

以圖像內容為基礎的重製圖像偵測與追蹤系統 – 使用概似最近搜尋法

邱一航 江昕鴻 胡鈞智 蕭人豪

中央研究院 資訊科學研究所

{yhchiu, cblta, s922611, jenhao}@iis.sinica.edu.tw

摘要

近年來，各種數位內容 (Digital Content) 被非法複製、傳播、使用的情形氾濫，使得數位內容創作者或提供者均不願釋出數位內容，以提供更進一步的加值應用，所以數位權利管理 (Digital Rights Management) 技術近年來受到廣泛的討論與注意。為了打擊對數位內容非法使用的行為，對受保護的數位內容作有效地偵測與追蹤，為現今不可或缺的措施。本論文提出以圖像內容為基礎的重製圖像偵測與追蹤系統 (Content-Based Image Detection and Tracking System, CBIDTS)，是一種網路重製圖像偵測與追蹤系統，主要利用圖像辨識的技術，搜尋網路上的任一相似圖像的網路使用者，是否合法註冊，但是由於網際網路上的圖像量過於龐大，每一張圖像都比對的方式太過耗時，所以系統中利用概似最近搜尋法 (Approximate Nearest Neighbor, ANN) 來改進搜尋相近圖像的速度。

關鍵詞：數位版權管理 (Digital Rights Management, DRM)、重製圖像的偵測與追蹤 (Copy Detection and Tracking and Tracking)。

1. 前言

1.1 數位內容概況

由於資訊科技的快速發展及資訊的數位化，電子數位產品不斷出現，透過網際網路無遠弗屆的便利性，使得所有數位資料得以快速方便的複製與傳遞大量的文字，繪畫與傳統媒體等均被轉換成數位檔案儲存，也因此產生出大量的數位內容 (Digital Content)。數位內容係指將圖像、文字、影像、語音等運用資訊科技加以數位化並整合運用之產品或服務。依照領域別可分為數位影音應用、電腦動畫、數位遊戲、行動應用服務、數位學習、數位出版典藏、內容軟體、網路服務等，如圖 1[1]。

數位內容在形式上有別於傳統的有形著作，必須面臨許多不可避免的問題與挑戰。經由電子化檔案形式存在，可讓各類影音、文件出版品與錄音著作的儲存媒介與電腦的儲存媒介相互整合，但也可以輕易地從網路上獲得各類影音及檔案，更可以毫不受限地重製受到著作權保護的著作，造成了著作權人利益的損害。

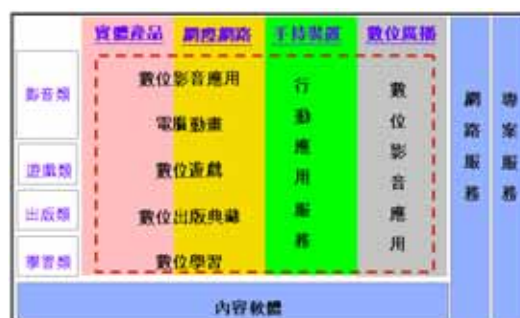


圖 1 數位內容領域範疇

(資料來源：資策會 MIC，2004 年 6 月)

數位版權管理 (Digital Rights Management, DRM) 技術近年來引起了廣泛的討論與注意，其所提出之數位內容保護架構，提供了著作權人一個可靠的數位智財保護方案，主要提出下列三項保護方向[10]：

- (1) 避免數位智財未經授權的複製濫用。
- (2) 有效的數位智財控管。
- (3) 侵權行為的偵測與追蹤。

1.2 數位版權管理技術

一般而言，目前數位權利管理技術架構主要針對密碼學、數位浮水印及權利語言三大技術作研究[11]。密碼學技術用來限制數位內容的存取，並進一步達到複製保護 (Copy Protection)；數位浮水印技術用來嵌入隱藏的版權資訊，將來若發生版權爭議時，就可以透過此一技術，將嵌入在數位內容中的認證資訊取出，作為版權認證的依據；權利語言則針對權利的種類 (使用者想要對數位內容進行什麼動作) 及權利的屬性 (允許的次數、允許的時間、授權使用的對象...等) 來訂定傳遞使用者相對應於數位內容的使用權利範圍的規則。

不過目前數位權利管理技術至今尚未完全成熟，數位浮水印在過去曾被視為數位智財保護的完整解決方案，但在今日各種不同的需求與應用中，單純的數位浮水印技術在數位內容的保護上顯得力有未逮，因為數位浮水印技術易受各種圖像處理攻擊，倘若要求數位浮水印對於目前成千上百種圖像攻擊方式都具有強韌性是過於苛求的[10]。因此，數位內容偵測與追蹤儼然成為最後一道防線。

目前並未出現比較好的網路重複數位內容偵測系統，而 CBIDTS 的概念就是網路重製圖像的偵測與追蹤系統，主要利用網路機器人的技術，搜尋網路上的任一相似圖像，以偵測網路使用者是否經過合法註冊。其相似圖像的判斷則是利用圖像本身不同位置顏色變化等圖像的獨特資訊，作為圖像的特徵值，來分辨圖像間的相似程度，進而找出可疑的非法使用者。

