

Maximum-Empty-Rectangle Spatial Index for Non-flat Broadcast

沈俊宏

亞洲大學資訊傳播系副教授
e-mail: shenj@asia.edu.tw

麥健棠

亞洲大學資訊傳播系研究生
e-mail: purelydinkum@gmail.com

陸清達

亞洲大學資訊傳播系教授
e-mail: lucas@asia.edu.tw

摘要

在無線通訊的環境中，非均勻廣播模式將熱門資料會比一般資料播放更多次，讓使用者能更快的查詢到熱門的資料。本文提出適用於非均勻廣播模式的 MER (Maximum Empty Rectangle) 空間索引方法，處理空間視窗查詢或連續視窗查詢。我們在熱門空間資料加入了索引資訊，其為一矩形包含了此區的資料並且往外延伸至附近的資料為止。這代表視窗查詢範圍若在此矩形之內，查詢只包含熱門資料，不必再查詢一般資料，因此可以減少查詢等待時間。我們驗證所提出的方法優於 SSI 索引方法。

關鍵詞：位置相關服務、無線數據廣播系統、非均勻廣播模式、視窗查詢。

Abstract

In the wireless communication environments, hot data are broadcast more times than regular data in the non-flat broadcast, such that mobile users can retrieve the hot data quickly. In this paper, we propose an MER (Maximum Empty Rectangle) spatial index methods to handle spatial window queries and continuous window queries in the non-flat broadcast. We interleave index information with hot spatial data. The index information provides a rectangle which contains hot data and extends its region maximally without touching the other data. If the region of a window query is within this rectangle and contains only hot data, no other data should be examined, resulting in the decrease of the query waiting time. We have verified that our proposed method outperforms the SSI method.

Keywords: location-based service, wireless data broadcast, non-flat broadcast, window queries.

1. 前言

隨著智慧型移動設備和無線網路的普及，如智慧型手機和平板電腦，移動設備能做到的事情越來越多 [8][14]。不少移動設備配有感知位置的功能，例如 GPS，能夠讓使用者在需要的時候得知他們目前的地理位置，並且根據目前的位置提供各種使用者需要的服務，稱為位置相關服務(LBSs) [10][11][14]。空間物件查詢為位置相關服務常見的應用，例如：使用者需要找到附近的加油站、提款機等等。

常見的空間物件查詢方法有四種：近鄰查詢 (k NN queries)、視窗查詢 (window queries)、連續近鄰查詢 (continuous NN queries)、連續視窗查詢 (continuous window queries)。近鄰查詢表示查詢 k 個最接近使用者的資料，例如：使用者查詢最接近的 3 個加油站 ($k=3$)。視窗查詢為查詢一範圍內的所有資料，例如：使用者查詢附近一英里內所有的加油站。而連續近鄰查詢和連續視窗查詢表示預估使用者在一移動時間內的所有近鄰查詢和視窗查詢，例如：使用者查詢由家裡到公司的路上每個路段最近的加油站。本文注重於解決視窗查詢和連續視窗查詢。

在我們的日常生活中，大量使用位置相關服務。常見的方法是向伺服器請求所需要的資料，再從伺服器下載下來。但每個使用者進行這行為時，都必須消耗伺服器的流量，如果大量使用者同時向伺服器要求資料，伺服器可能會不堪負荷。為了讓大量的使用者皆能存取伺服器的資料，無線數據廣播系統可以有效的解決這個問題 [3]。無線數據廣播系統是由伺服器在一個或多個頻道不斷的循環播放資料，而每個使用者擷取頻道中各自所需要的資料。這樣無論使用者有多少，伺服器皆只耗費固定的流量，有效地解決了使用者數量的問題。

另外，為了在無線廣播系統中節省移動設備的耗能，以及能讓使用者擷取到各自所需的資料，無線數據廣播系統中使用了索引結構 [4][15]。在廣播的資料前插入索引，而索引中記載了各個資料播放的時間點或其它索引播

放的時間點，使用者在不是自己所需要資料的播放時間，可以將設備切換成省電模式(doze mode) 減少耗能，而在需要存取資料的時間切換成運作模式(active mode)。如此就不需要存取廣播中所有播放的資料，僅存取使用者想要的資料，大幅減少了行動設備的耗能。

而無線廣播系統的效能由存取時間(access time) 和聽頻道時間(tuning time) 來評估 [2]。存取時間指的是由使用者向伺服器請求查詢後直到得到結果的時間，表示此無線廣播系統查詢速度的快慢；聽頻道時間則是在存取時間內設備停留在運作模式的時間，表示此無線廣播系統耗能的多寡。

在使用無線廣播系統處理空間資料查詢的文獻中，分佈式空間索引 DSI [14] 的索引以指數的方式支援了近鄰查詢、視窗查詢、連續近鄰查詢以及連續視窗查詢。NI [13] 以階層式區域的概念加入索引，改善了 DSI 索引以指數方式的視窗查詢以及連續視窗查詢的效能。DSPI [7] 利用以資料圍成的空間區塊，改善了 DSI 使用空間填充曲線，在視窗查詢時容易進行多餘查詢的問題。

而以上的方法皆是均勻廣播模式(flat broadcast)，指的是每個空間資料播放的頻率相同，在一個播放循環只出現一次。而 GDIN [6] 考慮了空間資料播放頻率不相同的方法，稱非均勻廣播模式(non-flat broadcast)。非均勻廣播模式會因為空間資料熱門程度，而讓每個空間資料播放的頻率不相同。在 MLAIN [5] 中同樣使用非均勻廣播模式改善了 GDIN 空間資料區塊劃分的問題。在 GDIN 和 MLAIN 中使用非均勻廣播模式將空間資料資料分為 2 類：熱門資料(hot data) 和一般資料(regular data)，而 SSI [12] 使用了 Acharya 等人 [1] 提出的廣播磁盤(Broadcast disks)，將空間物件根據使用者存取頻率分為多類，更加改善了效能。

