整合關鍵字與視覺特徵的反覆式影像檢索系統

A Hybrid Approach for Iterative Image Retrieval with Keywords and Visual Features

研 究 生：簡志宇　　Student：Jr-Yu Gen

指導教授：陳穎平　　Advisor：Ying-Ping Chen

國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

In

Computer Science

June 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年六月

國立交通大學
研究所碩士班

論文口試委員會審定書

本校 資訊科學與工程 研究所 簡志宇 君

所提論文:

整合關鍵字與視覺特徵的反覆式影像檢索系統

合於碩士資格水準、業經本委員會評審認可。

口試委員：孫春在 簧雲孝

許和寧 陳穎平

指導教授：陳穎平

所 長：曾文貴

中 華 民 國 九十五 年 六 月

Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
Hsinchu, Taiwan, R.O.C.

As members of the Final Examination Committee, we certify that we have read the thesis prepared by Jr-Yu Gen entitled A Hybrid Approach for Iterative Image Retrieval With Keywords and Visual Features

and recommend that it be accepted as fulfilling the thesis requirement for the Degree of Master of Science.

Committee Members:

Thesis Advisor:

Director:

Date: June, 2006

誌謝

首先要感謝我的指導教授陳穎平老師，在這段研究期間，引導我用系統化的方式，剖析一個問題、思考解決問題的方向。這些方法提供我在面對將來更多問題時，能夠用一個更有效率的方式，找出最佳的解決方案。

感謝口試委員們: 孫春在教授、洪炯宗教授、鍾雲恭教授所給與的各項指導與建議，有了你們的幫助，才讓這項研究有更佳的結果與成就。

感謝我的實驗室同袍: 明昌、長泰，有了你們的幫忙，讓我的口試和論文可以順利如期完成。

感謝我的好朋友們: 閒魚、賤王、彥志、龜欽、MarkR、噴糾、小恥、國手、kaogold3、小柯、神龍、大洋、衛斯理、謝肥、大光、台西、Zivv、Jerr、wakaw、asura、小光、緯凱、老爹、阿 C、sheviks、Dracula、maygin、雄哥、阿邦、小公主、Y8、19、小郭、Ruby、Neil、Irene、阿雅、老鼠、小黑、Lucky、Mark、Van、Sandra、小晶、Abby、佳燊、Michael、Paul、映晨、小馬。感謝你們在這段期間讓我不覺得孤獨。

最後要感謝我的家人，你們是我的原動力，希望我的一切能讓你們感到驕傲。

整合關鍵字與視覺特徵的反覆式影像檢索系統

研究生: 簡志宇　　指導教授: 陳穎平

國立交通大學 資訊科學與工程研究所

摘要

QBK 是從人對於圖片的高階語意描述出發的一種圖片搜尋系統，其優點在於以人類的語意為基礎出發，並輔以成熟的文字檢索技術。QBK 的缺點則在於圖片本身的內容對於檢索的影響可以說完全沒有，且圖片的文字描述並不能完全代表圖片本身所包含的內容。

CBIR 則是從圖片本身的視覺特徵出發的一種圖片搜尋系統，其優點在於檢索結果完全依靠圖片本身的內容為主，完全客觀。其缺點目前 CBIR 的基礎技術仍不夠成熟，無法完美的模擬人類的辨別能力。

本研究綜合 QBK 系統和 CBIR 系統的優點，整合視覺特徵與關鍵字檢索技術，提出一個較為接近人類語義且以影像內容為基礎的圖片檢索系統。

本研究將 QBK 系統的查詢結果，透過 CBIR 中視覺特徵的擷取，將擷取出來的特徵值，再以資料探勘中的分群演算法加以分群，以區分出代表不同語意的影像。最後加上關鍵字擷取的技術，以關鍵字建議引導使用者作反覆式的搜尋，找到更貼近使用者語意的搜尋目標。

關鍵字: 關鍵字式影像檢索、基於內容的影像檢索

A Hybrid Approach for Iterative Image Retrieval with Keywords and Visual Features

Student: Jr-Yu Gen Advisor: Dr. Ying-Ping Chen

Institute of Computer Science and Engineering
National Chiao Tung University

Abstract

QBK is an image search approach based on text description. The advantage of QBK is that it is based on semantics of mankind, and assisted by the matured text-based search technology. However, the disadvantage of QBK is that the result of image search is not affected by the content of the image itself. Besides, the text description does not represent the content of image fully.

CBIR is another image search approach which is based on the visual features of image itself. The advantage of CBIR is that the result of image search is all based on the content of the image and, it is objectively. The disadvantage of CBIR is that the basic technology is not matured enough. So the approach cannot imitate the recognition ability of human beings.

Our approach is the combination of QBK and CBIR which integrates the advantage of visual features and text description. This approach not only access the semantics, but also base on the content of image.

We extract the visual features with the method of CBIR from the result images of the QBK system. And then the images will be clustered by their visual features. Finally, users can iteratively search with keyword suggestions which are extracted from the description of clustered images.

Keywords: QBK, CBIR

目錄

圖目錄

表目錄

第一章 導論

1.1 研究背景

隨著網路普及與資訊科技的進步，多媒體內容的儲存以及如何搜尋，是一個值得深究的熱門話題。也由於影像資料的快速增加，如何有效率將影像組織化，透過適當的索引，以協助使用者從茫茫大海的影像資料庫中，找到想要的影像，這一類的應用與技術，也如雨後春筍般出現[1-6]。

由於傳統的關鍵字檢索技術已非常成熟，利用圖片的註解與描述文字搭配關鍵字檢索技術的圖片搜尋系統(Query by Keyword、簡稱QBK)，已成為圖片搜尋的商業應用主流。如 Google Image Search[7]。如圖 1所示，QBK系統以先針對圖片資料庫的文字描述部份，利用現行的關鍵字檢索技術(Indexing)，製成可供快速比對(matching)的indexed description資料庫。其後使用者輸入關鍵字，便可和indexed description作比對，並找出相關的項目並顯示。從QBK的流程中，我們不難發現，對整個系統來說，人為的描述才是整個系統檢索的主體，圖片本身的內容並不會影響系統的運作。因此我們可以說，對一個Image QBK系統而言，圖片僅是一個表現層的物件，和系統邏輯層完全無關。

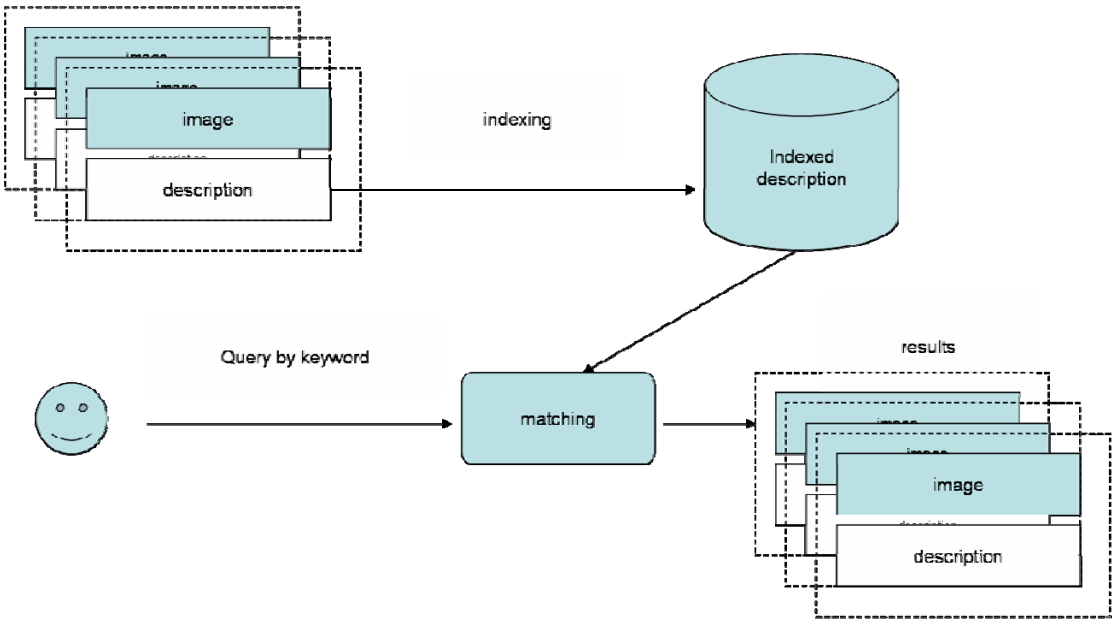

圖 1 QBK system

QBK 的圖片搜尋系統並沒有完全滿足人類對圖片搜尋的深度需求。其原因在於人類對於圖型的認知是經過長時間視覺經驗累積而成的主觀意識。同一張圖片對於不同的人，可能就會產生不同的認知標準，因此圖片的文字描述並不能完全代表圖片本身所包含的內容。就算為圖片文字描述制定單一標準，面對來自網路的大量圖片資料庫，也需要透過大量的人力才有辦法完成。更甚者，有些圖型檢索之應用不管花再多人力也無法以文字描述來達成，如: 指紋辨認、人臉辨認等。對於人類最原始的需求"找想要的圖片"，QBK system 只能說是一個可用的系統，我們無法稱之為完整的解決方案。

基於傳統QBK圖片搜尋系統的這些弱點，以圖片內容為基礎的圖片檢索(Content-based Information Retrieval, CBIR)[1, 5, 6, 8-10]，因而成為新一代影像檢索技術的焦點。如圖 2所示，在一個CBIR的系統中，首先將大量的圖片，透過視覺特徵擷取技術(Visual Features Extraction)，擷取出以顏色(Color)、形狀(Shape)、紋理(Texture)三大

視覺特徵方向為主的各項視覺特徵值，再將之以數位化的方式儲存於特徵值資料庫中，以供使用者檢索與搜尋。使用者可透過以圖片範例(Query by Example)或對特徵值的人為定義等描述圖型本身內容的方式來提出查詢(Query by Content)，系統將查詢轉化成各項視覺特徵值後，將之與視覺特徵資料庫作比對，並回報給使用者搜尋結果。

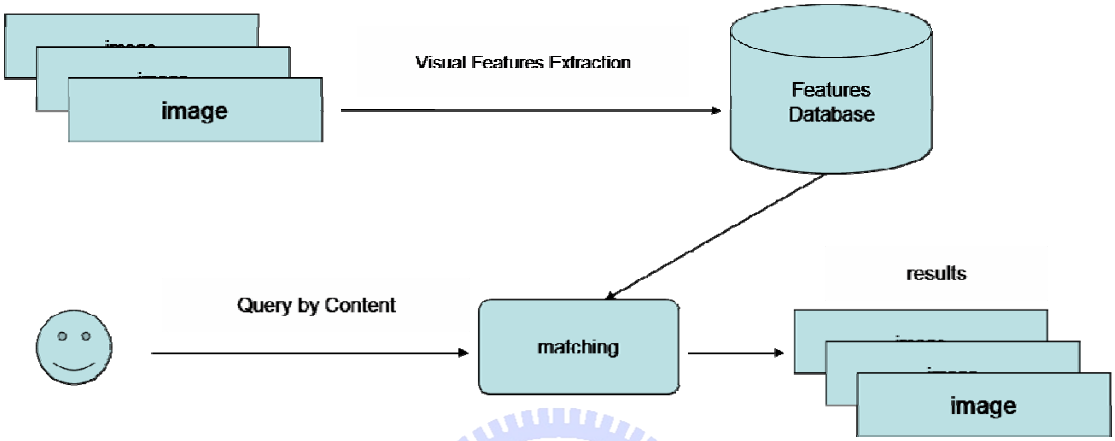

圖 2 CBIR System

為了要更接近人性化的需求，許多透過圖片低階特徵值的組成來描述更接近人類語意表達(Semantics) 的各項相關研究[11]，和更多不同的低階視覺特徵描述方式，如: 記錄顏色與空間分佈的關係的顏色空間特徵值(Color Layout)、降低儲存空間與計算複雜度的各式特徵值擷取演算法[12-15]，也相繼被發表。及應用在影像或影片的註解或搜尋系統中。如何把影像對應到正確的語意表達以及做出有效率的索引，是 CBIR 技術的兩大目標。

1.2 研究動機

由 1.1 的研究背景中，我們可以發現，利用圖片的註解與描述文字搭配關鍵字的檢索技術的圖片搜尋系統(QBK)，雖然是目前圖片搜尋的商業應用主流。但傳統的QBK系統仍無法滿足人類搜尋圖片的深度需求。因而衍生出CBIR等直接擷取圖片低階特徵值的相關研究，並透過圖片低階特徵值的組成來描述更接近人類語意表達(Semantics)。然語意表達層次較直接擷取低階特徵值更為複雜許多，一張圖片可以表現的意義，從不同的角度、內含的物體、不同的人來解讀，有各種不同的文字描述方式，而對一個CBIR系統來說，同一張圖片永遠只有一種數據描述。以圖 3為例，可能有的人會以"小鳥"來描述這張圖片，而有的人會以"藝術"來描述，也有可能以"去台北旅行看到的街景"來描述這張圖片的意義。可是對同一個CBIR的演算法來說，這張圖片永遠只會有一種數據資料。

圖 3 一張圖片的意義可代表不同的語意

以現行的CBIR演算法來說，主要分成三種層次，第一層次是完全專注於最低階的視覺特徵擷取與編碼技術，可區分為三大方向: 顏色(Color)，紋理(Texture)，形狀(Shape)。雖然第一層次的擷取可以完全依賴圖片本身的raw data處理。但其實還有其待改善的空間。如圖 4所

示的十二張圖片的Color Histogram都是類似的，但其代表的意義差異卻非常大。又如圖 5所示的二張圖片，其語意上所代表的意義非常相近，只是圖型被旋轉了。但對CBIR的形狀描述子(Shape Descriptor)來說，這兩張圖片卻無任何的相關性。諸如此類第一層次的問題研究與解決方案也不斷被提出，但為了能夠更完整的接近人類語意表達，也有人提出結合各項第一層次的視覺特徵的第二層次演算法。第二層次的演算法結合了Image segmentation[16, 17]的概念，並試圖辨示出圖片中所包含的物件內容、背景(如圖 6所示)與物件的空間關係(如圖7所示)。

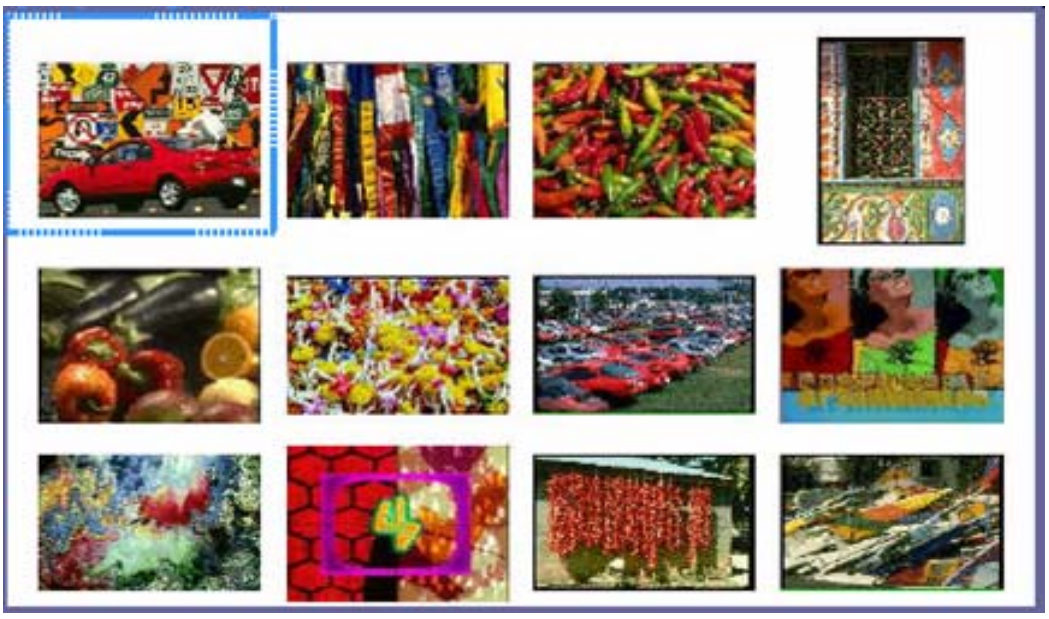

圖 4 12 張擁有類似 Color Histogram 的影像

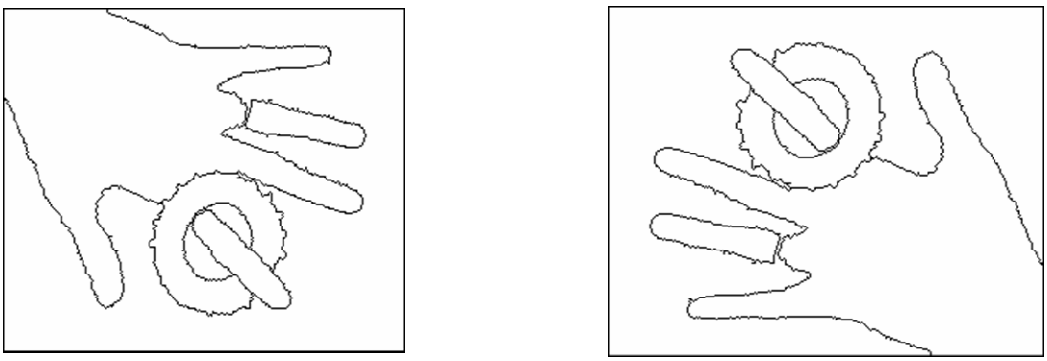

圖 5 相同形狀但不同方向的兩張影像

圖 6 影像中的物件內容

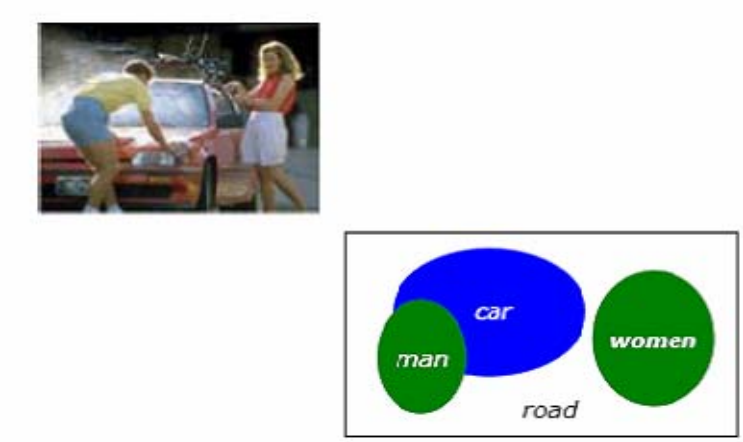

圖 7 影像中的物品與空間關係

雖然第二層次的圖片分析已讓CBIR更接近人類語意的表達，但目前第二層次的CBIR系統仍侷限於某一特定應用之範圍內[18-22]，如:檢查圖中是否含有裸女、公車、獅子、找人臉、找指紋等圖片中包含特定物件。仍沒有一個可供廣泛類型圖片的通用型CBIR系統。其主要原因在於，對於不限制類型的物件分割，我們仍找不到一個通用的最佳的解決方案，正如同圖 8所示，很多語意上的單一物件，電腦的分割結果與人類的辨認結果並不能保證是完全一樣的。

圖 8 影像分割與人類語意認定並不是完全一樣

我們可以將之視為是一種樣式辨認問題(pattern recognition problem)，對人類來說樣式辨認能力是與生俱來、潛移默化、經過長時間的經驗歸納累積而成的經驗法則。雖然電腦處理資料的量及速度遠遠超過人類，但對於樣式辨認，如: 圖片中包含那幾種物件(objects)，何為背景、同一物件的不同角度、不同顏色的變化，電腦系統不容易感受到這樣子變化所代表的意義，進而呈現在人類面前。也因此對於綜合各項第二層次結果的第三層次 CBIR 研究與應用，如: 快樂的舞會照片、沉悶的考試照片，更是少之又少。

1.3 問題描述

綜合以上 1.2 與 1.3 節各項討論，我們可以了解 QBK 主要是從人對於圖片的高階語意描述出發的一種圖片搜尋系統，其優點在於以人類的語意為基礎出發，並輔以發展成熟的文字檢索技術，而成為大型或商業圖片檢索應用主流。QBK 的缺點則在於圖片本身的內容對於檢索的影響可以說完全沒有，而且人類對於圖型的認知是經過長時間視覺經驗累積而成的主觀意識，因此圖片的文字描述並不能完全代表圖片本身所包含的內容。就算為圖片文字描述制定單一標準，面對來自網路的大量圖片資料庫，也需要透過大量的人力才有辦法完成。更甚者，有些圖型檢索之應用不管花再多人力也無法以文字描述來達成，如: 指紋辨認、人臉辨認等。

CBIR 則是從圖片本身的視覺特徵出發的一種圖片搜尋系統，其優點在於檢索結果完全依靠圖片本身的內容為主，完全客觀。其缺點目前 CBIR 的基礎技術仍不夠成熟，無法完美的模擬人類的辨別能力，進而達到滿足人類的需求。這兩大類的系統各有其現階段的優缺

點，所以本研究的主要目的，即是希望能綜合 QBK 系統和 CBIR 系統的優點，提出一個整合視覺特徵與關鍵字的圖片檢索系統，希望高低階演算法的組合，提出一個較為接近人類語義且以影像內容為基礎的圖片檢索系統。

1.4 研究方法

由於文字檢索技術的成熟，本論文將以一個完整的 QBK 系統: Google Inc.所提供的 Google Image Search[7]為出發點。使用者輸入關鍵字後，經過文字檢索比對，取得多張於文字描述中帶有關鍵字之圖片。有了這些圖片與相對應的文字描述，我們再用選擇並改良現有的各項低階視覺特徵值擷取演算法，取得每張圖片的低階特徵值並正規化。在視覺特徵與關鍵字之間，我們採用資料探勘(Data mining)[23]的分群技術(Clustering)作連結。我們會建立一個分群演算法，將每張圖片依低階特徵座標作分群。在分群結果產生後，我們會整合資訊擷取(Information Retrieval)的技術[24]，從分群中擷取能代表該群的關鍵字。最後我們將以上各項工作結果作流程整合，提供一個讓使用者可以藉由組織化分群瀏覽搜尋結果，並由系統提供進一步查詢的建議關鍵字(Keyword Suggestion)以改善檢索結果的反覆式圖片檢索系統。研究步驟如下:

1. 低階視覺特徵值的演算法研究
2. 低階視覺特徵值擷取與正規化
3. 分群演算法之研究
4. 建立適合的分群演算法
5. 關鍵字擷取之研究
6. 建立適合的關鍵字擷取演算法

7. 整合流程
8. 結果驗證

1.5 論文架構

本論文結構如下: 第二章為相關文獻、工具的概要與探討。第三章為本研究的系統架構與演算法詳細論述。第四章系統雛型與結果探討。第五章為結論與未來工作。第六章為參考文獻。

第二章 文獻探討

本章將介紹與本研究相關的應用與演算法，包括以內容為基礎的影像搜尋系統、低階影像視覺特徵值擷取、影像分割、分群演算法。

2.1 以內容為基礎的影像搜尋系統

由於網路的興起，各種資料的傳遞與交換變得無遠弗屆，且因為儲存設備的進步，如數位相機、數位攝影機等的普及，帶動了多媒體資訊的快速暴增，如影像、音樂、影片等，進而需要一個好的系統來有效率地管理這些資料。在傳統的影像搜尋系統中，以查詢的方式而言，過去有許多系統是以文字來作為查詢的索引(Query By Keyword, QBK)，此類系統會根據使用者輸入的文字當作是搜尋索引的目標，進而找到相關的影像。但當使用者不知道如何下達正確的關鍵字時，或是圖片與文字之間多對多的關係，會使用查詢的結果無法滿足使用者的需求。故近年來由影像內容作為查詢基礎(Query By Content, QBC)的研究漸成為影像查詢系統研究的主流。而由影像內容作為查詢基礎的系統，可再細分為由實際影像作查詢(Query By Example)，或是可以由使用者來描繪出所想要的意涵，甚至也有進一步加入使用者習性作為參考的查詢系統。這種以影像為查詢索引的系統，便稱為以內容為基礎的影像搜尋系統。而現今比較有名的有以下幾個:

- QBIC[25](Query By Image Content，由 IBM Almaden Research Center 所開發): QBIC 是第一個商業型泛用的 CBIR 系統，因此採用的技術較為單純，以四種單獨的低階視覺特徵值為索引搜尋，

分別為 Average Color、Color Histogram、Positional Color、及 Texture。下面四張圖分別為四種不同視覺特徵的搜尋範例。

