

以臨界值控制區域函式分水嶺演算法為基礎之

彩色影像分割

Color Image Segmentation based on Watershed Algorithm using Threshold-Controlled Segmentation Function

孫振東

中國文化大學資訊科學系

jdusun@faculty.pccu.edu.tw

莊書瑋

中國文化大學資訊科學系

Edward.ch@msa.hinet.net

摘要

彩色影像分割是一項快速發展的數位影像處理技術，影像分割目的是將影像切割為數個不相重疊而各具有相同特性的區域或物件，典型的應用有物件抽取、物件辨識、以物件為基礎的影像壓縮、以及以內容為基礎的影像擷取等，近來有許多影像分割的相關技術與演算法被提出，其中分水嶺轉換(Watershed transformation)能有效地用在灰階與彩色影像分割，然而傳統的分水嶺轉換可能會產生嚴重的過度分割，也就是因原影像有許多局部最小點(Local minima)經轉換後產生許多小區域，本論文提出一個在梯度影像(Gradient image)裡去除無相關最小點(Irrelevant minima)的演算法，我們用梯度臨界值控制分割函式(Gradient-threshold-controlled segmentation function)之分水嶺轉換，全域梯度臨界值是由區域函式對臨界值的一次導數而得，根據實驗結果顯示本演算法對具不同特徵的影像，諸如是否有非均勻亮度、紋理、輪廓、與陰影等，皆可有效地改進分割精確性。

關鍵詞：梯度臨界值、彩色影像分割、分水嶺

Abstract

Color image segmentation is a rapidly developing technique of digital image processing. The goal of image segmentation is to divide the image into non-overlapping homogeneous regions or objects. Object extraction, object recognition, object-based compression, and content-based image retrieval are typical applications. Recently, a large number of techniques and algorithms have been proposed for image segmentation. Among them, those based on watershed transformation can be effectively used for segmentation of grey-scale and color images. However, conventional watershed

transform may produce a severe over-segmentation of the image, i.e., many small regions are produced due to many local minima in the input image. In this paper we propose an algorithm for eliminating irrelevant minima in the resulting gradient images. We use gradient-threshold-controlled segmentation function for watershed transformation. The global gradient threshold is obtained from the first derivative of the region-function with respect to threshold. The experimental results show that the proposed algorithm can effectively improve segmentation accuracy for different image natures, such as presence or not of non-homogeneous illumination, texture, contours, and shadows.

Key words : Gradient threshold, Color image segmentation, Watershed

一、前言

彩色影像分割是一項快速發展的數位影像處理技術，影像分割目的是將影像切割為數個不相重疊而各具有相同特性的區域或物件，典型的應用有物件抽取、物件辨識、以物件為基礎的影像壓縮、以及以內容為基礎的影像擷取等，近來有許多影像分割的相關技術與演算法被提出，其中分水嶺轉換(Watershed transformation)能有效地用在灰階與彩色影像分割，然而傳統的分水嶺轉換可能會產生嚴重的過度分割，也就是因原影像有許多局部最小點(Local minima)經轉換後產生許多小區域，本論文提出一個在梯度影像(Gradient image)裡去除無相關最小點(Irrelevant minima)的演算法，我們先將原影像轉成形態梯度影像(Morphological gradient image)，以作為分水嶺轉換之輸入影像，稱之為梯度臨界值控制分割函式(Gradient-threshold-controlled segmentation function)，由區域函式對臨界值的一次導數變化特性決定全域梯度臨界值(Global gradient threshold)，用此全域梯度臨界值去除無相關最小點，達到較佳影像分割，根據實驗結果顯示本演算

法對具不同特徵的影像，諸如是否有非均勻亮度、紋理、輪廓、與陰影等，皆可有效地改進分割精確性。

二、原理分析

影像分割的主要工作就是利用影像中的訊息將不同的物件分割出來，灰階影像分割的方法比較常見的有[1][2]: (1)基於長條圖的(histogram-based)分割法[3]: 是利用影像的灰階統計長條圖，找出臨界值以區分物件及背景，(2)基於邊緣的(edge-based)分割法[4][5]: 是利用物件和背景間，影像的亮度會有明顯的變化之特性，來找出物件的邊緣，(3)基於區域的(region-based)分割法[6][7]: 是利用同一物件中之影像亮度值相近的特性，來找出物件，以及(4)混合的(hybrid)分割法: 即為混合以上的方法來分割影像，以分水嶺分割法[8][9]為代表。

分水嶺分割法可說是目前最有效的影像分割法之一[10][11]，原是由 Vincent and Soille 所提出[8]，它的觀念是將一張影像看成是立體的地形圖，把影像的灰階值當成是地形的高度，如圖 1，猶如高低起伏的山脈、河川、及平原等地理景觀，每個區域最小值的點當成是每個盆地(Catchment basin)的最低點，模擬水從最低的盆地開始慢慢漲上來，若漲到某個盆地的最低點，該盆地也開始漲水，等到來自不同盆地的水即將要匯流時，便建立一個水壩(dam)將之隔離，如圖 2，此水壩就是我們要找的分水嶺線。

