

地図データの更新に対応した組み込みデータベースの データ配置方式

伊藤 大輔 谷崎 正明 嶋田 茂[†]

あらまし カーナビゲーションシステムに代表される地図データベースを扱う情報端末のストレージが DVD-ROM から HDD に移行している。HDD は DVD-ROM より高速かつ更新可能という特長があり高機能化が進む情報端末の構成要素として望ましい。しかし現状では地図データをファイルとして格納しており、HDD を用いても新しい道路の追加のような差分更新を行えない。また、情報端末に DBMS を応用することで差分更新は可能になるが、更新に伴いデータのフラグメント化が進み参照速度が低下する。そこで更新後にもフラグメントを生じにくい重複 ID 近傍配置表を提案し、机上での性能見積りを行うことで参照速度が低下しない見通しを得た。

キーワード 組み込みデータベース、DBMS、地図データベース、モバイル、
カーナビゲーションシステム、更新

Proposal of Data Allocation Method Supporting Perpetually Updated Map Data for Embedded Database

Daisuke ITO Masaaki TANIZAKI and Shigeru SHIMADA[‡]

Abstract Storage devices used by handheld terminal that treats a geographic database as typified by a car navigation system is changing from DVD-ROM to HDD. HDD is updatable and faster than DVD-ROM, suit for the handheld terminal. But the current handheld terminal can't update the geographic database using only differential data because the handheld terminal treats the geographic database as a large file. Even if applying a DBMS to the handheld terminal and change it updatable, there is remains a fragmentation problem of a database incidental to long operation. To solve this problem, we propose a duplicate ID nearly located table having a tolerance to fragment and also present an estimate evaluation of our approach.

Keyword Embedded Database, DBMS, Geographic Database, Mobile, Car Navigation System, Update

1. はじめに

携帯電話やカーナビゲーションシステムなど情報端末の高機能化に伴い、情報端末に組み込まれるソフトウェア（組み込みソフトウェア）の開発規模が増大しており、ソフトウェア危機が叫ばれている。組み込みソフトウェアでは約半分のコードがデータ管理に費やされるとの報告もあり、情報端末向けのデータ管理ミドルウェアへの期待は高い。

特にカーナビゲーションシステムに代表されるモバイル向け地図コンテンツを扱う情報端末はデータ管理への性能要求が高い。これは情報端末ではユーザ操作に対する応答速度が求められる一方、地図コンテンツが数 GB に及び、その件数も数百万件に及ぶ巨大かつ件数の多いコンテンツであるためである。

このような大量の地図コンテンツを格納するためには、従来は CD-ROM や DVD-ROM といったデバイスが不可欠であったが、近年ではより高速な HDD が利用できる。しかし、現状のカーナビゲーションシステムでは、たとえ HDD を用いても新しい道路や建物の追加といった地図データの差分更新が行えない。これは HDD 上のファイルシステムに地図データを単一の

ファイルとして格納しているため、ファイルの部分的な更新が行えない事が原因である。それでもなおモバイル向け地図コンテンツを差分的に更新したいというニーズは世界的に広まっており、一例として現在は後進市場である欧州においても差分更新可能な地図データフォーマットの標準化が進んでいる。

モバイル向け地図コンテンツの管理にデータベース管理システム (DBMS) を導入することで、地図データの差分更新が可能になる。さらに読み込みや更新といった基本的な入出力処理だけではなく、条件を指定した複雑な検索操作も宣言的な問い合わせ言語を用いて容易に行う事ができる。そのため、複雑化の進む情報端末のデータ管理ミドルウェアとして DBMS を導入することは、組み込みソフトウェアの開発期間短縮に大きく貢献する。

しかし、現状の DBMS をそのまま導入すると、更新後にデータのフラグメント化が進み、参照速度が低下するという問題がある。この参照速度低下を避けるためには定期的なデータの再整理処理が必要になるが、コンシューマ機器である情報端末においては煩雑なメンテナンス操作は受け入れられない。つまり情報端末

