

在無線廣播環境查詢反向最近者

Search RNN on Air

林聯發

高苑科技大學資訊傳播系

中山路1821號

高雄縣821路竹鄉

lienfa@cc.kyu.edu.tw

Lien-Fa Lin

Dept. of Information Communication, University of Kao Yuan

No.1821, Jhongshan Rd.,

Lujhu Township, Kaohsiung County 821

lienfa@cc.kyu.edu.tw

摘要

提供與物體位置有關的查詢服務被稱為 Location Based Services (LBSs)，而支援LBS 的查詢稱為與位置相關查詢(Location Dependent Queries-LDQ)。反向最近鄰居的查詢(Reverse Nearest Neighbor Queries-RNN)，即是屬於這類服務。就如最近鄰居查詢(Nearest Neighbor Queries-NN)，RNN 查詢也出現在很多的實際應用方面，包括決策支援系統、市場決策、資料庫文件搜尋、生物資訊等。因此提供有效率的方法來處理資料庫RNN查詢有其必要性。雖然RNN的問題在傳統有線以磁碟為基礎的主從架構(disk based client-server)環境有很好的研究，在無線的廣播環境並沒有處理過RNN 的問題。

本論文中，我們討論如何在無線廣播的環境有效的組織與位置相關的資料以及回答 RNN 查詢的問題。無線廣播線性存取以及行動裝置必需考量節省電力的特性使得這個問題特別有趣也更具有挑戰性。我們針對無線廣播環境提出一個可適應線性存取與有效節省電力稱為Jump-Rdnn tree 的索引架構以及相對的RNN查詢演算法。我們設計了大量的實驗來驗證我們的方法，實驗結果驗證我們方法在效能方面有顯著的提升。

關鍵詞：索引結構，資料廣播，能源管理，行動計算

Abstract

Location-based services (LBSs) provide information based on location information specified in a query. Queries that support for LBS are called Location-Dependent Queries (LDQ). One such query is the Reverse Nearest Neighbor (RNN) query that returns the objects that have a query object as their closest object. Just like the Nearest Neighbor (NN) queries, the RNN queries appear in many practical applications such as decision support system, continuous referral systems, profile-based marketing, maintaining document repositories, bioinformatics, etc. Thus efficient methods for the

RNN queries in database are required. While the RNN is well studied in the traditional wired, disk-based client-server environment; it has not been tackled in a wireless broadcast environment. In this paper, the issues involved with organizing location dependent data and answering RNN queries on air are investigated. The linear property of wireless broadcast media and power conserving requirement of mobile devices make the problem particularly interesting and challenging. An efficient data organization, called Jump-Rdnn Tree, and the corresponding search algorithm are proposed. Performance of the proposed Jump-Rdnn Tree and other traditional indexes (enhanced for wireless broadcast) is evaluated using both uniform and skew data. The result shows that Jump-Rdnn Tree substantially outperforms the traditional indexes.

Keywords: index structure, data broadcast, energy-conserving, mobile computing

壹、緒論

隨著無線網路通訊與行動裝置技術的蓬勃發展以及定位技術的逐漸成熟，LBS 漸漸在業界與學術上成為發展的重點[7, 15, 18, 20, 21, 22]。在 "IT Roadmap to a Geospatial Future" [12]的報告中，"LBSs 將迎接任意時間與任意地點皆能計算 (pervasive computing) 的時代，並且在即將到來的十年將重新的改造大眾宣傳媒介、行銷以及不同的社會面向"。雖然 LBSs 在傳統的計算環境裡就已經存在 (例如，Yahoo! Local)，但其實它最大的發展潛力還是在於行動計算方面。因為這樣的環境，可提供使用者充分的行動性與隨處可得的資訊。

隨著所需技術的日漸成熟以及預期廣佈於各地的 3G 無線通訊的基礎架構 (wireless communication infrastructure)，LBSs 將會成為行動網路中一項不可或缺的應用。提供關於 LBSs 的查詢，我們稱為位置相關性查詢 (Location Dependent Query-LDQ)。LDQ 的應用包括了範圍查詢 (Range Query)、最近鄰居查詢 (Nearest Neighbor Query)、k 個最近鄰居查詢 (K-Nearest Neighbor Query-KNN)、連續最近鄰居查詢 (Continuous Nearest Neighbor Query-CNN)、連續 k 個最近鄰居查詢 (Continuous K-Nearest Neighbor Query-CKNN) 以及反向最近鄰居查詢 (Reverse Nearest Neighbor Query-RNN) 等。

過去 LDQ 的研究，關於 NN [6, 18, 25]，KNN [4, 11, 19, 24, 25]，CNN [9, 17, 23, 29]，CKNN [17, 29] 的查詢研究可說是相當得熱門且豐富。近年來有關 Reverse Nearest Neighbor (RNN) Query [3, 6, 10, 13, 16, 27, 28] 的問題也開始受到眾人的注意。所謂的 RNN 查詢是指給予某個物件集合 S 和某一個查詢物件 q，找出在 S 中 q 是它們的最近鄰居 (NN) 的所有物件集合。在 [13] 提到許多關於 RNN 查詢的應用實例。例如，銀行欲在某地開設新的分行。假設，客戶們總是會選擇離他們較近的分行洽公，那麼這個新開設的分行，應該設在多數的客戶到此分行的距離，要比起到其他銀行的距離還靠近的位置上。另外，舉日常生活中計程車司機如何選擇載運客人的應用實例。假設，計程車可以藉由無線裝置得知欲搭乘計程車的客人所在位置，若從競爭者的角度來思考，尋找 RNN 會比尋找 NN 來得有意義許多，以圖 1 來說明，a 車的 NN 是乘客 c，但並不表示 a 車最有可能載到 c

乘客。因為還有離c乘客更近的b車在附近。相反的，a車應該去載d乘客，因為就d而言，a車才是他的NN。也就是說，a的RNN為d，a車可以比起其他的計程車更快到達乘客d。

