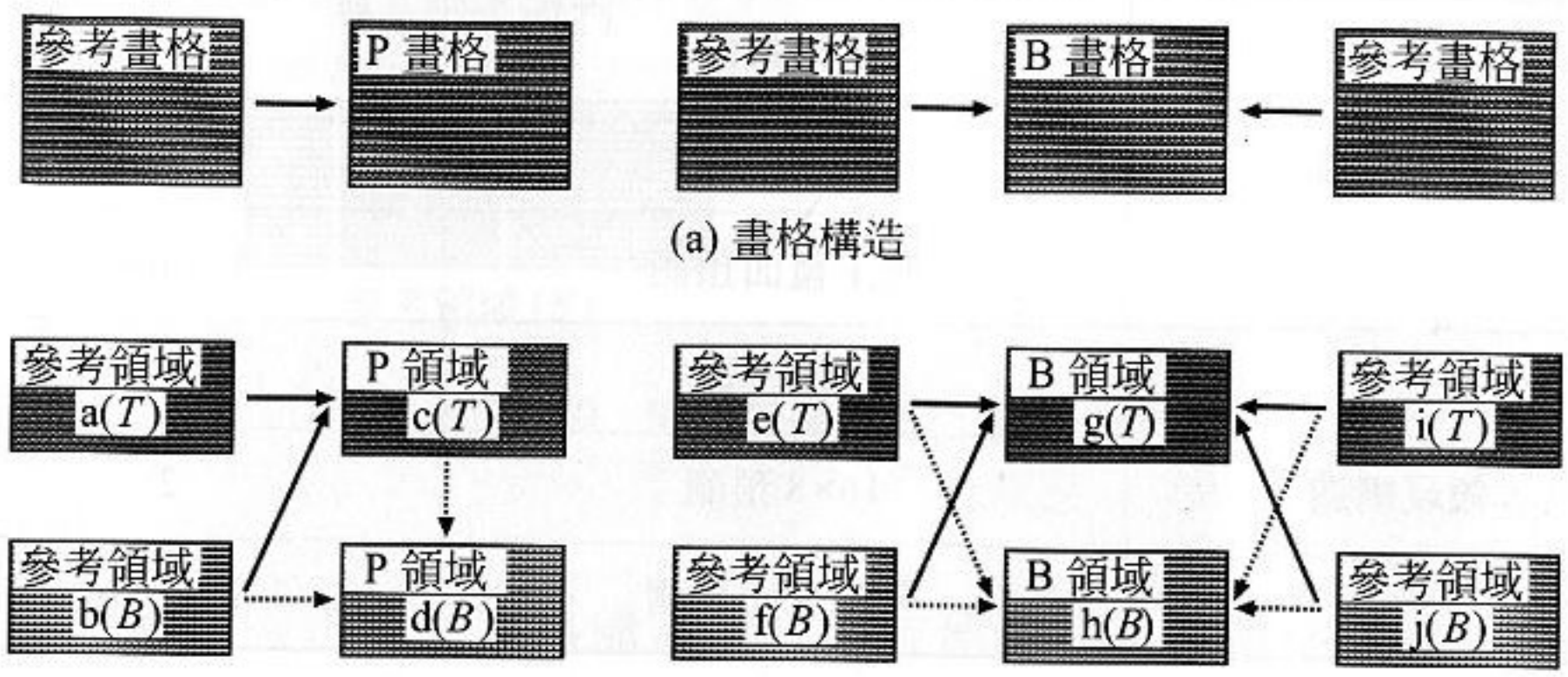
144.如此一來，在畫格構造的情形中，圖像對應至畫格，在領域構造的情形中，對應至領域(配合種類稱為 I、P、B 畫格，或是 I、P、B 領域)。在構造時，處理的圖像數變為兩倍。

關於畫格構造與領域構造之動態補償預測

圖 10.18 所示者為**畫格構造**與**領域構造**中之 MC 適用方法，特別是圖像之參考方法(圖中的箭頭指向MC之參考方向)。畫格構造(**參考的圖像為畫格**)中，參考圖像之選擇方式與 MPEG-1 相同，在 P 畫格為前一個I／P畫格，在B畫格所參考者為前一個與後一個I／P畫格。另外，在領域構造(**參考的圖像為領域**)中，因係以領域為單位進行編碼，參考圖像之選取與前者相比，較為複雜。

具體言之，一幅畫格所屬的兩幅領域基本上屬同一種圖像種類(IPB)[145](在圖 10.18(b)中，領域c與d都屬P圖像，同一圖g與h都屬B圖像)。在 P 領域的情形中，在已編碼之 I／P 領域中，目標領域係**參考時間上最接近之 *T* 或是 *B* 領域**。舉例來說，在圖 10.18 中，對 T 領域 c 進行 MC 的情況中，係參考前一個 I／P 畫格所屬之*T*領域 a 或是 B 領域 b，可以自任何一方取得對應區塊。在對*B*領域 d 進行 MC 的情況中亦適用相同規定，參考之 T 領域為與本身重疊之 c，因此使用參考 B 領域之前一個 I／P 影像中之 B 領域，對其中任何一者進行 MC。另外，在領域構造對 B 圖像進行 MC 之過程中，可參考前後之已完成編碼之 I／P 畫格之*T*／*F*領域(*T*領域 g 與 B 領域 f 相同，可參考 e、f、i、j)在 b 圖像中，與 MPEG-1 相同，可選擇順向／逆向／內插 MC，在內插 MC 的情形中舉例來說參考領域可為 e 與 j 之組合。

註　145.對 I 圖像編碼時為僅有的例外，容許以F領域為 I 圖像，與其對應之B領域為 P 圖像(參考重疊之 T 領域)加以編碼。



(a) 畫格構造

(b) 領域構造 (T)爲 T 領域，(B)爲 B 領域

圖 10.18　畫格構造與領域構造

作爲對上述之畫格／領域構造進行MC預測之具體方法，共規定了五種方法(模式)。這些方法的共通點是 MC 的精度固定爲半像素精度，而不是像MPEG-1 可選擇整數像素／半像素精度。表 10.15 歸納了可適用於各畫格／領域構造之MC預測模式，與附屬於單一 16×16MB 之MV的最大數目(P 圖像之情形)。模式可以 MB 爲單位而加以選擇[146]，當然該資訊包含在位元串中(在領域預測中參考領域(不管是選擇T／B)也一樣地寫入位元串)。此處介紹具有代表性之三種模式。