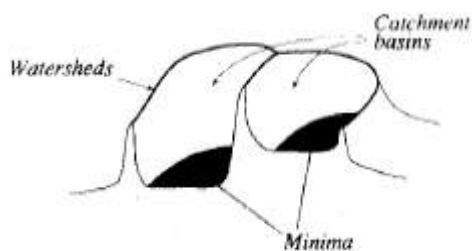


圖 1 (來源取自[8])

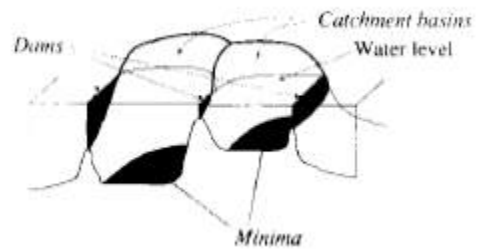


圖 2 (來源取自[8])

分水嶺演算法主要分為兩個步驟:

步驟一、排序(sorting)

首先將一張灰階影像依照像素的灰階值由小到大排列，存入一個陣列裡，必須可以對影像中的任何像素及其鄰居像素的值做隨機存取。

步驟二、模擬漲水(flooding)

排序完成後，開始模擬漲水的動作，假設水漲到高度 h ，此時已進水的蓄水盆地之個別最小值高度一

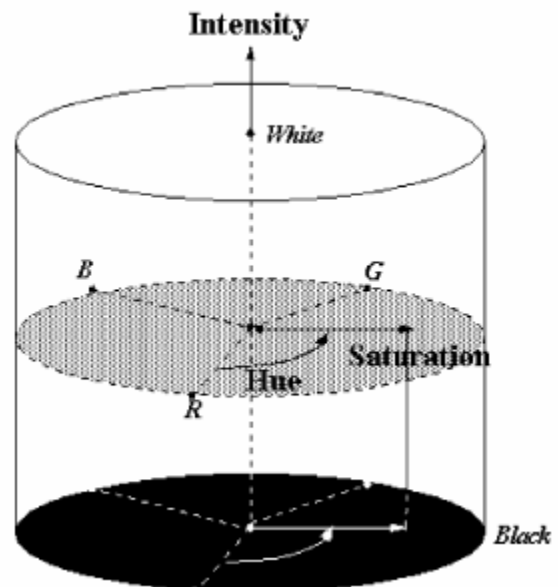


圖 3 HSI 色彩空間

定是小于等於 h ，這些最小值的點都會給予一個獨特的標籤(label)。由於之前完成排序，即可以對 $h+1$ 高度(即灰階值為 $h+1$)的像素做存取，給予這些像

素一個標籤值 MASK，此時需要一個 FIFO(First in first out)的 queue，將那些標為 MASK 且鄰居像素已有被標上 label 的像素推入 queue，然後一個一個拿出來，根據它的鄰居像素來決定它的標籤值，如果鄰居像素也有標上 MASK 的，也把它丟進 queue 裡，屬於同一個蓄水盆地的像素都會被標上相同的標籤值，如此便會從蓄水盆地的最低點慢慢往外延伸。而那些雖然是標成 MASK 但鄰居像素都沒有被標過 label 的像素，它就是一個新的區域最小值，我們給予它們不同於其他蓄水盆地的標籤值，經過此步驟後，不屬於任何蓄水盆地的 pixel 都會被標成分水嶺點，最後產生出分水嶺線。

原分水嶺分割方法是用在灰階影像，若將其用在彩色影像而又不願失去彩色訊息，則會發生上述步驟一的排序問題，為解決此問題，參考文獻[12]提出以向量排序，將常用的紅-綠-藍(R,G,B)色彩空間(color space)轉成與人類視覺較相近的色彩空間(H,S,I)，H 為色相(Hue)，S 為彩度(Saturation)，I 為亮度(Intensity)，其幾何描述如圖 3 所示，(R,G,B)和 (H,S,I) 的轉換公式如下：

RGB 轉成 HIS：

$$H = \begin{cases} \theta & \text{if } B \leq G \\ 360 & \text{if } B > G \end{cases}$$

$$\theta = \cos^{-1} \left\{ \frac{[(R-G) + (R-B)]/2}{[(R-G)^2 + (R-B)(G-B)]^{1/2}} \right\}$$

$$S = 1 - \frac{3}{(R+G+B)} [\min(R, G, B)]$$

$$I = (R+G+B)/3$$

HSI 轉成 RGB：

$$0^\circ \leq H < 120^\circ$$

$$B = I(1-S)$$

$$R = I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right]$$

$$G = 3I - (R+B)$$

$$120^\circ \leq H < 240^\circ$$

$$H = H - 120^\circ$$

$$R = I(1-S)$$

$$G = I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right]$$

$$B = 3I - (R+G)$$

$$240^\circ \leq H \leq 360^\circ$$

$$H = H - 240^\circ$$

$$G = I(1-S)$$

$$B = I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right]$$

$$R = 3I - (G+B)$$

向量排序(Vector ordering)規則如下：

- (1) 先以 I 由小到大排序
- (2) 若 I 相同，則以 S 由大到小排序
- (3) 若 I 與 S 皆相同，則以 H 由小到大排序

在做分水嶺演算法時，僅需考慮 I 即可[12]，因若對 H 與 S 亦同時做，對分割效果並沒有顯著改進，為了要解決過度分割的問題，需先對輸入的影像做一些前處理，在此之前我們先定義數學形態學中的 Gray-level Erosion 和 Dilation，

