第八章
空間資料結構設計

內　容

- 8.1　前　言
- 8.2　黑白影像表示法
- 8.3　影像加密
- 8.4　灰階影像表示法
- 8.5　作　業

8.1 前 言

- 將影像切割成許多的規律區塊，可用較省記憶體的空間資料結構 (Spatial Data Structures) 來表示這些區塊。

- 空間資料結構除了省記憶體的優點外，他還保有不需解壓的影像運算等方面的有效應用。

8.2 黑白影像表示法

- 四分樹切割

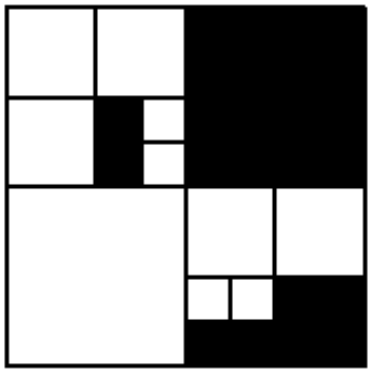


圖 8.2.1 黑白影像

圖 8.2.2 四分樹表示法

- 四分樹的正規化

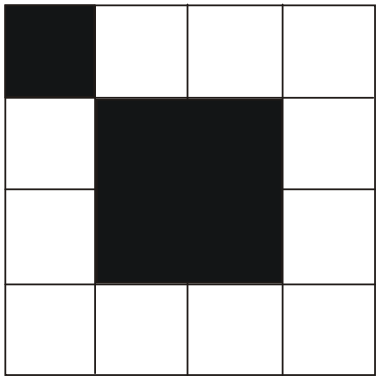
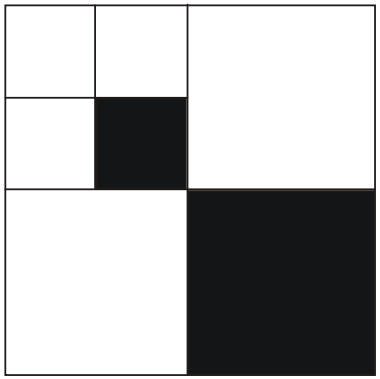
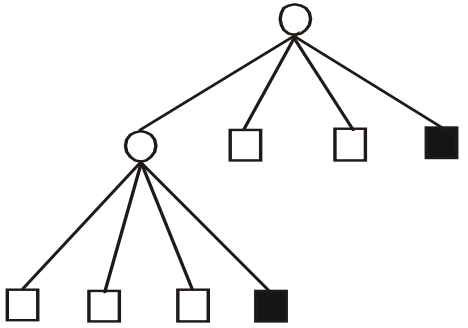