註　146.正確地說，可以在畫格／領域構造的範圍內進行選擇。

第一種模式爲「對畫格構造進行畫格預測」，這種基本上與MPEG-1中使用的MC爲完全相同的模式。如圖 10.19 所示，目標畫格以 16×16MB 一致地進行分解，在參考畫格上搜尋對應區塊。如此一來，對一個MB，P 圖像上最多對應一個MV，在 B 圖像上最多對應兩個MV。

表 10.15　MPEG-2 之 MC 預測模式

構造	預測模式	*MV*數目*
畫格構造	畫格預測	1
	領域預測	2
	雙主畫面預測	1 + *dmv*
領域構造	領域預測	1
	16×8 預測	2
	雙主畫面預測	1 + *dmv*

*在 P 圖像的情形中，單一之 16×16MB 所附屬的*MV*之最大數目。

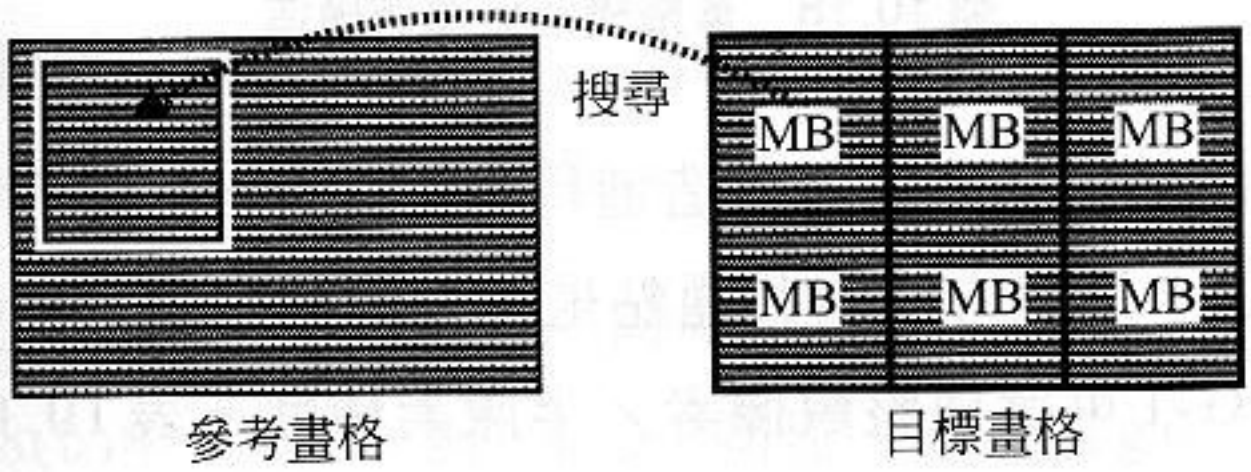


圖 10.19　對畫格構造進行畫格預測

第二種模式爲「對領域構造進行領域預測」，這種基本上相當於將前一種「對畫格構造進行畫格預測」中之處理把「畫格」置換爲「領域」。舉例來說將 T 領域當作 P 圖像進行編碼的情形中，如圖 10.20 所示，對應的領域以 16×16MB 一致地進行分解，在參考的*T*或*B*領域上搜尋對應區塊(參照圖 10.18(b))[147]。單一MB所附屬的*MV*之數目與「對畫格構造進行畫格預測」相同。

註　147.如此一來，在原影像上，與「對畫格構造進行畫格預測」的情形相較，可以看作等價地對在縱向為兩倍大之區塊進行 ME／MC。

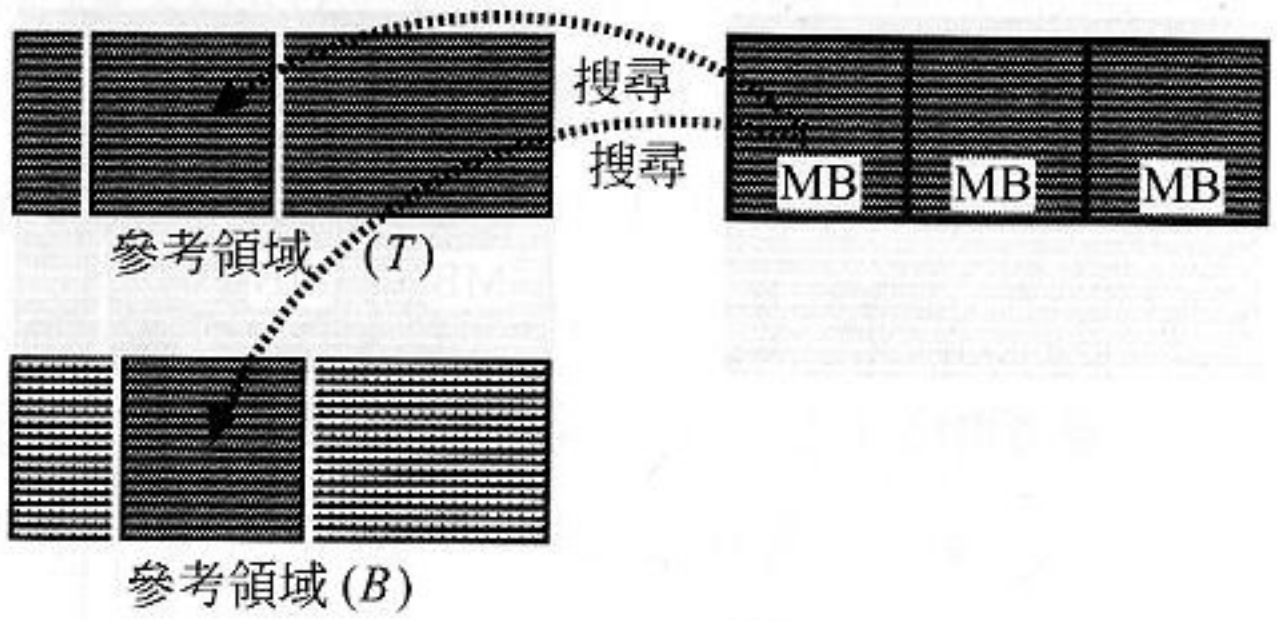


圖 10.20　對領域構造進行領域預測

第三種模式爲「對畫格構造進行領域預測」。在畫格構造中之一幅圖像分割成 MB 後，如圖 10.21 所示，單一 MB 之各掃描線係交互地配屬於T領域與B領域[148]。因此，在本模式中，如圖 10.21 所示，將各MB之掃描線分成T與B各自在參考畫格之T／B領域上搜尋。如此一來，在 P 圖像內之單一 MB 有兩個附屬MV [149]。

