領域塗り手続きを用いた図形の構造化

大久保　明　伸*

A Method To Make Data Structures for Shape Using FloodFill Procedure

Akinobu OOKUBO

Abstract

This paper presents a method of data structures for bitmap shape. To make data struture, floodfill precedure is used. This floodfill method has ability to paint in a shape and get a data structure (edge points, area, center point and etc). This data structure is S-expression of Lisp and is adapted to AI system, expert system or data base for image processing. As algorithm of this procedure is described by Pascal, it is able to translate to the other structure language.

1．はじめに

パーソナルコンピュータの発達により，従来大型の計算機でしか実現できなかったいろいろなデータ処理がパソコン上で可能になってきた．特に画像処理やそのデータベース化は従来大型のシステムを前提にして開発されてきた．数値データや記号データのほかに積極的に画像データが利用されるOAやFAシステム等では小型の計算機(EWSやパソコン)を現在では，中心にして構成されている．画像処理技術は，大きく分けて蓄積や伝送のための圧縮や符号化技術，画像理解のための認識技術，これらの前処理のための雑音処理や強調処理の技術等に分けられる．これらの技術に共通している考え方は次のようになる．画像を点や線の単なる集合と考えるより，画像をひとつのデータ構造としてとらえる方が大域的な処理や局所的な処理のいずれにも有利なことが多い．最近ではこのような観点にたって，これらの研究が盛んにおこなわれるようになってきた[1][2][3]．また構造化された画像はデータベースやエキスパートシステムの対して比較的適応しやすくなる．

本稿では，パーソナルコンピュータ上に画像データベースを構築する目的で画像データの基本的な処理技法とデータ構造に関するプログラム技法について報告する．また今回対象にする画像はイメージスキャナや簡易CCDカメラより得られる2値画像を前提にしたものを処理の対象にする．処理プログラムは，パソコン上で動作するターボパスカルで記述してユニット化した．2章では領域塗りによる画像データの分離とそれを用いた実行例について述べる．3章では画像データの構造化のために領域塗りの手続きを利用して，LISP処理系で使用されるS式の生成法とその具体例について述べる．

2．領域塗りによる連結成分の抽出

パソコン上で動作する各種のソフトウェアの基本的な図形処理関数の中には，領域塗りの機能はあるが，通常は単に与えられた領域を塗りつぶすだけである．しかし，この機能を利用すればもっと多くの画像情報を同時に得ることができる．例えば，連結成分の面積，中心，外接箱，輪郭線および図形同士の包合関係などの情報を得る機能を埋め込むことができる．以下にこれらの情報を獲

*宇部工業高等専門学校電気工学科

得する技報について述べる.

2.1 領域塗りの基本アルゴリズム

領域とは8近傍により同一の色（または色の集合）により連結した領域をさす．8近傍領域塗りのための再帰アルゴリズムは手続き1に示すようなものである．ただし,パソコンではスタック領域の確保量が64KB程度であるのでこの再帰アルゴリズムでは大きな領域を塗ることは出来ない．scolorは定義した領域の色（または色の集合）で，ocolorは塗りかえる色である．またこの手続きの初期座標（x，y）の色はscolorである．初期座標（x，y)を発見して，その8近傍を検査して同色の点を再帰的に塗りつぶしていく.

このアルゴリズムの欠点は，手続き呼び出しによりただ1ドットしか塗りつぶせないし，再帰呼び出しのための局所変数の確保領域が大きくなる．また，連結領域が大きい場合に再帰のレベルが深くなりすぎることである.