2. 相關文獻探討

2.1 網路機器人

網路機器人 (Spider, 又稱 Robot 或 Crawler), 是專門用來搜集網際網路上各種網頁資料與檔案的一種軟體系統, 能自動地在網際網路上搜索與偵測, 並將搜集的訊息傳給伺服器。一般而言, 網路機器人從給定的初始 URL 位址出發, 根據其原始 HTML 中指向其它 URLs 位址的超連結而搜尋其它的 URLs, 進而不斷地探詢和擴展其他網路位置, 基本上會遍佈整個網路。搜索演算法主要有深度優先和廣度優先兩種基本的搜索策略, 其演算法的主要步驟可以如下描述:

(1) 網路機器人從給定 URLs 列表中取出 URL, 並從所取出 URL 中獲取其 HTML 文檔內容。

(2) 從 HTML 文檔中擷取需要之檔案或網頁資料, 並放入資料收集的資料庫中。

(3) 從 HTML 文檔中找尋指向其它位置的 URL, 並加入到 URLs 列表中。

(4) 重複以上三個步驟, 直到沒有再找到新的 URLs 或超出了一定限制條件 (如時間或硬碟空間的限制) 就停止。

2.2 重製圖像的偵測與追蹤技術 (Copy Detection and Tracking)

在數位智財保護, 近年來被廣泛的討論, 對於數位內容的非法使用, 必須有效的偵測與追蹤。目前在保護數位圖像主要的技術有浮水印技術, 與以圖像本身為基礎的重製圖像的偵測與追蹤技術。浮水印技術是指在數位圖像散佈之前, 對於所有數位圖像增加額外的資訊, 使日後能有效的擷取出資訊, 來證明數位圖像所有權。以圖像為基礎的重製圖像的偵測與追蹤技術, 則是不用增加額外的資訊, 直接針對圖像本身的獨特內容資訊, 來辨識出近似的圖像的方式。

數位浮水印技術有從以展頻通訊觀念來嵌入浮水印 (Spread Spectrum Watermark), 到利用向量投影概念的量化浮水印 (Quantization Watermarking), 從抽取浮水印需要原始數位內容資訊到盲目偵測 (Blind Detection) [2] 等等不同方式嵌入獨特的資訊到圖像之中。

重製圖像的偵測與追蹤技術, 主要是利用樣式辨識中圖像辨識技術, 對於需保護的圖像, 經演算法擷取圖像的特徵值, 而此圖像的特徵值必須具有

獨特性, 不同的圖像必須會產生不同的特徵值, 所以圖像之間, 就以此特徵值的資訊相互比對, 當圖像特徵值之間的距離達到所接受的差距, 則視為此圖像為相似的。

目前不論是數位浮水印技術, 或是以圖像本身為基礎的重製圖像的偵測與追蹤技術, 當在第三方對於原始圖像做修改時, 例如對原始圖像檔案格式的轉換, 或是做圖像幾何的改變, 如鏡射、裁減、調整大小、旋轉等其他修改, 都會降低上述技術的強韌性, 也就是造成對於相似或相同圖像的正確辨識率下降。所以目前的研究, 就是針對這些圖像的攻擊, 找出應對演算法加以克服, 增加演算法的強韌性及安全性。其中在樣式辨識中, 以圖像辨識技術找出重製圖像的方法中, 擷取圖像特徵值的方式大略有下列幾種:

2.2.1 以圖像顏色資訊

在圖像處理上, 每個像素顏色常以 RGB 模式或 YCbCr 模式表示各種顏色。將顏色以 RGB 模式或 YCbCr 模式量化後, 就可依量化值以數理統計方式來比對圖像之間量化值的相似程度, 如顏色直方圖。不過此方法, 只記錄原圖的顏色, 所以當發生強烈顏色變化或圖像遭裁剪時, 辨識率極差, 且當顏色相近時, 但非重製的圖像中, 可能會有判別為一樣的情況。所以此方法在重製圖像偵測上效果並不是很好。

2.2.2 以圖像形狀資訊

表示形狀方式藉由樣式辨識方式來萃取其特徵點, 來代表圖像中的形狀。特徵點不僅能夠描繪形狀整體形式, 就像面積、外形、主要軸線定位等, 而且也能表示形狀邊界區域元素, 像形狀尖角、邊的獨特點等, 因此, 在這種方法上是考慮找出形狀特徵空間上有效點。進而找出兩形狀間相似程度, 可以經由數學距離公式計算出形狀上兩點間差距或是以機率統計方式算出近似程度[7]。在此方法, 當圖像作旋轉, 鏡射時能有效偵測, 不過在圖像中物體有類似形狀, 可能會發生判別成一樣, 且在計算的時間上通常會較長。