在本文中，我們提出了一個改良 SSI 的 MER (Maximum Empty Rectangle Containing Query Points) 空間索引方法，使用非均勻模式的無線廣播系統於空間資料查詢，並且支援視窗查詢和連續視窗查詢。MER 空間索引方法主要在解決非均勻廣播模式中視窗查詢或連續視窗查詢的範圍容易包含熱門資料區塊和一般資料區塊的問題。在 SSI 方法中，因為使用空間填充曲線劃分空間資料，導致每個資料對應的空間區塊範圍都相同。這使得使用者在進行視窗查詢或連續視窗查詢時，容易造成查詢視窗雖然包含了空間資料所屬區塊，但卻不包含空間資料，使得查詢處理程序必須做多餘的

檢查。舉例來說，若查詢視窗包含了熱門資料區塊和一般資料區塊，但查詢視窗在一般資料區塊內不包含一般資料，則查詢還是必須確認一般資料區塊內有無資料。事實上，這是個多餘的查詢，因為查詢視窗在一般資料區塊內無任何一般資料。再者，一般資料區塊因為非均勻廣播模式播放頻率較低，要查詢到一般資料區塊需要較多的時間，整體下來大幅增長了存取時間。

在熱門資料區塊附近，上述的情形會大量的發生，因為空間填充曲線導致熱門資料區塊和一般資料區塊大小一樣的關係，使得查詢視窗很容易包含多個區塊。若每次查詢熱門資料區塊都還要確認一般資料區塊，則索引結構便失去非均勻廣播模式的優勢，效能大幅降低。

為了處理這個問題，我們在熱門資料區塊加上了 MER 索引資訊。MER 索引資訊為一矩形包含了此區的資料，並且往外延伸至附近的資料為止。這代表視窗查詢或連續視窗查詢的範圍若在此矩形之內，便可以確定查詢只包含熱門資料，不必再查詢一般資料。換句話說，是將熱門資料區塊有效範圍加大的概念。這概念可以套用在其他非均勻廣播模式的索引結構中，將熱門資料區塊有效範圍加大。實驗結果驗證本文提出的 MER 索引結構比 SSI 索引結構於存取時間(access time) 和聽頻道時間(tuning time) 皆更有效率。

本文接下來的安排如下：在第 2 節，我們介紹了本文會使用的無線廣播環境處理空間查詢的相關技術。第 3 節，介紹我們是出的 MER 索引結構的方法。第 4 節，我們呈現了實驗結果。最後，第 5 節為結論。

2. 相關技術

在無線廣播的環境中，資料是以一維的方式播放，播放順序為循序性。要將二維的地圖資料轉換成一維的播放順序，我們使用 Hilbert curve 空間填充曲線。在無線廣播的環境中，若每個資料播放的頻率相同，稱為均勻廣播模式。若要將想要的資料多播放幾次，造成每項資料播放頻率不同，稱為非均勻廣播模式。在非均勻廣播模式中，由於熱門資料在一個播放循環內(Broadcast cycle) 播放較多次，可以減少存取時間。如圖 1 (a)，每項資料在一個播放循環中皆播放一次，稱為均勻廣播模式。圖 1 (b)，資料 A 在播放循環中播放了 2 次，造成每個資料播放的頻率不同，稱為非均勻廣播模式。Acharya 等人 [1] 提出了廣播磁盤

(Broadcast disks) 能夠有效的處理非均勻廣播模式中資料如何排程。Hilbert curve 及 Broadcast disks 簡易的介紹如下:

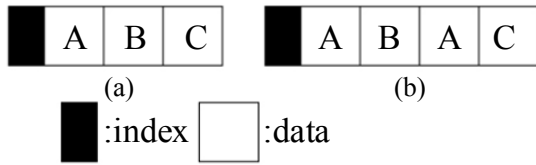


圖 1 播放模式：(a)均勻廣播模式(b)非均勻廣播模式。

2.1 Hilbert curve

Hilbert curve 空間填充曲線能將空間中的資料轉換為一維的廣播排列順序，並且能維持一定程度的空間區域性。Hilbert curve 是由一連續路徑通過多維空間，並將路徑上的點編號（以下簡稱 hc ），且能夠使用階層 n (order n) 調整空間中的曲線填充密集度，如圖 2 (a) 為 $n=1$ 的空間編碼方式。在二維空間中空間物件距離接近，則空間物件編碼後的 hc 值也會接近，如此便能將空間轉換為一維的排列順序。Hilbert curve 的階層 n 表示空間中每邊 2^n 個 hc ，且 hc 範圍為 0 到 $(2^{n*2}-1)$ 。如圖 2 (b) 所示，Hilbert curve 階層 $n=2$ 表示 hc 範圍由 0 到 15。

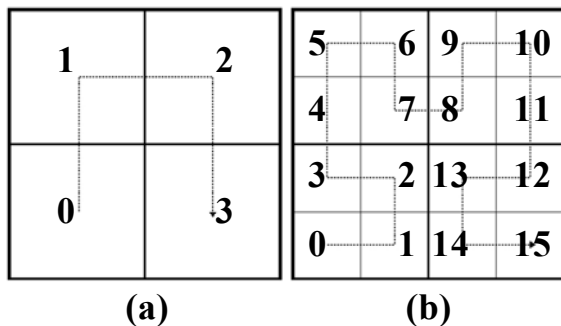


圖 2 Hilbert curve: (a) order 1; (a) order 2.

2.2 Broadcast disks

Acharya 等人的 Broadcast Disks 方法 [1] 將資料依存取頻率分成幾個類別，最熱門的類別包含最少的空間物件，但在同一廣播循環內出現的次數最多，較不熱門的類別包含較多的空間物件，但出現的次數較少。此方法能讓使用者決定資料需要分為幾區，以及每區的播放的頻率，最後得出我們需要的廣播排列順序。先決定磁盤的數量 S (D_1-D_S)，以及磁盤廣播頻率 λ_i ($1 \leq i \leq S$)，而每個磁盤在廣播上的播放頻率最高由 D_1 至 D_S 。接著，每個資料依序放入磁盤中，每個磁盤中根據 $(\lambda_1-\lambda_S)$ 的最小公倍數 L 再將資料分區。最後分為 L 個小播放循環

