

在廣播環境查詢反向最近鄰居

林聯發

高苑科技大學資訊傳播系，成功大學資工所博士班

lienfa@cc.kyit.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)。Reverse Nearest Neighbor (RNN) queries，即是屬於這類服務。RNN queries 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。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挑戰性。我們針對無線廣播提出一個可適應線性存取與有效節省電力稱為 Jump-Rdnn tree 的索引架構與相對的 RNN 查尋演算法。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：索引架構，資料廣播，能源管理，行動計算

1. Introduction

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 [1, 14, 21, 24]。在 "IT Roadmap to a Geospatial Future" [5] 的報告中，the Computer Science and Telecommunications Board (CSTB) 預測 "LBSs 將迎接 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，Google! 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世界各地的 3G 無線通訊基礎建設，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 LDQ (Location-Dependent Query)。LDQ 的應用包括了 Range query、Nearest Neighbor (NN) query、K-Nearest Neighbor (KNN) query 以及 Reverse Nearest Neighbor (RNN) query 等。過去 LDQ 的研究，關於 NN [13, 22]，KNN [4, 13, 23]，CNN [3, 17, 18]，CKNN [17, 18] 的研究是相當熱門且豐富的，近年來有關 Reverse Nearest Neighbor (RNN) Query [11, 15, 16, 18, 22] 的問題也開始受到眾人的注意。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 nearest neighbor 的所有物件集合。在 [11] 提到許多關於 RNN query 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要近的位置上。另外，舉一個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。

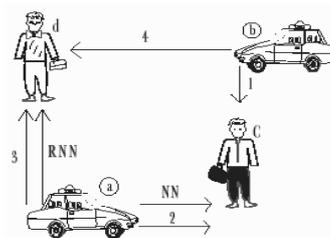


圖 1. Motivation of RNN Search

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。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 doze mode 直到所需資料到達才 active 抓取資料，不用一直 active 等待所需的資料到達而浪費電力。這樣利用廣播索引的技術使 MU 可利用 selective tuning 達到電力節省的研究相當的豐富且熱門 [8, 25, 26, 27]。從這些研究我們歸納出一些設計一個好的廣播索引的準則。我們利用這些準則來設計可以有效處理 RNN Search 的廣播索引。經由豐富的實驗，證明我們的方法比起改良後適用於廣播環境的 Rdnn-tree index 效能有明顯的提昇。此一研究，就我們知識所及是第一研究如何在廣播環境的有效回答 RNN 的查詢。本論文其它節的架構如下，在下一節介紹 Background and Related Work。第 3 節提出一些設

計 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。

2. Background and Related work

在下面各小節將分別介紹 RNN query、Wireless Data Broadcast 與 On Air Index 等與廣播環境 RNN Search 的相關研究議題。

2.1 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 nearest neighbor 的所有物件集合。RNN queries 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[11, 15, 22]提到許多關於 RNN query 的應用實例。

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[11]的特別 R-tree 來處理 RNN 的問題。Congjun Yang[22]等改善[11]的作法，提出 Rdn-tree 的架構，可以同時處理 NN 與 RNN 的問題，Rdn-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。

2.2 Wireless Data Broadcast

無線廣播的環境，有兩種方法提供 MU 抓取 LBSs 的資料。On-demand access 是由 MU 對 server 提出查詢要求。針對 MU 的需求，server 負責過濾找出 MU 所需的資料並回傳結果給 MU。Broadcast and Filter 則是資料以週期的方式在公共無線頻道上廣播。MU 需要對 server 提出需求，而是直接切換進入廣播頻道聽取所需的資料。On-demand access 是用基本的 client-server model，server 負責處理查詢並透過點對點的方式直接回傳結果給使用者。然而 On-demand access 適用在對於頻寬的 contention 與 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 radio and TV 工業的無線廣播，不管使用者的人數多寡少，對 server 的負載是一樣，server 還是只傳送一份資料，是解決 bandwidth problems 與 scalability 的方法。

除此之外，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 active mode 與 doze mode[8]兩種操作模式。典型的 wireless PC card 在 doze mode 消耗約 60mW 以

及在 active mode 消耗約 805-1400mW[2]。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。藉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 doze mode 直到所需的資料播放才進入 active mode 抓取資料。