Erosion:

$$E_G(A, B) = \min_{[j,k] \in B} \{a[m+j, n+k] - b[j, k]\}$$

Dilation:

$$D_G(A, B) = \max_{[j,k] \in B} \{a[m-j, n-k] + b[j, k]\}$$

A 為影像， B 為形態學裡的結構元素(structure element)，常用的是 3×3 和 5×5 大小的結構元素， $a[x,y]$ 為 A 影像中像素 (x,y) 的像素值， $b[x,y]$ 為結構元素 B 中像素 (x,y) 的像素值， (m,n) 為 A 中正在處理的像素座標， (j,k) 為 B 中正在處理的像素座標，

- (3) 若 I 與 S 皆相同，則以 H 由小到大排序
2. 執行閉合(Closing)，再將影像單純化，使易於分割。
本步驟僅需考慮 I 即可，因若對 H 與 S 亦同時做此步驟，對分割效果並沒有顯著改進：

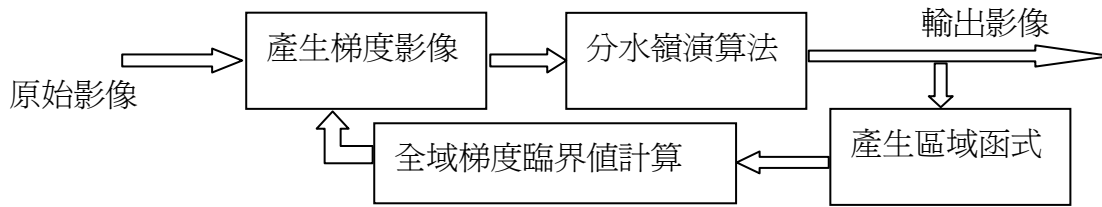


圖 4 系統流程

Erosion 的做法是把結構元素 B 的中心點移到欲處理的像素 (m,n) 上，例如結構元素為 3×3 ，包括 (m,n) 以及鄰居 8 像素，分別和相對應的結構元素之像素值做相減運算，最後由此 9 個像素中取出像素值最小者取代 (m,n) 的像素值，Dilation 則以此類推。本研究用的結構元素為 3×3 ，像素值均為 0。接著定義 Closing，Closing 的做法是對影像做 Dilation 後再 Erosion。我們首先對輸入的影像做單純化處理，以產生易於分割的影像，方法是將原圖做 Closing 運算後的影像做為遮罩影像，然後把經過 Dilation 後的原始影像不斷的做 Erosion，每做一次就和遮罩影像的每一像素比較，取最大值，直到沒有任何像素有改變為止，最後得到一張簡單化的影像。接著利用剛才得到的單純化影像來產生梯度影像(gradient image)，我們使用數學形態學中梯度差運算，分別對影像做 Dilation 和 Erosion 處理，將兩張處理後的影像做差集得到梯度差影像，經過此處理後，物件的邊緣會被顯現出來。不過用剛才得到的梯度影像去做分水嶺運算還是會出現太多的分割，主要因為有太多的區域極小值，所以如何把梯度影像中 intensity 太小的像素去除，亦即將這些像素的 intensity 改成 0，是本研究的重點。

- (1) Closing 的做法是對影像做 Dilation 後再做 Erosion。
- (2) 影像單純化的做法是以上述 Closing 運算後的影像做為遮罩影像，然後把經過 Dilation 後的原始影像不斷的做 Erosion，每做一次就和遮罩影像的每一像素比較，取最大值，直到沒有任何像素有改變為止，最後得到一張簡單化的影像。
3. 執行形態學梯度運算，產生梯度影像。使用數學形態學中梯度差運算：對步驟 2 所得簡單化影像分別做 Dilation 和 Erosion 處理，將兩張處理後的影像做差集得到梯度影像。
4. 設定各個全域臨界值，執行分水嶺演算法，產生區域函式。
 - (1) 臨界值 t 的初始值設為 0。
 - (2) 梯度影像中凡像素之 I 值小於臨界值者，將這些像素的 I 設成 0。
 - (3) 執行分水嶺演算法，得到分割區域數(即 Label 數， L)。
 - (4) 逐步增加臨界值 t ，重複執行上述(2)(3)步驟，即可得到臨界值與分割區域數的對應關係 (t, L) ，稱之區域函式(Region function) f ， $L = f(t)$ 。
5. 計算區域函式對臨界值的一次導數，出現第一次平坦導數值(連續四個導數在額定誤差 Δ 範圍內)的臨界值即得最佳全域臨界值 t 。

$$\begin{aligned} |f(t+i+1) - f(t+i)| &\leq \Delta \\ i &= 0, 1, 2, 3 \quad t = 1, 2, \dots, 252 \end{aligned}$$

根據實驗結果顯示，本演算法對具不同特徵的影像，諸如是否有非均勻亮度、紋理、輪廓、與陰影等，皆可有效地改進分割精確性。

三、系統設計

本研究的系統流程如圖 4 所示，經分水嶺演算法輸出的影像，可以得到分割區域數與臨界值的關係，稱之區域函式(Region function)，由區域函式對臨界值的一次導數變化特性決定全域梯度臨界值(Global gradient threshold)，用此全域梯度臨界值去除無相關最小點，達到較佳影像分割，詳細步驟如下：