そこで，手続き2に非再帰プログラムとして実現したものを示す．この手続きの基本は点（x，y）をocolorに置き換える代わりに点(x,y)とおるx軸に平行なocolorの直線で換えることである．その後その直線上の上下1ドットの範囲を調べ,scolorとscolorではない境界を発見して，scolorの座標をスタックに積む．次にスタックに積まれた座標に対して同様のことを行う．スタックが空であれば終わりである.手続きgetbd(x1, xr, y, scolor)は（x1，y）から（xr，y）の範囲でscolorとそれ以外の色になる境界を求めて,スタックにその座標を格納する.この領域塗り手続きを拡張して，塗りつぶした領域に外接する箱領域を返す手続きflood-fill-getboxと領域の重心と面積を返す手続きflood-fillを製作した.外接箱の獲得は塗りつぶしたx,y座標の最大値と最小値を獲得することと同じである．また，重心と面積は簡単な演算であるので省略する.

```
procedure floodfill  (x，y：座標；
                      scolor：色の集合；
                      ocolor：ぬりつぶし色)；
begin
  if getpixel (x,y) in scolor then
    begin
      putpixel (x,y, ocolor);
      floodfill (x+1,y    , scolor, ocolor);
      floodfill (x+1,y+1, scolor, ocolor);
      floodfill (x    ,y+1, scolor, ocolor);
      floodfill (x-1,y+1, scolor, ocolor);

      floodfill (x-1,y    , scolor, ocolor);
      floodfill (x-1,y-1, scolor, ocolor);
      floodfill (x    ,y-1, scolor, ocolor);
      floodfill (x+1,y-1, scolor, ocolor);
    end
  end；
```

手続き1　再帰による8近傍領域塗り手続き

2.2 領域塗りを利用した輪郭線の抽出

領域塗り手続きを利用して，その連結成分の輪郭線を獲得することが出来る．連結成分の形状をビットパターンで記憶するより輪郭線として記憶する方が有利である.

```
procedure flood-fill-simple  (x，y：座標；
                              scolor：色の集合；
                              ocolor：塗りつぶし色)；
var x1, xr：(x,y)を通るの直線の左右の座標；
            {(x1,y)から(xr,y)までocolorにする。}
begin
  if getoixel (x,y) in scolor then
    begin
      sp=0；{スタックの初期化}
      push (x,y)；
      while sp <>0 do
      begin
      pop (x,y)；
      x1：=x-1；while getpixel (x1,y)=scolor do
      x1：=x1-1
      xr1：=x+1；while getpixel (xr,y)=scolor do
      xr：=xr+1
      line (x1+1,y,xr-1,y)；
      getbd (x1,xr,y-1,scolor)；
      getbd (x1,xr,y+1,scolor)；
      end；
  end；
```

手続き2　手続き1の非再帰手続き

この手続きの原理は

(1) 点(x, y)の4近傍がscolor(色の集合)かocolorでない場合，輪郭線の1点である．

(2) (1)の条件を満たす点集合を再びscolorで描く．

(3) この点集合の1点から再び領域塗りをおこない，新しい点集合を獲得する．これが連結成分の1つの輪郭線である．

(4) (3)の操作を繰り返し行い，全ての輪郭線を獲得する．

連結成分の輪郭線郡の点集合を獲得する(2)の手続きをget-pointsとし，手続き3にまとめておく．
ただし，pointとshapeの型は図2に示す．

プログラム3の手続きを使用して，輪郭線の集合を獲得する．この手続きはコンピュータの画像面を作業領域に使用するために15色のなかの2色を作業色として予約しておく．

3．画像への構造化への適用

領域塗り手続きを利用して，画像の構造化の例を示す．画像の構造形式はLispのS式とした．ある1つの図形(個体)のS式は次の様に表現される．

((外接箱の座標，重心の座標，面積，復元するための座標，自分自身の色，(輪郭線の座標郡))，(包含関係にある個体のリスト))

である．(包含関係にある個体のリスト)の各要素は，個体である．したがって，画像構造は再帰構造で表わすことになる．CRTディスプレイ上に3つの個体がある場合のデータ構造はつぎの様になる．

((0, 0, 639, 399), (399, 199), 256000, (1, 1), 0,
((0, 0, 639, 0, 639, 399, 399, 0)),
(SHAPE1, SHAPE2, SHAPE3))

ただし，SHAPE1, SHAPE2, SHAPE3もまた同様の構造を持ったS式である．このS式で作られたトップレベルのS式はディスプレイを表わす．輪郭線のリストのCDR部がnilの場合は輪郭線は一つである．最も単純な図形のS式は最後がnilである．