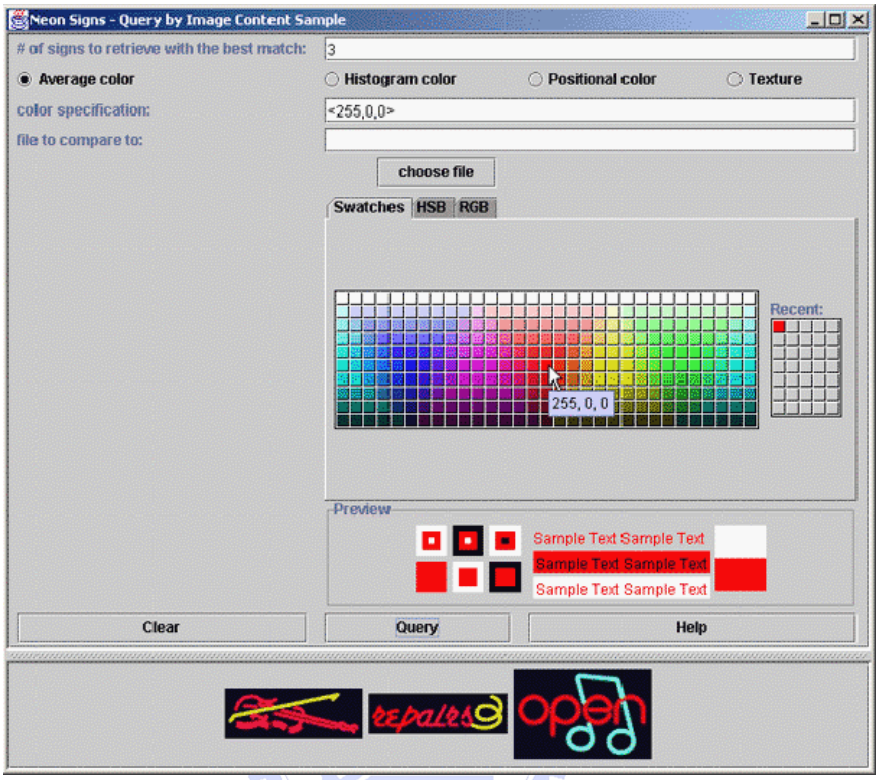

圖 9 QBIC Average Color 搜尋範例

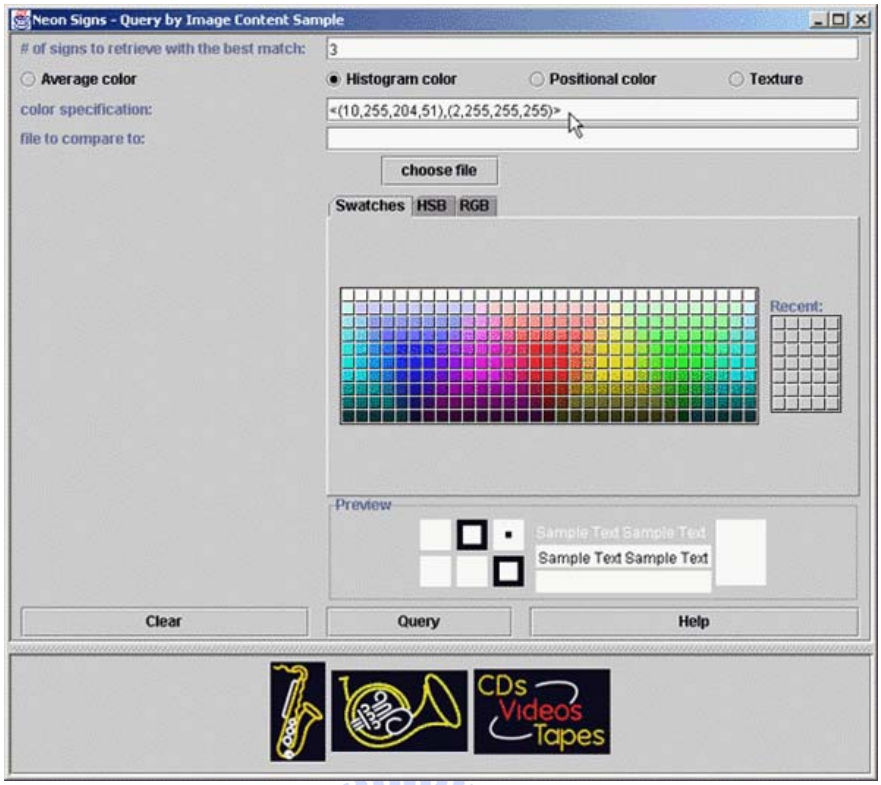

圖 10 QBIC Histogram Color 搜尋範例

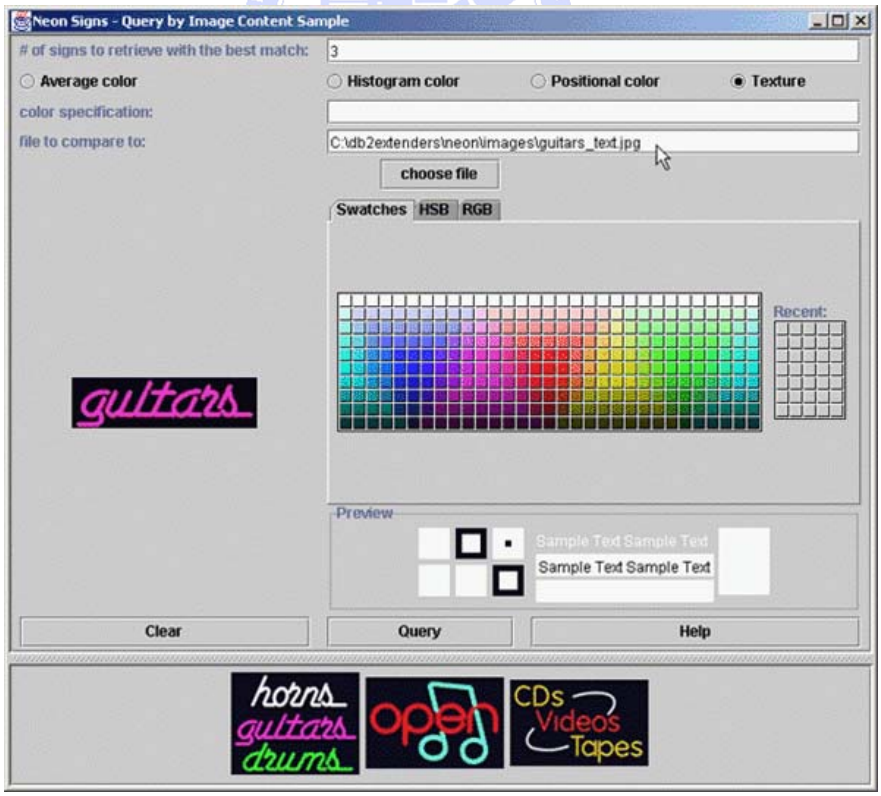

圖 11 QBIC Positional Color 搜尋範例

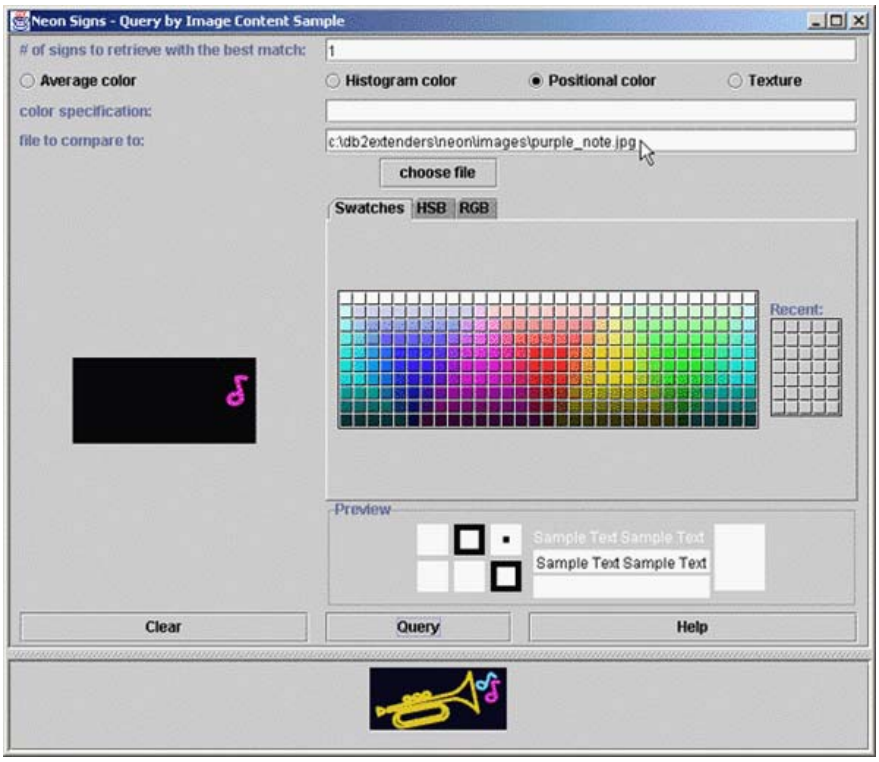

圖 12 QBIC Texture 搜尋範例

- VIR Image Engine[26](由 Virage Inc. 所開發): VIR Image Engine 是第二個商業型泛用 CBIR 系統。其主要特色在於除了提供全域的 Color, Texture, Shape 特徵搜尋，還加入了特定區域內的 Color, Texture, Shape 的特徵搜尋。相較於 QBIC 每一次搜尋只能指定以某一個特徵值作搜尋，VIR Image Engine 可指定多個特徵並設定不同比重作搜尋。而 VIR Image Engine 還加入修改特徵比重值形成 iterative query 的概念。
- VisualSeek[6](由 Department of Electrical Engineering, Columbia University 所開發): VisualSeek 與上述兩個系統最主要的差異在於: 加入了色彩的空間關係為搜尋索引。VisualSeek 的查詢是以手繪的方式來定義色塊間的空間與大小關係。查詢介面與結果範例如下二圖所示。

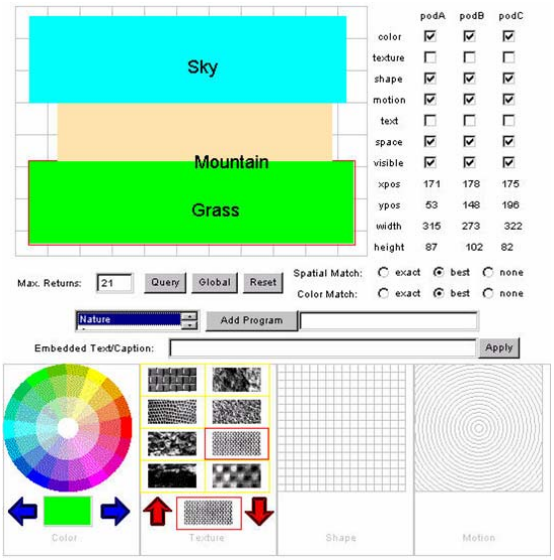

圖 13 VisualSeek 搜尋介面

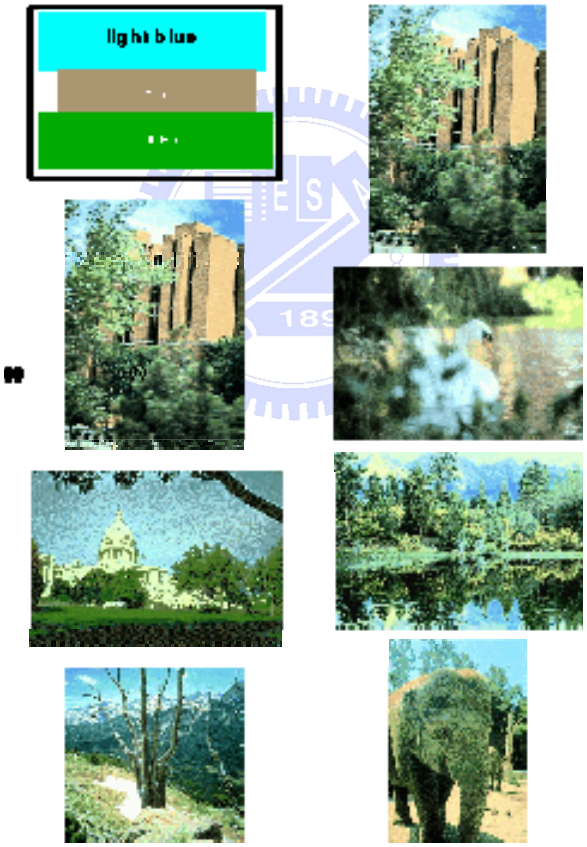

圖 14 VisualSeek 搜尋結果範例

- NeTra[4](由 Department of Electrical and Computer Engineering, University of California 所開發): NeTra 是以一個事先經過良好組織的影像資料庫 Corel photo collection 的 CBIR 系統。所有的圖片

都已事先分成 25 個分類，每個分類中包含 100 張圖片。每張圖片業已經過事先分割為多個不規則形狀的 region。使用者透過先選取一張圖片(Query by Example)為搜尋起點，接著可再選取這張圖片中的某個 region，並指定要採用這個 region 的 Color、Shape、Location、Texture 的那幾樣特徵值作為搜尋目標。Netra 因建構於一個事先經過良好組織的圖片庫，因此搜尋的結果可以達到相當高的精確度。下圖為搜尋範例。

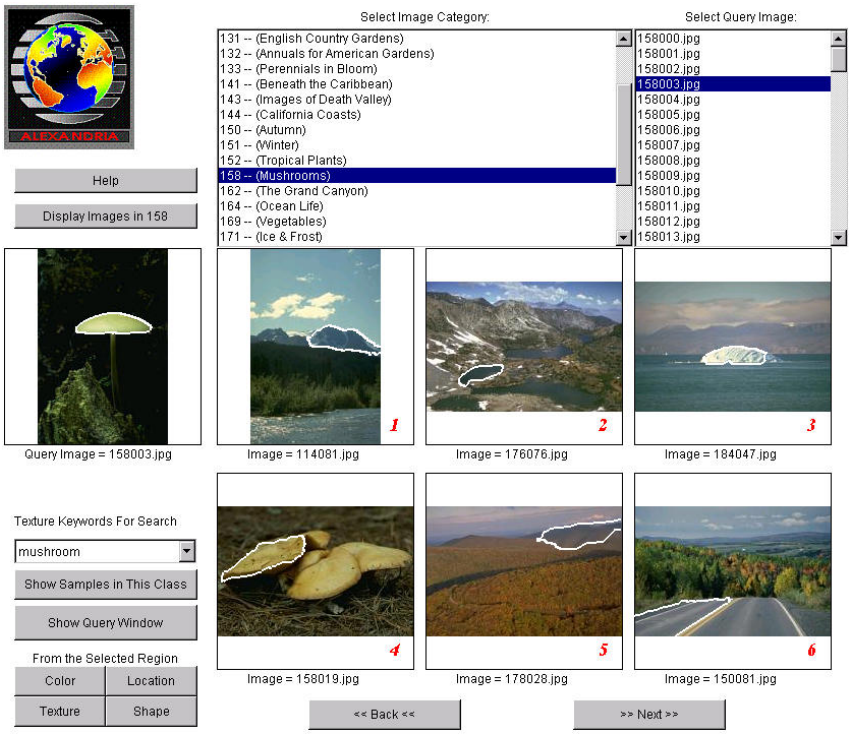

圖 15 NeTra 搜尋範例

- MARS[27](Multimedia Analysis and Retrieval System，由 Backman Institute for Advanced Science and Technology, University of Illinois 所開發): MARS 除了有兩種低階的視覺特徵值，Color 和 Texture 外，另外加入了一個手繪形狀的特徵值搜尋。下圖為搜尋範例。

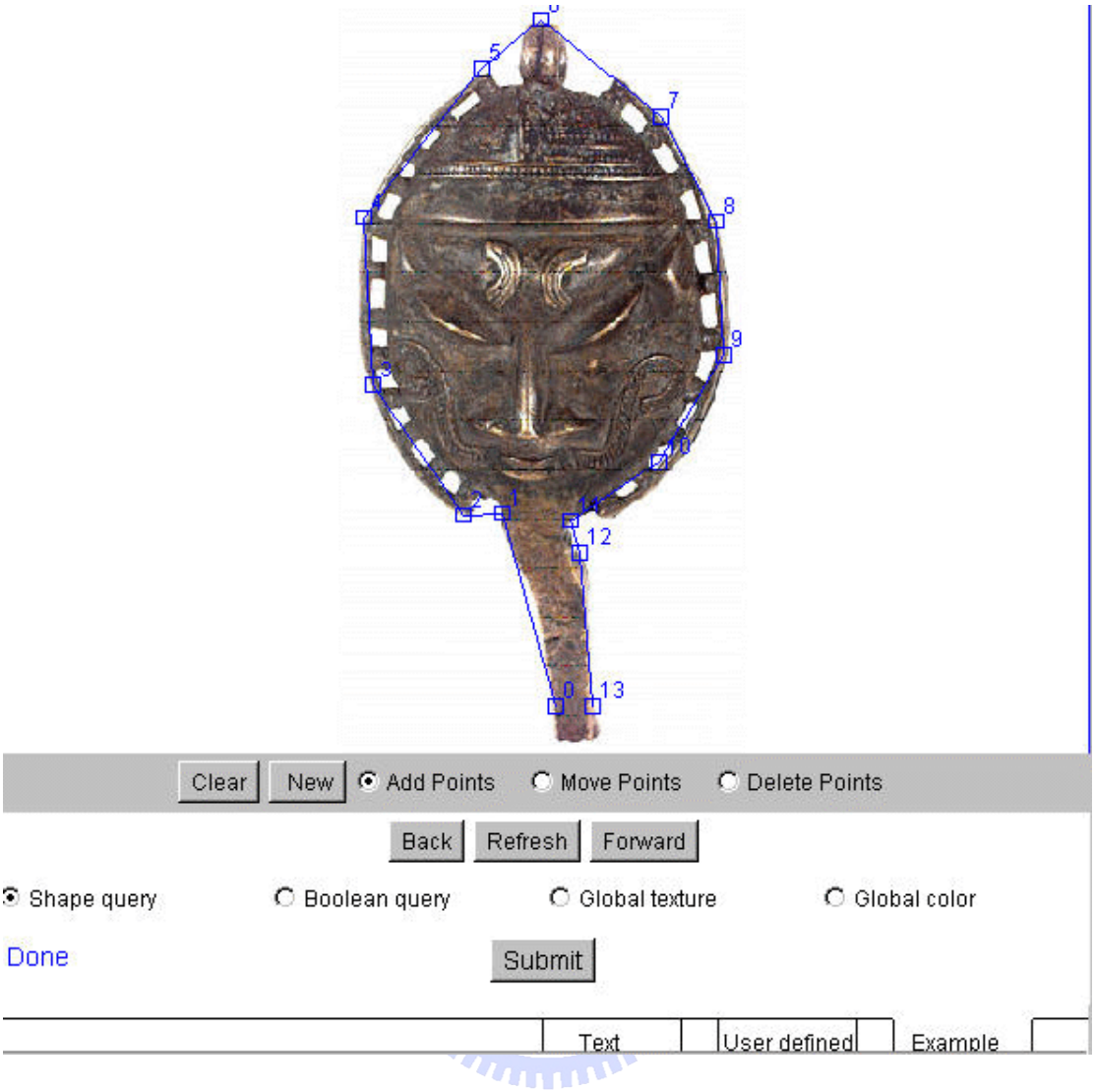

圖 16 MARS 搜尋範例

2.2 低階影像視覺特徵擷取

本節中我們將介紹本研究所應用到的低階影像視覺特徵擷取的各項技術，包含: 視覺特徵擷取的演算法、視覺特徵擷取的工具、與視覺特徵值的儲存格式。

2.2.1 低階影像視覺特徵擷取演算法

影像所能提供的低階視覺特徵有許多種，主要分為三種類形: 顏

色(Color)[12-15, 28, 29]、形狀(Shape)[30-32]、及紋理(Texture)[12, 33]。低階影像視覺特徵擷取演算法的詳細說明請參考附錄，以下就本研究使用到的視覺特徵值作一簡單的介紹:

- Color Layout[14]: 以低計算量表現出各個顏色的空間分佈狀態。首先影像會被分割為一個個 8x8 的區塊，每個區塊的 Dominant Color 以 YCbCr 顏色表示系統儲存著。每個色頻裡的 Dominant Color 再套用 DCT(Discrete Cosine Transform)，以 DCT 的係數值作為特徵值。
- Color Structure[13]: 表現出影像裡顏色的內容以及該內容的架構，可應用在矩形影像，特殊輪廓(非矩型影像)影像，以及非連接型影像(如影像為兩個不相連的區塊)之間的比對。以 8x8 畫素為一個單位視窗，在影像內滑動並且記錄下來視窗裡的顏色特性(以 double-coned HMMD 為顏色表示系統)。與 Color Histogram 不同的是，Color Structure 可以表現出畫素伴隨出現的特性(因為 Color Structure 不是以畫素為單位來記錄資訊，而是以 8x8 視窗為單位)。
- Contour Shape[32]: 利用封閉的曲線來描述 2-D 物件的輪廓，並以 Curvature Scale Space(CSS)來呈現輪廓的形狀
- Dominant Color[29]: 只要用影像中局部性的顏色特徵就可以表達整張影像的顏色訊息。
- Edge Histogram[34]: 表現空間中，五種邊(四個有向邊，一個無向邊)的分佈狀態，可以應用在非一致性的邊分佈的影像比對。將影像分割為 16 個子影像，分別計算邊(無方向性)在 0°、45°、90°、135°時的個數。
- Homogeneous Texture[12]: 表現出影像中材質紋理的特性。利

用 Gabor 過濾函式，作出紋理的走向趨勢(共 5 個)及規模(共六個)過濾器來過濾影像，在 Frequency Domain 所表現出來的第一及第二時刻能量被記錄下來成為特徵值。

- Region Shape[31]: 這個特徵值不但可以描述單一封閉區域(如圖 17左)，也可以描述有鏤空(如圖 17中)，或沒有相連的區域(如圖 17右)。

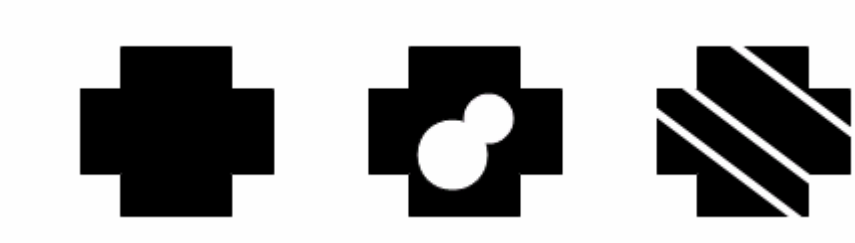

圖 17 三種 Region Shape 的影像範例

相似度比較方面，圖 18左及圖 18中是會被視為相似度比較高的，但跟圖 18右就會被視為差異性大。

圖 18 Region Shape 的相似度比較

而圖 19都會被視為相似度高。

圖 19 Region Shape 的相似度比較

這個特徵值不但使用空間小，而且在擷取和比對上都有不錯的表現。Region Shape 不但能精簡且有效率的描述多個不相

連的區域，即使在作過影像切割後，可以保留著原始影像的特性。

- Scalable Color[15]: 利用 Haar Transform，在 HSV 顏色表示系統下記錄顏色的分佈狀態。

2.2.2 低階影像視覺特徵擷取工具與儲存格式

我們採用 MPEG-7[35]的實作和標準為擷取工具和擷取結果的儲存格式。MPEG 是（Moving Picture Experts Group）動態影像專家團體的縮寫，這個團體創建於 1988 年，早期主要是為了 CD 建立視訊和音訊的標準，其中成員主要為視訊、音訊及系統領域的專家，今天我們所指的 MPEG-X 版本，是指由 ITU（International Telecommunication Union）和 ISO（International Standarization Organization）制定發佈的視訊、音訊的壓縮標準，如: MPEG-1、MPEG-2、MPEG-4、MPEG-7。MPEG-1、MPEG-2、MPEG-4 等標準，著重在影音資料的壓縮上，重點是如何達到高壓縮率，並同時兼顧一定的畫質和音質。當資料以驚人的速度的成長，直到資料多得讓人找不到時，資料也就變的沒用的東西。而 MPEG-7 標準要解決的就是隨著多媒體時代的來臨而產生繁雜的資料量，也就是如何在繁雜的資料中找到最符合自己需求的資料。MPEG-7 可獨立於其它 MPEG 標準使用，而不是取代之前的 MPEG 標準。

MPEG-7 的正式名稱為"multimedia content description interface"，其重點在於影音內容的描述和定義，以有彈性、具延伸性、多層次及明確的資料結構和語法來定義影音資料的內容，經由 MPEG-7 的定義格式，使用者可以有效率地搜尋、過濾和定義想要的影音資料。由於在不同的使用者或應用的影音內容會有不同的意義，所以在相同的媒

體上可能會有不同的影音內容的定義，如：同一隻狗在屋裏吠和在屋外吠兩段影片，在低階的特徵上有相同的音頻，但在高階的特徵上則不同，一個是在屋裏另一個是在屋外。這些高階的特徵做為與使用者互動的重要依據。MPEG-7 也使用 XML 當做陳述影音資料的語言，並以 XML Schema 當做 DDL(Description Definition Language)的基礎，使其整個架構更具有其延展性。而本研究所採用的即是 MPEG-7 中，包含 2.2.1 節所介紹的各項演算法實作與儲存標準。

2.3 影像分割演算法

在影像處理方面，有一部分的研究重點在於如何作影像分割，分割出有意義、有代表性的個體，稱之為物件(Object)、區塊(Blob)、及分割(Segment)等。分割的技巧在於先各個擊破(Divide-and-Conquer)，將整個問題縮小到某幾個子影像，在影像分割過程大致上可分為四大方向:

1. 邊緣偵測: 邊緣偵測技術一開始先去除影像雜訊，再透過如 LoG[36]或 Sobel[37]過濾器等來產生邊緣對應圖。而邊緣對應圖只能指出每個區域可能的邊界位置，所以需要進一步將邊緣與區域邊界結合，成為一封閉區以及將不必要的邊緣線移除。
2. 區域擴張及分裂: 先定義畫素的關係條件，如顏色、密度等，再針對畫素所成的集合作處理。依此方法，輸入影像會被畫分成為一些基本區域的集合，再遞迴式地將相鄰且同性質的區域合併成較大的區域，即擴張作用。而分裂作用則相反，一開始整張影像被視為一個區域，接下來再遞迴式地分割成較小且同性質的區域。邊緣偵測與此方法皆是以區域為基礎

的影像分割方法，這種方法雖然較費時，但由於效果佳，所以目前的影像分割演算法有許多是這種方法的變化型。

3. 聚合: 藉由聚合特徵向量中同性質的畫素達到分割影像的方法。首先將影像分成若干個區間(interval)或是能量單位(bin)，再計算這些局部性的畫素顏色分佈特性統計圖，接著利用貝式定理(Bayesian Theory)把每個統計圖統整為一個顏色分佈模型。這樣的方法較上述二法簡單，所需要的計算量也比較小。
4. 其他非圖型基礎的最佳化方法: 最佳化的技巧在於定義一個全域函式，來決定及找尋理想的分割。[38]便是一個利用馬可夫隨機域(Markov random field)的影像分割方法。首先影像被視為馬可夫隨機域函式，這個函式中包含著景物之間的空間分佈資訊。這類方法不但需要知道許多影像之間景物的資訊，同時計算量也相當繁複。

