木構造を用いた画像データ処理

大久保　明　伸*

Image Data Processing Using Tree Structure

Akinobu OOKUBO

Abstract

We previously proposed an algorithm of makeing structure for bitmap image using floodfill procedure. This paper presents a method to make data structure for image data. This data structure is called QTREE. Qtree completely keeps bit pattern of image data on CRT display. We define fundamental operations for QTREE, and QTREE is represented on memory by strings and it is translated huffman code when is saved into external file.

1．まえがき

パーソナルコンピュータは年々高性能になってきた．今までは大型のシステムでしか実現できなかった処理が多かったが，この数年で十分実現可能なものの一つに文書画像データベースがある．これまでのデータベースのほとんどは，テキストを中心に検索されるものがほとんどであった．新聞，雑誌や種々の書籍が持つ情報量のかなりの部分は画像や図面が保持している．したがって，テキスト情報のみの検索では不十分なデータベース機能しか利用できないことがわかっていたが，コストパフォーマンスの関係で一部の大型や中型計算機システムにおいて試作検討されてきた．

現在，パソコンもこれらを実現する十分な性能を，低価格で持つようになりつつある．教育現場でも，パソコンの台数や性能がそろってきて，プログラミング教育ばかりでなくデータベース等を利用した広範囲の教育分野に利用することが可能になってきた．教育分野で利用されるシステムに画像の蓄積機能や処理機能が不可欠である．CAIや教材製作等のシステムばかりでなく，学校事務等においても重要である．これらは一部のソフトウェアで実現されているものの，より高度な機能が望まれている．これらのシステムに画像の蓄積や検索等の可能なデータベースを組み込むことが可能になれば，パソコンの利用範囲も容易に拡大することは明かである．我々はこのような現状をふまえて，特に文書画像のデータベース化を行うための基礎研究を行っている．

前に画像データの構造の一手段として，領域塗り手続きを利用した図形や文字を抽出し，これを構造化する手法について報告した．

今回の報告はイメージスキャナから得た画像データを，ビットパターンと同等の機能を持ちながら，種々の画像演算が可能なQTREEと呼ぶデータ構造の構成法，その処理関数とデータ圧縮について報告する．

2．QTREEの定義と基本処理関数

ここでは画像のQTREEの定義とこの構造への変換法についてのべる．画像としては，イメージスキャナから収集したデータでパソコン画像は2値（白，黒）とする．

2.1　QTREEの定義

QTREEとは**図1**のような構造をいう．ノードの下に4つの成分があり，それぞれの成分もまたQTREEであるような再帰構造をしている．木の先端すなわち葉はblack

*宇部工業高等専門学校電気工学科

アトムか，whiteアトムである．最も単純なQTREEはblackアトムかwhiteアトムのみである．

図1に示す構造を文字列で表現するために，LISPで用いるS式を採用する．またblackアトムはb，whiteアトムはwで表す．QTREEの正確な定義を次に示す．

［定義］QTREEとは次の構文を持つものである．

QTREE：：＝b
　　｜ w
　　｜(QTREE QTREE QTREE QTREE)

定義よりQTREEは一般に4つの要素を持つ構造化されたデータである．アトムwとアトムbはいつでも同じ4つの要素に分解できるものとする．QTREEの4つの要素をそれぞれA1，A2，A3，A4とよぶことにする．これらの各要素はまた4つの要素に分解可能である．アトムをさらに要素に分解しても，4つの成分は同じアトムから構成されているものと考えることができる．すなわち，(b,b,b,b) や (b,(b,b,b,b)，(b,b,b,b)，b) はアトムbに等しい．アトム以外で構造を持ったQTREEのうち，最も単純なものは (b,b,b,w) (b,b,w,b) (b,b,w,w) (b,w,b,b)……(w,w,w,b) の14種類である．これを疑似アトムという．あとで示すように，この疑似アトムに適当な文字（A，B，C，……N）を割り当てて表すこともある．

