

植基於距離測量之影像檢索系統

黃國峰¹ 黃慧鳳² 林嘉文³

¹ 國立臺中科技大學多媒體應用系研究所

^{2,3} 國立臺中科技大學資訊工程系研究所

³ likeakid104@gmail.com

摘要

以內容為基底的影像檢索(content-based image retrieval)是近二、三十年來相當受到注意的影像處理技術之一。許多研究者透過萃取影像的顏色、紋理、形狀或空間位置得到影像的特徵，再利用特徵之間的比對，來找到資料庫裡與輸入影像最類似的數張影像。研究者們鑽研出許多不同的特徵萃取方法，以及各種不一樣的距離比對方式，希望能夠簡單又快速的獲取更佳的影像檢索準確率。

在這篇論文中，我們提出一個新的距離比對方式，利用第一次檢索過後的結果，找出參照特徵，再次進行檢索並對結果排序(re-rank, 簡稱 RRRK)，能夠非常輕易的提升影像檢索的效率。通過實驗，結果證明了此方法是相當有效的，而且我們相信這個方法可以被應用在其他的影像檢索技術中，提升其影像檢索效能。

關鍵詞：影像檢索系統、CBIR、HSV 色彩空間、灰階共現矩陣、距離比對

1. 前言

隨著時代的演變及網際網路的興盛，人類的溝通工具，除了傳統的文字以外，影像或是圖片已經成為最重要的媒體格式之一了。在數位時代的快速發展之下，人們愈發習慣利用影像或圖片記錄生活週遭的一切，因此，網路上充斥著成千上萬的圖片；而獲取資訊幾乎是現代人們的基本需求，於是，影像檢索技術(image retrieval)成為了一個熱門的研究項目。

一般來說，影像檢索技術可以區分為三大類，分別是以內文為基礎的方式(text-based)、以內容為基礎的方式(content-based)及以含義為基礎的方式(semantic-based)[1]。

由於以內文為基礎及以含意為基礎的影像檢索的影像檢索方法，分別存在著耗費大量人力及技術難度尚無法克服等缺點，以內容為基礎的影像檢索仍然是目前最重要的一個方式。

在CBIR(content-based image retrieval)系統的改進上，許多學者都針對特徵值的擷取提出了改善的方式，如 Wang 等人[3]提出了一個結合顏色、紋理及形狀特徵的影像檢索系統。Lin 等人[8]則提出了結合顏色和紋理特徵的影像檢索系統。Liu 等人[1]提出了一個稱為 micro-structure descriptor(MSD)的方法，他們模擬了人類視覺處理系統的早期階段，成功的結合了顏色、紋理、形狀及顏色分布資訊成

為一個整體特徵，再利用這個特徵進行影像間的距離比對，提升了檢索的效率。

有些學者則針對距離計算的方式提出改進的方法。Xiang 等人[9]在關於形狀比對的論文中提出一個重新計算距離的方法。形狀比對的系統和影像檢索系統非常類似，在形狀的描述子確認之後，系統萃取每個形狀圖片的特徵，接著 query image 則和資料庫的每一張影像進行距離計算，然後再根據資料庫裡每一張影像和 query image 距離，進行排序，最後得出和 query image 最相像的數張影像。Xiang 等人認為這種傳統的距離計算方式無可避免的會產生一定的錯誤，即資料庫的某一張影像雖然在與 query image 的距離計算上無法取得足夠相近的距離，但此張影像卻和 query image 的許多鄰居有著相當接近的距離，此影像理應納入與 query image 最相近的幾張影像之一，但因為傳統的直接的距離比對，而得到較差的排序。於是 Xiang 等人提出了距離重新計算方法，其主要概念是每兩張影像之間的距離，不應該只考慮其中直接的距離計算，而應該考慮整個資料庫內所有影像中彼此的距離(Graph Transduction)。Xiang 等人的實驗證明，他們提出的距離重新計算方法能夠獲取相當顯著的進步。Wang 等人[10]則繼承了 Xiang 等人提出的想法，加以改善之後，得到了更令人滿意的成果。

相較於透過改進顏色、紋理、形狀等特徵取得方式而取得效率的提升，改善 CBIR 系統距離量測方式來增進系統效能的相關文獻，顯得相當少見。我們相信這是一個改進 CBIR 系統效能的嶄新視點，透過距離量測系統的改進，應能為 CBIR 系統帶來更多的效率提升。因此，我們以此觀點設計一個新的 CBIR 系統，希望能夠取得更加的效能。

2. 植基於距離測量之影像檢索系統

本章將介紹本文提出的影像檢索系統。我們參考了 Xiang 等人的論文，提出一個適用於 CBIR 系統的特徵距離計算方式，由於此距離計算方法仍需要一個影像檢索系統當作平台，所以我們簡單的發展一個影像檢索系統，加上本文提出來的距離測量方法，再透過實驗測試新方法的成效。

