7J-4

属性つきの境界点列からの図形復元手法

深田陽司

ＡＴＲ視聴覚機構研究所

1．まえがき

二値図形の復元はその応用の広さから古くより数多くの研究がなされてきたが、これまでの多くの手法は、連続する境界点の局所状況しかみていないため複雑であったり、図形のホール部分の情報が壊されてしまうとか、点列の他に図形の内部点を指示しなければならない等の問題を抱えている(1),(2)(3)。本稿では、境界点列の並びを大局的にみることにより、シンプルで見通しのよい復元手法を述べる。

2．入力情報と出力情報

本稿では入力として図形領域の境界点列Ｓ＝Ｐ1 Ｐ2 --Ｐn Ｐn+1, Ｐi=（ｘi,ｙi）とその属性が与えられるとする。ここで、Ｐn+1=Ｐ1 であり、属性とは種類（外周／内周）と方向（反時計／時計）を示す。境界点列Ｓ（ n >=2 ）に対して以下の条件が成立しているとする。

1）連続する二点Ｐi とＰi+1 は隣接する。つまり

$|x_{i+1} - x_i| <= 1$ & $|y_{i+1} - y_i| <= 1$

2）連続する二点Ｐi とＰi+1 は同一点ではない。

3）境界点列は互いに接することはあっても交差することはない。

点列が与えられた時、境界を含めた領域の内部をラスタグラフィックス的に塗りつぶすための情報が出力情報である。つまり、図形領域を水平のランデータで表現した場合のランの左端点と右端点の座標値のペアである。

ペアを求めるには左端点リスト（l-list）と右端点リスト（r-list）を用意しておいて、Ｐi=（ｘi,ｙi）が左（右）端点と判定された時、l-list（r-list）のｙi 行に関するリストにそれまでに格納されているｘ座標との間でｘi を昇順に並べて格納していく。左右端点（左端点かつ右端点）の場合は l-list と r-list の双方にそれぞれ昇順に格納する。その後、l-list と r-list との間で対応する要素を並んでいる順にペアリングすればよい。

3．三連続点の並び方

点列が外周を反時計方向に追跡しているとすると、進行方向の左手に領域を見ており、領域の左側では下方に、右側では上方に向かっている。それ故、画像の左上角を原点にとると、ｙについて正変化している点はランの左端であり、負変化している点は右端であるので、基本的にはこれらの点を l-list, r-list に格納していけばよい。しかしディジタル図形では境界点の全てがランの端点となる訳ではないので、各境界点について端点か内部点かを注目点の前後の状況から判定しなければならない。まず基本的な状況である、注目点とその直近の前後の点からなる三点の並び方を調べる。Ｐi への入口変化（ ystep1=ｙi －ｙi-1 ）、出口変化（ ystep2=ｙi+1 －ｙi ）は条件１より各々±１、０の３通りあり、条件２より同一点が連続することはないから、存在しうる並び方は９種類ある。

4．アルゴリズム

本稿では属性として外周反時計方向と内周時計方向のみを考える。他の属性については同様に記述できる。

点列は共に領域を左手に見ながら境界を追跡しているので、ystep が一様に変化 down（up）しているケースではＰi の左（右）隣点は背景に属する点であるから、Ｐi はランの左（右）端点である。まとめると、

Ｓ１） ystep1 = ystep2 = ＋1 の時

Ｐi を l-list に格納

Ｓ２） ystep1 = ystep2 = －1 の時

Ｐi を r-list に格納

次に図１に示すような一点極値（ turn ）のケースを考える。同図においてｘ変化は

xstep = ｘi+1 －ｘi

で定義されている。左手に領域を見ているので、ハッチ側が図形である。図から分かるように、○印をつけた境界点はランの左右端点であるが、Ｘ印はランの内部点である。これらのケースをまとめると、

Ｓ３）ystep1+ystep2 = 0 & xstep≠ystep2 の時

Ｐi を l-list と r-list に格納

A Region Filling Method Represented by the Sequences of Border Points

Youji FUKADA

ATR Auditory and Visual Perception Res. Lab.

