

影像資料庫系統之空間推論與相似度檢索植基於RS⁺-strings

Spatial Reasoning and Similarity Retrieval for Image Database Systems Based on RS⁺-strings

簡永仁
靜宜大學資訊管理研究所
yrjean@pu.edu.tw

白質元
靜宜大學資訊管理研究所
g9571017@pu.edu.tw

摘要

在影像資訊系統中，為了使影像檢索達到比對更精確與查詢更快速的目標，發展一套良好的空間知識表示法更顯得重要，透過空間知識表示法所儲存的空间資訊作為影像索引，可支援空間推論與相似檢索，以滿足使用者的需求。而RS-string與2D Z-string乃著名的空間知識表示法，RS-string主要植基於極座標系統，能夠達到旋轉不變之特性，而植基於卡笛生座標系統的2D Z-string則藉由零切割機制，不僅縮減字串長度且加入物件大小、位置、距離等計量資訊，兩者都克服了2D string相關研究的一些缺點。然而，我們發現在某些情況下，RS-string仍有精確度不足之處，以致會做出相似比對的誤判。因此，在本文中，我們以RS-string為基礎，排除在極座標系統中引用2D Z-string概念的困難，提出RS⁺-string空間知識表示法，同時擁有了RS-string與2D Z-string的優點，進而能夠達到更精確的空間推論與相似檢索。

關鍵詞：RS-string、RS⁺-string、2D Z-string、極座標系統、相似度檢索

Abstract

In image information system, it is important to develop a well spatial knowledge representation in order to attain the goal that image retrieval can be much precisely and quickly. By spatial knowledge representation to store the spatial information, it can support spatial reasoning and similarity retrieval to help users query image to satisfy their request. RS-string and 2D Z-string are the two famous spatial knowledge representation. RS-string is based on polar coordinate system, so it can arrive at rotation-invariant. But, 2D Z-string is based on Cartesian coordinate system, its feature not only reduces the length of string by using zero-cutting but also records the metric information that includes the size of object、location and distance. Both of them overcome the drawbacks of the relative researches of 2D string. However, we find that RS-string can not attain much precisely similarity retrieval in some condition, so it may result in misjudging of similarity retrieval. Thus, in this paper, we propose a spatial knowledge representation which is based on RS-string

and we name it as RS⁺-string. At the same time, we overcome the difficulty to apply the concept of 2D Z-string to the polar coordinate system. In addition, RS⁺-string also has the advantages of RS-string and 2D Z-string, therefore it can attain much precisely spatial reasoning and similarity retrieval.

Keywords：RS-string、RS⁺-string、2D Z-string、polar coordinate system、similarity retrieval

1. 前言

隨著電腦網路不斷的蓬勃發展，頻寬不斷地提升，使得多媒體的相關應用日益普及，在傳統的資料庫系統上是以文字方面的資料擷取為主，大部分在多媒體擷取上較無支援，另外，隨著許多新的應用上的興起，例如：辦公室自動化、電腦輔助設計、電腦輔助製造、地理資訊以及工業零件搜尋等應用皆需要大量的圖形資訊，使得在多媒體資料庫系統上的相關需求與日俱增。

在影像資料庫系統上，當系統要識別影像時，必須先經過影像處理取得影像中存在哪些物件，進而透過這些物件的空間關係來做影像上的識別，而空間推論(spatial reasoning)與相似度檢索(similarity retrieval)對影像資料庫而言是兩個相當重要的功能，因此，如何從影像裡萃取其特徵並且設計一種適當的圖像索引(iconic indexing)來幫助影像儲存在資料庫裡以及如何透過圖像索引來輔助做空間推論與相似度檢索也就成為影像資料庫上的相當重要的議題。

在過去，學者們對於圖像索引的發展提出了許多相關的研究成果，其中包含了以空間知識表示法(spatial knowledge representation)的相關研究，因此，藉由以空間知識表示法為基礎的圖像索引，則可以用來支援空間推論與相似度檢索，以幫助使用者做影像查詢。然而，在此方面的相關研究，其最早可以追溯到 Chang [14].所提出的 2D string。2D string 的精神為：以零大小(zero-size)的物件來表示影像上的物件，然後，再以直角系座標的投射方式將物件分別投射在 x 軸與 y 軸上，並且使用 {'<', '=', '>'}-這三個空間知識來表達影像物件在 x 軸與 y 軸上的空間關係，因此，也就產生了兩個維度的字串索引，其優點為保留了影像上物件之間的空间關係(spatial relationship)以支援做空間推論。

然而，2D string 僅以三個空間運算子 (spatial operator) 來支援空間知識的表達，當應用在複雜的影像時，則 2D string 無法給予較完整地空間描述。因此，Jungert and Chang[13] 藉由導入新的空間運算子來對 2D string 做擴充，即為 2D G-String，因此，2D G-string 的空間知識表示法比 2D string 更加完善，使得在影像的表達上能更有效率。然而，雖然 2D G-string 解決了 2D string 在表達更複雜影像上的不足，但是，由於 2D G-string 其本身使用的切割線 (cutting-line) 架構，會將影像上物件重疊的關係—無論是完整重疊 (complete overlap) 或是只有部分重疊 (partly-overlap)，都切割成許多個子物件 (subpart)，因此，在字串長度上，會增加儲存空間的負擔。

為了減少 2D G-string 做切割線的次數以及經切割後所產生出的子物件數量，Lee and Hsu[15] 提出了 2D C-string，其表達方式也是以切割線架構為基礎，但是，2D C-string 只針對當物件之間存在著部分重疊的關係時才對物件做切割，因此，在使用切割線架構上，2D C-string 會比 2D G-string 更有效率，並且也減少因切割而產生出大量子物件的情況。同時，Lee and Hsu 也定義出七個空間運算子如表(1)，來描述在單維上物件彼此的空間關係。但因為 2D C-string 在本質上還是使用了切割線的架構，因此，假如使用過多切割線的話，還是會造成儲存空間的負擔。

此外，因 2D C-string 沒有紀錄物件的大小 (size)、位置 (location) 與物件之間的距離 (distance) 資訊，使得 2D C-string 的表達上會產生不明確的狀況，因此，Huang and Jean[9] 提出了 2D C⁺-string，其字串索引記載著計量資訊 (metric information)，即：(1) 物件的大小 (2) 距離資訊。因此，2D C⁺-string 在字串索引的表達上會比 2D C-string 更為精確。

雖然 2D C⁺-string 克服了 2D C-string 不明確情況問題的發生，但是，2D C⁺-string 與 2D C-string 在本質上還是使用了切割線架構，因此，在最糟的情況下 (worst case)，還是會因為使用過多的切割線而產生大量的子物件，導致字串長度過長，造成儲存空間的負擔。因此，為了克服使用切割線所產生的缺點，Anthony and Chiu[1] 提出了 2D Z-string。2D Z-string 因架構在零切割 (zero-cutting) 的方式之上，使得物件不需要被切割成多個子物件，取而代之的是將物件一一做合併 (merge)。因此，2D Z-string 在字串長度上比 2D C⁺-string 較為精簡，故可降低儲存空間的需求，而克服了 2D C⁺-string 因使用過多的切割線所產生出的缺點。此外，2D Z-string 也記載著計量資訊，因此，也保有了 2D C⁺-string 的優點。

上述的 2D 系列的字串都是植基於卡笛生座標系統 (Cartesian coordinate system)，其共同的缺點為：當影像經過旋轉後，則所產生的字串索引可能會與旋轉前的字串索引不一致，而導致在相似度檢索上的誤判，有鑑於此，Petraglia[4] 提出了 2D R-string。2D R-string 乃植基於極座標系統 (Polar coordinate system)，由一旋轉中心物件 (rotation center object) 為起始，分別在同心圓方向 (ring-direction) 與扇形方向 (sector-direction) 產生出對應的空間關係字串，其字串索引的表達方式與 2D C-string 的切割線架構相似。而在 2D R-string 的結

構上，是以一對的 c-string 與 s-string 來表達物件在同心圓方向與扇形方向的空間關係，故 2D R-string 有著極座標系統之旋轉不變 (rotation-invariant) 的特性而解決了卡笛生座標系統的不足。

然而，2D R-string 的缺點在於其字串索引並沒有完整地表達出中心物件，因此，缺少了中心物件與其他物件之空間關係的訊息，使得會將原本不相似的影像誤判成相似，此外，當選定不同的中心物件時，將會產生不同的字串索引，因而容易將原本相似的影像誤判成不相似。

在 Hung and Jean[10] 所提出的 RS-string 改善了 2D R-string 的缺點。RS-string 在其字串索引中，記載了中心物件的資訊，故可以完整地表達出中心物件與其他物件的空間關係，同時，RS-string 也維持了 2D R-string 之旋轉不變的優點，使得 RS-string 在相似度檢索上能夠比 2D R-string 更加精確。但是，假如使用者希望能夠針對查詢影像做更進一步精確比對的話，此時，便可發現 RS-string 在精確比對上的不足。另外，RS-string 並無記載計量資訊，因此，會和 2D C-string 一樣，可能會產生不明確的情況。

在本篇論文中，我們提出了植基於 RS-string 而改良的空間知識表示法，稱為 RS⁺-string。RS⁺-string 應用了 2D Z-string 之零切割架構，因此，在字串的長度上，會比原本 RS-string 精簡，並擁有了 RS-string 在極座標系統上之旋轉不變的特性。同時，因 RS⁺-string 在其字串中，參考了 2D Z-string 的計量資訊，因此，在相似度比對時，能夠比 RS-string 更加精確且更能符合使用者所下達的影像查詢之需求。

2. 先前研究

2D Z-string

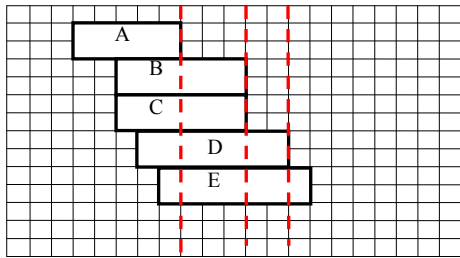
2D Z-string 在空間知識上的表達與 2D C-string、2D C⁺-string 一樣使用了如表(1)所示的七個空間運算子，透過這七個空間運算子來表達了物件之間的空間關係。2D Z-string 在字串中也有著 2D C⁺-string 記載計量資訊的優點，並且，因 2D Z-string 採用零切割的方法，改善了 2D C⁺-string 在某些情況下，可能因過多切割線造成大量子物件，導致字串長度過長，增加儲存空間負擔的缺點，例如下圖(1)所示，將物件們投射在 x 軸，比較 2D Z-string 與 2D C⁺-string 在 x 軸的字串長度，可以觀察出 2D Z-string 可降低字串長度的複雜度，然而，2D C⁺-string 因使用較多的切割線，而使得在字串長度上會比較耗儲存空間。因此，2D Z-string 的零切割架構更突顯出其優點。