1. 輸入彩色影像，轉至 HSI 色彩空間，將像素做向量排序。
向量排序(Vector ordering)規則如下：
 - (1) 先以 I 由小到大排序
 - (2) 若 I 相同，則以 S 由大到小排序

四、測試結果

首先我們用既有簡單分水嶺方法來實作，如 MATLAB (Image Processing Toolbox)所提供的分水嶺方法，下列是我們用工具軟體 MATLAB 7.0 所撰寫的 MATLAB 程式碼及對輸入影像

Pepper(size:256×256)執行的結果(如圖 5 所示)，MATLAB Toolbox 所用的分水嶺演算法是原創者 (Vincent and Soille)的方法。

```

rgb=imread('Pepper256Color.bmp');
hsi=rgb2hsi(rgb);
H=hsi(:, :, 1);
S=hsi(:, :, 2);
I=hsi(:, :, 3);
%
fi=double(I);
L=watershed(fi);
wr=L==0;
%
r=rgb(:, :, 1);
g=rgb(:, :, 2);
b=rgb(:, :, 3);
r(wr)=255;
g(wr)=255;
b(wr)=255;
rgb=cat(3, r, g, b);
imshow(rgb)

```

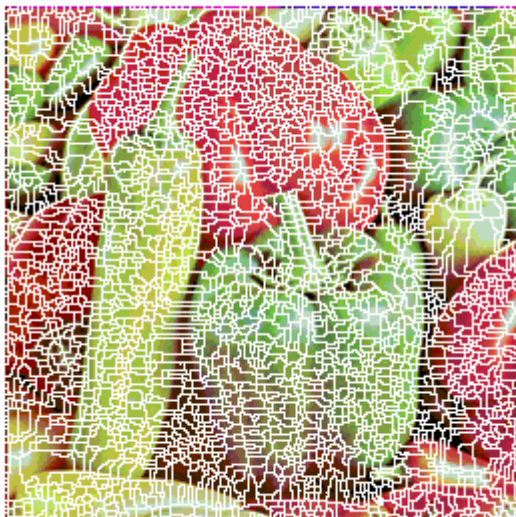


圖 5

我們再進一步將影像做線性空間濾波(Laplacian Filter)，再經 Close-opening filtering 前處理，目的是將太亮或太暗的小區塊濾除(小於 3×3 結構元素的區塊)，下列是我們用工具軟體 MATLAB 7.0 所

撰寫的 MATLAB 程式碼及對輸入影像 Pepper(size:256×256)執行的結果(如圖 6 所示)。

```

rgb=imread('Pepper256Color.bmp');
hsi=rgb2hsi(rgb);
H=hsi(:, :, 1);
S=hsi(:, :, 2);
I=hsi(:, :, 3);
w=fspecial('laplacian', 0.5);
I_filtered=imfilter(I, w, 'replicate');
fi=double(I_filtered);
g2=imclose(imopen(fi, ones(3, 3)), ones(3, 3));
L2=watershed(g2);
wr2=L2==0;
r=rgb(:, :, 1);
g=rgb(:, :, 2);
b=rgb(:, :, 3);
r(wr2)=255;
g(wr2)=255;
b(wr2)=255;
rgb=cat(3, r, g, b);
imshow(rgb)

```



圖 6

本研究之演算法是用 Borland C++ Builder 實現，對影像 Pepper 之實驗所得區域函式如下，關係序偶

(t, L) 為：

(0, 1159), (1, 1159), (2, 1008), (3, 864), (4, 755), (5, 674), (6, 579), (7, 509), (8, 449) *, (9, 398), (10, 344), (11, 301), (12, 272)

設定 $\Delta = 60$ ，得 $t = 8$ ，分割影像如圖 7 所示，與上述既有簡單分水嶺方法所得結果顯著不同與改進。

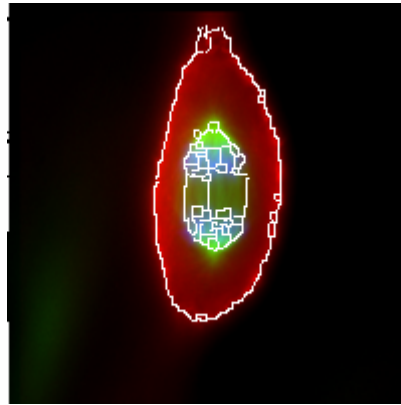
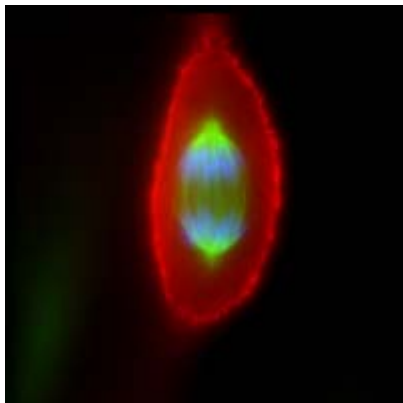


