

3N-6

動的インデックス方式による
図面処理用領域分割型図形データ構造

宮武孝文 松島 整 大谷成子 江尻正員
(株)日立製作所 中央研究所

1. はじめに

近年、図面認識について数多くの研究が行われているが^{1) 2)}、記号や線経路の認識では折線近似で表現した図形データを対象とすることが多い。大量の図形データの中で認識処理を効率良く行うためには、高速に図形データを検索できるようなインデックス機構が必要となる。従来セル法³⁾と呼ばれる多次元点データ管理手法があったが、本稿ではそれを拡張し、線・面のデータが取り扱える領域分割型図形データ構造を提案し、それを適用した検索、追加、削除の方法とこの構造を用いた場合の基本特性を示す。

2. 領域分割型図形データ構造

線や面の図形は2次元的な広がりを持つため、領域分割を行うと個々の図形データは複数の領域(セルと呼ぶ)にまたがる。このため処理の簡素化に、通常は①線や面の重心位置座標値を用いて所属するセルを1つに決めるか、②またがる図形データは特別扱いとして別に記憶する方法がとられる。検索を例にとると、①の場合検索しようとする位置のセル内に重心位置が無い場合検索漏れが生じ、その防止策が必要となる。また②では特別扱いしたデータが常に検索の対象となるので検索速度が上がらない。

そこで本稿の方式では個々の図形が通過または包含するすべてのセル番号をその図形データのインデックスとして持たせるようにした。具体的なデータ構造を図1に示す。本構造は通常の図形データテーブルとセル毎に図形識別子を集合化したインデックステーブルからなる。例では図形識別子として折線(以降セグメントと呼ぶ)毎に番号を付けそれを用いている。検索はセル番号に基づいて行う。

本構造は図形データの検索だけでなく、追加や削除が頻繁な動的な環境に対しても高速性が発揮出来るように工夫がしてあるのが特長である。

3. 検索・追加・削除操作

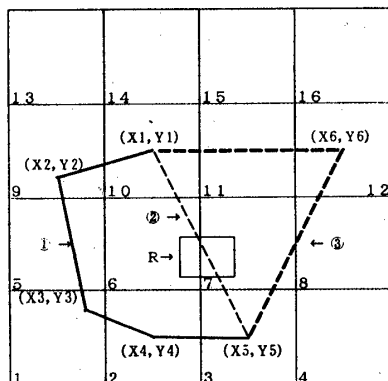
<検索>検索範囲Rに含まれるセグメントの抽出

step1: Rと重なりを持つセルの番号を割り出し、該当セルテーブルから候補セグメントの番号をすべて読み出す。

step2: 読み出したセグメント番号群の内、重複しているセグメントを除く。

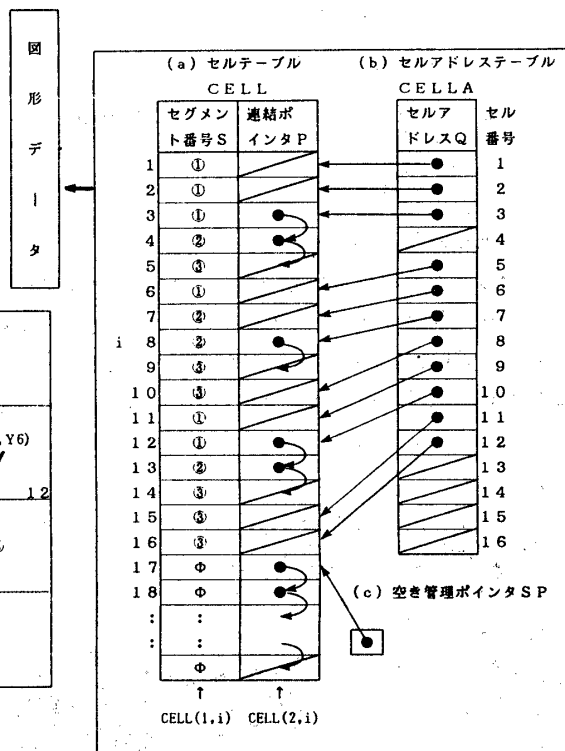
step3: 重複除去後の個々のセグメント番号についてそのセグメントの実際の座標(X,Y)とRとの包含関係を調べ、検索結果を出力する。

セル内の平均セグメント数をFとすると、検索コストはオーダーF; O(F)となる。(但し検索範囲はセルサイズ以下を仮定)



セグメント①:セル番号1, 2, 3, 5, 9, 10
セグメント②:セル番号3, 7, 6, 10
セグメント③:セル番号3, 7, 8, 10, 11, 12

(a) 図形



(b) インデックス

図1 領域分割型図形データ構造

<追加>セグメントの追加とインデックスの変更

セグメント番号: S, セル番号: CELLNOとする。
 step1:セルテーブルへのSの追加; CELL(1,SP)←S
 step2:空きリストの連結ポインタ待避;
 P←CELL(2,SP)
 step3:セルの連結ポインタ更新;
 CELL(2,SP)←CELLA(CELLNO)
 step4:セル毎のリスト先頭アドレス更新;
 CELLA(CELLNO)←SP
 step5:空き管理ポインタ更新; SP←P