雖然，2D Z-string 在字串的表達會比 2D C⁺-string 更有效率，然而，它還是植基於卡笛生座標系統，而卡笛生座標系統這一系列技術的共同特點為：必須先將影像物件以直角系座標的投射方式分別地投射在 x 軸與 y 軸上，然後，透過物件在這兩軸上的空間關係，分別地產生 x 軸與 y 軸上的字串索引，最後再以這兩軸的字串索引推論出任兩物件在影像上的二元空間關係。因此，一旦將影像經過旋轉後，

則所得出來的字串索引可能會與原先的不一致，而導致在相似度檢索上產生誤判。

表(1) 空間運算子的定義

Notations	Conditions
$A < B$	$end(A) < begin(B)$
$A = B$	$begin(A) = begin(B), end(A) = end(B)$
$A B$	$end(A) = begin(B)$
$A \% B$	$begin(A) < begin(B), end(A) > end(B)$
$A [B$	$begin(A) = begin(B), end(A) > end(B)$
$A] B$	$begin(A) < begin(B), end(A) = end(B)$
A / B	$begin(A) < begin(B) < end(A) < end(B)$



2D C⁺ u-string : $A_5|B_3|C_3|D_2|E_1|B_3=C_3=D_3=E_3|D_2=E_2|E_1$
 2D Z u-string : $((A_5/3(B_6=C_6))/_5D_7)/_6E_7$

圖(1)

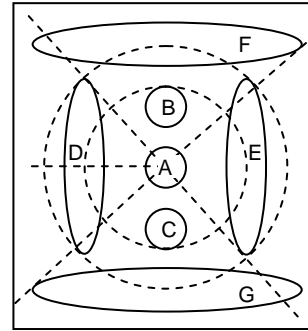
RS-string

由於 2D R-string 其字串索引缺乏了中心物件的資訊，因此，無法完整的表達出全部物件之間的二元空間關係(binary spatial relationships)，然而，RS-string 在字串索引上則加入了中心物件的資訊，使得在推論物件之間的二元空間關係能夠更加完善。因此，RS-string 解決了 2D R-string 在相似度檢索時，會將原本不相似的影像誤判成相似的缺失。另外，RS-string 在選擇中心物件時，會以較接近原點的物件為中心物件，因此，在中心物件的選擇上，會比 2D R-string 更加明確，所以，可以避免因選擇不同的中心物件而產生不一致的字串索引，故 RS-string 也克服了 2D R-string 另一項缺點—即會將原本相似的影像誤判成不相似。

RS-string 其結構是由一對的 r-string 與 s-string 所組成的，而 r-string 和 s-string 分別表達了影像物件在同心圓方向與扇形方向的空间關係，並且，RS-string 在字串索引的表達上會與 2D C-string 有相似的表達結構。RS-string 在字串索引的表達方式如下圖(2)所示。

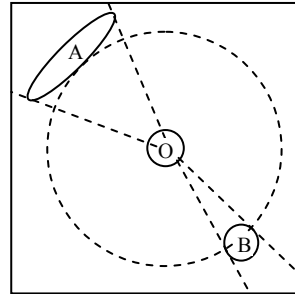
然而，RS-string 在某些的情況下，仍會造成相似檢索的誤判，例如下圖(3.a)、圖(3.b)所示，這兩張影像有相同的字串索引，因此，在 RS-string 的架構下，此兩張影像會被判定為相似，然而，如圖所示般的顯而易見，我們可以發現在圖(3.a)上的 A、B 兩物件在扇形方向上的角度大小與圖(3.b)上的 A、B 兩物件在扇形方向上的角度大小是不相等的，另外，在圖(3.a)上的 A 物件，在同心圓方向上離中心物件比較遠，然而，圖(3.b)上的 A 物件，在同心圓方向上卻離

中心物件比較近(B 物件亦然)，因此，基於更加精確比較的觀點，圖(3.a)和圖(3.b)是不相似的。



r-string : $A < B = C | D = E | D = E | F = G | F = G$
 s-string : $A = (D' | F | F) | (B < E) | E | G | G | (C < D'') | D''$

圖(2)



圖(3.a)

RS-string

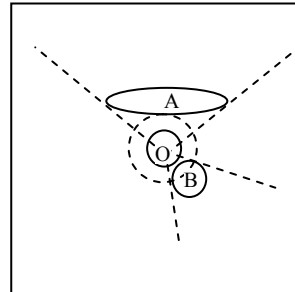
r-string : $O < B / A$

s-string : $O \% (A < B)$

RS⁺-string

r⁺-string : $O_1 <_8 A_1 / B_2$

s⁺-string : $O_{360} \%_{625} (A_{40} <_{160} B_{15})$



圖(3.b)

RS-string

r-string : $O < B / A$

s-string : $O \% (A < B)$

RS⁺-string

r⁺-string : $O_1 <_1 A_1 / B_2$

s⁺-string : $O_{360} \%_{630} (A_{100} <_{60} B_{35})$

3. RS⁺-string

雖然 RS-string 在字串索引的表達上，加入了中心物件的資訊，而使得更能完整地表達出全部物件之間的空间關係，進而克服了 2D R-string 之缺點，然而，正如同上述的圖(3.a)、圖(3.b)所示，以肉眼的角度來看，該兩張圖可以被認定為不相似，但經由 RS-string 的比對方式，則該兩張圖會被判定為相似，故 RS-string 在其相似度檢索上，仍存在著盲點，因此，在本研究中，我們主要參考 2D Z-string 的計量資訊以及零切割的概念，針對 RS-string 改良成為一個新的空間知識表示法—RS⁺-string。

在卡笛生座標系統與極座標系統對於計量資訊的表達上，雖然皆以數字型態的量化方式來表達，但實際上，兩者如何求出計量資訊的方式卻有所不同。卡笛生座標系統主要以直角系座標的投射方式將物

件投射到相對應的 x 軸與 y 軸，因此，就 x 軸與 y 軸而言，此兩軸會有一致性的計量資訊之產生方式。然而，在極座標系統上，同心圓方向與扇形方向在產生計量資訊的方式，並非如同在 x 軸與 y 軸上，具有一致的生成方式，因此，同心圓方向與扇形方向對於計量資訊的產生方式會有所不同。以下分別介紹同心圓方向與扇形方向在計量資訊的產生方式。

◆ 同心圓方向計量資訊的產生方式：

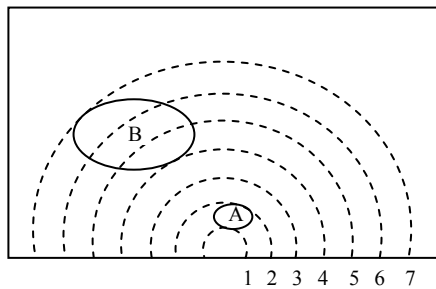
1. 首先是以原點為起始點，沿著同心圓方向往外擴散出許多不同的同心圓。
2. 每一個物件與同心圓有著兩個正相切的點即為該物件的開始位置 (begin-bound) 與結束位置 (end-bound)。
3. 透過每個物件在同心圓上的開始位置與結束位置來求出物件之間的計量資訊。

下圖(4.a)為同心圓方向上產生計量資訊的實例，可以透過同心圓之間單位距離來推論出 A 物件的大小為 1 (2-1)，B 物件的大小為 3 (7-4)，而 A、B 兩物件之間的距離為 2 (4-2)。

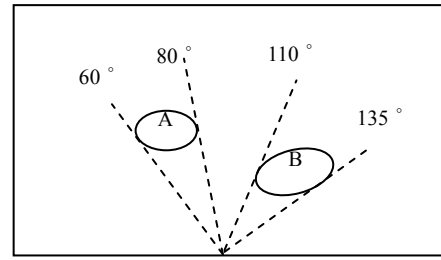
◆ 扇形方向計量資訊的產生方式：

1. 首先是以原點為起始點，繪製一半直徑(half line)並且將它放置於如圖(5.a)所示。半直徑從這個初始位置做順時針的旋轉。
2. 當半直徑旋轉時，則每個物件會有兩個放射狀的線段與其正相切而形成扇形面，而較接近初始位置的放射狀線段稱為物件在扇形方向上的開始位置，而另一個放射狀線段稱為結束位置。
3. 透過每個物件在扇型上的開始位置與結束位置來求出物件之間的計量資訊。

下圖(4.b)為扇形方向上產生計量資訊的實例，可以透過角度資訊來推論出 A 物件的大小為 20°，B 物件的大小為 25°，而 A、B 兩物件之間的距離為 30°。在圖中，可以明顯觀察出同心圓方向與扇形方向在產生計量資訊的不同點在於，同心圓方向是以單位距離計算出計量資訊，然而，扇形方向卻以角度資訊來計算出計量資訊。因此，同心圓方向與扇形方向在產生計量資訊的方式上，會有所差異。



圖(4.a)



圖(4.b)

我們的 RS⁺-string 主要是針對 RS-string 加入上述的計量資訊，以求能透過更精確的比對，解決如圖(3.a)及圖(3.b)相似比對誤判的問題。原本的 RS-string 是由一對 r-string 與 s-string 所組成的，其分別表達了影像物件在同心圓方向與扇形方向上的空間關係，而 RS⁺-string 在結構上與 RS-string 類似，其是由 r⁺-string 與 s⁺-string 所組成的。r⁺-string、s⁺-string 不但改良了 r-string、s-string 所欠缺的不足，並且也和 r-string、s-string 一樣，分別表達了物件在同心圓方向與扇形方向上的空間關係。RS⁺-string 除了改良了原本 RS-string 無法達到更加精確的相似度檢索之外，並且也維持了旋轉不變的特性。總之，RS⁺-string 參考了在 2D Z-string 裡所提到的計量資訊，使得在相似度檢索上能更加精確。此外，RS⁺-string 也應用了 2D Z-string 之零切割的架構，因此，在字串索引的長度上會比 RS-string 精簡。以下我們列出 2D Z-string[1]的字串生成演算法：

Algorithm : String generation

Input : $O_1(B_1E_1), O_2(B_2E_2), \dots, O_n(B_nE_n)$

Output : a 2D u-string(or v-string)

1. Sort in non-decreasing order all the begin-bound points and end-bound points $B_n, E_n, i = 1, 2, \dots, n$.
2. Group the same value points into a same-value-list. Form a same-value-list sequence.
3. Loop from step 4 to step 8 for each same-value-list according to the non-decreasing order.
4. If there is no end-bound in the list, process the next same-value-list.
5. Find the dominating object from the objects in the same end-bound list so that the corresponding begin-bound of the dominating object is the smallest of them. Compute the size of the dominating object.
6. If there exist no objects partly overlapping with the dominating object, find the objects covered by the dominating object (including the dominating object itself). Call the *template object generation algorithm* with the covered objects as the input parameter.
7. If there exist objects partly overlapping with the dominating object, choose among them the object with the smallest end-bound. Let the chosen object be P. Perform the following three phrases.
 - (a) Find the objects covered by object P (including object P itself). Then call the *template object generation algorithm* with the covered objects as the input parameter. Let the returned template object be T_1 .
 - (b) Find the objects covered by the dominating object but not by object P (including the

dominating object itself). Then call the *template object generation algorithm* with the covered objects as the input parameter. Let the returned template object be T_2 .

