VIII. Data Compression (B)

◉ 8-A Differential Coding for DC Terms, Zigzag for AC Terms

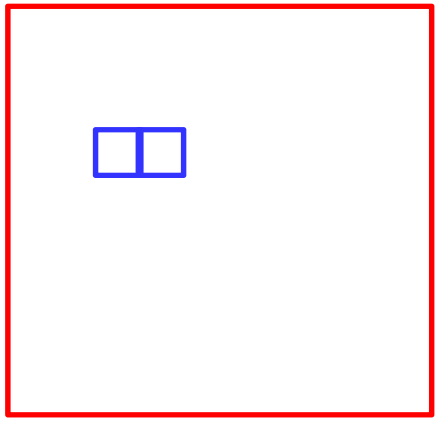
這兩者可視為 JPEG Huffman coding 的前置工作

Differential Coding (差分編碼)

If the DC term of the $(i,\ j)^{th}$ block is denoted by $DC[i, j]$, then

encode $DC[i, j] - DC[i, j\text{-}1]$

Instead of $DC[i, j]$



(也是運用 space domain 上的一致性)

Zigzag scanning

將 2D 的 8x8 DCT outputs 變成 1D 的型態

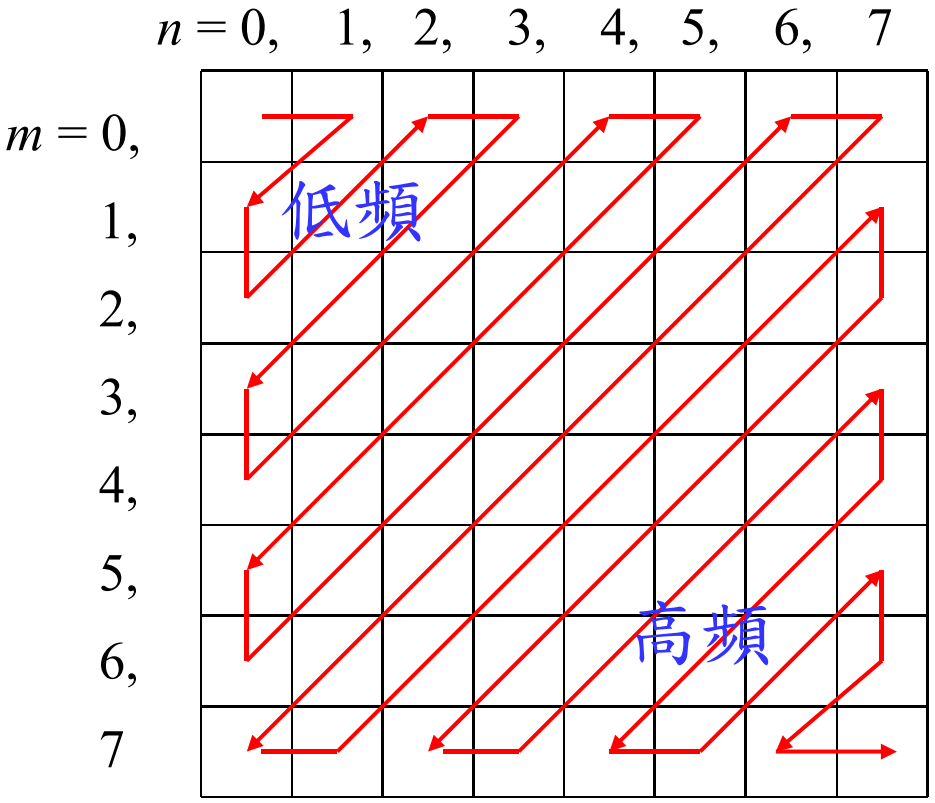
但按照 “zigzag” 的順序 (能量可能較大的在前面)

8x8 DCT output:

$C[m, n]$

$m = 0, 1, \ldots, 7$

$n = 0, 1, \ldots, 7$



(也是運用 frequency domain 上的一致性)

◉ 8-B Lossless Coding

Lossless Coding: The original data can be perfectly recovered

Example:

direct coding method

Huffman coding

Arithmetic coding

Shannon–Fano Coding, Golomb coding, Lempel–Ziv, ...

◉ 8-C Lossless Coding: Huffman Coding

- **Huffman Coding 的編碼原則: (Greedy Algorithm)**

(1) 所有的碼皆在 Coding Tree 的端點，再下去沒有分枝
(滿足一致解碼和瞬間解碼)

(2) 機率越大的， code length 越短；機率越小的， code length 越長

(3)假設 S_a 是第 L 層的 node，S_b是第 L+1 層的 node

則 $P(S_a) \geq P(S_b)$ 必需滿足

不滿足以上的條件則交換

原始的編碼方式：

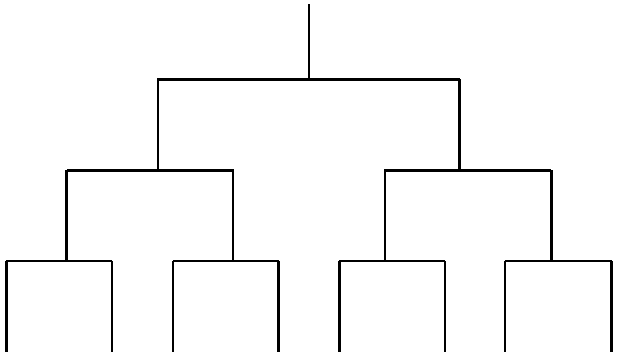
若 data 有 M 個可能的值，使用 k 進位的編碼，
則每一個可能的值使用 floor($\log_k M$) 或 ceil($\log_k M$) 個 bits 來編碼

floor: 無條件捨去

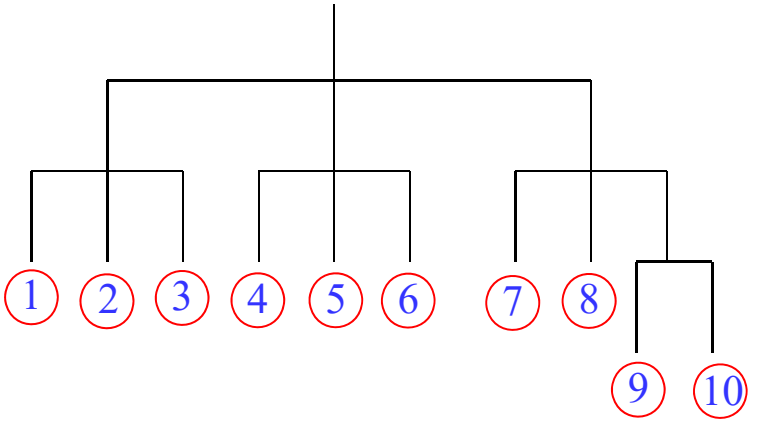
ceil: 無條件進位

Example:

若有 8 個可能的值，在2進位的情形下，需要 3 個 bits



若有 10 個可能的值，在3進位的情形下，需要 2 個或 3 個 bits



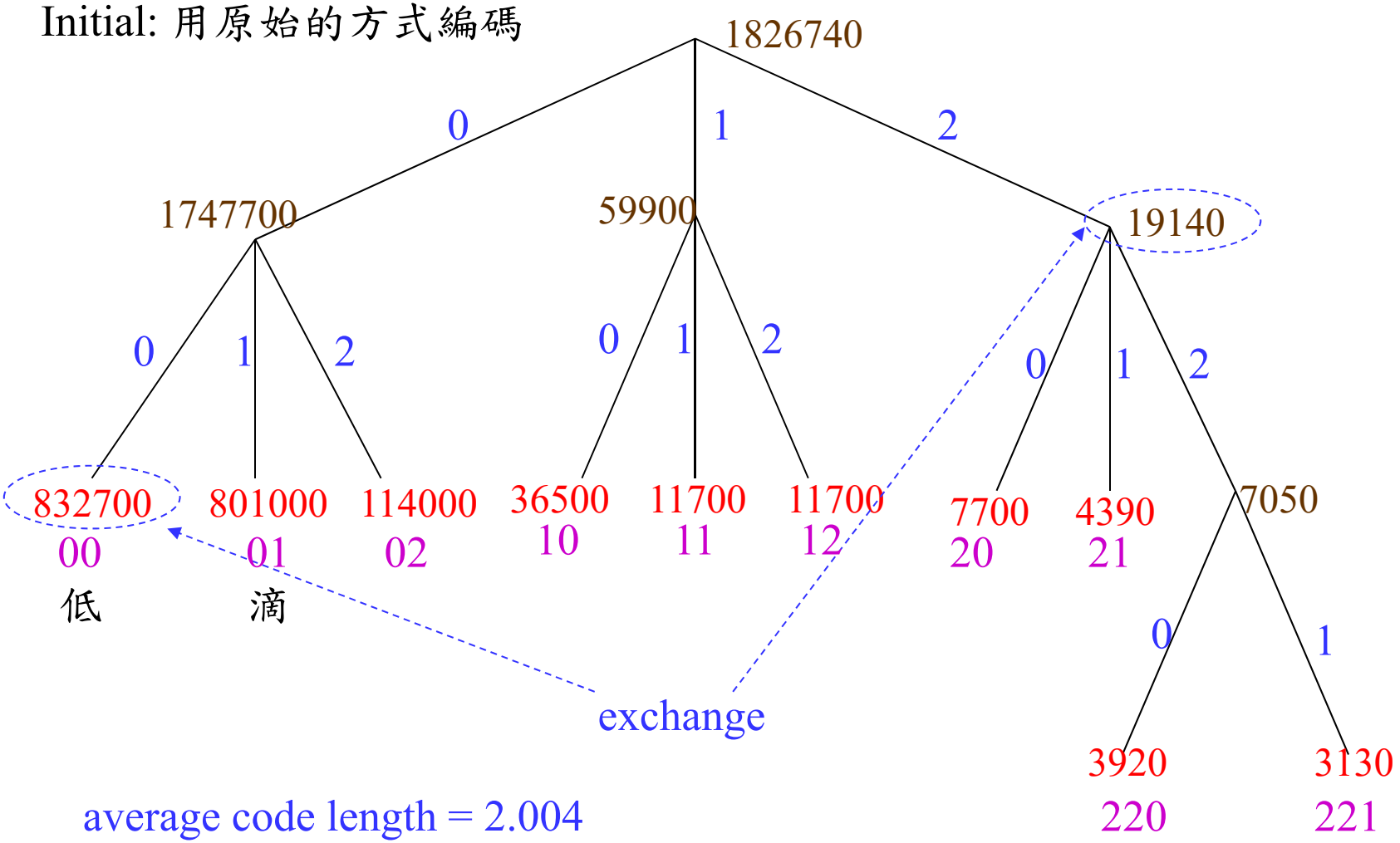
Example:

低	滴	氐	羝	鞮
832700	801000	114000	7700	4390
磾	祗	葧	墑	熵
3920	11700	11700	3130	36500

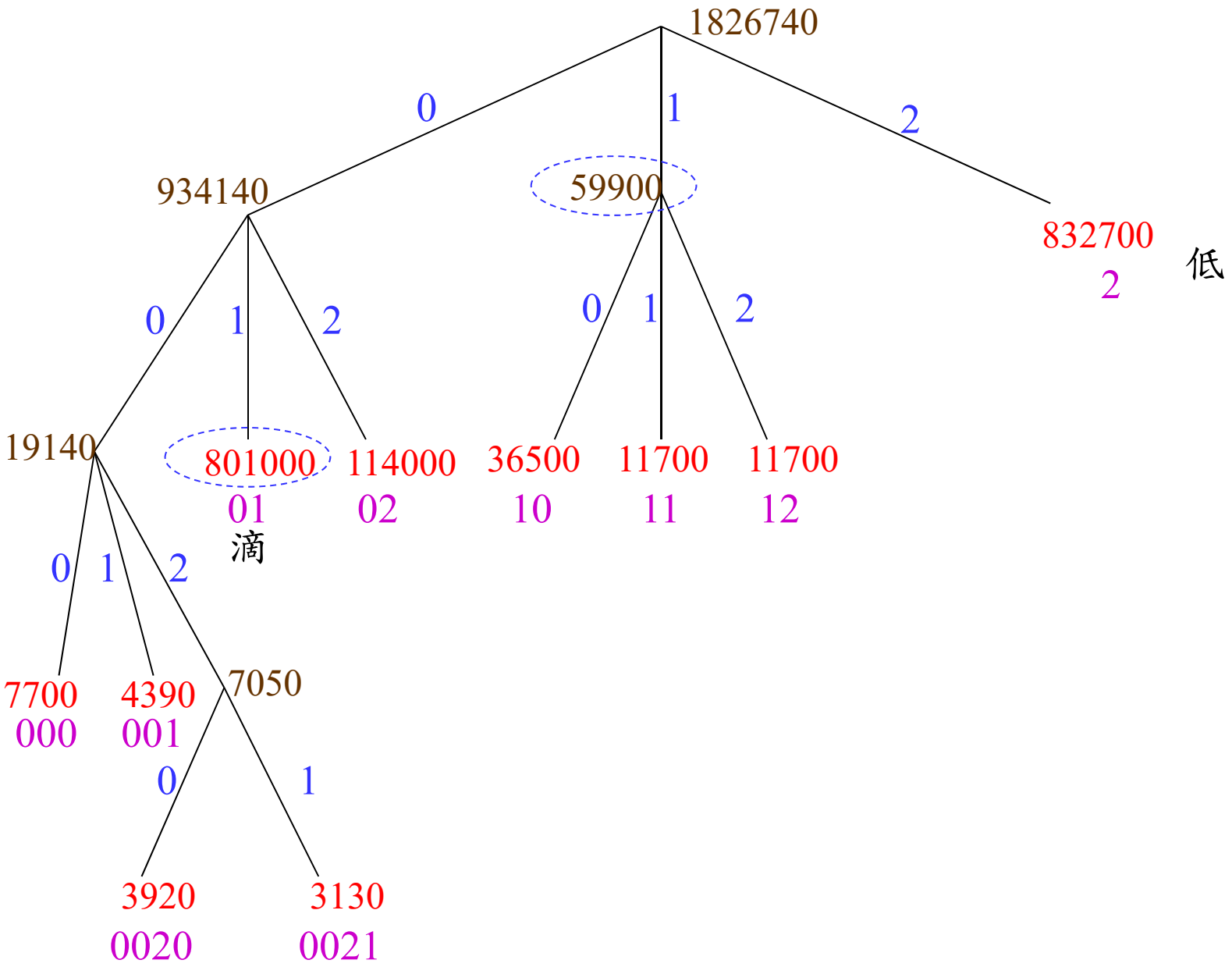
他們 3進位的Huffman Code 該如何編

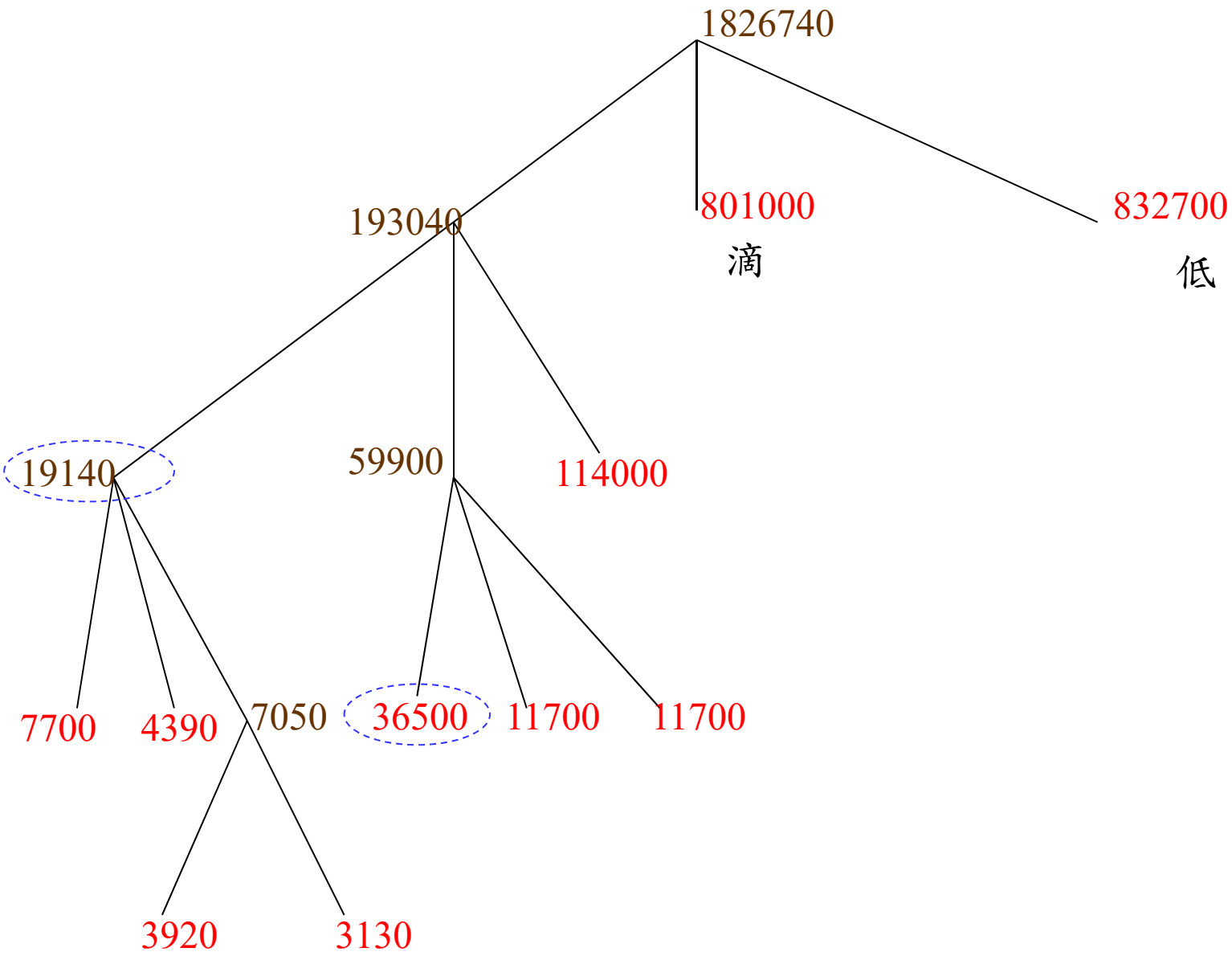
Huffman coding by the greedy algorithm

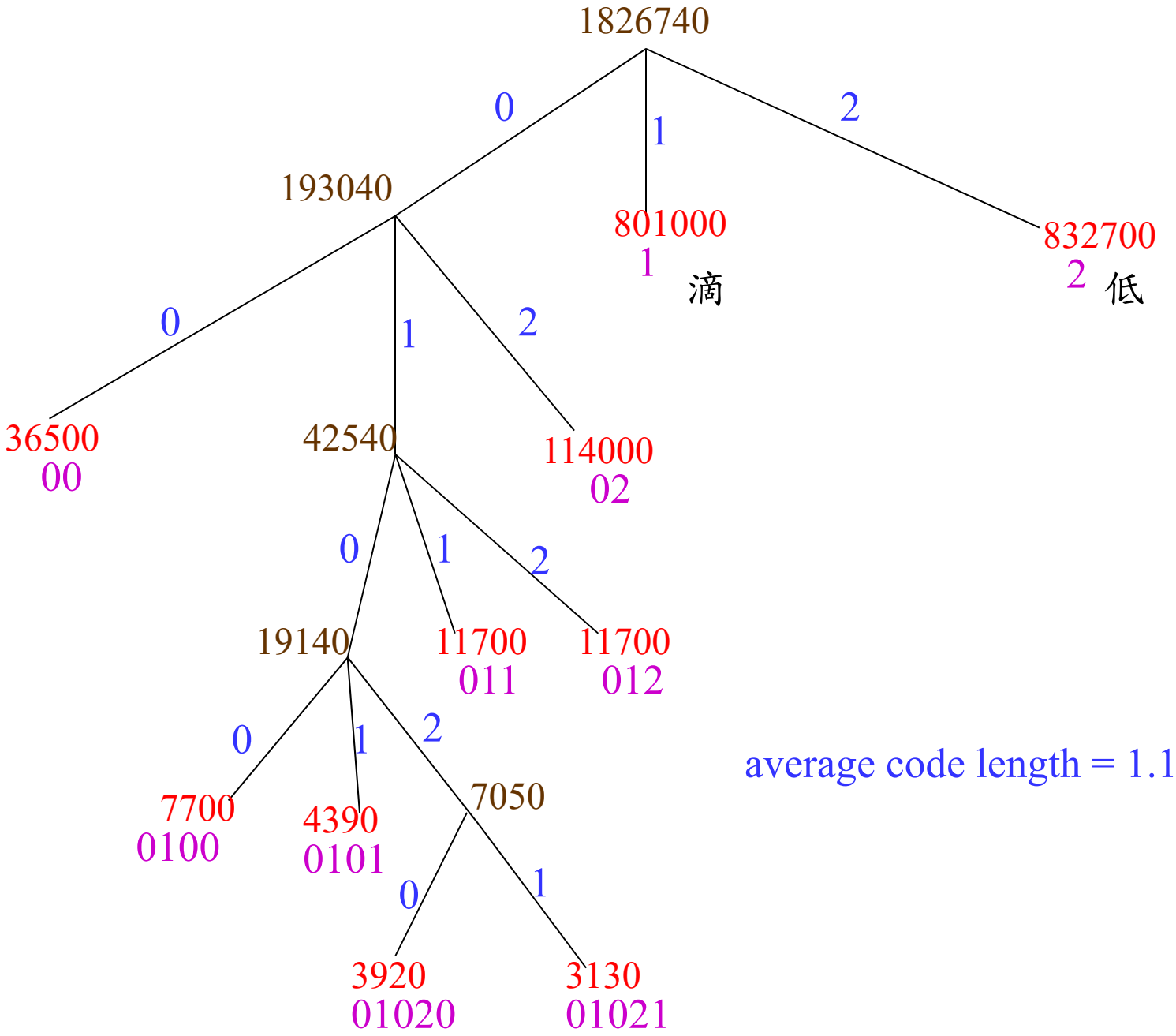
Initial: 用原始的方式編碼



average code length = 2.004


1826740
0
1
2
934140
59900
832700
2
低
0
1
2
0
1
2
19140
801000
01
滴
114000
02
36500
10
11700
11
11700
12
0
1
2
7700
000
4390
001
7050
0
1
3920
0020
3130
0021


1826740
193040
801000
滴
832700
低
19140
59900
114000
7700
4390
7050
36500
11700
11700
3920
3130



average code length = 1.143

思考： 郵遞區號是多少進位的編碼？

電話號碼的區域碼是多少進位的編碼？

中文輸入法是多少進位的編碼？

如何用 Huffman coding 來處理類似問題？

◉ 8-D Entropy and Coding Length

- Entropy 熵；亂度 (Information Theory)

註：此處 log 即 ln
和 $\log_{10}$ 不同

$$entropy = \sum_{j=1}^{J} P(S_j) \log \frac{1}{P(S_j)}$$

P: probability

$P(S_0) = 1$, entropy = 0

$P(S_0) = P(S_1) = 0.5$, entropy = 0.6931

$P(S_0) = P(S_1) = P(S_2) = P(S_3) = P(S_4) = 1/5$, entropy = 1.6094