圖 7

我們選擇了各類型 21 張圖進行測試，顯示順序為：圖形編號，出處，最佳臨界值，原圖，分割圖，測試數據(圖形編號，臨界值，區域數)，其中*表示最佳分割

1. http://www.itg.uiuc.edu/exhibits/gallery/images/atlas/as/anaphase_02.jpg

臨界值：06

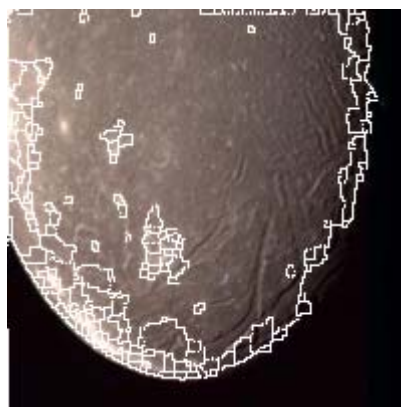


(1,0,215), (1,1,221), (1,2,119), (1,3,78), (1,4,65), (1,5,50), (1,6,45) *, (1,7,37), (1,8,34), (1,9,30), (1,10,28), (1,11,23)

2.

<http://www.itg.uiuc.edu/exhibits/gallery/images/atlas>

臨界值：07



(2,0,882), (2,1,882), (2,2,743), (2,3,556), (2,4,414), (2,5,311), (2,6,236), (2,7,188) *, (2,8,137), (2,9,122), (2,10,94), (2,11,79), (2,12,71), (2,13,58)

3. <http://www.ls.eso.org/images/lasilla.jpg>

臨界値：14

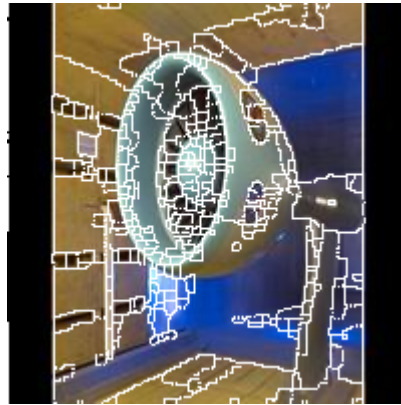


(3,0,840), (3,1,841), (3,2,562), (3,3,373), (3,4,259),
(3,5,191), (3,6,144), (3,7,115), (3,8,96),
(3,9,81), (3,10,74), (3,11,61), (3,12,55), (3,13,51),
(3,14,44) *, (3,15,37), (3,16,33), (3,17,30)

4.

http://grcimagenet.grc.nasa.gov/GRCDigitalImages/1995/1995_01896L.jpg

臨界値：9



(4,0,907),(4,1,906),(4,2,724),(4,3,563),(4,4,444),(4,5,
357),(4,6,308),(4,7,258),(4,8,229),(4,9,195) *,
(4,10,176),(4,11,161),(4,12,148),(4,13,136),(4,14,128
) ,(4,15,114)

5.

http://grcimagenet.grc.nasa.gov/GRCDigitalImages/1996/1996_00950L.jpg

臨界値：19

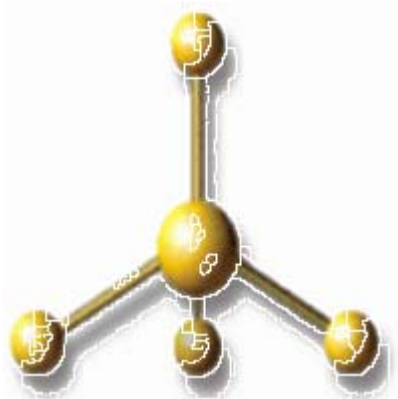
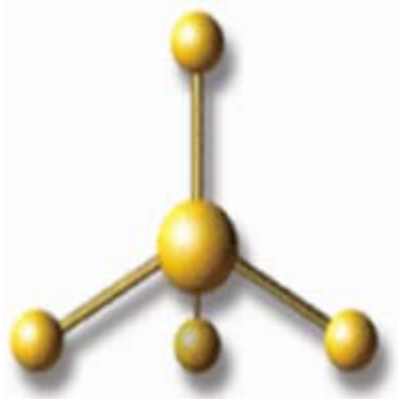


(5,0,786),(5,1,782),(5,2,483),(5,3,317),(5,4,242),(5,5,
196),(5,6,156),(5,7,143),(5,8,118),
(5,9,112),(5,10,106),(5,11,100),(5,12,85),(5,13,78),(5,

14,75),(5,15,71),(5,16,65),(5,17,64),
 (5,18,59),(5,19,52) *, (5,20,49), (5,21,48), (5,22,39),
 (5,23,35),(5,24,33),(5,25,32)

6. <http://www.indigo.com/graphics/molecule.gif>

臨界值：19



(6,0,259),(6,1,259),(6,2,231),(6,3,215),(6,4,207),(6,5,
 189),(6,6,178),(6,7,171),(6,8,160),(6,9,153),
 (6,10,139),(6,11,133),(6,12,126),(6,13,113),(6,14,109
),(6,15,98),(6,16,70),(6,17,64),(6,18,61),(6,19,58) *,
 (6,20,54),(6,21,50),(6,22,47)

7. http://www.eso.org/paranal/images/ut1fl_400.jpg

臨界值：10



(7,1,661),(7,2,472),(7,3,363),(7,4,300),(7,5,231),(7,6,
 162),(7,7,129),(7,8,108),(7,9,93),(7,10,83) *,
 (7,11,77),(7,12,69),(7,13,66),(7,14,63),(7,15,60)

8.