考量到系統的有效性，如果只萃取單一種影像的低階特徵，恐怕無法取得令人滿意的準確率，所以本系統將萃取影像的顏色及紋理特徵來加以比對；同時，我們也盡量簡化系統，希望能夠取得較少的計算成本。

本章幾個小節將如下安排：1.系統概觀 2.萃取顏色特徵 3.萃取紋理特徵 4.重新測距的方法。

2.1 系統概觀

本論文建置的影像檢索系統，其架構和其他的影像檢索系統並無太大差異。首先我們萃取 query image 和影像資料庫裡的每一張影像的顏色及紋理特徵，形成特徵向量，接著 query image 和資料庫裡的每一張影像進行特徵向量間的距離計算，然後根據每張影像和 query image 間的距離進行排序，如此我們便得到和 query image 最相像的數張影像。不同於傳統的影像檢索系統的是，我們在最後的排序步驟之後，再加上我們所提出的重新測距(RRK)的方法，在重新測距之後，我們得到新的排序，而此新的排序，即是我們最後得到的結果。

2.2 萃取顏色特徵

2.2.1 HSV 彩色直方圖

因為 HSV 色彩空間(HSV color space)能夠符合人類的視覺感知，因此在許多 CBIR 系統中都被廣泛的運用[2][4][5][6]。HSV 利用三個值來描述一個顏色：H 為色相(hue)、S 為飽和度(saturation)、V 為亮度(value)。

我們首先將 RGB 值轉變成 HSV 值。接著，基於節省計算成本考量，我們將 HSV 值做簡單的量化：Hue 代表顏色的不同，我們認為這屬於顏色中的比較重要的部分，所以平均的量化為 m_1 等份；Saturation 及 Value 則分別平均量化為 m_2 、 m_3 等份。因此，每一張影像我們得到一個 M 個顏色($M = m_1 \times m_2 \times m_3$)的 HSV 色彩空間直方圖(HSV color space histogram)，這個直方圖我們用 M 維的特徵向量表示。

2.2.2 HSV 彩色直方圖的距離計算

我們設計的系統力求有效性及降低計算成本，故我們選擇 L_1 距離計算法[1]來進行 HSV 彩色直方圖的距離計算。我們萃取 query image 的顏色特徵，得到一個 M 維度的特徵向量 $Q_{color} = [Q_{c1}, Q_{c2}, \dots, Q_{cM}]$ ，同時，資料庫的每一個影像的顏色特徵被萃取出來，得到一個 M 維度的特徵向量 $T_{color} = [T_{c1}, T_{c2}, \dots, T_{cM}]$ 。這兩個向量間的距離計算方式，如公式(1)所列：

$$D_{color}(Q_{color}, T_{color}) = \sum_{i=1}^M |Q_{ci} - T_{ci}| \quad (1)$$

L_1 距離計算法的計算方式非常簡單，不需要進行平方或是平方根的運算，可以非常有效的降低計算成本，很適合用在大型的影像資料庫。在我們提出的系統中，根據後來實驗的結果，我們將 M 設定為 256，即 $M = 8 \times 4 \times 4 = 256$ (hue 平分為 8 等份，saturation 及 value 分別平分為 4 等份)。

2.3 萃取紋理特徵

2.3.1 灰階共現矩陣(gray level co-occurrence matrix)

在 CBIR 系統中，利用顏色特徵來區別及辨識影像是簡單而有效的。但是，僅僅依靠一種低階特徵進行影像檢索，往往無法取得令人滿意的準確度，因此，為了得到較佳的影像檢索效能，我們的系統除了顏色特徵以外，也將萃取影像的紋理特徵。

Haralick 等人[15]所提出的灰階共現矩陣(gray level co-occurrence matrix, GLCM)，是一個相當著名的紋理分析方法，我們以此為基礎，發展本系統紋理特徵萃取的方法。

灰階共現矩陣產生方式是統計影像空間內，兩相對位置的像素在符合預先設定的條件下(例如：方向、距離)出現相同灰階值的次數；也就是說，灰階共現矩陣是以像素分佈機率統計歸納成矩陣形式，檢視在相同幾何位置上兩相對像素的灰階相關性，即可達到描述影像灰階值的幾何分布情形之目的。圖 2.2 是四個方向的灰階共現矩陣的示意圖。