目前比較有名的影像分割演算法為下面幾個，除了演算法概念說明外，我們以圖 20為原始影像，實際透過下列三個分割演算法分割後，比較其不同的分割結果:

圖 20 影像分割範例圖片

1. JSEG[16]: 這是一種區域擴張及熔合的影像分割方法。圖 21 為JSEG的執行結果範例。

圖 21 JSEG 影像分割結果

2. NCut[17]: 這是一個正規化切割的圖型理論影像切割方法。圖22為NCut的執行範例。

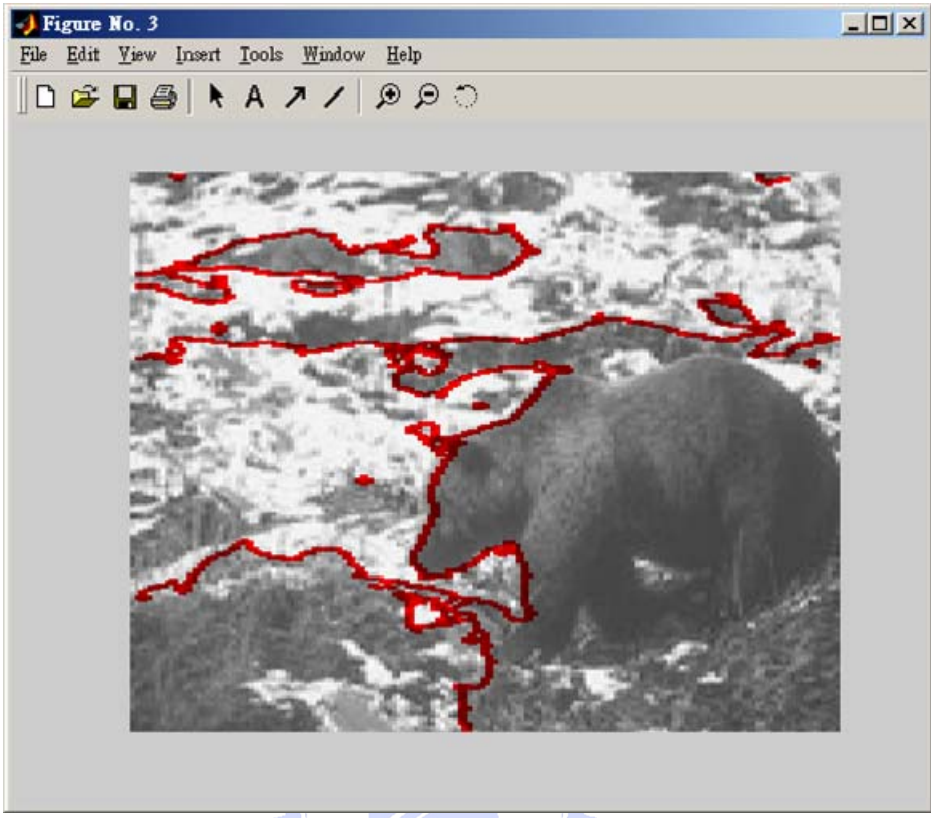

圖 22 NCut 影像分割結果

3. BSE[39]: BSE是一個以影像中的亮度(Brightness)、顏色(Color)、與紋理(Texture)為基礎的影像分割演算法。圖 23為執行範例。

圖 23 BSE 影像分割結果

由上面三個演算法的執行結果，我們可以發現，其分割後的區塊相較於人類語意中的物件仍有一段距離，且執行影像分隔演算法所需要的計算量較多。分割演算法對於本研究的中心主軸: 結合關鍵字與視覺特徵值的優點，是屬於輔助與加強的作用。一個好的分割演算法固然可以讓本研究的搜尋精確度好上加好，但目前來看一個適合本研究的分割演算法並沒有出現。因此我們會降低影像分割在本研究中的重要性，將之列為是未來本研究可繼續進行改善的方向之一。

2.4 分群演算法

分群(Clustering)為資料探勘(Data Mining)[23]領域中的一環。其主要概念為，將多個實體或虛擬的物體(objects)，透過分群演算法，分成數組(groups)，每一組內部的物體之間，相對於外部物體，都是較相似的。本研究採用的分群演算法為 k-medians clustering[40]，演算法說明如下:

假設有數個物體，給定物體與物體之間的距離，欲將數個物體分成 k 組，需執行以下步驟:

1. 先將物體亂數分成 k 組。
2. 每一組中，找出一個物體作為 medoids，也就是該組的代表物體。代表物體必須具備以下條件: 相較於同組內的其他物體，此物體到同組內其他物體的距離總合，必須是最短的。
3. 將每個物體重新分配至與代表物體距離最近的那一組。
4. 重覆執行 2-4 數次。

經由本章各節的探討，我們可以得知目前大型的 CBIR 系統，仍停留在以 bottom-up 的方式來解決影像檢索所遇到的各項問題。並從其中衍生出影像分割的研究領域。但至目前為止，影像分割對於泛用型 CBIR 系統的幫助，仍然有待更多研究的累積來加強。而分群演算法是資料探勘領域中的一環。本研究的目前即是組織各領域的研究成果，並提出一個整合關鍵字與視覺特徵的影像檢索系統。

第三章 整合關鍵字和視覺特徵的影像檢索

本章我們將介紹本研究所提出的系統架構與演算法，參考圖24。本系統共分為四大部分:

1. 一個現存的以關鍵字為基礎之影像搜尋系統。
2. 影像視覺特徵之擷取與正規化。
3. 影像分群。
4. 關鍵字擷取與關鍵字建議。

以下我們將介紹本研究的架構與演算法的詳細內容。

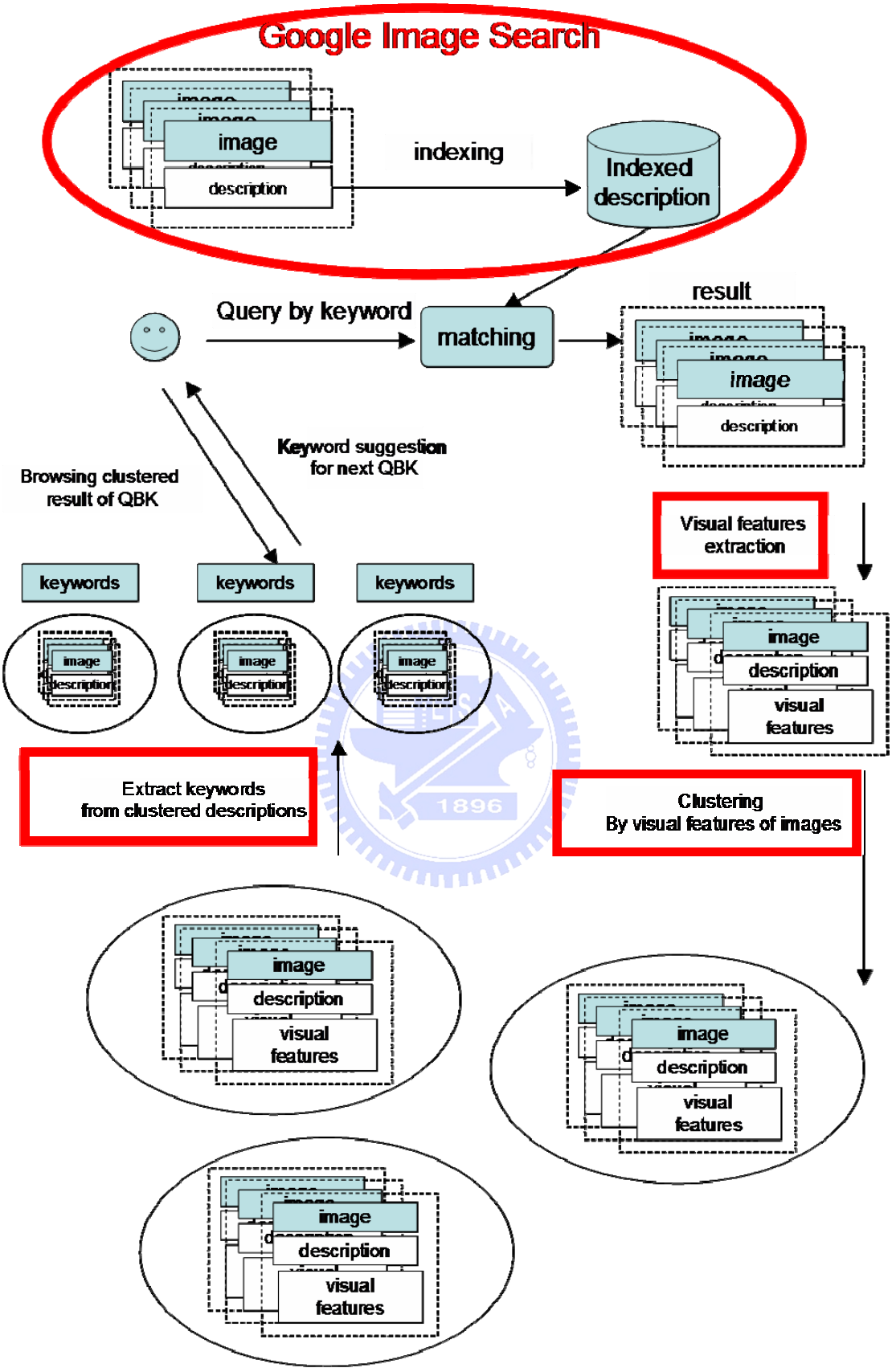
Google Image Search
image
description
indexing
Indexed description
Query by keyword
matching
result
image
description
Keyword suggestion for next QBK
Browsing clustered result of QBK
Visual features extraction
keywords
keywords
keywords
image
description
visual features
Extract keywords from clustered descriptions
Clustering By visual features of images

圖 24 本研究系統架構

3.1 方法架構

本節我們將分述本研究所提出的系統架構與演算法，我們分成四個部分來說明:

1. 一個以關鍵字為基礎的影像搜尋系統: 本研究採用 Google Image Search 作為關鍵字影像搜尋作為影像資料庫來源。其優點在於 Google Image Search 影像資料來自於全世界之網頁，影像取樣來源具多樣性，因此更能全方位的驗證本系統之正確性與可行性。然而，採用 Google Image Search 作影像資料庫也有其缺點，正因為其取樣來源多樣，影像的大小與格式較為不統一。另外，其影像所夾帶之文字訊息(通常來自影像所屬網頁上下文)，也是由全世界的網頁製作者共同提供，語意上的定義標準非常不統一。
2. 影像視覺特徵擷取與正規化: 我們將第一階段 Google Image Search 的搜尋結果中的每張圖片，依照 2.2.1 節的演算法，取出該張圖的 Color Layout, Color Structure, Contour Shape, Dominant Color, Edge Histogram, Homogenous Texture, Region Shape, Scalable Color 八項視覺特徵值，並以 2.2.2 節之 MPEG-7 規範的 XML 格式來儲存。由於八項視覺特徵值的數值記錄方式各有其定義，八項記錄方式與單位是互相獨立而不統一的。因此我們必須將八項記錄方式統一作正規化，如此每張影像之間，便可以定義出一個有意義的視覺特徵距離。有了每張影像之間的距離值，我們便可以透過正規化的影像距離進行影像分群。
3. 影像分群: 在取得正規化的影像距離後，我們利用 k-medoids

演算法將影像分群。影像分群是本研究重要的一環，分群結果的優劣將直接影響關鍵字與視覺特徵的對應關係。因此，在分群的過程中，我們將研究不同的分群參數對影像分群所帶來的影響，並以一組已結構化的圖片資料庫來進行一連串的實驗，幫助系統取得最佳的影像分群結果。

4. 關鍵字擷取與關鍵字建議: 在取得影像分群之結果後，我們已將圖片的視覺特徵，透過分群的方式，整合於本系統中。接著我們便以關鍵字擷取的技術，來連結關鍵字與影像視覺特徵之間的關係。我們將從每個分群中的圖片描述，取出能代表該分群的關鍵字。最後將每個分群的代表關鍵字顯示於使用者介面中，作為關鍵字建議。使用者將可由介面中，透過視覺化的方式，看見依照不同視覺特徵分群的影像資料，其所代表的語意，也可以透過分群代表關鍵字而得知。如欲得到某一特殊語意或視覺特徵的圖片，可直接點選該分群的代表關鍵字，透過本系統反覆式的進行更深度的搜尋。

以下我們將介紹整個系統與演算法的詳細內容和使用到的工具。

3.2 以關鍵字為基礎之影像搜尋系統

本研究採用Google Image Search作為關鍵字影像搜尋作為影像資料庫來源。Google Image Search的搜尋結果畫面請參考圖 25。我們放大搜尋結果中的單一項目，如圖 26，分析Google Image Search所能提供本系統的影像與相關資訊。我們從圖 26發現，每一個影像資訊包含四個項目:

1. 影像縮圖與連往影像詳細資訊的超連結。
2. 影像相關描述。
3. 影像維度、檔案大小、檔案格式。
4. 所在網域。

我們將擷取儲存於Google Image Search的影像縮圖，作為視覺特徵擷取的來源檔案，並作為本系統使用者介面的顯示圖片。而在代表影像語意的文字描述部分，我們發現Google Image Search所擷取的文字描述過於簡短，最長不超過十個英文單字，所能提提供的語意資訊有限。因此我們進一步查詢影像詳細資訊的超連結，以找出更多關於影像的語意資訊。如圖 27所示，上半部為上一頁已提供之摘要資訊，下半部則是原始圖片來源之所屬網頁。之外並無其他更結構化的語意資訊可供利用。因此在影像的語意資訊上，我們只能透過擷取原始圖片所屬網頁，解析原始網頁與相關網頁語法，以取得更多能代表該圖片的語意資訊，細節作法我們將在 3.5 節中說明。

圖 25Google Image Search 搜尋範例

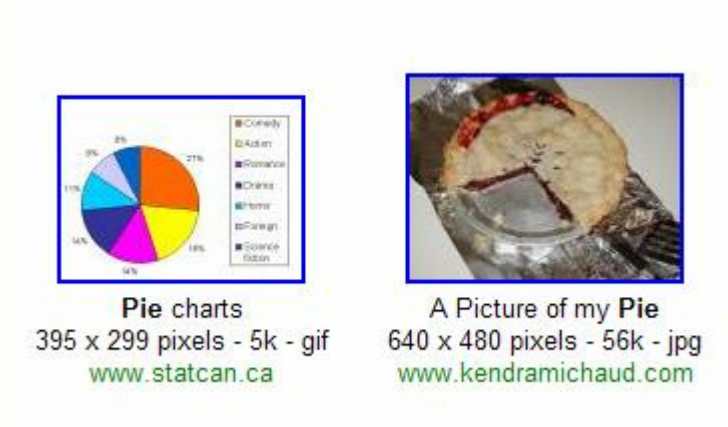

圖 26Google Image Search 搜尋範例放大

圖 27Google Image Search 影像來源網頁

3.3 影像視覺特徵之擷取與正規化

從 Google Image Search 取得影像檔案後，我們便透過 MPEG-7 的工具，擷取每一張影像的視覺特徵值，並以 XML 格式儲存。

在分別取出每一張影像的八項視覺特徵值後，我們依照MPEG-7 對於每一個視覺特徵的距離定義，分別計算在單一視覺特徵內，影像之間的距離值。在這個階段中，我們面臨到一個問題: MPEG-7 對於每一個特徵距離值的單位，並不是統一的。以圖 28為例子。

(A) (B) (C) (D)

Distance / items	A⇔B	A⇔C	A⇔D
Color Layout	79.372894	62.023926	74.620842
Color Structure	4.450980	6.427451	6.752941
Dominant Color	4.3667e+07	5.0087e+07	5.7213e+07

圖 28 未經正規化的視覺特徵值

我們可以從以上例子發現，影像 A, B, C, D 在 Color Layout、Color Structure、與 Dominant Color 三個不同的特徵值下，經由 MPEG-7 定義所計算出來的距離值，數字量級差距非常大。因為 MPEG-7 在單位上的不統一，造成我們在分群時，無法平衡的綜合八項視覺特徵值，產生有意義的影像視覺距離。因此我們必須定義一個正規化的方式，

來平衡這八個不同量級的距離值。我們假設影像 i 和 j 是經由同一個 QBK 搜尋結果中的其中兩張影像。其中:

- x[i][j]: 影像 i 和影像 j 在特徵 x 中，由 MPEG-7 所定義的原始距離值。
- max(x): 在同一個 QBK 搜尋結果中，特徵 x 距離的最大值。
- X[i][j] = x[i][j] / max(x[i][j]): 我們定義影像 i 和影像 j 在特徵 x 中的正規化距離值 X[i][j] = x[i][j] / max(x)

藉由以上的定義，正規化後的特徵距離值，永遠只會落在 0 和 1 之間。在分別取得八個特徵的正規化距離值後，我們綜合八個維度的向量，取其平方和開根號，定義為影像 i 和影像 j 在八個特徵維度上的距離 D[i][j]。

- D[i][j] = Sqrt((X1[i][j])^2 + (X2 [i][j])^2 + (X3 [i][j])^2 + (X4 [i][j])^2 + (X5 [i][j])^2 + (X6 [i][j])^2 + (X7 [i][j])^2 + (X8[i][j])^2)

有了綜合八個特徵值的影像距離後，我們便可以進入下一階段，利用數據化的視覺特徵值進行影像分群。

3.4 影像分群

在取得所有影像的視覺特徵距離陣列後，我們將此陣列帶入 2.4 節中所介紹的 k-medoids 演算法。演算法說明如下:

假設有數個物體，給定物體與物體之間的距離，欲將數個物體分成 k 組，需執行以下步驟:

1. 先將物體亂數分成 k 組。
2. 每一組中，找出一個物體作為 medoids，也就是該組的代表物體。代表物體必須具備以下條件: 相較於同組內的其他物

體，此物體到同組內其他物體的距離總合，必須是最短的。

3. 將每個物體重新分配至與代表物體距離最近的那一組。

4. 重覆執行 2-4 數次。

為了不致於因為分群個數過多，造成使用上的不友善，我們選擇把分群個數，也就是步驟 1 中的 k，固定為 7。而在未來我們將把分群的個數調整，加入到使用者介面中，讓使用者可以根據不同的檢索結果，自行調整最佳的分群個數。

3.5 關鍵字擷取與關鍵字建議

影像在經過視覺特徵擷取，與利用視覺特徵值完成分群後，我們便開始組織影像本身相關的文字描述，來連接關鍵字與視覺特徵的關係。由 3.2 節的分析，我們可以發現，Google Image Search 本身所提供的文字描述，並沒有提供足夠的資訊供本研究使用，因此在本節中，我們將透過分析影像本身所屬的網頁內容，擷取出代表該影像的關鍵字。再從同一群的影像與其代表關鍵字中，擷取能代表該群的關鍵字。

當我們在分析影像所屬的網頁內容時，首先我們須區分，那些內容是和影像有關的，那些是無關的。因為對於一個網頁來說，影像可能只佔其中一小部分，並不一定和整個網頁的文字都有關係。我們先將所有網頁中的 HTML 語法部分全部過濾，因為 HTML 決大部分都是控制著網頁的配置與排版，通常不會帶有過多的關鍵訊息。

在過濾掉 HTML 語法後，我們得到一個完整的純文字檔案，我們將這個純文字檔經過一個 Token 分析器，切成一個一個單字的 Token。例如: There are four people in the Lin’s family. 在經過 Token 分析器後，會變成一個包含”there”, “are”, “four”, “people”, “in”, “the”,

“Lin’s”, “family” 的字串陣列。在得到這個 token 化的字串陣列後，我們再將一些無意義的單字，如: be 動詞、介系詞、代名詞、冠詞等過濾掉，成為一個有意義的 token 化字串陣列。

雖然我們已經得到一個類似關鍵字的 token 陣列，然而這樣的資訊還是有點過多而無法精確地代表影像的語意。因此，我們依照網路上的網頁製作習慣，作一個簡單的假設: 網頁中離影像越近的文字，越能代表該影像所描述的語意。

在為影像擷取代表關鍵字的最後一步，我們依照影像在該網頁的原始位置中，往上與往下各取 n 個有意義的單字，作為代表這張影像的關鍵字陣列。n 值越大，所取出的影像代表關鍵字越多，但其精確度也越低，反之亦然。我們在此不會事先定義怎樣的 n 值才是最佳的結果，因為這已經 Information Retrieval 的研究範圍。然而不可否認的是，一個好的 n 值可以影響本系統搜尋結果。因此，我們將 n 值的設定提供在本系統的使用者介面中，讓使用者依照搜尋的結果，自我調整 n 值的大小。如果使用者覺得取出的關鍵字過少，則可透過調高 n 值來修正。如果覺得關鍵字的結果不夠精確，則可透過調低 n 值來提高關鍵字精確度。

有了每張影像的代表關鍵字後，我們把每一群的影像代表關鍵字聯集起來，依照每個關鍵字出現的頻率排列，當某個關鍵字出現的頻率高於 f 時，系統便會選取此關鍵字作為該分群的代表關鍵字。和 n 值的概念類似的是，一個好的 f 值一樣可以影響系統的搜尋結果，但如何設定一個好的 f 值，本身就是一個很大的問題。我們在此也先將 f 值的設定提供於使用者介面中，由使用者依照每次的搜尋結果，自己調整 f 值。f 值越大，其分群代表關鍵字越多。f 值越小，其分群代表關鍵字的精確度則越高。

3.6 系統整合

至此我們已經完成了本研究的各個重要模組，我們將第三章各節所提到的各個模組，按所示的的流程加以整合，形成一個完整可用的系統雛型。其系統整體運作流程說明如下:

1. 使用者透過本系統的使用者介面，輸入查詢關鍵字(Query By Keyword)
2. 本系統的使用者介面將使用者輸入的關鍵字傳給系統控制單位(Control Unit)
3. 控制單位將關鍵字送出至 Google Image Search
4. Google Image Search 傳回查詢結果，由控制單位分析並擷取所需的影像和網頁資料
5. 控制單位將步驟 4 所得到的影像資料，交由視覺影像擷取工具(Visual Features Extractor)。
6. 視覺影像擷取工具擷取每張影像的視覺特徵值並回傳給控制單位。
7. 控制單位將步驟 6 得到的視覺特徵值正規化後，交由負責分群的程式，進行分群。
8. 控制單位將步驟 4 所得到的網頁，交由關鍵字擷取器，擷取每張影像代表關鍵字並回傳。
9. 控制單位將步驟 8 所得到的影像代表關鍵字，配合步驟 7 所得到的結果，交由分群關鍵字擷取器，擷取每一分群的代表關鍵字並回傳。
10. 控制單位將經過分群後的影像與代表關鍵字，交由使用者介面顯示。

11.使用者瀏覽分群後的影像與代表關鍵字。

12.若想得到更精確的影像，可再點選分群代表關鍵字，系統將重覆步驟 1~步驟 12 的各項流程，形成反覆式的搜尋。

系統雛型的實際運作與討論，我們將在第四章中作更詳細的介紹。

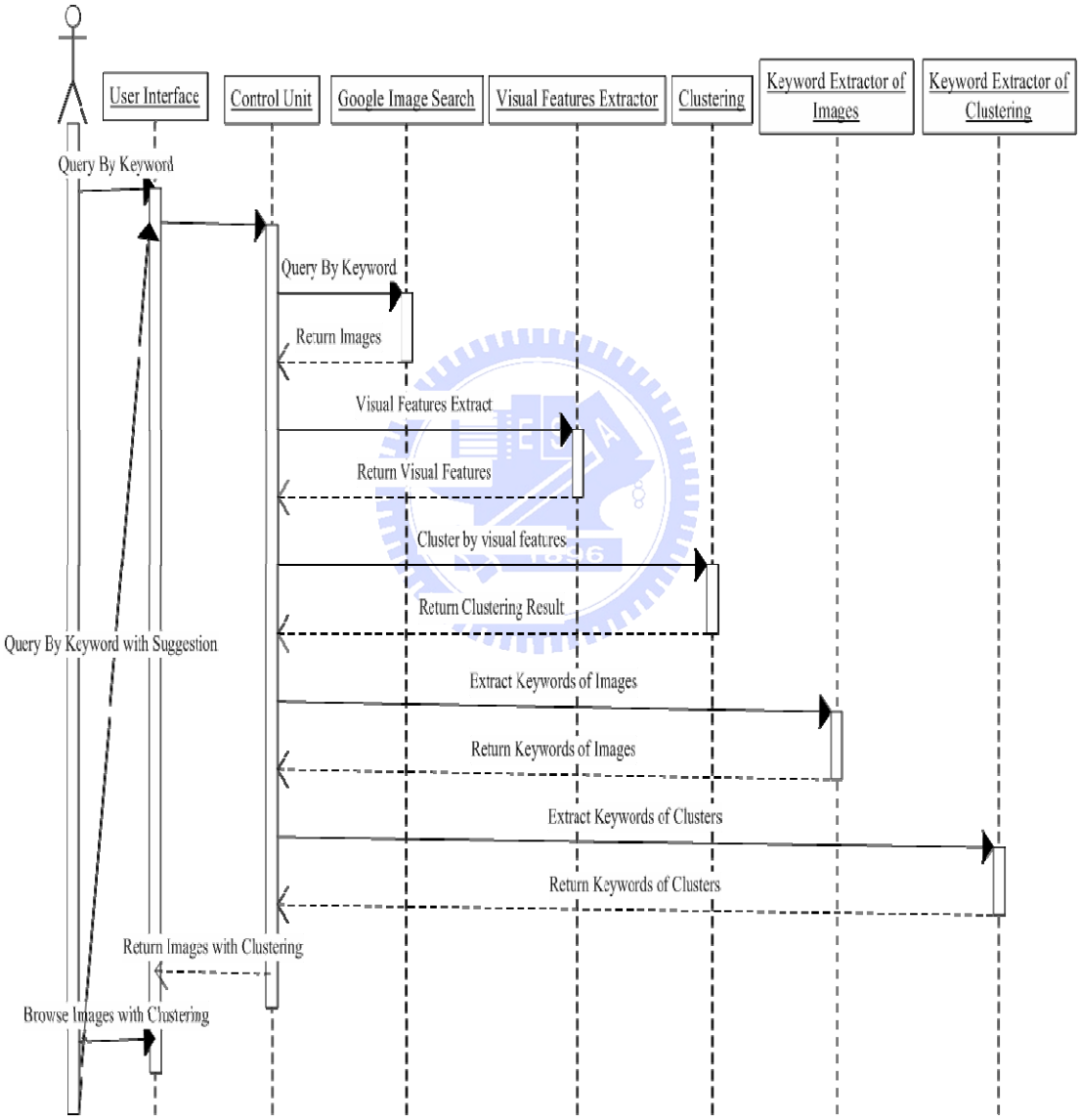

圖 29 系統執行流程圖

第四章 系統雛型與成果

為了能確定驗證本研究的各項，本研究實作一系統雛型，於 4.1 節中說明。我們在 4.2 節中，透過七個實際的測試案例，來驗證系統的運作成果。

4.1 系統雛型

為了能確定驗證本研究的各項理論，本研究實作一系統雛型，如圖 30所示，第一行為關鍵字輸入欄位，第二行之數值為 3.5 節中所提到的n值，使用者可依 3.5 節之說明，自行調整n值的大小，一開始的預設值為 150，代表系統在作關鍵字擷取時，將從影像所屬的網頁上，經過token化處理後，往下與往下各取 150 個字串，作為該影像的代表關鍵字。第三行之數值為 3.5 節中所提到的f值，預設為 5，代表若某一關鍵字在該群的影像關鍵字中，若超過 5 張影像擁有該關鍵字，就將此關鍵字列為分群代表關鍵字。本系統仍為一雛型，在執行速度上仍未最佳化，因此系統整體執行速度會比較慢。我們將在第五章中討論影響系統執行效率的各項因素，並提出可以改進的方向。