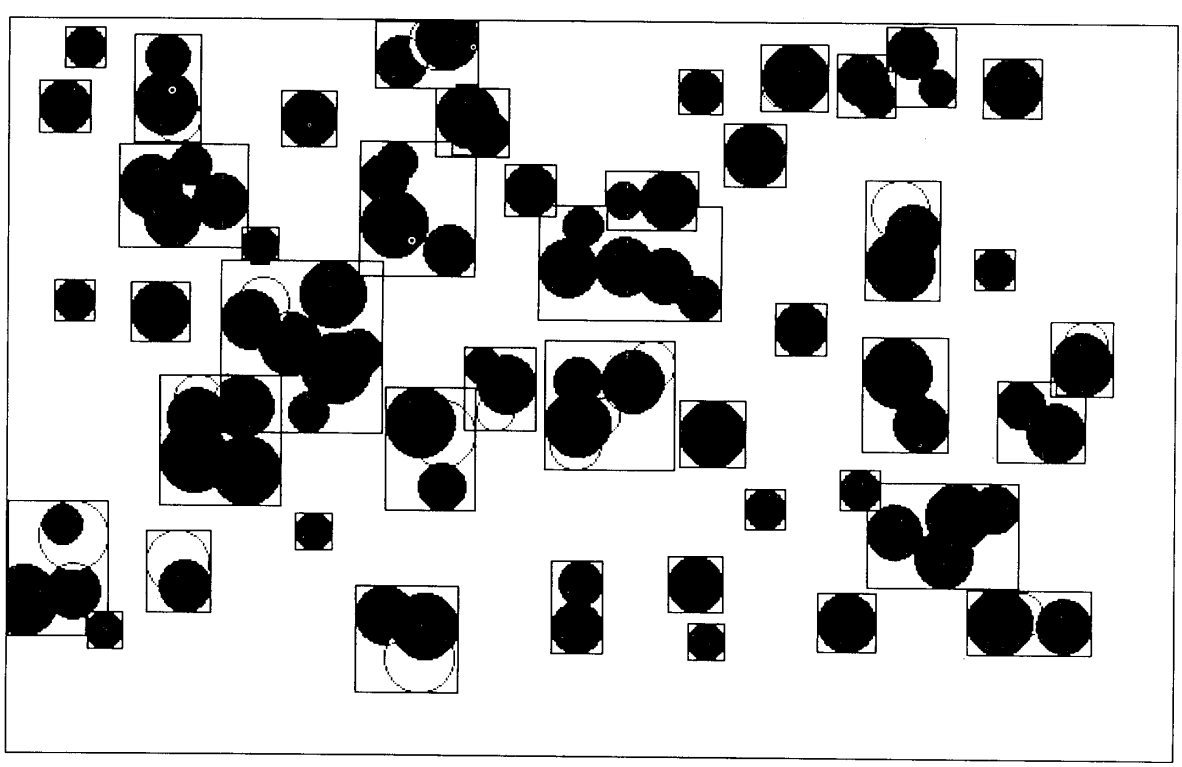

図1　領域塗りによる外接箱の獲得

```
procedure get-points (point：point-type；
                      scolor：色の集合；
                      scolor：塗りつぶし色)
var x1,xr：integer；
begin
  push(point)；
  while  sp<>0  do
    begin
      pop(point)；
      if (pの4近傍がscolorかocolor)=true then
    begin
        pを点集合に加える。
    end；
  x1：=p.x−1；
  while getpixel(x1,p.y)=scolor do begin
  if ((x1,p.y+1)と(x1,p.y−1)の色がscolorか
        ocolor)=false then
    begin
        (x1,p.y)を点集合に加える。
    end；
  x1：=x1−1；
  end；
xr：=p.x+1；
while getpixel(xr,p.y)=scolor do
begin
  if ((xr,p.y+1)と(xr,p.y−1)の色がscolorか
        ocolor)=false then
    begin
        (xr,p.y)を点集合に加える。
    end；
  xr：=xr+1；
end；
line (xl+1,p.y,xr−1,p.y)；
line (x1+1,p.y)と(xr−1,p.y)を点集合に加える。
end；
```