9種類のうちの残りのケースは、図2にあげた図例の局所状況として出現する。図2はy方向の変化が0である境界点が2点以上連続（以後 flat と呼ぶ）して、 flat が極値や停滞を構成する例を示している。局所的に見る場合でも、その先がどうなっているかによって、Pi が端点か内部点かを判定することは可能であるが、場合分けが細かくなりすぎて手法としては複雑になってしまう。そこでこれらのケースについては、図2の如く大局的な状況により判定する。

flat はPk からPm （ k < m ）の点列とすると、入口変化と出口変化は次のように定義される。

$$ystep1 = y_k - y_{k-1}, \quad ystep2 = y_{m+1} - y_m$$

更に、 flat のうちx方向に対して最小値、最大値をとる点をPi, Pj とすると、 flat には2点以上存在するから、 $i \neq j$ で $k \leqq i, j \leqq m$ である。x変化は次のように定義する。

$$xstep = \begin{cases} 1 & i < j \\ -1 & i > j \end{cases}$$

すると、 flat の点のうちで注目すべき点であるPi とPj がランの端点かどうかは図2の場合分けにより簡単に判定できる。まとめると、

S4） ystep1+ystep2

$$= \begin{cases} 0 \ \& \ xstep \neq ystep2 \text{ の時} & \\ \quad \text{Pi を l-list, Pj を r-list に格納} & \\ 2 \quad \text{Pi を l-list に格納} & \\ -2 \quad \text{Pj を r-list に格納} & \end{cases}$$

先ほど一点極値を別に考えたが、 flat における極値の一例（Pi とPj が同一点）であり、端点かどうかの判定S3）は、x変化に対する定義を次のようにすると、S4）と全く同じになる。

$$xstep = \begin{cases} x_{i+1} - x_i & i = j \\ 1 & i < j \\ -1 & i > j \end{cases}$$

5．実験

手続は VAX11/730 Pascal で実現し、図3の例に対して適用した。前節では一つの閉領域に対する手法しか述べていないが、アルゴリズムとしては一枚の画像を表す点列群に対処するものを開発している。点列集合としてはS＝（S3, S4, S1, S6, S5）を与え、その属性は外周・内周ともに時計方向を与えた。処理結果は例えばS1 に対しては、0.48秒であった。各領域のペアを求める時間は（S1, S3, S6）、（S2, S4）、（S5）のそれぞれに対して .60, .32, .05 秒である。点列集合S（総点数は 1010 ）から最終的に一枚の画像を復元するまでの全処理時間は 2.15 秒である。

6．あとがき

属性つきの点列から領域を埋めるアルゴリズムを述べた。本稿では点の並びを大局的にみることにより、シンプルで見通しのよい手法を実現している。又、点列を一度しか走査しないので高速である。

謝辞：本研究を進めるにあたり貴重な助言を頂いた当研究所淀川社長に感謝します。

参考文献

(1) 岩田、安居院：” 閉曲線図形の内外部識別方法”、信学論、J63-D, 11, pp. 946-953, 1980.
(2) T.Pavlidis: "Filling algorithms for raster graphics", CGIP, 10, 2, pp. 126-141, 1979.
(3) 末永康仁：” 連結領域のぬりつぶし及び番号づけに関する一考察”、信学技報、IE78-10, 1978.

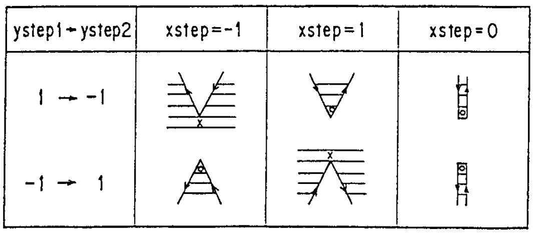

図1． 一点極値

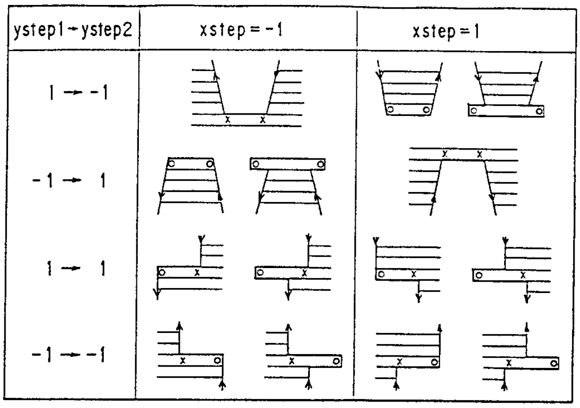

図2． 平坦部

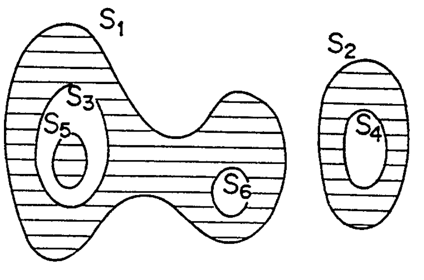

図3． 二値図形