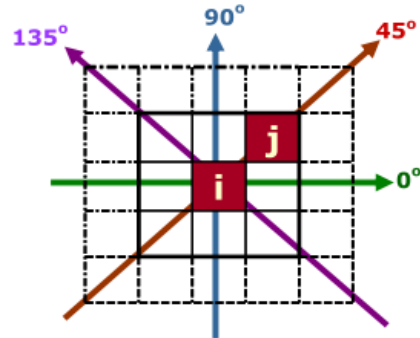


圖 2.1 四個方向的灰階共現矩陣示意圖

以圖 2.1 為例，圖中每個格子皆代表一個像素， i 和 j 為影像灰階像素值，矩陣即為統計 (i, j) 在條件為像素間距離的距離 d 和角度為 θ 組合下所出現的次數。灰階共現矩陣因為角度及位置不同，會產生不同的灰階共現矩陣。圖 2.2 為灰階共現矩陣運算範例，條件設定為 $d=1, \theta=90^\circ$ 。圖 2.2 中(a)為一張 5×5 大小的一張灰階影像，灰階值設定為 4 階，(0~3)，(b)為簡單的計算範例，即 i, j 值都為 3 時，且距離 $d=1$ ，角度 $\theta=90^\circ$ 的分布情形(紅色框框)，(c)的紅框即為(b)計算的結果，其他值則依此類推計算。

1	0	3	3	1
2	3	3	1	2
3	3	2	3	1
3	2	0	3	1
1	1	0	1	0

(a) 尺寸為 5×5 大小的灰階影像

1	0	3	3	1
2	3	3	1	2
3	3	2	3	1
3	2	0	3	1
1	1	0	1	0

(b) i, j 值都為 3 時，且距離 $d=1$ ，角度 $\theta=90^\circ$ 的分布情形

j \ i	0	1	2	3
0	1	1	1	0
1	0	1	2	3
2	0	2	0	2
3	1	1	1	4

(c)根據(a)圖產生的灰階共現矩陣

圖 2.2 灰階共現矩陣計算範例

矩陣內的每一個值代表每一組相對位置的像素值分布在一張影像內出現的次數，一般做法是利用統計的方式，對共現矩陣再進行運算，得到對比度(Contrast)、相關度(Correlation)、熵(Entropy)、能量(Energy)、同質度(Homogeneity)和最大機率(Max probability)等紋理特徵，再利用這些特徵來進行影像間的比對。不過，在本論文中，我們設計的方法力求簡單有效，所以我們將根據灰階共現矩陣直接進行運算。

透過灰階共現矩陣的計算，根據設定的距離及方向，每張影像即可產生一個灰階共現矩陣，但若距離及方向改變，則能夠產生不一樣的灰階共現矩陣，例如：假設距離設定為 2 種，方向設定為 4 個，則能產生 8 個(2 x 4)不一樣的灰階共現矩陣。我們參考以往相關的論文，將距離設定為 1，角度則分別設為 0°、45°、90°、135°，分別進行實驗，結果發現四個角度雖然有著不一樣的準確度，但其實相差不大，若是將四個角度通通考量進行運算，增加了四倍的運算量，準確率卻絲毫沒有改善，於是，我們最終將參數設定距離為 1，角度則為 90°。

我們萃取影像的紋理特徵的第一個步驟是將彩色影像轉換成灰階影像，我們使用的轉換方程式如公式(2)所示[2]，轉換後的灰階值將介於 0~255 之間，但如果以此灰階值計算灰階共現矩陣，我們將得到 256x256 大小的矩陣，這樣將造成過大的計算成本，而且，這樣將過份注重紋理的細節，除了不符合人類的視覺感知系統外，也未必能得到更精確的效果，所以我們將 0~255 的灰階值量化為 0~15(16 階)，再依設定的距離(1)和角度(90°)來取得一個 16x16 大小的灰階共現矩陣。最後，我們直接將灰階共現矩陣中共 256 個值視為一個 256 維的特徵向量，此特徵向量則代表一張影像的紋理特徵。

$$\text{grey}=0.29 \times R+0.587 \times G+0.114 \times B \quad (2)$$

2.3.2 灰階共現矩陣的距離計算

在紋理特徵距離測量中，我們選擇如論文[7]所提到的距離算法，如公式(3)所示。我們萃取 query image 的紋理特徵，得到一個 M 維度(我們設定 M=256)的特徵向量 $Q_{\text{texture}}=[Q_{t1}, Q_{t2}, \dots, Q_{tM}]$ ，同時，資料庫的每一個影像的紋理特徵被萃取出來，得到一個 M 維度的特徵向量 $T_{\text{texture}}=[T_{t1}, T_{t2}, \dots, T_{tM}]$ 。這兩個向量間的距離計算方式，如公式(3)所列：

$$D_{\text{texture}}(Q_{\text{texture}}, T_{\text{texture}}) = \sum_{i=1}^M \left| \frac{Q_{ti} - T_{ti}}{Q_{ti} + T_{ti} + \alpha} \right| \quad (3)$$