2.3 On Air Index

傳統磁碟存取的環境，通常利用 back-tracking 來增進查詢的效能。然而這在只能線性存取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。因此，當演算法抓取索引資料的順序與它們被廣播的順序相反，則使用者必需等到該索引資料下一次的播放。相反的，在傳統的資料庫，索引資料儲存在如 disk 或 memory 的常駐儲存體，可以在任何的時間抓取。因為線性的存取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。例如圖 2 R-tree index 的廣播順序是 root, R1, R2。給一個查詢點 q_2 ，則拜訪的順序(visit sequence)是先 root，然後 R2，最後 R1。因廣播順序與查詢的順序不同將如圖 3(a) 所示導致很長的 access latency。因此，branch-and-bound 的 search approach 對 access latency 而言不是很有效率。另外一個選擇是，我們可以如圖 3(b) 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 index search performance。例如，針對查詢點 q_1 ，抓取 R1 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。

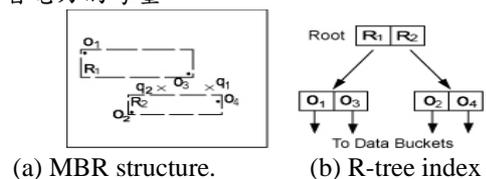


圖 2. A running example of R-tree index

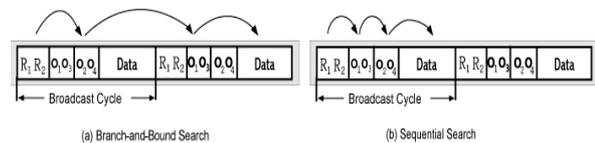


圖 3. Linear access on a wireless broadcast channel.

3. Design Rules for On Air Index

Access latency 與 Tuning time 是兩個用來衡量廣播索引效能的標準。Access latency 是使用者下達查詢直到抓取到所有滿足查詢所需的資料的這段時間。Tuning time 則是使用者以 active mode 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，MU 抓取資料將包括下述 3 個步驟[24]。(1)**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 initial probe waiting。(2)**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) **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儘可能降低這 3 個步驟所花的時間。

■ **降低 initial probe waiting time**: initial probe waiting 是使用者等待索引資料出現的時間，在 broadcast cycle 複製多份索引，可增加索引出現的機率，可降低 initial probe waiting time，Imielinski 等 [8]利用 (1,m) interleaving 重複配置 m 份索引資料的方式來降低 initial probe waiting time。

■ **減少索引的資料量**: Index Search 的時間與整個索引資料的大小有關，索引資料愈小，則 index search time 將愈短，整個 broadcast cycle 將相對愈短，故平均的 access latency 也將愈小。如 [8]只複製 index-tree 上面 k 層，來降低整個索引資料的大小。[25]利用則 signature 技術縮小索引資料。

■ **Efficient data placement 降低 Data retrieval 的時間**: Chen [26]證明不同資料不同的播放順序將影響平均的 access latency，提出 ORD algorithm 來降低平均的 access latency。而 [27]依 access frequency 安排 broadcast data 的先後次序，來降低 access latency。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 個步驟的效能改進設計一個新的廣播索引來處理廣播的 RNN Search。

4. A New Index for On Air RNN Search.

某一 query q 的 RNN 是那些以 query q 為 NN 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 的距離還近，則這些物件即為 query q 的 RNN。Rdnn-tree 與 R-tree 不同之處在於，它另外儲存了有關各個物件的 NN 資訊，可以直接判斷 leaf node 是否為查詢的結果，不像 R-tree 無法直接判斷 leaf node 是否為查詢的結果必須利用 branch-and-bound 的技術而有 back-tracking 的問題，故我們可將 Rdnn-Tree 依據設計有效的 On Air Index 的規則進一步的改良。

4.1 Rdnn-Tree

R-Tree[7]早期是發展在空間資料庫領域下的索引，被 [11]改良成 Rdnn-Tree，用來加速 RNN Search 的處理。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 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 rectangle 將這一群的物體包含住，此 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。Rdnn-Tree 的內部節點存放的是 MBR，只要是在它底下的節點都會被它包含住。Rdnn-Tree 的 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_b) 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 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 non-leaf node 的部分，則存放 (ptr, rect, MaxDnn)，其中 ptr 向 child node 的位址，rect 則是包含此 node 以下所有 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。

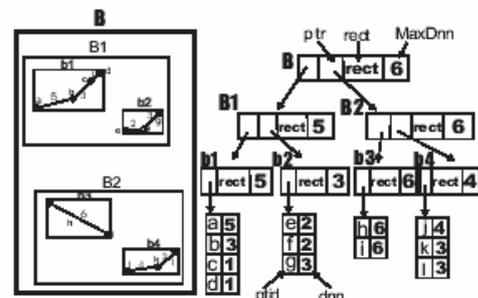


圖 4. Data structure of Rdnn-Tree.