2.2 QTREEに対する基本処理関数

ここではQTREEに対する基本処理関数についてのべる．QTREEのなかの一部の構造である部分QTREEをとりだしたり，検査や，ある種の演算を行う関数群である．

（1）関数 a1,a2,a3,a4
　　引数Xのそれぞれの要素を取り出す．
　　もしXがアトムならXそのものを返す．
（2）関数 atomp
　　引数Xがアトムならtrue，それ以外はfalseを返す．
（3）関数 eqp
　　引数X，Yが構造的に等しい場合はtrue，それ以外はfalseを返す．
（4）関数 eq4
　　引数X1，X2，X3，X4がすべて等しい場合はtrue，それ以外はfalseを返す．

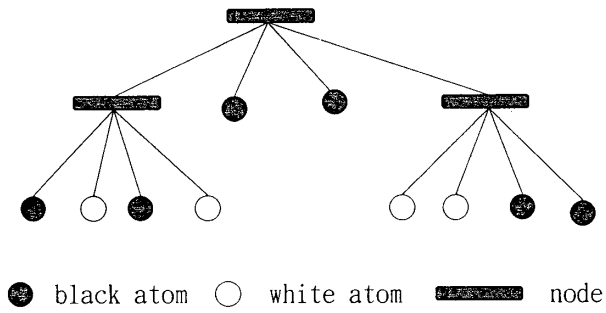

図1 QTREEの構造例

（5）関数 shift
　　引数Xの各要素をリング状にシフトする．結果はシフトされたQTREEである．
（6）nott
　　引数Xのアトムwとbを入れ換える
（7）andt
　　引数XとYの共通部分にwの部分を残したQTREEを構成して返す．
（8）ort
　　引数XとYのwの部分を残したQTREEを構成して返す．
（9）make4 (a1，a2，a3，a4)
　　任意の4つのQTREEを部分木にする新しいQTREEを構成して返す．

QTREEとこれらの基本処理関数を使用して，ビットパターンを構造化されたデータに変換する．

QTREEはふつうのビットパターンと違って，近傍のブロックをまとめて記憶することができる．従ってandt，ortやnottなどの演算ではブロック単位での演算が可能であり高速に実行できる特徴がある．これらの基本演算を利用して，図形同士の合同や相似の判定，図形の抽出が容易に実現できる．

3．画像のQTREEへの変換と復元処理

ここでは画像の変換処理と復元処理について述べる．QTREEは成分A1，A2，A3，A4の4つを持っているので，ある画像領域**図2**のように割り当てる．それぞれの領域はさらに再帰的に4つの成分に分割される．細分化の処理はアトムになるか，1ドットになれば終了する．

3.1 変換処理手続き

変換する領域を矩形とする．矩形領域を表すための座標を(XS，YS，XE，YE)で表す．この矩形領域の各辺を2等分してゆき，必要な矩形まで細分化された時点でwまたはbアトムを割り当てる．最も細分化された矩形領域は1ドットである．細分化された矩形領の近傍はまた同じアトムであることが多い．そこで，あるアトムの近傍（A1，A2，A3，A4）が同じであれば，この部分のQTREEを1つのアトムにする．これにより矩形領域の統合化ができることになる．この統合化により作られたQTREEの様子を**図3**に示す．

目的の矩形領域が2のべき乗になっていれば，細分は完全に1ドットまで到達できる．しかし，一般に矩形領域は2のべき乗になっていないこととが多く，**図4**のに示すような分割領域が最終的に残ることがある．この場合は，細分化できるところまで処理して，細分化が不可能な場合は仮想的に同じドットパターンがあるものとして処理する．以上の手続きをまとめたものがアルゴリズム1である．