<http://lisar.larc.nasa.gov/IMAGES/SMALL/EL-1996-00050.jpeg>

臨界值：09





(8,0,688),(8,1,686),(8,2,226),(8,3,110),(8,4,74),(8,5,54),(8,6,41),(8,7,35),(8,8,25),(8,9,16) *,
(8,10,10),(8,11,9),(8,12,7),(8,13,5),(8,14,5)

9.

<http://lisar.larc.nasa.gov/IMAGES/SMALL/EL-1996-00082.jpeg>

臨界值：18



(9,0,1026),(9,1,1026),(9,2,768),(9,3,571),(9,4,430),(9,5,336),(9,6,280),(9,7,249),(9,8,216),(9,9,193),(9,10,163),(9,11,141),(9,12,120),(9,13,102),(9,14,91),(9,15,85),(9,16,79),(9,17,70),(9,18,64) *,

(9,19,58),(9,20,50)

10. _

<http://www.dfrc.nasa.gov/gallery/photo/Facility-FLL/Small/EC94-42421-20.jpg>

臨界值：05



(10,0,965),(10,1,964),(10,2,739),(10,3,453),(10,4,313),(10,5,278) *,
(10,6,232),(10,7,217),(10,8,200),(10,9,185),(10,10,167)

11.

臨界值：07



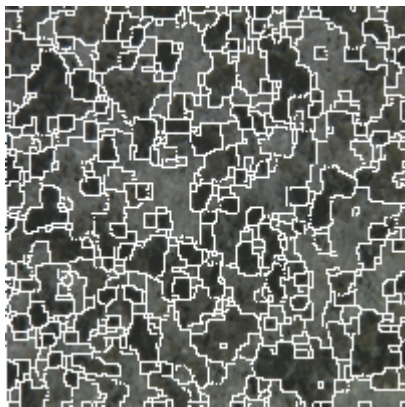
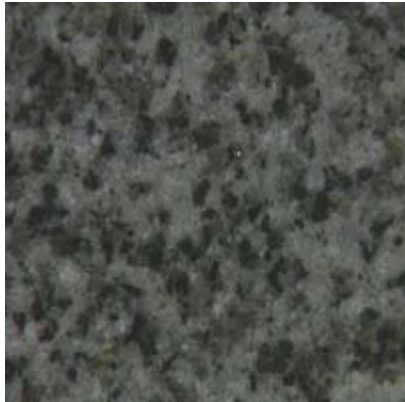


(11,0,1159),(11,1,1160),(11,2,1008),(11,3,864),(11,4,755),(11,5,674),(11,6,579),(11,7,509) *,
(11,8,449),(11,9,398),(11,10,344),(11,11,301),(11,12,272)

12

<http://www.outex.oulu.fi/db/color/scaled/granite001-inca-300dpi-00.bmp>

臨界値：10



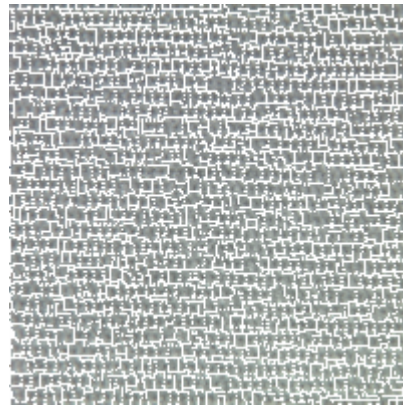
(12,0,1288),(12,1,1287),(12,2,1258),(12,3,1204),(12,4,1125),(12,5,1045),(12,6,954),(12,7,865),
(12,8,739),(12,9,629),(12,10,531)

*,(12,11,423),(12,12,335)

13

<http://www.outex.oulu.fi/db/color/scaled/paper001-inca-300dpi-00.bmp>

臨界値：1



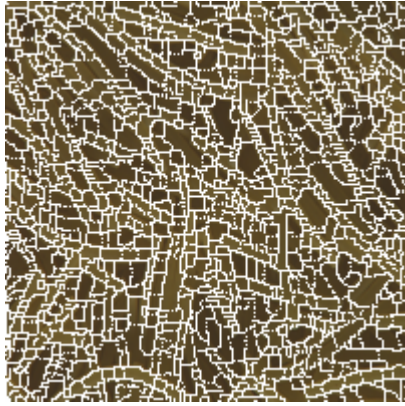
(13,0,1265),(13,1,1266) *,(13,2,1089)

14

<http://www.outex.oulu.fi/db/color/scaled/pasta001-inca-300dpi-00.bmp>

臨界値：1



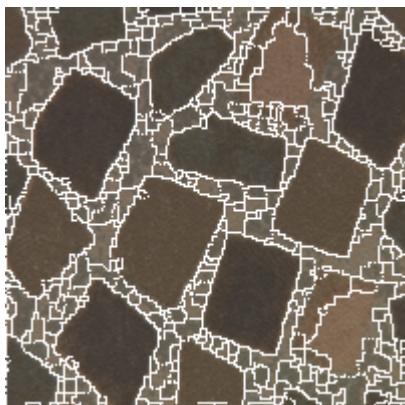


(14,0,1032),(14,1,1033) *,(14,2,752)