註　148.這就是「對畫格構造進行畫格預測」之效率低落的主因。

149.乍見之下，這種模式與「對領域構造進行領域預測」非常接近，但是由於 MB 之等價尺寸有所差異，特別是圖 10.18(b)之領域 d，無法參考與本身相重疊之領域這點為最大不同。

其他的模式尚有「對領域構造進行 16×8 預測」及「對畫格／領域構造進行雙主畫面(DP)預測」(參照表 10.15)。前者爲「對領域構造進行領域預測」中將 MB 之尺寸取爲 16×8 之模式，後者爲在不包含 B 圖像的情形中，在上下方向爲等速移動之假設下，將自兩個領域取得之預測區塊的平均值作爲該MB之預測值之模式。詳情請參照文獻[62]～[65]。

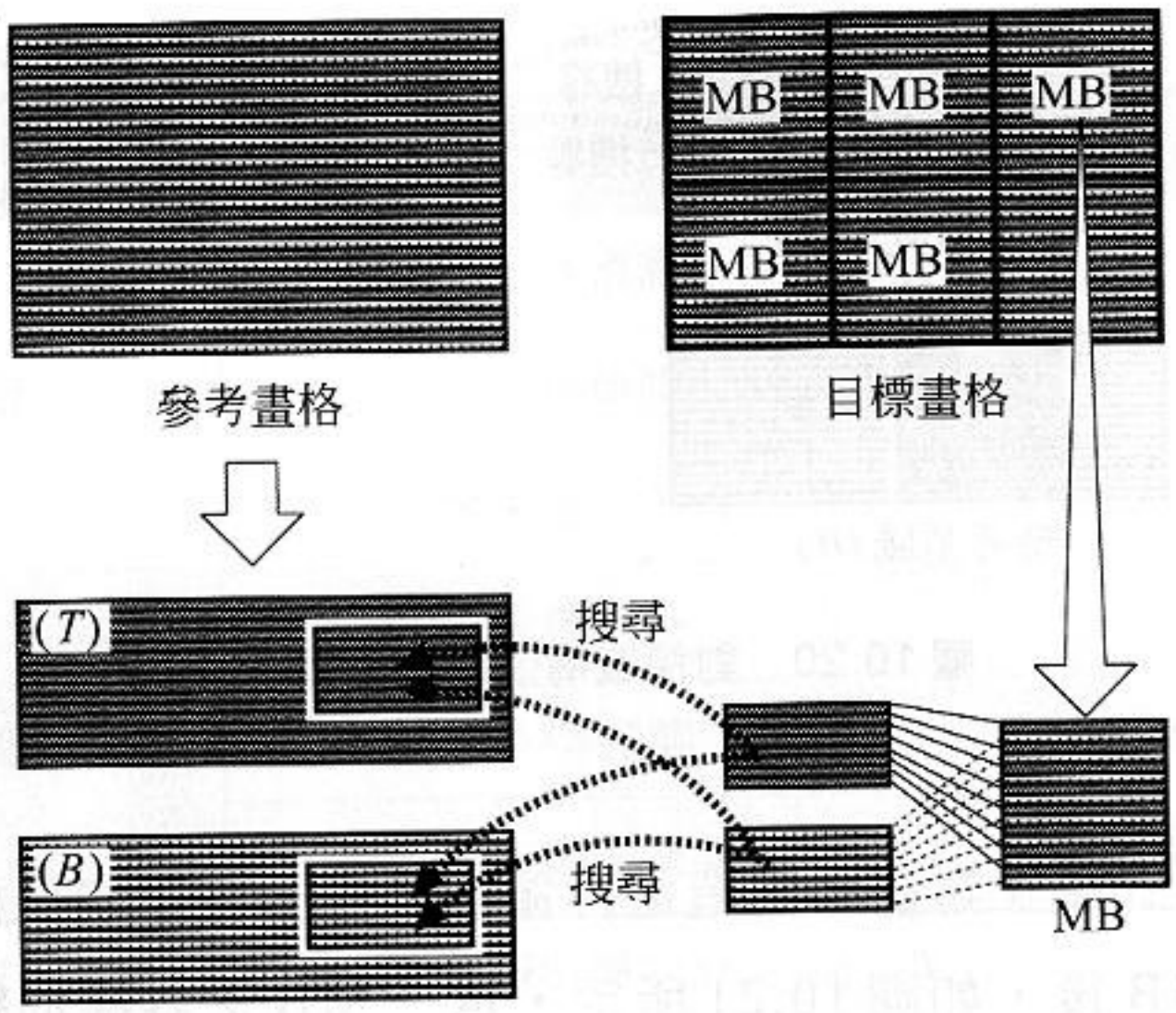


圖 10.21　對畫格構造進行領域預測

由以上所得到的各MV，進行與 MPEG-1 相同之差分編碼，寫入位元串中的MB層。但是，由於與MPEG-1 相較具有複數個預測模式，因此在差分編碼中所使用之預測$MV(pMV)$之求取方式也隨之擴張。另外，預測影像與MPEG-1 相同，在求取目標影像之差分後，求取誤差影像，再對此進行DCT 編碼。

MPEG-2 編碼演算法中的其他特徵

前述之預測影像，依序經DCT、量子化、熵編碼之處理，各別配合交錯的特性進行改良。在此觀察 MPEG-1 與 MPEG-2 主要的不同點。

首先，在MPEG-2 中，在畫格構造中，對輝度Y 分量之MB 可選擇要進行**畫格 DCT** 或是**領域 DCT**。在畫格構造中，不管是畫格預測／領域預測或是內部編碼，構成一個 MB 之掃描線係依T／B領域之各掃描

