

## 利用者インタフェースとしての文字自動配置機能

Automatic Character Placing Function as a User Interface

今井浩 青沼裕美 加藤研児 神代伸彦 上林彌彦

九州大学工学部

### 1. はじめに

文字や数字に加え、図形や画像などをデータベースで扱うマルチメディア・データベースの研究が近年盛んに行なわれている。なぜなら種々の地図や論理回路図、発電所や工場の設備配置図等の図面をデータベースで管理することは、頻繁な更新や検索を容易にし、関連情報（例えば地図では建物の住所や電話番号、設備配置図では機器の定格など）との対応づけもおこなえ、たいへん有用だからである。その中でも特に、地図を扱う地図データベースは応用範囲が広く、利用者からの要求も強いため各方面で研究が進められ、市販されているシステムも多い。しかし地図には文字と図形という異種メディアが含まれるため、それらの相互作用から文字の可読性に問題が生じ、現在のシステムは満足のものとは言い難い。この問題に対処するためには文字自動配置機能が必要である[5]。

筆者らは、前回の当研究会で点図形、線図形、領域図形に対する一通りの文字配置法を述べた。本稿では次の4点について述べる。

- ・領域図形に対する配置法を $O(n^2)$ から $O(n \log n)$ に改善したアルゴリズム。
- ・前回の点図形に対するアルゴリズムは自由度2であったが、自由度を3とした場合の文字の配置可能性問題はNP完全であること。
- ・離散的な自由度でなく、点を含むある大きさの長方形内であれば連続的に配置を許すアルゴリズム。
- ・実際の地理データの点図形に自由度2で配置した出力例と考察。

### 2. 地理データベースと文字自動配置機能

図面中に自動的に文字を配置する機能を文字自動配置機能という。地図、論理回路図、計算機の構成図、各種グラフなどの図面を計算機で処理するために文字自動配置機能は有効であろう。その理由は次の2つに分けられる。

#### <その1. 地図以外の図面>

図面の入力には大変な時間と労力を必要とする。なぜなら図形はシンボル化によってかなり対応できるが、複雑な図面に対して読み易く文字を配置することはいまだ人手で行われているからである。ゆえに文字の情報を個々の図形（例えばAND素子など）の属性情報としてもたせ、自動的に文字を配置できれば入力の手間が軽減される。一旦作成された図面はそのまま記憶してしまえば次回検索、表示のみでよい。

#### <その2. 地図>

地図においては、その1で述べた入力の手間に加えて拡大、縮小に対応しなければならないという問題が起こる。

計算機上で地図を処理することによって拡大/縮小、表示させる要素の選択、幾何的質

問処理など、従来の地図にはなかった柔軟な表示が可能となる。つまり、利用者の要求に合わせた地図を動的に作ることができる。しかしその柔軟性のために、文字と図形の相互作用という新たな問題が起こってしまう。例えばある縮尺の地図を縮小する時に、文字を文字としてでなく、画像やベクトルの形で扱い、図形と同縮尺で拡大／縮小を施すと、読めないほど小さな文字、または画面いっぱいの大きな文字など意味のない表示がなされてしまう。一方、各文字または文字列に固定点を与え、端末のフォントを用いて表現した場合、縮小しても文字の大きさは変わらないために、読みづらいほど文字同士が重なる恐れがある。

このように、文字の可読性（読み易さ）を考慮せずに柔軟な対応のみを実現すると、任意の要求に対してわかりやすい表示を期待できない。この文字可読性問題を解決するためには利用者の要求がくる度に次の操作を行なうことが必要である。

- step1: 要求（縮尺、用途等）に応じて表示データをデータベースから選択。  
 step2: 選択されたデータから白地図中に文字を動的に配置し、配置できなかったデータになんらかの対処を行う（数字で示して枠外に文字表示、印のみ表示しクリックされれば文字表示など）。