- (c) Merge T_1 and T_2 into a new template object by separators '(' and ')' with operator '/'. The distance associated with operator '/' is equal to the end-bound of object T_2 minus the begin-bound of the object T_1 . The begin-bound of the template object is equal to the begin-bound of object T_2 . The end-bound of the template object is equal to the end-bound of object T_1 .
8. Collect the begin-bounds and end-bounds of the objects into the same-value lists.
9. Call the *template object generation algorithm* with all the remaining objects as the input parameter. Output the representation of the final object. This is the u- (or v-)string.

由於 RS^+ -string 改善 Hung and Jean 所提出的 RS-string，因此，在字串索引生成的步驟與 RS-string 極為相似，以下分別描述 r^+ -string 和 s^+ -string 的字串生成演算法：

演算法： 在同心圓方向之字串生成演算法
輸入： 一張被分割的影像與極座標系統的原點
輸出： r^+ -string

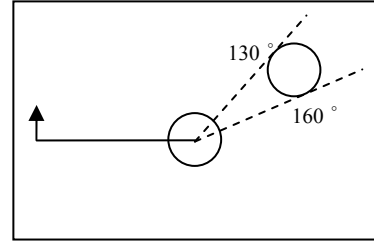
1. 以原點為起始點沿著同心圓方向往外擴散出許多不同的同心圓，每一個物件與同心圓有著兩個正相切的點即為該物件的開始位置(begin-bound)與結束位置(end-bound)，而較接近原點的點稱為開始位置，較遠的點為結束位置。
2. 假如有一個物件包含著原點，則定義該物件在同心圓方向的開始位置為 0。
3. 得到一系列 $S_1(B_1E_1), S_2(B_2E_2), \dots, S_n(B_nE_n)$ ，其中 S_i 是物件符號， B_i 和 E_i 是物件 S_i 在同心圓方向的開始位置與結束位置， $i = 1, 2, \dots, n$ 。
4. 應用 2D Z-string 的字串生成演算法得到一串 r^+ -string。

演算法： 在扇形方向之字串生成演算法
輸入： 一張被分割的影像與極座標系統的原點
輸出： s^+ -string

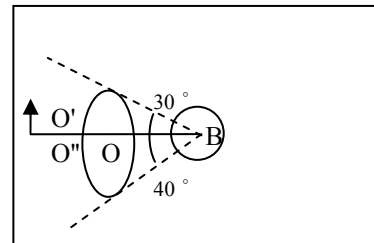
1. 以原點為起始點，繪製一半直徑(half line)並且將它放置於如圖(5.a)所示。半直徑從這個初始位置做順時針的旋轉而最後會停止在相同的位置。在扇形方向上任何位置的座標軸是介於 0° 到 360° 之間。當半直徑旋轉時，則每個物件會有兩個放射狀的線段與其正相切而形成扇形面，而較接近初始位置的放射狀線段稱為物件在扇形方向上的開始位置，而另一個放射狀線段稱為結束位置。
2. 假如有一個物件 O 在初始位置為 0 的地方被半直徑切割的話，則 O 物件會被視為有兩個邊對邊相接的 O' 物件與 O'' 物件，如圖(5.b)所示， O' 的開始位置會與 O'' 的結束位置相同。在圖(5.b)中， O' 物件、 O'' 物件與中心物件 B 的開始位置與大小分別

為 $O'(0,30)$ 、 $O''(320,40)$ 和 $B(0,360)$ 。

3. 假如有一個物件包含著原點，則分別定義該物件在扇形方向的開始位置與結束位置為 0° 與 360° 。
4. 得到一系列 $S_1(B_1E_1), S_2(B_2E_2), \dots, S_n(B_nE_n)$ ，其中 S_i 是物件符號， B_i 和 E_i 是物件 S_i 在扇形方向的開始位置與結束位置， $i = 1, 2, \dots, n$ 。
5. 應用 2D Z-string 的字串生成演算法得到一串 s^+ -string。

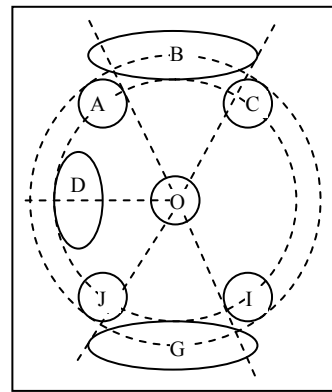


圖(5.a)



圖(5.b)

下圖(6)為一個 r^+ -string 與 s^+ -string 在表達方式上的實例，其分別表達了影像物件在同心圓方向與扇形方向的空間關係。在 RS^+ -string 的字串中，因記載了計量資訊(metric information)，所以可以觀察出其在字串索引的表達上比 RS-string 能更加精確。此外，在圖(3.a)與圖(3.b)中，假如以 RS^+ -string 的方式來表達其字串索引的話，則在相似度比對時，可判別出此兩張影像是不相似的，進而破除了原本的 RS-string 在相似度比對上的盲點。同時，我們也在 RS^+ -string 中應用了 2D Z-string 之零切割(zero-cutting)的架構，因此， RS^+ -string 與 2D Z-string 一樣在字串索引的表達上，會比較精簡。



r^+ -string : $(O_1 <_2 ((D_2 /_1 (A_2 = C_2 = I_2 = J_2)) /_1 (B_2 = G_2)))$
 s^+ -string :
 $(O_{360} = (D'_{25} <_{10} (((A_{25} /_5 B_{60}) /_5 C_{25}) <_{90} (((I_{25} /_5 G_{60}) /_5 J_{25}) <_{10} D''_{25}))))$

圖(6)

4. 影像重建演算法

在這一小節裡主要介紹影像重建演算法(picture reconstruction algorithm)，透過影像重建我們可以將 RS⁺-string 轉換成符號圖像(symbolic pictures)以幫助使用者在下達影像查詢的指令後，可以針對從資料庫檢索出來的影像做視覺化的瀏覽(visual browsing)。

以下說明在影像重建演算法裡所使用到的字串物件 W(string object)與符號物件(symbolic object)O 之符號標記。字串物件 W 代表著一個給定的 r⁺-string 或 s⁺-string 裡的符號，在此所提及的符號包含有在字串中代表影像物件的符號或是字串中代表空間運算子的符號，而每一個字串物件 W 則有兩個相對應的欄位，分別為 W.sym 和 W.size。W.sym 表達了在字串裡的符號(影像物件與空間運算子)，然而 W.size 表達了在字串中每一個影像物件所相對應的大小以及空間運算子的距離資訊—即 (1)當 W.sym 所表達的是影像物件時，例如圖(6)中的 s⁺-string 裡的 O 物件，則 W.sym='O'以及 W.size=360；(2)當 W.sym 所表達的是空間運算子時，例如圖(6)中的 r⁺-string 裡的 '<'運算子，則 W.sym='<'以及 W.size=2。而在這些空間運算子當中，除了 '<'、'%'和 '/'之外，其餘的運算子('='、'|'、 '['、']')的 W.size 欄位一律設置為 0。

相似地，影像物件 O 表達了符號圖像裡的物件，每一個影像物件有三個相對應的欄位—分別為 O.sym、O.size、與 O.location。O.sym 表達了符號物件(symbolic object)，O.size 表達了符號物件的大小，O.location 則是表達符號物件的空間位置。RS⁺-string 在字串索引的主體上，乃植基於 2D Z-string 之零切割架構，因此，在影像重建的結構上，會與 2D Z-string 的影像重建有相似的流程。在同心圓方向的影像重建執行步驟源自於 2D Z-string[1]。以下為 r⁺-string 與 s⁺-string 之影像重建演算法執行步驟：

演算法：同心圓方向影像重建演算法

輸入：含有 n 個元素的 r⁺-string： string = (W₁, W₂, ... W_n)

輸出：一串列的影像物件(O₁, O₂, ... O_m)

1. Loc = 0; Object_List = null; Stack = null; i = 1; j = 0; MoreOperators = False; /*Initialization*/
2. 假如在字串中還有任何剩餘的元素內容的話，則 goto 4; 否則輸出 Object_List 並且離開程式。
3. Loop from 步驟3.(a) ~ 步驟3.(b)

假如 MoreOperators = true, 則：

 - (a) i = i + 1; MoreOperator = false /* next operator */
 - (b) case W_i.sym
 - "%" : Loc = Loc + W_i.size; i = i + 1;
 - "<" : Loc = Loc + PreviousObjectSize + W_i.size; i = i + 1;
 - "/" : Loc = Loc + PreviousObjectSize - W_i.size; i = i + 1;
 - "|" : Loc = Loc + PreviousObjectSize; i = i + 1;

"]" :

假如 W_{i+1}.sym ≠ "(", 則：

TemplateSize = W_{i+1}.size;

else

TemplateSize = GetTemplateSize(i+1, string);

Loc = Loc + PreviousObjectSize - TemplateSize; i = i + 1;

"=" or "[" : i = i + 1;

"]" :

從堆疊移出一個 template object W;

Loc = W.beginBound;

PreviousObjectSize = W.size;

MoreOperators = True;

4. Loop from 步驟 4.(a)~步驟 4.(c)

(a) 假如 W_i.sym = '(', 則：

建立一個相對應的 template object W;

W.beginBound = Loc;

W.size = GetTemplateSize(i, string);

(b) 將此 template object W 放入到堆疊中;

(c) i = i + 1;

5. 假如 W_i 是個字串物件的話，則：

j = j + 1;

建立一個新的物件 O_j

O_j.sym = W_i.sym;

O_j.size = W_i.size;

O_j.beginBound = Loc;

將此物件 O_j 附加到 Object_List;

6. PreviousObjectSize = W_i.size;

7. MoreOperators = True; go to Step 3

演算法：扇形方向影像重建演算法

輸入：含有 n 個元素的 s⁺-string： string = (W₁, W₂, ... W_n)

輸出：一串列的圖示物件(O₁, O₂, ... O_m)

1. 與同心圓方向影像重建相同的(1)~(7)步驟。
2. 假如有存在著兩個物件 O_i 與 O_j，而此兩物件是被半直徑在初始位置為 0 的地方所切割的物件之子物件的話，並且 O_i 的位置為 0 的話，則：
 - (a) 合併 O_i、O_j 兩物件，則 O_i 開始位置會成為 O_j 的開始位置。
 - (b) 從 Object_List 移除 O_j。