(B_1-B_L)，將資料按照 λ_i 依序放入 (B_1-B_L) 成為一個完整的播放循環。本文應用此方法將空間物件配置為非均勻式廣播。

3. MER (Maximum Empty Rectangle) 空間索引方法

在 MER 空間索引環境中，伺服器會處理一個二維區域中 N 個空間物件 $\{d_1, \dots, d_N\}$ ，轉換成一維的無線廣播索引結構，並交給基地台。基地台會分為 2 個頻道，其中較大頻寬的頻道 (High-bandwidth) 負責不斷重覆發送索引結構與空間物件，讓使用者存取各自所需的資料；而較小頻寬的頻道 (Low-bandwidth) 負責接收使用者回饋查詢了哪些資料。而伺服器接收到使用者的回饋，便可以排序出每筆空間物件的查詢頻率，並且定時調整索引結構，再經由基地台發送。如圖 3 所示。

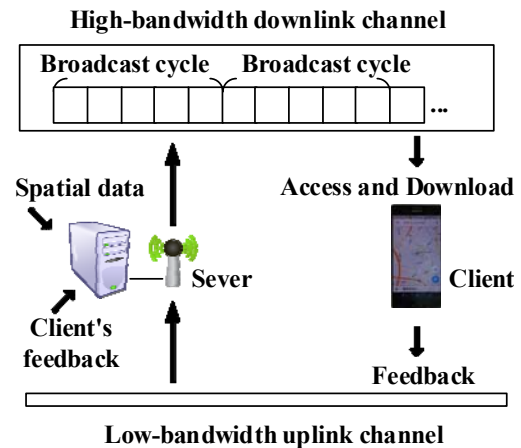


圖 3 無線廣播模式介紹

MER 空間索引方法根據 SSI [12] 為基礎來改良，主要分為 4 步驟，並在接下來各的小節詳述：

1. 將二維空間資料轉為一維排序。根據 Hilbert Curve 劃分地圖為區塊 (cells)，並配置內含空間物件的區塊一個 hc 值。
2. 將資料根據 Acharya 等人 [1] 的磁盤方式產生非均勻式廣播，並同時於廣播資料內配置索引資訊空間。
3. 加入索引內容，在索引上加上索引項目。
4. 查詢處理機制，包含視窗查詢和連續視窗查詢。

3.1 資料劃分

在無線廣播的環境中，無線資料廣播的播放順序為循序性，為了將二維空間中的空間物件轉換為一維的廣播排列順序，並保有空間的

區域性，我們採用 Hilbert curve 將二維空間物件轉換為一維播放順序。首先，將空間中的每筆資料劃分後，對應到 Hilbert curve 的 hc 值 (以下簡稱 hc) 來排序。以 Hilbert curve 的 n 階層 (以下簡稱 n) 來將所有資料遞迴分為 $2^n \times 2^n$ 區塊 (cell)，每個區塊中包含上限 η 個資料，即所有資料及區塊都有對應 hc 及 n 。

以圖 4 (a) 為例，地圖中有 13 家餐廳的資料 $\{d_1, \dots, d_{13}\}$ ，並且經過回饋得知每筆資料查詢頻率的排序，如： d_2 和 d_3 為查詢頻率最高， d_8 和 d_{12} 次高。接著，假設 $\eta = 2$ ，代表分割空間後每一區塊最多只能包含 2 個空間物件。圖 4 (b) 中每點代表資料 d_1-d_{13} ，這時因為範圍中包含 13 個資料，將區塊分成 $\{C(1,0), C(1,1), C(1,2), C(1,3)\}$ 。其中 $C(i,j)$ ， i 代表 Hilbert Curve 的階層， j 代表在該階層 i 中此區的編碼 hc 值。在 Hilber Curve 第 1 階層的劃分空間後，這時 $C(1,0)$ 有 4 個資料，大於上限 $\eta = 2$ ，再將區塊分成 4 塊 $\{C(2,0), C(2,1), C(2,2), C(2,3)\}$ ，此 4 區內資料皆符合上限 $\eta = 2$ ，不用再劃分。而其他區塊 $\{C(1,1), C(1,2), C(1,3)\}$ 也照此方法處理，直到符合上限 $\eta = 2$ 。此空間物件經由 Hilbert Curve 第 2 階層劃分完成後如圖 4(c) $\{C(2,1), \dots, C(2,14)\}$ 所示。由於經過回饋得知每筆資料查詢頻率的排序，接下來將想要多次播放的熱門的區塊選出，如圖 4 (c) 內的 $\{C(2,2), C(2,8), C(2,13)\}$ ，命名為 $\{H(2,2), H(2,8), H(2,13)\}$ 。

3.2 排程空間資料

在前一節二維空間已劃分為區塊 $\{C(2,1), H(2,2), \dots, C(2,14)\}$ ，接下來將這些區塊以非均勻式廣播模式配置無線頻道上的播放順序，並同時配置好索引資訊空間，主要分為 4 個步驟：

1. 採用 Acharya 等人 [1] 的廣播磁盤 (Broadcast disks) 產生在廣播週期中空間區塊資料的播放時間表。廣播磁盤以圖 5 (a) 為例：假設我們將熱門程度分為 3 個類別 ($S=3$)，分成 3 個磁盤 $\{D_1, D_2, D_3\}$ ， $\{D_1, D_2\}$ 為熱門磁盤。若磁盤播放頻率 $\lambda_1=4, \lambda_2=2, \lambda_3=1$ ，三個磁盤播放頻率最小公倍數 L 為 4。在 D_1 磁盤中，空間物件分為 1 單位 ($L/\lambda_1 = 4/4 = 1$)；在 D_2 磁盤中，資料分為 2 單位 ($L/\lambda_2 = 4/2 = 2$)；在 D_3 磁盤中，資料分為 4 單位 ($L/\lambda_3 = 4/1 = 4$)，見圖 5 (b)。每一播放循環內有四個 ($L=4$) 子循環 B_1-B_4 ，每一磁盤依序放一單位至子循環內，見圖 5 (c)。