[†]株式会社日立製作所中央研究所, {d.ito, tanizaki, shimada}_at_crl.hitachi.co.jp

[‡]Central Research Lab., Hitachi, Ltd., {d.ito, tanizaki, shimada}_at_crl.hitachi.co.jp

下のような簡単な SQL 命令によって記述される。

```
select * from 地図表 where 区画番号
between 100 and 108;
```

このような宣言的インターフェースを用いる場合に付随して生じるメリットとして、ソフトウェアのモジュラリティが高まることも挙げられる。近年、組み込みソフトウェアの開発規模が拡大しているにもかかわらず開発期間を短縮するニーズが高く、このニーズを満たすためにはモジュラリティを高める事が必須であると考えられている。そのため、DBMS を導入することで得られるメリットは大きい。

しかし、従来の DBMS は宣言的なインターフェースを持つ一方で物理的なデータ配置を制御できないため、DBMS を用いない場合と比較して、経路探索処理に伴う近傍 n 区画読み込みの際のランダムアクセスの割合が増加する可能性がある。たとえ HDD であってもシークを伴うランダムアクセスはシーケンシャルアクセスと比べて遅く、結果的に経路探索処理の応答速度が遅くなる。例えば HDD 上で単純に DBMS を用いてランダムアクセスが大量に発生した場合、DVD-ROM 上のシーケンシャルアクセスを用いた場合よりも同一データ量の読み込み所要時間が長くなってしまふ事が経験的に分かっている。

また、更新および挿入処理の回数に依存して検索処理の所要時間が劣化する事も予想される。長さの異なる行の更新および挿入処理に伴いデータのフラグメントが発生し、検索処理の際のシークの回数が増加し、所要時間が劣化する。シークの回数は更新および挿入処理の回数におおよそ比例して増加するため、結果として運用期間に伴い検索処理の所要時間が劣化することになる。従来の DBMS はデータセンターのような場所を想定し、定期的にフラグメント解消のためのメンテナンス時間を確保できる運用を想定しており、夜間のような処理負荷の低い時間帯にデータの再整列処理を行うことでこの問題に対処してきたが、コンシューマ機器ではメンテナンス時間は容易に確保できず、カーナビゲーションシステムにおいても、車検など 2~3 年に一度しかメンテナンスは行えない。そのため、運用期間に伴う検索処理所要時間の劣化は避けられない。

2.3. 問題解決に向けた課題の整理

以上のことから、モバイル向け地図データベースを用いる情報端末に DBMS を用いる場合の課題は以下の 2 つである。

1. 近傍 n 区画の読み込み処理のシーク回数を削減すること
2. 2~3 年間はメンテナンスフリーとすること

本研究では、これら 2 つの課題を満たす表の構成法を提案する。

3. 重複 ID 近傍配置表の提案

3.1. 重複 ID 近傍配置表

本研究で想定する DBMS では図 3 に示すページとセグメントという物理的な単位によって構成されるデータ領域を用いる。ページはデータ I/O の最小単位を表し、セグメントは表やインデックスに領域を割り当てる際の最小単位を表す。表を構成する各行は、ページ

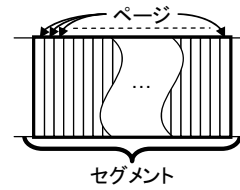


図 3: ページとセグメントからなるデータ領域

ID	...	道路種別	...	座標
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮

重複キーを含む列

図 4: 交差点表の例

の中に格納される。DBMS へのデータロード操作では、後の更新の為にページやセグメントの中に未使用領域を残しておく事がある。本研究で提案する手法でも、この運用に倣い、未使用領域を残しておく。また、区画に区切られた交差点データを格納する表を図 4 に示す。ここで ID 列は区画識別子を表し、重複キーを含むことが分かっている。

ここで、2.3 節に記した目的および課題を解決するために、指定した列の値が重複する行を連続領域に配置する機能を有する重複 ID 近傍配置表を新たに導入することを提案する。先に記したバッファを用いた一括読み込み機能により、図 5 の例では ID=13 の行の集合が連続読み込みによって高速に読み込まれる。なお、図 5 の隣接セグメントからなる同一 ID の行集合を、以降の説明の中では図 6 のように略記する。