圖 30 系統雛型查詢介面

4.2 成果驗證

在這一節中，我們將幾個關鍵字，帶入系統雛型執行，並觀察系統雛型的執行，來驗證本研究的成果。我們選擇以下幾個關鍵字，作為系統雛型的測試案例: pie, formula, windows, opera, nano, redhat, taiwan。每個測試案例的細節測試過程說明如下。

4.2.1 測試案例 1: pie

關鍵字pie的執行結果如圖 31。

圖 31 關鍵字 pie 執行結果

由圖 31我們可以發現，在前兩群中的影像中，主要以電腦文書處理的影像、背景偏白為主。後面四群的來源大部分是一般照片影像為主，其中內容大多是和食物有關的照片。我們將其中兩個較有代表性之分群放大，以方便觀察。如圖 32、圖 33所示。我們由圖 32發現，該分群中共有 14 張影像，其中 7 張都是圓餅圖，剩下 7 張則較無統一的語意概念。而其代表關鍵字也偏向與圓餅圖相關的關鍵字，如:

bar, data, charts, chart等。而圖 33的影像與食物比較相關，其 17 張影像中，有 14 張都是派的照片。分群代表關鍵字也都偏向食物的概念，如: apple, baked, food, cream等。

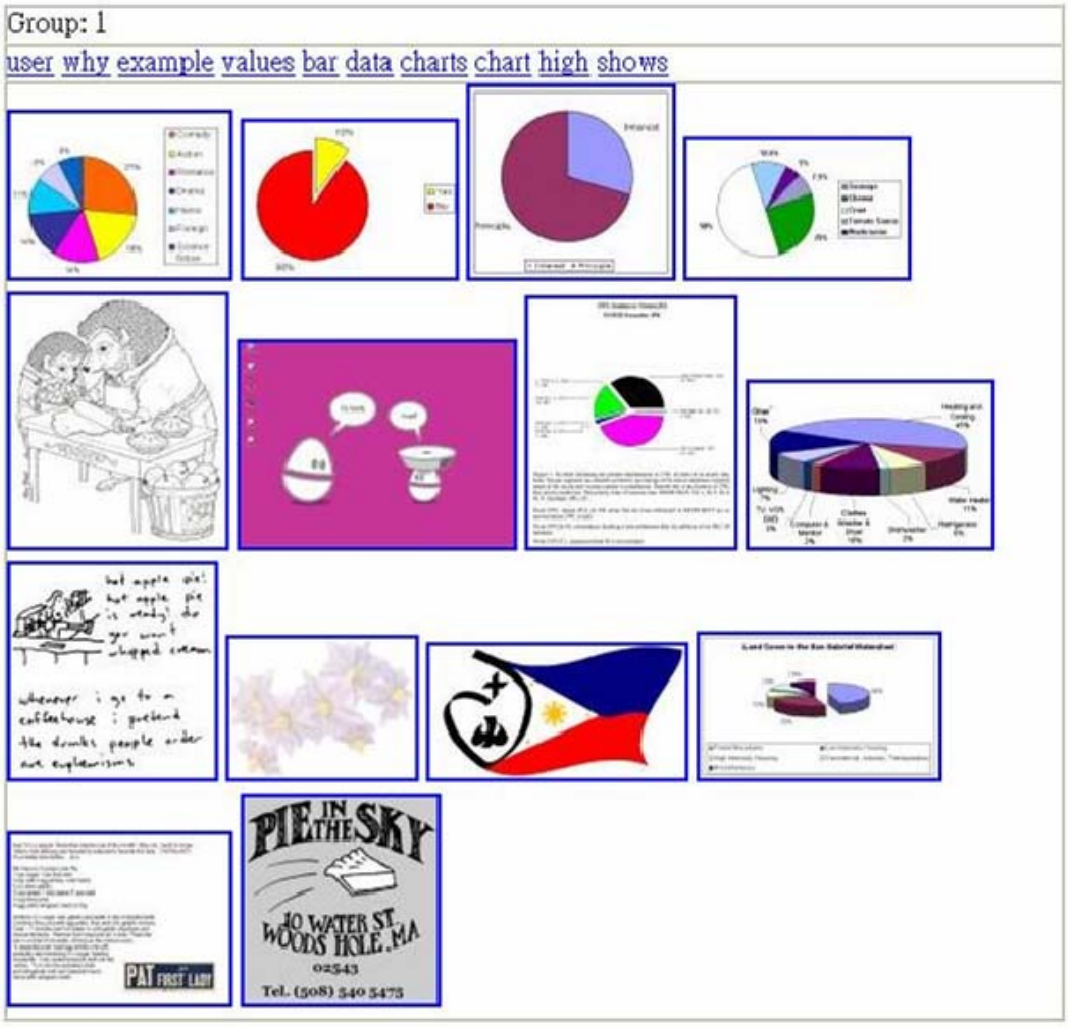

圖 32 關鍵字 pie 的分群結果之一，以圓餅圖為主

圖 33 關鍵字 pie 的分群結果之一，以派為主

一般在QBK系統中，若我們輸入pie為關鍵字，就會得到包含食物的派和圓餅圖兩種概念的影像。而透過本系統，我們可以藉由兩種不同概念影像在視覺上的差異，而將兩種不同語意的影像，透過本系統作分群區分。使用者若只想單獨查詢某一語意的影像，則可點選某一分群的代表關鍵字。例如我們若想查詢的影像，是和圖 32類似的圓餅圖，我們便可以點選其中的關鍵字chart，而得到如圖 34的查詢結果。若我們想查詢的是蘋果派的照片，則可以點選圖 33中的apple，而得到如圖 35都是蘋果派的搜尋結果。

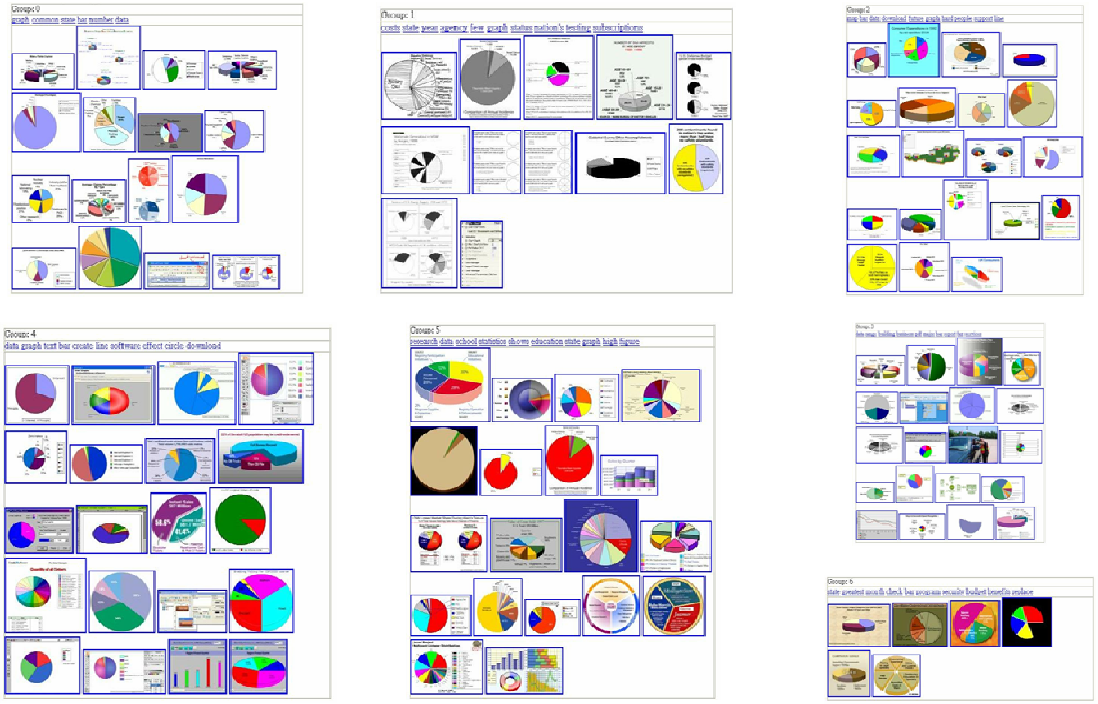

圖 34 關鍵字 pie+建議關鍵字 chart 搜尋結果

圖 35 關鍵字 pie+建議關鍵字 apple 搜尋結果

4.2.2 測試案例 2: formula

關鍵字formula的執行結果如圖 36。

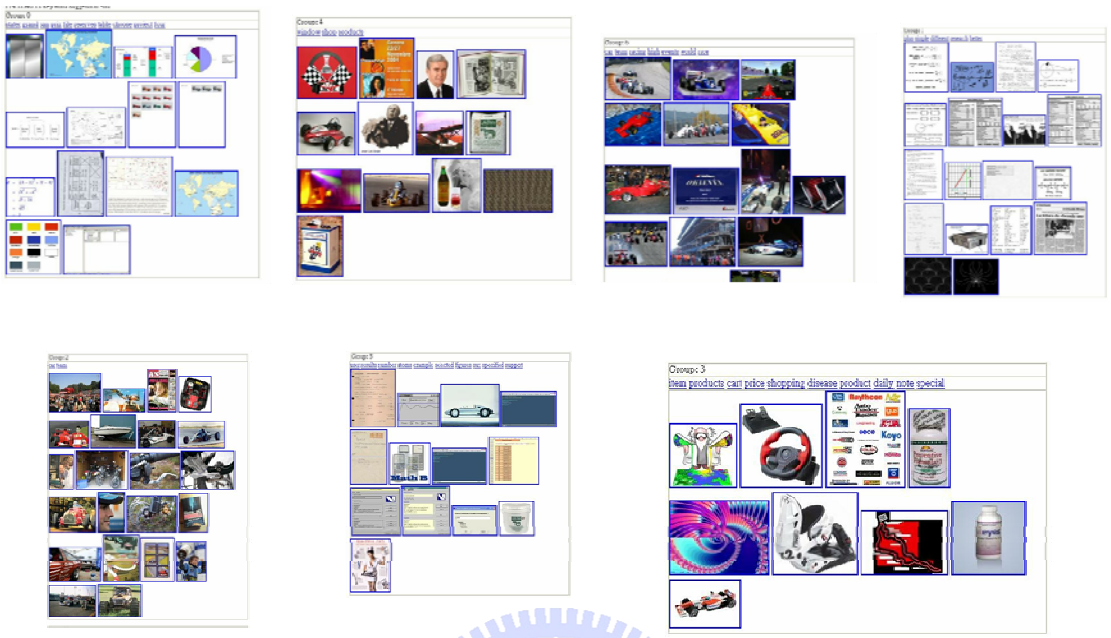

圖 36 關鍵字 formula 搜尋結果

為了方便觀察，我們一樣將其中兩個較具代表性的分群放大，如圖 38和圖 39所示。圖 38的影像內容以白底黑字為主，主要為化學與數學的公式和營養表格，在 18 張影像中，佔有 14 張。其關鍵字為simple, different, research, better等。圖 39的內容則明顯的以方程式賽車為主，其關鍵字包含: car, team, racing, race, world等。我們一樣模擬兩種不同的搜尋目的，分別搜尋以科學研究和賽車為主的影像。圖 39為點選分群關鍵字research的搜尋結果，我們可以發現其內容都和賽車全部無關。圖 40為點選分群關鍵字racing的結果，內容包含賽道、車體、和一些方式程有關的照片等。

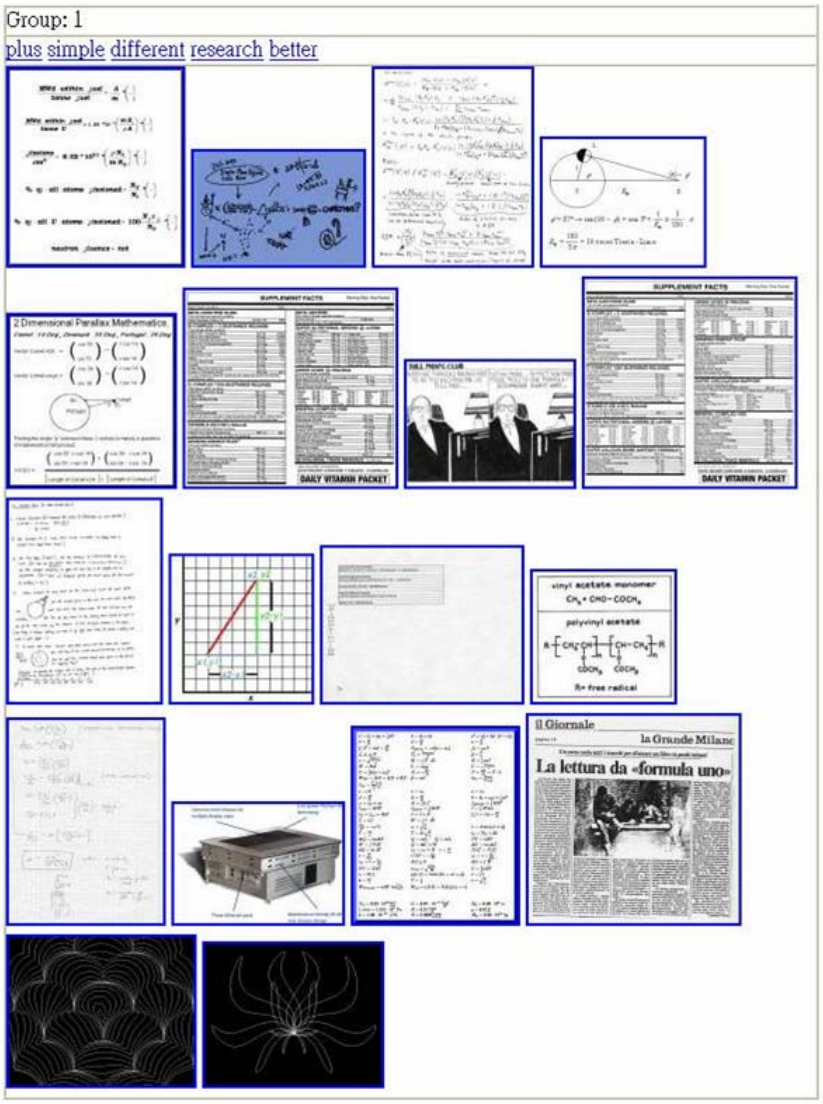

圖 37 關鍵字 formula 的分群結果之一，以公式和營養表為主

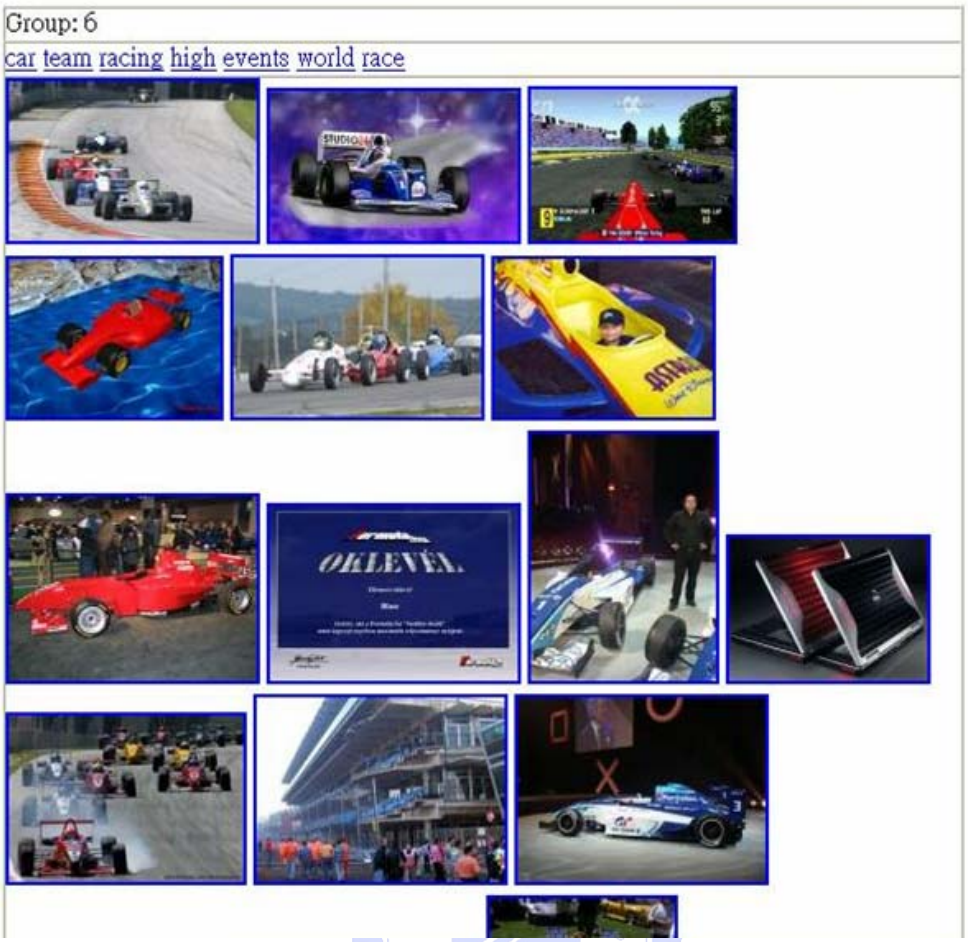

圖 38 關鍵字 formula 的分群結果之一，以方程式賽車為主

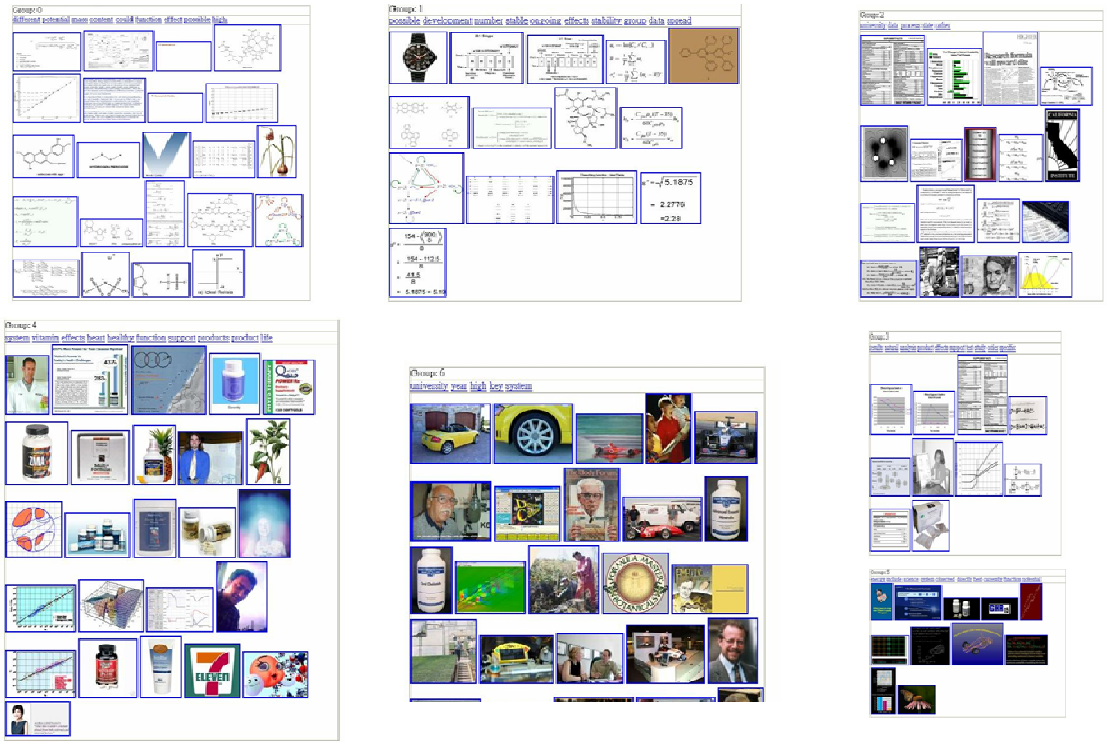

圖 39 關鍵字 formula+建議關鍵字 research 搜尋結果

圖 40 關鍵字 formula+建議關鍵字 racing 搜尋結果

4.2.3 測試案例 3: windows

關鍵字windows的執行結果如圖 41。

圖 41 關鍵字 windows 執行結果

我們一樣將其中兩個分群結果放大觀察，如圖 42和圖 43所示。圖 42中的影像內容以實體建築物的窗戶為主，代表關鍵字包含doors, visit, products等。而圖 43則主要以微軟公司的windows軟體為主，其代表關鍵字包含internet, security, microsoft等。這個測試案例相當完美，在兩個不同的分群中的影像，各自代表一個獨立的語意，該分群中也沒有出現例外的影像，其原因在於，從QBK系統所得到的影像來源中，這兩個不同語意的視覺特徵差異性夠大。我們按此兩種不同的搜尋目標，分別點選doors和microsoft兩個分群代表關鍵字，得到如圖 44的實體窗戶和圖 45視窗軟體，兩種不同語意的搜尋結果。