その2で述べた地図用の文字自動配置機能の方が実用性が大きく、興味深い。この機能はstep1で選択された文字をできるだけたくさん、またわかりやすく地図中に配置しなければならない。一方、利用者の要求に実時間で応答するためには処理の高速性も要求される。そのため我々は計算幾何学を基礎として文字自動配置機能の検討を行っている。

### 3. 線分の $L_\infty$ 距離Voronoi図を用いた領域への文字配置法

著者らは領域図形の感覚的中央付近に文字列を配置するアルゴリズムを既に示した<sup>[1]</sup>が、今回同じ結果を得るアルゴリズムにVoronoi図を導入し効率を改善した。

#### 3.1 a:b最大長方形と $L_\infty$ 距離Voronoi図

地図中に現われる領域図形(多角形)Rの名称を、領域内の中央付近に配置するために、配置文字列を縦a横bの長方形と考える。これによって縦と横の比がa:bでx軸y軸に平行な辺をもつ長方形(a:b長方形と呼ぶ)のうち、多角形に含まれる最大のものを求めれば、その中心の縦a横bの長方形は、感覚的な「領域の中心付近」に近いと思われる。(図3.1(a))。

a:b最大長方形を求めるために、Rをx軸方向にa/bの割合で縮小する。この多角形をR'とよぶ。

これによって、R中のa:b最大長方形を求める問題は、R'中に含まれる最大正方形を求める問題に帰着される

(図3.1(b))。よって以降では、多角形中の最大空正方形を求める問題に焦点を当て、そのためにR'を構成する各線分を母線とする  $L_\infty$  距離Voronoi図の構成法を検討する。

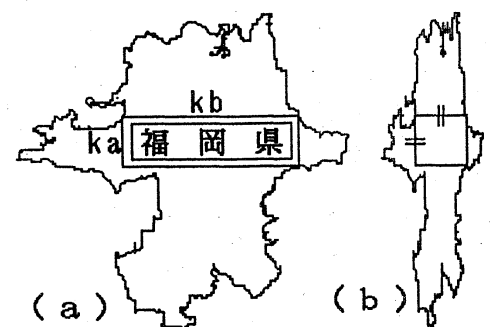


図3.1(a)a:b最大長方形

(b)変換された領域図形

<L<sub>∞</sub> 距離> 点P<sub>i</sub>(x<sub>i</sub>, y<sub>i</sub>)と、点P(x, y)のL<sub>∞</sub> 距離d<sub>∞</sub> は次のように定義される<sup>[4]</sup>.

$$d_{\infty}(P_i, P) = \max(|x_i - x|, |y_i - y|)$$

<線分と点のL<sub>∞</sub> 距離> 線分lと点P(x, y)のL<sub>∞</sub> 距離d<sub>∞</sub> を、次のように定義する.

$$d_{\infty}(l, P) = \min\{d_{\infty}(Q, P)\}$$

ここに、Qはl上の点であり、minは全てのQについてとる.

<線分のL<sub>∞</sub> 距離Voronoi図> 平面上のn本の母線l<sub>i</sub>(i=1, ..., n)が与えられた時、線分l<sub>i</sub>の「勢力圏」V<sub>n</sub>(l<sub>i</sub>)を、

$$V_n(l_i) = \{P \mid \forall j \neq i, d_{\infty}(l_i, P) < d_{\infty}(l_j, P)\}$$

で定義し、V<sub>n</sub>(l<sub>i</sub>)による平面の分割をL<sub>∞</sub> 距離Voronoi図ということにする. また、3つ以上の母線から等距離にある点をVoronoi点という.

この問題では多角形を構成する線分がl<sub>1</sub>, ..., l<sub>n</sub>である.