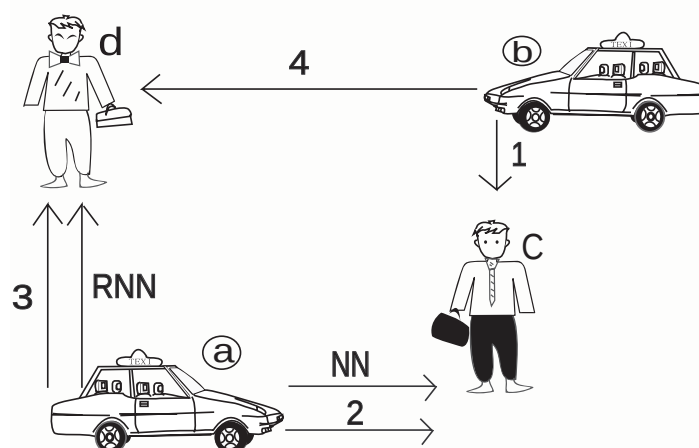


圖1. RNN 查詢範例。

隨著愈來愈多的行動裝置使用人口，如何能不因使用人數增加而仍能有效提供 LBSs 的服務是一大的挑戰。無線廣播技術是解決此一問題的方法隨著愈來愈多的行動裝置使用人口，可以不因使用人數增加而仍能提供 LBSs 的服務是一大的挑戰。無線廣播技術是解決此一問題的方法[1, 5, 8]。將資料利用廣播頻道播出，可以允許同一時間任一數目的行動使用者(Mobile User-MU)抓取資料。另外，為了有效率節省行動裝置的電力，一般性的作法是將資料與索引(on air index)交錯的透過廣播頻道播放。透過索引資料MU可以知道所需的資料何時播放，故可進入省電的睡眠模式(doze mode)直到所需資料到達才醒來以行動模式(active mode)抓取資料，不用一直以行動模式等待所需的資料到達而浪費電力。這樣利用廣播索引技術(on air index technology)使MU可利用選擇性切換不同的模式(selective tuning)達到電力節省的研究相當的豐富且熱門[5, 15, 16, 19, 26]。Guttman 於1984年提出R-tree[2]是目前使用最廣泛的以物件為基礎的索引方法。此索引架構雖然能處理各種類別的LDQ，但當應用在如 NN、KNN 與 RNN 的查詢時，需要利用回溯追蹤(back-tracking)的技術來加速查詢的處理。但是，回溯追蹤的技術不適用在只能提供線性存取(linear access)的廣播環境。無線環境下行動裝置由於電力有限如何有效的節省電力是一重要的課題，因此針對一般的查詢處理如何能有效的進行行動裝置能源管理的文獻相當豐富[18, 19, 22, 26]。從這些研究我們歸納出一些設計一個好的支援RNN 查詢的廣播索引(on air index)的準則。除此之外，經由模擬的實驗，證明我們的方法比起改良後適用於廣播環境的Rdnn-tree 索引效能有明顯的提昇。此一研究，就我們知識所及是第一個研究如何在廣播環境可以同時有效的支援RNN查詢的廣播索引架構。本論文其它節的架構如下，在

下一節介紹相關研究。第3節提出一些設計有效率的廣播索引(On Air Index)的規則。第4節介紹我們設計支援廣播環境找尋RNN的索引架構。第5節介紹實驗的結果。最後做一簡單的結論並說明我們未來繼續研究的方向。

貳、相關研究

本論文相關的研究包括有RNN查詢、無線資料廣播(Wireless Data Broadcast)以及廣播索引(On Air Index)等三方面。在下面各小節將分別介紹這三方面的相關研究議題。

(一) 反向最近鄰居查詢(Reverse Nearest Neighbor Query)

所謂的RNN query是指給予某個物件集合 S 和某一個查詢物件 q 找出在 S 中 q 是它們的最近鄰居(NN)的所有物件集合。RNN查詢應用相當廣泛，包括決策支援系統、市場決策、資料庫文件搜尋、生物資訊等。在[6, 10, 14]提到許多關於RNN查詢的應用實例。

若利用暴力(brute force)的方法去找RNN，首先必需先找到每一物件的最近鄰居，然後再看看這些物件中有那些物件的最近鄰居是查詢點。不過這個方法，當要處理的資料量太大時則顯得不可行，因為利用此方法的時間複雜度是 $O(N^3)$ 。故一般性的方法是利用一個稱為RNN tree[10]的特別R-tree來處理RNN的問題。Congjun Yang[6]等改善[10]的作法，提出Rdnn-tree的架構，可以同時處理NN與RNN的問題，Rdnn-tree與R-tree不同之處在於，它另外儲存了有關各個物體的NN資訊，可以更有效率處理RNN的問題。

(二) 無線資料廣播(Wireless Data Broadcast)

在無線廣播的環境，有兩種方法可以提供MU來抓取廣播中的資料：

需求性存取(On-Demand Access):由MU對伺服器提出查詢要求(submits query to server)。伺服器可以利用以磁碟為基礎的空間索引(disk-based spatial index)來加速查詢的處理，增進資料存取的效能。針對MUs的需求，由伺服器端負責過濾找出MUs所需的資料並回傳結果給MUs。

廣播與過濾(Broadcast and Filter):資料以週期性的方式在公共無線頻道上廣播。MU並不需要對伺服器提出需求，而是直接切換進入廣播頻道(tune into the broadcast channel)聽取他所需的資料。

需求性存取是利用基本的主從架構模式(client-server model)，由伺服器負責處理查詢(query)並藉由點對點(point-to-point)的專屬頻道(dedicate channel)直接回傳結果給

使用者。然而需求性存取適用在當對於無線頻寬的競爭(contention)與伺服器的行程處理(server processing)不是太嚴重，負載較輕的系統。當使用者的人數增加，則系統的效能將快速的遞減。至於已經被使用在廣播與電視(radio and TV)工業的無線廣播，則不管使用者的人數有多少，對伺服器的負載是一樣，伺服器還是只傳送一份資料，是一種很自然用來解決使用人數的擴充性(scalability)與頻寬問題(bandwidth problems)的方法。

除此之外，因為行動裝置的電能有限，有效的節省電力是無線環境下行動裝置的一大課題。為了節省電力，行動裝置一般設計成行動模式(active mode)與睡眠模式(doze mode)[26]兩種操作模式。以典型的無線個人電腦網路卡(wireless PC card)在睡眠模式消耗約60mW以及在行動模式消耗約 805-1400mW[20]。在無線的廣播環境節省電力的作法是利用在廣播資料中加入索引資料提供資料在廣播頻道的播出時間。借由查詢索引資料使用者知道所需資料的播放時間，因此可以進入較省電的睡眠模直到所需的資料播放才進入行動模式抓取資料。

(三) 廣播索引(On Air Index)