15

<http://www.outex.oulu.fi/db/color/scaled/plastic001-inca-300dpi-00.bmp>

臨界值：5



(15,0,996),(15,1,996),(15,2,684),(15,3,594),(15,4,464),(15,5,380) *,(15,6,289),(15,7,213)

16

<http://www.outex.oulu.fi/db/color/scaled/wood001-inca-300dpi-00.bmp>

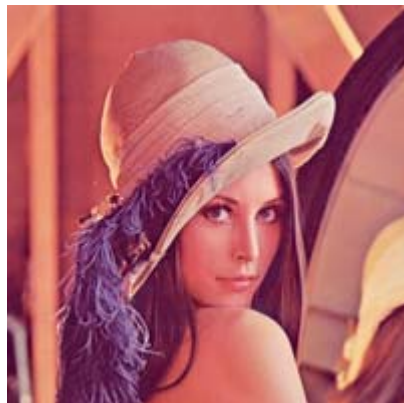
臨界值：1



(16,0,987),(16,1,987) *,(16,2,525)

17

<http://neuron.et.ntust.edu.tw/homework/91/NN/91NNHomework2/M9109005/main/ben/mark/lenna.JPG>





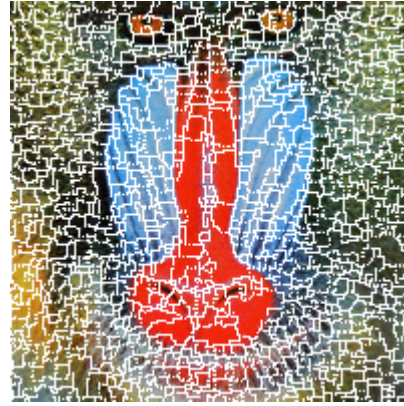
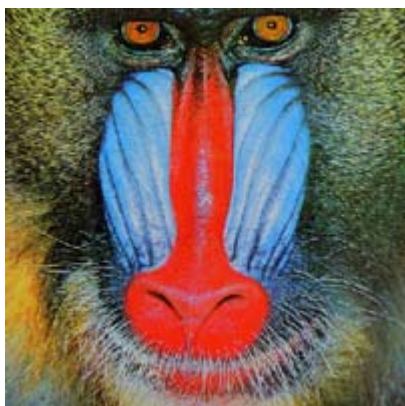
臨界値：03 臨界値：07

(17,0,1066),(17,1,1064),(17,2,888),(17,3,722)
*,(17,4,600),(17,5,493),(17,6,409),(17,7,357)
*,(17,8,312)

18

<http://neuron.et.ntust.edu.tw/homework/91/NN/91NNHomework2/M9109005/main/ben/mark/monkey.bmp>

臨界値：3



(18,0,1314),(18,1,1314),(18,2,1236),(18,3,1140) *,
(18,4,1033),(18,5,929),(18,6,818),(18,7,730),(18,8,644)
(18,9,572)

19 <http://www.ppl.s.chiba-u.jp/%7Eesyoshida/sy-face.jpg>

臨界値：4



(19,0,980),(19,1,979),(19,2,714),(19,3,580),(19,4,484)
*,(19,5,414),(19,6,366)

20 http://www.angola.org.uk/images/taag_face.jpg

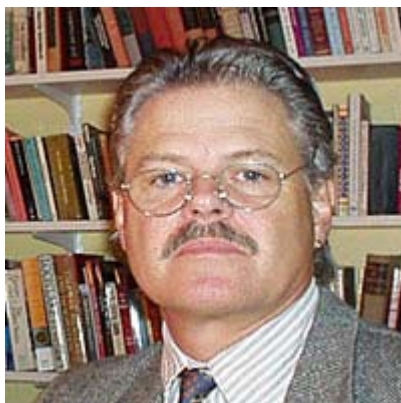
臨界值：3



(20,0,829),(20,1,830),(20,2,714),(20,3,582)
*,(20,4,495),(20,5,425)

21 <http://www.acornresume.com/face.jpg>

臨界值：9



(21,0,1320),(21,1,1320),(21,2,1230),(21,3,1153),(21,4,1067),(21,5,974),(21,6,867),(21,7,789),(21,8,702),(21,9,622) *,(21,10,561),(21,11,508),(21,12,461) °

五、分析與討論

分析：

我們列出測試 12 與測試 21 的區域函式，可以發現到全是紋理影像(測試 12)的區域函式較陡峭如圖 8 所示，而僅具部分紋理影像(測試 21)的區域函式則先陡峭而後趨於平坦如圖 9 所示，在測試 12 中，臨界值為 9 之後的 4 個導數值分別為 88、89、87、及 86，平均為 87.5，誤差為 1.5，是第一次出現平坦導數值，則臨界值 9 定為最佳臨界值，在測試 21 中，臨界值為 9 之後的 4 個導數值分別為 57、57、58、及 53，平均為 56.25，誤差為 1.75，是第一次出現平坦導數值，則臨界值 9 定為最佳臨界值，我們以目測觀察上述測試皆有較佳的分割結果，根據此演算法，無論什麼特徵的影像，其區域函式第一次導數趨於平滑時，即連續對應四個臨界值的導數相近時，即可得最佳分割。