圖 8.2.3　4x4 黑白影像



(a) 移動後的結果



(b) 移動後的四分樹表示法

圖 8.2.4　移位後的效果

- 深先表示法

內部節點 ➔ 輸出G
白色外部節點 ➔ 輸出W
黑色外部節點 ➔ 輸出B

圖8.2.2的四分樹可表示成GGWWWGBWBWBWGWWGWWBBB。

圖8.2.2　四分樹表示法

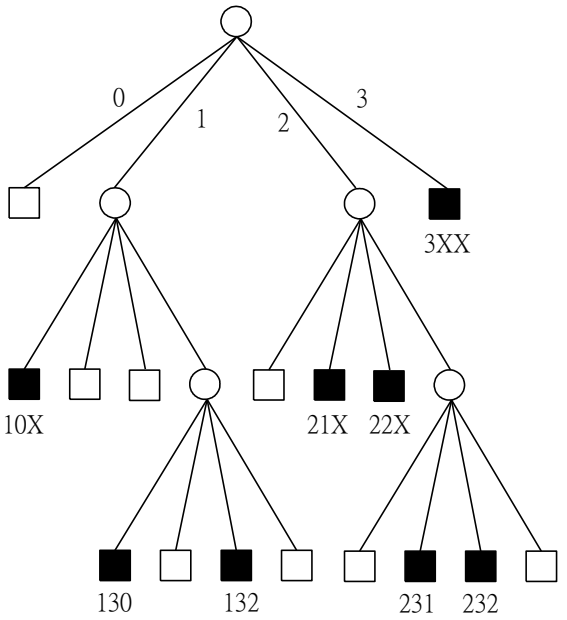
- 線性四分樹

利用深先搜尋方式，圖 8.2.2 的可表示為 030，032，1XX， 322，323，33X。

圖 8.2.2　四分樹表示法

10X, 130, 132, 21X, 22X, 231, 232, 3XX

解碼



8.3 影像加密

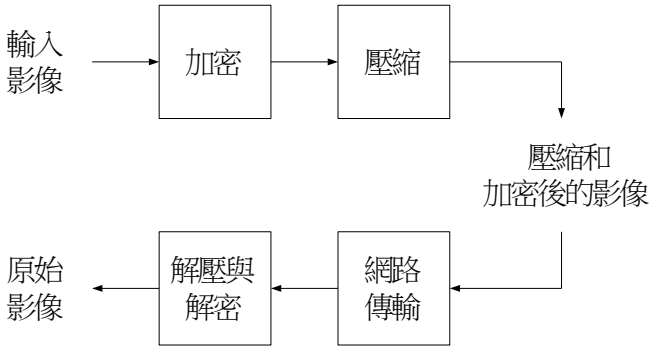
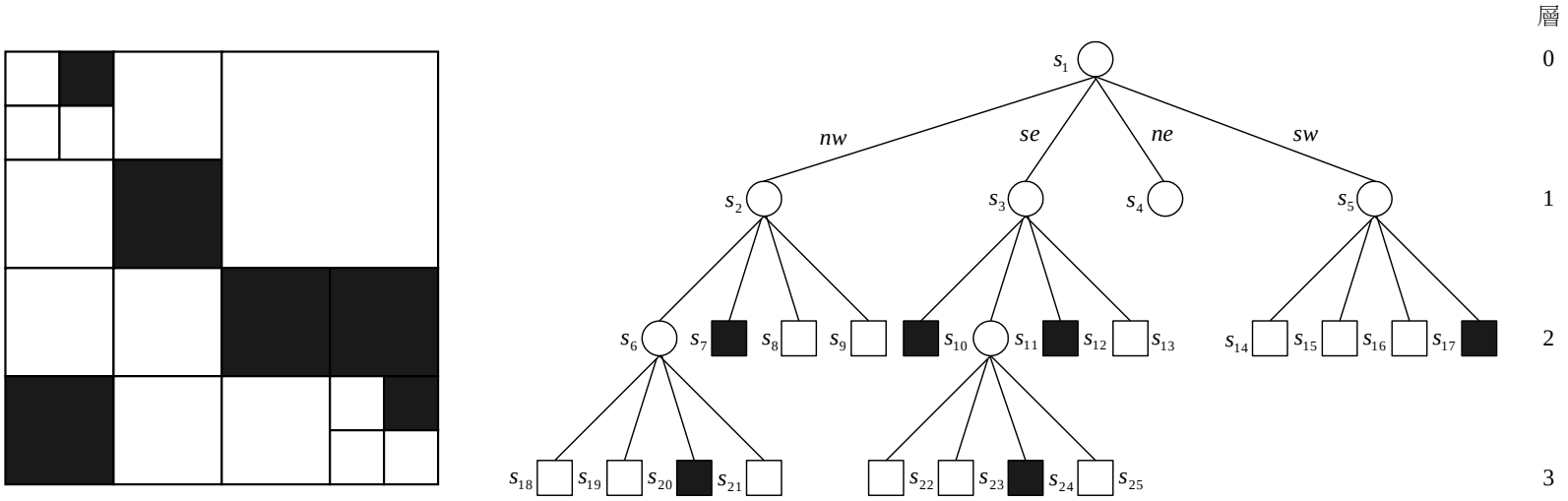