線交錯配置(圖 10.22(a))。對此，有與 MPEG-1 相同，單純地四等分進行DCT的模式之畫格DCT(同一圖(b))，以及配合各掃描線奇偶數之別，將掃描線再行配置模式之領域 DCT(同一圖(c))。特別是圖 10.17(a)、(b)當領域間之移動量大的情形中，以畫格處理的話，上下掃描線間的相關程度較低，以領域 DCT 較爲有利。畫格／領域 DCT 可依 MB 不同進行選擇。另外，對色差分量一律使用畫格 DCT。

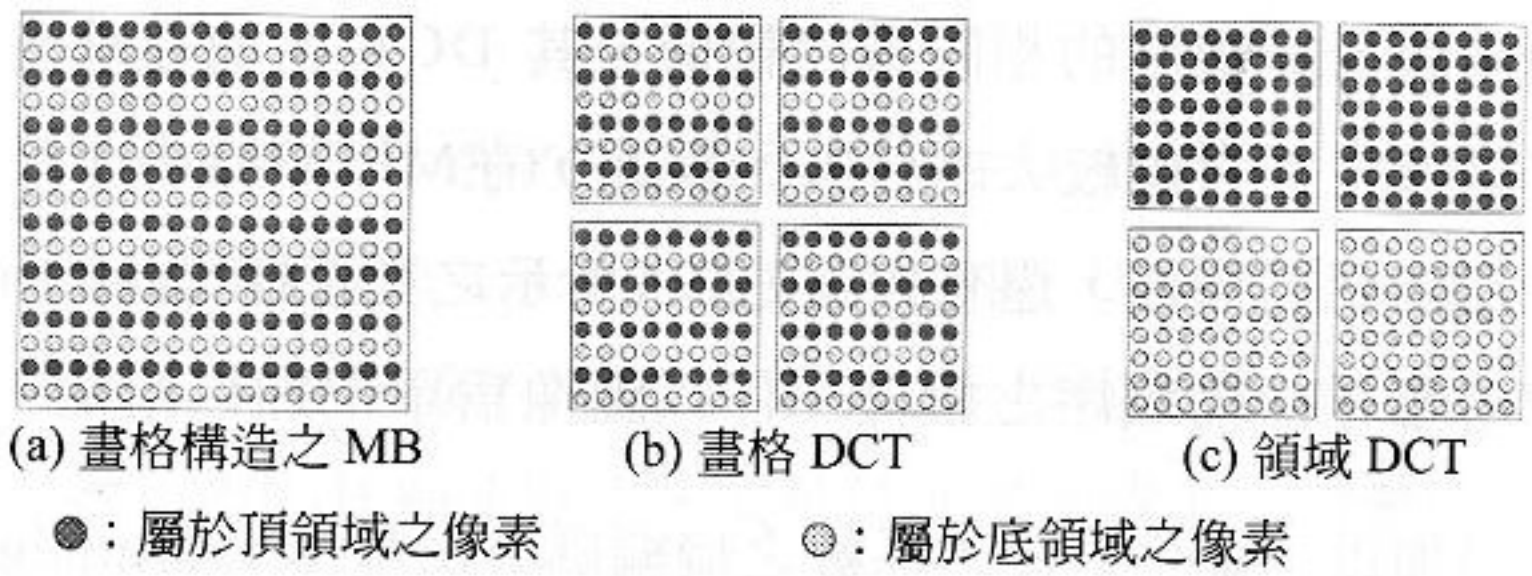


圖 10.22　畫格 DCT 與領域 DCT

對 DCT 係數進行量子化的方法幾乎與 MPEG-1 相同，但是在內部編碼之 DC 係數量子化增加幅度可以選擇爲 8、4、2 這點，以及量子化尺寸q_s爲非線性這點不同。具體而言，內部編碼 DCT 係數以：

$$F'_{\text{Intra}}(0,0)=\text{int}\left(\frac{F_{\text{Intra}}(0,0)}{S}+0.5\cdot\text{sgn}(F_{\text{Intra}}(0,0))\right) \quad (10.13)$$

$$(S=8,4,2)$$

$$F'_{\text{Intra}}(u,v)=\text{int}\left(\frac{16\times F_{\text{Intra}}(u,v)}{q_s\times Q_{\text{Intra}}(u,v)}+0.5\cdot\text{sgn}(F_{\text{Intra}}(u,v))\right) \quad (10.14)$$

非內部編碼 DCT 係數以

$$F'_{\text{Nonintra}}(u,v)=\text{int}\left(\frac{16\times F_{\text{Intra}}(u,v))}{q_s\times Q_{\text{Intra}}(u,v)}\right) \quad (10.15)$$

進行量子化。預設的量子化表，使用式(10.1)與式(10.2)。此處q_s的值，依照位元串中所給的 quantizer_scale_code 決定，可選擇如圖 10.23 所示之直線／非直線的兩種特性。選擇哪一種特性，係以圖像爲單位由旗標加以指示。

量子化之後的 DCT 係數，與 MPEG-1 相同依照掃描順序加以一維化。交錯式影像與漸次掃描影像相較，不管是使用畫格／領域構造在上下方向的兩條掃描線間的相關程度較低，其 DCT 區塊在垂直方向較垂直方向容易產生絕對值較大的係數。因此，在MPEG-2 中，除了之前的圖 6.10 所示之斜向掃描，還有如圖 10.24 所示之交替掃描(alternate scan)順序。使用哪種掃描順序之旗標，係以圖像爲單位加以給定。

