組込 DBMS における空間データの k 最近傍検索手法

k-Nearest Neighbor Search Method of Spatial Data in Embedded DBMS

林 秀樹♠　伊藤 大輔♠　谷崎 正明♦
木村 耕治†　梶山 尚紀‡

Hideki HAYASHI, Daisuke ITO,
Masaaki TANIZAKI, Kohji KIMURA,
Hisanori KAJIYAMA

本論文では，車載情報端末へ適用可能な組込 DBMS における k 最近傍検索手法を提案した．提案手法は，空間索引に適用可能な二つの従来手法である(1)RKV 法と(2)HS 法に基づき，組込機器向けに検索時間の短縮とメモリ使用量の削減を実現する．シミュレーション実験の結果より，提案手法のディスクアクセスの回数は，HS 法と同じで，RKV 法より最大 12%少ないことを確認した．また，提案手法のメモリ使用量は，HS 法より，最大 68%少ないことを確認した．さらに，実機評価の結果より，提案手法は，車載情報端末に適用可能である見通しを得た．

In this paper, we proposed a k-nearest neighbor search method in an embedded DBMS for in-vehicle information terminals. Our method based on two conventional methods, (1)the RKV method and (2)the HS method, applicable to any spatial index shortens the search time and reduces the size of the memory usage. Our simulation results showed that our method needed the same number of disk accesses with the HS method and up to 12% smaller number than the RKV method. Our method needed up to 68% smaller size of the memory usage than the HS method. Additionally, results of our actual experiment showed that our method was applicable to in-vehicle information terminals.

1. はじめに

1.1　研究背景

近年，大容量のハードディスクドライブ(HDD)を搭載したカーナビゲーション装置やマルチメディアレコーダが普及している．また，モバイル端末に搭載可能な小型のフラッシュメモリが大容量化してきている．組込機器で管理すべきデータ量とデータの種類が急増しているため，高度なデータ管理機能が組込ソフトウェアに要求されている．

♠正会員　（株）日立製作所 中央研究所
{hideki.hayashi.xu,daisuke.ito.mq }@hitachi.com
♦非会員　（株）日立製作所 中央研究所
masaaki.tanizaki.tj@hitachi.com
†非会員　（株）日立製作所 ソフトウェア事業部
kohji.kimura.zn@hitachi.com
‡非会員　日立ソフトウェアエンジニアリング（株）
h-kajiyama@hitachisoft.jp

組込ソフトウェアのステップ数は，大規模になると，500万行以上になる[1]．その約半分は，データ管理である．組込ソフトウェアの開発期間が短縮される現状では，開発工数の削減に加え，信頼性の向上は急務である．この課題を解決するため，組込機器のミドルウェアとして，データベース管理システムが製品化されている[2,3,4]．開発者が組込機器用のデータベース管理システム（以下，組込DBMSと呼ぶ）を利用することで，開発工数を削減できる．

1.2　研究動機と目的

組込DBMSの適用先は，大容量の記憶装置を備え，かつデータの件数と種類が多いコンテンツを管理する端末となる．その代表例として，地図コンテンツを管理するカーナビゲーション装置やPND(Personal Navigation Device)等の車載情報端末が挙げられる．現在，車載情報端末では，アプリケーションの開発効率を高めるため，その開発にDBMSを導入する動きがある．本研究では，車載情報端末の地図コンテンツの管理に，組込DBMSを適用した場合を想定する．

車載情報端末でニーズの高い地図検索として，k最近傍検索がある．k最近傍検索は，地図で指定した地点（以下，質問地点と呼ぶ）に距離の近い上位k件の地点データを取得する検索である．例えば，「現在地から距離の近い上位10件の駐車場」や「目的地から距離の近い上位20件の和食レストラン」という検索に用いられる．車載情報端末では，地域毎に異なる施設の存在密度に関わらず，限られたメモリ使用量と処理時間で，指定した件数を取得できることが求められる．