2.2.3 以圖像內容資訊

以圖像內容, 則是綜合顏色與形狀與相對位置, 為組成元素做為圖像中的特徵值。此方法是以樣式辨識方法判別圖像中各式組成物件, 其中每個物件因顏色變化可判別出物件邊界的點之向量, 來代表此圖像的特徵點, 而圖像之間則由每個特徵點構成相對位置與方向來判定相似度。如兩張圖像之間, 計算所有代表圖像的特徵點的幾何距離, 當特徵點的總差距甚小, 意味著這兩張圖像相似。其中擷取特徵值有 RIME[9]的方式, 來計算出特徵值。其主要是以離散小波變換 (Discrete Wavelet Transform, DWT), 針對每個圖像計算, 存放整體圖像的統計資料, 在單純對圖像調整大小、改變色系或是改變格式下有很好的效果, 不過在當圖像被裁

減、旋轉、水平翻轉等幾何改變，常會被視為不相符。除了以 RIME 的演算法外，另外以圖像之地區性描述子 (local descriptors) 相關演算法，如 Harris corner detector [5]、Lowe's Difference of Gaussian (DoG) detector[16]等演算法，來萃取特徵點，來加強辨識圖像中物件的特徵向量，而此方法對於重製圖像的辨識效果良好，對於大部分圖像攻擊都具有強韌性，不過由於其特徵值維度通常較大，相對於找尋相似圖像時間上的運算也較為久。

2.3 概似最近搜尋法 (ANN)

在多維向量空間下，每個空間上的點之間相近程度，是以距離的遠近作為代表，距離較近為較相近，距離較遠為較不相近。其中在 n 維空間上，判斷某一點與空間上所有存在點的相近點，可以利用線性搜尋法或是概似最近搜尋法等方式來比較。線性搜尋法，也就是利用計算點與點之間幾何距離的方式來比對空間上的所有存在的點，而一個點會有 k 個維度，若有 n 個存在的點，進行一次搜尋相近點 (Nearest Neighbors) 的時間複雜度將會達到 $O(kn)$ ，因此當點的維度或是點的數目過於龐大，使用線性搜尋法方法查詢相近點的時間，是相當耗時的。

以概似最近搜尋演算法的判別相近點，查詢空間上某點與其他點的相近點。當空間為較高的維度，或是空間上有大量資料點的時候，其速度是較為快速。概似最近搜尋法定義為在空間上所有點集合 S 中，其中 S 有 n 個點，從 S 取任一點 p 與給予查詢點 q ，計算 p 點與 q 點兩點之間距離，當距離為所給定的範圍之內，也就是視為 p 點為 q 點相近點， p 點即為所求。概似最近搜尋法為利用切割空間成子空間的方式，首先將空間中的每一個點先做分類於每個子空間，進而依據使用者輸入的查詢點，能快速找尋查詢點位於的子空間而找出相近的點，而不用找尋空間上的每一個點。目前常用的概似最近搜尋演算法有以 KD-Tree 為資料結構的演算法及 LSH (Locality Sensitive Hashing) 等演算法。

2.3.1 KD-tree

KD-tree 是一個針對 k 維度的多維度空間所設計的二元搜尋樹，樹中的階層是以資料點為基準分割維度空間，且每一個對應的子空間最少須包含一個資料點，並以此資料點來代表此一子空間。

KD-tree 的提出主要是為了解決關連性搜尋 (Associative Searching) 的問題。關連性搜尋問題的描述如下：假設檔案 F 共有 n 筆紀錄，每一筆紀錄都有 k 個屬性值 ($V_0, V_1, V_2, V_3, \dots, V_i, V_{k-1}$)，而使用者可以使用系統進行查詢，此種查詢工作就稱為關連性搜尋[3]。

KD-tree 是一種藉由遞迴的方式所建立的樹，根據資料的維度，分別比照第一維度，第二維度到第 N 維度，每一維度皆將資料分成兩邊。若分割到最後一軸時，樹的深度尚未滿足使用者所設定的值時，分割必須回到第一次分割資料集時所參考的軸，如此輪迴，直到滿足樹的深度為止。當資料點

為 n ，KD-tree 的概似最近搜尋的時間複雜度約略為 $O(\log_2 n)$ [12]，較線性搜尋時間快。KD-tree 也被廣泛應用在資料分群的問題中[15]。

2.3.2 LSH (Locality Sensitive Hashing)

LSH[4][14]主要的想法為空間中的每一個點經過雜湊函數的計算，可以讓每一個點與較相近的點碰撞的機率提高，反之若兩點的距離很遠則碰撞的機率則會下降。所以在查詢時，可以藉著雜湊函數來判斷鄰近的點，並且從雜湊桶裡得到所需相近點的資料。LSH 在動態修改方面，如插入，刪除和修改點等功能，相當容易使用，時間複雜度也不高。LSH 文獻中主要探討都使用歐氏空間來計算兩點間的距離，但是也可依據使用不同情況下，使用不同的距離計算公式，來定義不同 LSH 的雜湊函數，判斷點與點之間的相近程度。