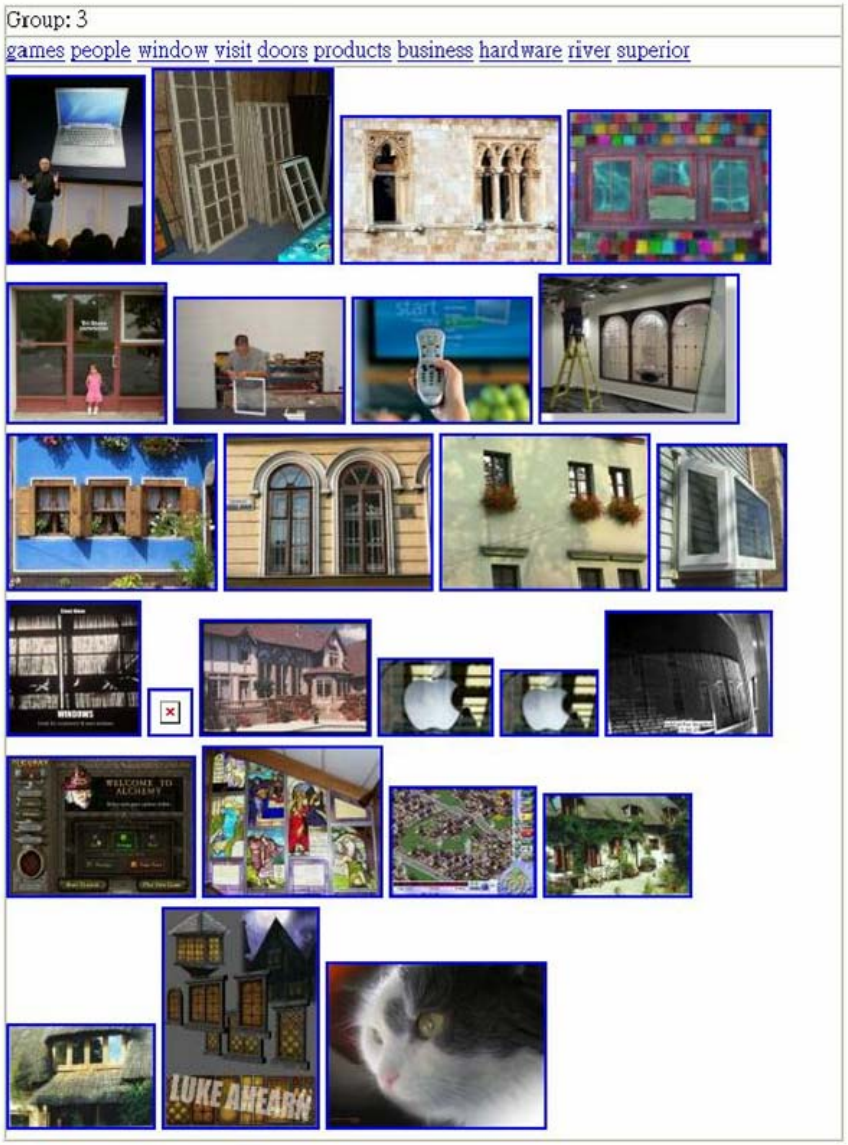

圖 42 關鍵字 windows 的分群結果之一，以窗戶為主

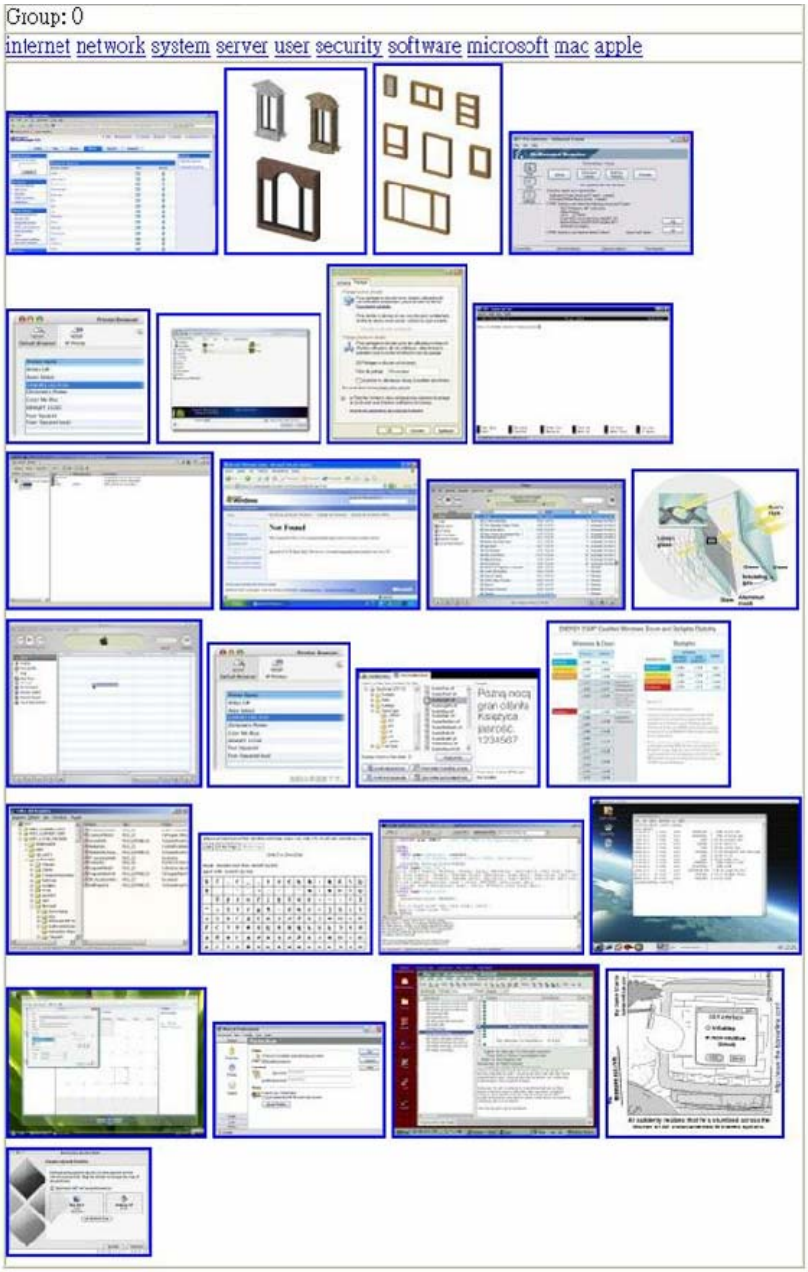

圖 43 關鍵字 windows 的分群結果之一，以視窗軟體為主

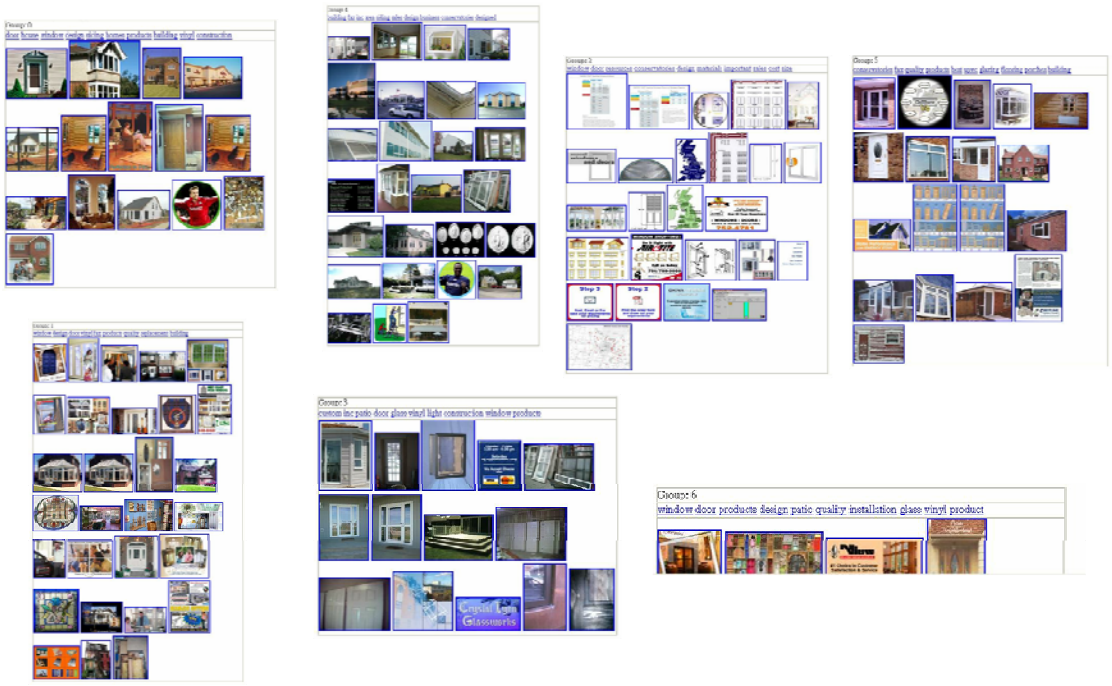

圖 44 關鍵字 windows+建議關鍵字 doors 搜尋結果

圖 45 關鍵字 windows+建議關鍵字 microsoft 搜尋結果

4.2.4 測試案例 4: opera

關鍵字opera的執行結果如圖 46。

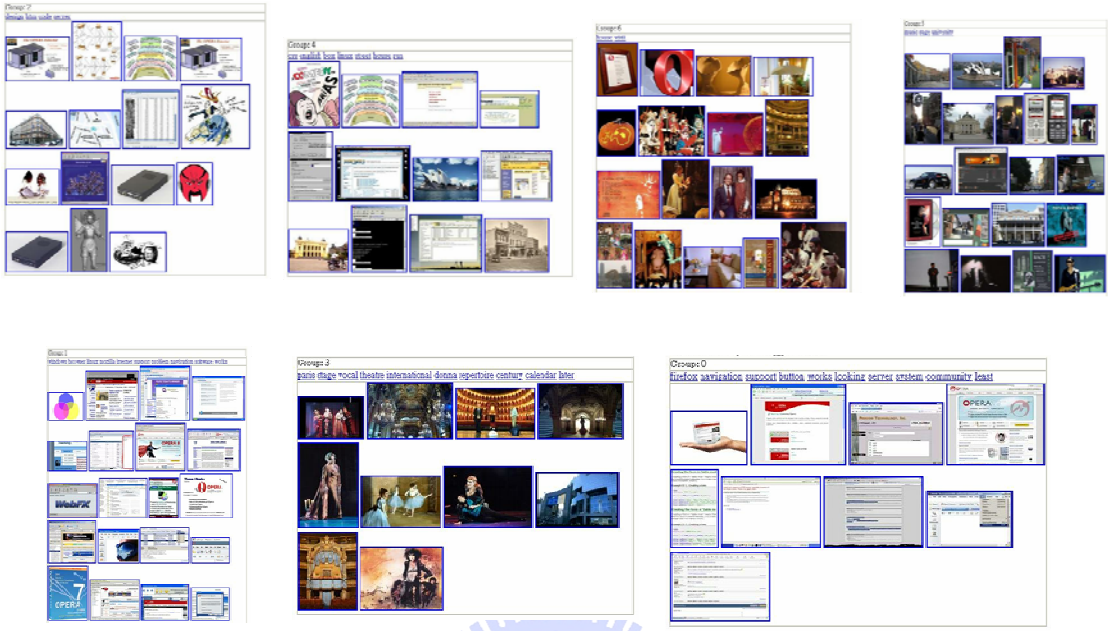

圖 46 關鍵字 opera 執行結果

我們一樣將其中兩個分群結果放大觀察，如圖 47和圖 48所示。圖 47中的影像內容以opera這個網路瀏覽器為主，代表關鍵字包含windows, browser, internet, software等。而圖 48則主要以歌劇相關影像為主，其代表關鍵字包含state, vocal, theatre等。這個測試案例如果 4.2.3 一樣，兩個分群所代表的語意，其影像的視覺差異有相當大的距離，也是一個相當完美的案例。我們也按此兩種不同的搜尋目標，分別點選browser和theatre兩個分群代表關鍵字，得到如圖 49的opera瀏覽器和圖 50歌劇相關影像，兩種不同語意的搜尋結果。

圖 47 關鍵字 opera 的分群結果之一，以 opera 瀏覽器為主

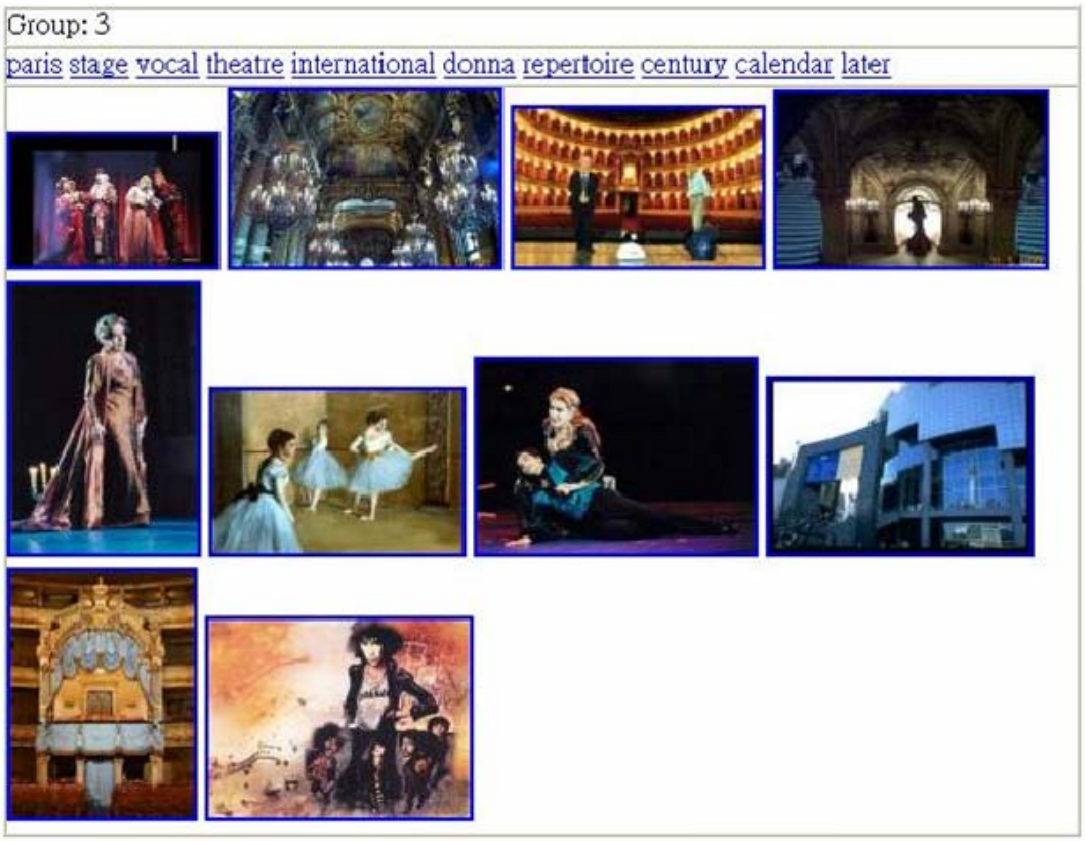

圖 48 關鍵字 opera 的分群結果之一，以歌劇為主

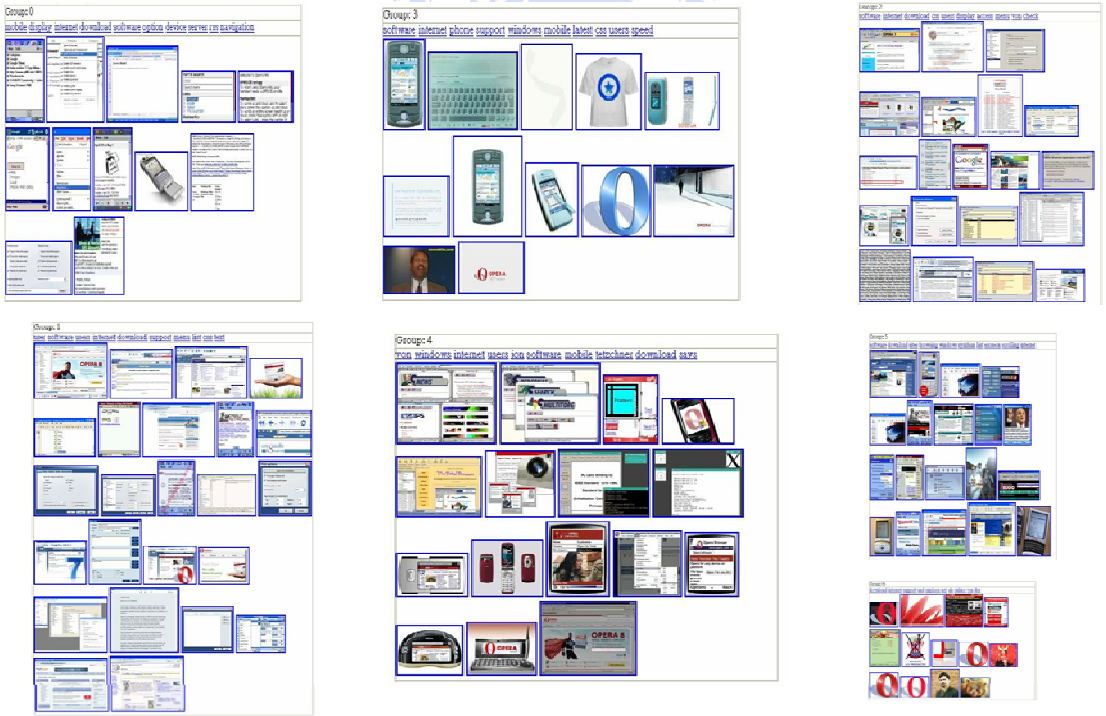

圖 49 關鍵字 opera+建議關鍵字 browser 搜尋結果

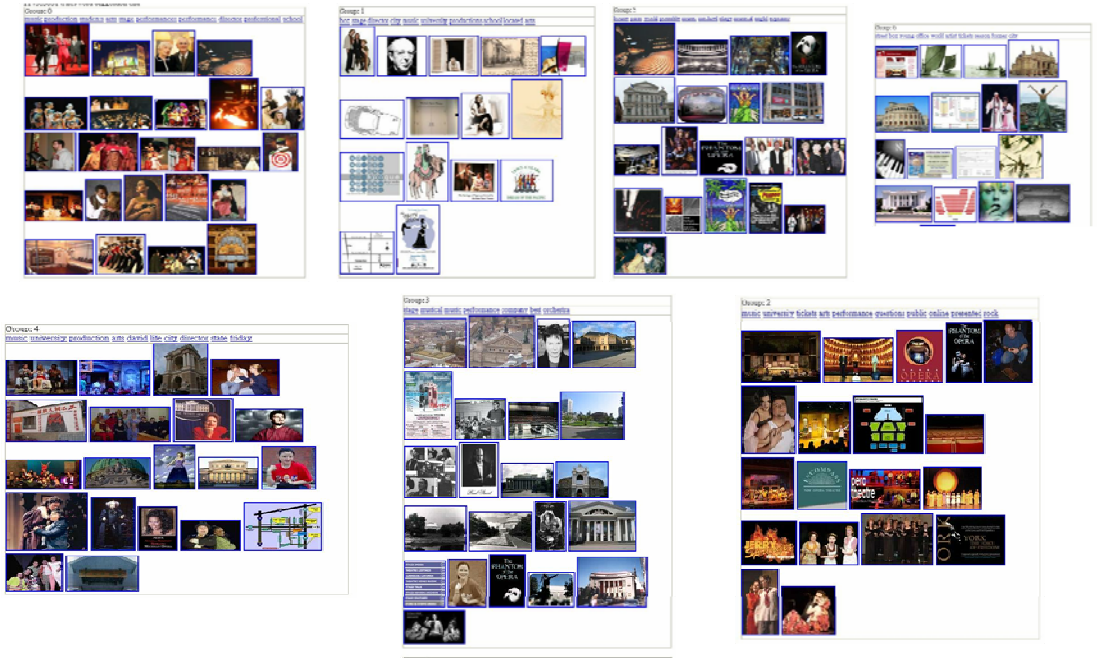

圖 50 關鍵字 opera+建議關鍵字 theatre 搜尋結果

4.2.5 測試案例 5: nano

關鍵字nano的執行結果如圖 51。

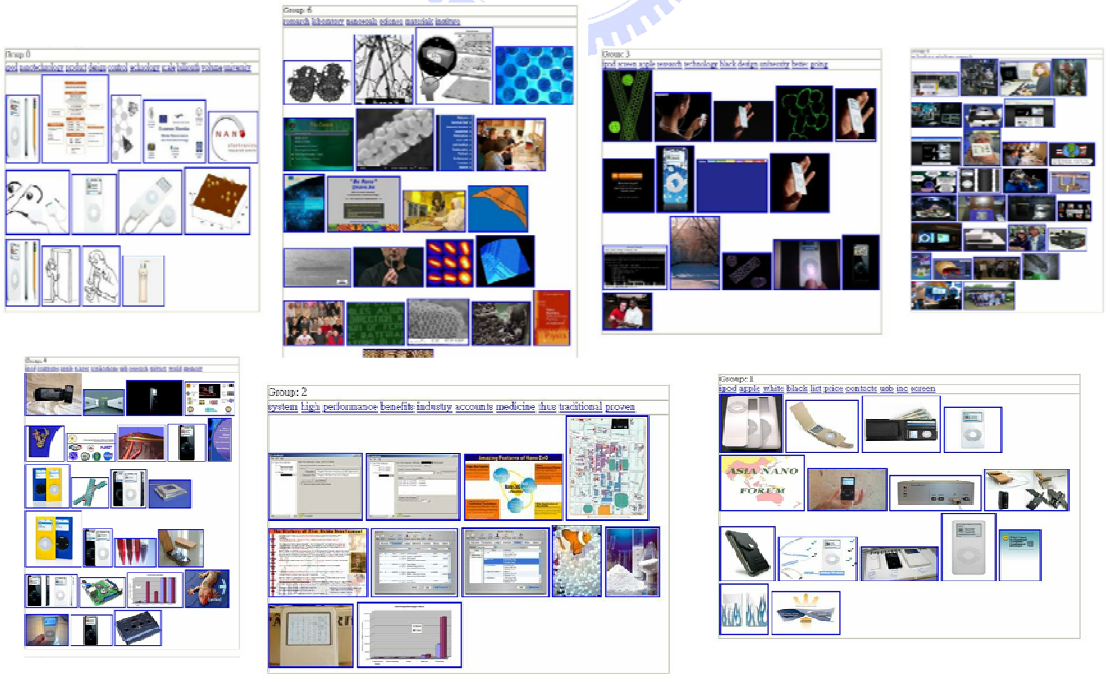

圖 51 關鍵字 nano 執行結果

我們將其中兩個分群結果放大觀察，如圖 52和圖 53所示。圖 52中的影像內容以Apple公司的ipod nano產品為主，代表關鍵字包含ipod, apple, price, usb等。而圖 53則主要和科學相關的奈米技術為主，其代表關鍵字包含research, laboratory, science, nanoscal等。我們在這兩個分群中可以發現，對應於這兩個不同語意的影像來源，其視覺化特徵的差異性頗大，例如以ipod來說，很多是產品搭配不同背景的照片，因此也造成了分群結果較為混亂的情況產生。不過我們還是可以藉由分群代表關鍵字，來進一步取得更精確的相關影像，如圖 54和圖 55所示。

圖 52 關鍵字 nano 的分群結果之一，以 ipod nano 產品為主

圖 53 關鍵字 nano 的分群結果之一，以奈米科學為主

圖 54 關鍵字 nano+建議關鍵字 ipod 搜尋結果

圖 55 關鍵字 nano+建議關鍵字 science 搜尋結果

4.2.6 測試案例 6: redhat

關鍵字redhat的執行結果如圖 56。

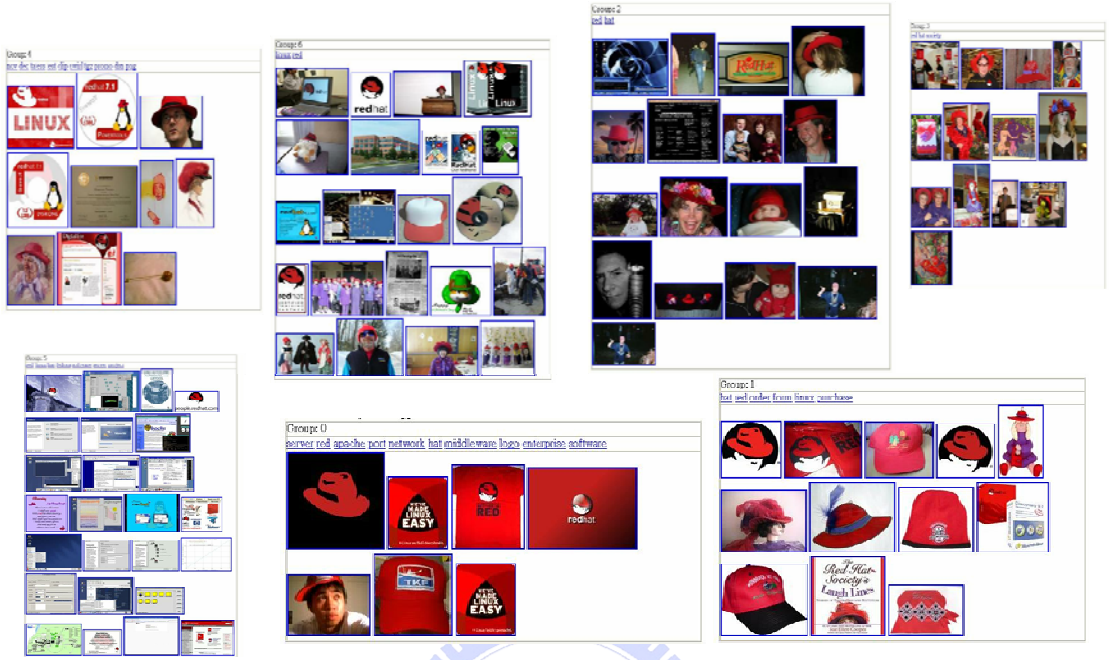

圖 56 關鍵字 redhat 執行結果

我們將其中兩個分群結果放大觀察，如圖 57和圖 58所示。圖 57中的影像內容為人戴著紅色帽子居多，代表關鍵字包含red, hat, society等。而圖 58則主要和著名linux公司redhat相關軟體為主，其代表關鍵字包含linux, fedora, software, server, project 等。我們按此兩種不同的搜尋目標，分別點選hat和linux兩個分群代表關鍵字，得到如圖 59和圖 60兩種不同語意的搜尋結果。我們從圖 60的第二個分群可以發現，該分群的影像主要以redhat的安裝畫面為主，若使用者想進一步查詢與redhat linux相關的安裝畫面，可以透過點選關鍵字建議再作第三次的精確查詢。