在傳統磁碟存取(disk-based)的環境，回溯追蹤(back-tracking)通常被利用在查詢演算法來增進查詢的效能。然而這在只能線性存取(linear-access)的廣播頻道將會產生問題。在無線廣播的環境，只有當索引的資料正在廣播時，使用者才能存取索引資料。因此，當演算法抓取索引資料的順序(traversal sequence)與它們被廣播的順序(broadcast sequence)相反，則使用者必需等到該索引資料下一次的播放才能抓取，因此浪費較長的存取時間延遲(access latency)。相反的，在傳統的資料庫，索引資料儲存在如磁碟(disk)或記憶體(memory)的常駐儲存體，可以在任何的時間抓取。因為線性的存取並不是傳統索引結構的設計考量，故目前應用在磁碟存取支援空間索引的演算法無法達到有效能源節省的需求。以圖2為例，查詢空間(search space)包括 O_1 、 O_2 、 O_3 、及 O_4 等四個物件如圖2(a)。而在這些物件上建立的 R-Tree 索引架構如圖2(b)所示。若將圖2(b) R-tree 索引資料經由廣播頻道廣播則廣播順序為 root, R_1 , R_2 , O_1 , O_3 , O_2 , O_4 如圖3(a)所示。若給一個查詢點 q_2 ，若要找距離查詢點 q_2 最近的物件則最近的物件索引查詢(index traversal)拜訪的順序(visit sequence)是先拜訪根節點(root)，然後拜訪離 q_2 較近的節點 R_2 接下來拜訪葉節點(leaf node)內的物件 O_2 , O_4 ，發現 O_2 與 O_4 皆不是 q_2 的最近物件故必需回溯拜訪節點 R_2 以及 R_1 的葉節點(leaf node)內的物件 O_1 , O_3 最後找到最近的物件 O_3 。然而因為廣播順序與查詢的順序不同，當拜訪 R_2 葉節點(leaf node)內的物件 O_2 , O_4 發現 O_2 與 O_4 皆不是 q_2 的最近物件要拜訪 R_1 的葉節點(leaf node)內的物件 O_1 , O_3 時發現 R_1 , O_1 , O_3 已經播過，導致如圖3(a)所示使用者必需要等到下一廣播周期(broadcast cycle)才能抓取到所需的資訊，故使用者會有很長的存取時間延遲(access latency)。因此，傳統用於空間查詢分支與設定範圍(branch-and-bound)的查詢方法(search approach)對存取時間延遲(access latency)而言不是很有效率。另外一個選

擇是，也是目前大部份文獻用來解決空間查詢的方法，那就是我們可以如圖3(b)循序(sequentially)的方式直接存取 MBRs。然而，這個方法會帶來不必要的 MBRs traversal，將不會有最好的索引查詢效能(index search performance)。例如，針對查詢點 q_1 找尋NN，真正的NN是在MBR R_2 中的 O_4 ，很明顯抓取 R_1 是無謂的浪費資源。故針對無線廣播環境必須設計一套新的索引機制，才能有效適應廣播環境現性存取與節省電力的考量。

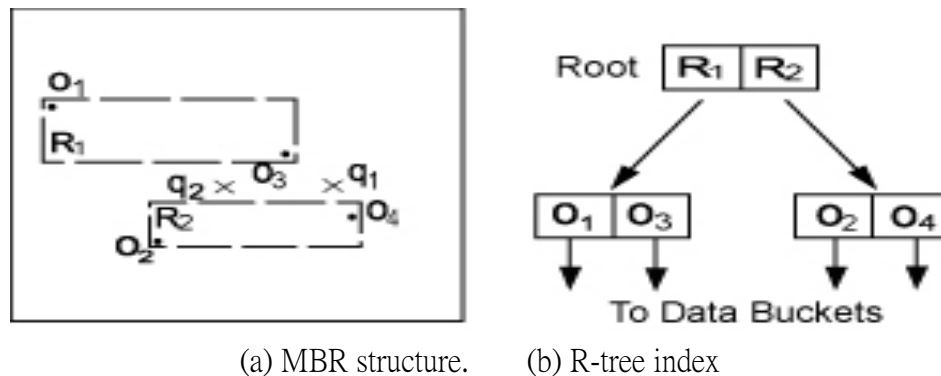


圖 2. R-tree 索引的執行範例.

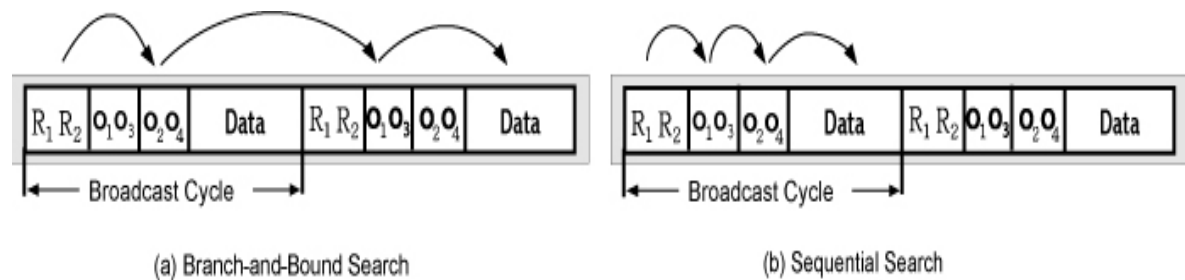


圖3.無線廣播頻道的線性存取方式.

參、有效率廣播索引的設計規則

資料存取的延遲時間 (access latency)與行動裝置在行動模式(active mode)下所花的時間(tuning time)是兩個用來衡量廣播索引效能的標準。資料存取的延遲時間(access latency)是從使用者下達查詢開始直到抓取到所有滿足查詢所需的資料的這段時間。行動裝置在行動模式(active mode)下所花的時間(tuning time)則是使用者以行動模式 接收所需資料的時間。廣播索引將廣播資料與索引混合播放，行動使用者抓取資料將包括下述3個步驟[16]。

(1)初始進行資料探索(Initial probe):使用者在廣播的任一時間點進入廣播頻道一直等到播放索引資料，這一段時間稱為初始進行資料探索的等待時間(initial probe waiting)。

(2)索引搜尋(Index search):使用者在索引到達時，抓取索引資料並根據查詢需求選擇性存取一些索引資料，找出所需資料所在的位置。

(3)資料抓取(Data retrieval):當所需的資料到達時，使用者下載並存取。

這3個步驟所花的時間將會影響到廣播索引的效能。因此設計一個有效的廣播索引必須降低這3個步驟所花的時間，底下我們從文獻的整理與分析歸納出針對這3個步驟如何降低該步驟所花的時間。

■ **降低初始進行資料探索的時間:**初始進行資料探索的等待時間是使用者等待索引資料出現的時間，在整個廣播周期(broadcast cycle)複製多份索引，可增加索引出現的機率，可降低初始進行資料探索的等待時間，1997 Imielinski 等人[8]利用如圖 4 (1,m) 交錯(interleaving)重複配置 m 份索引資料的方式來降低初始進行資料探索的等待時間。