LSH 是由 Indyk 和 Motwani 所提出，其主要目的在於解決在多維空間中搜尋某一點的臨近點之問題。LSH 在多維度的空間中，搜尋最近相鄰點時，其時間複雜度，會小於線性搜尋法的複雜度。

3. 以圖像內容為基礎的重製圖像偵測與追蹤系統 (Content-Based Image Detection and Tracking System, CBIDTS) 之架構

以圖像內容為基礎的重製圖像偵測與追蹤系統主要提供給圖像所有人，先將要受保護圖像利用重製圖像偵測與追蹤演算法取出特徵值，再轉存 LSH 所需形式，以減少之後特徵值比對時間，再利用網路機器人等圖像搜尋引擎，在網際網路上，週期性的收集圖像，針對所搜尋到每個圖像取出的特徵值且轉存 LSH 形式，並在資料庫紀錄每個圖像的特徵值以及所抓取的 URLs，再與之前存入之受保護圖像的特徵值，以 LSH 演算法方式判斷近似的圖像，假如為近似圖像的話，以 URLs 列出懷疑名單，而圖像所有人可再依照事先註冊的名單或其他認證技術，來判斷此圖像是否合法，如圖 2 所示。

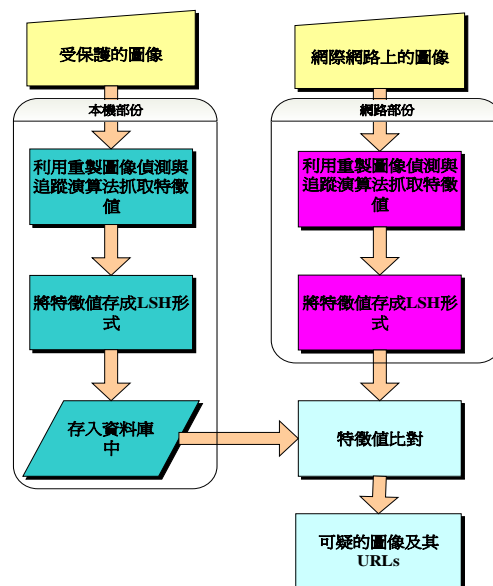


圖 2 CBIDTS 架構

3.1 重製圖像偵測與追蹤演算法

針對樣式辨識中圖像特徵值萃取演算法，目前有許多不同演算法用來判斷數位內容間的近似程度。為了應付網路上成千上萬的圖像，CBIDTS 必須能搜尋比對各種不同的圖像格式的數位內容，以及增強重製圖像的偵測與追蹤演算法抗圖像攻擊之能力。以達到有效性、多樣性、整合性與強韌性的反盜版系統功能，所以針對圖像特徵值萃取演算法主要採用以圖像內容為基礎的重製圖像偵測與追蹤中，以 Kim 取圖像特徵值[6]的概念，其演算法有一定的強韌性，且所得出特徵值的維度較小，與地區性描述子 (local descriptors) 相關演算法相比較，演算法的複雜度與特徵值的維度相對低很多。為了要應付網路上大量圖像，本系統採用以圖像內容為基礎的重製圖像偵測與追蹤演算法，來減少系統運算時所需花的時間。

以圖像內容為基礎的重製圖像偵測與追蹤演算法主要先切割圖像，並將切割的圖像利用離散餘弦轉換 (Discrete Cosine Transform, DCT) 轉換轉出離散餘弦轉換的係數，其中離散餘弦轉換轉換，在此系統以快速離散餘弦轉換 (fast DCT) 來加快離散餘弦轉換轉換的速度，其係數值的大小順序值，就是當做此圖像的特徵值。

CBIDTS 中的詳細重製圖像偵測與追蹤演算法如下圖 3 所示，

1. 將任一圖像作為輸入，把圖像轉成 YCbCr 模式，並把圖像等分成 8x8 的子圖像，而每一子圖像都取得其 Y 值之平均值，得一 8x8 的二維矩陣。
2. 8x8 的二維矩陣，由快速離散餘弦轉換轉換，得到 64 個係數，包含 1 個 DC 係數與 63 個 AC 係數。
3. 取得 63 個 AC 係數，並作遞增排序，知道每一個 AC 係數為全部 AC 係數的大小順序，將每一個 AC 係數的大小順序值作記錄的一個 63 個大小順序值，此順序值作為此圖像的特徵值。