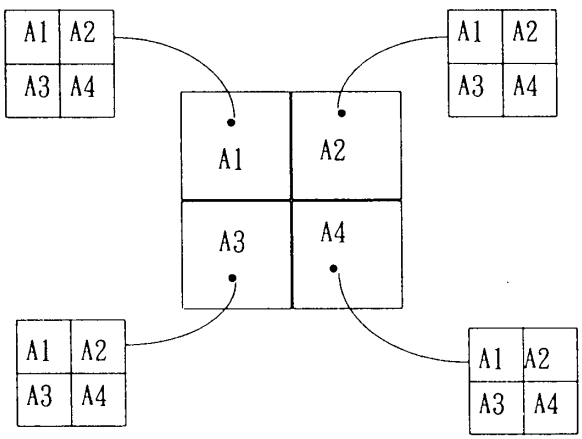

図2　画像面とQTREEの関係

［アルゴリズム1］

```
function mk_qtree (xs,ys,xe,ye：矩形領域)：
QTREEポインタ；
var a1,a2,a3,a4：QTREEポインタ；
        xm,ym：矩形領域の中心；
         dx,dy：［0,1］；
         Patom：疑似アトムへのポインタ；
begin
  if(xs=xe)and(ys=ye) then
   begin
```

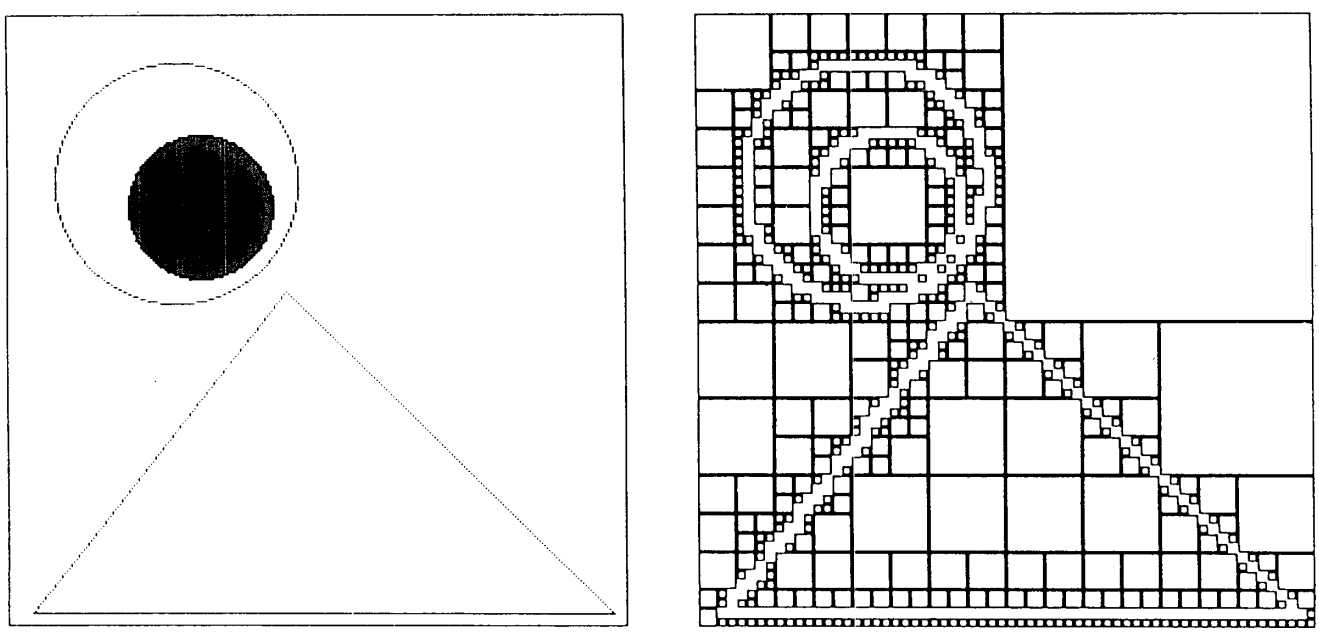

図3　矩形領域の統合化

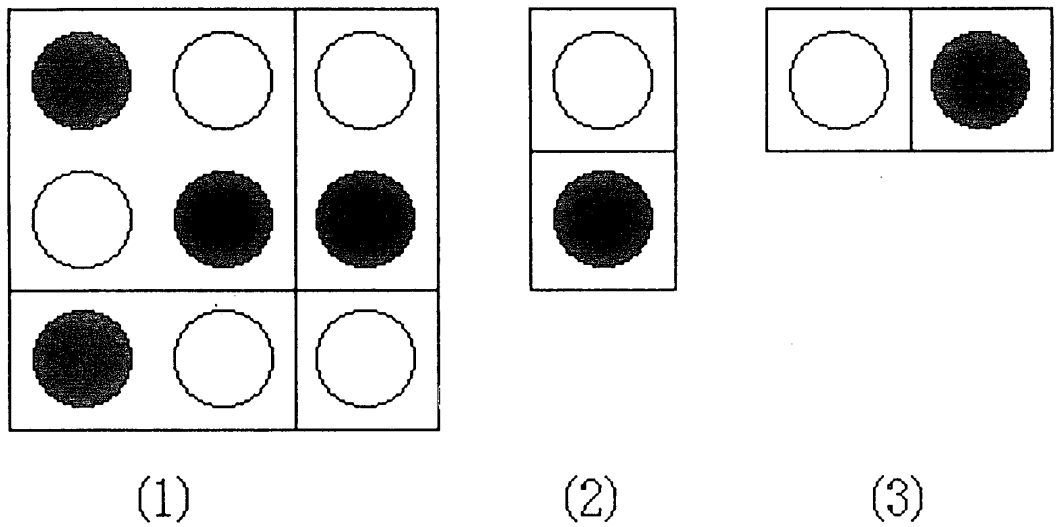

図 4 矩形領域の分割が不完全な例

```
if getpixel(xs,ys)=white then
 begin
   mk_qtree:=white_atom ポインタ;
 end
 else
 begin
  mk_qtree:=black_atom ポインタ;
 end:
end
else
begin
 mk_qtree:=node ポインタ;

xm:=(xs+xe) div 2;ym:=(ys+ye) div 2;
if xs=xe then dx:=0 else dx:=1;
if ys=ye then dy:=0 else dy:=1;

a1:=mk_qtree(xs  ,ys  ,xm  ,ym);