本論文では，車載情報端末で適用可能な組込DBMSにおけるk最近傍検索手法を提案する．提案手法では，組込機器向けに検索時間の短縮とメモリ使用量の削減を実現する．その際，提案手法は，領域と位置の対応関係が不規則で，階層的に領域を管理する四分木やR-tree[5]等の空間索引へ適用できる．検索時間の短縮は，検索時間の大半を占めるメモリ等の主記憶装置とHDD等の二次記憶装置の間の入出力回数を削減することで，実現する．一方，メモリ使用量の削減は，k最近傍検索の処理中に，主記憶装置にk件の地点データのみを保持することで実現する．なお，提案手法はR-treeにも適用できるが，本論文では，車載情報端末のk最近傍検索に適した四分木に適用する場合で，提案手法を説明する．また，シミュレーション実験では，提案手法と従来手法を比較し，提案手法の有用性について述べる．さらに，提案手法を（株）日立製作所と日立ソフトウェアエンジニアリング（株）で共同開発した組込DBMS製品であるEntier[2]に実装し，実機評価で車載情報端末への適用可能性について検討する．

以下では，2章で，提案手法の説明に用いる空間索引について述べる．3章で従来のk最近傍検索技術について説明し，4章で提案手法について述べる．5章で提案手法の性能評価を示し，最後に6章で，本論文のまとめを述べる．

2. 空間索引

本章では，車載情報端末のk最近傍検索に適した四分木のデータ構造について説明する．また，その四分木を利用した空間検索について述べる．

2.1　四分木のデータ構造

四分木は，空間領域を再帰的に四分割した際にできた分割領域を，木構造で表現したものである．これまでに，様々な四分木のデータ構造が提案されているが，その中で，車載情

報端末のk最近傍検索に適したデータ構造を用いる．

まず，車載情報端末のハードウェア構成から，四分木のデータ構造を検討する．車載情報端末は，地図コンテンツを格納する二次記憶装置と，実行中のアプリケーションプログラムや，それに必要なデータを格納する主記憶装置から構成される．このような構成では，空間検索を実行した場合に，二次記憶装置と主記憶装置の間の入出力回数を削減するため，バケット方式の四分木が適する．バケット方式の四分木では，距離の近い地点データの集合を一つのバケットで二次記憶装置に格納し，主記憶装置への入力をバケット単位で行う．

次に，車載情報端末のk最近傍検索の特徴から，四分木のデータ構造を検討する．前章で述べたように，車載情報端末では，地点データの種別を指定したk最近傍検索のニーズが高い．具体的には，「現在地から距離の近い上位10件の駐車場」といった検索が該当する．このようなk最近傍検索を実現できるように，四分木のデータ構造に地点データの座標だけでなく，種別の情報も含める．

以上の点を考慮し，具体的な四分木のデータ構造を示す．図1は四分木による空間領域の分割を，図2は図1に対応した木構造を示す．図1に記載のアルファベットは地点データの識別子，数字は分割領域の識別子を表す．図2の木構造は，分割領域の情報を含む「枝ノード」（図2の丸）と，地点データの情報を含む「葉ノード」（四2の四角）から構成される．枝ノードと分割領域は，一対一の対応になる．枝ノードは，分割領域の範囲，四分割した座標，下位の枝ノードの識別子，葉ノードへのポインタ等を含む．枝ノードの子は，その枝ノードに対応した分割領域の下位の分割領域に対応する．

葉ノードは，最下位の分割領域に対応した枝ノードに隣接する．葉ノードは，その分割領域に含まれる地点データの位置と種別の情報を含む．葉ノードはバケットに相当し，1ページを割り当てる．1ページに格納可能な地点データの個数を超えると，その分割領域を四分割する．四分割する場合，その分割領域に含まれる全地点データの重心座標を分割点とする．重心座標で四分割する理由は，分割後の四つの領域内の地点データの件数をできる限り等しくするためである．

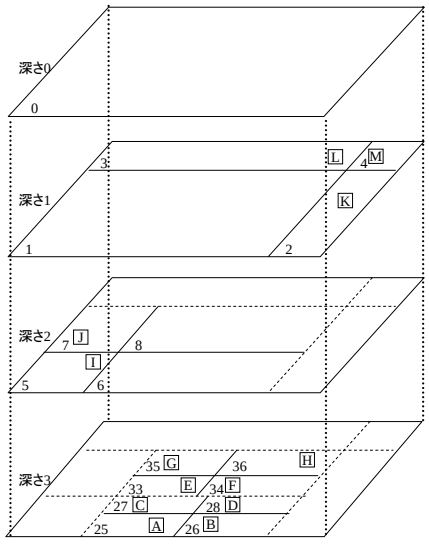

図1　四分木による空間分割
Fig.1 Spatial partitoning in quadtree.