以掃描加以一維化之後的係數之熵編碼方法基本上與mpeg-1 相同，但是以對內部編碼 MB 的 AC 係數份追加了新的可變長度對照表爲首，施以數個小改良。

non-uniform quantization

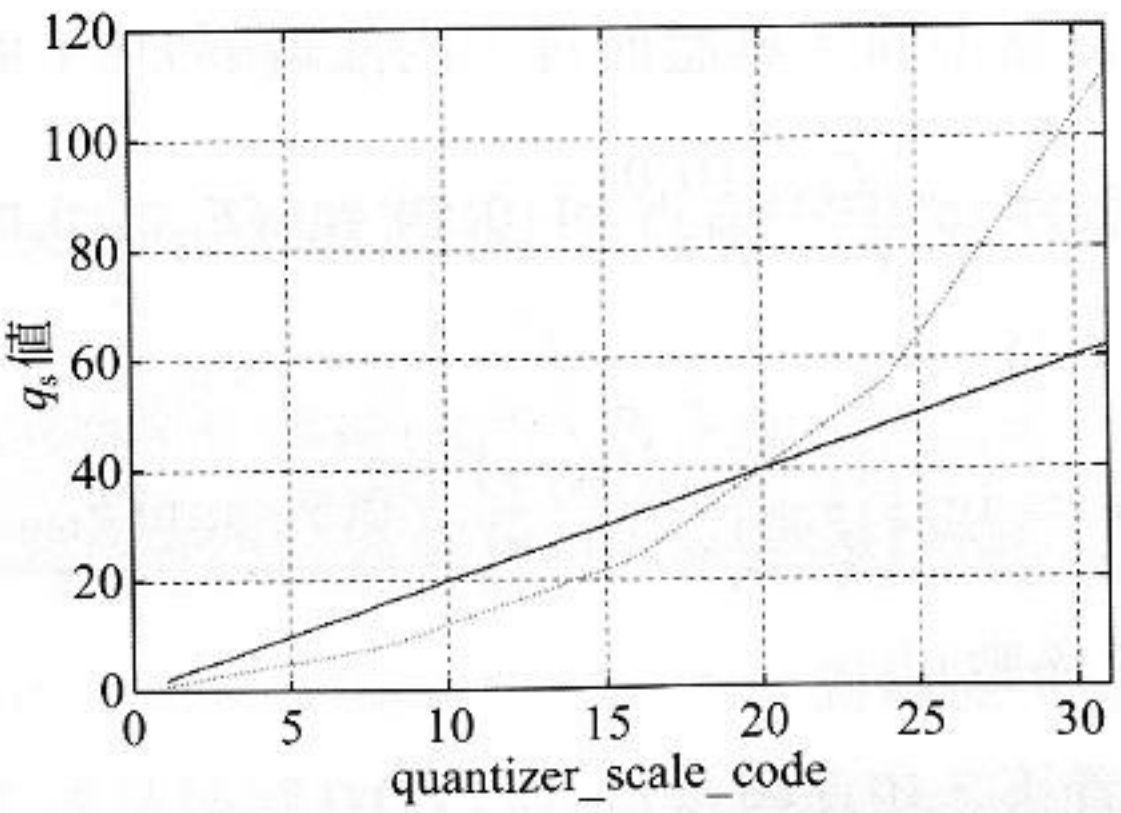


圖 10.23　q_s之直線與非直線特性

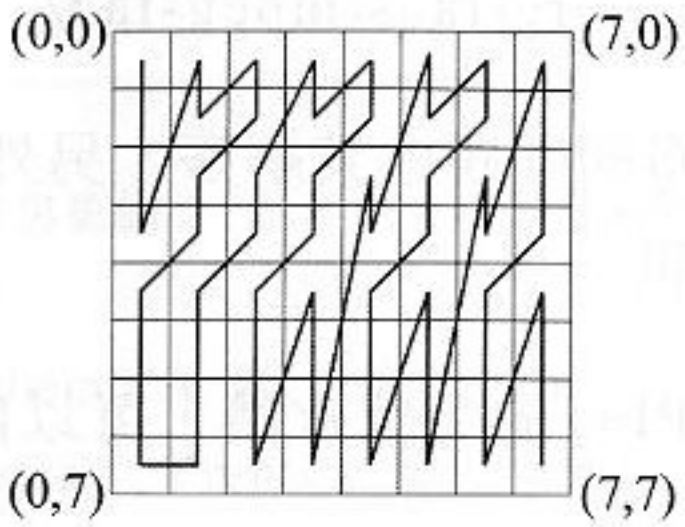


圖 10.24　交替掃描順序

其他，在 MPEG 中的YC_bC_r格式僅容許 4:2:0，在 MPEG-2 中可進一步使用使用 4:4:2 或是 4:4:4 的格式。另外，切片不能如圖 10.12(c)所示有 MB 橫跨水平掃描線的情形，切片一定得從影像左端的 MB 起始。MPEG-2 更進一步的詳細演算法，或是位元串組成等，請參照文獻[62]～[65]。

結　語

在本章中，以MPEG-1 方式爲中心，介紹基本編碼之演算法與位元之大致組成。但是因爲受限於篇幅，僅簡介MPEG-2。但是，要想完全了解這些規格，還是有許多不足之處。舉例來說，使用虛擬緩衝區之資料流率控制方法，或是在MPEG-2 的縮放功能，以及關於輪廓與等級的規定都不得不加以割愛。關於這些在必要時請參照參考文獻。

將本章的內容當作預備知識，即使是難懂的 MPEG 規格書(這樣講好像沒禮貌！)，也應該在攻讀時不會感到那麼困難。

最後，與 JPEG 那章一樣，介紹兩個在網際網路上有用的網站。其中之一爲關於 MPEG 之 FAQ 網站：

http://www.faqs.org/faqs/mpeg-faq/

在此列出許多關於MPEG的問題與其解答。另外，經由其連結，可以連到其他網站之有用的資訊。

在本章所使用的MPEG的免費軟體，可以自MSSG的網站取得：

http://www.mpeg.org/MPEG/MSSG/

mpeg2vidcodec_v12tar.gz爲其原始碼，通常可以用C語言編譯器簡單地加以編譯，俾供參考。

参考文献

[1] 辻井重男監修："ディジタル信号処理の基礎"，電子情報通信学会 (1988)

[2] Oppenheim, A. V. and Schafer, R. W.: "Discrete-time Signal Processing", Prentice-Hall (1989)

[3] Rosenfeld, A. and Kak, A. C.: "Digital Picture Processing", Academic Press Inc. (1976) (長尾真監訳："ディジタル画像処理"，近代科学社 (1978))

[4] 竹村裕夫："CCD カメラ技術入門"，コロナ社 (1997)

[5] 藤子不二雄："ドラえもん 10 巻―夜を売ります―"，pp. 63-66，てんとう虫コミックス (1976)

[6] 池田光男："色彩工学の基礎"，朝倉書店 (1980)

[7] 佐々木玲一，宮崎栄一："テレビジョン"，テレビジョン学会編，コロナ社 (1989)

[8] 伊藤安雄："カラーテレビと色再現"，照明学会雑誌，Vol. 62，No. 12，pp. 33-40 (1978)