$P(S_0) = P(S_1) = P(S_2) = P(S_3) = 0.1$, $P(S_4) = 0.6$, entropy = 1.2275

同樣是有 5 種組合，機率分佈越集中，亂度越少

- Huffman Coding 的平均長度

$$mean(L)=\sum_{j=1}^{J}P(S_j)L(S_j)$$ $P(S_j)$: S_j 發生的機率， $L(S_j)$: S_j 的編碼長度

- **Shannon 編碼定理：**

$$\frac{entropy}{\log k}\le mean(L)\le\frac{entropy}{\log k}+1$$ 若使用 k 進位的編碼

- Huffman Coding 的 total coding length $b=mean(L)N$ N: data length

$$N\frac{entropy}{\log k}\le b\le N\frac{entropy}{\log k}+N$$

都和 entropy 有密切關係

Entropy: 估計 coding length 的重要工具

$$N\frac{entropy}{\log k} \cong \text{bit length}$$

◉ 8-E Arithmetic Coding

- **Arithmetic Coding (算術編碼)**

Huffman coding 是將每一筆資料分開編碼

Arithmetic coding 則是將多筆資料一起編碼，因此壓縮效率比 Huffman coding 更高，近年來的資料壓縮技術大多使用 arithmetic coding

K. Sayood, *Introduction to Data Compression*, Chapter 4: Arithmetic coding, 3rd ed., Amsterdam, Elsevier, 2006

編碼

若 data X 有 M 個可能的值 ($X[i] = 1, 2, \ldots,$ or M)，使用 k 進位的編碼，且

P_n : the probability of $x = n$ (from prediction)