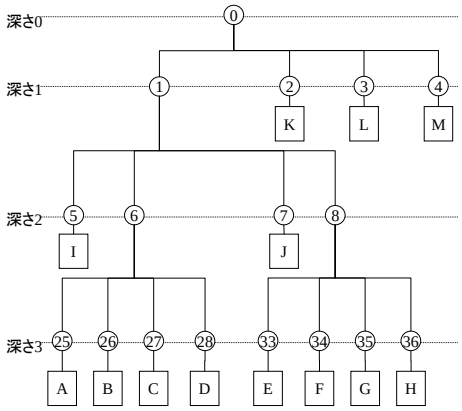

図2　四分木の木構造
Fig.2 Quadtree structure.

2.2　四分木を利用した空間検索

四分木の枝ノードと葉ノードの情報は，HDD等の二次記憶装置にページ単位で格納されている．空間検索の処理で，ある分割領域の地点データを取得する場合，まず，その分割領域に対応した葉ノードまでの枝ノードを格納しているページを主記憶装置に読み込む．そして，その分割領域に対応した葉ノードのページを二次記憶装置から主記憶装置に読み込む．本論文では，このような読み込み処理をディスクアクセスと呼ぶ．一般にディスクアクセスの処理時間は計算処理の時間より非常に長い．車載情報端末では，ディスクアクセスの処理時間が15ミリ秒程度であるのに対し，k最近傍検索で多発する二点間の距離計算の処理時間は10マイクロ秒程度である．そのため，ディスクアクセスの回数を削減できれば，検索時間を短縮できる．

図1の分割領域33の地点データEを検索する場合を考える．この場合，枝ノード0，1，8，33の順に探索し，枝ノード33に接続している地点データEを含む葉ノードにアクセスする．その際，枝ノード0，1，8，33を格納するページと，地点データEを含む葉ノードを格納するページに対し，ディスクアクセスが発生する．なお，アクセスしたページは，メモリ上のバッファ（空間バッファ）に格納される．空間バッファ内のページには，ディスクアクセスが発生しない．

3. 従来の k 最近傍検索技術

空間データベースの分野では，これまでに多数のk最近傍検索技術が提案されている．古くは，指定した任意の地点から距離の近い上位k件の地点データを取得する手法[6,8]が提案されている．さらに，指定した道路などの線や，建物や施設などの面から距離の近い上位k件の地点データを取得する手法が提案されている[7,9,10]．

本研究では，組込DBMS向けのk最近傍検索の最初の取組みとして，任意の地点から距離の近い上位k件の地点データを取得する手法を提案する．このようなk最近傍検索に関しては，文献[6,8]の手法が基盤になっている．以下では，文献[8]の手法をRKV法，文献[6]の手法をHS法とし，2.1節の四分木に適用する場合で説明する．

3.1　RKV法

本節では，RKV法の特徴について述べる．

・深さ優先探索で木構造を探索

検索処理中に四分木を探索する際，同じ深さの節の中で，質問地点に最も近い分割領域に対応した節を選択し，葉まで辿る．葉に到達すると，一つ上位の節に戻る．そして，未探索の節の中で，質問地点に最も近い分割領域に対応した節を選択し，同様に処理する．この探索では，未探索の節に対応した分割領域の情報をスタックで管理する．

・木構造の探索時に枝刈りを実行
ディスクアクセスの回数を削減するため，枝刈りを行う．枝刈り判定は，質問地点と節に対応した分割領域との距離と，質問地点とk番目の地点データとの距離を比較する．前者の距離が大きい場合，当該節を探索しない．これは，当該節に対応した分割領域内の地点データが，上位k番以内に入らないためである．これは，質問地点と分割領域との距離は，質問地点と当該分割領域内の地点データとの距離より，必ず短くなるためである．分割領域に地点データが必ず含まれるという四分木の特徴を利用している．

・主記憶装置に上位k件のみ地点データを保持
メモリ使用量を削減するため，主記憶装置に保持する地点データは，上位k件のみとする．主記憶装置にk件の地点データを格納している状態で，検索結果になり得る地点データを見つけると，質問地点に最も遠い地点データを破棄する．この処理は，ヒープ構造で地点データを管理することで，効率的に行える．このヒープ構造では，質問地点から最も遠い地点データの情報が，最も高い優先度となり，ヒープ構造の先頭要素に配置される．