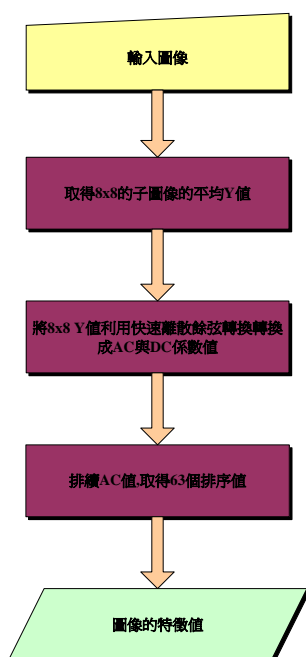


圖 3 重製圖像偵測與追蹤演算法

3.2 特徵值之比對

重製圖像偵測與追蹤演算法取出特徵值以後，都是以 63 個值代表，其每一個值代表某一個維度上的大小，所以也就是以 63 維空間向量來代表數位圖像的特徵值，但是當大量比較圖像時，這些資料量是非常多的，不僅會用掉電腦大部分的效能，也會浪費使用者大量的時間在比對相似圖像上，更甚會減少使用者想使用此系統的意願。如何解決問題即成為非常重要的一個課題，為了兼顧特徵值的強韌性，63 個維度的特徵值是必須的，所以無法從減少特徵值的維度方面著手改善，改變特徵值比對的搜尋方法來加快整體的速度成為最重要的議題。這裡將利用概似最近搜尋演算法中的 Locality Sensitive Hashing (LSH) [4][14] 來加快速度，並與線性搜尋法比較。

3.2.1 線性搜尋法

當取出 63 個維度的特徵值來代表每一個數位圖像，其中每一個特徵值也就是 63 維度的向量，所以線性搜尋法，就是將資料庫每一個圖像的特徵值與查詢圖像的特徵值之間，都需要兩兩去比較相似程度。其中兩個圖像間的比較，也就是在空間上兩個 63 維度的點，以歐幾里得長度來計算距離值，代表兩個圖像之間的相似程度。

3.2.2 概似最近搜尋法 — Locality Sensitive Hashing (LSH)

在 CBIDTS 中，LSH 主要是針對擷取圖像的特徵值之後，將特徵值利用雜湊函數建製雜湊表。也就是將保護的圖像與網路機器人所抓取的圖像轉換成特徵值，將特徵值與圖像來源資料存在資料庫中之後，再利用 LSH 演算法所產生的鍵值與圖像來源資料記錄在資料庫中，以便之後查詢相似圖像使用。

LSH 主要應用機率概念的方法，分類空間上的所有點，使得相近的點歸類在同一類的可能性較大，反之不相似的可能性較小。CBIDTS 中利用重製圖像偵測與追蹤演算法所擷取出的特徵點具有 63 個維度，所以可以利用 LSH 較快速搜尋概似最近搜尋法的方式，來改進所有圖像比較特徵值的時間。LSH 的流程主要如下，首先會對於每一張圖像的特徵值中各個維度乘上一個隨機變數，接著會根據每一個維度大小編成漢明碼，這也是之後會使用到的鍵值，接著會根據這些鍵值來建立雜湊表，另外 LSH 演算法中，並不會只建立單一雜湊表，雜湊表建立的個數會根據維度，產生多個雜湊表。其 LSH 演算法的運作，主要運作流程如下 [4][8][13][15]：

3.2.2.1 建立圖像資料庫

首先會將圖像的位置當作資料庫的鍵值和特

徵值當作資料被儲存起來，接著會將這些資料取出來建立雜湊表，將特徵值放入雜湊函數產生鍵值，而會根據鍵值來找到資料放在相對應的雜湊桶裡，而雜湊桶裡會儲存圖像的位置。查詢時可以根據圖像位置去找到圖像。接著會將建立好的雜湊表儲存起來。因為資料庫無法儲存結構性的資料。因此使用特別的方法建立鍵值來達到 LSH 的結構。建立鍵值的方法主要先將每個雜湊表編號，雜湊表的編號和雜湊函數所產生的鍵值做結合就會產生儲存雜湊表所需要的鍵值值（鍵值 = 雜湊表值 + 雜湊函數鍵值），因為會使用到重複的鍵值值所以將資料庫設定必須設定為可以使用相同鍵值儲存資料，如圖 4。

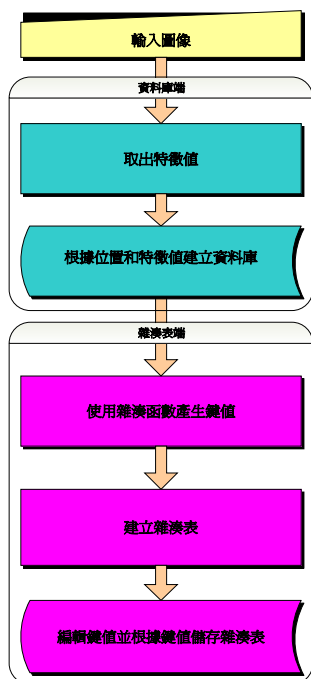


圖 4 建立圖像資料庫

3.2.2.2 查詢圖像

圖像首先被轉換成 63 個維度的特徵值，接著雜湊函數會根據特徵值產生所對應的鍵值，使用得到的鍵值到以儲存的雜湊表裡尋找放在相同雜湊桶的其他圖像位置，接著會根據這些位置到另外一個資料庫取出需要的特徵值，接著使用歐幾里得距離來比較圖像的相似度，如圖五。