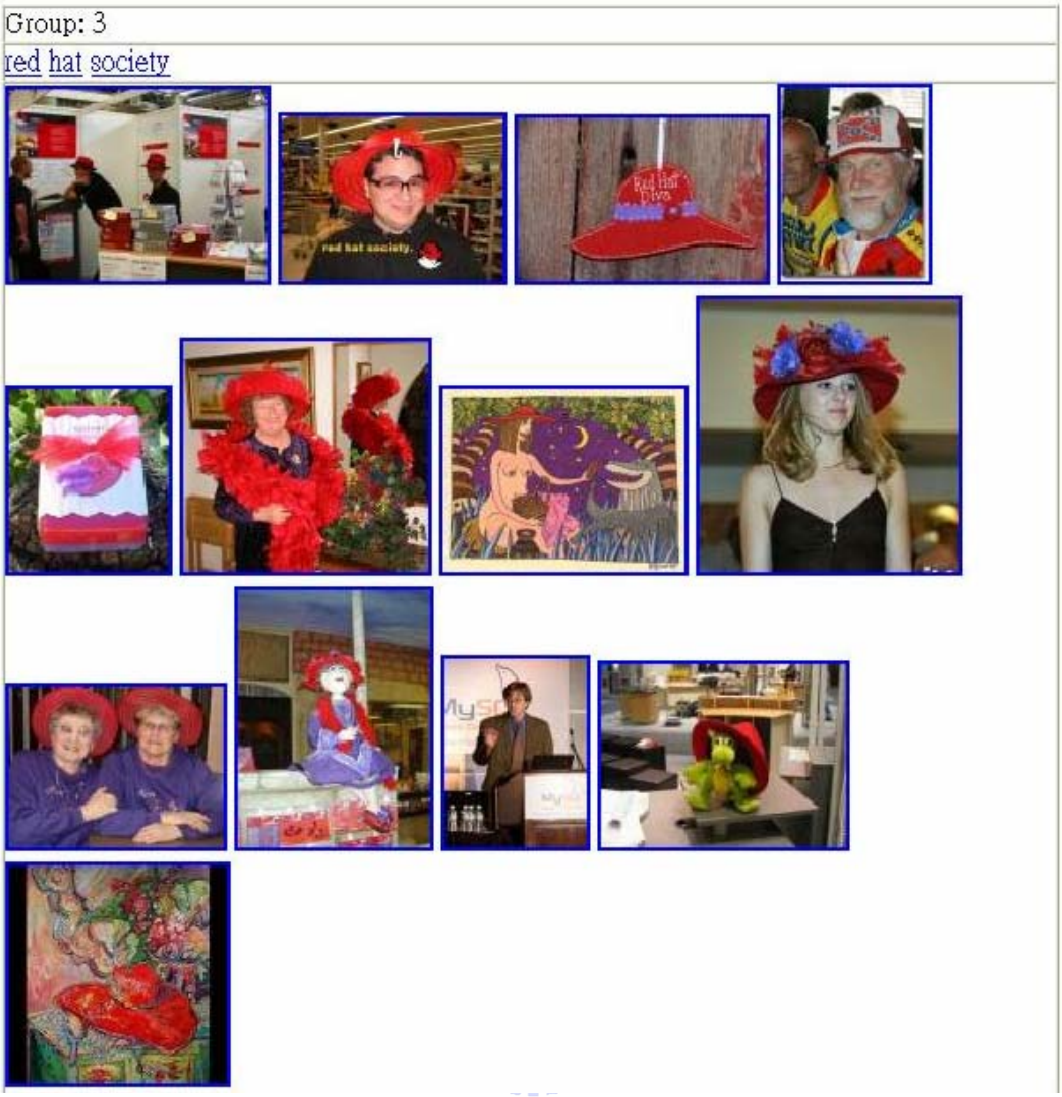

圖 57 關鍵字 redhat 的分群結果之一，以紅色的帽子為主

圖 58 關鍵字 redhat 的分群結果之一，以 redhat linux 為主

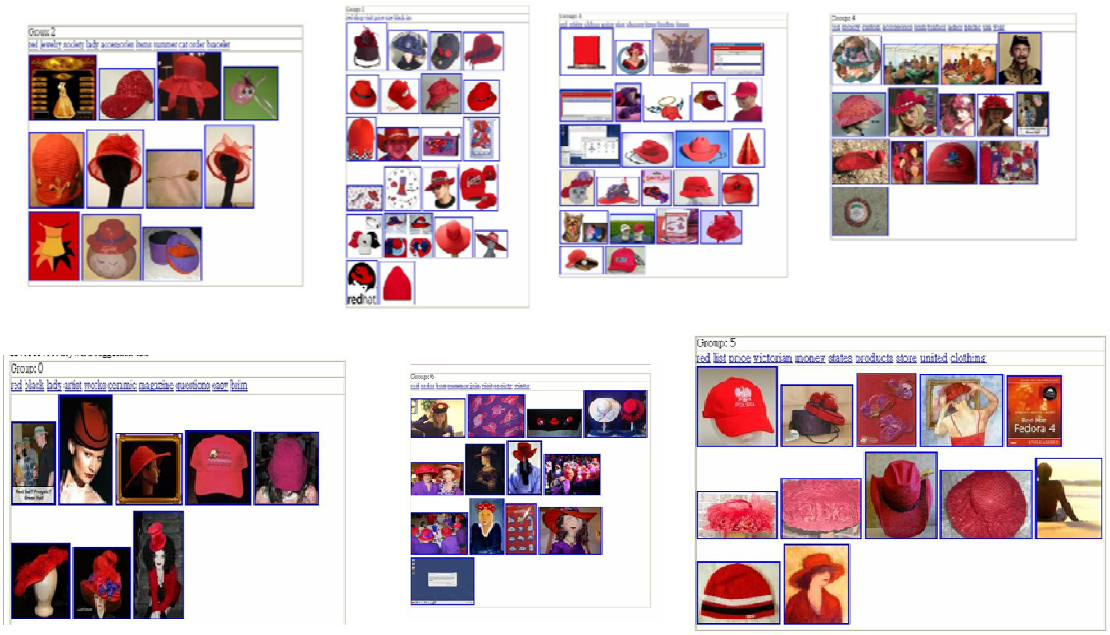

圖 59 關鍵字 redhat+建議關鍵字 hat 搜尋結果

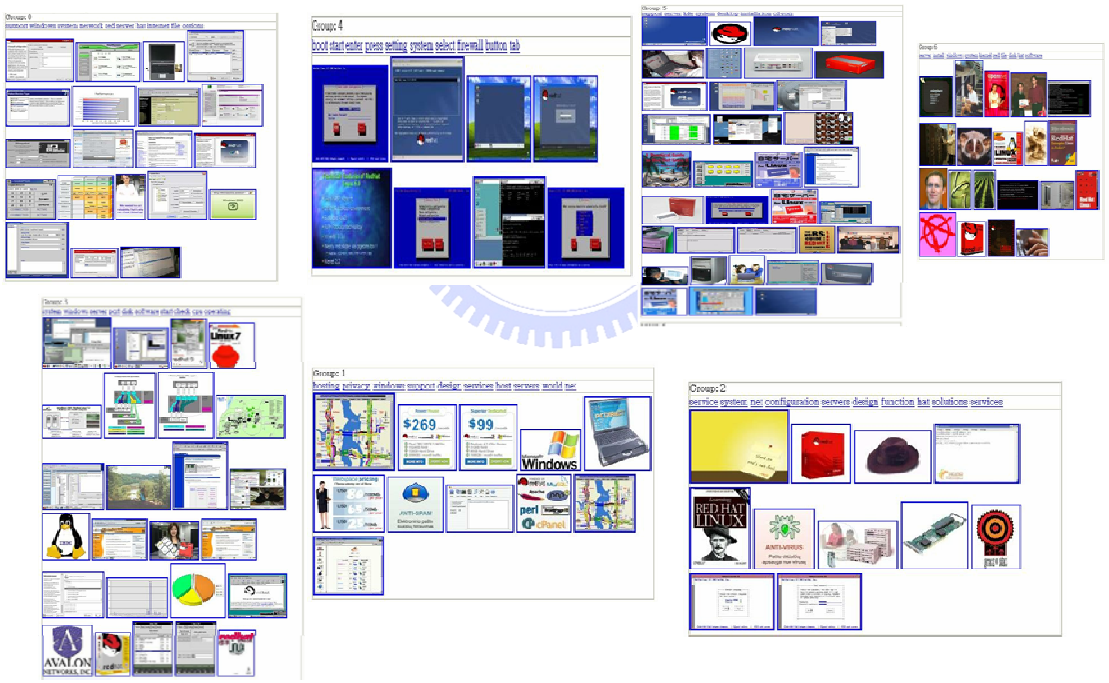

圖 60 關鍵字 redhat+建議關鍵字 linux 搜尋結果

4.2.7 測試案例 7: taiwan

關鍵字taiwan的執行結果如圖 61。

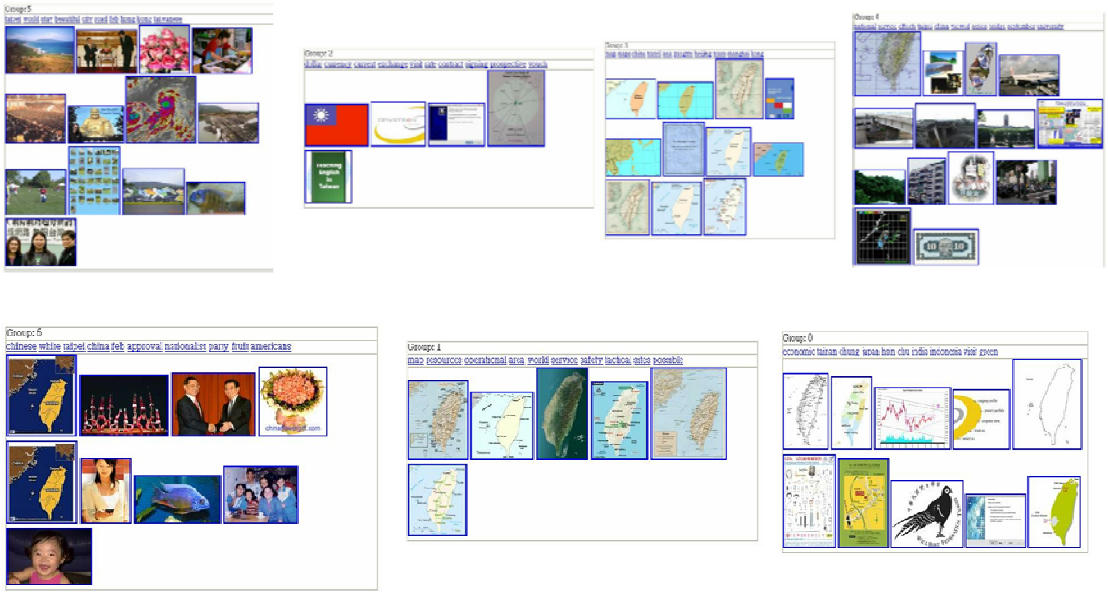

圖 61 關鍵字 taiwan 執行結果

我們將其中兩個分群結果放大觀察，如圖 62和圖 63所示。圖 62中的影像內容和台灣地圖相關，代表關鍵字包含map, area, world等。而圖 63的影像內容相當不一致，其代表關鍵字如taipei, beautiful, city, road, hong kong等，也不偏向於某個特殊的語意概念，只能說都跟taiwan有關係而已。我們還是模擬兩種不同的搜尋目標，分別點選map和beautiful兩個分群代表關鍵字，得到如圖 64和圖 65兩種不同語意的搜尋結果。其中圖 64的結果已相當精確，全部是和map概念有關的台灣地圖。而圖 65則依舊維持非常不統一的各種影像概念。

圖 62 關鍵字 taiwan 的分群結果之一，以台灣地圖為主

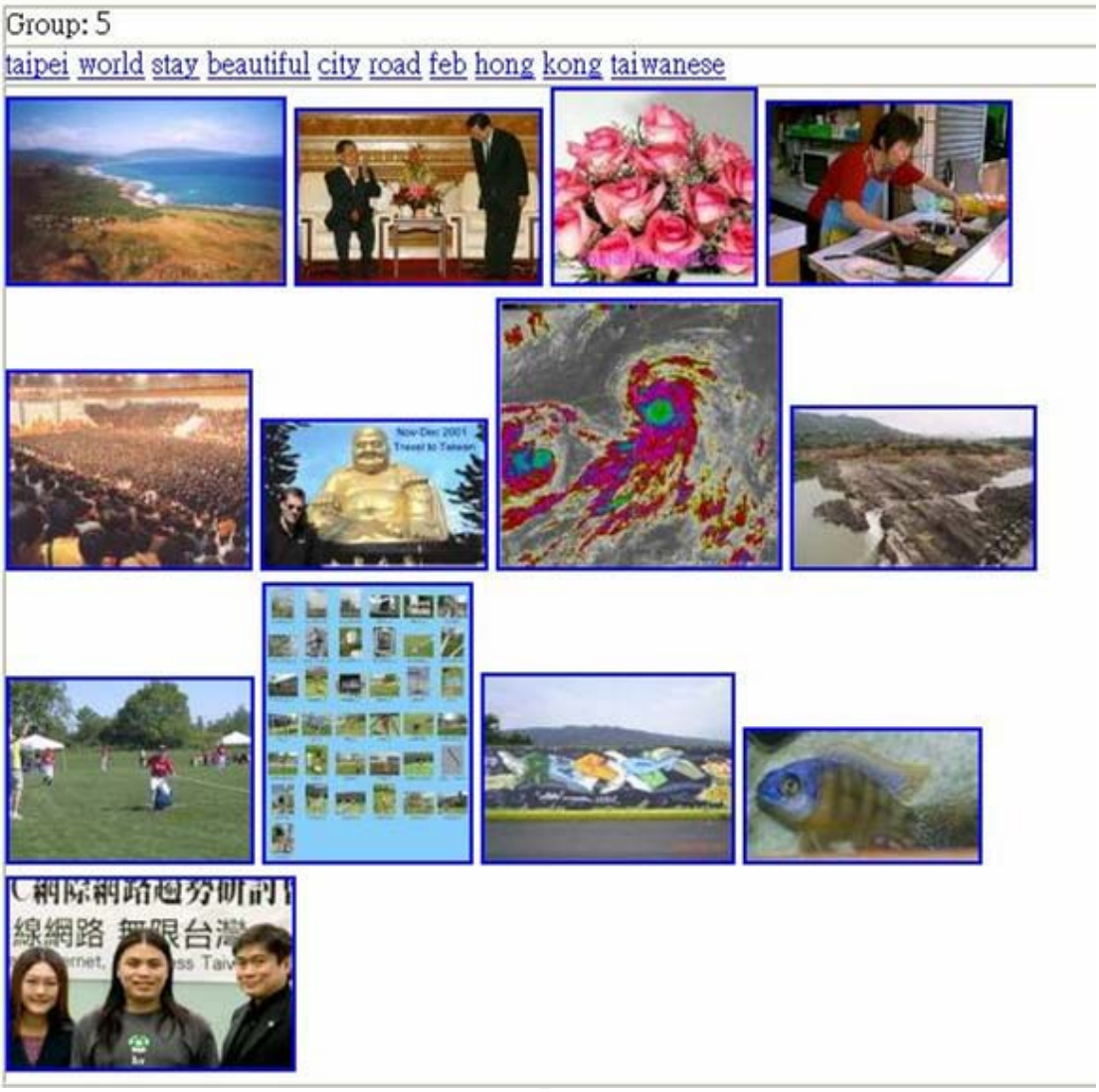

圖 63 關鍵字 taiwan 的分群結果之一，較無統一的概念

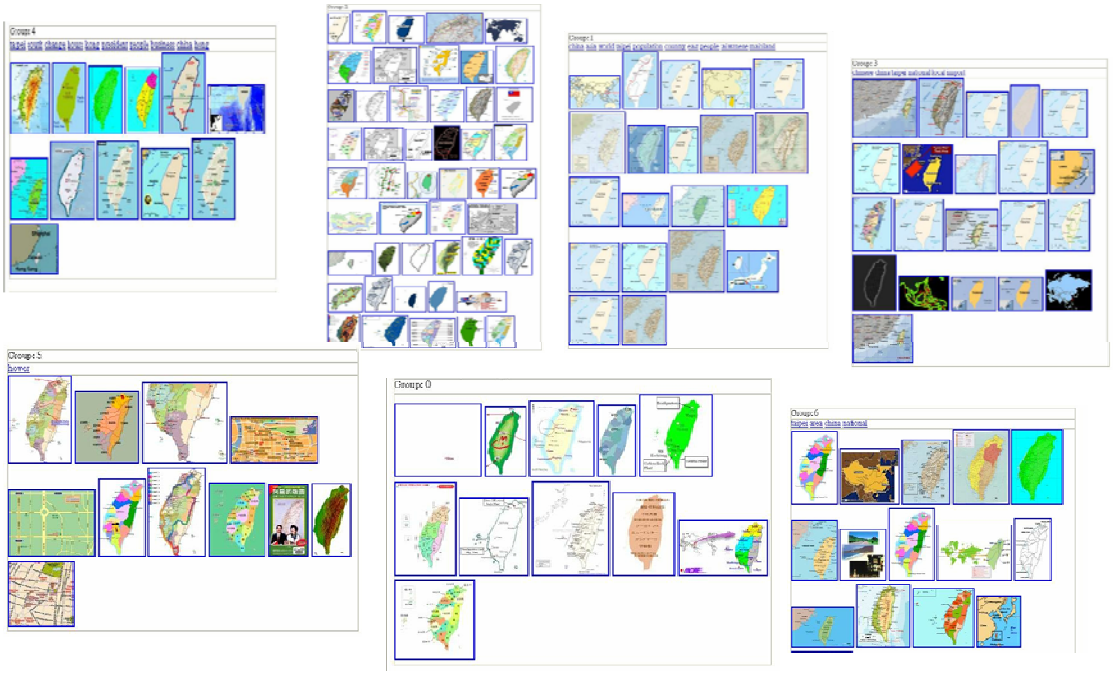

圖 64 關鍵字 taiwan+建議關鍵字 map 搜尋結果

圖 65 關鍵字 taiwan+建議關鍵字 beautiful 搜尋結果

4.3 系統雛型測試結論

由以上七個實作操作本系統的測試案例，我們可以發現，透過本系統的運作，能將本來無組織化的 QBK 搜尋結果。透過影像的視覺特徵結合關鍵字的技巧，取得較接近人類語意的搜尋結果，並透過視覺化方式表現。另外，透過關鍵字建議，加速使用者進一步取得更精確的影像搜尋結果。

在傳統的 QBK 影像搜尋系統中，使用者若想得到類似的影像搜尋結果，必須在沒有任何建議或提示下，選擇非常精確的關鍵字作搜尋，才有辦法得到一樣的搜尋結果。我們以 4.2.1 節中的 pie 關鍵字為例，若某使用者的原始搜尋目標就是找圓餅圖，在傳統的 QBK 中，他必須在穿插很多蘋果派的影像中，一張一張的找出他所要的圓餅圖。而本系統在輸入關鍵字後，即能依視覺特徵將圓餅圖與食物類的影像完全分離。使用者只需點選圓餅圖的分群即可。

第五章 結論與未來工作

在第一章中，我們曾經提到，QBK 主要是從人對於圖片的高階語意描述出發的一種圖片搜尋系統，其優點在於以人類的語意為基礎出發，並輔以發展成熟的文字檢索技術，而成為大型或商業圖片檢索應用主流。QBK 的缺點則在於圖片本身的內容對於檢索的影響可以說完全沒有，而且人類對於圖型的認知是經過長時間視覺經驗累積而成的主觀意識，因此圖片的文字描述並不能完全代表圖片本身所包含的內容。就算為圖片文字描述制定單一標準，面對來自網路的大量圖片資料庫，也需要透過大量的人力才有辦法完成。

CBIR 則是從圖片本身的視覺特徵出發的一種圖片搜尋系統，其優點在於檢索結果完全依靠圖片本身的內容為主，完全客觀。其缺點目前 CBIR 的基礎技術仍不夠成熟，無法完美的模擬人類的辨別能力，進而達到滿足人類的需求。這兩大類的系統各有其現階段的優缺點，所以本研究的主要目的，即是希望能綜合 QBK 系統和 CBIR 系統的優點，提出一個整合視覺特徵與關鍵字的圖片檢索系統，希望高低階演算法的組合，提出一個較為接近人類語義且以影像內容為基礎的圖片檢索系統。

我們將一般 QBK 系統的查詢結果，透過 CBIR 中視覺特徵的擷取，將擷取出來的特徵值，再以資料探勘中的分群演算法加以分群，以區分出代表不同語意的影像。最後加上關鍵字擷取的技術，以關鍵字建議來引導使用者作反覆式的搜尋，以找到更貼近使用者語意的搜尋目標。由第四章的 7 個關鍵字查詢結果，我們可以說本系統之運作原理是確實可行的，尤其以 windows 和 opera 等類似這類的關鍵字，

其語意有雙重以上的代表意義，而不同語意的影像的視覺特徵之間，有很明顯且特殊的視覺特徵差異。其搜尋成效會特別明顯。

而從本研究的發展過程中，我們也意外的發現了一些不同研究領域上的收獲。在傳統的搜尋過程中，我們只是從單一的搜尋需求出發，在決定了關鍵字之後所得到的搜尋結果，我們會自己過濾掉我們認為不符合此關鍵字的部分結果。但透過本系統以分群結果的方式來呈現，我們可以幫助使用者了解一個關鍵字，在不同領域中的應用情況。我們以 4.2.5 的 nano 關鍵字為例，使用者原先可能只是想找 ipod nano 這項產品的相關影像。他並不了解 nano 這個字原始的意義。而透過本系統，使用者就不難發現，原來 nano 這個字在科學領域上有著另一個意義，也能因此更了解 ipod nano 這個產品的命名由來。

在本研究的過程中，我們也發現了更多值得進一步探討的方向:

1. 如果我們可以限定影像資料庫的範圍，或事先得到所有影像資料，則我們可以透過事先擷取每一張影像的視覺特徵值與影像代表關鍵字擷取的方式，來加快使用者在執行搜尋的速度。以本研究所採用的 Google Image Search 為例，由於影像本身都是儲存在 Google 的伺服器中，每一次的查詢，我們都必須把查詢的結果一張一張從 Google 的伺服器上下載，才能開始作下一步視覺特徵值的擷取。如果我們可以事先就得到整個 Google Image Search 的影像資料庫，並擷取每一張影像的視覺特徵值並事先儲存於資料庫中，那麼對於每一次的查詢動作，我們只要將視覺特徵值由資料庫中讀出，就可以直接進入分群的階段。
2. MPEG-7 工具本身也僅是一個概念的實作雛型，在研究過程中，我們分析了 MPEG-7 工具的內部演算法，發現其中還有

很多可以作調整以加強整體系統效率的部分。比如說: MPEG-7 對於每一項特徵值的擷取，均是重新讀取一次影像檔的 raw data。而對於像我們這樣的系統，如果可以將影像讀取出來，並存放於記憶體中，再輪流交由每一個演算法來使用，雖然這樣的作法會造成系統在記憶體使用量上的增加，但對於特徵值擷取的速度，在效率有很大的加強效果。

3. 雖然影像代表關鍵字的擷取演算法並非本研究之重點，但是從研究中我們發現到，每一張影像代表關鍵字的擷取時間，約佔單次查詢的 1/3 強。由此我們可以發現，一個好的網頁關鍵字擷取技術，對本系統來說，是非常重要的基礎。好的網頁關鍵字擷取技術，可以加強系統在擷取影像代表關鍵字的效率。而且，每張影像的代表關鍵字，可以說是關鍵字建議的候選。如果我們可以採用一個更好的關鍵字擷取技術，能讓語意與視覺特徵的連結性更加強烈，對於關鍵字建議的品質提升，有非常大的幫助。舉個例子來說，如果在 4.2.1 節中，在圓餅圖的分群裡，如果系統本身沒有找出 chart 的關鍵字建議，我們可能就無法因此再進一步找到更多更精確的圓餅圖影像。

4. 在 4.2 節的各項實驗中，我們可以發現，雖然大體上來說，每一個分群，都可以找出一個獨立的語意來代表整個分群的概念。但是我們還是可以從中發現幾張在該分群中，不屬於該語意的例外影像。又或者我們可以在全部的影像中，發現某幾張根本不知道要放在那一個分群的影像。在分群演算法的研究中，有一系統關於這些問題的解法，通常我們稱之為 outlier detection。透過 outlier detection，我們可以過濾很多對

於系統幫助不大，甚至會影像分群結果的小部分影像。Outlier detection 對於最佳化分群結果，我們認為將會非常有幫助。

5. 第三章分群演算法中，我們將分群數目定為 7。從細節上看來，如果要讓使用者可以從系統中得到更完美的分群結果，我們可以從系統的使用者介面中，加入動態調整分群數目的功能。使用者如判斷某兩分群屬於同一語意，可以透過系統介面命令這兩個分群執行合併的動作。相反地，使用者若認定某一分群中，包含了二種以上的語意，也可以透過系統介面命令該分群作分裂的動作，以產生足夠多的分群數目，達到最佳的分群效果。而為了能加速使用者得到最佳分群結果，我們更可以加入建議分群合併的功能。透過分析每個分群代表關鍵字的重覆頻率，系統可以得知那幾個分群在語意上是類似的，並依此在系統介面上提出合併分群的建議給使用者，讓使用者可以透過系統的幫助，更快速的修正合適的分群數目。

6. 在本研究，我們探討了 QBK 與 CBIR 的相關研究背景，因而發現到兩者的所遇到的困境，透過加入其他領域的研究，提出一個能綜合 QBK 和 CBIR 優點的系統。但是從本研究中，我們發現視覺特徵的比例和語意比例，並不一定是相同的。例如，背景的顏色常常是一張影像中，佔最大比例的部分，然對人類的語意來說，其背景所代表的語意並不如其視覺特徵的比例那麼大。因此，我們如果可以透過一個較完整且通用於各式影像的影像分割演算法，將每張影像作適當的分割，我們就可以將本系統從以每張影像為演算的基礎單位，再細分為針對影像所包含的物件為單位。我們還可以根據物

件本身的語意，調整其視覺特徵值在本系統中的比重，進而得到一個更接近人類語意的影像檢索系統。

綜合本研究所述，我們提供了一個整合關鍵字與視覺特徵的影像檢索方法，且實作一系統雛型並驗證本研究的理論是確實可行的。本系統採用模組化的概念，任何一個小模組或演算法的改進，都可以在最快速的時間內，以不同的角度提高本系統各項效率，或提高影像檢索的搜尋品質。我們期望在未來，藉由本系統所提出的各項概念，衍生出一個更符合需求的影像檢索系統。