圖 8.3.1 影像加密系統



(a) 8×8 黑白影像　　(b) 四分樹結構

圖8.3.2 影像加密的例子

掃描語言

假設影像的大小為$2^n \times 2^n$，掃描語言可被定義為文法$G = \langle V_N, V_T, P, S \rangle$

$V_N = \left\{ S, \bigcup_{i=1}^{n} L_i \right\}$代表非終結符號集

L_i代表四分數中第i層的掃描圖案

$V_T = \left\{ \bigcup_{i=1}^{n} \Omega_i^{4^{i-1}} \middle| \Omega_i = \left\{ R_j^i \middle| 1 \le j \le 4^{i-1} \right\} \right\}$終結符號集

$\Omega_i^{4^{i-1}} = \left\{ \Omega_i \Omega_i \cdots \Omega_i \ (\Omega_i \text{連乘} 4^{i-1} \text{次}) \right\}$

R_j^i 圖8.2.3中定義的24個掃描圖案中的一個

S 代表起始符號

P 代表文法G中的產生規則

SP_i 為24個掃描圖案中的第i個掃描圖案

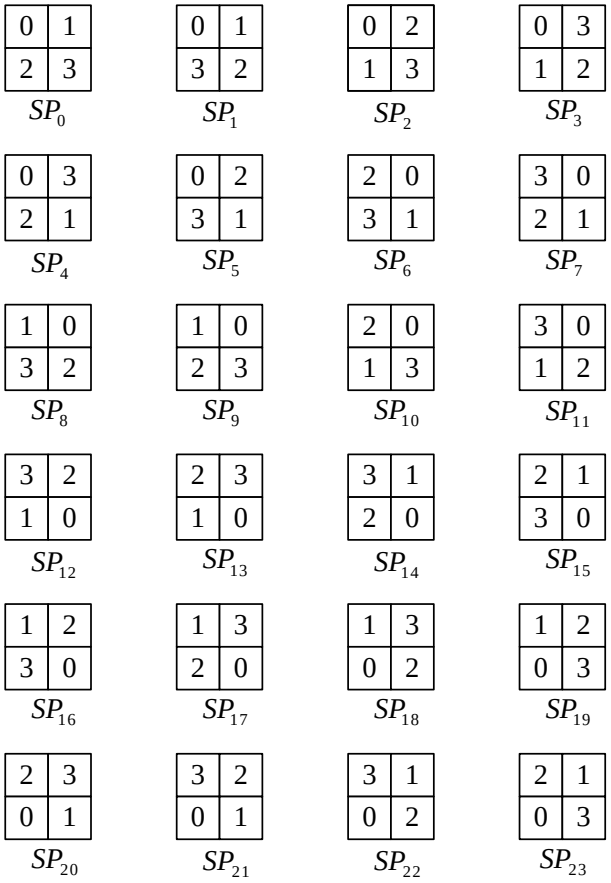


圖8.2.3　24 個掃瞄圖案

- 模擬例子

給定一組產生規則:

$$S \to L_1 L_2 L_3$$

$$L_1 \to R_1^1$$

$$L_2 \to R_1^2 R_2^2 R_3^2 R_4^2$$

$$L_3 \to R_1^3 R_2^3 R_3^3 R_4^3 R_5^3 R_6^3 R_7^3 R_8^3 R_9^3 R_{10}^3 R_{11}^3 R_{12}^3 R_{13}^3 R_{14}^3 R_{15}^3 R_{16}^3$$

$$R_1^1 = SP_1$$

$$R_1^2 = SP_{23}, \quad R_2^2 = SP_2, \quad R_3^2 = SP_4, \quad R_4^2 = SP_7$$

$$R_1^3 = SP_1, \quad R_2^3 = SP_{11}, \quad R_3^3 = SP_{13}, \quad R_4^3 = SP_1$$