手続き3　輪郭線の点集合の獲得手続き

```
point-type=                    pointの型
    record
      x：integer；
      y：integer；
    end
```

x	x座標
y	y座標

```
shape-type=
  record
  xs,ys,xe,ye ：integer；
  xc,yc       ：integer；
      s       ：longinteger；
  part        ：arry [0..n,0..m] of point；
end；
  end；
```

xs,ys,xe,ye	外接箱
xc,yc	重心
s	面積
part	複数の輪郭線

図2　点と形状の構造

画像からこの構造を得る手続きは手続き4のようになる．またS式からもとの画像のを復元する手続きを手続き5に示す．

4．むすび

領域塗りを利用して，画面に存在する物体の基本的な情報（重心，面積，外接箱，輪郭線）を抽出する方法について述べた．またこれを用いて画面全体を構造化する一方法を示した．この手法の問題点は輪郭線の座標を，

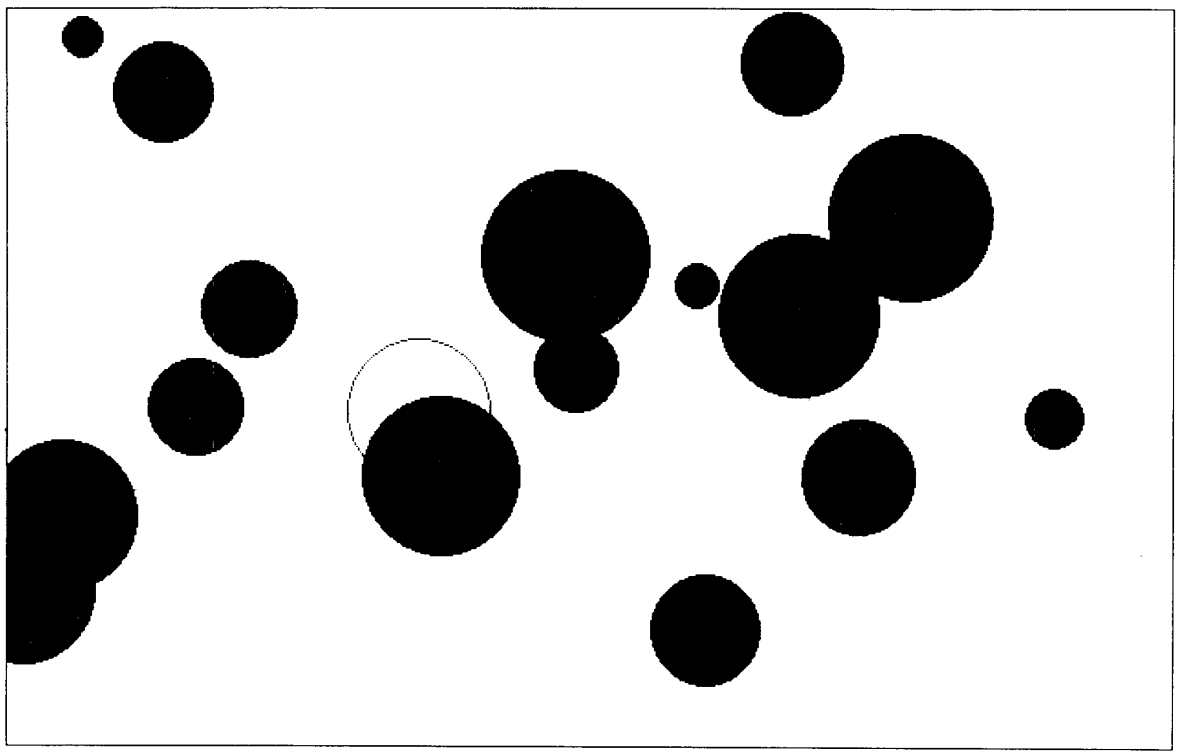

図 3　輪郭線処理前の画像

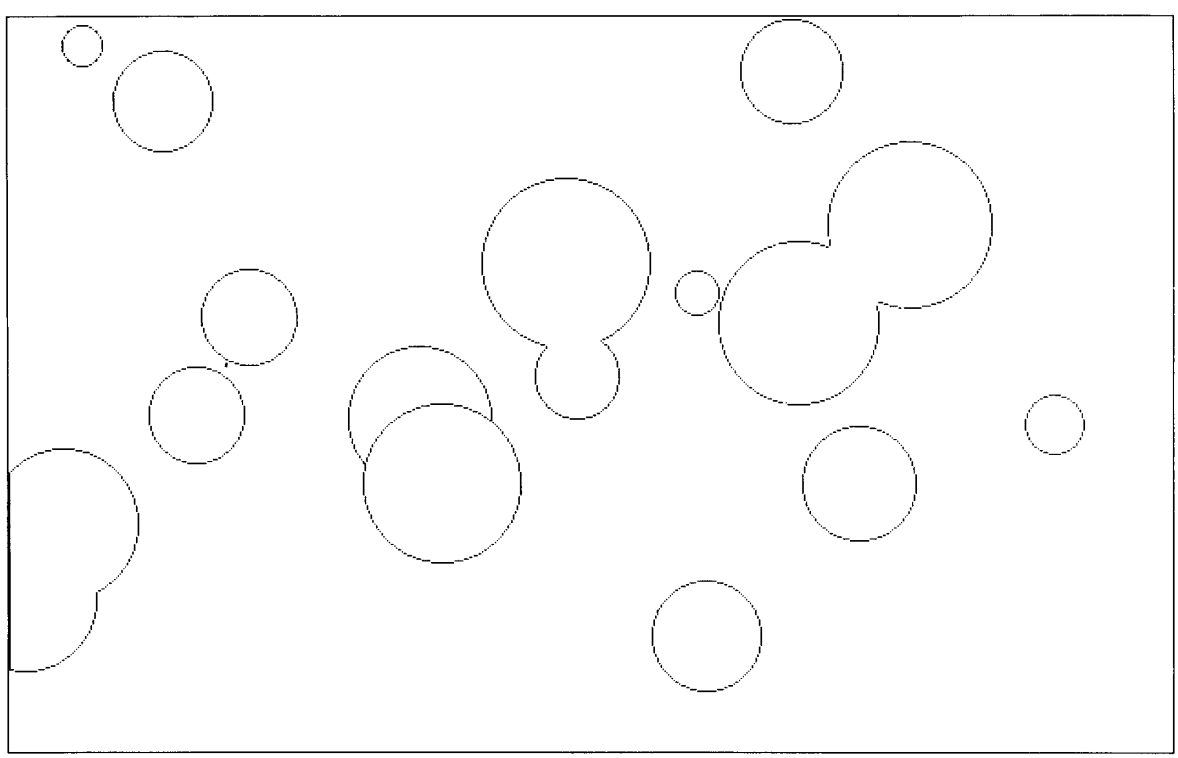

図 4　図 3 の輪郭線

```
procedure get-sexpr(var s：S式)；
begin
while sp<>0 do
begin
point：=pop(sp)；