交差点表に対して提案方式を用いた場合の更新もしくは挿入操作の手順を示すフローチャートを図 7 に、チャート中の各分岐のデータ処理の例を図 8 に示す。提案方式においては、図 7 の分岐 B のように、表作成

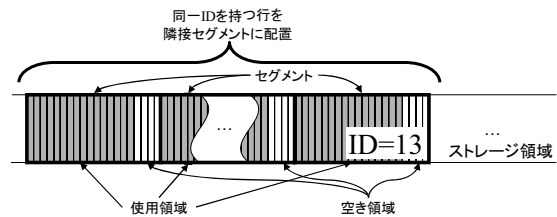


図 5: 同一 ID を持つ行を隣接セグメントに配置

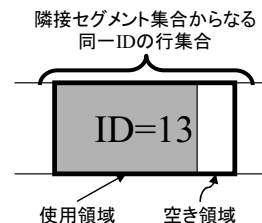


図 6: 図 5 の略記

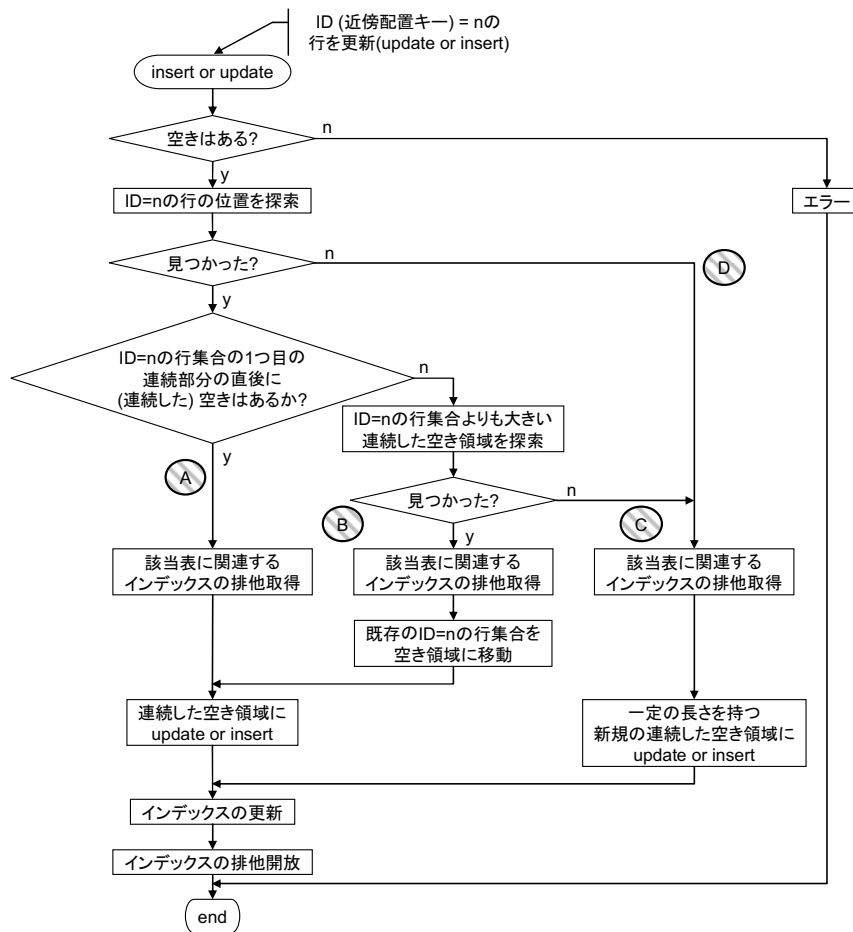


図 7:提案方式の更新および挿入の手順

時の予測以上の更新があった場合にも図 8 のパターン B のように指定した列の値が重複する行の集合をまとめて連続領域に移動させることで連続性を保障できる場合がある。なお、図 7 の例では十分なサイズの空き領域が見つからず分岐 C に分岐した場合には非連続が発生する。しかしこのような場合はまれであり、またその後の更新操作や削除操作に伴い、新たに十分なサ