3.2.2.3 新增圖像

在新增的過程和建立資料庫和雜湊表的步驟相似，但為了減少建立雜湊表所花費的時間所以在每次新增資料時並不重新建立雜湊表，而重新建立雜湊表的時機是根據雜湊表是否可以容納目前資料筆數。若無法容納則重新建立雜湊表並把圖像位置和特徵值儲存到資料庫裡，若可以則使用雜湊函數產生鍵值，再根據鍵值將圖像位置儲存入雜湊表裡。另外特徵值會根據圖像位置被儲存入資料庫裡，如圖 5。

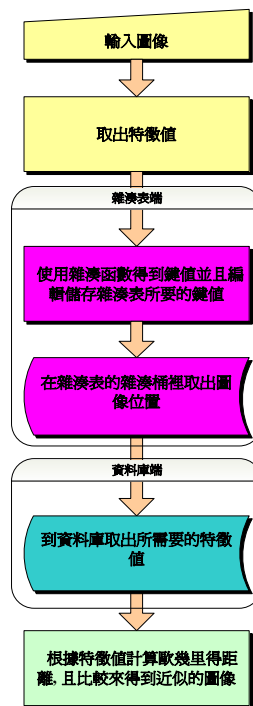


圖 5 查詢圖像

3.2.2.4 刪除圖像

在刪除圖像也是根據雜湊表容納程度來重建雜湊表。若雜湊表可以使用更小的容量，來容納現在的資料筆數，則重新建立雜湊表。若無法則會直接在現今的雜湊表裡進行刪除的動作，如圖 6。

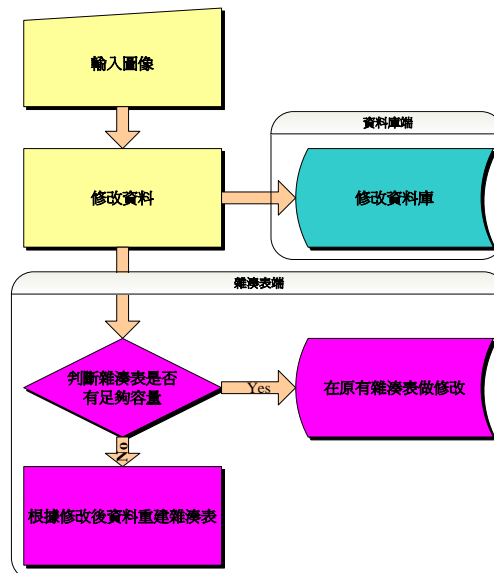


圖 6 新增與刪除圖像

4. 實驗結果

實驗硬體採用 Pentium 2.4 GHz, RAM 512 MB。圖像則採用 Corel Gallery 圖庫中隨機取 52148 張圖像當作網路上搜尋到的圖像，其中選取 100 圖像，當作被保護的圖像，每張保護的圖像利用 Adobe Photoshop 7.0 做 12 種圖像變化攻擊，包含對圖像作馬賽克、雜訊、放大兩倍、Jpeg 壓縮、水平反轉、

圖上畫線、海浪效果、殘影、縮小兩倍、旋轉 180 度、旋轉 2 度、水彩效果等圖像攻擊，這些攻擊圖像也就是代表可能由網路上搜尋到的重製圖像。實驗圖像中，重製圖像加上原圖共 13 張相似的圖像，如圖七。

實驗將所有圖像特徵值放入資料庫中，共 53348 張圖，其中 100 張為當作保護的圖像。搜尋資料庫，看是否每一張保護圖像是否能找尋到本身圖像以及重製的圖像，同時其他非重製的圖像，也盡量要視為較不相似，也就是差距較大。其中每個圖像的特徵值比較方式，分別前面章節所述利用線性搜尋法與概似最近搜尋法中的 LSH 方式搜尋不同的相似程度，也就是圖像間不同的容許差距，分別做實驗測試，得出查全率 (Recall) 與查準率 (Precision)，以及其搜尋花費的平均時間。其查全率與查準率的定義如下，

查全率 (Recall) = 實際正確重製圖像的數目 / 所有重製圖像的數目

查準率 (Precision) = 實際正確重製圖像的數目 / 偵測出重製圖像的數目



圖七 原圖與重製圖像 (1)原圖 (2)馬賽克 (3)放大兩倍 (4)縮小兩倍 (5)雜訊 10.0 (6)Jpeg 壓縮 50% (7)水平反轉 (8)海浪效果 (9)殘影 (10)旋轉 180 度 (11)旋轉 2 度 (12)水彩效果 (13)圖上畫線

在實驗 53348 張圖像中，使用 LSH 方式搜尋 100 原圖不同相似程度與不同攻擊方式，找尋到重製圖像數目的結果如表 1-1 與表 1-2，其中相似程度定義為圖像特徵點之間的距離正規化後的值，介於 0 到 1 之間，當值越小表越相近，反之值越大表越不相近。此外，不同相似程度的查全率與查準率，如表 2 與圖 8。由實驗得知，CBIDTS 中，以線性