<Voronoi図と最大空正方形> 3本の線分l<sub>1</sub>, l<sub>2</sub>, l<sub>3</sub>のL<sub>∞</sub> 距離Voronoi図を描いた時、そのVoronoi点をPとする. 定義より、Pは3つの線分にL<sub>∞</sub> 距離で等距離である. その距離をdとすると、Voronoi点を中心とし、3本の線分に接する1辺の長さ2dの正方形が描ける. このとき、1辺2dの正方形は、これら3線分の凸包の中に中心点をもち、内部にどの

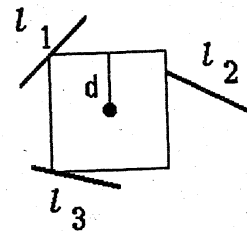


図3.2 Voronoi点と空正方形

l<sub>i</sub>も含まない正方形のうち最大のもの(極大空正方形と呼ぶ)

である(図3.2). また、多角形中の極大な正方形は、多角形を成す3本以上の線分と接する<sup>[1]</sup>ので、極大空正方形の中心は、Voronoi点でないVoronoi辺上には存在しない.

n本の線分を考えると、Voronoi点を成す3線分以外の線分は、Voronoi点に対してl<sub>1</sub>, l<sub>2</sub>, l<sub>3</sub>よりも遠いので、正方形に交差しない. ゆえに、多角形の各辺を母線とするL<sub>∞</sub> 距離Voronoi図を求め、各Voronoi点について極大空正方形を調べれば、多角形に含まれる最大空正方形(最大の極大空正方形)が求められる. 文字配置問題に即していえば、多角形中に含まれるVoronoi図のみ求めればよい.

Voronoi点は、O(n)個(平面グラフのEulerの定理より)なので、Voronoi図を求めた後、最大空正方形は線形時間で求められる.

### 3.2 L<sub>∞</sub> 距離Voronoi図の性質

2本の線分l<sub>1</sub>, l<sub>2</sub>の二等分線B<sub>1,2</sub>は、2本の線分にL<sub>∞</sub> 距離で等距離である点の軌跡である. B<sub>1,2</sub>上の点Pの条件は、Pを頂点とする同じ大きさの正方形が2線分に接することなので、Pは両線分に接する正方形の中心点である(図3.3). 一般にB<sub>i,j</sub>は、両線分の端点から伸ばした傾き±1の直線を境界とする、3本の線分と2本の半直線で成る. n本の線分l<sub>i</sub>(i=1, ..., n)のL<sub>∞</sub> 距離Voronoi図は、B<sub>i,j</sub>(i≠j)の重ね合わせによって求められる(図3.4(b)).

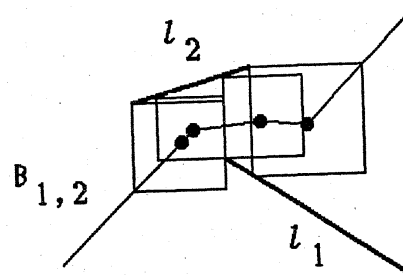


図3.3 二本の線分の二等分線

### 3.3 平面走査による $L_\infty$ 距離Voronoi図作図法の検討

平面走査を用いて  $L_2$  距離Voronoi図を作図するアルゴリズム<sup>[3]</sup>の、 $L_\infty$  距離Voronoi図への適用を以下で検討する。

母線  $l_i$  の2 端点の  $y$  座標のうち、小さい方を  $y_i^-$  とすると、 $y_i^-$  の昇順で母線  $l_i$  ( $i=1, \dots, n$ ) を並べる。平面上の各点  $(x, y)$  に以下の写像\*