3.2 HS法

本節では，HS法の特徴について述べる．

・最良優先探索で木構造を探索
質問地点から近い分割領域に対応した節を優先的に探索する．未探索の節に対応した分割領域の情報をヒープ構造で管理する．このヒープ構造では，質問地点に最も近い分割領域の情報が最も高い優先度になり，ヒープ構造の先頭要素に格納される．分割領域の情報には，分割領域の識別子と，質問地点と分割領域との距離が含まれる．ヒープ構造の先頭要素を取り出すことで，質問地点に最も近い分割領域に対応した節を取得できる．そして，取得した分割領域に対応した節の子に対応した分割領域の情報をヒープ構造に挿入する．

・探索済みの分割領域内の全地点データを保持
上記のヒープ構造には，分割領域の情報の他に，地点データの情報を格納する．地点データの情報には，地点データの識別子と，質問地点と地点データとの距離が含まれる．ヒープ構造から取得した分割領域が最下位の節の場合，当該分割領域内の全地点データをヒープ構造に挿入する．

・検索結果になり得る地点データが存在しない場合，検索処理を終了
検索処理の実行中，ヒープ構造の先頭が地点データならば，その地点データは質問地点に最も近い地点データとなり，検索結果になる．この場合，当該地点データをヒープ構造から取り出し，検索結果のリストに保持する．検索結果のリストにk件格納された時点で，検索処理が終了となる．この処理は，質問地点と分割領域との距離が，質問地点と当該分割領域内の地点データとの距離より，必ず短いという四分木の特徴に基づく．

4. 提案手法

本章では，組込DBMS向けのk最近傍検索手法を提案する．組込DBMSでは，短い検索時間と小さなメモリ使用量が要求される．提案手法は従来手法と異なり，検索時間だけでなく，メモリ使用量も考慮する必要がある．

以下では，まず，提案手法の特徴について説明した後，詳細なアルゴリズムについて述べる．次に，提案手法の特性を説明し，RKV法やHS法と比較する．

4.1 特徴

提案手法はRKV法とHS法の長所を備え，以下の特徴をもつ．

・最良優先探索で木構造を探索
ディスクアクセスの回数を削減するため，質問地点から距離の近い分割領域に対応した節を優先的に探索する．最良優先探索では，未探索の節に対応した分割領域の情報をヒープ構造で管理する．提案手法の分割領域の情報の管理は，HS法と同様である．ただし，HS法と違い，このヒープ構造には，分割領域の情報のみ管理する．これは，次の項目で述べるように，ヒープ構造内の分割領域と地点データの優先度の定義が異なるためである．

・主記憶装置に上位k件のみ地点データを保持
メモリ使用量を削減するため，RKV法と同様の方法で，主記憶装置に保持する地点データは，質問地点に近い上位k件のみとする．ヒープ構造内の地点データの優先度は，質問地点から距離の遠い地点データほど高くなる．

・検索結果になる上位k件の地点データを発見できた時点で検索処理を終了
ディスクアクセスの回数を削減するため，上位k件の地点データを発見できた時点で検索処理を終了する．この判定は，未探索の分割領域の中で，質問地点に最も近い分割領域との距離と，質問地点とk番目の地点データの距離を比較する．前者が長い場合，k番目の地点データより質問地点に近い地点データが存在しないため，処理を終了する．

さらに提案手法では，地点データの種別を指定したk最近傍検索を実行できる．これは，2.1節の四分木の葉ノードに地点データの種別を含むためである．具体的には，地点データを管理するヒープ構造に，検索条件に指定した種別に一致する地点データのみ保持することで，実現できる．

4.2 アルゴリズム

提案手法では，分割領域と地点データを管理する四分木，結果候補になり得る地点データを含む分割領域の情報を格納するヒープ構造（候補領域ヒープ），結果候補になり得る地点データを格納するヒープ構造（候補地点ヒープ）を用意する．

図3に提案手法のアルゴリズムを示す．候補領域ヒープでは，質問地点との距離が最短の分割領域の情報がヒープの先頭要素に格納される．先頭要素から順に分割領域を取得することで，質問地点から近い順に分割領域を探索できる．

候補地点ヒープでは，上位k件の地点データの情報を格納する．その際，質問地点までの距離が最長の地点データの情報が先頭要素になる．これにより，候補地点ヒープに地点データがk件存在する状態で，新たに地点データをヒープに挿入できるかの判定処理は，当該地点データの距離とヒープの先頭要素の距離との比較になる．