$$R_5^3 = SP_4, \quad R_6^3 = SP_1, \quad R_7^3 = SP_0, \quad R_8^3 = SP_7$$

$$R_9^3 = SP_1, \quad R_{10}^3 = SP_{10}, \quad R_{11}^3 = SP_1, \quad R_{12}^3 = SP_{21}$$

$$R_{13}^3 = SP_{11}, \quad R_{14}^3 = SP_1, \quad R_{15}^3 = SP_{13}, \quad R_{16}^3 = SP_{15}$$

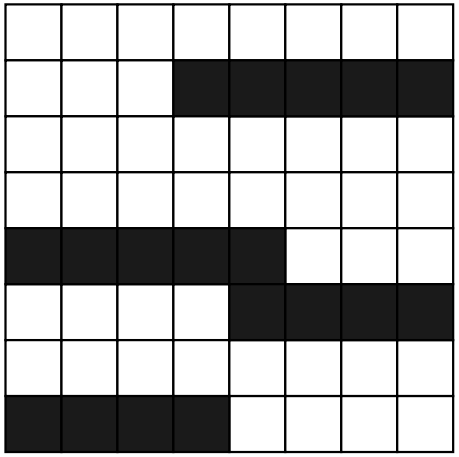


圖8.3.4　加密後的結果

則圖 8.3.2(a) 的黑白影像被加密成圖 8.3.4。將圖 8.3.4 予以壓縮。利用列掃瞄的方式，圖 8.3.4 可表示成
0000000000011111000000000000000111110000000
1111000000011110000，進而用 011516574844 來表示圖 8.3.4。

8.4 灰階影像表示法

- 一維線性內插

  圖 8.4.1 中的 O 點被表示為 (1, 5)，此處 1 表示 x 軸的位置而 5 表示灰階值；C 點被表示為 (11, 13)。假設 A 點的位置為 4，則 $\dfrac{\overline{OA}}{\overline{OC}} = \dfrac{\overline{AB}}{\overline{CD}}$

由 $\overline{AB} = \dfrac{\overline{OA} \times \overline{CD}}{\overline{OC}} = \dfrac{3 \times 8}{10} = 2.4$