■ **減少索引的資料量:**索引搜尋的時間與整個索引資料的大小有關，索引資料愈小，則索引搜尋時間將愈短，故整個廣播周期將相對愈短，故平均的資料存取的延遲時間也將愈小。如 1997 Imielinski 等人[26]只複製索引樹(index-tree)上面 k 層，來降低整個索引資料的大小。Quinglong Hu 等人[19] 則利用抓取特徵性(signature)的技術縮小索引資料。

■ **有效的置放廣播資料(Efficient data placement):** 2003 年 Ming-Shan Chen 等人[18]證明不同資料不同的播放順序將影響平均的資料存取的延遲時間，提出 ORD 演算法來降低平均的資料存取的延遲時間。而 2002 年 Jianting 與 Le Gruebwald [22] 依存取頻率安排廣播資料(broadcast)播放的先後次序，來降低資料存取的延遲時間。

目前廣播索引的研究通常只針對單一步驟效能的改進，並沒同時考量3個步驟的效能改進，本論文將同時考量3個步驟的效能改進以及考量廣播索引只能線性存取的特性，設計一個新的廣播索引來處理廣播的RNN 查詢，詳細的方法將在下一節詳細的說明。

肆、廣播環境下執行 RNN 查詢的新索引架構

某一查詢(query) q 的RNN是那些以查詢點(query point) q 為NN(nearest neighbor)的那些物件的集合，若能事先知道物件與其NN距離，那麼只要query q 與物件的距離，比起這些物件到其NN的距離還要近，則這些物件即為 query q 的RNN。因此若要能更快速找尋RNN，則物件本身必須先算出物件本身與其NN的距離。Rdnn-tree與R-tree不同之處在於，它另外儲存了有關各個物體的NN資訊(各物體到其最近物件的距離)，可以直接判斷葉節點(leaf node)是否為查詢的結果，不像如前一節所談的 R-tree 無法直接判斷葉節點 (leaf node) 是否為查詢的結果必須利用分支與限定範圍(branch-and-bound)的技術而有回溯追蹤(back-tracking)的問題，不適合應用在只能線性存取(linear access)的廣播環境，故

我們論文採Rdnn-Tree的索引架構並依據設計有效率的廣播索引(On Air Index)的規則進一步的改良以增進廣播環境下查詢RNN的效能。

(一) Rdnn-Tree

R-Tree[2]早期是發展在空間資料庫(spatial database)領域下的索引，2000年被 Korn與 Muthukrishnan[10]改良成 Rdnn-Tree (R-tree containing Distance of Nearest Neighbors)，用來加速RNN 查詢的處理。作者提出只需利用單一的索引結構(Index Structure-亦即Rdnn-tree)，便可以同時解決 NN查詢 與 RNN查詢。Rdnn-tree 與 R-Tree 不同的地方是多儲存了每一節點到其NN 的距離資訊。Rdnn-Tree的結構，如圖4，根據物體座標相似性將所有物體分成一群一群存放在樹葉節點(leaf node)，也就是座標相近的物體分在同一群。然後，以一個最小的矩形(rectangle)將這一群的物體包含住，此最小的矩形(rectangle)稱為MBR (Minimum Bounding Rectangle)。接下來將相近的MBR再分成一群一群，以一個更大的MBR將一群含蓋較小範圍的MBR包含住。以此類推，直到所有物體皆被包含在同一個MBR 為止。在Rdnn-Tree的內部節點(internal node)存放的是 MBR，只要是在它底下的節點都會被它包含住。Rdnn-Tree 的根節點(root)會將所有物體都包含住。至於每個MBR 都是記錄其左下點的座標(M_l, M_d)跟右上點的座標 (M_r, M_u)，這樣就能知道整個MBR的大小範圍。在葉節點(leaf node)的部分，存放的是(ptid, dnn)，ptid代表資料集中物件點的編號，dnn 則是該物體與其 NN的距離。在非葉節點(non-leaf node)的部分，則存放(ptr, rect, MaxDnn)，其中ptr 指向子節點(child node)的位址，rect 則是包含此節點以下所有子節點(child nodes)的MBR，MaxDnn 是此子樹中所有物體其dnn 的最大值，代表該子樹底下的所有物體到其 NN 的最大距離不會超過 MaxDnn。圖4是 Rdnn-Tree 的範例，在此範例中共有a 到 l等12個物件。其中物件 a, b, c, d 被 MBR b1 所包含，物件e, f, g 被MBR b2 所包含，物件h, i 被MBR b3 所包含，物件j, k 被MBR b4 所包含，又MBR b1與 MBR b2 以及MBR b3與MBR b4 又分別被一更大的 MBR B1以及MBR B2 所包含，最後 MBR B1與 MBR B2 則被最大的MBR B 包含。

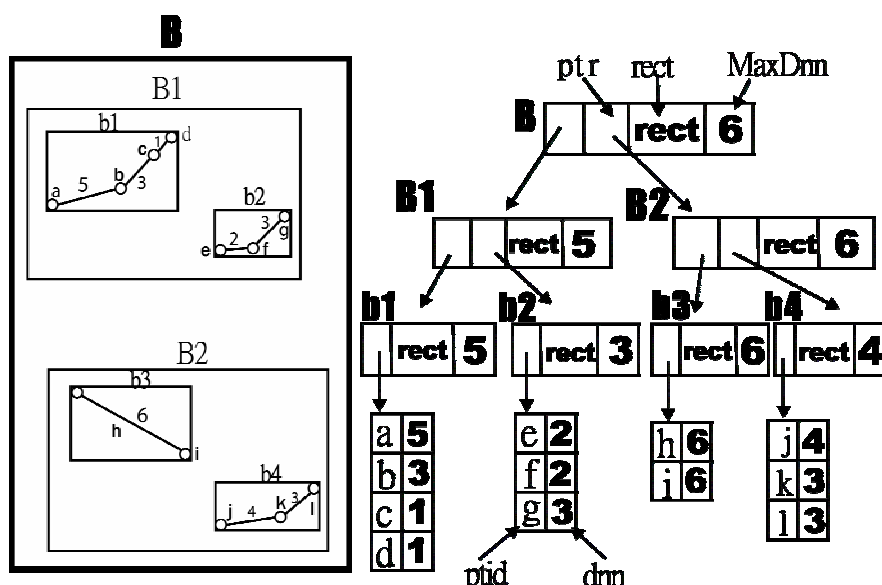


圖 4. Rdn-Tree的資料結構.