[9] 日下秀夫監修："カラー画像工学"，先端技術の手ほどきシリーズ，映像情報メディア学会編，オーム社 (1997)

[10] Poynton, C.: "Frequently Asked Questions about Gamma", http://Home.InfoRamp.Net/poynton/GammaFAQ.html

[11] 坂田晴夫，磯野春雄："視覚における色度の空間周波数特性（色差弁別閾)"，テレビジョン学会誌，Vol. 31，No. 1，pp. 29-35 (1977)

[12] 原島博監修："画像情報圧縮"，先端技術の手ほどきシリーズ，映像情報メディア学会編，オーム社 (1991)

[13] Poynton, C.: "Frequently Asked Questions about COLOR", http://Home.InfoRamp.Net/poynton/ColorFAQ.html

[14] Recommendation ITU-R BT. 601, "Studio Encoding Parameters of Digital Television for Standard 4 : 3 and Wide-screen 16 : 9 Aspect Ratios"

[15] Winkler, S.: "Issues in Vision Modeling for Perceptual Video Quality", Signal Processing, Vol. 78, pp. 231-252 (1999)

[16] Recommendataion ITU-R BT. 500, "Methodology for the Subjective Assessment of the Quality of Television Pictures"

[17] Berger, T.: "Rate Distortion Theory", Prentice-Hall Electrical Engineering Series, Prentice-Hall (1971)

[18] 有本卓："現代情報理論"，電子情報通信学会 (1978)

[19] Peebles Jr., P. Z.: "Probability, Random Variables, and Random Signal Processing", McGraw-Hill (1980)（平野信夫訳，"電子・通信工学のための確率論序説"，東京電機大学出版局 (1980)）

[20] Benedetto, J. J. and Frazier, M. W. (ed.): "Wavelets: Mathematics and Applications", CRC Press Inc. (1994)（山口昌哉監訳："ウェーブレット理論と応用"，シュプリンガー・フェアラーク東京 (1995)）

[21] Topiwala, P. N.: "Wavelet Image and Video Compression", Kluwer Academic Publishers (1998)

[22] Akansu, A. N. and Medley, M. J. (ed.): "Wavelet, Subband and Block Transforms in Communications and Multimedia", Kluwer Academic Publishers (1999)

[23] 南　敏："情報理論（第2版）"，産業図書 (1993)

[24] Hamming, R. W.: "Coding and Information Theory", Prentice-Hall (1986)

[25] 大石進一："例にもとづく情報理論"，講談社サイエンティフィク (1993)

[26] Huffman, D. A.: "A Method for the Construction of Minimum Redundancy Codes", Proc. IRE, Vol. 40, No. 9, pp. 1098-1101 (1952)

[27] Salomon, D.: "Data Compression (2nd Edition)", Spriger-Verlag (2000)

[28] 植松友彦："文書データ圧縮アルゴリズム入門"，CQ出版社 (1994)

[29] 安田浩編著："マルチメディア符号化の国際標準"，丸善株式会社 (1991)

[30] Jayant, N. S. and Noll, P.: "Digital Coding of Waveforms",

Prentice-Hall Signal Processing Series, Prentice-Hall (1984)

[31] Gersho, A. and Gray, R. M.: "Vector Quantization and Signal Compression", Kluwer Academic Publishers (1992)

[32] Wintz, P. A.: "Transform Picture Coding", Proc. IEEE, Vol. 60, No. 7, pp. 809-820 (1972)

[33] Ahmed, N., Natarajan, T. and Rao, K. R.: "Discrete Cosine Transform", IEEE Trans. Comput., Vol. C-23, No. 1, pp. 90-93 (1974)

[34] Strang, G.: "Linear Algebra and its Appications", Academic Press (1976) (山口昌哉監訳："線形代数とその応用", 産業図書 (1978))

[35] Huang, J. J. Y. and Schultheiss, P. M.: "Block Quantization of Correlated Gaussian Random Variables", IEEE Trans. Communications Systems, pp. 289-296 (1963)

[36] 嘉納秀明："システムの最適理論と最適化", コロナ社 (1987)

[37] Papoulis, A.: "Probability, Random Variables and Stochastic Processes", McGraw-Hill, Inc. (1984) (中山謙二, 根本幾, 町田東一共訳, "確率とランダム変数", パポリズ応用確率論 1, 東海大学出版会 (1992))

[38] 西垣久實, 洲之内治男："改訂マトリクスとその応用", コロナ社 (1973)

[39] 笠原晧司："線形代数と固有値問題", 現代数学社 (1972)

[40] Ray, W. D. and Driver, R. M.: "Further Decomposition of the Karhunen-Loeve Series Representation of a Stationary Random Process", IEEE Trans. Information Theory, Vol. IT-16, No. 11, pp. 663-668 (1970)

[41] Rao, K. R. and Yip, P.: "Discrete Cosine Transform Algorithm, Advantage, Applications", Academic Press Inc (1990) (安田浩, 藤原洋共訳："画像符号化技術—DCT とその国際標準—", オーム社 (1992))

[42] Chen, W. H., Smith, C. H. and Fralick, S. C.: "A Fast Computational Algorithm for the Discrete Cosine Transform", IEEE Trans. Communications, Vol. COM-25, No. 9, pp. 1004-

1009 (1977)

[43] 吹抜敬彦："TV 画像の多次元信号処理", 日刊工業新聞社 (1988)

[44] Natrajan, T. R. and Ahmed, N.: "On Interframe Transform Coding", IEEE Trans. Communications, Vol. COM-25, No. 11, pp. 1323-1329 (1977)

[45] Westwater, R. and Furht, B.: "Real-time Video Compression, Techniques and Algorithm", Kluwer Academic Publications (1997)

[46] Koga, J., Iinuma, K., Hirano, A., Iijima, Y. and Ishiguro, T.: "Motion Compensated Interframe Coding for Video Conferencing", Proc. National Telecommunications Conference, pp. G 5.3.1-5.3.5 (1981)

[47] Furht, B., Greenberg, J. and Westwater, R.: "Motion Estimation Algorithm for Video Compression", Kluwer Academic Publications (1997)

[48] Davisson, L. D.: "Rate-distortion Theory and Application", Proc. IEEE, Vol. 60, No. 7, pp. 800-808 (1972)