得知

A 點的灰階值約為 7=(5+2)。

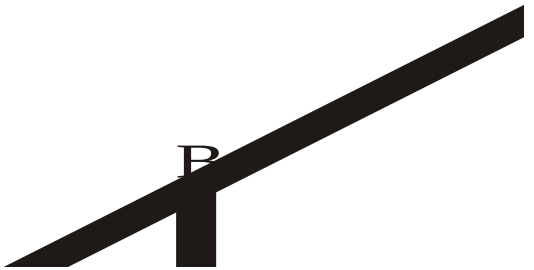


圖8.4.1 一維的線性內插

- 二分樹切割的條件

$$|g(x,y)-g_{est}(x,y)|\leq \varepsilon$$

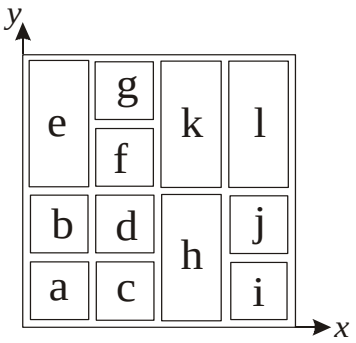


圖8.4.2　同質的區塊分割圖

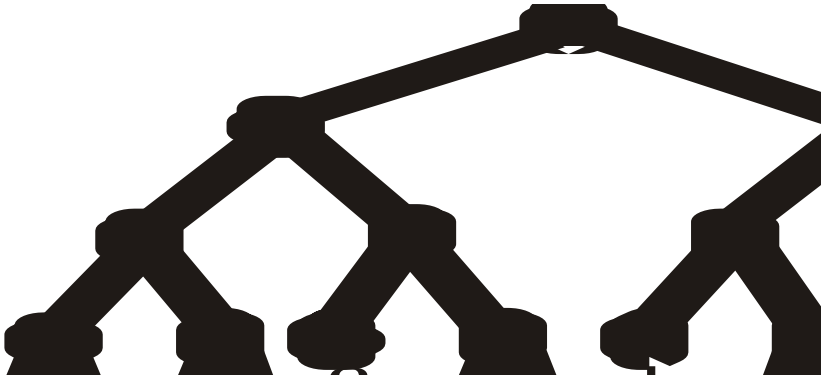


圖8.4.3　二分樹的表示法

- 求算 $g_{est}(x,y)$

  假設一個區塊的四個角點分別如下:

	位置	灰階值
左上	(x_1, y_1)	g_1
右上	(x_2, y_1)	g_2
左下	(x_1, y_2)	g_3
右下	(x_2, y_2)	g_4

  利用線性內插可以得到 $g_{est}(x,y) = g_5 + \frac{g_6 - g_5}{y_2 - y_1}(y - y_1)$ ，

  此處 $g_5 = g_1 + \frac{g_2 - g_1}{x_2 - x_1}(x - x_1)$ 和 $g_6 = g_3 + \frac{g_4 - g_3}{x_2 - x_1}(x - x_1)$ 。

- 廣先搜尋

  圖 8.4. 3 的二分樹用 S 樹表示如下:

  線性樹表：000000000101011111111111

  顏色表：$(e_{ul}, e_{ur}, e_{bl}, e_{br})$, $(h_{ul}, h_{ur}, h_{bl}, h_{br})$, … , $(j_{ul}, j_{ur}, j_{bl}, j_{br})$

- 重疊策略

採用重疊策略來降低區塊效應的影響。將原影像的最右邊一行和最底下一列重複一次，這使得原先 $2^n \times 2^n$ 大小的影像放大成 $(2^n+1)\times(2^n+1)$ 的大小。這個重疊策略 (Overlapping Strategy) 會使影像經二分樹分割後，鄰近的兩區塊會重疊一個像素的寬度，解壓出來後確可大幅降低區塊效應的影響。

- 實　驗

在 ε=21 時，圖 8.4.4 對應樹所需的 bpp (Bit Per Pixel) 約為 1.35 bits，這與原始影像一個像素需 8 個 bits 相比，壓縮改良率為 83%。

圖8.4.4　ε=21 得到的還原影像圖

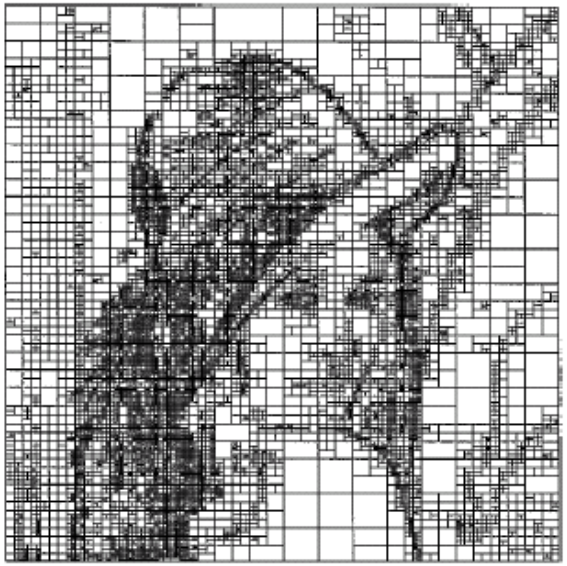


圖8.4.5　二元分割後的區塊示意圖

8.5 作 業

- 作業一：寫一 C 程式以完成將黑白影像轉成線性四分樹的實作。
- 作業二：寫一 C 程式以完成影像加密的實作。