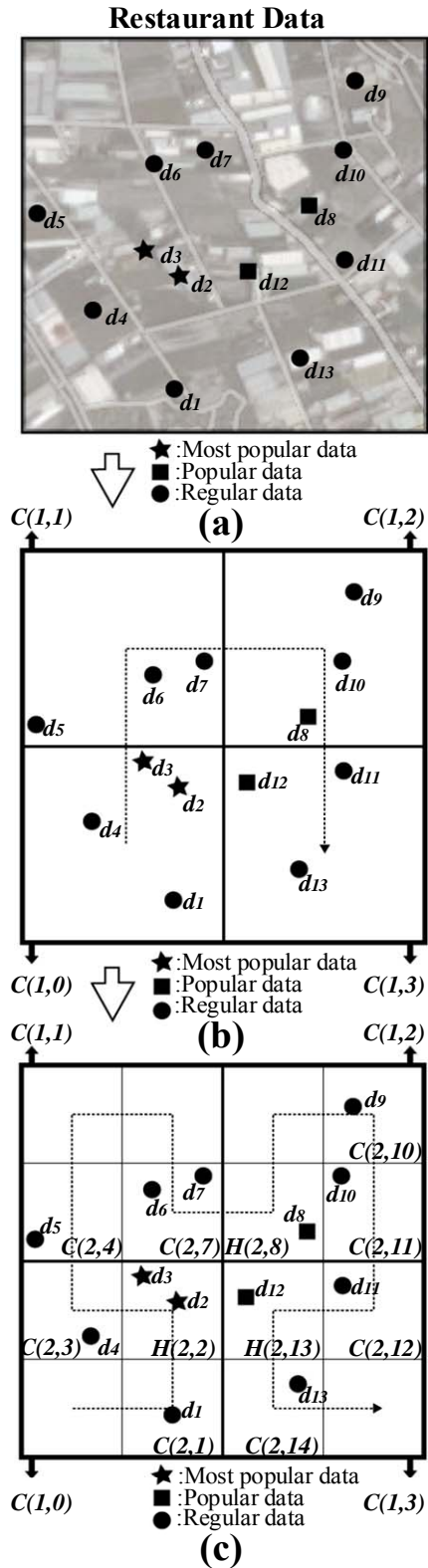


圖 4 資料劃分步驟：(a)地圖上擁有資料；(b)第一次劃分資料；(c)第二次劃分資料，並決定熱門區塊。

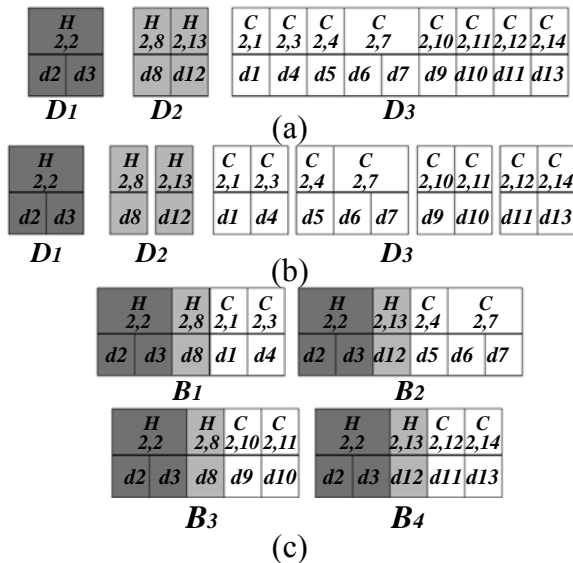


圖 5 排列資料: (a) 決定磁盤數量並排序資料; (b) 按照播放頻率將磁盤分區; (c) 依序將資料排進(B₁-B₄)。

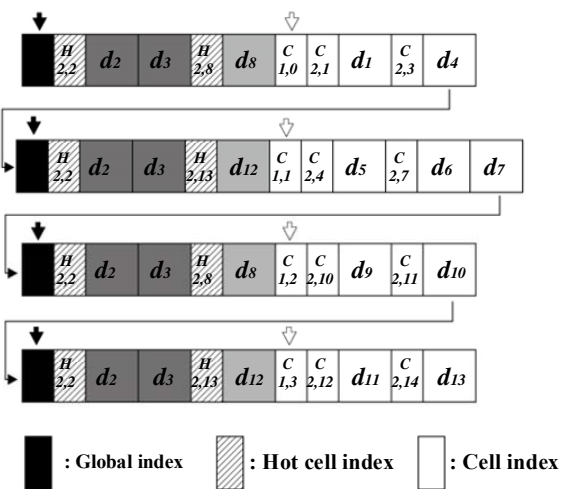


圖 6 根據排好的資料放入索引

2. 穿插全域資料索引 (Global index) 在每個小播放循環之前, 如圖 6 黑箭頭所示。
3. 在熱門磁盤中根據 hc 值以及 Hilbert Curve 階層 n 插入熱門區塊索引 (Hot cell index), 如斜線索引所示。
4. 在剩餘磁盤中根據 hc 值及 Hilbert Curve 階層 n 插入區塊索引 (Cell index), 如白色索引所示。並且在每個區塊 hc 值為下一階層第一個出現的資料, 根據 hc 值以及 Hilbert curve 階層 1 至 Hilbert curve 階層 $(n-1)$ 插入資料索引 (Cell index)。例如圖 6 白色箭頭所示, $\{C(2,0), C(2,1), C(2,2), C(2,3)\}$ 為同一區, 必須放入 $C(1,0)$, 而 $C(2,1)$ 為此區第一個出現有資

料的索引, 故在 $C(2,1)$ 前插入 $C(1,0)$ 。

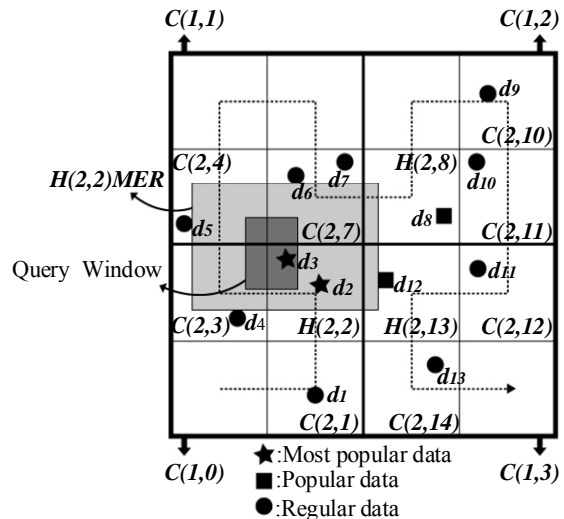


圖 7 地圖資料以及查詢視窗

3.3 索引結構

在設計非均勻式資料廣播的索引結構時, 因為熱門區塊會被多次播放, 若查詢範圍盡量落在熱門區塊上, 可以減少存取時間。但因為 Hilbert Curve 要維持空間排序, 導致熱門區塊和一般區塊的大小是一樣的, 進而造成查詢範圍非常容易包含熱門區塊和一般區塊。若是查詢範圍太容易包含一般區塊, 會使查詢時間大幅增加, 造成非均勻式資料廣播之優勢會大幅下降。