5. 空間推論

Lee 和 Hsu[15]指出任兩個物件在單一維度上總共會有 13 種的空間關係。相似地，在極座標系統上任兩個物件在同心圓方向或在扇形方向上也會有 13 種的空間關係。藉由推論物件之間的空间關係，可以幫助使用者做相似度檢索的查詢。由於在 RS⁺-string 的字串中，記載了物件的計量資訊，因此，一旦將字串索引經由影像重建後，可以得到每一個物件重建後的位置、大小，之後再將兩兩物件透過 Huang and Jean[9]所提出的決策樹(圖(7)所示)以求出正確的空

間關係。由於極座標系統之旋轉不變的特性，使得在同心圓方向與扇形方向的空間推論(spatial reasoning)演算法有些許的不同，以下分別介紹同心圓方向與扇形方向的空間推論演算法：

演算法：同心圓方向空間推論演算法

輸入： O_A 與 O_B 兩個影像物件

輸出：A、B 兩物件在同心圓方向之二元空間關係

- $X_A = O_A.beginBound;$ $l_{Ax} = O_A.size;$
 $X_B = O_B.beginBound;$ $l_{Bx} = O_B.size;$
- 使用決策樹找出 A 物件與 B 物件之間之二元空間關係。

演算法：扇形方向空間推論演算法

輸入： O_A 與 O_B 兩個影像物件

輸出：A、B 兩物件在扇形方向之二元空間關係

- $B_A = O_A.beginBound;$ $S_A = O_A.size;$
 $B_B = O_B.beginBound;$ $S_B = O_B.size;$
- ```

If((0<=B_B<B_A+S_A-360)&&(B_B+S_B>B_A+S_A-360))
 then B_B = B_B + 360;
If((0<=B_A<B_B+S_B-360)&&(B_A+S_A>B_B+S_B-360))
 then B_A = B_A + 360;

If((B_B >= 0 && (B_B+S_B < B_A+S_A-360))
 then B_B = B_B + 360;
If((B_A >= 0 && (B_A+S_A < B_B+S_B-360))
 then B_A = B_A + 360;

If((B_B >= 0 && (B_B+S_B == B_A+S_A-360))
 then B_B = B_B + 360;
If((B_A >= 0 && (B_A+S_A == B_B+S_B-360))
 then B_A = B_A + 360;

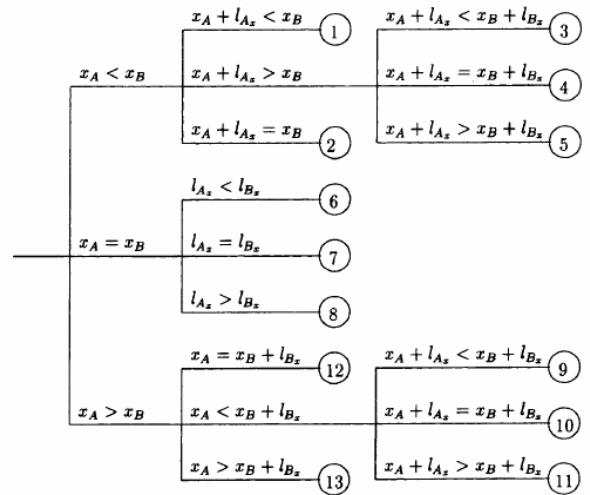
If((B_B == B_A+S_A-360))
 then B_B = B_B + 360;
If((B_A == B_B+S_B-360))
 then B_A = B_A + 360;

If((B_B == 0 && B_B+S_B == 360) && (B_B+S_B < B_A+S_A))
 then S_B = S_B * 2;
If((B_A == 0 && B_A+S_A == 360) && (B_A+S_A < B_B+S_B))
 then S_A = S_A * 2;

```
- $X_A = B_A;$      $l_{Ax} = S_A;$   
 $X_B = B_B;$      $l_{Bx} = S_B;$
- 使用決策樹找出 A 物件與 B 物件之間之二元空間關係。

在同心圓與扇形方向空間推論演算法中， $X_A$  代表物件 A 的開始位置(begin-bound)， $l_{Ax}$  則是物件 A 的大小(size)； $X_B$  代表物件 B 的開始位置， $l_{Bx}$  則是物件 B 的大小。在圖(5.b)中，字串索引的表達為( $B_{360}=(O'_{30} <_{290} O''_{40})$ )，值得注意的是，因為 O 物件被半直徑(half line)在初始位置為 0 的地方切割成兩個子物件，因此，O 物件被切割成  $O'$  與  $O''$  兩個子物件，而此兩子

物件的開始位址與大小分別為  $O'(0,30)$  與  $O''(320,40)$ ，因為  $O'$  與  $O''$  是相同物件的子物件，因此，在扇形方向之空間推理演算法中必須將它們合併為單一的物件，故合併後的開始位址與大小為： $O(320,70)$ ，而中心物件 B 的開始位置與大小為  $B(0,360)$ ，因為 O 物件是由  $O'$  和  $O''$  兩個子物件所組成，故在扇形方向的空間推理必須將 B 物件的計量資訊做暫時性的局部微調(tuning)，以求出 O 物件與 B 物件正確的空間關係。在圖(8)的幾種情況也需要將物件的計量資訊做暫時性的局部微調，以得到正確的空間關係。這幾種情況的微調與 RS-string 的微調情況相似，但因為 RS-string 在空間推論使用了開始列(begin-rank)與結束列(end-rank)的概念來代表影像物件在邏輯上的開始位置(begin-bound)與結束位置(end-bound)，然而， $RS^+$ -string 在其字串中，則因多了計量資訊的記載，故其意義上會比 RS-string 更加精確地表達出物件在空間上的開始位置與結束位置。



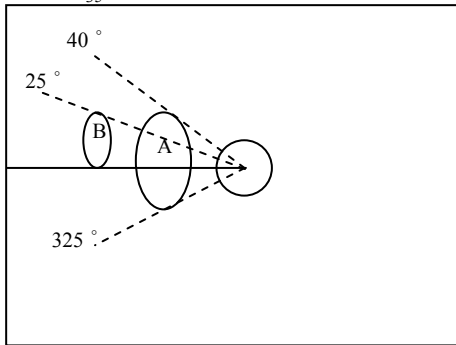
圖(7)

## 6. 相似度檢索

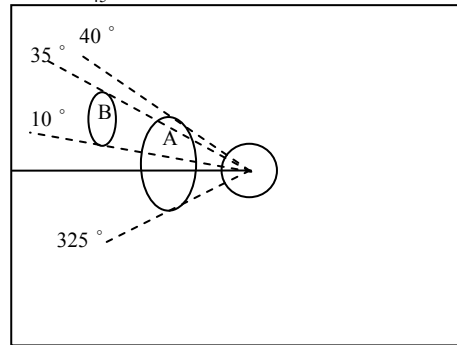
相似度檢索(similarity retrieval)是影像資料庫系統的重要功能之一，當使用者對影像資料庫系統下達相似度檢索的查詢時，則系統會在查詢影像與資料庫影像之間做相似度比對，以過濾出相似的影像。當任兩張影像被判斷相似時，則查詢影像中的物件集合必定為資料庫影像中物件集合的子集合，另外，空間關係在影像辨識上，亦是相當重要的特徵，所以，可透過這兩項特點來過濾許多不相似的影像。

除此之外，角度旋轉的因素也會影響相似度檢索的結果，例如 2D string 這一系列相關的字串索引研究，其主要是透過直角系座標的投射方式來判斷物件之間的空間關係，因此，假如將影像經過旋轉後，則產生出的字串索引往往會與旋轉前不一致，進而產生影像檢索的誤判。因此，對於一個良好的影像資訊系統而言，即使影像經過任何角度的旋轉後，則也不會影響其影像比對的結果。由於  $RS^+$ -string 是植基於極座標系統，因此，有著旋轉不變的優點。另外，因  $RS^+$ -string 在字串中，額外記載了計量資訊，所以，

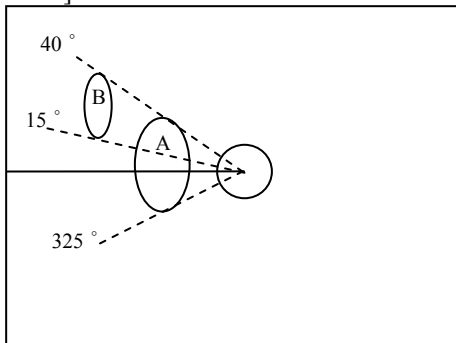
A(325, 75) and  
 B(0, 25)  $\Rightarrow$  B(360, 25)  
 $\rightarrow A \%_{35} B$



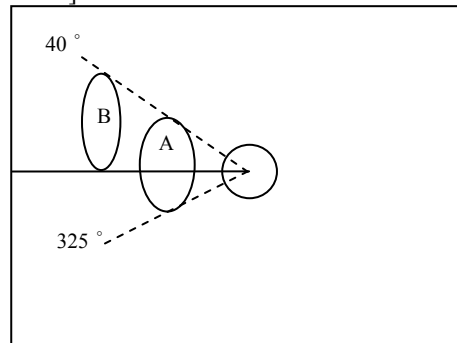
A(325, 75) and  
 B(10, 25)  $\Rightarrow$  B(370, 25)  
 $\rightarrow A \%_{45} B$



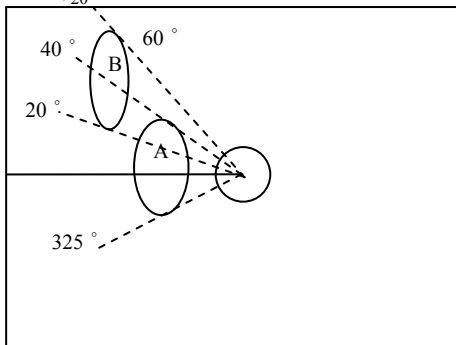
A(325, 75) and  
 B(15, 25)  $\Rightarrow$  B(375, 25)  
 $\rightarrow A | B$



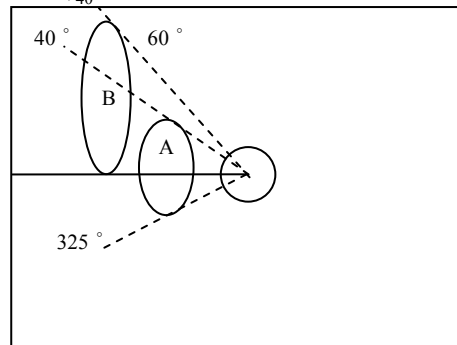
A(325, 75) and  
 B(0, 40)  $\Rightarrow$  B(360, 40)  
 $\rightarrow A | B$