第六章 參考文獻

1. Niblack, C.W., et al., *QBIC project: querying images by content, using color, texture, and shape.* Proceedings of SPIE, 2003. **173**: p. 1993.
2. Jeon, J., V. Lavrenko, and R. Manmatha, *Automatic image annotation and retrieval using cross-media relevance models.* Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, 2003: p. 119-126.
3. Laaksonen, J., M. Koskela, and E. Oja, *PicSOM-self-organizing image retrieval with MPEG-7 content descriptors.* Neural Networks, IEEE Transactions on, 2002. **13**(4): p. 841-853.
4. Ma, W.Y.T. and B.S.T. Manjunath, *NeTra: A toolbox for navigating large image databases.* Multimedia Systems, 1999. **7**(3): p. 184-198.
5. Pentland, A.A., R.W.A. Picard, and S.A. Sclaroff, *Photobook: Content-based manipulation of image databases.* International Journal of Computer Vision, 1996. **18**(3): p. 233-254.
6. Smith, J.R. and S.F. Chang, *VisualSEEk: a fully automated content-based image query system.* Proceedings of the fourth ACM international conference on Multimedia, 1997: p. 87-98.
7. *Google Image Search* http://images.google.com, Google Inc.
8. Rummukainen, M., J. Laaksonen, and M. Koskela, *An efficiency comparison of two content-based image retrieval systems, GIFT and PicSOM.* Proceedings of International Conference on Image and Video Retrieval (CIVR 2003): p. 500?09.
9. Muller, H., et al., *Performance evaluation in content-based image retrieval: Overview and proposals.* Pattern Recognition Letters, 2001. **22**(5): p. 593?01.
10. Sun, Y. and S. Ozawa, *Semantic-meaningful content-based image retrieval in wavelet domain.* Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval, 2003: p. 122-129.
11. Barnard, K. and D. Forsyth, *Learning the semantics of words and pictures.* International Conference on Computer Vision, 2001. **2**: p. 408-415.
12. Manjunath, B.S., et al., *Color and texture descriptors.* Circuits and Systems for Video Technology, IEEE Transactions on, 2001. **11**(6):

p. 703-715.
13. Priese, L. and P. Sturm, *Introduction to the Color Structure Code and its Implementation.* Koblenz Marz, 2004. **4**.
14. Kasutani, E. and A. Yamada, *The MPEG-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval.* Image Processing, 2001. Proceedings. 2001 International Conference on, 2001. **1**: p. 674-677.
15. Albuz, E., E. Kocalar, and A.A. Khokhar, *Scalable color image indexing and retrieval using vector wavelets.* IEEE Transactions on Knowledge and Data Engineering, 2001. **13**(5): p. 851-861.
16. Deng, Y., B.S. Manjunath, and H. Shin, *Color image segmentation.* Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on., 1999. **2**: p. 451.
17. Shi, J. and J. Malik, *Normalized Cuts and Image Segmentation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000. **22**(8): p. 888-905.
18. Antani, S., et al., *Medical Validation and CBIR of Spine X-ray Images over the Internet.* Proceedings of SPIE, 2006. **6061**: p. 60610J.
19. Xu, X., et al., *A Hybrid Approach for Online Spine X-ray Image Retrieval Based on CBIR and Relevance Feedback.* IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE, 2005: p. 1.
20. Moustakas, J., et al., *A Two-Level CBIR Platform with Application to Brain MRI Retrieval.* Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on, 2005: p. 1278-1281.
21. Tanaka, K., M. Hirayama, and F. Kondo, *Query-based occupancy map for SVM-CBIR on mobile robot image database.* Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on, 2005: p. 868-874.
22. Dukkipati, P. and L. Brown, *IMPROVING THE RECOGNITION OF GEOMETRICAL SHAPES IN ROAD SIGNS BY AUGMENTING THE DATABASE.* Proceedings of the 3 rdIntl. Conf. on Computer Science and its Applications, June, 2005: p. 8-13.
23. Hand, D.J., *Principles of data mining*. 2001: MIT Press Cambridge, Mass.
24. Frakes, W.B. and R. Baeza-Yates, *Information retrieval: data structures and algorithms*. 1992: Prentice-Hall, Inc. Upper Saddle River, NJ, USA.
25. *QBIC* http://wwwqbic.almaden.ibm.com/, IBM Almaden Research Center.
26. *VIR Image Engine* http://www.virage.com/products/vir-irw.html,

Virage Inc. .
27. Huang, T.S., S. Mehrotra, and K. Ramchandran, *Multimedia analysis and retrieval system (MARS) project.* Proc of 33rd Annual Clinic on Library Application of Data Processing-Digital Image Access and Retrieval, 1996.
28. Ojala, T., M. Aittola, and E. Matinmikko, *Empirical evaluation of MPEG-7 XM color descriptors in content-based retrieval of semantic image categories.* Proc. 16th International Conference on Pattern Recognition, Quebec, Canada, 2002. **2**: p. 1021-1024.
29. Wong, K.M. and L.M. Po, *MPEG-7 dominant color descriptor based relevance feedback using merged palette histogram.* Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on, 2004. **3**.
30. Zhang, D. and G. Lu, *Content-based shape retrieval using different shape descriptors: a comparative study.* Multimedia and Expo, 2001. ICME 2001. IEEE International Conference on, 2001: p. 1139-1142.
31. Foulonneau, A., et al., *Geometric shape priors for region-based active contours.* Image Processing, 2003. Proceedings. 2003 International Conference on, 2003. **3**.
32. Bober, M., *MPEG-7 visual shape descriptors.* Circuits and Systems for Video Technology, IEEE Transactions on, 2001. **11**(6): p. 716-719.
33. Ryan, T.W., et al., *Image compression by texture modeling in the wavelet domain.* Image Processing, IEEE Transactions on, 1996. **5**(1): p. 26-36.
34. Won, C.S., D.K. Park, and S.J. Park, *Efficient use of MPEG-7 Edge Histogram Descriptor.* ETRI Journal, 2002. **24**(1): p. 23-30.
35. Salembier, P. and J.R. Smith, *MPEG-7 multimedia description schemes.* Circuits and Systems for Video Technology, IEEE Transactions on, 2001. **11**(6): p. 748-759.
36. Liqin, S., et al., *Edge detection on real time using LOG filter.* Speech, Image Processing and Neural Networks, 1994. Proceedings, ISSIPNN'94., 1994 International Symposium on, 1994: p. 37-40.
37. Kanopoulos, N., N. Vasanthavada, and R.L. Baker, *Design of an image edge detection filter using the Sobel operator.* Solid-State Circuits, IEEE Journal of, 1988. **23**(2): p. 358-367.
38. Panjwani, D.K. and G. Healey, *Markov random field models for unsupervised segmentation of textured color images.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995. **17**(10): p. 939-954.

39. Fowlkes, C.C., *Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues.* Learning, 2004. **26**(5): p. 530-549.
40. Jain, A.K. and R.C. Dubes, *Algorithms for clustering data*. 1988: Prentice-Hall, Inc. Upper Saddle River, NJ, USA.

附錄

1.顏色（Color）Descriptors

主要提供四種 Descriptor 來描述顏色的特徵，分別為 DominantColor、ScalableColor、ColorLayout、ColorStructure，包括了 ColorSpace 和 ColorQuantization Descriptor 兩種顏色描述的工具。每種顏色的 Descriptor 都可以用來描述任意大小影像的顏色特徵。其詳細的內容如下：

1.1 Color space

主要是在描述影像的顏色空間分布，包括了 RGB、YCrCb、HSV、HMMD、Linear transformation matrix with reference to RGB、Monochrome。DDL 的語法如下：

```
<complexType name="ColorSpaceType" final="#all">
<choice>
<element name="ColorTransMat" minOccurs="0">
<simpleType>
<restriction>
<simpleType>
<list itemType="mpeg7:unsigned16">
</simpleType>
<length value="9"/>
</restriction>
</simpleType>
</element>
</choice>
<attribute name="colorReferenceFlag" type="boolean" use="default" value="false"/>
<attribute name="type">
<simpleType>
<restriction base="string">
<enumeration value="RGB"/>
<enumeration value="YCbCr"/>
<enumeration value="HSV"/>
<enumeration value="HMMD"/>
<enumeration value="LinearMatrix"/>
<enumeration value="Monochrome"/>
</restriction>
</simpleType>
</attribute>
</complexType>
```

colorReferenceFlag：這個欄位的值為布林值（true、1 或 false、0），預設值為 false，這個值通常搭配 type=”RGB”時使用，如果其值為 1 時，則將 RGB 轉換成 CIE（Commission Internationale de l’Eclairage）XYZ color space，表格 1 為其轉換表。若其值為 0，則不變。

	Red	Green	Blue	D65
x	0.6400	0.3000	0.1500	0.3127
y	0.3300	0.6000	0.0600	0.3290
z	0.0300	0.1000	0.7900	0.3583

表格 1 Color Space

type：

‘**RGB**’：為紅色（R）、綠色（G）、藍色（B）三種光的強度所組合成的顏色空間。

‘**YCbCr**’：將 RGB 的值線性轉換成 YcbCr 的值，其公式如下：

$$Y = 0.299*R + 0.587*G + 0.114*B$$

$$Cb = -0.169*R - 0.331*G + 0.500*B$$

$$Cr = 0.500*R - 0.419*G - 0.081*B$$

‘**LinearMatrix**’：必須與 ColorTransMat 的值配合，利用線性矩陣轉換將 RGB 顏色空間轉換成相對應的 components。其公式如下：

$$C1 = ColorTransMat[0][0]*R + ColorTransMat[0][1]*G + ColorTransMat[0][2]*B$$

$$C2 = ColorTransMat[1][0]*R + ColorTransMat[1][1]*G + ColorTransMat[1][2]*B$$

$$C3 = ColorTransMat[2][0]*R + ColorTransMat[2][1]*G + ColorTransMat[2][2]*B$$

‘**HSV**’：為Hue（色調）、Saturation（飽和度）、Value（亮度）的縮寫，將RGB標準化的值非線性轉換成HSV的值。HSV在顏色空間

上的關係如圖 66，Hue是從紅色開始的旋轉角度，Saturation是從圓心到圓周的距離其值為 0 到 1，Value為其高度。轉換的公式如下：

Max = max(R, G, B);

Min = min(R, G, B);

Value = max(R, G, B);

if(Max == 0) then

Saturation = 0;

else

Saturation = (Max-Min)/Max;

if(Max == Min) Hue=0; /* achromatic */

otherwise:

if(Max == R && G > B)

Hue = 60*(G-B)/(Max-Min)

else if(Max == R && G < B)

Hue = 360 + 60*(G-B)/(Max-Min)

else if(G == Max)

Hue = 60*(2.0 + (B-R)/(Max-Min))

else

Hue = 60*(4.0 + (R-G)/(Max-Min))

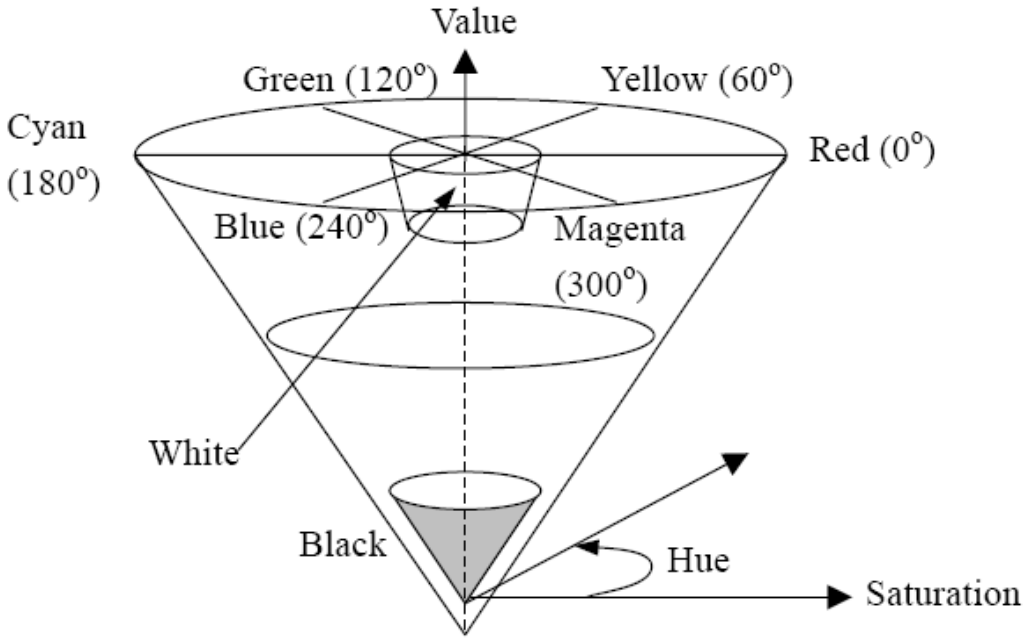

圖 66 HSV 在顏色空間上的關係

‘**HMMD**’：為 Hue、Max、Min、Difference 的縮寫，其值由 RGB 顏色空間逆向轉換而來，HMMD 包含了五個值如下：Hue（與 HSV 的 Hue（色調）相同）、Max（有多黑，指陰暗的感覺，清色（pure color）加黑變暗）、Min（有多白，指明亮的感覺，清色（pure color）加白變淡）、Diff（包含多少灰色，接近清色（pure color）的程度，指彩度（colorfulness）的感覺）、Sum（模仿顏色的明度（brightness）），其中 Hue、Max、Min 和 Hue、Diff、Sum 兩種組合都可以定義顏色空間，Hue、Max 和 Min 的轉換與 HSV 中的 Hue、Saturation 和 Value 的轉換相同，Diff 和 Sum 的轉換公式如下：

$$Diff = Max - Min;$$

$$Sum = (Max + Min)/2;$$

HMMD可由暗度（blackness）、明度（whiteness）、彩度（colorfulness）和色調（hue）所構成的雙圓錐體表達的顏色空間如圖 67。

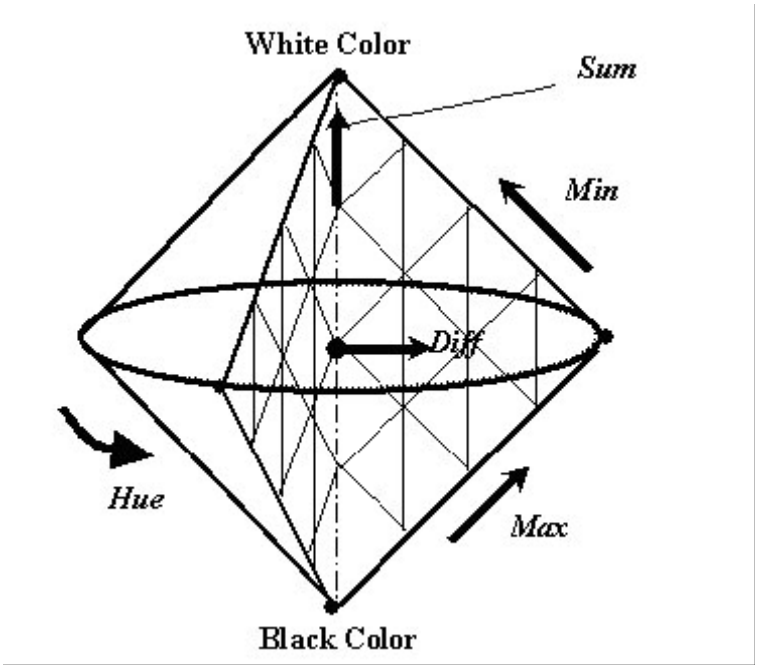

圖 67 HMMD 顏色空間

‘**Monochrome**’：黑白色階的形式，轉換公式如下：

$$Y = 0.299*R + 0.587*G + 0.114*B$$

ColorTransMat：當 type=”LinearMatrix”時，此值才出現，ColorTransMat 為一個矩陣，矩陣內元素的值為-1 到 1 之間。

1.2 Color quantization

對顏色空間定義一致性量化，可與 Dominant Color 結合，使得 Dominant Color 所表達的值更具有意義。DDL 的語法如下：

```
<complexType name="ColorQuantizationType" final="#all">
<sequence maxOccurs="unbounded">
<element name="Component">
<simpleType>
<restriction base="string">
<enumeration value="R"/>
<enumeration value="G"/>
<enumeration value="B"/>
<enumeration value="Y"/>
<enumeration value="Cb"/>
<enumeration value="Cr"/>
<enumeration value="H"/>
<enumeration value="S"/>
<enumeration value="V"/>
<enumeration value="Max"/>
<enumeration value="Min"/>
```

<enumeration value="Diff"/>
<enumeration value="Sum"/>
</restriction>
</simpleType>
</element>
<element name="**BinNumber**" type="mpeg7:unsigned12"/>
</sequence>
</complexType>

Component：指的被量化的顏色元件（Component），當顏色空間指定為Monochrome，則元件個數為一，其餘為三，當顏色空間指定為HMMD時，則元件可以是（H、Max、Min）和（H、Diff、Sum）的組合，每個顏色空間與元件的關係如表格 2。

type	Component1	Component2	Component3	Component4	Component5
RGB	R	G	B	N/A	N/A
YCbCr	Y	Cb	Cr	N/A	N/A
HSV	H	S	V	N/A	N/A
HMMD	H	Max	Min	Diff	Sum
LinearMatrix	C1	C2	C3	N/A	N/A
Monochrome	Y	N/A	N/A	N/A	N/A

表格 2 顏色空間與元件關係

BinNumber：每個顏色元件 bin 的個數。

1.3 Dominant color

這個 Descriptor 用來呈現只要用影像中局部性的顏色特徵就可以表達整張影像的顏色訊息，主要目的用在顏色的影像內容檢索（content-based retrieval）。DDL 的語法如下：

```
<complexType name="DominantColorType" final="#all">
<complexContent>
<extension base="mpeg7:VisualDType">
<sequence>
<element name="ColorSpace" type="mpeg7:ColorSpaceType" minOccurs="0"/>
<element name="ColorQuantization" type="mpeg7:ColorQuantizationType"
minOccurs="0"/>
<element name="SpatialCoherency" type="mpeg7:unsigned5"/>
<element name="Values" maxOccurs="8">
<complexType>
<sequence>
<element name="Percentage" type="mpeg7:unsigned5"/>
<element name="ColorValueIndex">
<simpleType>
<restriction>
<simpleType>
<list itemType="mpeg7:unsigned12"/>
</simpleType>
<length value="3"/>
</restriction>
</simpleType>
</element>
<element name="ColorVariance" minOccurs="0">
<simpleType>
<restriction>
<simpleType>
<list itemType="mpeg7:unsigned1"/>
</simpleType>
<length value="3"/>
</restriction>
</simpleType>
</element>
</sequence>
</complexType>
</element>
</sequence>
<attribute name="size">
<simpleType>
<restriction base="positiveInteger">
<minInclusive value="1"/>
<maxInclusive value="8"/>
</restriction>
</simpleType>
</attribute>
</extension>
</complexContent>
</complexType>
```

ColorSpace：如 1.1 所述。

ColorQuantization：如 1.2 所述。

SpatialCoherency：對每一個dominant color的空間凝聚性（spatial coherency）配置一個加權值，其值越高代表dominant color的空間凝聚性越高，其值越低則反之；如圖 68，左圖紅色的點在區塊中較為分散，所以SpatialCoherency的值較低，相反的右圖紅色的點在區塊中較為集中，所以SpatialCoherency的值較高，這個特徵通常用來做相似度的擷取。

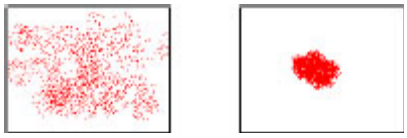

圖 68 dominate color 的空間凝聚性

Values：這個為陣列的形態，主要是由 Percentage、ColorValueIndex 和 ColorVariance 所組成。

Percentage：在區塊中 Dominant color 所用到的顏色像素所佔的百分比。

ColorValueIndex：為一個整數陣列，其值為所選擇顏色空間的 dominant color 的索引，陣列維度的大小依據所選擇顏色空間而改變。

ColorVariance：為一個整數陣列，在所定義的顏色空間中，第 j 個 color component 的變異值（相對於其它像素的顏色的值）計算方式如下：

$$CV_j = \frac{1}{N}\sum_{k=0}^{N-1}\left(m_j - p_{kj}\right)^2$$

m_j是第j個Dominant color的component，p_{kj}是第k個像素的第j個component，N是此Dominant color中全部像素的數目。其中j的維度視所定義的顏色空間而定。

size：指定了 dominant colors 在影像區塊中的個數，其最大的 dominant colors 個數為 8，最小為 1。

1.4 Scalable color

這個 Descriptor 主要是在呈現 HSV 顏色空間的 color histogram，它對於 image-to-image matching 的搜尋方式很有幫助。DDL 的語法如下：

```
<complexType name="ScalableColorType" final="#all">
<complexContent>
<extension base="mpeg7:VisualDType">
<sequence>
<element name="Coefficients" type="mpeg7:integerVector"/>
</sequence>
<attribute name="numberOfCoefficients" type="mpeg7:unsigned3"/>
<attribute name="numberOfBitplanesDiscarded" type="mpeg7:unsigned3"/>
</extension>
</complexContent>
</complexType>
```

Coefficients：主要是儲存影像在 HSV 顏色空間經由 Haar transform 轉換後的有號整數（signed integers）向量值，它量化後的精確度會隨著 numberOfBitplanesDiscarded 的值而變。

numberOfCoefficients：這個屬性主要說明在 Scalable color 中係數（Coefficients）的數目。可能的值有 16、32、64、128。

numberOfBitplanesDiscarded：這個屬性主要說明在 Scalable color 的每個係數（Coefficients）中省略的 bitplanes 的個數。可能的值有 0、1、2、3、4、6、8。假如配置給係數（Coefficients）的 bits 的數目小於 numberOfBitplanesDiscarded 指定的數目，則只有係數（Coefficients）的 sign bit 會被保留。

1.5 Color layout

以非常簡潔的方式描述顏色在空間上的配置，目的在於快速的擷取和瀏覽,它的目的在於提供 image-to-image 和 video clip-to-video clip matching（那些須要重覆計算相似度的工作），也可用在 color layout-based 擷取如 sketch-to-image matching（先用手畫一個概略的輪廓，再依此輪廓對資料庫搜尋）。DDL 的語法如下：

```
<complexType name="ColorLayoutType" final="#all">
<complexContent>
<extension base="VisualDType">
<sequence>
<element name="YCoeff">
<complexType>
<element name="YDCCoeff" type="mpeg7:unsigned6"/>
<element name="YACCoeff" type="mpeg7:acCoeffType"/>
</complexType>
</element>
<element name="CbCoeff">
<complexType>
<element name="CbDCCoeff" type="mpeg7:unsigned6"/>
<element name="CbACCoeff" type="mpeg7:acCoeffType"/>
</complexType>
</element>
<element name="CrCoeff">
<complexType>
<element name="CrDCCoeff" type="mpeg7:unsigned6"/>
<element name="CrACCoeff" type="mpeg7:acCoeffType"/>
</complexType>
</element>
</sequence>
<attribute name="numOfYCoeff" type="mpeg7:numberOfCoeffType"
use="default" value="6"/>
<attribute name="numOfCCoeff" type="mpeg7:numberOfCoeffType"
use="default" value="3"/>
</extension>
</complexContent>
</complexType>
<simpleType name="numberOfCoeffType" base="mpeg7:positiveInteger">
<enumeration value="1"/>
<enumeration value="3"/>
<enumeration value="6"/>
<enumeration value="10"/>
<enumeration value="15"/>
<enumeration value="21"/>
<enumeration value="28"/>
<enumeration value="64"/>
</simpleType>
<simpleType name="acCoeffType">
<restriction>
<simpleType>
<list itemType="mpeg7:unsigned5"/>
```

```
</simpleType>
<maxLength value="63"/>
</restriction>
</simpleType>
```

YDCCoeff、YACCoeff、CbDCCoeff、CbACCoeff、CrDCCoeff、CrACCoeff：

這些為整數陣列，儲存著 zigzag-scanned DCT（Discrete Consine Transform）係數的值。

YDCCoeff：Y component 第一個被 DCT 量化的係數。

YACCoeff：Y component 第二個和第二個以後被 DCT 量化的係數。

CbDCCoeff：Cb component 第一個被 DCT 量化的係數。

CbaACCoeff:Cb component 第二個和第二個以後被 DCT 量化的係數。

CrDCCoeff：Cr component 第一個被 DCT 量化的係數。

CrACCoeff:Cr component 第二個和第二個以後被 DCT 量化的係數。

numOfYCoeff：這個屬性指定 color component（Y）係數的數目，可能的值為 1、3、6、10、15、21、28、64，若沒有指定則預設值為 6。

numOfCCoeff:這個屬性指定 color component（Cb、Cr）係數的數目，可能的值為 1、3、6、10、15、21、28、64，若沒有指定則預設值為 3。

1.6 Color structure

主要描述影像的顏色內容（color content）與影像的顏色內容在空間的結構關係，主要功能在image-to-image matching，可以使用在靜態影像的擷取。Color structure主要是利用Structuring Element來描述顏色內容在空間的結構，如圖 69影像中color bins可記錄 8 種不同的顏

色，Structuring Element的大小是 8x8 pixels，圖 69顯示了在影像中央的一個Structuring Element，這個Structuring Element含蓋了三種顏色c1、c3、c7，於是在color bins中c1、c3、c7 分別加一，以此類推。

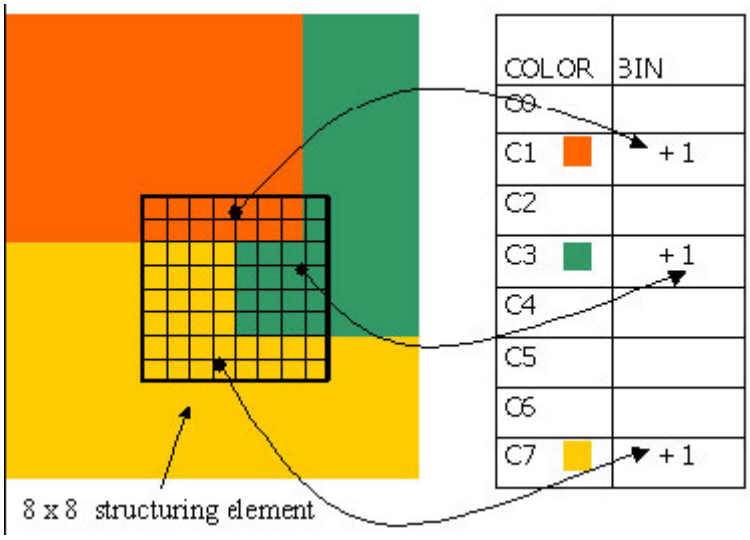

圖 69 Color Structure

圖 70顯示了兩張影像具有相同的color histogram（兩張影像中c_m的個數相等），卻有不同的Color Structure，左邊顯示了較有結構性的Color Structure，右邊則不具結構性，左邊的顏色比較集中，所以利用Structuring Element在計算color bins時，所得到的color bins值較小，因為只有在Structuring Element經過左上角時，color bins的c_m才會被加一，相反的右邊的顏色非常的分散，幾乎所有的Structuring Element都會含蓋c_m（color bins的c_m值每一次都會被加一）。