$$S_0 = 0, \quad S_n = \sum_{j=1}^{n} P_j$$

現在要對 data X 做編碼，假設 length(X) = N

Algorithm for arithmetic encoding

initiation: $lower = S_{X[1]-1}$ $upper = S_{X[1]}$

for $i = 2 : N$

$$lower = S_{X[i]-1} \times (upper - lower)$$

$$upper = S_{X[i]} \times (upper - lower)$$

end

(continue)…

Suppose that

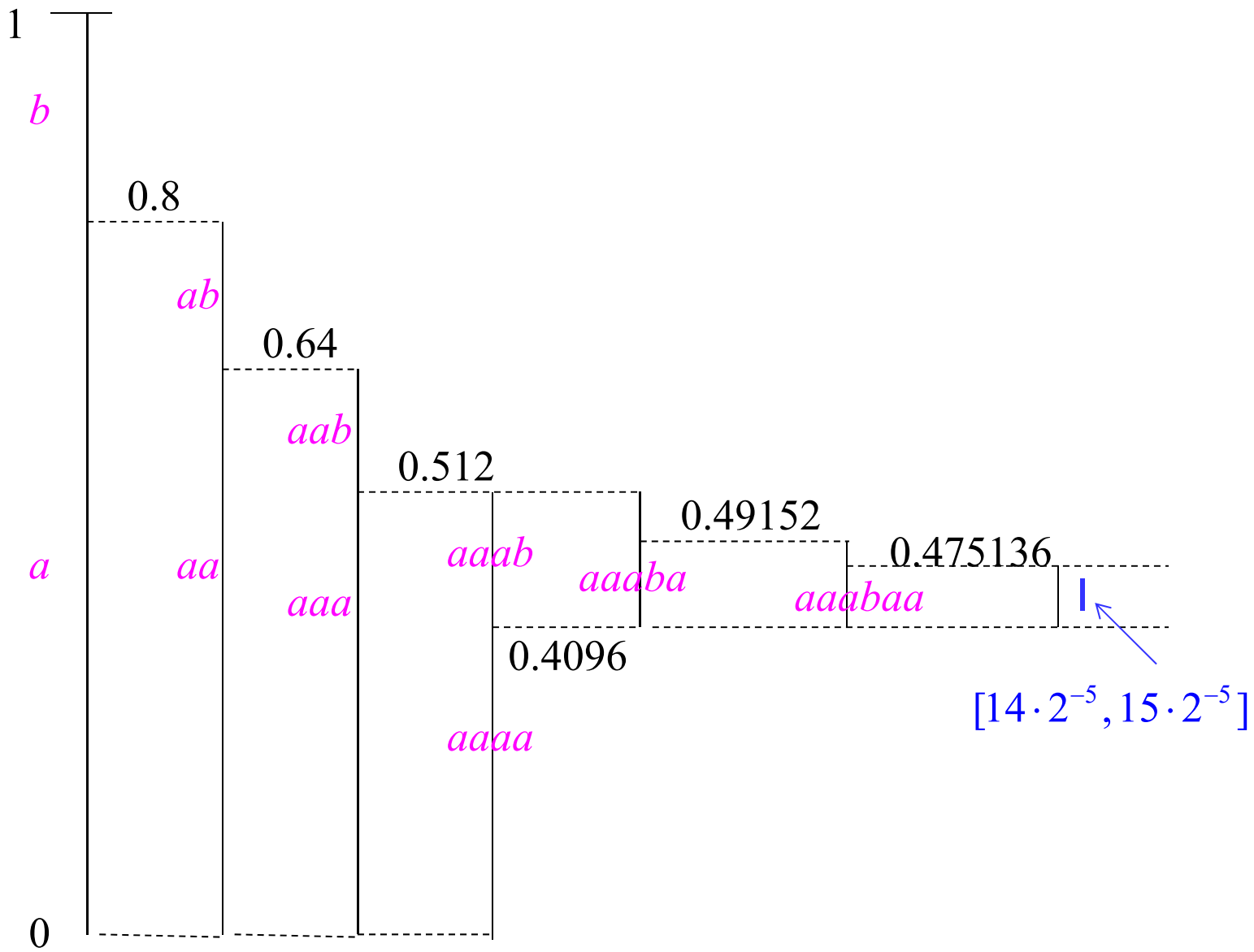
$$lower \le C \cdot k^{-b} < (C+1) \cdot k^{-b} \le upper$$

where C and b are integers (b is as small as possible), then the data X can be encoded by

$$C_{(k,b)}$$

where $C_{(k,b)}$ means that using k-ary (k 進位) and b bits to express C.

(註： Arithmetic coding 還有其他不同的方式，以上是使用其中一個較簡單的 range encoding 的方式)


1
b
0.8
ab
0.64
aab
0.512
aaab
0.49152
aaaba
0.475136
aaabaa
a
aa
aaa
0.4096
aaaa
0
$[14\cdot 2^{-5}, 15\cdot 2^{-5}]$

Example:

假設要對 X 來做二進位 $(k=2)$ 的編碼

且經由事先的估計，$X[i]=a$ 的機率為 0.8, $X[i]=b$ 的機率為 0.2

$\Rightarrow$ $P_1=0.8,\quad P_2=0.2,\qquad S_0=0,\quad S_1=0.8,\quad S_2=1$

若實際上輸入的資料為 $X=a\ a\ a\ b\ a\ a$

Initiation $(X[1]=a)$: $lower=0,\quad upper=0.8$

When $i=2$ $(X[2]=a)$: $lower=0,\quad upper=0.64$

When $i=3$ $(X[3]=a)$: $lower=0,\quad upper=0.512$

When $i=4$ $(X[4]=b)$: $lower=0.4096,\quad upper=0.512$

When $i=5$ $(X[5]=a)$: $lower=0.4096,\quad upper=0.49152$

When $i=6$ $(X[6]=a)$: $lower=0.4096,\quad upper=0.475136$

由於　　$lower = 0.4096, \quad upper = 0.475136$

$$lower \le 14 \cdot 2^{-5} < 15 \cdot 2^{-5} \le upper$$

$$0.4375 \qquad 0.46875$$

所以編碼的結果為

$$14_{(2,5)} = 01110$$

2 進位　5 個 bits

解碼

假設編碼的結果為 Y, length(Y) = b

其他的假設，和編碼 (see page 280) 相同　　使用 k 進位的編碼

Algorithm for arithmetic decoding

initiation:　$lower = 0$　　$upper = 1$　　$j = 1$

$lower\ 1 = 0$　　$upper\ 1 = 1$

for i = 1 : N　　% loop 1

check = 1;

while check = 1　　% loop 2

if there exists an n such that

$lower + (upper\text{-}lower)S_{n\text{-}1} \leqq$ lower 1　and

$lower + (upper\text{-}lower)S_n \geqq$ upper 1　are both satisfied,

then

check = 0;

(continue)….

else

$$upper\ 1 = lower\ 1 + (upper\ 1 - lower\ 1)(Y[j] + 1)k^{-j}$$

$$lower\ 1 = lower\ 1 + (upper\ 1 - lower\ 1)Y[j]k^{-j}$$

$$j = j+1$$

end

end % end of loop 2

$X(i) = n;$

$lower = lower + (upper\text{-}lower)S_{n\text{-}1}$

$upper = lower + (upper\text{-}lower)S_n$

end % end of loop 1

Coding Length for Arithmetic Coding

假設 P_n 是預測的 $X[i]=n$ 的機率

Q_n 是實際上的 $X[i]=n$ 的機率

(也就是說，若 length(X) = N, X 當中會有 $Q_n N$ 個 elements 等於 n)

則

$$upper - lower = \prod_{m=1}^{M} P_m^{Q_m N}$$

$\prod$: 連乘符號

另一方面，由於 (from page 281)

$$k^{-b} \le upper - lower < (2k)k^{-b}$$

想一想，為什麼

$$-\log_k (upper - lower) \le b < -\log_k (upper - lower) + 1 + \log_k 2$$

$$ceil\left(-N\sum_{m=1}^{M} Q_m \log_k P_m\right) \le b \le floor\left(-N\sum_{m=1}^{M} Q_m \log_k P_m + \log_k 2\right) + 1$$

$$ceil\left(-N\sum_{m=1}^{M}Q_m \log_k P_m\right) \le b \le floor\left(-N\sum_{m=1}^{M}Q_m \log_k P_m + \log_k 2\right)+1$$