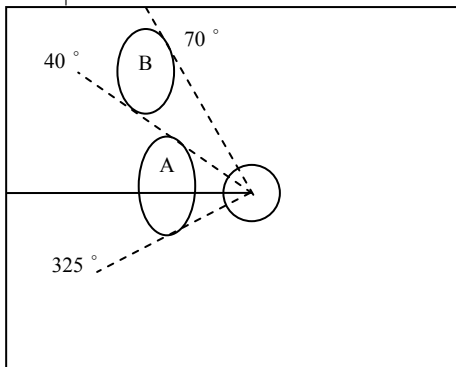
A(325, 75) and  
 B(20, 40)  $\Rightarrow$  B(380, 40)  
 $\rightarrow A /_{20} B$



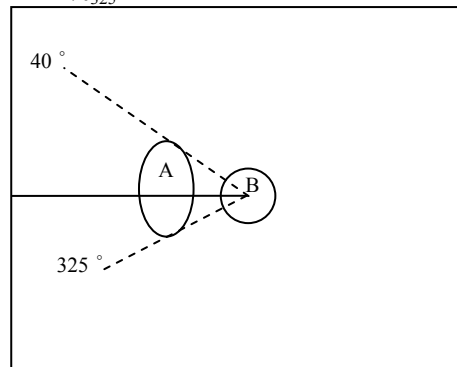
A(325, 75) and  
 B(0, 60)  $\Rightarrow$  B(360, 60)  
 $\rightarrow A /_{40} B$



A(325, 75) and  
 B(40, 30)  $\Rightarrow$  B(400, 30)  
 $\rightarrow A | B$



A(325, 75) and  
 B(0, 360)  $\Rightarrow$  B(0, 720)  
 $\rightarrow A \%_{325} B$



圖(8) 8



我們可允許使用者在下達相似度檢索的查詢時，可以動態調整所容許的誤差值，使得影像資訊系統在做影像查詢時，不至於因計量資訊的條件太嚴苛而難以找出相似的影像，此外，也因為在 RS<sup>+</sup>-string 加入了計量資訊進行相似度檢索，故使用者可以針對查詢影像與資料庫影像做進一步的精確比對。以下介紹了 RS<sup>+</sup>-string 在相似度檢索時所用到的一些符號標記以及在相似度之型別的定義式，而這些符號標記以及定義式是源自於 Huang and Jean[10] 所提出的 RS-string。稍後，我們也會介紹 RS<sup>+</sup>-string 在精確比對上所定義的符號標記與相似度之型別的定義式。

- $r_{AB}^R$  : A物件與B物件在同心圓方向上的空間關係。
- $r_{AB}^S$  : A物件與B物件在扇形方向上的空間關係。
- $C^f(r_{AB}^R, r_{AB}^S)$ : A物件與B物件在影像 f 上的類別空間關係(category spatial relation)之集合，而  $r_{AB}^R$  是同心圓方向上的空間關係， $r_{AB}^S$  是扇形方向上的空間關係， $C^f \in \{ \text{DISJOINT, JOIN, PARTOVL, CONTAIN, BELONG} \}$  並且這五個類別空間關係可以分別可以被縮寫成  $d、j、p、c、b$ 。
- $O_{f_q}$  : 在查詢影像  $f_q$  裡的全部物件的集合。
- $O_{f_d}$  : 在資料庫影像  $f_d$  裡的全部物件的集合。
- $R_{f_q}^{rc}$  : 在查詢影像  $f_q$  中，其以C物件為旋轉中心(rotation center)在同心圓方向的全部二元空間關係的集合(binary spatial relations)。
- $R_{f_q}^{sc}$  : 在查詢影像  $f_q$  中，其以C物件為旋轉中心在扇形方向的全部二元空間關係的集合。
- $R_{f_d}^{rc}$  : 在資料庫影像  $f_d$  中，其以C物件為旋轉中心在同心圓方向的全部二元空間關係的集合。
- $R_{f_d}^{sc}$  : 在資料庫影像  $f_d$  中，其以C物件為旋轉中心在扇形方向的全部二元空間關係的集合。

#### 定義 5.1

假如任兩個物件  $A、B \in O_{f_q}$ ，同時  $A、B \in O_{f_d}$  以致於  $C^f(r_{AB}^R, r_{AB}^S) = C^d(r_{AB}^R, r_{AB}^S)$ ，則資料庫影像  $f_d$  與查詢影像  $f_q$  是類別型相似(category-similar)。

#### 定義 5.2

假如資料庫影像  $f_d$  與查詢影像  $f_q$  符合類別型相似，並且  $R_{f_q}^{rc} \subseteq R_{f_d}^{rc}$  與  $R_{f_q}^{sc} \subseteq R_{f_d}^{sc}$ ，則資料庫影像  $f_d$  與查詢影像  $f_q$  是空間型相似(spatial-similar)。

由於 RS-string 在相似度檢索時，最多只能比對到任兩物件之空間型態的相似，所以，假如將影像中任兩物件之間的距離資訊列為相似度檢索的考量時，則可以發現 RS-string 無法滿足此方面的需求。因此，為了讓 RS<sup>+</sup>-string 能夠達到更進一步的精確比對，以下我們定義了 RS<sup>+</sup>-string 在精確比對上所使用到的符號標記以及在相似度之型別的定義式。

- $O_{f_q}^{rdc}$  : 在查詢影像  $f_q$  中，其以C物件為旋轉中心(rotation center)，在同心圓方向之全部物件的計量資訊(metric information)的集合。以下為其定義式： $\{ O_1 r O_2 (O_1\text{-size, } O_2\text{-size, distance}), O_1 r O_3 (O_1\text{ size, } O_3\text{ size, distance}), O_1 r O_n (O_1\text{-size, } O_n\text{-size, distance}), \dots, O_{n-1} r O_n (O_{n-1}\text{-size, } O_n\text{-size, distance}) \}$ ， $r$  為兩物件之間的二元關係(binary relation)並且  $n \geq 2$ 。
- $O_{f_q}^{sdc}$  : 在查詢影像  $f_q$  中，其以C物件為旋轉中心，在扇形方向之全部物件的計量資訊的集合。以下為其定義式： $\{ O_1 r O_2 (O_1\text{-size, } O_2\text{-size, distance}), O_1 r O_3 (O_1\text{ size, } O_3\text{ size, distance}), O_1 r O_n (O_1\text{-size, } O_n\text{-size, distance}), \dots, O_{n-1} r O_n (O_{n-1}\text{-size, } O_n\text{-size, distance}) \}$ ， $r$  為兩物件之間的二元關係，並且  $n \geq 2$ 。
- $O_{f_d}^{rdc}$  : 在資料庫影像  $f_d$  中，其以C物件為旋轉中心，在同心圓方向之全部物件的計量資訊的集合。以下為其定義式： $\{ O_1 r O_2 (O_1\text{-size, } O_2\text{-size, distance}), O_1 r O_3 (O_1\text{ size, } O_3\text{ size, distance}), O_1 r O_n (O_1\text{-size, } O_n\text{-size, distance}), \dots, O_{n-1} r O_n (O_{n-1}\text{-size, } O_n\text{-size, distance}) \}$ ， $r$  為兩物件之間的二元關係，並且  $n \geq 2$ 。
- $O_{f_d}^{sdc}$  : 在資料庫影像  $f_d$  中，其以C物件為旋轉中心，在扇形方向之全部物件的計量資訊的集合。以下為其定義式： $\{ O_1 r O_2 (O_1\text{-size, } O_2\text{-size, distance}), O_1 r O_3 (O_1\text{ size, } O_3\text{ size, distance}), O_1 r O_n (O_1\text{-size, } O_n\text{-size, distance}), \dots, O_{n-1} r O_n (O_{n-1}\text{-size, } O_n\text{-size, distance}) \}$ ， $r$  為兩物件之間的二元關係，並且  $n \geq 2$ 。

#### 定義 5.3

假如資料庫影像  $f_d$  與查詢影像  $f_q$  符合空間型相似，並且  $O_{f_q}^{rdc} \subseteq O_{f_d}^{rdc}$ ，但是不符合  $O_{f_q}^{sdc} \subseteq O_{f_d}^{sdc}$ ，則它們為同心圓型計量相似(Ring-metric-similar)。

#### 定義 5.4

假如資料庫影像  $f_d$  與查詢影像  $f_q$  符合空間型相似，並且  $O_{f_q}^{sdc} \subseteq O_{f_d}^{sdc}$ ，但是不符合  $O_{f_q}^{rdc} \subseteq O_{f_d}^{rdc}$ ，則它們為扇狀型計量相似(Sector-metric-similar)。

#### 定義 5.5

假如資料庫影像  $f_d$  與查詢影像  $f_q$  符合空間型相似，並且同時符合  $O_{f_q}^{rdc} \subseteq O_{f_d}^{rdc}$  與  $O_{f_q}^{sdc} \subseteq O_{f_d}^{sdc}$ ，則它們為 RS 型計量相似(RS-metric-similar)。

在本文中所提的 RS<sup>+</sup>-string 之相似度檢索演算法，主要是以原本 RS-string 的相似度檢索演算法為基礎，進而再做其他方面的擴充，因此，在相似比對時，可以比 RS-string 過濾出更相似的影像。以下介紹 RS<sup>+</sup>-string 在相似度檢索演算法的執行步驟。

**演算法：**相似度檢索演算法

**輸入：**一張查詢影像以及一連串用 RS<sup>+</sup>-string 所表達的資料庫影像

**輸出：**Category\_Similar\_List  
Spatial\_Similar\_List  
Ring-Metric\_Similar\_List  
Sector-Metric\_Similar\_List  
RS-Metric\_Similar\_List