(二) Jump-Rdn Tree (Jump Rdn-tree)

設計一個良好的廣播索引(On Air Index)如第3節所述包括減少初始進行資料探索的時間、減少索引資料量以及有效率置放廣播資料三個階段。底下說明我們如何在這三個階段來改良Rdn-Tree。

■ 減少初始進行資料探索的時間：傳統的作法是利用複製索引的方法，增加索引出現的機率。然而這樣的做法會使得整個廣播週期(broadcast cycle)變長，導致平均的資料存取的延遲時間增長。我們的作法是，是將廣播資料以我們提出的索引架構建成一棵所謂的 Jump-Rdn tree，以每一棵子樹(sub-tree)為單位將資料與索引混合播出，該子樹的索引播完緊接著播該子樹下面的資料，拉近資料與索引的距離，不是等整個索引資料播完，才開始播所有的資料。以圖 4 為例，廣播排程將是 B, B1, b1, a, b, c, d, b2, e, f, g, B2, b3, h, i, b4, j, k, l。另外採用所謂的循環性的索引結構(cyclic index structure)，每個索引節點(index node)先根據深先搜尋(Depth First Search-DFS traversal)的順序算出下個要拜訪的索引節點的位置，即是所謂的跳躍指標(Jump pointer)如圖 5 所示，因為每一索引節點皆有跳躍指標，索引節點透過跳躍指標相互連結形成循環，故索引查詢可從任何索引節點開始，不用從根節點開始搜尋，同時索引樹查詢依指標循序搜尋不會有回溯追蹤的問題。

■ 減少索引資料量：如前面所述，我們採用循環性的索引架構故整個廣播週期不需複製任何索引資料，只需一份索引資料，故索引資料量很小。另外，一般性的索引樹的架構，若該索引樹每一層最大的子樹有 f 個代表該索引樹的 fan-out 是 f ，每一索引樹若有 fan-out 個子樹(sub-trees)則需 fan-out 個指標(pointers)紀錄各個子樹的位址。無線廣播環境因為線性存取的特性，MU 只有兩種行為以行動模式

抓取資料或是以睡眠模式略過目前資料等待下一次以行動模式再抓取資料，也就是繼續下一個的動作(Next)與略過不抓取的跳躍動作(Jump)，不符合 Jump 的條件，就是 Next，故 Next 的行為不用特別紀錄。因此，任何搜尋樹的非葉節點只要保留一個跳躍(Jump)的指標即可，其它 f-1 個子樹的指標皆可去直接去除，降低索引資料量。

■ 有效率置放廣播資料：查詢點是根據整個查詢空間(search space)任意產生，每區域發生查詢的機會將影響到查詢的效能，若在查詢點發生愈多的區域將對應的索引資料先播出，則整個索引查詢的效能將愈好。假設，查詢發生的點位置是呈常態分布(uniform distribution)。因此，我們根據每一個子樹所在的 MBR 所佔面積安排子樹的播放順序，因為 MBR 所佔的面積愈大，代表此區域會發生查詢的機率愈高。

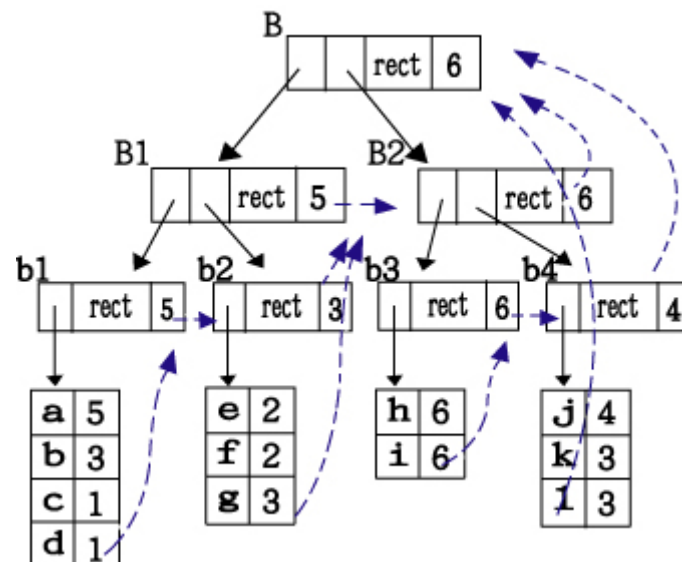


圖 5. Jump Rdn-Tree的資料結構

將Rdn-Tree 經由上述改善，除了不再有回溯追蹤的問題，而具循環性架構的Rdn索引樹適合廣播的線性存取。至於在廣播中查詢RNN(RNN Search on Air)詳細演算法如圖6所示，其中 $D(q, ptid)$ 代表查詢點 q 到物件 $ptid$ 的距離， $D(q, rect)$ 則代表查詢點 q 到MRB $rect$ 的距離。

RNN-Search-On-Air (Node n, Point q)

Case 1 :

if n 是葉節點(leaf node) then

For all在 n 裡的每一個 (data-item , dnn)

If $D(q, ptid) < dnn$ then data-item 是 q 的 RNN點

Case 2 :

If n 是 非葉節點(non-leaf node) then

For all n 的每一個 branch $B = (\text{ptr}, \text{MBR}, \text{Maxdnn})$,

If $D(q, \text{rect}) < \text{Maxdnn}$ call $\text{RNN-Search-On-Air}(B, \text{ptr}, q)$

圖 6. Algorithm for RNN Search on Air .

伍、實驗研究(Performance Study)

(一) 實驗環境(Experiment Environment)

(1) 兩種不同的實驗資料集(uniform & skew)

爲了驗證我們所提的方法在不同樣式的資料分布情況下的效能表現，我們使用兩個不同的資料集來進行實驗的評估。如圖7(a)所示第一個資料集 (UNIFORM)，我們在歐幾里德的平面採常態分布的方式(square Euclidean space uniformly)產生1,000個點。第二個實驗資料集(SKEW)如圖7(b)所示，我們利用Zipf分布產生1,000個點，其中Zipf分布的傾斜參數(skewness parameter) 設爲1.2。

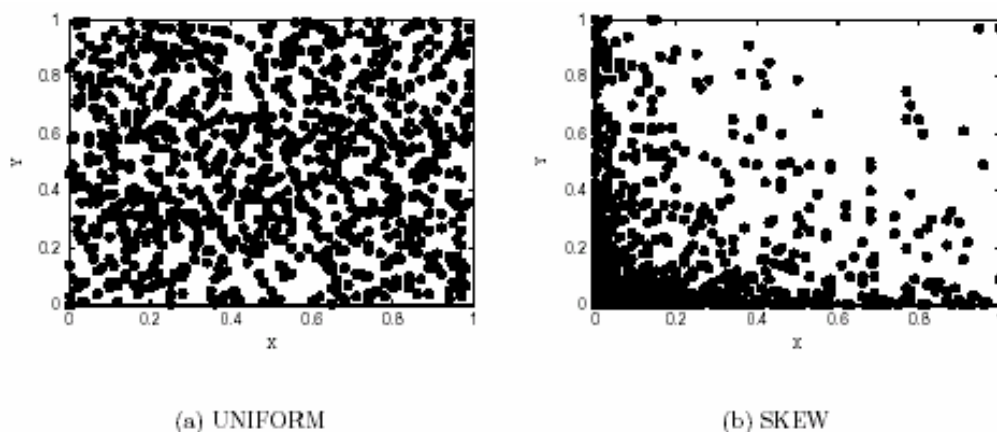


圖 7. Uniform and Skew data sets.