在機率的預測完全準確的情形下， $Q_m = P_m$

Total coding length b 的範圍是

$$ceil\left(-N\sum_{m=1}^{M}P_m \log_k P_m\right) \le b \le floor\left(-N\sum_{m=1}^{M}P_m \log_k P_m + \log_k 2\right)+1$$

$$ceil\left(N\cdot\frac{entropy}{\log k}\right) \le b \le floor\left(N\cdot\frac{entropy}{\log k} + \log_k 2\right)+1$$

Arithmetic coding 的 total coding length 的上限比 Huffman coding 更低

◉ 8-F MPEG

MPEG： 動態影像編碼的國際標準　　全名： Moving Picture Experts Group

MPEG standard： http://www.iso.org/iso/prods-services/popstds/mpeg.html

MPEG 官方網站： http://mpeg.chiariglione.org/

人類的視覺暫留： 1/24 second

一個動態影像，每秒有 30個或 60個畫格 (frames)

例子：

Pepsi 的廣告

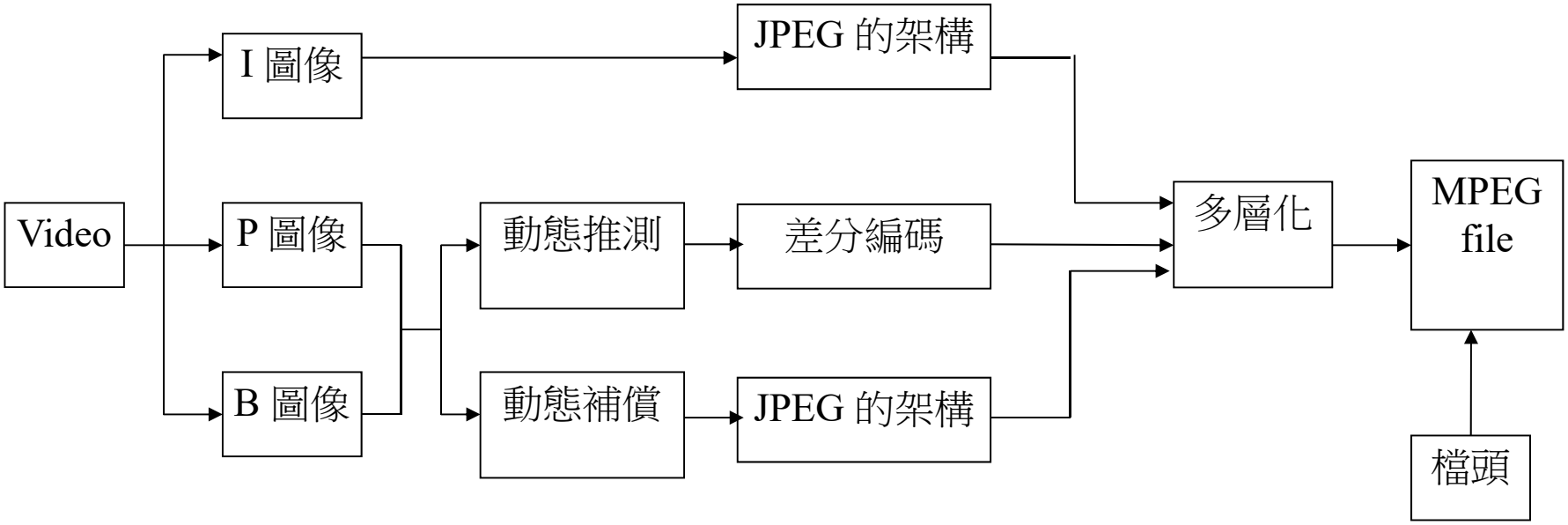
Size: 160 × 120　Time: 29 sec　一秒 30 個 frames