1. 分別以查詢影像  $f_q$  裡的每一個物件為旋轉中心產生所相對應的 RS<sup>+</sup>-string。
2. 在每一個以 RS<sup>+</sup>-string 表達資料庫影像  $f_d$  之中，假如  $O_{f_q} \subseteq O_{f_d}$ ，並且  $f_d$  的旋轉中心也是在  $O_{f_q}$  當中，則
  - (a) 在同心圓方向與扇形方向使用空間推論演算法來推斷出在  $R_{fd}^{rc}$  與  $R_{fd}^{sc}$  中的全部二元空間關係，然後再從  $R_{fd}^{rc}$  與  $R_{fd}^{sc}$  推論出類別型空間關係  $C_{fd}$ 。
  - (b) 在步驟 1 對  $f_q$  所產生出的 RS<sup>+</sup>-string 之中，挑選出與目前  $f_d$  有相同旋轉中心的 RS<sup>+</sup>-string，接著透過同心圓方向與扇形方向的空間推論演算法來推斷出在  $R_{fq}^{rc}$  與  $R_{fq}^{sc}$  中的全部二元空間關係，最後再從  $R_{fq}^{rc}$  與  $R_{fq}^{sc}$  推論出類別型空間關係  $C_{fq}$ 。
  - (c) 假如  $f_d$  與  $f_q$  是類別型相似的話，則
    - (1) 將  $f_d$  加入到 Category\_Similar\_List。
    - (2) 假如  $f_d$  與  $f_q$  是空間型相似的話，則將  $f_d$  加入到 Spatial\_Similar\_List。
      - (2.1) 假如  $f_d$  與  $f_q$  滿足空間型相似，並且  $f_d$  與  $f_q$  符合同心圓型計量相似，但不符合扇狀型計量相似，則將  $f_d$  加入到 Ring-Metric\_Similar\_List。
      - (2.2) 假如  $f_d$  與  $f_q$  滿足空間型相似，並且  $f_d$  與  $f_q$  符合扇狀型計量相似，但不符合同心圓型計量相似，則將  $f_d$  加入到 Sector-Metric\_Similar\_List。
      - (2.3) 假如  $f_d$  與  $f_q$  滿足空間型相似，並且  $f_d$  與  $f_q$  同時符合同心圓型計量相似與扇狀計量相似，則將  $f_d$  加入到 RS-Metric\_Similar\_List。
3. 回傳 Category\_Similar\_List、Spatial\_Similar\_List、Ring-Metric\_Similar\_List、Sector-Metric\_Similar\_List 和 RS-Metric\_Similar\_List。

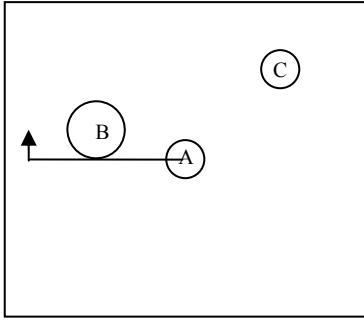
由於 RS<sup>+</sup>-string 在字串索引上，加入了計量資訊，所以，更能精確地表達任兩物件在影像上的空間關係，例如在圖(10.p)中，中心物件 A 與非中心物件 B 是「|」的空間關係，但是，假如以 RS-string 的方式來表

達的話，則中心物件與非中心物件都固定為「%」的空間關係。因此，RS<sup>+</sup>-string 比 RS-string 更能精確地表達任兩物件在影像上的空間關係，同時，RS<sup>+</sup>-string 也有著 RS-string 在極坐標系統上之旋轉不變的特性。

因此，基於上述的觀點，當使用者下達影像查詢時，RS<sup>+</sup>-string 在扇形方向上，必須忽略中心物件與非中心物件的比對。例如，在圖(10.p)與圖(10.q)是兩張相似的影像，然而，圖(10.q)是圖(10.p)的順時針旋轉 45° 的版本，我們可以發現  $R_{fp}^{sA} \not\subset R_{fq}^{sA}$  ( $\cdot: A[B \neq A\% B]$ )，因此，假如將扇形方向上的中心物件與非中心物件列為相似度比對項目的話，則圖(10.p)與圖(10.q)比對後的結果會被判定為不相似，而這樣的結果嚴重地違反極坐標系統之旋轉不變的特性，所以，RS<sup>+</sup>-string 在扇形方向的相似度檢索上，必須忽略中心物件與非中心物件的比對。同理，在扇形方向上，中心物件與非中心物件之計量資訊的比對亦是如此，例如，在圖(10.p)與圖(10.q)中，可以觀察出  $O_{fp}^{sdA} \not\subset O_{fq}^{sdA}$ ，所以，RS<sup>+</sup>-string 在扇形方向之精確的比對上，必須忽略中心物件與非中心物件之計量資訊的比對。因此，為了不違反旋轉不變之特性的情況下，在 RS<sup>+</sup>-string 的扇形方向上，我們只比對非中心物件彼此之間的空間關係以及其計量資訊。

在圖(11)中，我們列舉了 RS<sup>+</sup>-string 在相似度檢索演算法的例子，其中， $f_q$  是一張使用者所下達的查詢影像，而  $f_1$ 、 $f_2$ 、 $f_3$ 、 $f_4$ 、 $f_5$ 、 $f_6$ 、 $f_7$  則是資料庫裡的影像。以下分別敘述相似度比對後的結果：

- $f_1$  與  $f_q$  不相似：因為  $O_{fq} \not\subset O_{f1}$ ，所以  $f_1$  與  $f_q$  為不相似。
- $f_2$  與  $f_q$  不相似：因為  $C_{fq} \not\subset C_{f2}$ ，所以  $f_2$  與  $f_q$  為不相似。
- $f_3$  與  $f_q$  為類別型相似：因為  $C_{fq} \subset C_{f3}$ ，並且  $R_{fq}^{sA} \subset R_{f3}^{sA}$ ，但是  $R_{fq}^{rA} \not\subset R_{f3}^{rA}$ ，所以  $f_3$  與  $f_q$  為類別型相似。
- $f_4$  與  $f_q$  為空間型相似：因為  $C_{fq} \subset C_{f4}$ ，並且  $R_{fq}^{rA} \subset R_{f4}^{rA}$ ，以及  $R_{fq}^{sA} \subset R_{f4}^{sA}$ ，所以  $f_4$  與  $f_q$  為空間型相似。
- $f_5$  與  $f_q$  為同心圓型計量相似：因為  $f_5$  與  $f_q$  符合空間型相似的條件，並且  $O_{fq}^{rdA} \subseteq O_{f5}^{rdc}$ ，但是  $O_{fq}^{sdc} \not\subset O_{f5}^{sdc}$ ，所以  $f_5$  與  $f_q$  為同心圓型計量相似。
- $f_6$  與  $f_q$  為扇狀型計量相似：因為  $f_6$  與  $f_q$  符合空間型相似的條件，並且  $O_{fq}^{sdc} \subseteq O_{f6}^{sdc}$ ，但是  $O_{fq}^{rdA} \not\subset O_{f6}^{rdA}$ ，所以  $f_6$  與  $f_q$  為扇狀型計量相似。

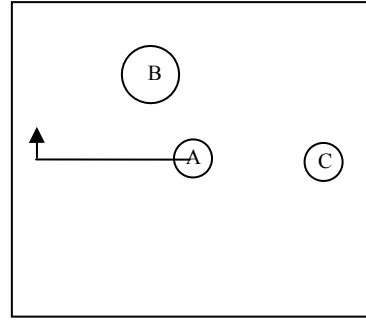


S<sup>+</sup>-string : ( A<sub>360</sub> [ ( B<sub>35</sub> <<sub>90</sub> C<sub>20</sub> ) )

R<sub>fp</sub><sup>SA</sup> : { A[B, A%<sub>0</sub>C, B<<sub>0</sub>C] }

O<sub>fp</sub><sup>sdA</sup> : A<sub>360</sub>[B<sub>35</sub> (360,0,35)  
 A<sub>360</sub><<sub>125</sub>C<sub>20</sub> (360,125,25)  
 B<sub>35</sub><<sub>90</sub>C<sub>20</sub> (35,90,20)

圖(10.p)

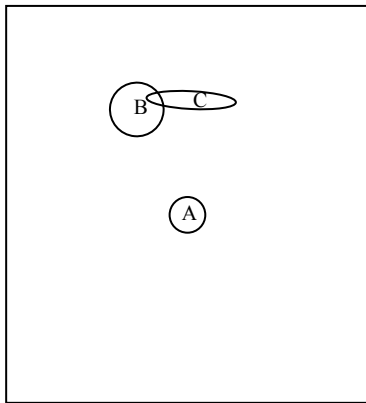


S<sup>+</sup>-string : ( A<sub>360</sub> %<sub>45</sub> ( B<sub>35</sub> <<sub>90</sub> C<sub>20</sub> ) )

R<sub>fq</sub><sup>SA</sup> : { A%<sub>0</sub>B, A%<sub>0</sub>C, B<<sub>0</sub>C }

O<sub>fq</sub><sup>sdA</sup> : A<sub>360</sub><<sub>45</sub>B<sub>35</sub> (360,45,35)  
 A<sub>360</sub><<sub>170</sub>C<sub>25</sub> (360,170,25)  
 B<sub>35</sub><<sub>90</sub>C<sub>20</sub> (35,90,20)

圖(10.q)



f<sub>q</sub>

Center object : A

r<sup>+</sup> : ( A<sub>1</sub> <<sub>4</sub> ( B<sub>3</sub> %<sub>1</sub> C<sub>1</sub> ) )

s<sup>+</sup> : ( A<sub>360</sub> %<sub>50</sub> ( B<sub>25</sub> /<sub>5</sub> C<sub>45</sub> ) )

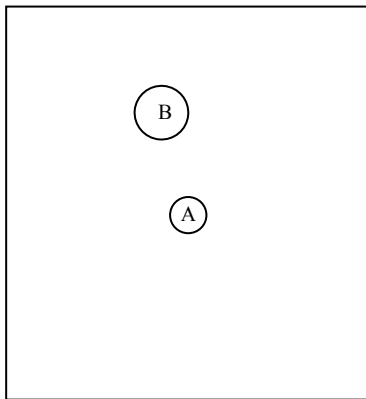
C<sub>fq</sub> = { AdB , AdC , BpC }

O<sub>fq</sub><sup>rdA</sup> : { A<<sub>4</sub>B ( 1 , 4 , 3 ) , A<<sub>5</sub>C ( 1 , 5 , 1 ) , B%<sub>1</sub>C ( 3 , 1 , 1 ) }

O<sub>fq</sub><sup>sdA</sup> : { A%<sub>50</sub>B ( 360,50,25 ) , A%<sub>70</sub>C ( 360,70,45 ) , B/<sub>5</sub>C ( 25,5,45 ) }

R<sub>fq</sub><sup>rA</sup> : { A<B , A<C , B%<sub>0</sub>C }

R<sub>fq</sub><sup>sA</sup> : { A%<sub>0</sub>B , A%<sub>0</sub>C , B/C }



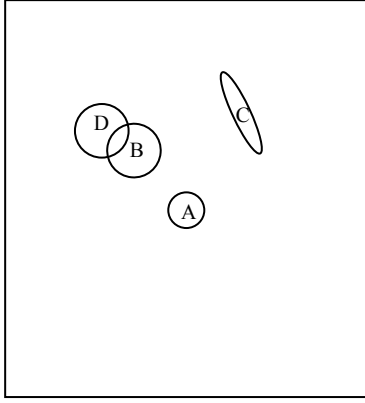
f<sub>1</sub>

O<sub>fq</sub> ⊄ O<sub>f1</sub>

f<sub>1</sub> is not category-similar to f<sub>q</sub>.

f<sub>1</sub> is not spatial-similar to f<sub>q</sub>.