$$(x, y)^* = (x, y + d_\infty(x, y))$$

( $d_\infty(x, y)$  は点  $(x, y)$  から最も近い母線までの  $L_\infty$  距離) を行い、変換された  $B_{i,j}$  である  $B^*_{i,j}$  を用いてVoronoi点を求める。 $B_{i,j}$  は、5本の線分  $B^*_{i,j}$  に写される。母線は母線に、母線  $l_i$  のVoronoi領域  $R(l_i)$  は  $l_i$  の下端点を最下点とする領域  $R^*(l_i)$  に写される。変換された平面を下から上に向かって水平直線で走査してゆき新たな母線  $l_i$  をヒットしたら  $l_i$  の隣の母線との二等分線を考慮にいれ、二等分線の交点 (Voronoi点の候補) を予測しておく。走査しながら予測したVoronoi点の候補がVoronoi点であるか否かを調査していく。このようにして作られるグラフを  $V^*$  とすると、 $V^*$  の位相構造はVoronoi図と同じである。この手法により、 $O(n \log n)$  の手間でVoronoi図が求められる。

図3.4(a)は、線分  $l_1, l_2, l_3$  と  $B^*$ 、図3.4(b)の実線はVoronoi図を示している。

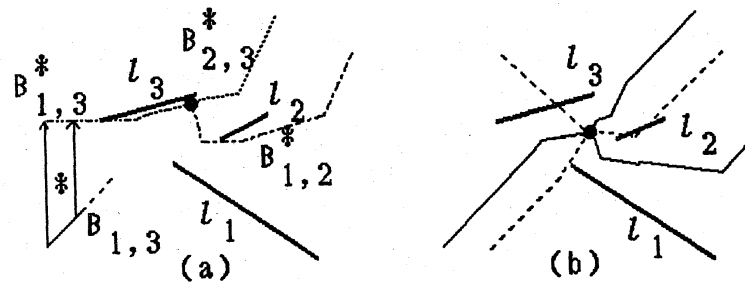


図3.4  $l_1, l_2, l_3$  の  $B^*$  と Voronoi 図

### 4. 2 または 3 自由度の文字配置問題の難しさ

地図中の点図形に地名などの文字列を配置するとき、その位置の候補は本来は連続的に無限に存在するが、高速化のために候補を多自由度で離散化し、さらに1つの部分図形への候補を点の右と左など2つに限って自由度2の文字配置を仮定する。すると、文字どうしの重なりを避けて点図形の全てに候補のどちらかに配置が可能であるか否かを、交差グラフを用いて判定できる。この判定は点図形の数を  $n$  とすれば、 $O(n(\log n)^2)$  の手間で解けることを青沼、今井[1]らは示した。

図面上に  $M$  個の施設点 (点図形) があるとする。それらの施設点の中の  $P$  ( $\leq M$ ) 個に対して文字列を配置できるかどうかを判定する問題を配置数判定問題といい、中でも  $P=M$  の特殊な場合を配置可能性判定問題ということにする。以下では上述の限定された問題について、自由度2の場合の配置数判定問題及び自由度3の場合の配置可能性判定問題がNP完全であることを示す。

#### 4. 1 基本的事項

指定された自由度2あるいは3の施設点を、2あるいは3節点の部分グラフで表す。配置候補を節点で表し、配置候補の重複関係を枝で表した交差グラフを構成する(図4.1)。ただし、他の施設点と重複するような配置領域には配置不能とする。また、前処理として他の施設点の配置領域と重複しない配置領域には、文字列を強制的に配置するようにする。

浅野<sup>[2]</sup>は平面上の様々なオブジェクト(単位正方形, 線分等)によって構成される交差グラフの独立点集合を求める問題がNP完全であることを示した。その証明方法の概略は次のようなものである。

まず、NP完全問題である平面3充足性問題を、ある制約された平面グラフの最大独立点集合を求める問題に帰着する。次にそのグラフを平面上の様々なオブジェクトによる交差グラフに帰着する。ここで、制約された平面グラフは、偶数個の節点を持つ循環グラフを図4.3のような3個の節点による部分グラフで結合した形で表される。この部分グラフは3つの循環グラフと交差する節点a, b, cがすべて真である(独立点である)ときのみ独立点を持たず、その他の場合は1つの独立点を持つという性質がある。