イズの空き領域が見つかった場合には、分岐 B に分岐した際に再配置が行われ、非連続が解消される。

さらに、従来方式の Z 字順連続配置 (2.1 節, 図 2) を提案方式に応用することも可能である。この場合は図 8 の分岐 B において移動対象となる行集合が Z 字順の近傍全体となる。

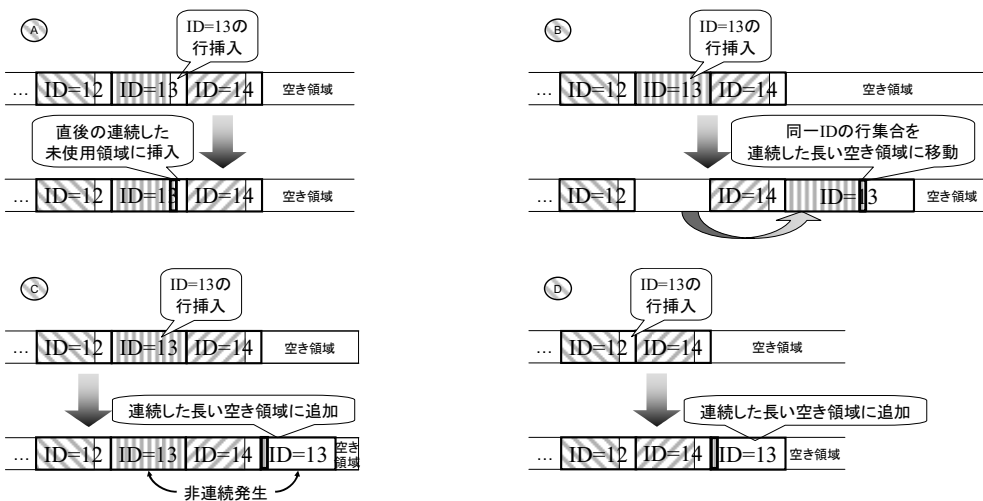


図 8:提案方式の動作例

3.2. 効果見積り

提案した重複 ID 近傍配置表を用い、地図の区画番号を重複 ID とすることで、現在地を含む区画の読み込みや、現在地近傍 n 区画の読み込みに伴うシーク回数が削減され、読み込み処理の所要時間が短縮される事が予想される。ここでは、運用期間を 3 年とし、読み込み処理の時系列変化を見積もる。

見積りの前提を表 1 に示す。表 1 に従い環境構築を行った場合、B-Tree インデックスは 4 段となった。一般に情報端末ではキャッシュメモリを十分に確保できず、数十 KB ~ 百 KB 程度しか使えない。そのため、インデックス部分で 3 回の実 read およびシークが発生すると見積もる。

また、運用期間の満期である 3 年間経過後に想定し

表 1: 見積りの前提

項目		値
HDD の I/O 単価	シーク	20msec
	read	4MB/sec(32Mbits/sec)
想定 I/O パターン	種別	現在地の近傍 1 区画読み込み
	サイズ	100 件/区画(初期値) 280 件/区画(3 年後)
データソース	種別	日本全土の道路情報
	サイズ	428MB,500 万行(初期値) 493MB,576 万行(3 年後)
挿入	頻度	2 ヶ月毎
	サイズ	3.6MB, 4.2 万行(年率 5%)
運用期間		3 年,36 ヶ月(18 回更新)