公式(3)中， α 為一個極小的數值，是為了避免分母為 0 所設定的。雖然這個距離計算法不像 L_1 距離計算法那樣簡單，但這個算法仍然避免了平方或是平方根的運算，也能很有效的降低計算成本，且能取得不錯的準確率。

2.4 重新測距的方法(RRK)

在這個小節，我們將介紹我們設計重新測距方式的概念，並將此方式導入到本論文建置的影像檢索系統。

2.4.1 特徵距離的正規化

由於本論文設計的 CBIR 系統萃取了顏色及紋理兩個特徵來進行影像間的距離比對，同時這兩種特徵分別有不同的距離比對方式，在結合兩種距離時，為了避免兩種距離的值域不一所造成的比重錯誤，我們必須先將顏色及紋理距離進行正規化，正規化的程序如公式(4)(5)所示：query image 和資料庫中每一張影像之間的距離將一律除以 query image 和資料庫中某張影像的最大距離，如此一來，新的距離值域將介於 0~1 之間，顏色、紋理皆是如此。最後，query image 和資料庫中每一張影像之間的距離則如公式(6)所示，其中 W_1 、 W_2 為權重 ($W_1+W_2=1$)，根據後來的實驗，我們設定 $W_1=0.6$ ， $W_2=0.4$ ，此設定能夠取得最佳的結果。

$$D'_{\text{color}}(\text{query}, \text{dataset}_i) = \frac{D_{\text{color}}(\text{query}, \text{dataset}_i)}{\max(D_{\text{color}}(\text{query}, \text{dataset}_i))} \quad (4)$$

$$D'_{\text{texture}}(\text{query}, \text{dataset}_i) = \frac{D_{\text{texture}}(\text{query}, \text{dataset}_i)}{\max(D_{\text{texture}}(\text{query}, \text{dataset}_i))} \quad (5)$$

$$D(\text{query}, \text{dataset}_i) = W_1 \times D'_{\text{color}}(\text{query}, \text{dataset}_i) + W_2 \times D'_{\text{texture}}(\text{query}, \text{dataset}_i) \quad (6)$$

2.4.2 重新測距

跟其他的 CBIR 系統相同，本系統在輸入一張 query image 之後，會從資料庫檢索出最接近於 query image 的 k 張影像，接下來我們將從這 k 張影像中提取出參照特徵值(reference feature)。

首先要說明的是本論文重新測距的概念。一般的 CBIR 系統，皆是輸入一張 query image，希望透過設計好的特徵萃取比對方式，得到和 query image 最相近的影像。這個方式的距離比對，通常是一對一的，只考慮到 query image 和資料庫中某影像的直接距離，並未考慮到資料庫中其他的影像，這樣的方式可能造成某張影像和 query image 不存在著相當接近的直接距離，但此影像卻和 query image 的鄰居(和 query image 極近似的影像)有著極接近的距離，此影像理應接近於 query image，卻因直接距離過大而遭到排除或有著較差的排序。在文獻 [9][10] 中，他們考量了資料庫中所有影像兩兩之間的距離，藉此產生新的距離，取代 query image 和資料庫中每一張影像的直接距離，認為這樣的距離是更為適當的。

本論文根據 [9][10] 中的精神，決定 RRK 的概念：一般的 CBIR 系統檢索和 query image 同類的影像，將其列舉出來；本論文的方法則先透過檢索系統確

立了 query image 的「類別」，再檢索此「類別」的影像，得到最後的結果。這樣的方式有著[9][10]的精神，卻簡化了其中相當繁複的計算。

RRK 的步驟如圖 2.6 所示。步驟 1~3 跟一般的 CBIR 系統沒有兩樣，而在絕大部份的系統中，檢索出來的影像，其準確率與排序順位成反比，例如：排序順位的第一張，往往都和 query image 是一張，毫無疑問的命中檢索條件；而第二順位的影像，則往往有著 90% 以上的準確率；而隨著順位次序的增加，第二十張順位的影像，可能只剩下 50% 的準確率。基於這樣的特性，本論文假設檢索結果的前 k 張有著極高的準確率，意即此 k 張影像有很高的機率是和 query image 屬於同一類的影像，換個角度來說，我們可以說 query image 就屬於此 k 張影像這個類別，既然我們已經知道 query image 屬於某個類別，那我們就再將這個類別的影像檢索出來即可。步驟 4 即為本論文檢索該類別影像的方法，前 k 張影像的平均特徵值將被設定為參照特徵值 (reference feature)。參照特徵值的產生將根據公式 (7)(8) 來取得。根據本論文後來的實驗，我們將 k 值設定為 1~10 分別測試，最後我們設定 $k=6$ ，因為此時可以得到最佳的實驗結果。步驟 5 的進行方式可以用公式 (9) 來表示。新的距離將以參照特徵值取代 query image 特徵值來獲得，值得一提的是，這樣的新距離有時會造成和 query image 最近似的影像卻不是 query image 本身。因此，重新排序時，並不會更改原始排序的第一順位，會由第二順位重新排序，以免造成不合理的結果。在步驟 5 的處理之後，理論上，本系統將得到更佳的檢索結果，而根據 RRK 的精神，我們可以一次又一次的重覆進行步驟 5、6，取得一次又一次更好的結果。