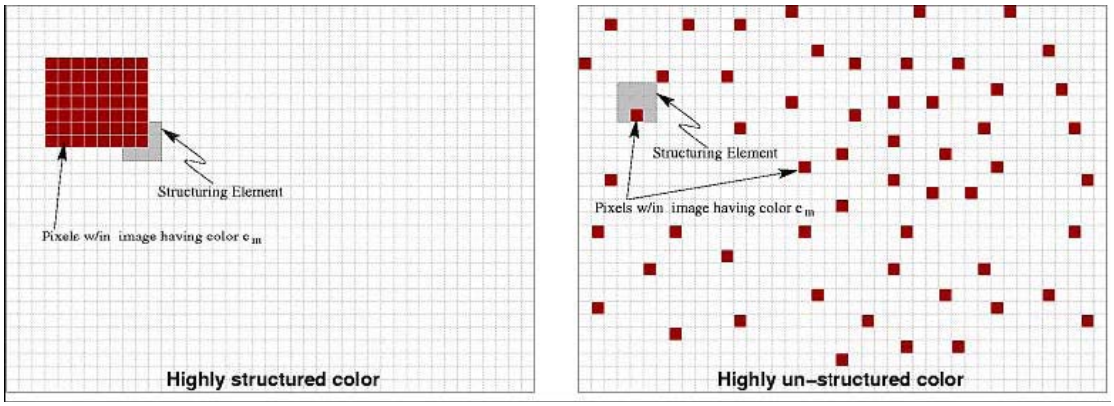

圖 70 相同 Color Histogram，不同的 Color Structure

DDL的語法如下：

```
<complexType name="ColorStructureType" final="#all">
<complexContent>
<extension base="mpeg7:VisualDType">
<sequence>
<element name="Values" >
<simpleType>
<restriction>
<simpleType>
<list itemType="mpeg7:unsigned8"/>
</simpleType>
<minLength value="32"/>
<maxLength value="256"/>
</restriction>

</simpleType>
</element>
</sequence>
<attribute name="colorQuant" type="mpeg7:unsigned3" use="required"/>
</extension>
</complexContent> </complexType>
```

Values：儲存Color Structure數值。

colorQuant：這個屬性指定Values在HMMD的顏色空間的數目。

2. 紋路（Texture）Descriptors

主要有 Homogeneous Texture 和 Edge Histogram Descriptors。

2.1 Homogeneous texture

使用the energy and energy deviation（in a set frequency channels）來描述區域性的紋路特徵，可以應用在相似度的搜尋與擷取。

Homogeneous texture在影像中的紋路徵主要是以頻率空間（frequency space）來表示，頻率空間在角度上以 30 度為單位切割成六等份，在半徑上用octave division切割成五等份，總共切割成 30 個頻率空間稱為特徵頻道（feature channels），如圖 71中的C_i（i=1 到 30）。在正規化的頻率空間（0≤ω≤1）中，每個特徵頻道在角度上的中點頻率以 30 度為一單位區隔，表示方法為$\theta_r = 30° \times r$，其中$r \in \{0, 1, 2, 3, 4, 5\}$，也就是每個特徵頻道離起始頻道的角度大小，如$C_6$離$C_1$有 30x5=150 度（$C_1$是起始頻道）；每個特徵頻道在半徑上的中點頻率以octave scale為單位區隔，表示方法$\omega_s = \omega_0 \cdot 2^{-s}, s \in \{0,1,2,3,4\}$，也就是每個特徵頻道在半徑的中點到圓點的距離，$\omega_0$為距離圓心最遠的距離為 3/4，如$C_1$的中點距離圓心$\omega_0 x2^{-0}=3/4x1=3/4$，$C_7$的中點距離圓$\omega_0 x2^{-1}=3/4x1/2=3/8$；而特徵頻道在半徑上的octave頻寬以$B_s = B_0 \cdot 2^{-s}, s \in \{0,1,2,3,4\}$表示，$B_0$代表最大的頻寬為 1/2，如$C_1$在半徑上的頻寬長度為$B_0 x2^{-0}=1/2x1=1/2$，$C_7$在半徑上的頻寬長度為$B_0 x2^{-1}=1/2x1/2=1/4$，r、s與特徵頻道索引i的關係為$i = 6 \times s + r + 1$，整個關係如圖 71。

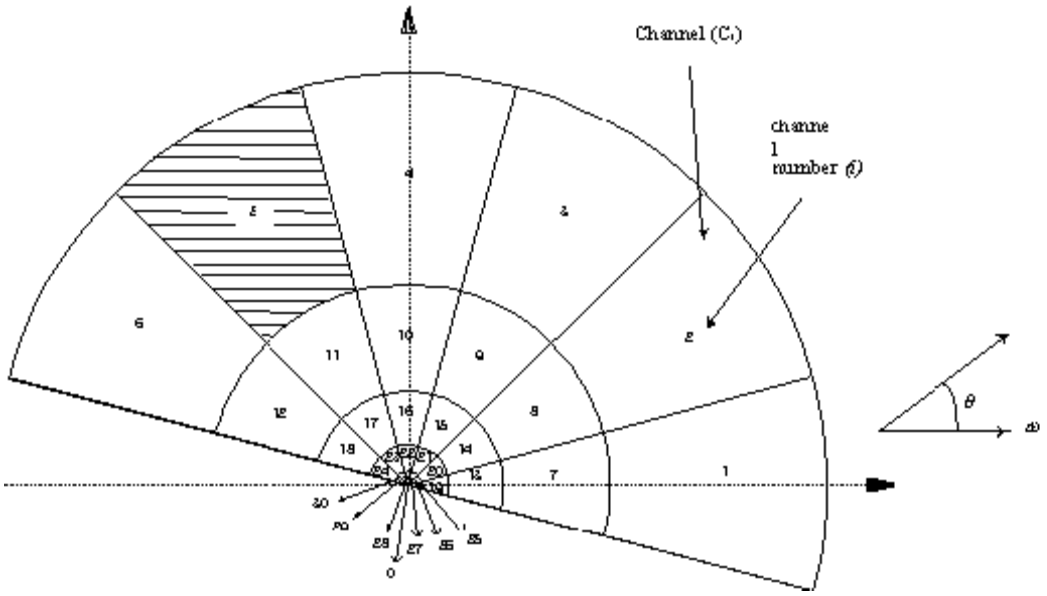

圖 71 Homogeneous Texture

在最上層的特徵頻道運用 2D Gabor function 來計算如下：

$$G_{P_{s,r}}(\omega,\theta)=\exp\left[\frac{-(\omega-\omega_s)^2}{2\sigma_{\rho_s}^2}\right]\cdot\exp\left[\frac{-(\theta-\theta_r)^2}{2\sigma_{\theta_r}^2}\right]$$

在鄰近特徵頻道（與本身有接觸）的Gabor function（在角度和半徑上）一半的最大值決定了Gabor function的標準差（standard deviations），在角度上 σ_{θ_r} 為一個常數 $15^\circ/\sqrt{2\ln 2}$，在半徑上 σ_{ρ_s} 的值依賴著octave頻寬而變，其值為 $\sigma_{\rho_s}=\frac{B_s}{2\sqrt{2\ln 2}}$，Gabor function與特徵頻道的關係如表格 3和表格 4。

Radial index (s)	0	1	2	3	4
Center frequency (ω_s)	$\frac{3}{4}$	$\frac{3}{8}$	$\frac{3}{16}$	$\frac{3}{32}$	$\frac{3}{64}$
Octave bandwidth (B_s)	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$
σ_{ρ_s}	$\frac{1}{4\sqrt{2\ln 2}}$	$\frac{1}{8\sqrt{2\ln 2}}$	$\frac{1}{16\sqrt{2\ln 2}}$	$\frac{1}{32\sqrt{2\ln 2}}$	$\frac{1}{64\sqrt{2\ln 2}}$

表格 3 Gabor function 與特徵頻道關係

Angular index (r)	0	1	2	3	4	5
Center frequency (θ_r)	0°	30°	60°	90°	120°	150°

Angular bandwidth	30°	30°	30°	30°	30°	30°
σ_{θ_r}	$\frac{30^\circ}{2\sqrt{2\ln 2}}$	$\frac{30^\circ}{2\sqrt{2\ln 2}}$	$\frac{30^\circ}{2\sqrt{2\ln 2}}$	$\frac{30^\circ}{2\sqrt{2\ln 2}}$	$\frac{30^\circ}{2\sqrt{2\ln 2}}$	$\frac{30^\circ}{2\sqrt{2\ln 2}}$

表格 4 Gabor function 與特徵頻道關係

第ith個特徵頻道的energy 被定義成一個影像的Gabor-filtered transform係數平方的log-scaled總和，公式如下：

$$e_i = \log_{10}[1 + p_i]$$

where

$$p_i = \int_{\omega=0^+}^{1} \int_{\theta=0^{0+}}^{360^0} [G_{P_{s,r}}(\omega,\theta) \cdot P(\omega,\theta)]^2$$

P(ω,θ)是代表著在polar frequency domain影像的Fourier transform，也就是 $P(\omega,\theta) = F(\omega\cos\theta, \omega\sin\theta)$ ，其中 $F(x,y)$ 是在Cartesian coordinate system的Fourier transform，第ith個的energy誤差 d_i 被定義成一張影像的Gabor-filtered Fourier transform係數平方的log-scaled標準差，其公式如下：

$$d_i = \log_{10}[1 + q_i]$$

where

$$q_i = \sqrt{\int_{\omega=0^+}^{1} \int_{\theta=0^{0+}}^{360^+} \left\{ \left[G_{P_{s,r}}(\omega,\theta) \cdot P(\omega,\theta) \right]^2 - p_i \right\}^2}$$

最後 Homogeneous texture Descriptor 由影像強度的平均值與影像強度的標準差構成了特徵頻道的 energies e_i 和特徵頻道的 energy d_i 誤差。

DDL 的語法如下：

```
<complexType name="HomogeneousTextureType" final="#all">
<complexContent>
<extension base="mpeg7:VisualDType">
<sequence>
<element name="Average" type="mpeg7:unsigned8"/>
<element name="StandardDeviation" type=”mpeg7:unsigned8"/>
<element name="Energy" type="mpeg7:textureListType"/>
<element name="EnergyDeviation" type="mpeg7:textureListType" minOccurs="0"/>
</sequence>
</extension>
</complexContent>
</complexType>
<simpleType name="textureListType">
<restriction>
<simpleType>
<list itemType="mpeg7:unsigned8"/>
</simpleType>
<length value="30"/>
</restriction>
</simpleType>
```

Average：這個元素代表影像像素強度的平均值。

StandardDeviation：這個元素代表影像像素強度的標準差。

Energy：這個元素儲存著特徵頻道的 energy 的值。

EnergyDeviation：這個元素儲存著特徵頻道的 energy 的標準差。

2.2 Edge histogram

主要是在呈現局部影像區塊上五種形式線條的特徵，如圖 72 Edge Histogram有四個方向性和一個無方向性的edge，原始的影像被切割成 4x4 的子圖（sub-image）（沒有重疊）如圖 73，每個子圖（sub-image）都用五種形式的edge來表示，也就說每個子圖（sub-image）產生 5 個bins的edge histogram，就整張影像來說有 16 個子圖（sub-image）共有 16*5=80 的histogram bins，子圖（sub-image）又被切割成更小的image-blocks，而image-block是用來擷取edge形式的資訊。

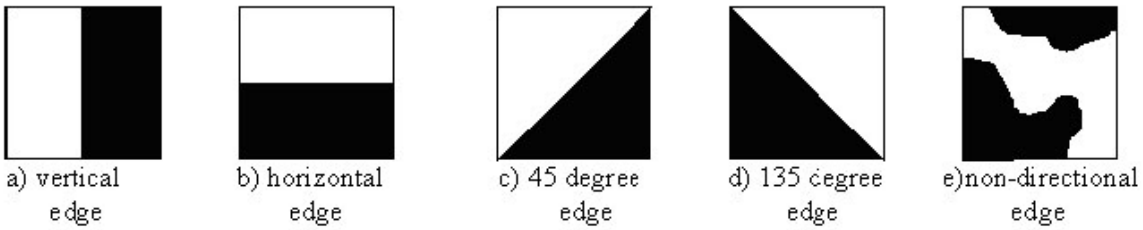

圖 72 Edge Histogram

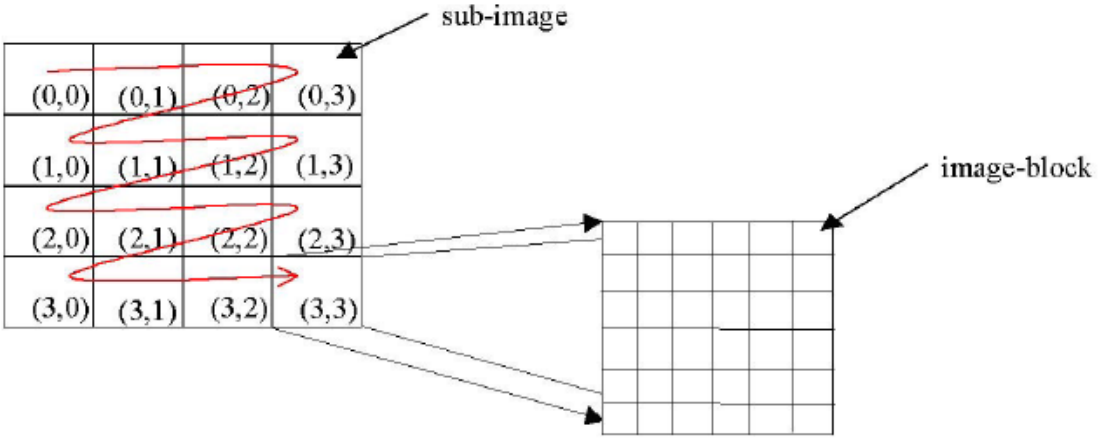

圖 73 Edge Histogram

DDL 的語法如下：

```
<complexType name="EdgeHistogramType" final="#all">
<complexContent>
<extension base="mpeg7:VisualDType">
<sequence>
<element name="BinCounts">
<simpleType>
<restriction>
<simpleType>
<list itemType="mpeg7:unsigned3"/>
</simpleType>
<length value="80"/>
</restriction>
</simpleType>
</element> </sequence>
</extension>
</complexContent> </complexType>
```

BinCounts：用來儲存 80 個histogram bins的資料如表格 5。

BinCounts[k]	Semantics
BinCounts[0]	Vertical edges in sub-image (0,0)
BinCounts[1]	Horizontal edges in sub-image (0,0)
BinCounts[2]	45 degree edges in sub-image (0,0)
BinCounts[3]	135 degree edges in sub-image (0,0)
BinCounts[4]	Non-directional edges in sub-image (0,0)
BinCounts[5]	Vertical edges in sub-image (0,1)
•	•
BinCounts[74]	Non-directional edges in sub-image (3,2)
BinCounts[75]	Vertical edges in sub-image (3,3)
BinCounts[76]	Horizontal edges in sub-image (3,3)
BinCounts[77]	45 degree edges in sub-image (3,3)
BinCounts[78]	135 degree edges in sub-image (3,3)
BinCounts[79]	Non-directional edges in sub-image (3,3)

表格 5 Histogram bins

3. 外形（Shape）Descriptors

主要有二種 Region shape、Contour shape。

3.1 Region shape

主要是在描述影像所有的像素的配置的位置，許多同質的像素構成一個區域（Region），而單一或多個區域構成外形，它可以描述單一相連的區域構成的外形如圖 74 (a)、(b)，或區域內有洞圖 74(c)，或多個不相連的區域圖 74 (d)、(e)，除了這些以外它也可以描述物邊緣較小的變形，如圖 74 (g)、(h)、(i)都是杯子，(g)的握把下方有裂縫，(i)的握把是填滿的，但Region shape Descriptor認為(g)和(h)是相同的，而(i)是不同的；(j)、(k)、(l)是影片中兩個圓盤分離的影像Region shape Descriptor會把它們都視為一樣是兩個圓盤。

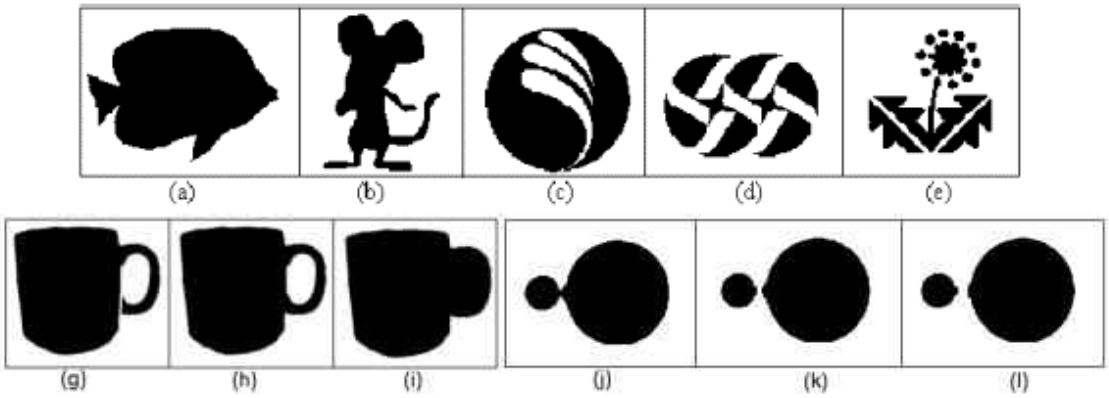

圖 74 Region Shape

Region shape Descriptor 利用 ART (Angular Radial Transform)係數的集合。ART 是定義單位圓盤（unit disk）的極向座標（polar coordinates）的複雜 2-D 轉換過程。

$$F_{nm} = \langle V_{nm}(\rho,\theta), f(\rho,\theta) \rangle = \int_0^{2\pi} \int_0^1 V_{nm}^*(\rho,\theta), f(\rho,\theta) \rho d\rho d\theta$$

$f(\rho,\theta)$ 是一個在 polar coordinates 的 image 函數（function），$V_{nm}(\rho,\theta)$ 是 ART 的基本函數，ART 的基本函數是延著角度（angular）和半徑（radial）的方向被分離出來的如下：

$$V_{nm}(\rho,\theta) = A_m(\theta)R_n(\rho)$$

角度和半徑的基本函數定義如下：

$$A_m(\theta) = \frac{1}{2\pi}\exp(jm\theta)$$

$$R_n(\rho) = \begin{cases} 1 & n = 0 \\ 2\cos(\pi n\rho) & n \neq 0 \end{cases}$$

總共有 12 個角度和 3 個半徑函數被使用，如圖 75。

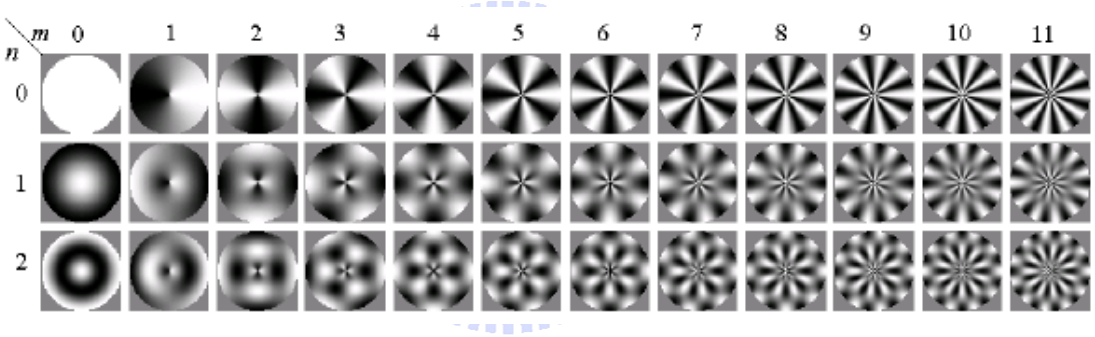

圖 75 Region Shape

DDL 的語法如下：

```
<complexType name="RegionShapeType" final="#all">
<complexContent>
<extension base="mpeg7:VisualDType">
<element name="ArtDE"> <simpleType>
<restriction base="mpeg7:listOfUnsigned4Type"> <length value="35"/>
</restriction>
</simpleType>
</element>
</extension>
</complexContent>
</complexType>
```

ArtDE：這個元素存放著 35 個 4 bits的整數陣列，35 個數值對應（如之前所述）的關係如表格 6，k=0 到 34，扣除掉n=0 和m=0。

k	0	1	2	3	4	…	30	31	32	33	34
n	1	2	0	1	2	…	1	2	0	1	2
m	0	0	1	1	1	…	10	10	11	11	11

表格 6 Region Shape

3.2 Contour shape

Contour shape Descriptor是利用封閉的曲線來描述2-D物件的輪廓。Contour shape Descriptor利用Curvature Scale Space(CSS)來呈現輪廓的形狀。Nsamples等距的點集用來建立輪廓外形CSS的描述，Nsamples是從輪廓線條上的任意點開始延著順時鐘的方取等距離的點，Nsamples點集的x座標和y座標形成X和Y座標序列，然後輪廓上X、Y序列逐步被平滑（藉著重覆利用low-pass filter with the kernel（0.25,0.5,0.25）到X、Y座標序列），由於平滑的過程使輪廓凹的部份逐漸變平，直到變凸的為止；filtering的過程也因為輪廓變凸而停止。圖 76顯示了輪廓發展過程與CSS image的關係。

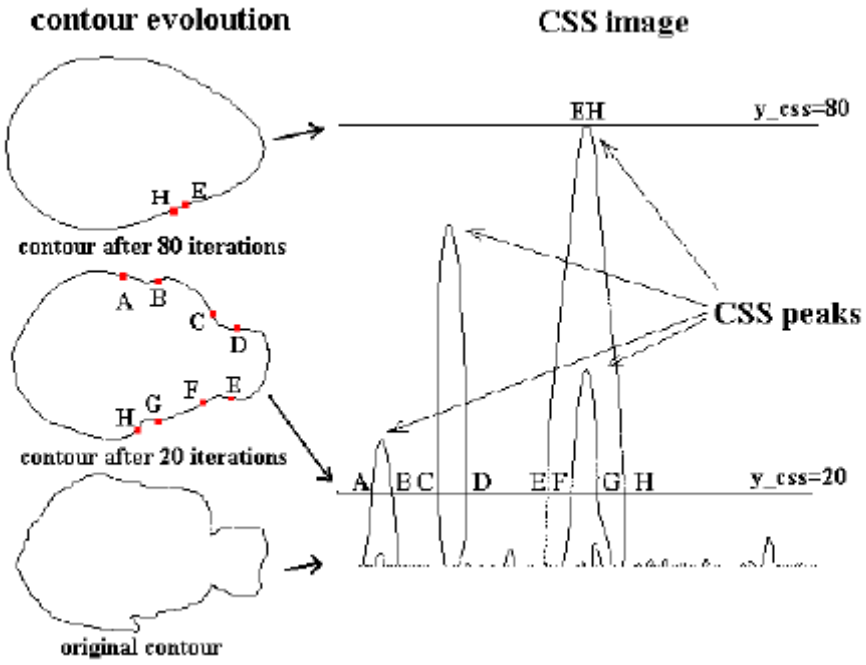

圖 76 Contour Shape

水平座標（x_css）代表 Nsamples 點集（1,…,Nsamples），垂直座標(y_css)代表 filtering 的次數,每一次平滑過程都會計算 curvature function 的 zero-crossings，zero-crossings 的點區分出輪廓凹與凸的部份，當有 zero-crossing 發生時，y_css=k 的水平線（指的是第 k 次被 filtering 平滑的輪廓)就被標誌住,這些被標誌住的水平線組成了 CSS image，在水平線上 x_css 座標也就是 zero-crossing 的位置。在 CSS image 中凸起的尖端座標（x_css,y_css）被稱為 peaks，Figure 38 的左邊顯示了最初的輪廓和被 filtering 20 次與 80 次的輪廓，右邊顯示了 y_css=20 的水平線和 8 個 zero-crossing 點（A,B,…,H）。

DDL 的語法如下：

```
<complexType name="ContourShapeType" final="#all">
<complexContent>
<extension base="mpeg7:VisualDType">
<element name="GlobalCurvatureVector" type="mpeg7:curvatureVectorType"/>
<element name="PrototypeCurvatureVector" type="mpeg7:curvatureVectorType"
minOccurs="0"/>
<element name="HighestPeak" type="mpeg7:unsigned7"/>
<element name="Peak" maxOccurs="62">
<complexType>
<element name="xpeak" type="mpeg7:unsigned6"/>
<element name="ypeak" type="mpeg7:unsigned3"/>
</complexType>
</element>
<attribute name="numberOfPeaks" type="mpeg7:unsigned6"/>
</extension>
</complexContent>
</complexType>
<simpleType name="curvatureVectorType">
<restriction base="mpeg7:listOfUnsigned6Type">
<length value="2"/>
</restriction>
</simpleType>
```

GlobalCurvatureVector：這個元素指定了輪廓全域的參數，也就是 Eccentricity 和 Circularity。

PrototypeCurvatureVector：這個元素指定了 prototype contour 的 Eccentricity 和 Circularity，prototype contour 指的是曲線利用 filtering 的平滑過程直到變成凸的輪廓。

HighestPeak：這個元素指定了最高峰的 peak 的參數。

Peak: HighestPeak、 xpeak、ypeak。

numberOfPeaks：這個屬性主要是描述在 CSS image 中 peaks 的個數。