[49] Girod, B.: "The Efficiency of Motion-compensating Prediction for Hybrid Coding of Video Sequence", IEEE Journal of Selected Areas in Communications, Vol. SAC-5, No. 7, pp. 1140-1151 (1987)

[50] Girod, B.: "Motion-compensating Prediction with Fractional-pel Accuracy", IEEE Trans. Comnunications, Vol. 41, No. 4, pp. 604-611 (1993)

[51] Ortega, A. and Ramchandran, K.: "Rate-distortion methods for Image and Video Comperession", IEEE Signal Precessing Magazine, pp. 23-51 (1998)

[52] Schuster, G. M. and Katsaggelos, A. K.: "Rate-distortion Based Video Compression", Kluwer Academic Publishers (1997)

[53] ISO 10918-1: "Digital Compression and Coding of Continuous-tone still images" (1991)

[54] Hamilton, E.: "JPEG File Interchange Format", C-Cube Microsystems

[55] 藤原洋："ポイント図解式最新 MPEG 教科書"，株式会社アスキー (1994)

[56] 小野定康，鈴木純司："わかりやすい JPEG/MPEG 2 の実現法"，オーム社 (1995)

[57] http://www.ijg.org/

[58] ISO/IEC 11172-2, "Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/s" (1993)

[59] ISO/IEC 13818-2, "Generic Coding of Moving Pictures and Associated Audio Information" (1996)

[60] Mitchell, J. L., Pennebaker, W. B., Fogg, C. E. and LeGall, D. J.: "MPEG Video Compression Standard", Digital Multimedia Standards Series, Chapman & Hall (1996)

[61] http://www.mpeg.org/MPEG/MSSG/

[62] Haskell, B. G., Puri, A. and Netrvali, A. N.: "Digital Video: An Introduction to MPEG-2", Digital Multimedia Standards Series, Chapman & Hall (1997)

[63] 安田浩編著："MPEG/マルチメディア符号化の国際標準"，丸善株式会社 (1994)

[64] 小野文孝，渡辺裕："国際標準画像符号化の基礎技術"，高度映像技術シリーズ 1，コロナ社 (1998)

[65] テレビジョン学会編："MPEG"，総合マルチメディア選書，オーム社 (1996)

[66] Chen, C. F., Pang, K. K.: "The Optimal Transform of Motion-Compensated Frame Difference Images in a Hybrid Coder", IEEE Trans. Circuits & Systems II, Vol. 40, No. 6, pp. 393-397 (1993)

索　引

數字、字母順序

4:2:0 格式 1-20
4:2:2 格式 1-20
4:4:4 格式 1-20
A/D 轉換器 1-11
AC 係數 6-2
AC 基底 6-2
B 圖像 10-5
CCD 1-7
CIF 1-22
CRT 1-15
DCT(離散餘弦轉換) 5-33
DCT 係數之熵編碼 6-12
DCT 係數量子化 6-9
DC 係數 6-2
DC 基底 6-2
DHT 9-11
DQT 9-11
EOB 6-14,9-9
EOI 9-11
ESC 碼 2-12
Extended 系統 9-1
Full-pel ME 7-17
f 碼 10-20
GOP(Group Of Pictures) 10-7
GOP 層 10-31
H.261 10-2
halfl-pel ME 7-17
Huffman 表 3-13
Huffman 編碼 3-12
Huffman 樹 3-13
Independent Function 9-2
Interframe coding 7-4
Interlaced scanning 7-1
Intraframe coding 7-4
I 圖像 10-6
JFIF 9-13
JPEG 9-1
Karhunen-Loeve 轉換 5-1,5-16
KLT(Karhunen-Loeve Transform) 5-1,5-16
Kraft-McMillan 不等式 3-8
Kronecker 積 5-22
Lagrange 乘子法 5-14
Lossless 1-23
lossless 編碼 2-3

lossy 1-23
lossy 編碼 2-3
L-R 形算術編碼 3-18
MB(Macroblock) 7-10
MB_ESC 10-22
MB_STUFFING 10-24
MBAI(Macro Block Address Increment) 10-24
MB 型式(MBT) 10-22
MB 層 10-22,10-30
MC(Motion Compensation) 7-8
MCU(Minimum Coded Unit) 9-12
ME(Motion Estimation) 7-8
Motion compensation predition 7-8
MPEG 10-1
MPEG-2 10-39
MSE(Mean Square Error) 1-25,2-4
MV(Motion Vector) 7-9
NTSC 1-14
PAL 1-14
Parseval 性質 5-11
pel(picture element) 1-7
pel 1-7
pixel 1-7
progressive scanning 7-1
PSD(Power Spectral Density Function) 8-3
PSNR(Peak Signal to Noise Ratio) 2-5
P 圖像 10-6
QCIF 1-22
QM 碼 3-19
QSIF 1-22
Q 表 6-9
reference frame 7-12
run length 6-14
run 6-14
SECAM 1-14
SIF 1-22
SNR(Signal to Noise Ratio) 1-25
SNR 可縮放性 10-40
SOF 9-11
SOI 9-11
SOS 9-11
Spatial 方式 9-1
target frame 7-12
YC_bC_r 1-19
YC_bC_r 空間 1-19
ZRL 9-9
γ 修正 1-15

一劃

一次轉換 5-1
一致量子化 4-3
一致解碼性 3-7

二劃

二進位編碼 3-2

三劃

三次搜尋法 8-16
三原色刺激值 1-13
三原色原理 1-13
三維正交轉換 7-4
小波轉換 2-23

四劃

不可逆 1-23
不可逆編碼 2-3
中點水平型 4-3
中點垂直型 4-3
內部處理編碼 7-4
分色稜鏡 1-15
分佈圖 2-22
切片層 10-27,10-33
逆向 ME／MC 10-4
反轉換 5-6
巴塞瓦關係 5-11

五劃

比對誤差 7-14
比對誤差曲面 7-14
比對錯誤 10-14
主觀評估 1-25
加法調色 1-12
半像素精度動態推測 7-18
可逆 1-23
可逆編碼 2-3
可縮放編碼 10-39
可變長度編碼 2-21,3-3,3-13
平均平方差 1-25
平均編碼長度 3-3
平移 7-9
未涵蓋領域 7-16
正交 5-4
正交矩陣 5-6
正交轉換 5-4
正定值矩陣 5-30
正規 5-4
正規正交基底 5-4
正轉換 5-6
目標畫格 7-14