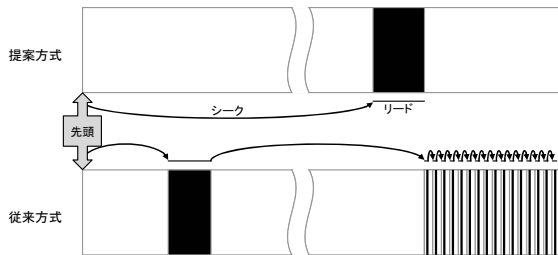


図 9: 3 年後のデータ配置とアクセスパターン

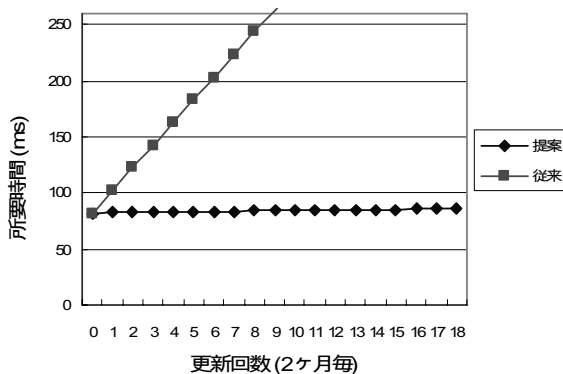


図 10: 現在地近傍 1 区画読み込み処理時間の見積り

た HDD 上のデータ配置とアクセスパターンを図 9 に示す。このように定期的な更新処理によってデータ領域のフラグメントが進行する。最後に表 1 の前提に従い、3 年間運用した場合の現在地近傍 1 区画読み込み処理時間の変化を図 10 に示す。図 10 から、提案方式を用いることで、運用期間中は顕著な性能低下が生じない事が分かる。

以上の机上見積りから、重複 ID 近傍配置表を用いることで、地図データベースを扱う情報端末に DBMS を適用した場合に、運用期間中にメンテナンス処理を行わなくとも更新に伴う参照速度の低下を生じないことを DBMS 側で保証する事が出来る見通しを得た。

4. 関連研究

ディスクのフラグメントによる性能劣化の解消を目指す研究は、データベースの再編成の研究を中心に行われてきた。また、近年においては、再編成処理の所要時間を短縮することを目的とし、フラグメントが激しい箇所のみを再編成する手法が提案され、効果が確認されている [1].

また、動画に代表されるマルチメディアデータを扱うコンシューマ機器においてもディスクのフラグメントによる性能劣化は問題となる。また、フラグメントが生じない場合においてもディスクの内周部と外周部では性能が異なるため、機器の性能設計時には最悪値を用いて設計しなければならない。そこで、マルチメディアデータの I/O パターンに特化した性能劣化の解消法として、ディスクをいくつかのゾーンに分割し、それらへ分割して単一のファイルを配置することで、極端なフラグメントを生じることなく、運用機関中の性能を均一に保つ技術も開発されている [2].

本研究では、モバイル向け地図コンテンツを扱う情報端末における地図更新に着目し、フラグメントが発生しないことを更新時に保証することによって運用期間中の性能を均一に保つ手法を提案したことから、独自性が高いといえる。

5. まとめ

本研究ではカーナビゲーションシステムに代表される従来のモバイル向け地図コンテンツを扱う情報端末の特徴と問題点、および従来の DBMS の特徴と情報端末に適用した際の問題点を示し、従来の DBMS を情報端末に適用した際の問題点を解決する機能である重複 ID 近傍配置表を提案した。重複 ID 近傍配置表は情報端末で用いられるストレージデバイスの I/O 特性および情報端末のアクセスパターンの特徴を前提として、性能上のボトルネックとなっていたシークの回数を削減することが可能である。そのためモバイル向け地図コンテンツを扱う情報端末に重複 ID 近傍配置表を導入した DBMS を適用することで、情報端末の想定ユースケース期間内にメンテナンスを行うことなく更新回数に伴う性能低下を招かない運用を行える。本研究ではストレージデバイスの I/O 単価を用いた見積りによって性能低下を招かない見通しを得た。

文 献

- [1] 合田和生, 喜連川優, “構造劣化の局所性を活かしたデータベース部分再編成の提案,” 電子情報通信学会第17回データ工学ワークショップ 第4回日本データベース学会年次大会 (DEWS2006), 4C-o4, Mar.2006
- [2] 西村章, 佐々剛, 田村哲也, 鈴木一也, “HDDに「AVモード」を実装,” 日経エレクトロニクス, 2005-12-5, pp.113-118, Dec.2005