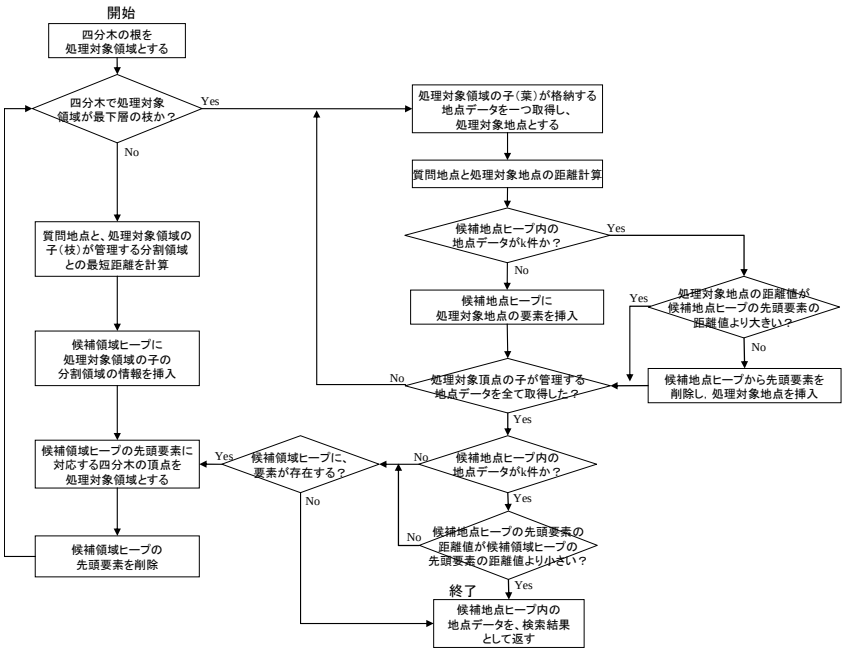

図3 提案手法のアルゴリズム
Fig.3 Algorithm of proposed method.

検索処理の終了判定は，候補領域ヒープの先頭要素の距離と，候補地点ヒープの先頭要素の距離を比較する．前者が大きい場合，候補領域ヒープに格納される分割領域の地点データを調べても，上位k件に入らないため，処理を終了する．

4.3 従来手法との比較

表1に示すように，提案手法の特徴は，ディスクアクセスの回数が少なく，かつメモリ使用量が少ない点である．なお，表1の木構造探索と地点データ管理の項目の括弧内は，その機能を実現するために用いるデータ構造を示す．

表1 k最近傍検索手法の比較
Tab.1 Comparison of k-nearest neighbor search method.

	提案手法	RKV法	HS法
木構造探索	最良優先 （ヒープ）	深さ優先 （スタック）	最良探索 （スタック）
地点データ管理	上位k件のみ （ヒープ）	上位k件のみ （ヒープ）	探索済み全て （ヒープ）
ディスク アクセス回数	少ない	多い	少ない
メモリ使用量	小さい	小さい	大きい

5. 性能評価

本章では，まず，提案手法の有効性を示すために行ったシミュレーション実験の結果について述べる．次に，提案手法の実機評価の結果について述べる．

5.1 シミュレーション実験

5.1.1 実験環境

シミュレータは，地点データの集合から四分木を作成し，提案手法，RKV法，HS法のいずれかのk最近傍検索を実行する．そして，k最近傍検索を実行した際のディスクアクセスの回数やメモリ使用量の性能を測定する．

本実験の評価項目は，ディスクアクセスの回数とメモリ使用量である．ディスクアクセスの回数は，上位k件の地点データの検索処理を終えるまでに，葉ノード，または枝ノードを格納しているページを二次記憶装置から主記憶装置に読み込む回数を示す．メモリ使用量は，分割領域と地点データの情報を保持する主記憶装置上のサイズを示す．

表2は実験のパラメータ設定を示し，括弧内の値はデフォルト値を示す．この設定は，車載情報端末における検索処理を想定したものである．本実験では，正方形の空間領域内に一様乱数で発生させた地点データを用いた．地点データが250,000件の場合には東京郊外の国分寺駅前の密度，750,000件の場合には新宿駅前の密度にそれぞれ相当する．そして，質問地点を空間領域内に一様乱数で50 回設定し，50 回測定したディスクアクセスの回数とメモリ使用量の平均値を実験結果とした．空間バッファの管理方法は，LRU (Least Recently Used) に従う．

表2 パラメータの設定
Tab.2 Parameter configuration.