$$\text{reference feature}_{\text{color}} = \frac{1}{k} \sum_{i=1}^{i=k} \text{feature}_{\text{color}}(\text{IMG}_{i\text{th}}) \quad (7)$$

$$\text{reference feature}_{\text{texture}} = \frac{1}{k} \sum_{i=1}^{i=k} \text{feature}_{\text{texture}}(\text{IMG}_{i\text{th}}) \quad (8)$$

$$\text{new } D(\text{query}, \text{dataset}_i) = D(\text{reference feature}, \text{dataset}_i) \quad (9)$$

Cluster s	Semantic name	A sample of the images in the each cluster
1	African people village	
2	Beach	
3	Building	
4	Busses	
5	Dinosaurs	
6	Elephants	
7	Flowers	
8	Horses	
9	Mountains and Glaciers	
10	Food	

圖 3.1 Wang 圖片資料庫種類及範例

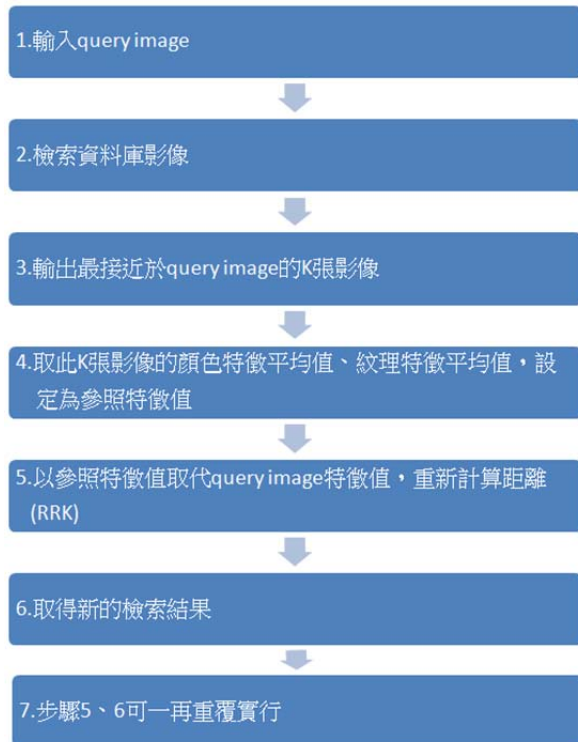


圖 2.6 重新測距(RRK)流程圖

3. 實驗結果與分析

3.1 實驗環境及資料庫

首先介紹本論文用於實驗的機器設備，我們使用一般常見的 PC，硬體配備：Intel i5 2.5GHz 之 CPU，容量為 8G 的 RAM，並搭配 Windows 7 的作業系統，撰寫程式的環境為 Adobe Flash Builder v4.5。本次實驗所用之資料庫為 WANG database，來源為 <http://wang.ist.psu.edu/docs/related/>。這個影像資料庫常常被用來進行 CBIR 系統實驗 [8][11][12][13][14]，其影像總數為 1000 張，共分為 10 個種類，如圖 3.1 所示。

3.2 實驗的評估標準

相似於多數的 CBIR 系統，本實驗以常用的準確率 (precision) 和召回率 (recall) 來判定系統效能。

準確率 (precision) 意即當檢索系統返回的影像與 query image 屬於同一類時，系統即判定為準確命中，我們設定系統返回影像張數為 k ，此 k 張圖片中準確命中的張數相對於 k 的比率即為準確率，如公式 (10) 所示。與我們將比較的其他論文一樣，我們設定 $k=20$ 。

$$\text{precision} = \frac{\text{the number of matched images}}{\text{the number of returned images } k} \quad (10)$$

召回率 (recall) 意即返回影像中正確命中的張數與該類別的影像總數之比率，如公式 (11) 所示。

$$\text{recall} = \frac{\text{the number of matched images}}{\text{the number of all relevant images in the database}} \quad (11)$$

本實驗將資料庫中每一張影像皆當作 query image 輸入系統並取得檢索影像，計算其準確率，最後再分別計算每個類別的平均準確率及整體資

料庫平均準確率。召回率方面，則觀察返回影像張數不一時的平均召回率。

3.3 實驗結果與討論

3.3.1 原始的 CBIR 系統(未使用 RRK)