六劃

交錯 9-11
交疊 1-5

交疊現象 1-5
光學濾鏡 1-7
向量量子化 4-3
自協方差函數 2-14
色差分量 1-19

七劃

位元比較法 3-15
位元串 2-2
位元組分配 9-11
宏塊 7-10
局部解碼器 7-22
序列層 10-30

八劃

依序掃描 7-1
例外碼 2-12
具有死帶之一致量子化(器) 4-3,10-14
刷新 7-20
協方差 2-14
協方差矩陣 5-19
取樣 1-3
取樣定值 1-3
取樣訊號 1-3
固定資訊來源 3-3
底領域 10-40
延伸 6-14
延伸長度 6-14
波形編碼 2-2
空間取樣 1-2
空間縮放性 10-39
表示順序 10-29
附加位元 9-5
非一致量子化 4-3
非交錯式 9-11
不定資訊來源 3-3
非相關化 2-21
非破壞性 1-23
非記憶性資訊來源 3-3
非線性量子化 4-3

九劃

亮度分量 1-18
係數 2-10,5-6
客觀評估 1-25
省略 10-25
相關係數 2-18

十劃

差分編碼 6-13
時間縮放性 10-40
特殊播放功能 10-3
特徵向量 5-18
破壞性 1-23
純量量子化 4-3

蚊狀雜訊 6-17
記憶性資訊來源 3-3
起始碼 10-7,10-29
高速 DCT 6-18
高速搜尋演算法 7-23
高斯雜訊 8-10

十一劃

僞輪廓 1-11
漂移 7-21
動態推測 7-6
動態補償 7-6
動態補償預測 7-6
區域編碼器 7-22
區塊比對 7-11
區塊結束編碼 6-15
區塊層 10-33
區塊雜訊 6-16
參考畫格 7-11
基底向量 1-18
基層 10-39
基線系統 9-1
掃描 9-10
掃描標頭 9-12
斜向掃描 6-14
移動向量 7-8
移動向量搜尋範圍 7-12
符號 3-1
符號之縮減 3-12
粒狀誤差 4-5
粒狀雜音 4-5
頂領域 10-40

十二劃

最佳位元分配 5-12
最佳量子化器 4-5
最佳轉換 5-21
幾何轉換 7-5
減樣 1-20
無效碼 9-11
畫格 7-1
畫格 DCT 10-46
畫格內編碼 7-3
畫格速率 7-1
畫格間編碼 7-3
畫格構造 10-41
畫格標頭 9-12
等色 1-13
等色函數 1-13
絕對值誤差 7-12
量子化 1-2,1-7
量子化尺度 10-14
量子化步進幅度 1-9,4-2
量子化表 6-9
量子化指數 1-10
量子化矩陣 6-9

量子化等級 1-10
量子化雜訊 1-10
階層 10-8
順向 ME／MC 10-4
順向轉換 5-6

十三劃

溢位誤差 4-5
補碼 10-30
資料率－失眞曲線 8-4
資料率－失眞函數 8-2
資料率－失眞理論 2-10,8-2
資訊來源之擴大 3-11
資訊來源 3-1
資訊來源編碼定理 3-10
跳耀掃描 7-1
電能頻譜密度函數 8-3
預設 9-6
預設量子化表 10-14
預測影像 7-6
預測編碼 6-14

十四劃

像素 1-7
像素間相關 2-7
圖像 10-26
圖像層 10-29
對照表參照法 3-13
對稱矩陣 5-18
端點 3-7
語頭 3-7
領域 7-1
領域 DCT 10-45
領域構造 10-41

十五劃

影像 9-8
影像格式 1-17
影像編碼 2-1
影像壓縮編碼 2-1
數位影像 1-15
標頭 10-7
編碼(語言) 3-1
編碼順序 10-30
編碼樹 3-7
線性轉換 5-2
熵 3-9
熵編碼 2-22

十六劃

整合編碼方式 10-1
整數像素精度動態推測 7-15
頻帶限制 1-4
頻帶限制訊號 1-4
壓縮率 2-3

十七劃

檢索表 1-17
瞬間解碼性 3-7
縮放性 10-39
縮減 1-20
算數編碼 3-17
擴張層 10-40

十八劃

轉換 5-6
轉換係數 2-10,5-6
轉換區塊 5-6
離散Δ函數 2-18
離散餘弦轉換 5-33
雙向 ME／MC 10-4
雙向動態保障預測 10-4
雙重一次內插法 7-17
邊緣 2-6
類比影像 1-2

國家圖書館出版品預行編目資料

影像壓縮技術 / 酒井善則，吉田俊之原著 ：白執善編譯. -- 初版. -- 臺北市 ： 全華，民93

面 ； 公分

參考書目：面

含索引

譯自 ： 映像情報符号化

ISBN 957-21-4703-X (平裝)

1. 影像壓縮

448.68 93017768

影像壓縮技術

映像情報符号化

原　　著　酒井善則、吉田俊之

編　　譯　白執善

發 行 人　陳本源

出 版 者　全華科技圖書股份有限公司

地　　址　104 台北市龍江路 76 巷 20 號 2 樓

電　　話　(02) 2507-1300　(總機)

傳　　眞　(02) 2506-2993

郵政帳號　0100836-1 號

印 刷 者　宏懋打字印刷股份有限公司

登 記 證　局版北市業第〇七〇一號

圖書編號　05606

初版一刷　2004 年 10 月

定　　價　新台幣 300 元

I S B N　957-21-4703-x (平裝)

有著作權・侵害必究

版權聲明

Original Japanese edition.
Human Communication Kougaku Series, Eizou Jyouhou Fugouka.
Supervised by Hiroshi Harashima.
Edited by Denshi Jyouhou Tsushin Gakkai.
Writer by Yoshinori Sakai and Toshiyuki Yoshida.
Copyright © 2001 by Shadan Hojin Denshi Jyouhou Tsushin Gakkai.
Published by Ohmsha, Ltd.
This Chinese Language edition co-published by Ohmsha, Ltd. and CHUAN HWA SCIENCE & TECHNOLOGY BOOK CO., LTD.
Copyright © 2004. All rights reserved.

全華科技圖書
www.chwa.com.tw
book@ms1.chwa.com.tw

全華科技網 OpenTech
www.opentech.com.tw