以圖 7 為例, 圖中查詢視窗包含熱門查詢空間物件 d_3 , 查詢範圍與 $\{H(2,2), C(2,3), C(2,4), C(2,7)\}$ 這些區塊重疊, 除了熱門區塊 $H(2,2)$ 之外, $C(2,3), C(2,4), C(2,7)$ 皆為一般區塊。查詢處理程序會先至 $H(2,2)$ 確認 d_3 資料, 再至 $C(2,3), C(2,4), C(2,7)$ 確認是否有資料存在, 查詢處理程序才會結束。事實上, 查詢處理程序在 $H(2,2)$ 裡的空間物件 d_3 確認完便可結束, 但因為查詢範圍包含了其他區塊, 處理機制必須確認這些範圍內沒有含有資料才可結束, 導致存取時間與聽頻道時間增加。

為了解決上述問題, 我們提出了 MER 空間索引方法, 這可以套用在任何非均勻廣播模式的索引結構中。在本文中, 我們使用區塊內空間物件的 Maximum Empty Rectangle (MER) [9] 來解決上述的問題, MER 為含有這些空間物件往四周延伸的最大無空間物件的矩形範圍。

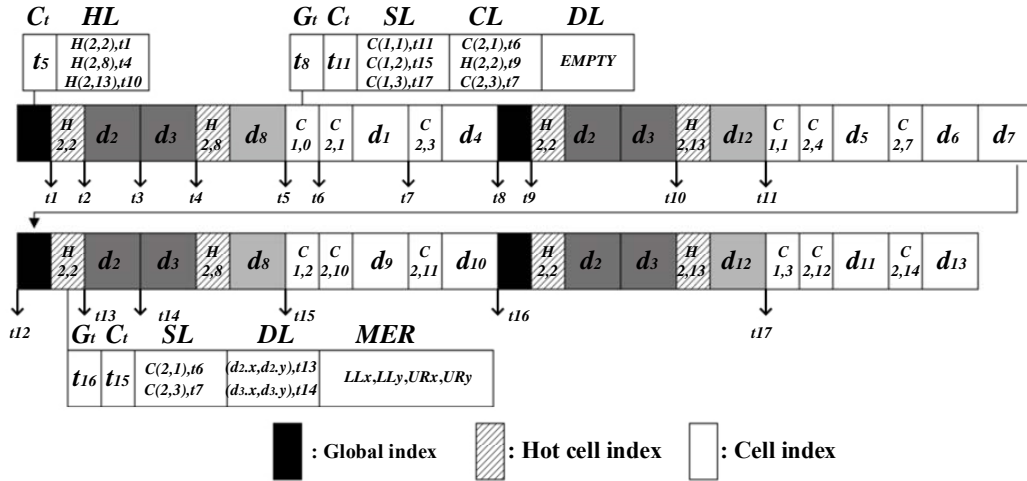


圖 8 MER 空間索引

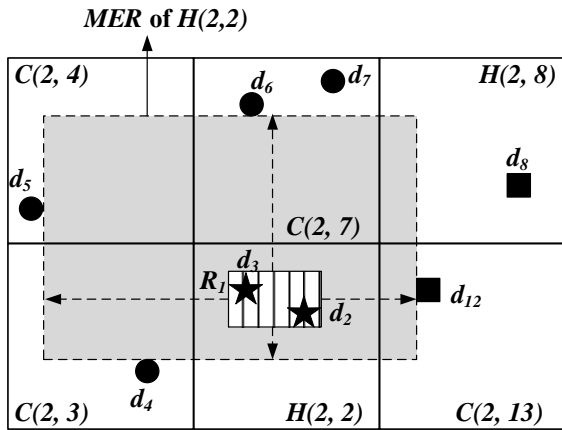


圖 9 Maximum Empty Rectangle

舉圖 9 為例， R_1 為包含熱門區域 $H(2, 2)$ 內的空間物件 d_2 與 d_3 的最小範圍矩形 (minimum bounding rectangle)， R_1 往四周延伸最大無空間物件的矩形即為熱門區域 $H(2, 2)$ 的 MER。若查詢範圍與 $H(2, 2)$ 重疊且 MER 包含此查詢範圍，則在檢查完 $H(2, 2)$ 後即可完成查詢處理程序，不須再檢查其他區塊。如此，可以減少存取時間與聽頻道時間，降低行動設備電力的消耗。

我們所提出 MER 空間索引方法有 3 種索引結構，如圖 8 所示。

1. Global index: $G = \langle Ct, HL \rangle$
 - Ct (next top-level cell time):
 Ct 為下個最高階層 $C(i, j)$ 的出現時間。
 - HL (hot-cell index list): $HL = \langle H(i, j), t \rangle$ 為所有 Hot cell index 出現的時間 t 。
2. Cell index: $C(i, j) = \langle Gt, Ct, SL, CL, DL \rangle$
 - Gt (next global index time):
 Gt 為下個 Global index 出現時間。
 - Ct (next top-level cell time): Ct 為下個最高

階層 $C(i, j)$ 及出現時間。

- SL (sibling list): $SL = \langle C(i, j), t \rangle$ 為相鄰同階層出現 $C(i, j)$ 與到達時間 t ，資料最多為 3 筆。
 - CL (child list): $CL = \langle C(i, j), t \rangle$ 為此區下一階層出現 $C(i, j)$ 與到達時間 t ，資料最多為 4 筆。
 - DL (data list): $DL = \langle (dx, dy), t \rangle$ 為此區出現資料的座標與到達時間，資料最多為 η 筆。
3. Hot cell index:
 $H(i, j) = \langle Gt, Ct, SL, DL, MER \rangle$
- $\langle Gt, Ct, SL, DL \rangle$ 如前 Cell index 所述。
 - $MER = \langle LLx, LLy, URx, URy \rangle$ ，為包含此區資料最大矩形左下及右上座標。