(2) 進行比較的演算法(Compared Algorithms)

就我們的知識所及目前在無線廣播的環境下並沒有處理 RNN search 的研究，然而傳統磁碟存取(disk-based)的環境發展出來的索引架構並不支援無線廣播線性存取的特性，因此我們將目前在磁碟存取(disk-based)的環境可以有效處理 RNN 查詢的 Rdn tree 的索引節點依深先搜尋DFS的排序方式循序式(sequentially)的方式存取以適應廣播環境必須線性存取的特性，同樣的搜尋過程中若不符合過濾(distance heuristics search)條件的子樹(sub-tree)將被過濾裁剪(pruned)掉。另外爲了降低減少初始進行資料探索的時間此處 Rdn-Tree 採取(1:m) Interleaving Technology的方式廣播我們稱此方法爲 Rdn-Tree(1:m)。

(3) 衡量效能的參數(Performance metrics)

一般測量廣播索引(on air index)效能使用的參數(metrics)是access time與tuning time，測量的單位是一個廣播單位packet(廣播的單位)。雖然tuning time可以大致反映行動裝置電能的消耗，但它只紀錄在active mode所花的電力不能真實的反應完整的電力消耗。雖然當MU在doze mode所耗的電力較少，但若等待的時間太長，則電力的消耗亦將很可觀。我們認為考量節省電力應該要看真正消耗的總電力，故我們實驗的效能評估參數(performance metrics)增加消耗總電力的參數(total power energy)。消耗總電力設為 $P = 1200 * \text{Time}_{\text{active}} + 60 * \text{Time}_{\text{doze mode}}$ 。為了簡化實驗的複雜度，在不影響實驗結果的前提下，我們忽略有關查詢處理其他會消耗電力的部份，假設1200mW包括全部抓取一廣播封包(broadcast packet)所需的電力，60mw為等待一廣播封包所需的電力。

(4) 參數設定(Parameters Setting)

實驗參數設定如表1所示，每一個廣播封包的封包編號(packet id)佔2 bytes，一個座標佔4 bytes，指標佔2 bytes。封包大小(packet size)變動從64 bytes到1024 bytes。Jump-Rdnn Tree的最小子樹(fan out)設為6。物件個數(Object#)變化從1000到5000，查詢是隨機的方式從整個查詢空間產生，使用者的初始探索的時間點為1到5000個廣播單位(broadcast unit)範圍中任意產生，最後的統計結果為30查詢的平均值[25]，實驗所用的程式是利用Gist[12]的R-tree程式碼加以改寫。

表 1. 實驗參數設定

Parameter	Description	Setting
Object#	資料物件的個數	預設值:1000 變動範圍:1000~5000
Packet Size	一個廣播封包的大小	預設值:256 bytes 變動範圍:64~1024
Fan out	Rdnn tree的子數個數	預設值:t:6

(二) 實驗結果(Performance Results)

(1) 封包大小對效能的影響

圖8、圖9是不同資料分布的實驗結果，就access time而言，不管資料分布是uniform或skew分布，隨著封包大小的變大，而有較小access time，這是因為大的封包容量(packet capacity)，可以容納更多的資料，因此廣播排程(broadcast program)的廣播週期(broadcast cycle)將相對較短所致。除此之外，我們的方法明顯優於Rdnn-Tree (1:m)的方法，我們的方法索引只播一次，相較於Rdnn-Tree採用(1:m)，將索引播m次的作法，廣播週期較短，有較小的access time。就tuning time而言，

隨著封包容量(packet capacity)的遞增，平均的封包存取時間(average packet access)遞減。但當資料是skew分布且封包容量大於1024 bytes時卻有tuning time些微遞增的效果，這是因為我們的方法採取資料與索引混合播出的方式，整個索引的資料是根據DFS 拜訪的順序分成幾個區段散落在整個廣播排程，當封包容量足夠大時可能有很多的索引區段不滿一個packet，但因為廣播排程必須區分索引封包(index packet)與資料封包(data packet)，這些不滿一個封包的索引區段仍然必須佔一個封包，這情況在資料是skew時情形較嚴重，故當封包容量變大時，整個平均的封包存取時間遞減，平均的封包存取時間的值較小時，因為比例的關係其效應較明顯。至於總消耗電力，不管資料的分布情形，我們的方法皆明顯優於Rdnn-Tree(1:m)的方法。雖然當資料是skew，封包容量大於1024 bytes時我們的方法tuning time稍為大於Rdnn-Tree (1:m)的方法，但比較總消耗電力我們的方法仍優於Rdnn-Tree(1:m)的方法，這符合我們的想法，當考量廣播索引的效能時必需考慮總消耗電力，因為這才能真正反應行動裝置的電力消耗。