圖(11)



$f_2$

$$r^+ : (A_1 <_2 (B_3 /_1 (C_3 = D_3))) \quad R_{f_2}^{rA} : \{ A < B, A < C, A < D, B/C, B/D, C=D \}$$

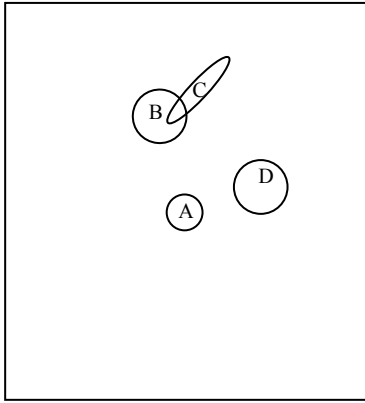
$$s^+ : (A_{360} \%_{35} ((B_{35} [D_{25}] <_{30} C_{35})) \quad R_{f_2}^{sA} : \{ A \% B, A \% C, A \% D, B < C, B [D, D < C \}$$

$$O_{f_2}^{rdA} : \{ A <_2 B (1,2,3), A <_4 C (1,4,3), A <_4 D (1,4,3), B /_1 C (3,1,3), \\ B /_1 D (3,1,3), C = D (3,0,3) \}$$

$$O_{f_2}^{sdA} : \{ A \%_{30} B (360,35,35), A \%_{100} C (360,100,35), A \%_{30} D (360,35,25), \\ B <_{30} C (35,30,35), B [D (35,0,25), D <_{40} C (25,40,35) \}$$

$$C_{f_2} = \{ AdB, AdC, AdD, BdC, BpD, CdD \}$$

$$C_{f_q} \not\subset C_{f_2} \rightarrow f_2 \text{ is not category-similar to } f_q.$$



$f_3$

$$r^+ : (A_1 <_2 ((D_3 /_2 B_3) /_2 C_4)) \quad R_{f_3}^{rA} : \{ A < B, A < C, A < D, B/C, D/B, D/C \}$$

$$s^+ : (A_{360} \%_{60} ((B_{30} /_{10} C_{30}) /_{40} D_{30})) \quad R_{f_3}^{sA} : \{ A \% B, A \% C, A \% D, B/C, B < D, C/D \}$$

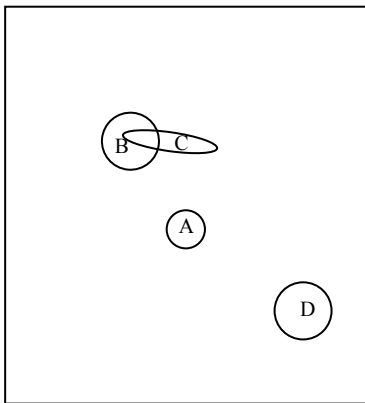
$$O_{f_3}^{rdA} : \{ A <_3 B (1,3,3), A <_4 C (1,4,4), A <_2 D (1,2,3), B /_2 C (3,2,4), \\ D /_2 B (3,2,3), D <_1 C (3,1,4) \}$$

$$O_{f_3}^{sdA} : \{ A \%_{60} B (360, 60, 30), A \%_{80} C (360, 80, 30), A \%_{150} D (360, 150, 30), \\ B /_{10} C (30, 10, 30), B <_{55} D (30, 55, 30), C <_{40} D (30, 40, 30) \}$$

$$C_{f_3} = \{ AdB, AdC, AdD, BpC, BdD, CdD \}$$

$$C_{f_q} \subset C_{f_3} \rightarrow f_3 \text{ is category-similar to } f_q.$$

$$R_{f_q}^{sA} \subset R_{f_3}^{sA} \text{ but } R_{f_q}^{rA} \not\subset R_{f_3}^{rA} \rightarrow f_3 \text{ is not spatial-similar to } f_q.$$



$f_4$

$$r^+ : (A_1 <_3 ((B_3 [C_2] /_1 D_3)) \quad R_{f_4}^{rA} : \{ A < B, A < C, A < D, B \% C, B/D, C [D \}$$

$$s^+ : (A_{360} \%_{40} ((B_{30} /_{20} C_{55}) /_{90} D_{25})) \quad R_{f_4}^{sA} : \{ A \% B, A \% C, A \% D, B/C, B < D, C < D \}$$

$$O_{f_4}^{rdA} : \{ A <_3 B (1,3,3), A <_3 C (1,3,2), A <_5 D (1,5,3), B \%_1 C (3,1,2), \\ B /_1 D (3,1,3), C [D (2,0,3) \}$$

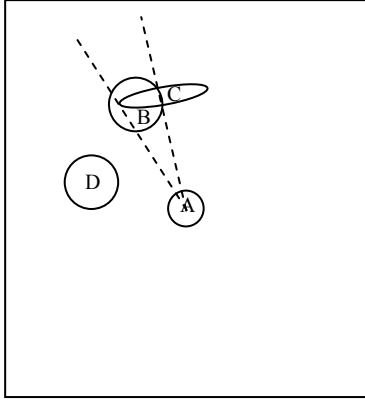
$$O_{f_4}^{sdA} : \{ A \%_{40} B (360, 40, 30), A \%_{55} C (360, 55, 55), A \%_{200} D (360, 200, 25), \\ B /_{20} C (30, 20, 55), B <_{125} D (30, 125, 25), C <_{90} D (55, 90, 25) \}$$

$$C_{f_4} = \{ AdB, AdC, AdD, BpC, BdD, CdD \}$$

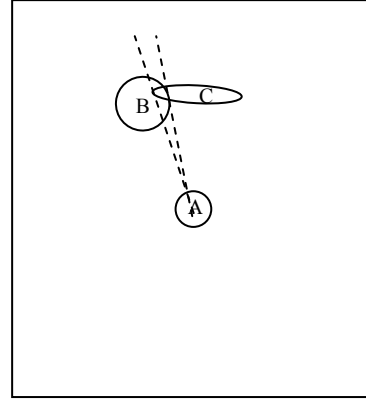
$$C_{f_q} \subset C_{f_4} \rightarrow f_4 \text{ is category-similar to } f_q.$$

$$R_{f_q}^{rA} \subset R_{f_4}^{rA} \text{ and } R_{f_q}^{sA} \subset R_{f_4}^{sA} \rightarrow f_4 \text{ is spatial-similar to } f_q.$$

圖(11)-continued.



$f_5$



$f_q$

$$r^+ : (A_1 <_3 (D_3 /_2 (B_3 \%_1 C_1)))$$

$$R_{f_5}^{rA} : \{ A < B, A < C, A < D, B \% C, D / B, D ] C \}$$

$$s^+ : (A_{360} [ (D_{30} <_{20} (B_{25} /_{20} C_{45})))$$

$$R_{f_5}^{sA} : \{ A \% B, A \% C, A [ D, B / C, D < B, D < C \}$$

$$O_{f_5}^{rdA} : \{ A <_4 B (1,4,3), A <_5 C (1,5,1), A <_3 D (1,3,3), B \%_1 C (3,1,1), D /_2 B (3,2,3), D ] C (3,0,1) \}$$

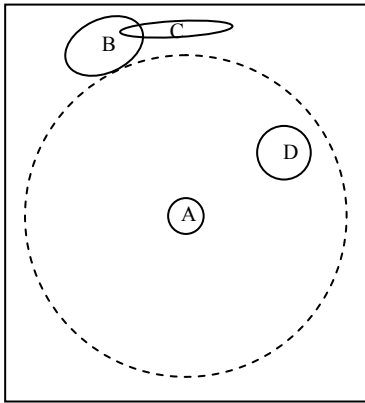
$$O_{f_5}^{sdA} : \{ A \%_{50} B (360, 50, 25), A \%_{55} C (360, 55, 45), A [ D (360, 0, 30), B /_{20} C (25, 20, 45), D <_{20} B (30, 20, 25), D <_{25} C (30, 25, 45) \}$$

$$C_{f_5} = \{ AdB, AdC, AdD, BpC, BdD, CdD \}$$

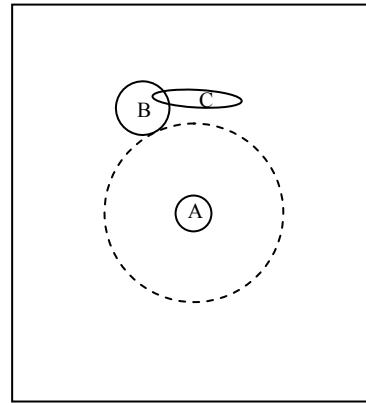
$$C_{f_q} \subset C_{f_5} \rightarrow f_5 \text{ is category-similar to } f_q.$$

$$R_{f_q}^{rA} \subset R_{f_5}^{rA} \text{ and } R_{f_q}^{sA} \subset R_{f_5}^{sA} \rightarrow f_5 \text{ is spatial-similar to } f_q.$$

$$O_{f_q}^{rdA} \subset O_{f_5}^{rdA} \text{ but } O_{f_q}^{sdA} \not\subset O_{f_5}^{sdA} \rightarrow f_5 \text{ is Ring-metric-similar to } f_q.$$



$f_6$



$f_q$

$$r^+ : (A_1 <_4 (D_3 <_1 (B_3 \%_1 C_1)))$$

$$R_{f_6}^{rA} : \{ A < B, A < C, A < D, B \% C, D < B, D < C \}$$

$$s^+ : (A_{360} \%_{50} ((B_{25} /_5 C_{45}) <_{30} D_{20}))$$

$$R_{f_6}^{sA} : \{ A \% B, A \% C, A \% D, B / C, B < D, C < D \}$$

$$O_{f_6}^{rdA} : \{ A <_8 B (1,8,3), A <_9 C (1,9,1), A <_4 D (1,4,3), B \%_1 C (3,1,1), D <_1 B (3,1,3), D <_2 C (3,2,1) \}$$

$$O_{f_6}^{sdA} : \{ A \%_{50} B (360, 50, 25), A \%_{70} C (360, 70, 45), A \%_{70} D (360, 145, 20), B /_5 C (25, 5, 45), B <_{70} D (25, 70, 20), C <_{30} D (45, 30, 20) \}$$

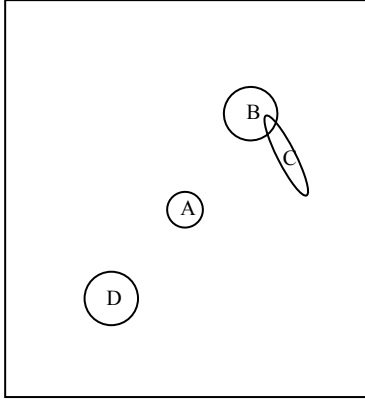
$$C_{f_5} = \{ AdB, AdC, AdD, BpC, BdD, CdD \}$$

$$C_{f_q} \subset C_{f_6} \rightarrow f_6 \text{ is category-similar to } f_q.$$

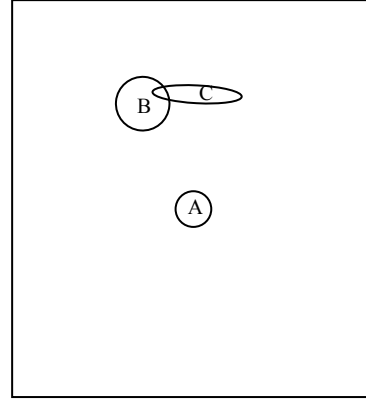
$$R_{f_q}^{rA} \subset R_{f_6}^{rA} \text{ and } R_{f_q}^{sA} \subset R_{f_6}^{sA} \rightarrow f_6 \text{ is spatial-similar to } f_q.$$

$$O_{f_q}^{sdA} \subset O_{f_6}^{sdA} \text{ but } O_{f_q}^{rdA} \not\subset O_{f_6}^{rdA} \rightarrow f_6 \text{ is Sector-metric-similar to } f_q.$$

圖(11)-continued.