4.2 Jump-Rdnn Tree (Jump Rdnn-tree)

設計一個良好的 On Air Index 如第 3 節所述包括減少 initial probe waiting、減少索引資料量以及 efficient data placement 三個階段。底下說明我們如何在這三個階段來改良 Rdnn-Tree。

■ **減少 initial probe waiting**: 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 broadcast cycle 變長，導致平均的 access latency 增長。我們的作法是，以每一 sub-tree 為單位將資料與索引混合播出，該 sub-tree 的索引播完緊接著播該 sub-tree 下面的資料，拉近資料與索引的距離，不是等整個 index tree 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 cyclic index structure，每一 index node 根據 DFS traversal 的順序算出下個要拜訪的 index node 的位置，即是所謂的 jump pointer 如圖 5 所示，因為每一 index node 皆有 jump pointer，index tree 透過 jump pointer 相互連結形成 cycle，故 index search 可從任何 index node 開始，不用從 root 開始搜尋，同時 index search 依 jump pointer 循序搜尋不會有 back-tracking 的問題。

■ **減少索引資料量**: 如前面所述，我們採用 cyclic index structure 故整個 broadcast cycle 不需複製任何

索引，只需一份索引資料，故索引資料量很小。另外，index tree 的架構，每一 index node 若有 fan-out 個 sub-trees 則需 fan-out 個 pointers 紀錄各個 sub-trees 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，就是 next 與 jump，不符合 jump 的條件，就是 next，故 next 不用特別紀錄，任何 search tree 的 non-leaf node 只要保留一個 jump pointer 即可，原來的 sub-tree 的 pointer 皆可去除，降低索引資料量。

■ Efficient data placement：query point 是根據整個 search space 任意產生，每區域發生 query 的機會將影響到查尋的效能，若 query 點發生愈多的區域對應的索引資料先播出，則整個 index search 的效能將愈好。因此我們根據每個 sub-tree MBR 所佔面積安排 sub-tree 的播放順序，因為 MBR 所佔的面積愈大，代表此區域會發生 query 的機率愈高。

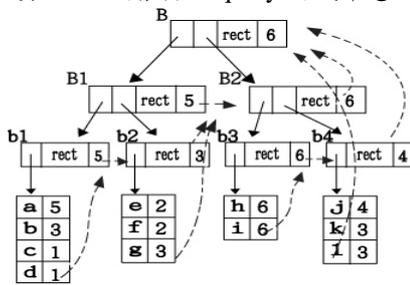


圖 5. Jump Rdn-Tree

Rdn-Tree 經由上述改善，除了不再有 back-tracking 的問題，cyclic 的 Rdn index tree 適合廣播的線性存取。至於 RNN Search on Air 詳細演算法如圖 6 所示，其中 $D(q, ptid)$ 代表 query q 到物件 $ptid$ 的距離， $D(q, rect)$ 則代表 query q 到 MRB $rect$ 的距離。

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

Case 1: 若 n 是 leaf node，判斷在 n 裡的每一個 (data-item, dnn) 若 $D(q, ptid) < dnn$ 表示 data-item 是 q 的 RNN 點

Case 2: 若 n 是 non-leaf node，判斷 n 的每一個 branch $B = (ptr, MBR, Maxdnn)$ ，若 $D(q, rect) < Maxdnn$ call RNN-Search-On-Air($B.ptr, q$)

圖 6. Algorithm for RNN Search on Air .

5. Performance Study

5.1. Experiment Environment

5.1.1 Two Data Set

我們使用兩個不同的 datasets 來進行實驗的評估。第一個 dataset (UNIFORM)，我們在 square Euclidean space uniformly 產生 1,000 個點。在第二個 dataset (SKEW)，我們利用 Zipf distribution 產生 1,000 個點，Zipf distribution 的 skew 參數設為 1.2。

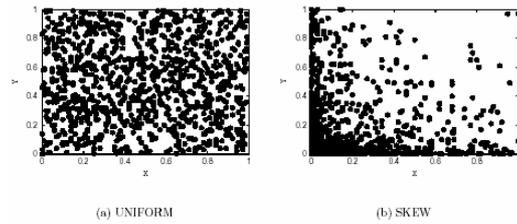


圖 7. Uniform and Skew data sets.