〝point始点として領域塗りを行い，包含関係にある図形
  を除いて情報を獲得する。〞；
〝領域の内側の点で背景色でない点があれば，塗りつぶ
  しget-sexpr(s)を実行する。〞；
〝領域内のS式を1つのリストにしてs式に加える。〞
end；
end；
```

手続き4　画像全体の構造化手続

```
procedure draw(p：sexpr)；
  begin
    〝輪郭線を描く。〞；
    〝復元座標と色を用いてぬりつぶす。〞；
    if 包含関係リスト<>nil then
      begin
        draw（car（包含関係リスト））；
        draw（cdr（包含関係リスト））；
      end；
  end；
```

手続き5　画像の復元手続

単なる配列として格納しているので，この部分の構造化が必要である．現在この部分の構造化のために，4分木構造を適用して完全な構造化データを構成する予定である．構造化された図形情報は比較的容易にデータベース化が可能である．今後これらの手法を文書画像に適用する予定である．

5．参考文献

1）田端，鶴岡，他：〝表の構造理解のはための罫線抽出と領域分け〞信学技報，NLC90-33，PRU90-73
2）秋山，増田：〝周辺分布特徴と線密度特徴を併用した反復文書画像領域分割法〞PRL85-41
3）成瀬，金井，他：〝帳票における枠罫線構造の記述と認識〞信学技報，PRU90-150
4）ターボパスカルはボーランド社（米国）の製品である．

（平成3年9月24日受理）