搜尋法方式時，對圖像單純改變色系、水平反轉、旋轉 180 度等不改變圖像整體存在內容的情況下，對重製圖像都有良好的辨識度。且當相似程度初始值設為 0.3 時，表現最佳，其查全率為 93.15%，查準率為 98.54%。

表 1-1 以線性搜尋法搜尋不同相似程度與不同攻擊方式，找尋到重製圖像張數

相似程度	馬賽克	放大兩倍	縮小兩倍	雜訊 10.0	Jpeg 壓縮 50%	水平反轉
0.2	100	100	100	100	100	100
0.25	100	100	100	100	100	100
0.3	100	100	100	100	100	100
0.35	100	100	100	100	100	100
0.4	100	100	100	100	100	100
0.45	100	100	100	100	100	100

表 1-2 以線性搜尋法搜尋不同相似程度與不同攻擊方式，找尋到重製圖像張數

相似程度	海浪效果	殘影	旋轉 180 度	旋轉 2 度	水彩效果	圖上畫線
0.2	19	100	100	4	36	55
0.25	60	100	100	14	58	76
0.3	93	100	100	56	76	86
0.35	100	100	100	83	92	97
0.4	100	100	100	97	97	98
0.45	100	100	100	98	99	99

表 2 線性搜尋法時不同相似程度的查全率與查準率

相似程度	查全率	查準率
0.2	78.00%	100.00%
0.25	85.23%	100.00%
0.3	93.15%	98.54%
0.35	97.85%	44.55%
0.4	99.38%	4.16%
0.45	99.69%	0.52%

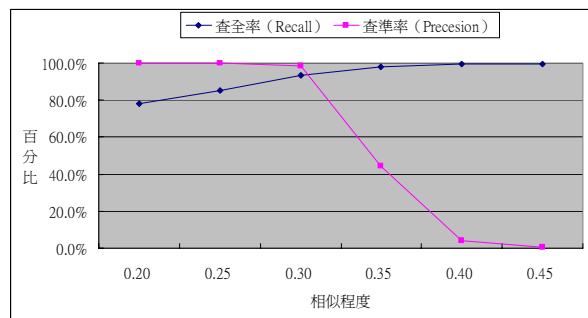


圖 8 線性搜尋法時不同相似程度的查全率與查準

率

在實驗 53348 張圖像中，使用 LSH 方式搜尋 100 原圖不同相似程度與不同攻擊方式，找尋到重製圖像數目的結果如表 3-1 與表 3-2，其查全率與查準率如表 4 與圖 9。由實驗得知，CBIDTS 中，以 LSH 方式時，其查詢到的重製圖像同線性搜尋法，對圖像單純改變色系、水平反轉、旋轉 180 度等不改變圖像整體存在內容的情況下，對重製圖像都有良好的辨識度。且當相似程度初始設為 0.3 時，表現最佳，其查全率為 93.15%，查準率為 98.70%。

表 3-1 以 LSH 方式搜尋不同相似程度與不同攻擊方式，找尋到重製圖像張數

相似程度	馬賽克	放大兩倍	縮小兩倍	雜訊 10.0	Jpeg 壓縮 50%	水平反轉
0.2	100	100	100	100	100	100
0.25	100	100	100	100	100	100
0.3	100	100	100	100	100	100
0.35	100	100	100	100	100	100
0.4	100	100	100	100	100	100
0.45	100	100	100	100	100	100

表 3-2 以 LSH 方式搜尋不同相似程度與不同攻擊方式，找尋到重製圖像張數

相似程度	海浪效果	殘影	旋轉 180 度	旋轉 2 度	水彩效果	圖上畫線
0.2	19	100	100	4	36	55
0.25	60	100	100	14	58	76
0.3	93	100	100	56	76	86
0.35	100	100	100	83	92	97
0.4	100	100	100	97	97	98
0.45	100	100	100	98	99	99

表 4 以 LSH 方式時不同相似程度的查全率與查準率

相似程度	查全率	查準率
0.2	78.00%	100.00%
0.25	85.23%	100.00%
0.3	93.15%	98.70%
0.35	97.85%	47.15%
0.4	99.38%	7.17%
0.45	99.69%	5.92%