以圖 7 為例，圖中查詢視窗包含熱門查詢點 d_3 ，並涵蓋了區塊 $\{H(2,2), C(2,3), C(2,4), C(2,7)\}$ 。在圖 8 中，假設行動設備聽頻道擷取第一個 Global index，此索引內的 HL 中有 $H(2,2)$ 的播放時間 t_1 ，接著行動設備會擷取 $H(2,2)$ 索引。經由此索引內的 MER 資訊，行動設備可判別查詢視窗在此 MER 範圍之內，不需再去檢查其他區的資料，可直接結束查詢處理程序，此次查詢結果為空間物件 d_3 。處理過程省去了搜尋 $C(2,3), C(2,4), C(2,7)$ 索引，大幅縮短了存取時間，以及不必要的聽頻道時間。

3.4 查詢處理機制

MER 空間索引方法可以支援視窗查詢與連續視窗查詢。視窗查詢為查詢一範圍內的空間資料。視窗查詢會先將與查詢視窗範圍重疊的區塊加入待檢查的集合 SC 中，並且依照索引順序檢查所有集合 SC 裡的區塊以及區塊內

的資料。詳細方法如下：

1. 將與查詢視窗範圍重疊的區塊加入待檢查的集合 SC 中。
2. 使用者等待索引並根據索引項目及待檢查的集合 SC 決定系統進入運作模式 (active mode) 或省電模式 (doze mode)。系統會依照階層劃分，查詢需要的索引，例如：需要查詢 $C(2,7)$ ，系統會經由確認熱門索引有無查詢區塊、查詢至同區階層、最後至同區階層下找到區塊。此例的查詢順序為 Global index、 $C(1,0)$ 、 $C(1,1)$ 、 $C(2,7)$ 。
3. 將索引要進入運作模式 (active mode) 的時間，也就是要查詢的時間依序加入 ST 中。
4. 依照 ST 查詢 Hot cell index，若 MER 索引項目涵蓋查詢範圍，則結束查詢並且得到相應的結果。
5. 依照 ST 查詢 Cell index，根據 SC 不斷加入 ST ，直至確認所有 SC 中的區塊。

以圖 10 的視窗查詢 $Q1$ 為例， $Q1$ 與 $H(2,2)$ 、 $C(2,7)$ 重疊，於是加入待檢查的 SC 中，即 $SC = \{H(2,2), C(2,7)\}$ 。假設行動設備在圖 8 的廣播循環一開始即聽無線頻道，取得 Global index，其內的 HL 索引資訊有 $H(2,2)$ 與到達時間 $t1$ 。接著， $H(2,2)$ 被排入 ST ，即 $ST = \{(H(2,2), t1)\}$ 。在 $t1$ 擷取 $H(2,2)$ 後，檢查 DL 項目 d_2 及 d_3 ，發現只有 d_2 在查詢範圍內。處理程序於 ST 內刪除 $(H(2,2), t1)$ ，並加入 $(d_2, t2)$ ，即 $ST = \{(d_2, t2)\}$ 。以及於 SC 內刪除 $H(2,2)$ ，即 $SC = \{C(2,7)\}$ 。原本因 SC 內還有需要檢查的範圍，行動設備尚需要繼續聽頻道擷取資料，但在 $H(2,2)$ 內的 MER 索引資訊發現查詢視窗 $Q1$ 在其 MER 範圍內，所以不必再檢查 $C(2,7)$ ，查詢處理程序在取得空間物件 d_2 後結束。

連續視窗查詢為預估一路段時間內的視窗查詢。連續視窗查詢的查詢路徑必須知道每個時間點上有哪些資料，也就是資料在哪段查詢路徑上查詢的到。我們使用 DSI 裡應用的 Minkowski regions [14] 方法來標示查詢結果有效區段。Minkowski Region 為以查詢結果的空間物件為中心，區域範圍與查詢範圍同大小，接著其區域範圍與路徑重疊的區段即是該空間物件的有效區段。以圖 11 為例，每筆空間物件皆有範圍與路徑重疊，即可得知其有效區段。例如：空間物件 d_1 在區段 \overrightarrow{AC} 有效，但

一超過 C 點則失效。在 \overrightarrow{BC} 區段 d_1 和 d_2 皆有效。而在區段 \overrightarrow{DE} 因為沒有任何資料有效範圍，所以在此區段不會查詢到任何資料。