為了將本論文提出的方法產生的結果和其他學者們提出的方法做比較，每一個方法，我們都有一樣的設定，即資料庫的每一張影像都當作 query image 被輸入 CBIR 系統，返回的前 k 張影像用來判定準確率($k=20$)，一共是 1000 張 query image，最後得到的比較結果如表 3.1 所示。

根據表 3.1 所示，本文提出的原始 CBIR 系統(未使用 RRK)，雖然只是簡單的萃取影像的顏色及紋理特徵，但仍以 22% 以上的差距大幅勝過[11][12] 兩篇論文；相較於約 10 年前發表的[11][12]，[8][13][14]皆為近 4 年來的論文，本文架構簡單的 CBIR 系統仍然在準確率上有 2.18%~5.78% 左右的提升。我們同時注意到，本文的方法在 10 個類別內，有 6 個類別都擁有相較於其他 5 篇論文之中的最高準確率(以紫色數據表示)，令人遺憾的是，因為第 2 類海灘及第 9 類高山有著許多藍天白雲等極相似的背景，他們存在著許多近似的顏色及紋理特徵，導致本文的方法在這些類別無法取得令人滿意的準確率。

表 3.1 本論文的方法和其他論文方法之比較

Average precision							
method/class	Jhanwar et al.[11]	Hung et al.[12]	OTDQRS method[15]	Lin et al.[8]	M.E.EAlami [14]	Proposed method (without re-rank)	Proposed method (with re-rank)
Africa	0.4525	0.4240	0.5620	0.6830	0.7030	0.8645	0.9100
Beaches	0.3975	0.4455	0.5360	0.5400	0.5610	0.4285	0.3665
Building	0.3735	0.4105	0.6100	0.5615	0.5710	0.6955	0.7885
Bus	0.7410	0.8515	0.8930	0.8880	0.8760	0.8800	0.9360
Dinosaur	0.9145	0.5865	0.9840	0.9925	0.9870	0.9915	0.9990
Elephant	0.3040	0.4255	0.5780	0.6580	0.6750	0.6375	0.6170
Flower	0.8515	0.8975	0.8990	0.8910	0.9140	0.9410	0.9430
Horses	0.5680	0.5890	0.7800	0.8025	0.8340	0.9575	0.9895
Mountain	0.2925	0.2680	0.5120	0.5215	0.5360	0.4050	0.3350
Food	0.3695	0.4265	0.6840	0.7325	0.7410	0.8150	0.8750
Average	0.5265	0.5325	0.7038	0.7271	0.7398	0.7616	0.7760

圖 3.2 為本系統實作的範例。如圖 3.2.(a)所示，圖片的最左上方影像為 query image，檢索結果返回的 20 張影像列於下面，故其檢索準確率為(100%)，這也說明了第 8 類影像在本文的檢索系統中有著相當高的準確率(0.9575)。圖 3.2.(b)為第 9 類高山影像的範例，準確率只達到 0.3(30%)。

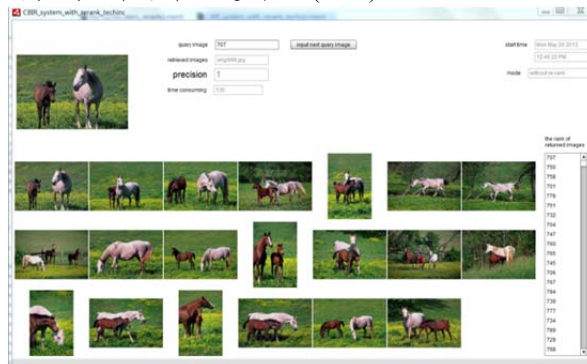


圖 3.2(a) 第 8 類馬匹影像的檢索範例

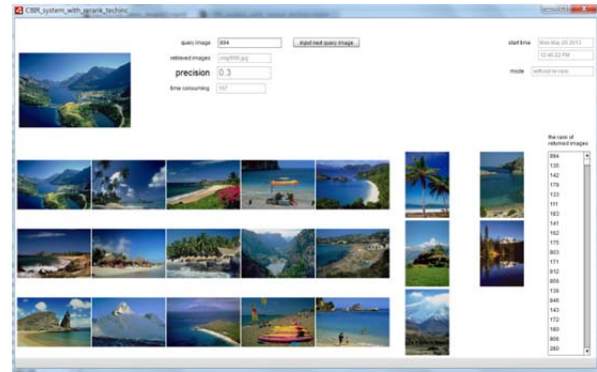


圖 3.2(b) 第 9 類高山影像的檢索範例

3.3.2 重新測距的效果

另一方面，從表 3.1 得知，本文提出的 RRK 方法證實是有效的。在重新測距並排序之後，本文的系統整體平均準確率從 0.7616 提升到 0.7760，有了 0.0144(1.44%)的提升，而紅色的數據部分代表了兩個含意，其一：重新測距後的準確率，紅色數據是所有方法中單類達到最高準確率的，一共有 7 類。其二：紅色數據也代表重新測距後，準確率得到提升的類別。7 個類別的提升說明了本文重新測距的概念是有效的，但是沒有得到提升的 3 個類別準確率卻下降了，這表示本文重新測距的方法存在著一些限制。