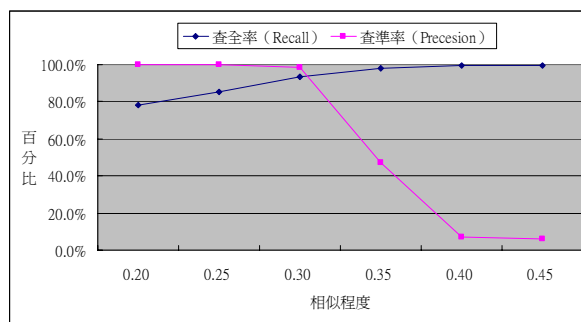


圖 9 LSH 搜尋法時的查全率與查準率

表 5 線性搜尋法與 LSH 之比較表

特徵值比較搜尋方式	線性搜尋法	LSH
平均需比較特徵值數	53348	219.78
平均搜尋時間	4.82 s	0.05s

綜合以上實驗結果，當圖像單純改變色系、水平反轉、旋轉 180 度等不改變圖像整體存在內容的情況下，對重製圖像都有良好的辨識度。且當相似程度初始設為 0.3 時，CBIDTS 表現最佳。由於 LSH 利用分割空間的方式，會將資料庫中每一個圖像先作分類，使得需要比較圖像的特徵值數較少，所以搜尋時間也減少，也造成 LSH 之查準率有所提升。利用線性搜尋法搜尋原圖的重製圖像平均所需比較特徵值數為 53358 個，所花時間為 4.82 秒，LSH 搜尋方式平均所需比較特徵值數為 219.78 個，所花時間為 0.05 秒，如表五。其中 LSH 搜尋方式比線性搜尋法快 96.4 倍，所以 CBIDTS 中，採用 LSH 搜尋近似特徵值方式，為較有效的方法。

5. 結論與未來展望

CBIDTS 為一個以圖像內容為基礎的重製圖像的偵測與追蹤系統，提供數位內容提供者、交易與權限控管中心、管理者或執法人員主動出擊，搜尋出非法使用者重製圖像的行為。其系統整合網路機器人、重製圖像的偵測與追蹤技術與概似最近搜尋法等相關技術，對任一圖像之數位內容，透過網路偵測及搜尋任一相似圖像，以判斷是否合法使用。

網路上的圖像有數以萬計，使得在蒐集網路圖像的特徵值時，若以集中式系統可能負擔過重，此時可利用分散式系統分散每個系統中每個圖像在此演算法中各項操作的負擔。此外，一但偵測有未經授權的非法使用者時，也要進而配合相對應的法律配套措施，與以告發制裁，才能達到遏止盜版的歪風。

誌謝

本研究計畫部份經費由行政院國家科學委員會補助，計畫編號：NSC95-2422-H-001-007，NSC95-2422-H-001-009，NSC95-2422-H-001-008，特此致謝。

6. 參考文獻

- [1] 台灣數位內容產業白皮書, 2004
- [2] 陳心渝, 李政宏, 邱一航, 林韋伶, “數位版權管理機制實作 - 以數位典藏管理系統為例”, 第四屆數位典藏技術研討會
- [3] 林文平, "文件倉儲系統的索引結構與其特性探討", 2003
- [4] A. Gionis, P. Indyk, and R. Motwani. "Similarity search in high dimensions via hashing", Proceedings of the 25th International Conference on Very Large DataBases (VLDB), 1999.
- [5] Chun-Shien Lu, Chao-Yong Hsu, Shih-Wei Sun, Pao-Chi Chang "Robust mesh based hashing for Copy Detection and Tracking and tracing of images", IEEE Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on, 2004
- [6] C. Kim, "Content-based Image Copy Detection and Tracking" Signal Processing: Image Communication, Vol. 18, pp. 169-184, 2003
- [7] Dong-Qing Zhang, Shih-Fu Chang "Detecting image near-duplicate by stochastic attributed relational graph matching with learning", Proceedings of the 12th annual ACM international conference on Multimedia, 2004
- [8] E2LSH package,
<http://web.mit.edu/andoni/www/LSH/index.html>
- [9] E. Y. Chang, J. Z. Wang. C. Li. and G. Wiederhold. "RIME: A Replicated Image Detector for the World-Wide Web ", IS&T/SPIE Symposium of Voice, Video, and Data Communications, Boston, November 1998.
- [10] Hsiao, Jen-Hao, "2004 Digital Watermark Competition: The Technical Report", Technical Report, IIS Academia Sinica, 2004 April.
- [11] Huang, Chun Hsiang; Wu, Ja-Ling, "Information Technologies for Digital Rights Managements : A Survey", Communication and Multimedia Laboratory, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, R. O. C. June, 2004
- [12] J. L. Bentley, "Multidimensional binary search trees used for associative searching", Commun. ACM 18, 9 (Sep. 1975), 509-517.
- [13] Locality Sensitive Hashing Applet,
<http://www.mit.edu/~vona/LSH-info.html>
- [14] M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni "Locality-sensitive hashing scheme based on p-stable distributions", DIMACS Workshop on Streaming Data Analysis and Mining, 2003.
- [15] Moore, A. W. "Very Fast EM-Based Mixture Model Clustering Using Multiresolution Kd-Trees", In M. S. Kearns, S. A. Solla, & D. A. Cohn (Eds.), Advances in Neural Information Processing Systems, 2, Cambridge: MIT Press (1999).
- [16] Yan Ke, Rahul Sukthankar, Larry Huston "Efficient Near-duplicate Detection and Sub-image Retrieval", Proceedings of the 12th annual ACM international conference on Multimedia, 2004