パラメータ	説明	値
空間領域のサイズ	地点データを配置する領域のサイズ	10km×10km
地点データの件数	空間領域に配置する地点データの件数	10,000～800,000 (250,000)
質問地点の設定領域	一様乱数で生成する質問地点の範囲	8km×8km
k	検索結果となる地点データの件数	5～30 (10)
検索対象の割合	地点データの総数に対する検索条件の種別に一致する件数の割合	0.05
空間バッファのサイズ	枝ノードと葉ノードのページを格納するメモリのサイズ	128kbyte

5.1.2 地点データの件数による影響

図4と図5に，地点データの件数を変化させた場合のディスクアクセスの回数とメモリ使用量をそれぞれ示す．

図4の結果より，提案手法のディスクアクセスの回数は，HS法と同じであることがわかる．これは，提案手法では，HS法と同じ最良優先探索を用い，四分木の探索する節が同じになるためである．一方，提案手法のディスクアクセスの回数は，RKV法より最大12%少ない（地点データが800,000件の場合）．この場合，提案手法は，RKV法より0.84回少ないことになる．車載情報端末の二次記憶装置をHDDとすると，1回のディスクアクセスの処理時間が15ミリ秒となり，提案手法の検索時間はRKV法より約13ミリ秒早いことになる．車載情報端末の検索の要求性能を考えると，検索処理のみで500ミリ秒以内が妥当である．この要求からすると，この差は小さいと言える．

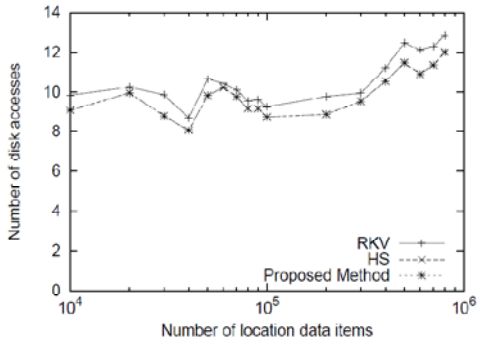

図4 地点データの件数とディスクアクセスの回数
Fig.4 Number of disk accesses vs. number of location data items.

表3に，地点データが800,000件で，各質問地点における提案手法とRKV法のディスクアクセスの回数を調べ，その差と頻度の結果を示す．この結果より，50回の実験で38回において，提案手法とRKV法に差がないことがわかる．残りの12回の実験において，両手法の差が出る．その差の最大は7回（検索時間にして105ミリ秒）になり，この差は大きいと言える．この傾向は，地点データが800,000件以外でも見られる．以上より，提案手法とRKV法のディスクアクセスの回数は大部分同じだが，その差が大きくなる可能性がある．提案手法とRKV法のディスクアクセスの回数の差が大きいとき，質問地点を四分木の分割領域の境界付近に設定した場合が考えられる．この場合，四分木で下位の枝ノードまで辿った後，上位の枝ノードに戻る必要がある．

表3 提案手法とRKV法のディスクアクセス回数の差
Tab.3 Difference in number of disk accesses between proposed method and RKV method.

回数の差	0	1	2	3	4	5	6	7
頻度	38	1	3	3	1	3	0	1

図5の結果より，提案手法のメモリ使用量は高々700バイト程度なので，車載情報端末で十分に適用可能な値である．また，提案手法は，HS法より最大68%小さい（地点データが20,000件）．提案手法とHS法のメモリ使用量の差を絶対値で示すと，最大970バイトとなる（地点データが90,000件）．車載情報端末への適用を想定すると，この差は小さいと言える．また，提案手法のメモリ使用量は，RKV法より最大24%大きい（地点データが700,000件）．提案手法とRKV法のメモリ使用量の差を絶対値で示すと，最大140バイトとなり（地点データの件数が70,000件），この差も小さいと言える．

以上の結果より，各手法間でメモリ使用量の絶対値の差は小さく，車載情報端末の性能に影響を与えるものではない．しかし，車両の移動を考慮し，継続的にk最近傍検索を実行する場合には，メモリ使用量の差が大きくなり，車載情報端末の性能に影響を与えるものと考えられる．

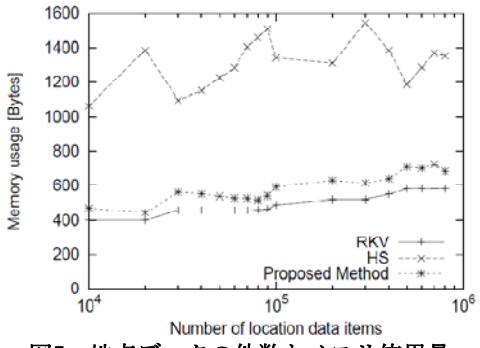

図5 地点データの件数とメモリ使用量
Fig.5 Memory usage vs. number of location data items.