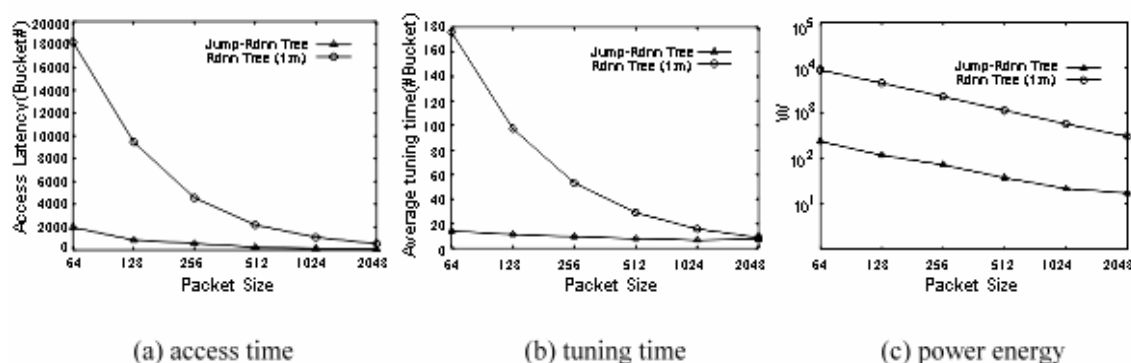


圖8. 資料物件呈常態分布下廣播封包容量的大小對效能的影響

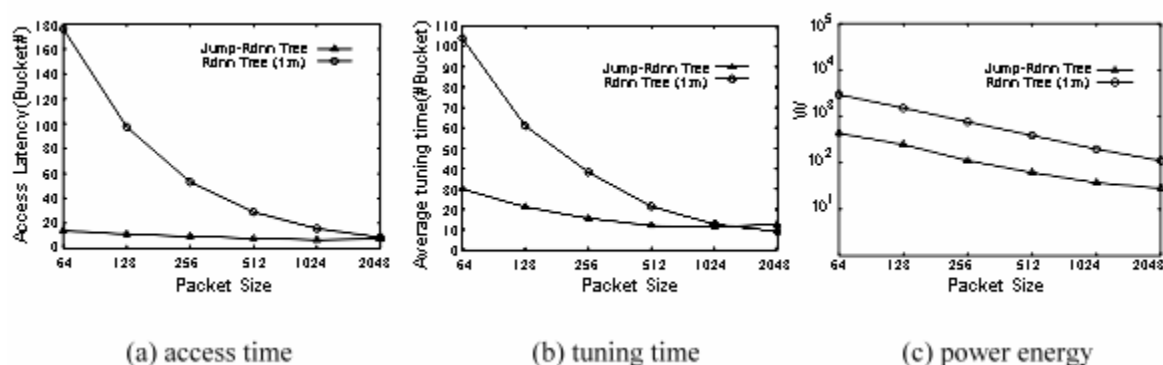


圖 9. 資料物件呈傾斜分布下廣播封包容量的大小對效能的影響

(2) 資料物件個數對實驗效能的影響

隨著物件(data objects)個數的增加，導致廣播排程的廣播週期變長故access time、tuning time以及總耗電力均因此增加。就access time而言，Rdnn-Tree (1:m)因為整個廣播週期變長，平均等待索引封包(index packet)播出的時間增長，且索引資料變大，其效果比是m倍的效果。我們的方法則增加的比較平緩。至於tuning time的效果亦如前面所述是因為提供使用者根據廣播資料是否符合使用者需求而選擇式切換以行動模式抓取資料或進入睡眠模式略過資料不抓取(selective tuning)。這種利用切換不同模式進行資料擷取的模式必需分索引封包與資料封包所致。因為我們的方法採取資料與索引混合播出的方式，整個索引的資料是根據DFS 拜訪的順序分成幾個區段散落在整個廣播排程，當封包容量足夠大時(預設256 bytes)又資料物件個數較少時(小於 2000個資料物件)相對的索引資料也較少，故有很多的索引區段不滿一個packet，但因為廣播排程必須區分索引封包(index packet)與資料封包(data packet)，這些不滿一個封包的索引區段仍然必須佔一個封包，因此才造成當物件資料較小時，Jump-Rdnn Tree 的 tuning time 會有較差的效能表現。於至於總消耗電力則不管物件個數為何，我們的方法仍明顯優於 Rdnn-Tree (1:m)。

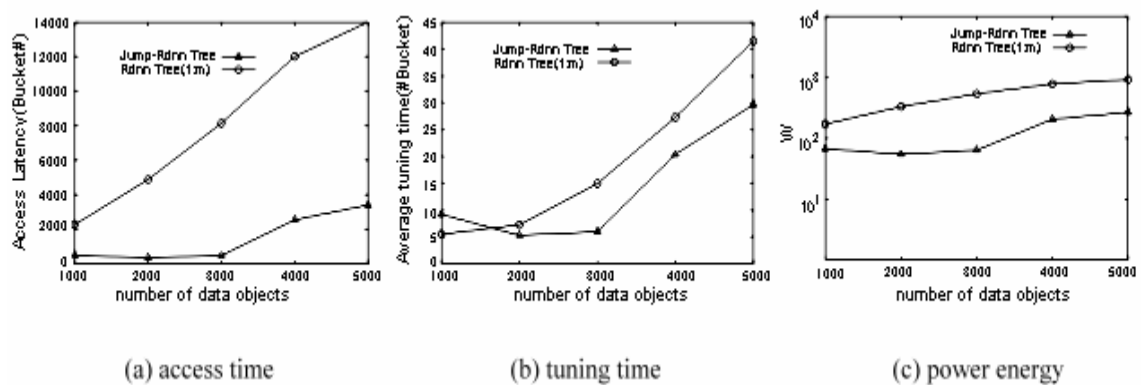


圖 10. 資料物件呈常態分布下資料物件多寡對實驗效能的影響

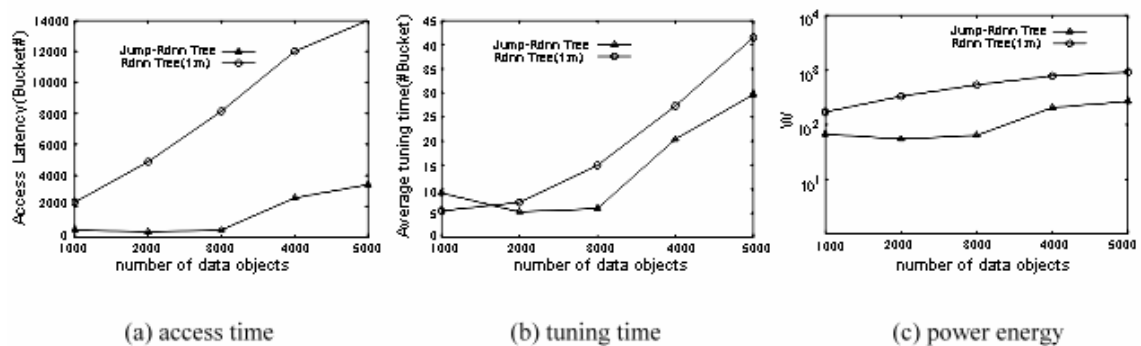


圖 11. 資料物件呈傾斜分布下資料物件多寡對實驗效能的影響

陸、結論與未來展望

RNN search 是行動計算領域一個很重要與實際的應用。雖然 RNN search 在傳統磁碟存取(disk-based)，client-server 的計算環境已經有很好的研究，在無線的廣播環境仍須更多的研究。此篇論文中，我們討論如何在無線廣播的環境有效的組織與位置相關的資料以及回答 RNN 查詢的問題。從以往的研究我們歸納出設計一個有效率廣播索引(on air index)的規則，根據這些規則，設計出適合在廣播環境查詢 RNN 的新索引架構Jump-Rdnn tree。試驗的結果證實我們的方法明顯優於目前廣為用於查詢RNN的Rdnn-Tree。目前此一研究只討論靜態查詢RNN的議題，未來我們將擴展至連續的查詢RNN。