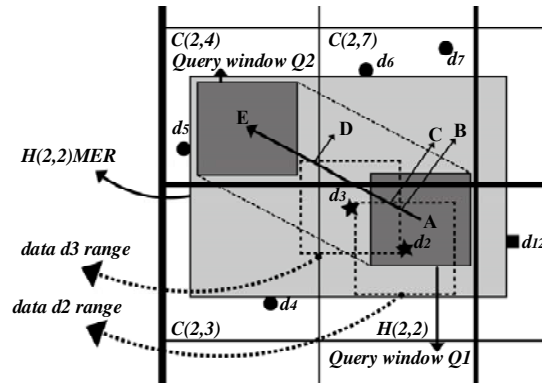


圖 10 擷取圖 4 地圖資料重點區域以及連續查詢視窗

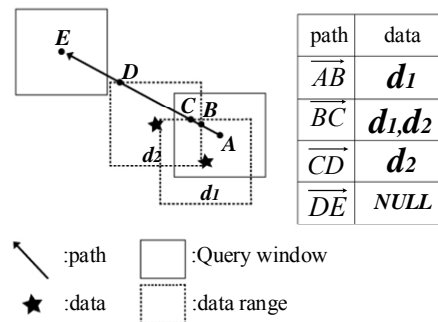


圖 11 Minkowski regions 介紹

連續視窗查詢和視窗查詢方法類似，最後使用 Minkowski regions [14] 指定空間物件有效區段。以圖 10 為例，連續查詢視窗 $Q1$ 至 $Q2$ 與 $H(2,2)$ 、 $C(2,3)$ 、 $C(2,4)$ 、 $C(2,7)$ 重疊，於是加入待檢查的 SC 中，即 $SC = \{H(2,2), C(2,3), C(2,4), C(2,7)\}$ 。假設行動設備於圖 8 內廣播循環一開始取得 Global index，其內的 HL 索引資訊有 $H(2,2)$ 與到達時間 $t1$ 。接著，將 $H(2,2)$ 加入 ST ，即 $ST = \{(H(2,2), t1)\}$ 。於 $t1$ 擷取 $H(2,2)$ 後檢查 DL 項目 d_2 及 d_3 ，發現 d_2 及 d_3 皆在查詢範圍內。於 ST 內刪除 $(H(2,2), t1)$ ，並加入 $t2$ 及 $t3$ ，即 $ST = \{(d_2, t2), (d_3, t3)\}$ 。以及於 SC 刪除 $H(2,2)$ ，即 $SC = \{C(2,3), C(2,4), C(2,7)\}$ 。原本因 SC 內還有需要檢查的範圍需要繼續執行查詢處理程序，但在 $H(2,2)$ 內的 MER 索引資訊發現查詢視窗 $Q1$ 至 $Q2$ 在其 MER 範圍內，所以不必再檢查 $C(2,3)$ 、 $C(2,4)$ 、 $C(2,7)$ ，查詢在取得空間物件 d_2 及 d_3 後結束。最後使用 Minkowski regions [14] 分配其有效區段， d_2 在區段 \overrightarrow{AC} 時有效， d_3 在區段 \overrightarrow{BD} 時有效。

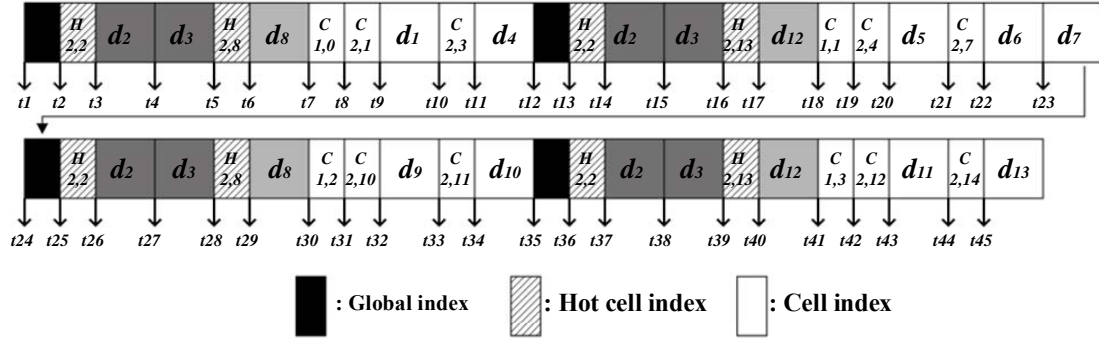


圖 12 播放結構

4. 效能分析

在 SSI 方法 [12] 中已與 MLAIN 方法 [5] 進行效能分析比較，其結果顯示 SSI 優於 MLAIN，其評估效能以平均存取時間 *Avg. AT* (access time)、平均聽頻道時間 *Avg. TT* (tuning time) 和能量 *E* (energy) 來評估。而本篇將與 SSI 進行效能分析比較。

4.1 系統模型

我們假設無線頻道上最小播放單位 (bucket) 的容量為 64 bytes，每個空間物件大小為 1024 bytes，索引項目內每項容量為 6 bytes。以圖 8 為例，每個空間物件為 $1024/64 = 16$ buckets。而 $C(1,0)$ 中的 C_t 、 G_t 、 DL 各為 6 bytes， SL 和 CL 各有 3 項為 18 bytes，總共為 $6 * 3 + 18 * 2 = 54$ bytes，未滿 64 bytes 以最小播放單位 1 bucket 計。本篇方法增加的索引項目 MER，加入後和 SSI 索引容量差異很小 (為熱門區塊左下與右上二個座標，一個索引項目)，此例一個索引皆以 1 bucket 計。

我們使用 SSI [12] 與 MLAIN [5] 的能量分析假設，考慮行動設備的 CPU 和網路卡之耗能，如表 1 所示。CPU 和網路卡在運作模式時 (active mode) 分別耗電為 400 mW/s 和 750 mW/s，總共為 1150 mW/s。而在省電模式時 (doze mode) CPU 和網路卡的耗電為 0.16 mW/s 和 24 mW/s，總共為 24.16 mW/s。考慮下載頻道 (downlink channel) 的頻寬為 1 Mbps。而平均能量為： $1150 * Avg. TT + 24.16 * (Avg. AT - Avg. TT)$ 。計算能量時，*Avg. AT* 與 *Avg. TT* 以秒為單位。平均存取時間 $Avg. AT = (total\ index\ AT * 1\ bucket + total\ data\ AT * 16\ buckets) / total\ buckets$ 。平均聽頻道時間 $Avg. TT = (total\ index\ TT * 1\ bucket + total\ data\ TT * 16\ buckets) / total\ buckets$ 。

表 1 系統耗能

Component	Doze mode	Active mode
CPU	0.16 mW/s	400 mW/s
Network interface card	24 mW/s	750 mW/s

4.2 實驗結果

我們以圖 10 為例，做效能分析。首先我們分析視窗查詢，以查詢視窗 $Q1$ 為例。查詢視窗 $Q1$ 必須要確認 $H(2,2)$ 和 $C(2,7)$ 區塊， $C(2,7)$ 為一般區塊在播放循環中只有出現一次，SSI 幾乎必須跑完整個播放循環才能確認 $C(2,7)$ 有無資料；而因為查詢視窗在索引項目 MER 範圍內，我們的方法只要索引找到 $H(2,2)$ 便可結束查詢。我們在每個時間點上算出存取時間 *AT* (access time) 與聽頻道時間 *TT* (tuning time)，如圖 12 所示。表 2 為視窗查詢的平均存取時間，表 3 為其相對應的平均聽頻道時間。兩者的結果都顯示 MER 方法效能優於 SSI 方法，MER 平均存取時間比 SSI 快了 2.8 倍左右。平均存取時間 *Avg. AT* 以 MER 為例，系統在所有 index 時間點的 bucket 總和為 1044，而在 data 時間點的 bucket 總和為 1482。索引結構總長度為 $24 (\text{index 個數}) * 1 + 21 (\text{data 個數}) * 16 = 360$ ，因 data 為 index 的 16 倍，套用公式為： $(1044 * 1 + 1482 * 16) / 360 = 68.767$ buckets。