若不作壓縮：　160 × 120 × 29 × 30 × 3 = 50112000 = 47.79 M bytes。

經過 MPEG壓縮：　1140740 = 1.09 M bytes。

只有原來的 2.276%。

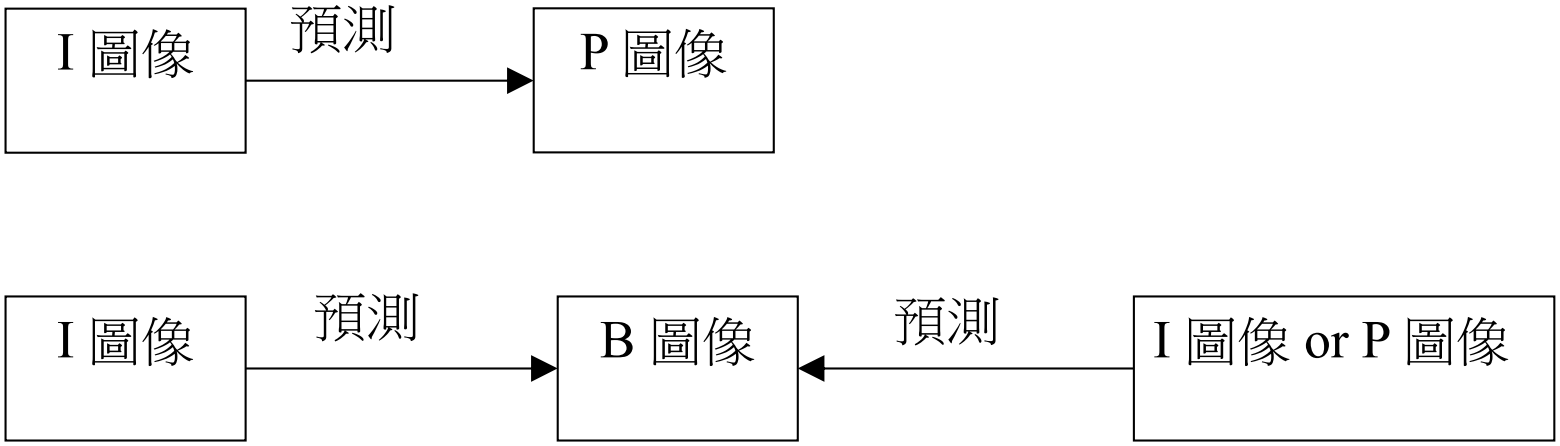
• Flowchart of MPEG Compression



I 圖像 (Intra-coded picture): 作為參考的畫格

P 圖像 (Predictive-coded picture): 由之前的畫格來做預測

B 圖像 (Bi-directionally predictive-coded picture): 由之前及之後的畫格來做預測

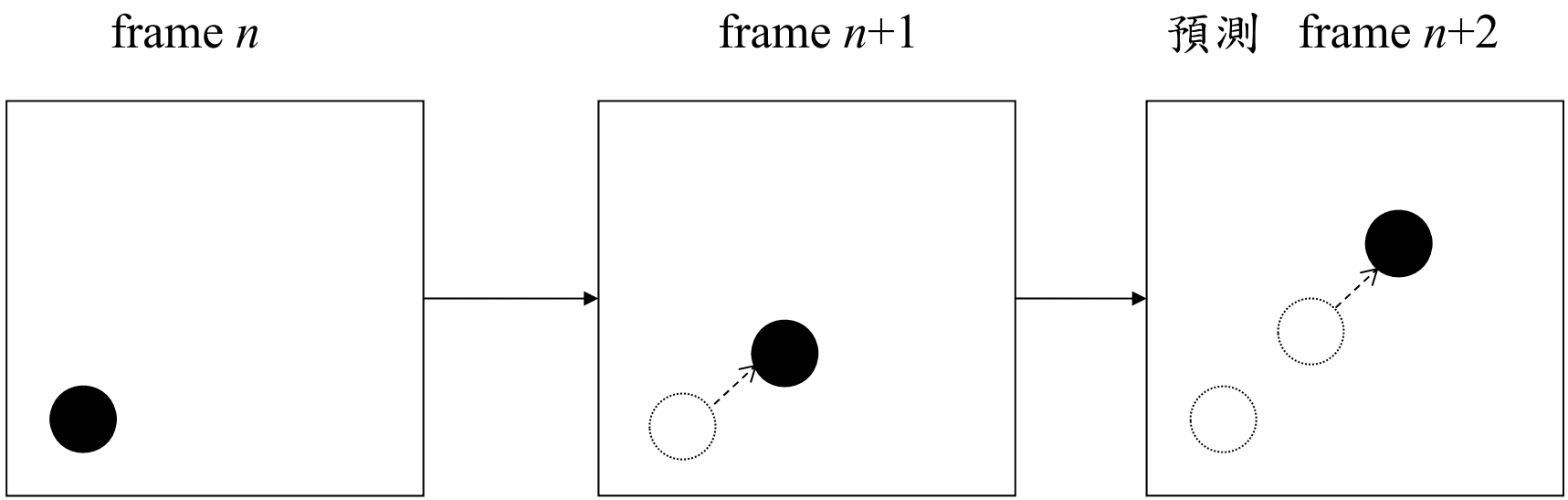


- **動態影像之編碼**

  原理：不同時間，同一個 pixel 之間的相關度通常極高
  只需對有移動的 objects 記錄 “motion vector”

- 動態補償 (Motion Compensation)

  時間相近的影像，彼此間的相關度極高



$F[m, n, t]$: 時間為 t的影像

如何由 $F[m, n, t]$， $F[m, n, t+\Delta]$ 來預測 $F[m, n, t+2\Delta]$?

(1) 移動向量 $V_x(m, n), V_y(m, n)$

(2) 預測 $F[m, n, t+2\Delta]$ ：

$$F_p[m, n, t+2\Delta] = F[m - V_x(m, n), n - V_x(m, n), t+\Delta]$$

(3) 計算 「預測誤差」

$$E[m, n, t+2\Delta] = F[m, n, t+2\Delta] - F_p[m, n, t+2\Delta]$$

對預測誤差 $E[m, n, t+2\Delta]$ 做 編碼

◉ 8-G Data Compression 未來發展的方向

Two important issues:

Q1: How to further improve the compression rate

Q2: How to develop a compression algorithm whose compression rate is acceptable and the buffer size / hardware cost is limited

附錄九：新的相似度測量工具：結構相似度
Structural Similarity (SSIM)

傳統量測兩個信號 (including images, videos, and vocal signals) 之間相似度的方式：

(1) maximal error $Max\left(\left|y[m,n]-x[m,n]\right|\right)$

(2) mean square error (MSE) $\frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left|y[m,n]-x[m,n]\right|^2$

(3) normalized mean square error (NMSE) $\frac{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left|y[m,n]-x[m,n]\right|^2}{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left|x[m,n]\right|^2}$

(4) normalized root mean square error (NRMSE) $\sqrt{\frac{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left|y[m,n]-x[m,n]\right|^2}{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left|x[m,n]\right|^2}}$

(5) signal to noise ratio (SNR)，信號處理常用

$$10\log_{10}\left(\frac{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}|x[m,n]|^2}{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}|y[m,n]-x[m,n]|^2}\right)$$

(6) peak signal to noise ratio (PSNR)，影像處理常用

$$10\log_{10}\left(\frac{X_{Max}^2}{\frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}|y[m,n]-x[m,n]|^2}\right)$$

X_{Max}: the maximal possible value of $x[m, n]$

In image processing, $X_{Max} = 255$

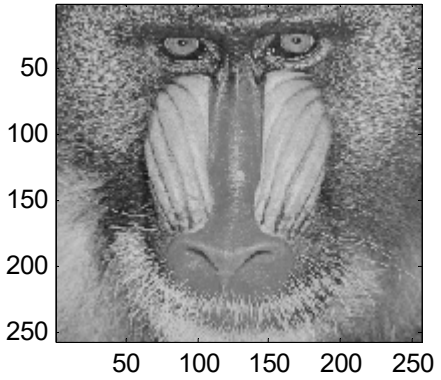
for color image:

$$10\log_{10}\left(\frac{X_{Max}^2}{\frac{1}{3MN}\sum_{R,G,B}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}|y_{color}[m,n]-x_{color}[m,n]|^2}\right)$$

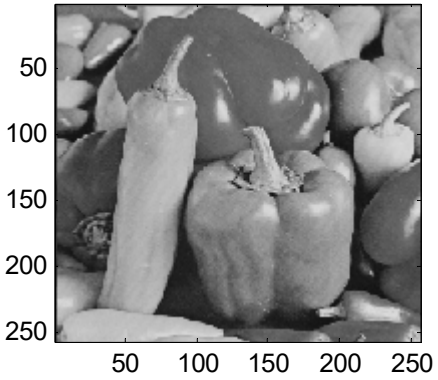
color = R, G, or B

然而，MSE 和 NRMSE 雖然在理論上是合理的，但卻無法反應出實際上兩個信號之間的相似度

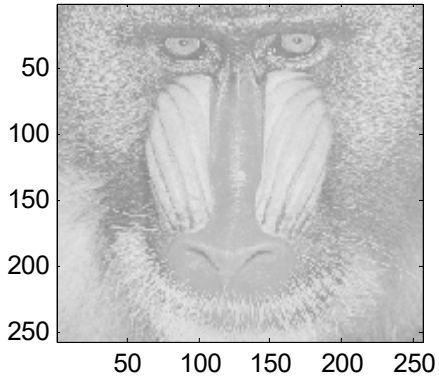
例如：以下這三張圖



圖一



圖二



圖三

圖三 = 圖一 × 0.5 + 255.5 × 0.5

照理來說，圖一和圖三較相近

然而，圖一和圖二之間的 NRMSE 為 0.4411

圖一和圖三之間的 NRMSE 為 0.4460

(7) Structural Similarity (SSIM)