本稿ではこの考え方を応用し、ステップ1)で帰着されたグラフが、文字配置問題の交差グラフに帰着できることを示す。自由度2の場合には点に接する任意の2つの領域が配置候補として可能であり、文字列の長さも任意である(配置領域が任意の長さの長方形に対応)場合について考察することにし、自由度3の場合には図4.1aのように配置候補の位置が指定され、また配置する文字列は1文字である(配置領域が正方形に対応)場合について考察することにする。

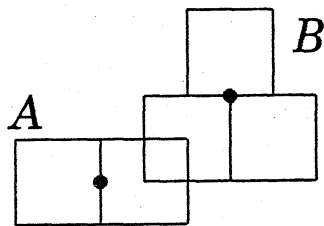
#### 4. 2 文字配置問題への変換方法

このグラフは自由度3の場合の文字配置問題の交差グラフに、次のようにして帰着できる。1)偶数個( $2k_i$ 個とする)の節点による循環グラフ( $n$ 個あるとする)は $k_i$ 個の2自由度の施設点による $n$ 個の循環した交差グラフに変換する。2)自由度の施設点は、実際には図4.2のように2つの自由度3の施設点で表す。これは文献<sup>[1]</sup>の前処理の後には文字列が配置された施設点と、自由度2の施設点となる。2)図4.3の部分グラフ( $m$ 個あるとする)は $m$ 個の自由度3の施設点に変換する(図4.4)。このように変換すると、平面3充足性問題は、 $K+m$ 個の文字配置を求める問題となる。

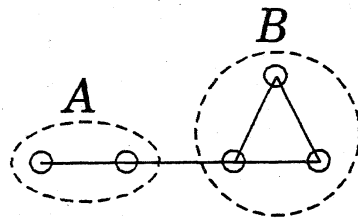
ただし、 $K = \sum_{i=1}^n k_i$ である。この場合、施設点は合計 $K+m$ 個あるので、問題は自由度3の配置可能性判定問題に帰着される。

また、自由度2の場合の文字配置問題には次のように帰着する。循環グラフの部分は自由度3の場合と同様に変換する。図4.3の部分グラフ( $m$ 個あるとする)はそのままでは変換できないので、図4.5のように変換する。この部分グラフはa, b, cがすべて真のときのみ4個の独立点を持ち、それ以外の場合は5個の独立点を持つ。このように変換すると、平面3充足性問題は $K+5m$ 個の文字配置を求める問題となる。この場合施設点は $K+6m (>K+5m)$ 個あるので、この問題は自由度2の配置数判定問題に帰着される。

以上より自由度2の配置数判定問題および自由度3の配置可能性判定する問題がNP完全であることが証明された。



a) 重複関係



b) 交差グラフ

図4.1 交差グラフの例 (A, Bは自由度2, 3の施設点及びその配置候補を表している)