5.1.2 kによる影響

図6と図7に，kを変化させた場合のディスクアクセスの回数とメモリ使用量の結果をそれぞれ示す．

図6の結果より，提案手法のディスクアクセスの回数はRKV法より少なく，最大14%（2.2 回）少ない（kが20）．この場合，検索時間にして33ミリ秒（2.2回×15ミリ秒）の差となり，小さいと言える．一方，kが20の場合で，各実験の提案手法とRKV法のディスクアクセスの回数差を調べると，最大12回（検索時間にして180ミリ秒）になることを確認した．この差は，車載情報端末の要求性能を考慮すると，大きいと言える．kの値が大きくなると，各手法ともに，ディスクアクセスの回数が増える．これは，上位k件を発見するまでに，探索すべき分割領域の数が増えるためである．

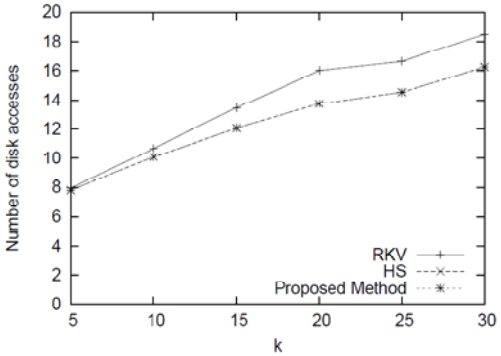

図6 kとディスクアクセスの回数
Fig.6 Number of disk accesses vs. k.

図7の結果より，提案手法のメモリ使用量は，HS法より最大58%削減できている（kが10）．提案手法とHS法のメモリ使用量の絶対値の差は最大1,242バイトになる（kが30）．この差は，車載情報端末への適用を想定すると，性能に影響を与える範囲ではない．しかし，図5の考察と同様に，車両の移動を考慮して，kの値を大きくし，かつk最近傍検索を継続的に実行する場合，メモリ使用量の差が性能に影響を与える

と考えられる．一方，提案手法のメモリ使用量はRKV法より最大20%大きい（kが15）．提案手法とRKV法のメモリ使用量の差は最大140バイトとなり（kが30），この差も小さいと言える．

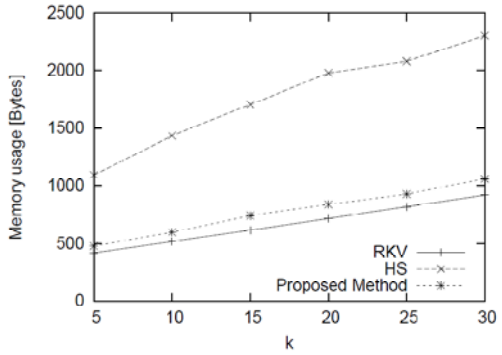

図7　kとメモリ使用量
Fig.7 Memory usage vs. k.

5.2　実機評価

5.2.1　実験環境

組込DBMS製品Entier[2]に提案手法のプロトタイプを実装し，PDA(Personal Digital Assistant)を用いた実験を行った．実験用のデータベースを，表2のデフォルト値で構築し，4ギガバイトで毎分3,600回転の1.0インチHDDに格納した．評価項目は，k最近傍検索の処理時間である．この処理時間は，k最近傍検索を実行してから，上位k件の地点データを特定するまでの時間を示す．

5.2.1　kによる影響

図8に，kの値を変化させた場合の処理時間の結果を示す．この結果より，空間検索の処理時間が高々140ミリ秒（kが30の場合）であることがわかる．この結果は，提案手法が車載情報端末への適用に十分な性能であることを示す．

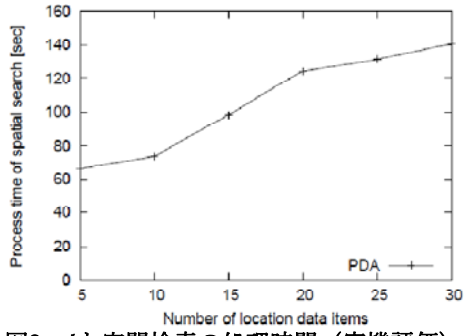

図8　kと空間検索の処理時間（実機評価）
Fig.8 Processing time of proposed method vs. k in actual evaluation.