1つのデータの追加コストはセグメント数に無関係であるので $O(1)$ と言える。

<削除>セグメントの削除及び不用となったインデックスの空き領域の回収

step1:ポインタ初期化; Pb←0, Pa←CELLA(CELLNO)
 step2:削除カラムサーチ;
 if Pa≠0 then
 if CELL(1,Pa)=S then
 step2.1:削除カラム発見,カラム削除;
 if Pb=0 then
 CELLA(CELLNO)←CELL(2,Pa)
 else
 CELL(2,Pb)←CELL(2,Pa)
 end if
 CELL(2,Pa)←SP
 SP←Pa
 削除処理終了;return
 else
 step2.2:セルテーブルのカラムアドレス更新;
 Pb←Pa, Pa←CELL(2,Pa)
 end if
 go to step2
 end if
 削除対象未発見で終了;return

削除の操作ではセル内でセグメント番号の探索が生じるため、削除コストは $O(F)$ となる。特にこの削除処理では、削除した領域は空き領域リストの先頭に記憶される。次回の追加操作ではこの削除した領域から使われ、メモリが有効に利用される。

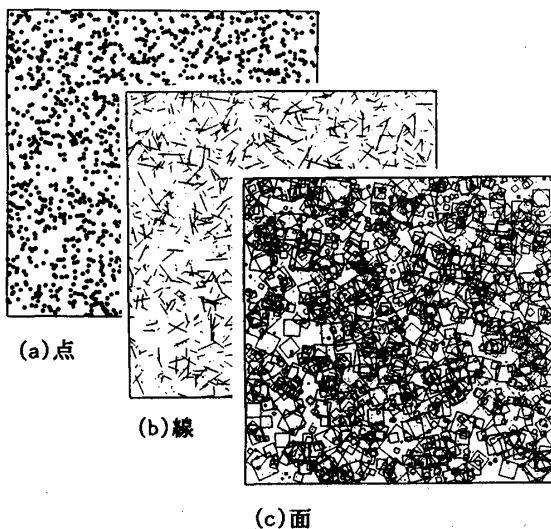


図2 実験用図形データ (点,線,面各2000個)

4. 実験結果

以上述べた領域分割型図形データ構造による検索時間, データ構造化時間, インデックスのメモリ量, 追加時間, 削除時間を, 大型計算機M200Hを用いて調べた。図2は点, 線, 面各2000個を乱数で発生させた実験図形データである。データは領域分割数, 即ちセル数を1から10000まで変化させた。図3はその結果である。検索時間, 削除時間はセル数に比例して短くなるのがわかる。またメモリ量は10000分割で100Kバイト程度である。セル数1の場合のデータは順次操作で処理をした場合とほぼ同じ値であることを考えると, 本提案のデータ構造の検索時間はその1/100程度となっている。

5. おわりに

図面認識を高速かつ効率良く実行するための領域分割型図形データ構造を提案し, 図形データの検索, 追加, 削除の処理が分割されたセル内のデータの数のみに比例する時間で実行出来ることを示した。今後, 図面認識の各種処理に導入し, その高速化を図る。

参考文献

- 1) 棟上, 佐藤: "図面の自動認識と理解", 情報処理, 24, 9, pp.1086-1094(1983)
- 2) 宮武 他2: "平行線抽出手法を用いた地図からの道路情報自動抽出", 信学論(D), J68-D No.2 94(1985)
- 3) Bentley, J.L.: "Data Structures for Range Searching", Computing Surveys, Vol.11, No.4, (1974)

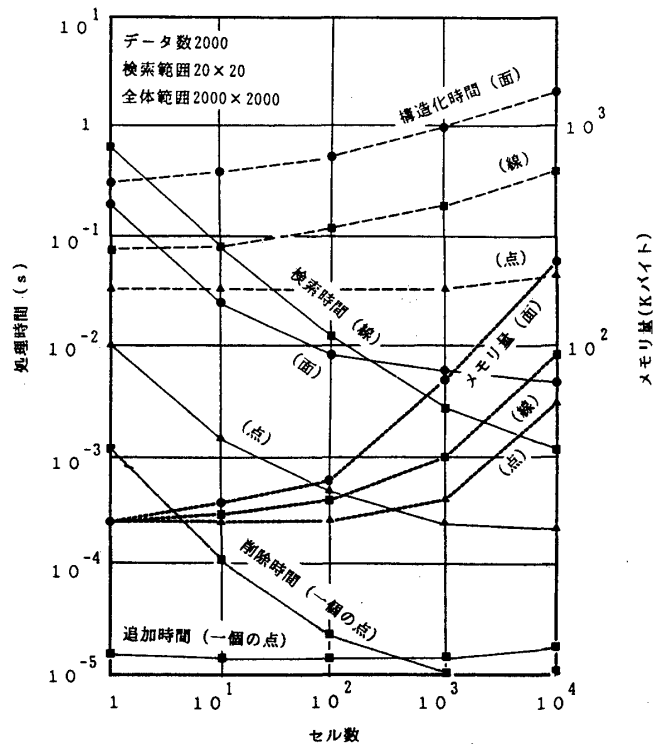


図3 基本特性 (検索, 構造化, 追加, 削除, メモリ量)