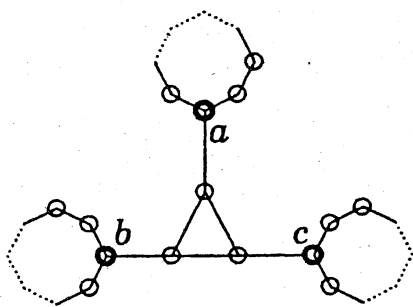


図4.3 結合部分のグラフ

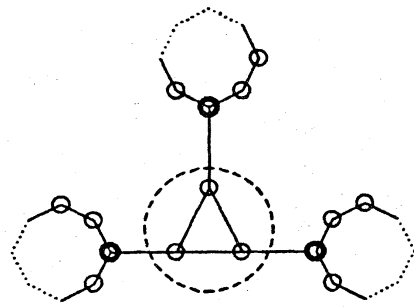


図4.4 自由度3以下の文字配置問題への変換

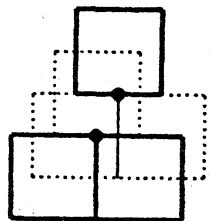
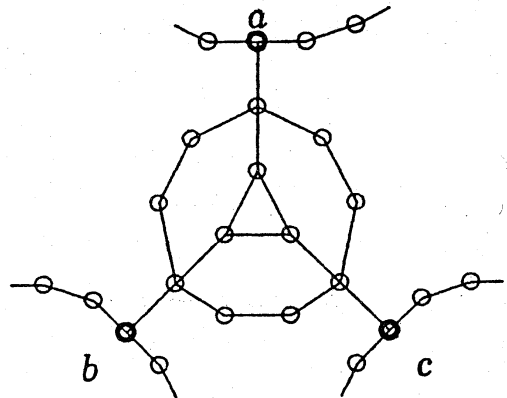
図4.2 自由度2の施設点の  
表し方 (点線は消去される領域)

図4.5 自由度2の文字配置問題への変換

## 5. 配置の許容領域を設定するアルゴリズムについて

4章では自由度2, 3の配置法についてその問題の複雑さを明らかにし, 離散的な配置の候補を設定すると自由度2の配置可能性判定問題が多項式アルゴリズムの限界であることを述べた. 5章では精密な配置が必要である場合に, 各点図形に対して点図形を含むある大きさの長方形を仮定し, 他の長方形と重ならない領域を求め, その領域内であれば文字列を配置してよい, という連続的な配置の候補をもつアルゴリズムについて述べる.

### 5.1 問題の定式化

各点データに対する文字は、それを隙間なく囲む長方形（固有長方形と呼ぶ）で代表する。すると、点データに対する文字配置問題は、各点の近くにその点に対応する長方形を配置する問題となり、異なる点に対応する長方形がそれと重ならないことが制約条件となる。各点データに対し、その点を中心とし、その点の固有長方形の縦横約2倍のサイズの長方形を考え、その中に固有長方形を置くことにする。各長方形で他の長方形と交わっていない領域をその専有領域と定めると、固有長方形が専有領域内に配置できるならそこに置くことによりこの点データを他と独立に扱うことができる（図5.1）。また、固有長方形を専有領域内に配置できなかった残りの点データに対しては、この方法を前処理として用いて、対象とする点データの数を減らした後に文献<sup>[1]</sup>の方法を適用するか、あるいは、長方形が相互に重なっているような点データ集合（連結成分という）を求めて処理することにする。

以下では、 $n$ 個の点データの長方形に対してその専有領域を求め、その中に固有長方形が配置できるかどうかを判定する問題に対して、平面走査法を用いたアルゴリズムを与える。