柒、參考文獻

- [1] A. Datta, D.E. Vandermeer, A. Celik, and V. Kumar, "Broadcast Protocols to Support Efficient Retrieval from Database by Mobile Users", ACM Transactions on Database Survey
- [2] A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching", Proceedings of the 1984 ACM SIGMOD international conference on Management of data, 1984, p.p.47-57.
- [3] A. Singh, H. Ferhatosmanoglu and A.S. Tosun, High Dimensional Reverse Nearest Neighbor Queries, Proceedings of the 20th International Conference on Information and Knowledge Management, New Orleans, LA, USA, 2003, p.p.91-98.
- [4] B. Zheng, W.C. Lee and D.L. Lee, "Search K Nearest Neighbors on Air", Proceedings of the 4th International Conference on Mobile Data Management, Melbourne, Australia January 2003, p.p. 181-195.
- [5] B. Zheng and D.L. Lee, "Information Dissemination via Wireless Broadcast", Communication of ACM, 48(5), May 2005, p.p. 105-110.
- [6] C. Yang and K.L. Lin, "An index structure for efficient reverse nearest neighbor queries", Proceedings of the 17th International Conference on Data Engineering, 2001, p.p. 485-492.
- [7] Computer Science and Telecommunication Board. IT Roadmap to a Geospatial Future, the National Academies Press, 2003.
- [8] D. Barbara, "Mobile Computing And Databases - A Survey", IEEE Transactions on Knowledge and Data Engineering, Vol. 11, No. 1, February 1999, p.p. 108-117.
- [9] D.L. Lee, W.C. Lee, J. Xu and B. Zheng, "Data Management in location dependent services", IEEE Pervasive Computing, Vol. 1, No. 3, September 2002, p.p. 65-72.
- [10] F. Korn and S. Muthukrishnan, "Influence sets based on reverse nearest neighbor queries", Proceedings of the ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, p.p. 201-212.
- [11] G. Iwerks, H. Samet and K. Smith, "Continuous K-Nearest Neighbor Queries for Continuously Moving Points with Updates", International Conference on Very Large Data Bases, 2003, p.p. 512-523.

- [12] Gist , available on line , <http://gist.cs.berkeley.edu/>
- [13] G. Kollios ,D. Gunopulos and Vassilis J. Tsotras , "Nearest Neighbor Queries in a Mobile Environment Spatial-Temporal Database Management" , 1999 , p.p. 119-134.
- [14] I. Stanoi , D. Agrawal and A.E. Abbadi , "Reverse Nearest Neighbor Queries for Dynamic Databases" , ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery , 2000 , p.p.44-53.
- [15] J. Xu , B. Zheng , W.C.Lee and D.L. Lee", "Energy efficient index for energy query location-dependent data in mobile environments", In Proceedings of the 19th IEEE International Conference on Data Engineering, Bangalore, India March 2003, p.p. 239-250.
- [16] J. Zhang, M. Zhu ,D. Papadias ,Y. Tao and D.L. Lee, "Location-based Spatial Queries", Proceedings of the 2003 ACM SIGMOD international conference on Management of data, San Diego, California, USA , June 9-12 2003,p.p. 443-454.
- [17] J. Xu and W.C. Lee and X. Tang , "Exponential Index: A Parameterized Distributed Indexing Scheme for Data on Air"., "Proceedings of the 2nd ACM/USENIX International Conference on Mobile Systems, Applications, and Services , Boston, MA, June 2004,p.p. 153-164.
- [18] J. Zheng and L. Gruenwald , "Prioritized Sequencing for Efficient Query on Broadcast Geographical Information in Mobile Computing", ACM GIS , 2002,p.p.88-93.
- [19] M.S. Chen ,P.S. Yu and K.L. Wu , "Optimizing index allocation for sequential data broadcasting in wireless mobile computing" , IEEE Transactions on Knowledge and Data Engineering , Vol. 15 , No. 1,January-February 2003, , p.p. 161-173.
- [20] M. A. Viredaz , L. S. Brakmo and W. R. Hamburger , "Energy management on handheld devices", ACM Queue , Vol. 1 , No. 7 , October 2003 , p.p. 44-52.
- [21] N. Roussopoulos , S. Kelley and Fr'ed'eric Vincent , Nearest neighbor queries , Proceedings of ACM SIGMOD International Conference on Management of Data , June 1995 , p.p. 71-79.
- [22] Q. Hu , W.C. Lee and D.L. Lee , "Index Techniques for Power Management in Multi-Attribute Data Broadcast" , Mobile Networks and Applications , Vo. 6,No. 2,2001,p.p.185-197.
- [23] R. Benetis , C.S. Jensen , G. Karciuskas and S. Saltenis , Nearest Neighbor and Reverse Nearest Neighbor Queries for Moving Objects , International Database Engineering and Applications Symposium , Canada , July 17-19 , 2002 , p.p. 44-53.
- [24] S. Chaudhuri and L. Gravano , Evaluating top-k selection queries , Proceedings of the 25th IEEE International Conference on Very Large Data Bases , 1999 , p.p. 297-410.
- [25] Sheldon Ross , Introduction to Probability and Statistics for Engineers and Scientists , 2000.
- [26] T. Imielinski ,S. Viswanathan and B. R. Badrinath , Data on Air:organization and access , IEEE Transactions on Knowledge and Data Engineering , Vol. 9 , No. 3 , May 1997 , p.p. 353-372.
- [27] Y. Tao, D. Papadias and Q. Shen, "Continuous Nearest Neighbor Search", International Conference on Very Large Data Bases,Hong Kong China,August 20-23,p.p. 279-290.
- [28] Y. Tao , D. Papadias and X.Lian , "Reverse kNN Search in Arbitrary Dimensionality",

Proceedings of 30th Very Large Data Bases, Toronto, Canada, August 29-September 3, 2004, p.p.279-290.

- [29] Z. Song and Nick Roussopoulos, K-Nearest Neighbor Search for Moving Query Point, Proceedings of 7th International Symposium on Advances in Spatial and Temporal Databases, LNCS2121, Redondo Beach, CA, USA, July 12-15,2001,p.p. 79-96.