6. まとめ

本論文では，車載情報端末で適用可能な組込DBMSにおけるk最近傍検索手法を提案した．提案手法は，組込機器向けに検索時間の短縮とメモリ使用量の削減を実現する．シミュレーション実験の結果より，提案手法のディスクアクセスの回数は，HS法と同じで，RKV法より最大12%少ないことを確認した．また，提案手法のメモリ使用量は，HS法より，最大68%少ないことを確認した．さらに，実機評価の結果より，提案手法は，車載情報端末に適用可能である見通しを得た．

今後の課題として，組込DBMS向けの空間索引技術を検討する予定である．本研究では，地点データの種別を考慮したk最近傍検索を実現するための一つの索引構造として，四分木を用いた．この四分木では，地理的に近い地点データをディスク上で近くに配置するといったディスク上の最適な配置が十分に行われているとは言えない．この点を改善できれば，空間検索のさらなる性能向上が見込める．

[文献]

[1] 経済産業省, “2007 年版組込みソフトウェア産業実態調査報告書 -プロジェクト責任者向け調査-” (2007).
[2] 日立 組み込みデータベース Entier, http://www.hitachi.co.jp/Prod/comp/soft1/ Entier/
[3] Oracle Database Lite, http://www.oracle.co.jp/database/Lite_Edition.html
[4] IBM DB2 Everyplace, http://www-06.ibm.com/jp/software/data/db2/everyplace/
[5] A. Guttman, “R-Trees: A dynamic index structure for spatial searching,” Proc. ACM SIGMOD'84, pp.47-57 (1984).
[6] G.R. Hjaltason and H. Samet, “Distance browsing in spatial databases,” ACM Trans. on Database System, vol.24, no.2, pp.265-318 (1999).
[7] H. Hu and D.L. Lee, “Range nearest-neighbor query,” IEEE Trans. on Knowledge and Data Engineering, vol.18, no.1, pp.78-91 (2006).
[8] N. Roussopoulos, S. Kelley, and F. Vincent, “Nearest neighbor queries,” Proc. of ACM SIGMOD'95, pp.71-79 (1995).
[9] Z. Song and N. Roussopoulos, “K-nearest neighbor search for moving query point,” Proc. of Int'l Symp. on Advances in Spatial and Temporal Databases, pp.79-96 (2001).
[10] Y. Tao, D. Papadias, and Q. Shen, “Continuous nearest neighbor search," Proc. of VLDB'02, pp.287-298 (2002).

林 秀樹　Hideki HAYASHI
2002 年大阪大学工学部電子情報エネルギー工学科卒業．2004 年同大大学院情報科学研究科博士前期課程修了．2006 年同大学院博士後期課程修了．同年（株）日立製作所入社．中央研究所にて空間情報システム，モバイルデータベース等の研究開発に従事．博士（情報科学）．電子情報通信学会，情報処理学会，日本データベース学会の各会員．

伊藤 大輔　Daisuke ITO
2000 年東京工業大学工学部情報工学科卒業．2002 年同大大学院情報理工学研究科博士前期課程修了．同年（株）日立製作所入社．中央研究所にてデータベース管理システムの研究開発に従事．日本データベース学会会員．

谷崎 正明　Masaaki TANIZAKI
1993 年神戸大学工学部計測工学科卒業．1995 年同大大学院工学研究科修士課程修了．同年（株）日立製作所入社．中央研究所にて空間情報システム，モバイルデータベース等の研究開発に従事．主任研究員．情報処理学会会員．

木村 耕治　Kohji KIMURA
1986 年立教大学理学部数学科卒業．同年（株）日立製作所入社．ソフトウェア事業部 DB 設計部主任技師．リレーショナルデータベース管理システムの設計開発に従事．情報処理学会会員．

梶山 尚紀　Hisanori KAJIYAMA
1995 年九州大学理学部数学科卒業．同年日立ソフトウェアエンジニアリング（株）入社．（株）日立製作所ソフトウェア事業部にてリレーショナルデータベースのシステム開発に従事．