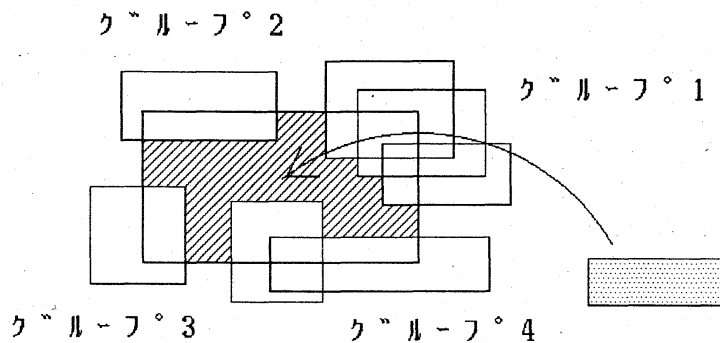


図5.1 専有領域の計算と長方形の配置判定、及び連結成分分解

### 5.2 専有領域の計算

専有領域の計算は平面走査法を用いて次のように行う。走査線は水平線とし、上から下へ平面上の $n$ 個の長方形を走査する。平面走査の際、走査線と長方形の左辺、右辺との交点を左から右へ並べた順序で2分探索木で管理する。各時点で走査線はそれらの交点により区間に分けられ、各区間ではそれを含む長方形の数が唯一つ定まる。この数を各区間毎にカウンタに覚えておく（図5.2）。カウンタの値が1となる区間は、ある長方形の専有領域に含まれており、そのような区間を平面走査して全て求めることにより、専有領域を求めることができる。このとき、カウンタの値は走査線が停止する度に更新するが、走査線がある長方形の上辺（下辺）と重なったとき、走査線上でのその辺の区間に含まれる区間のカウンタを+1(-1)すればよい。このとき更新にかかる手間は、その長方形と交わる長方形の個数に比例する。このことより、全体でのこのアルゴリズムの手間は、交差する長方形対の数を $k$ としたとき、 $O(n \log n + k)$ であることがわかる。

### 5.3 長方形の配置判定

5.2節で求めた  $n$  個の専有領域の中の1つに着目し、その固有長方形（以下単に長方形と記す）のサイズを縦  $a$  横  $b$  として専有領域内に配置できるかどうかを調べる。

予め水平線分には、それが専有領域（以下単に領域と記す）を構成する上辺であるか下辺であるかラベル付けをしておく。走査は  $x$  軸に平行な間隔  $a$  の2本の直線で上から下へ行なう。下側の走査線を  $h_B$ 、上側の走査線を  $h_T$  とする。もし領域内に長方形が配置できる場所があるならば、そのスペースをそのまま領域の下辺に接するまで下方へ平行移動させることができる。従って下辺の一部または全部が領域の下辺に接している状態での長方形の配置を考えればよい。

$h_B$  が、水平区間が  $[x_L, x_R]$  の下辺と重なった状態を考える。領域を  $h_B$  と  $h_T$  ではさまれた部分に制限し、 $h_B$  の各点で、垂直上方に最初に見えるもの（その点が領域内部なら領域の上辺か  $h_T$ ；外部なら何も見えない）に着目すると、 $h_B$  を同じものが見える区間に分割できる。これを平行ビューと呼ぶ（図5.3）。この平行ビューがあれば、水平区間  $[x_L - b, x_R + b]$  と  $h_T$  が見えている平行ビューの区間との共通部分で、幅  $b$  以上のものであるかないかでその下辺に接するところに長方形を配置できるか否かを判定できる。

平行ビューを、その区間の区切りの点を左から右へ並べたリストで表わすと、平面走査法の中で走査線が水平辺のうち上辺と重なったときにはその辺の端点をリストに挿入し、下辺と重なったときにはその辺の区間内に含まれる点をリストから削除しながら配置判定を行うことにより管理できる。領域の各点は、このリストに1度挿入・削除され、他には高々1回アクセスされるだけであるので、挿入するときの位置さえわかれば全体で領域の辺数に比例する手間で行える。

以上は、1つの専有領域に着目したときのアルゴリズムであったが、これを3節での専有領域を求める平面走査と同時に全体でまとめて行なう。すると、各専有領域の辺数の総和が  $O(n+k)$  なので、リストでの挿入位置がわかれば後の操作は全体で  $O(n+k)$  の手間で行える。挿入位置を求めることは、全体で  $O((n+k) \log n)$  の手間で行える。結局、 $O((n+k) \log n)$  の手間ですべての配置判定まで行えることがわかる。