本文的方法是假定原始 CBIR 系統檢索出的前幾張相似影像有著極高的命中準確率，因此我們根據前幾張影像來判定 query image 的類別，然後再根據這類別去檢索同類別的影像，而本實驗的海灘及高山這 2 個類別，因為原來的系統只能取得約 4 成的準確率，換句話說，其不準確率高達 6 成，所以 RRK 一開始就將 query image 辨識為錯誤的類別，也難怪經過 RRK 之後，準確率反而下降了。類別 6 大象影像原始的準確率為 0.6375，則屬於模擬兩可的狀況，RRK 後的結果，準確率則得到小幅度的下降，這是可以理解的。

雖然重新測距的方法存在著缺點，但這個方法仍是具有貢獻的。如果將準確率下降的 3 個類別忽略的話，其他 7 個類別的準確率則將從平均 0.8779(87.79%)提升至 0.9201(92.01%)，提升了 4.23%。此即說明，此方法不但容易理解且易於實作，而且是一個能夠有效提升準確率的好方法。

3.3.3 重覆實行 RRK 的效果

根據 RRK 的精神，本系統預期重覆實行 RRK 將能夠進一步提升 CBIR 系統的準確率，實驗中，我們實行 RRK 的次數為 1 至 5 次，實驗的結果如表 3.2 所示。

在表 3.2 中，準確率因多一次實行 RRK 而增加或持平時，數據以紅色表示。紅色數據的分布說明了資料庫內有七個類別的影像準確率因為重覆進行 RRK 而幾乎每一次都有所提升；如果原始的準確率高於 7 成，在每一次的 RRK 後大致上都能取得更佳的結果；但同時地，原始準確率若低於 6 成 2，在每一次的 RRK 後，準確率則漸漸下降，這樣的原因則如上一個小節所提及，在此不再贅述。

RRK 雖然有效，但若原始準確率不佳，則 RRK 將產生反效果。

表 3.2 RRK 重覆實行後準確率的變化

method/class	proposed method	proposed method +RRK 1 time	proposed method +RRK 2 times	proposed method +RRK 3 times	proposed method +RRK 4 times	proposed method +RRK 5 times
Africa	0.8645	0.9100	0.9145	0.9125	0.9125	0.9125
Beaches	0.4285	0.3665	0.3195	0.2970	0.2920	0.2895
Building	0.6955	0.7885	0.8245	0.8385	0.8415	0.8415
Bus	0.8800	0.9360	0.9485	0.9510	0.9510	0.9510
Dinosaur	0.9915	0.9990	0.9995	1.0000	1.0000	1.0000
Elephant	0.6375	0.6170	0.6040	0.5970	0.5980	0.5960
Flower	0.9410	0.9430	0.9470	0.9500	0.9535	0.9530
Horses	0.9575	0.9895	0.9945	0.9945	0.9945	0.9945
Mountain	0.4050	0.3350	0.2980	0.2815	0.2770	0.2775
Food	0.8150	0.8750	0.8830	0.8855	0.8825	0.8845
Average	0.7616	0.7760	0.7733	0.7708	0.7703	0.7700

3.3.4 系統穩定性

圖 3.3 為本論文提出的系統與其他五個系統的平均準確率、平均召回率比較結果，檢索的返回影像數量設定為 20、30、40、50、60、70、80、90、100。從圖 3.5 中可以發現，本文提出的系統無論有沒有實行 RRK，檢索效果都是六種方法中最優的。