5.1.2 Compared Algorithms

無線廣播線性存取的特性，傳統 Rdn tree 的 node 依 DFS sequentially 的方式存取，搜尋過程不符合 distance heuristics search 條件的 sub-tree 將被 pruned 掉。為了降低 initial probe waiting 此處 Rdn-Tree 採取 (1:m) Interleaving Technology。

5.1.3 Performance metrics

一般測量 on air index 效能的 metrics 是 access time 與 tuning time，測量的單位是 packet。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電力的消耗。雖然 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，故實驗增加消耗總電力的 metric。消耗總電力

$P = 1200 * \text{Time}_{\text{active}} + 60 * \text{Time}_{\text{doze mode}}$ 。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 broadcast packet 所需的電力，60mW 為等待一 broadcast packet 所需的電力。

5.1.4 Parameters Setting

實驗參數設定如表 1 所示，Packet size 變動從 64 bytes 到 1024 bytes。Jump-Rdn Tree 的 Fan out 設為 6。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 initial probe 為 1 到 5000 個 broadcast unit 範圍中 random 產生，最後的統計結果為 30 queries 的平均值 [16]，實驗所用的程式是利用 [6] 的 R-tree 程式碼加以改寫。

表 1. 實驗參數設定

Parameter	Description	Setting
Object#	The number of data item.	Default:1000 Varying:1000~5000
Packet Size	The capacity of one broadcast unit.	Default:256 bytes Varying:64~1024
Fan out	The fan out of Rdn tree	Default:6

5.2 Performance Results

5.2.1 The effect of packet size

圖 8、圖 9 是不同資料分佈的實驗結果，就 access time 而言，不管 data 是 uniform 或 skew 分佈，隨著 packet size 的變大，而有較小 access time，這是

因為大的 packet capacity，可以容納更多的 data，因此 broadcast program 的 broadcast cycle 將相對較短所致。除此之外，我們的方法明顯優於 Rdn-tree (1:m) 的方法，我們的方法索引只播一次，相較於 Rdn-tree 採用 (1:m)，將索引播 m 次的作法，broadcast cycle 較短，有較小的 access time。就 tuning time 而言，隨著 packet capacity 的遞增，average packet access 遞減。但當 data 是 skew 且 packet capacity 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 order 分成幾個區段散落在整個 broadcast program，當 packet capacity 足夠大時可能有很多的索引區段不滿一個 packet，但因為 broadcast program 必須區分 index packet 與 data packet，這些不滿一個 packet 的 index 區段仍然必須佔一個 packet，這情況在 data 是 skew 時情形較嚴重，故當 packet capacity 變大時，整個平均的 packet access 遞減，average packet access 的值較小時，因為比例的關係其效應較明顯。至於總耗電力，不管 data 的分佈情形，我們的方法皆明顯優於 Rdn-tree (1:m) 的方法。雖然當 data 是 skew，packet capacity 大於 1024 bytes 時我們的方法 tuning time 稍為大於 Rdn-tree (1:m) 的方法，但比較總耗電力我們的方法仍優於 Rdn-tree (1:m) 的方法，這符合我們的想法，當考量 on air index 的效能必需考慮總耗電力，因為這才能真正反應行動裝置的電力消耗。

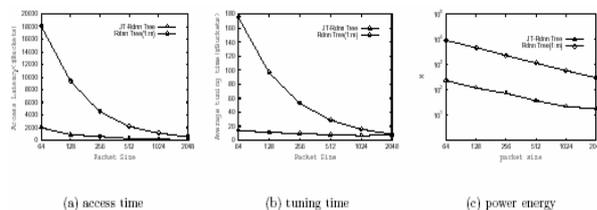


圖 8. Packet size affects for uniform distribution

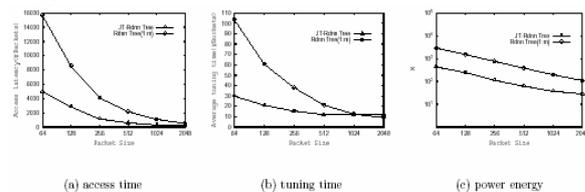


圖 9. Packet size affects for skew distribution

5.2.2 The effect of data objects

隨 objects 個數的增加，導致 broadcast program 增長故 access time、tuning time 以及總耗電力均因此增加。就 access time 而言，Rdn-tree (1:m) 因為整個 broadcast cycle 變長，平均等待 index packet 播出的時間增長，且 index 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 selective tuning 的資料擷取必需分 index packet 與 data packet 所致。至於總耗電力則不管 objects 個數