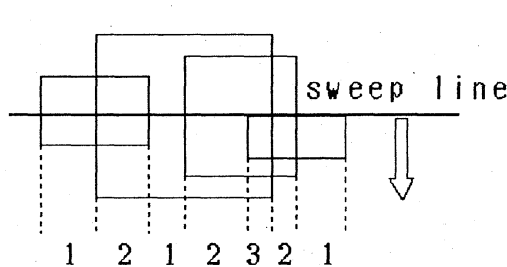


図5.2 カウンタの設定

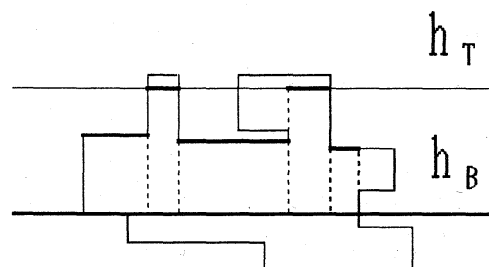


図5.3 平行ビュー

## 6. 自由度を2とした文字配置例

現在構築中の地理データベースについて、自由度を2とした素朴なアルゴリズムを用いた配置法の出力例とその考察を述べる。

このプログラムでは、利用者が出力したい情報（公共施設の位置および名称）を要求す



と、データベースから必要なデータを検索し、海岸線・行政界とともに表示する。文字の表示位置は点の右側を主とし、右に置けないものは左側に置く。文字が配置できないデータで数字なら配置できるものは、文字と同様に左右どちらかに数字を置き、図面の外に数字とともに名称を列挙する。文字も数字も置けないデータはその位置に印のみ表示する。



図6.1 自由度1の文字配置出力



- 1 下山門小学校
- 2 長丘小学校
- 3 大野小学校
- 4 大野東小学校
- 5 下大利小学校
- 6 香椎東小学校

図6.2 自由度2の文字配置出力

図6.1は自由度が1，すなわち文字を右にしか置かない場合の文字配置出力例で，図6.2は自由度が2の出力例である。データはどちらも同じもので，福岡市内（国土数値情報におけるメッシュコードが503022, 503023, 503032, 503033）の小学校を表示させている。

2枚の図の評価を行なうため，自由度を1増やしたときの改善率 $\rho$ を次式で定義する。

$$\rho = (d_i - d_{i-1}) / d_{i-1}$$

ただし， $d_i$ は自由度 $i$ のときの表示データ数で， $i \geq 2$ とする。

また，付加した自由度に従って新しく配置されたデータの数を，改善度と呼び $\lambda$ で表わす。

この $\rho$ と $\lambda$ を用いて左側にも文字を配置することを許した効果を調べると，このデータの場合，図6.1に文字が表示されているデータ数は55で図6.2では60であるから，自由度を1から2に上げたときの改善率および改善度は $\rho = 9\%$ ， $\lambda = 12$ であり，明らかに改善されていることがわかる。しかし，自由度を3にしたと仮定すると，図6.2から判断してあまり効果は期待できない。したがって，自由度2による配置でかなり満足のいく表示ができるものと思われる。

## 7. むすび

本稿では，これまでに開発した文字配置アルゴリズムを発展させ問題の難しさの解析，計算複雑度の改良，異なる配置の方針，実際の地図データを用いた出力例からの考察を述べた。今後は速さと美しさ，理論と実際の両側面から研究を進める必要がある。

## 謝辞

本研究で用いている国土数値情報を提供頂いた建設省国土地理院に感謝致します。尚，本研究の一部は文部省科研費による。

## 参考文献

- [1] Hiromi AONUMA, Hiroshi IMAI, Yahiko KAMBAYASHI: A Visual System of Placing Characters Appropriately in Multimedia Map Databases. IFIP TC-2 Working Conference on Visual Database Systems, April, 1989.
- [2] Tetsuo Asano : Difficulty of the Maximum Independent Set Problem on Intersection Graphs of Geometric Objects. 電子情報通信学会技術研究報告 COMP88-19.
- [3] S. Fortune: A Sweep-line Algorithm for Voronoi Diagram. Proc. 2nd ACM, Symp. on Comp. Geom, pp. 313-322, 1986.
- [4] 伊理正夫監修，腰塚武志編：計算幾何学と地理情報処理。bit別冊，共立出版，東京，1986.
- [5] 上林彌彦，有川正俊：図形データにおける概視機能とその実現。情報処理学会研究報告，86-DB-54，1987.