表 2 Access time for the static window query

Method	#index buckets	#data buckets	<i>Avg. AT</i> (#buckets)
MER	1044	1482	68.767
SSI	4148	4172	196.944

表 3 Tuning time for the static window query

Method	#index buckets	#data buckets	Avg. TT (#buckets)
MER	451	820	37.697
SSI	504	861	39.667

連續視窗查詢以圖 10 內查詢視窗 Q_1 至 Q_2 為例。連續視窗查詢必須確認區塊 $H(2,2), C(2,4), C(2,3)$ 以及 $C(2,7)$ ，最後再使用 Minkowski regions [14] 分配資料有效區段。 $C(2,4), C(2,3)$ 以及 $C(2,7)$ 皆為一般區塊，SSI 同樣因為要確認這些區塊而耗費整個播放循環的時間；因為連續視窗查詢範圍在索引項目 MER 內，我們的方法只要索引找到 $H(2,2)$ 便可結束查詢。我們在每個時間點上算出存取時間 AT (access time) 以及聽頻道時間 TT (tuning time)，表 4 為平均存取時間，表 5 為平均聽頻道時間。結果顯示 MER 效能優於 SSI ， MER 平均存取時間比 SSI 快了 3 倍左右。

表 4 Access time for the continuous window query

Method	#index buckets	#data buckets	Avg. AT (#buckets)
MER	1428	1818	84.767
SSI	5661	5411	256.214

表 5 Tuning time for the continuous window query

Method	#index buckets	#data buckets	Avg. TT (#buckets)
MER	840	1163	54.022
SSI	934	1238	57.617

最後我們依照先前的假設算出 MER 與 SSI 視窗查詢與連續視窗查詢的能量消耗，結果如表 6。結果顯示 MER 消耗能量皆比 SSI 少。能量計算我們以 MER 視窗查詢為例，套用公為： $1150 * 37.697 + 24.16 * (68.767 - 35.697) = 44102$ buckets。我們假設下載頻道 (downlink channel) 的頻寬為 1 Mbps，將結果乘上 $64/(1024^2/8)$ ，換算後得到結果 21.549 mW。

表 6 能量消耗比較

Method	Static	Continuous
MER	21.549 mW	30.712 mW
SSI	24.206 mW	34.793 mW

以上的結果驗證我們提出的 MER 空間索引方法效能優於 SSI 方法，接下來我們將會做更大量數據的效能分析。

5. 結論

在本文中，我們在無線數據廣播系統中使用非均勻廣播模式的 MER 空間索引方法，支援處理視窗查詢以及連續視窗查詢。本篇方法應用 MER 索引資訊，將熱門區塊有效範圍加大，解決了視窗查詢與連續視窗查詢的範圍容易包含熱門資料區塊和一般資料區塊的問題，減少查詢過程中不必要檢查的區塊資料之處理程序，縮短了存取時間與聽頻道時間。我們也進行效能評估，驗證我們的方法優於現有的方法。

參考文獻

- [1] Acharya, S., Franklin, M., Zdonik, S. and Alongso, R., "Broadcast disks: data management for asymmetric communications environments," *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, pp.199–210, 1995.
- [2] Aksoy, D. and Franklin, M., "Rwx: A scheduling approach for large-scale on-demand data broadcast," *IEEE/ACM Transactions on Networking*, Vol. 7, No. 6, pp. 846–860, 1999.
- [3] Imielinski, T., Viswanathan, S. and Badrinath, B.R., "Data on air-organization and access," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 9, No. 3, pp.353–372, 1997.
- [4] Imielinski, T., Viswanathan, S. and Badrinath, B.R., "Energy efficient indexing on air," *ACM SIGMOD Record*. Vol. 23. No. 2. ACM, 1994.
- [5] Im, S. and Choi, J., "MLAIN: multi-leveled air indexing scheme in non-flat wireless data broadcast for efficient window query processing," *Computers and Mathematics with Applications*, Vol. 64, No. 5, pp.1242–1251, 2012.
- [6] Im, S., Youn, H.Y., Choi, J. and Ouyang, J., "A novel air indexing scheme for window query in non-flat wireless spatial data broadcast," *Journal of Communications and Networks*, Vol. 13, No. 4, pp. 400–407,

2011.

- [7] In, K., Jung, H., Youn, H.Y., and Kim, U.M., “Efficient processing of spatial range queries on wireless broadcast streams,” *International Journal of Database Management Systems*, Vol. 6, No. 1, pp. 29–43, 2014.
- [8] Lee, B.G. and Chung, W.Y., “A smartphone-based driver safety monitoring system using data fusion,” *Sensors*, Vol. 12, pp. 17536–17552, 2012.
- [9] Naamad, A., Lee, D. T. and Hsu, W.-L., “On the maximum empty rectangle problem,” *Discrete Applied Mathematics*, Vol. 8, No. 3, pp. 267–277, 1984.
- [10] Park, K., “Efficient data access for location-dependent spatial queries,” *Journal of Computer Science and Technology*, Vol. 29, No. 3, pp. 449–469, 2014.
- [11] Park, K., “An efficient scalable spatial data search for location-aware mobile services,” *Journal of Information Science and Engineering*, Vol. 31, No. 1, pp. 165–178, 2015.
- [12] Shen, J.H. and Jian, M.S., “Spatial query processing for skewed access patterns in nonuniform wireless data broadcast environments,” accepted by *International Journal of Ad Hoc and Ubiquitous Computing*, Sept. 2014.
- [13] Shen, J.H., Lu, C.T. and Jian, M.S., “Neighbor-index method for continuous window queries over wireless data broadcast,” *Applied Mechanics and Materials*, Vols. 284–287, pp. 3295–3299, 2013.
- [14] Zheng, B., Lee, W.C., Lee, C.K., Lee, D.L. and Shao, M., “A distributed spatial index for error-prone wireless data broadcast,” *The VLDB Journal*, Vol. 18, No. 4, pp. 959–986, 2009.
- [15] Zheng, B., Lee, W.C. and Lee, D.L., “Spatial queries in wireless broadcast systems,” *Wireless Network*, Vol. 10, No. 6, pp. 723–736, 2004.