a2:=mk_qtree(xm+dx,ys  ,xe  ,ym);
a3:=mk_qtree(xs  ,ym+dy,xm  ,ye);
a4:=mk_qtree(xm+dx,ym+dy,xe  ,ye);

if eq4(a1,a2,a3,a4) then
begin
 mk_qtree:=a1;
end
 else
 if (a1,a2,a3,a4がアトム) then
 begin
  Latom:=make_Patom(a1,a2,a3,a4);
  mk_qtree:=Latom;
 end
end
```

3.2 QTREE から画像への変換

3.1の mk_qtree 手続きにより得られたQTREEをもとの画像に変換する手続きについて述べる.

QTREE はもはや座標情報は持っていない.獲得した矩形領域に関係なく画像を復元できる. とくに矩形領域が元の領域に相似であれば図形の拡大縮小も自由に行うことができる. またQTREE を画像に変換する場合, 一定のレベルで打ち切ればもとの画像に対して近似画像になっている. これを**図 5** に示す. また図形の90度単位の回転も前に述べた SHIFT 関数により簡単に実現できる.

画像変換手続きのアルゴリズムを次に示す.

[アルゴリズム 2]

```
procedure draw_qtree(p:QTREE ポインタ;xs,
ys,xe,ye:矩形領域);
  var xm,ym:矩形領域の中心;
      dx,dy:[0,1];
  begin
```