討論：

- (1) 本研究的演算法不需考慮圖內各區域的特徵，這是本演算法的優點。
- (2) 本演算法最大計算量是在計算區域函式，經初步實驗結果顯示並不需計算所有的臨界值，將

可節省計算量。

- (3) 本演算法的關鍵是在找第一次出現區域函式導數的平坦點，亦即連續四個導數在額定誤差範圍內，如何設定誤差範圍是關鍵，由於判定是否為最佳分割，我們是以主觀觀察而得，因此如何對最佳分割做客觀定量分析亦是本研究需要再深入研究的地方。

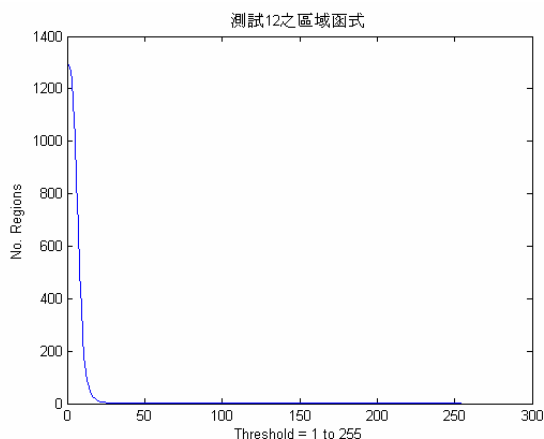


圖 8

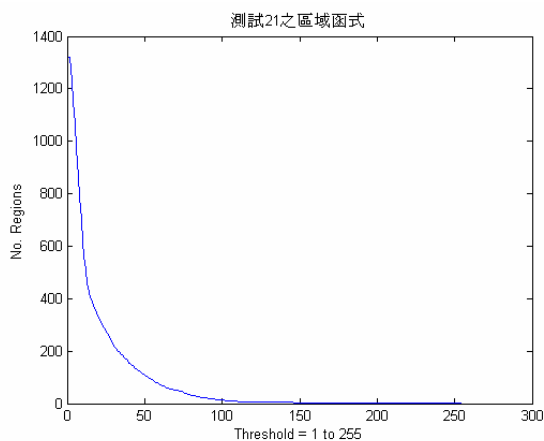


圖 9

六、參考文獻

- [1] S.K. Pal et al., A review on image segmentation techniques, Pattern Recognition, 29, pp.1277-1294, 1993.
- [2] R. Gonzalez and R. Woods, Digital Image Processing, Prentice-Hall, 2nd ed., 2002.
- [3] T. geraud, P.-Y. Strub, and J. Darbon, "Color image segmentation based on automatic morphological clustering," EPITA research and development Laboratory, in Proceedings of IEEE International Conference on Image Processing 2001, vol.3, pp. 70-73, 2001.
- [4] M.D. Heath, S. Sarkar, T. Sanocki and K.W. Bowyer, "A Robust Visual Method for Assessing the Relative Performance of Edge-Detection Algorithm," IEEE Transaction Pattern Analysis and Machine Intelligence, vol.19, no.12, pp. 1338-1359, 1997.
- [5] R.J. Qian and T.S. Huang, "Optimal Edge Detection in Two_Dimensional Images," IEEE Transaction Image Processing, vol. 5, no. 7, pp. 1215-1220, 1996.
- [6] J.F. Haddon and J.F. Boyce, "Image Segmentation by Unifying Region and Boundary Information," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 10, pp.929-948, 1990.
- [7] S.A. Hojjatoleslami and J. Kittler, "Region Growing: A New Approach," IEEE Transaction Image Processing, vol. 7, no. 7, pp. 1079-1084, 1998.
- [8] L. Vincent and P. Soille, "Watershed in digital spaces: An efficient algorithm based on immersion simulations," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.13,no.6, pp. 583-598, 1991.
- [9] Jos B.T.M Roerdink and Arnold Meijster, "The Watershed Transform: Definition, Algorithms and Parallelization Strategies," Fundamental Informatics, IOS Press, pp. 187-228, 2001.
- [10] M. Kazanov, "A new color image segmentation algorithm based on watershed transformation," Proc. Of the 17th international conference on pattern recognition, pp. 590-593, August 2004.

- [11]A. Hugué and M. Andrade, "Color-based watershed segmentation of low-altitude aerial images," Proc. Of the 17th international conference on pattern recognition, pp. 138-145, August 2004.
- [12]G.Louverdis,M.I. Vardavoulia,I.Andreadis, PH. Tsalides, "A new approach to morphological image Processing," Pattern Recognition, vol. 35, pp.1733-1741,2002.
- [13]R. Gonzalez, R. Woods, and S.Eddins, Digital Image Processing using MATLAB, Prentice-Hall, 2004.
- [14]Demin Wang, "A multiscale gradient algorithm for image segmentation using watersheds," Pattern Recognition, vol. 30, no. 12, pp. 2043-2052, 1997.