有鑑於 MSE 和 PSNR 無法完全反應人類視覺上所感受的誤差，在 2004 年被提出來的新的誤差測量方法

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1L)}{(\mu_x^2 + \mu_y^2 + c_1L)} \frac{(2\sigma_{xy} + c_2L)}{(\sigma_x^2 + \sigma_y^2 + c_2L)}$$

$$DSSIM(x,y) = 1 - SSIM(x,y)$$

μ_x, μ_y: means of x and y σ_x^2, σ_y^2: variances of x and y

σ_{xy}: covariance of x and y c_1, c_2: adjustable constants

L: the maximal possible value of x − the minimal possible value of x

Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Processing*, vol. 13, no. 4, pp. 600−612, Apr. 2004.

若使用 SSIM，且前頁的 c_1, c_2 皆選為 1

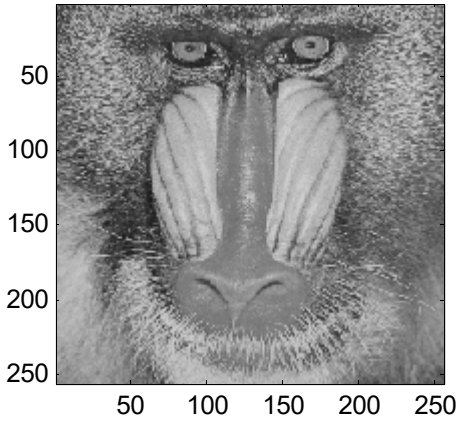
圖一、圖二之間的 SSIM 為 0.1040

圖一、圖三之間的 SSIM 為 0.7720

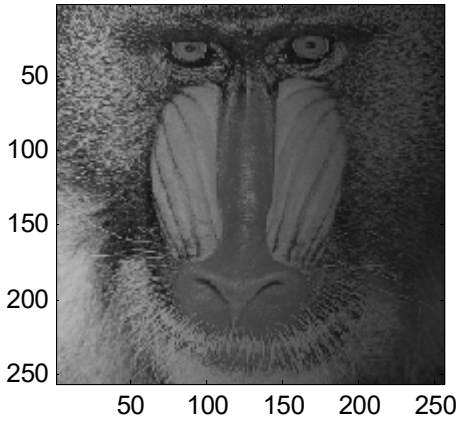
反應出了圖一、圖三之間確實有很高的相似度

其他幾個用 MSE 和 NRMSE 無法看出相似度，但是可以用 SSIM 看出相似度的情形

影子 shadow



圖四

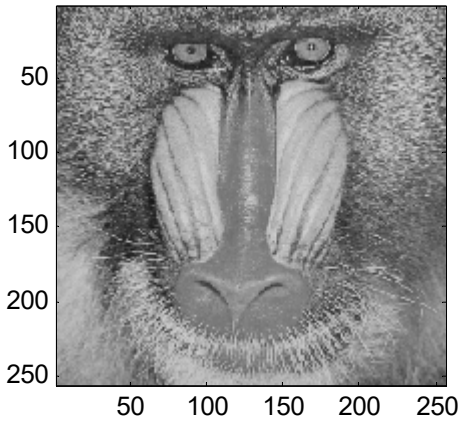


圖五

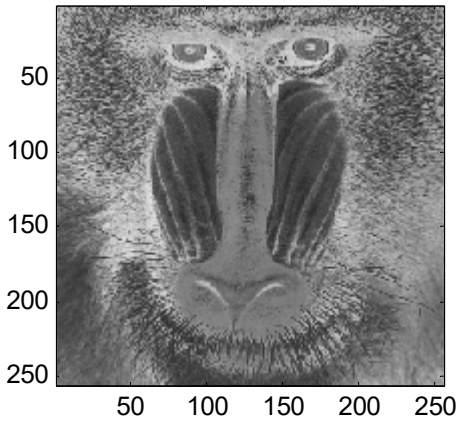
NRMSE = 0.4521 (大於圖一、圖二之間的 NRMSE)

SSIM = 0.6010

底片 the negative of a photo



圖六



圖七

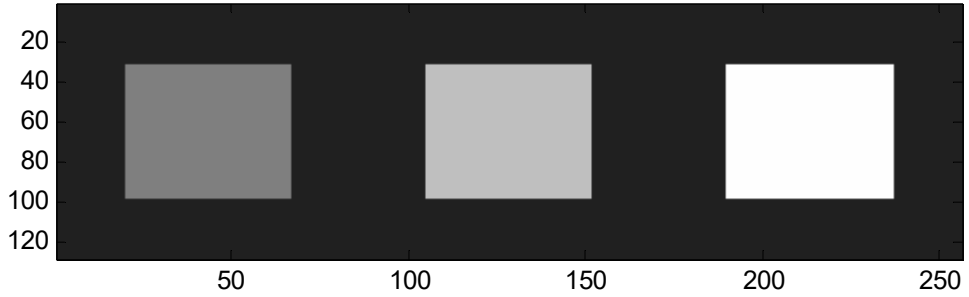
圖七 = 255 − 圖六

NRMSE = 0.5616 (大於圖一、圖二之間的 NRMSE)

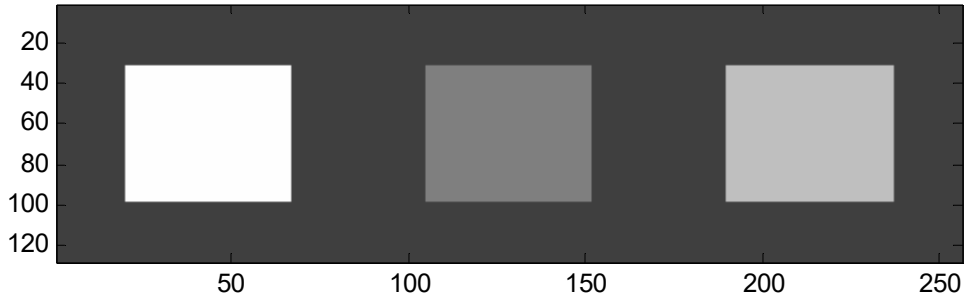
SSIM = -0.8367 (高度負相關)

同形，但亮度不同 (Same shape but different intensity)

圖八



圖九



NRMSE = 0.4978 (大於圖一、圖二之間的 NRMSE)

SSIM = 0.7333

思考：對於 vocal signal (聲音信號而言)

MSE 和 NRMSE 是否真的能反應出兩個信號的相似度？

為什麼？