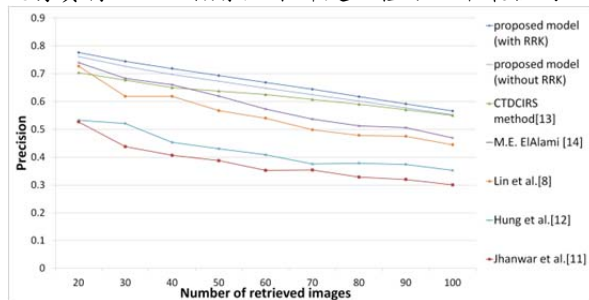


圖 3.3(a) 返回影像數量不同時，六種方法的平均準確率

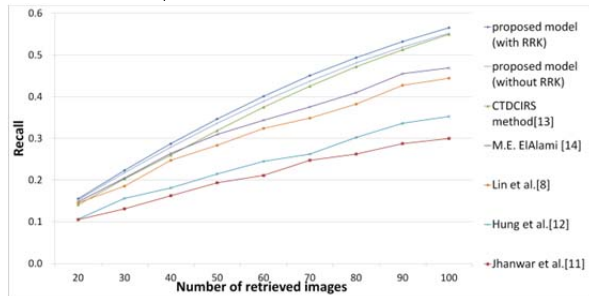


圖 3.3(b) 返回影像數量不同時，六種方法的平均召回率

3.3.5 RRK 的廣泛應用性

由於 RRK 的概念容易理解，實作上也並不複雜，我們大膽假設：凡是 CBIR 系統中，影像中被提取的特徵值是能夠加總平均的，皆能適用此方法。唯一該考量的地方是原始 CBIR 系統的檢索準確率，如果原始的平均準確率不能超過 7 成，或者原始系統不能有效處理資料庫中的各種影像種類，導致檢索準確率不穩定，都將導致 RRK 的效果低落，或甚至造成反效果。

4. 結論

本論文首先設計的一個簡單而有效的 CBIR 系統，此系統萃取影像的顏色及紋理資訊，產生 HSV 色彩空間直方圖及灰階共現矩陣，並轉化為特徵向量，藉此特徵向量進行影像比對。實驗結果顯示，本論文的 CBIR 系統，和其他五篇論文提出的方法相比，準確率有著 2% 至 22% 的提升。

我們另外提出了一個名為重新測距(RRK)的距離測量方法。這個方法改善了特徵間距離的比對方式。實驗結果證明了 RRK 能夠有效的提升原始 CBIR 系統的檢索準確率，而在 RRK 重覆實行的狀況下，準確率還能進一步提升。

值得一提的是，RRK 的設計原理相當簡單，我們認為在所有的 CBIR 系統中，只要萃取出來的影像特徵值是能夠被加總平均的，皆能套用 RRK 來取得效率的提升。

參考文獻

- [1] Liu, G.-H., Li, Z.-Y., Zhang, L. and Xu, Y., *Image retrieval based on micro-structure descriptor*. Pattern Recognition, 2011. **44**(9): p. 2123-2133.
- [2] Yue, J., Li, Z., Liu, L. and Fu, Z., *Content-based image retrieval using color and texture fused features*. Mathematical and Computer Modelling, 2011. **54**(3-4): p. 1121-1127.
- [3] Wang, X.-Y., Yu, Y.-J. and Yang, H.-Y., *An effective image retrieval scheme using color, texture and shape features*. Computer Standards & Interfaces, 2011. **33**(1): p. 59-68.
- [4] Singha, M. and Hemachandran, K., *Content Based Image Retrieval using Color and Texture*. Signal & Image Processing, 2012. **3**(1).
- [5] Smith, J.R. and Chang, S., *Transform Features for Texture Classification and Discrimination in Large Image Databases*. Proceeding, in *IEEE International Conference on Image Processing*. 1994. p. 407-411.
- [6] Liu, G.-H. and Yang, J.-Y., *Content-based image retrieval using color difference histogram*. Pattern Recognition, 2013. **46**(1): p. 188-198.
- [7] Wang, X.-y., Chen, Z.-f. and Yun, J.-j., *An effective method for color image retrieval based on texture*. Computer Standards & Interfaces, 2012. **34**(1): p. 31-35.
- [8] Lin, C.-H., Chen, R.-T. and Chan, Y.-K., *A smart content-based image retrieval system based on color and texture feature*. Image and Vision Computing, 2009. **27**(6): p. 658-665.
- [9] Xiang, B., *Learning Context-Sensitive Shape Similarity by Graph Transduction*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010. **32**(5): p. 861-874.
- [10] Wang, J., Li, Y., Bai, X., Zhang, Y., Wang, C. and Tang, N., *Learning context-sensitive similarity by shortest path propagation*. Pattern Recognition, 2011. **44**(10-11): p. 2367-2374.
- [11] Jhanwar, N., Chaudhuri, S., Seetharaman, G. and Zavidovique, B., *Content based image retrieval using motif cooccurrence matrix*. Image and Vision Computing, 2004. **22**(14): p. 1211-1220.
- [12] Huang, P.W. and Dai, S.K., *Image retrieval by texture similarity*. Pattern Recognition, 2003. **36**(3): p. 665-679.
- [13] Rao, M.B., Rao, B.P. and Govardhan, A., *CTDCIRS: Content based Image Retrieval System based on Dominant Color and Texture Features*. International Journal of Computer Applications, 2011. **18**(6): p. 40-46.
- [14] ElAlami, M.E., *A novel image retrieval model based on the most relevant features*. Knowledge-Based Systems, 2011. **24**(1): p. 23-32.
- [15] Robert M. Haralick, K Shanmugam, Its'hak Dinstein (1973). *Textural Features for Image Classification*. IEEE Transactions on Systems, Man, and Cybernetics. SMC-3 (6): 610-621.