$f_7$



$f_q$

$$r^+ : (A_1 <_4 (D_3 /_1 (B_4 \%_1 C_1))) \quad R_{f_7}^{rA} : \{ A < B, A < C, A < D, B \% C, B = D, D \% C \}$$

$$s^+ : (A_{360 \%_{110}} ((B_{25} /_5 C_{45}) <_{115} D_{25})) \quad R_{f_7}^{sA} : \{ A \% B, A \% C, A \% D, B / C, B < D, C < D \}$$

$$O_{f_7}^{rdA} : \{ A <_4 B (1,4,3), A <_5 C (1,5,1), A <_4 D (1,4,3), B \%_1 C (3,1,1), B = D (3,0,3), D \%_1 C (3,1,1) \}$$

$$O_{f_7}^{sdA} : \{ A \%_{110} B (360,110,25), A \%_{130} C (360,130,35), A \%_{290} D (360,290,25), B /_5 C (25,5,45), B <_{155} D (25,155,35), C <_{115} D (25,115,35) \}$$

$$C_{f_7} = \{ AdB, AdC, AdD, BpC, BpD, CdD \}$$

$C_{f_q} \subset C_{f_7} \rightarrow f_7$  is category-similar to  $f_q$ .

$R_{f_q}^{rA} \subset R_{f_7}^{rA}$  and  $R_{f_q}^{sA} \subset R_{f_7}^{sA} \rightarrow f_7$  is spatial-similar to  $f_q$ .

$O_{f_q}^{rdA} \subset O_{f_7}^{rdA}$  and  $O_{f_q}^{sdA} \subset O_{f_7}^{sdA} \rightarrow f_7$  is RS-metric-similar to  $f_q$ .

圖(11)-continued.

- $f_7$  與  $f_q$  為 RS 型計量相似：因為  $f_7$  與  $f_q$  符合空間型相似的條件，並且  $O_{f_q}^{rdA} \subset O_{f_7}^{rdA}$  以及  $O_{f_q}^{sdA} \subset O_{f_7}^{sdA}$ ，所以  $f_7$  與  $f_q$  為同心圓型計量相似。

以下為此範例在經過相似度檢索演算法後的最終結果：

$$\text{Category\_Similar\_List} = [f_3, f_4, f_5, f_6, f_7]$$

$$\text{Spatial\_Similar\_List} = [f_4, f_5, f_6, f_7]$$

$$\text{Ring-Metric\_Similar\_List} = [f_5]$$

$$\text{Sector-Metric\_Similar\_List} = [f_6]$$

$$\text{RS-Metric\_Similar\_List} = [f_7]$$

由上述結果可知，RS<sup>+</sup>-string 在影像檢索上，比 RS-string 更為完整，除了在比對上更加精確外，也更能符合使用者所下達的影像查詢之需求。

## 7. 結論

2D string 在圖像索引的技術上開啟了另一道新的研究領域，而之後相關的研究有 2D G-string、2D C-string、2D B-string、2D C<sup>+</sup>-string、2D Z-string、RCOS-string 等，這些研究都是植基於卡笛生座標系統(Cartesian coordinate system)的投射方式，將物

件個別投射在 x 軸與 y 軸，再從這兩軸所得到的字串索引來探討物間之間的空間關係，然而，在卡笛生座標系統的投射方式，其主要的缺點在於一旦影像經過旋轉後則所產生的字串索引方式可能會與旋轉前的字串索引不一致，而導致影像誤判的情況發生，因此，假如一張查詢影像與資料庫影像恰巧完全相同，但旋轉角度不同的話，則此兩張影像還是有可能會被誤判成不相似，對於一個良好的影像資料庫系統而言，它必須能夠自動處理此方面的狀況，而非要求使用者在下達影像查詢時必須提供角度方位的資訊。

在影像資訊系統中，為了使影像檢索能夠達到比對更精確與更快速查詢的目標，發展一套良好的空間知識表示法更加顯得重要，透過空間知識表示法所儲存的空間資訊作為影像索引，可支援空間推論與相似度檢索，以滿足使用者對於影像查詢的需求。而 RS-string 與 2D Z-string 兩者乃相當著名的空間知識表示法，RS-string 因植基於極座標系統，故，能夠達到旋轉不變之特性。另一方面，植基於卡笛生座標系統的 2D Z-string，其在字串中記載物件大小、位置、距離等計量資訊，並且藉由零切割的機制，使得其字串長度比 2D C<sup>+</sup>-string 精簡。對於 RS-string 與 2D Z-string 而言，兩者都克服了 2D string 相關研究的一些缺點。然而，我們發現在某些情況下，RS-string 仍有精確度不足之處，以致會做出相似比對的誤判，此外，2D Z-string 因受限於卡笛生座標系統的限制，一旦影像經過角度旋轉後，則可能會導致相似度比對的誤判。因此，在本



篇論文中，我們以 RS-string 為基礎，並且，排除在極座標系統中引用 2D Z-string 概念的困難，提出 RS<sup>+</sup>-string 空間知識表示法。RS<sup>+</sup>-string 應用了 2D Z-string 的零切割方式來精簡字串長度，並能加入物件大小、位置、距離等計量資訊。因此，RS<sup>+</sup>-string 在字串表示結合了 RS-string 與 2D Z-string 之優點，並且克服了此兩者空間知識表示法的缺點，因此，RS<sup>+</sup>-string 不但提升了儲存空間的上效益，同時也達到了更精確的空間推論與相似檢索，以符合使用者的需求。

## 參考文獻

- [1] Anthony J. T. Lee and H. P. Chiu, "2D Z-string: A new spatial knowledge representation for image databases." *Pattern Recognition Letters* 24(16): 3015-3026, 2003.
- [2] C. C. Chang and C. F. Lee, "Relative coordinates oriented symbolic string for spatial relationship retrieval." *Pattern Recognition* 28(4): 563-570, 1995.
- [3] Erland Jungert, "Extended Symbolic Projections as a Knowledge Structure for Spatial Reasoning." *Proceedings of the 4th International Conference on Pattern Recognition* 301: 343 – 351, 1988.
- [4] G. Petraglia, M. Sebillio, et al., "Towards normalized iconic indexing." *Proceedings of the 1993 IEEE Symposium on Visual Languages*: 392-394, 1993.
- [5] G. Petraglia, M. Sebillio, et al., "Rotation invariant iconic indexing for image database retrieval." *Progress in Image Analysis and Processing III*: 271-278, 1994.
- [6] G. Petraglia, M. Sebillio, et al., "Virtual images for similarity retrieval in image databases." *IEEE Transactions on Knowledge and Data Engineering* 13(6): 951-967, 2001.
- [7] P. W. Huang "Indexing pictures by key objects for large-scale image databases." *Pattern Recognition* 30(7): 1229-1237, 1997.
- [8] P. W. Huang and P. L. Lin "Visibility Inference Based on Spatial Knowledge Representation From Observer's Perspective." *International Journal of Intelligent Systems* 12: 191-202, 1997.
- [9] P. W. Huang and Y. R. Jean, "Using 2D C+-strings as spatial knowledge representation for image database systems." *Pattern Recognition* 27(9): 1249-1257, 1994.
- [10] P. W. Huang and Y. R. Jean, "Spatial reasoning and similarity retrieval for image database systems based on RS-strings." *Pattern Recognition* 29(12): 2103-2114, 1996.
- [11] S. K. Chang and A. Hsu, "Image information systems: where do we go from here?" *IEEE Transactions on Software Engineering* 4(5): 431-442, 1992.
- [12] S. K. Chang, C. W. Yan, et al, "An intelligent image database system." *IEEE Transactions on Software Engineering* 14(5): 681-688, 1988.
- [13] S. K. Chang, E. Jungert, et al., "Representation and retrieval of symbolic pictures using generalized 2D strings." *Technical Report*, University of Pittsburg, 1988.
- [14] S. K. Chang, Q. Y. Shi, et al., "Iconic indexing by 2-D strings." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 9(3): 413-428, 1987.
- [15] S. Y. Lee and F. J. Hsu, "2D C-string: a new spatial knowledge representation for image database system." *Pattern Recognition* 23(10): 1077-1087, 1990.
- [16] S. Y. Lee and F. J. Hsu, "Picture algebra for spatial reasoning of iconic images represented in 2D C-string." *Pattern Recognition Letters* 12(7): 425-435, 1991.
- [17] S. Y. Lee and F. J. Hsu, "Spatial reasoning and similarity retrieval of images using 2D C-string knowledge representation." *Pattern Recognition* 25(3): 305-318, 1992.
- [18] S. Y. Lee, F. J. Hsu, et al., "Similarity Retrieval by 2D C-Trees Matching in Image Databases." *Journal of Visual Communication and Image Representation* 9(1): 87-100, 1998.
- [19] S. Y. Lee, M. K. Shan, et al., "Similarity retrieval of iconic image database." *Pattern Recognition* 22(6): 675-682, 1989.
- [20] Wang, Y. H., "Image indexing and similarity retrieval based on spatial relationship model." *Information Sciences* 154(1-2): 39-58, 2003.
- [21] Y. I. Chang, B. Y. Yang, et al., "A generalized prime-number-based matrix strategy for efficient iconic indexing of symbolic pictures." *Pattern Recognition Letters* 22(6-7): 657-666, 2001.
- [22] Y. I. Chang, B. Y. Yang, et al., "A bit-pattern-based matrix strategy for efficient iconic indexing of symbolic pictures." *Pattern Recognition Letters* 24(1-3): 537-545, 2003.
- [23] Y. I. Chang, H. Y. Ann, et al., "A unique-ID-based matrix strategy for efficient iconic indexing of symbolic pictures." *Pattern Recognition* 33(8): 1263-1276, 2000.