為何，我們的方法仍明顯優於 Rdn-tree (1:m)。

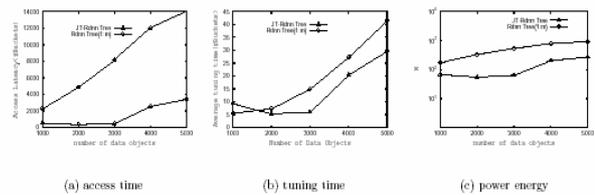


圖 10. Data objects affects for uniform distribution

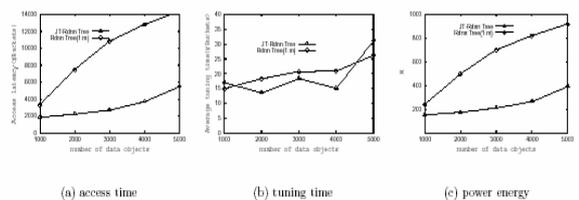


圖 11. Data objects affects for skew distribution

6. Conclusions and Future works

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 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 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-Rdn 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 Rdn-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。

7. Reference

- [1] Daniel Barbara, Mobile Computing And Databases - A Survey, IEEE Transactions on Knowledge and Data Engineering, Vol. 11, No. 1, February 1999, p.p. 108-117.
- [2] 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.
- [3] Rimantas Benetis, Christian S. Jensen, Gytis Karciauskas and Simonas 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.
- [4] Surajit Chaudhuri and Luis Gravano, Evaluating top-k selection queries, Proceedings of the 25th IEEE International Conference on Very Large Data Bases, 1999, p.p. 297-410.

- [5] Computer Science and Telecommunication Board. IT Roadmap to a Geospatial Future, the National Academies Press, 2003.
- [6] Gist , available on line , <http://gist.cs.berkeley.edu/>
- [7] Antonin 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.
- [8] T. Imielinski and 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.
- [9] Glenn Iwerks , Hanan Samet and Ken Smith , Continuous K-Nearest Neighbor Queries for Continuously Moving Points with Updates , International Conference on Very Large Data Bases , 2003 , p.p. 512-523.
- [10] George Kollios , Dimitrios Gunopulos and Vassilis J. Tsotras , Nearest Neighbor Queries in a Mobile Environment Spatial-Temporal Database Management , 1999 , p.p. 119-134.
- [11] Flip 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.
- [12] Dik Lun Lee , Wang-chien Lee , Jianliang Xu and Baihua Zheng , Data Management in location dependent services , IEEE Pervasive Computing , Vol. 1 , No. 3 , September 2002 , p.p. 65-72.
- [13] Nick Roussopoulos , Stephen Kelley , Fr'ed'eric Vincent , Nearest neighbor queries , Proceedings of ACM SIGMOD International Conference on Management of Data , June 1995 , p.p. 71-79.
- [14] Sheldon Ross , Introduction to Probability and Statistics for Engineers and Scientists , 2000.
- [15] Ioana Stanoi , Divyakant Agrawal , Amr El 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.
- [16] Amit Singh , Hakan Ferhatosmanoglu and Ali Saman 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.
- [17] Zhexuan 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.
- [18] Yufei Tao and Dimitris Papadias and Qiongmao Shen, "Continuous Nearest Neighbor Search", International Conference on Very Large Data Bases,Hong Kong China,August 20-23,p.p. 279-290.
- [19] Yufei Tao and Dimitris Papadias and Xiang 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.
- [20] Jianliang Xu and Wang-Chien Lee and Xueyan 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.
- [21] Jianliang Xu and Baihua Zheng and Wang-Chien Lee and and Dik Lun 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.
- [22] Congjun Yang and King-Ip Lin, "An index structure for efficient reverse nearest neighbor queries", Proceedings of the 17th International Conference on Data Engineering,2001,p.p. 485-492.
- [23] Baihua Zheng and Wang-Chien Lee and Dik Lun 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.
- [24] Jun Zhang and Manli Zhu and Dimitris Papadias and Yufei Tao and Dik Lun 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.
- [25] Ming-Syan Chen and Philip S. Yu and Kun Lung 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.
- [26] Jianting Zheng and Le Gruenwald , Prioritized Sequencing for Efficient Query on Broadcast Geographical Information in Mobile Computing, ACM GIS , 2002,p.p.88-93.
- [27] Quinglong Hu and Wang-Chien Lee and Dik Lun Lee , "Index Techniques for Power Management in Multi-Attribute Data Broadcast" , Mobile Networks and Applications , Vo. 6,No. 2,2001,p.p.185-197.