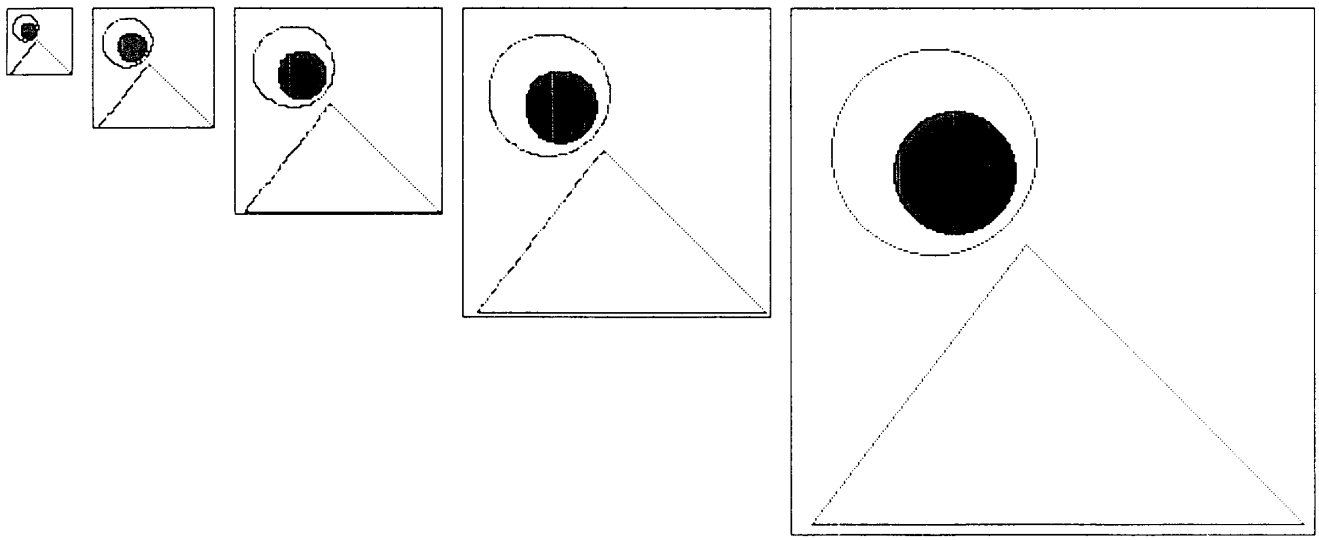

図5　QTREEを任意の矩形領域に展開した例

```
if not(pがアトム) then
 if p=white_atom then
   begin
    矩形領域を白で塗る
   end
end
else
 begin
  xm:=(xs+xe) div 2;ym:=(ys+ye) div 2;
  if xs=xe then dx:=0 else dx:=1;
  if ys=ye then dy:=0 else dy:=1;
  draw_qtree(a1(p),xs,ys,xm,ym);
  draw_qtree(a2(p),xm+dx,ys  ,xm  ,ym);
  draw_qtree(a3(p),xs  ,ym+dy,xm  ,ye);
  draw_qtree(a4(p),xm+dx,ym+dy,xe  ,ye);
 end;
end;
```

4．QTREEの内部構造と圧縮符号化

QTREEを計算機内部で直接実現するためには，ポインタを用いる方法が最も簡単であるがメモリの消費量が多くて，実際の画像を処理するのは不利である．そこでQTREEをt，w，bをそれぞれ木のノード，wアトム，bアトムに対応つけて符号化した．例えば（b（w b w b）b w）は，tbtwbwbbwで表す．さらにアトムが4個からなるようなQTREEの部分木は14種類しか存在しないので，これに適当な文字を割り当てる．(w b w b) をAに割り当てれば上のQTREEはt b A b wとなる．見かけ上のアトムを導入する事でQTREEを維持する文字列の長さはきわめて少なくなる．実際の処理はこの文字列を利用して行われる．しかし画像データがさらに多量になる場合やデータを外部のファイルに格納するときは，この文字列にたいしてハフマンの符号化を施した．ハフマンの符号化の問題点は符号化のための辞書を記憶しておく必要がある点にある．しかし，QTREEに現れる文字の種類は高々17種類なのでこのための記憶の消費量は少ない．またこのためにハフマンコードを複号化する場合も比較的高速に行うことができる．640×400ドットの文書画像を記憶するのに，2値のビットパターンを直接記憶するためには32kバイトが必要である．QTREEを使用した場合には15kバイトから4kバイト程度であった．またハフマンの符号化を行うと，これの1割程度の圧縮効果があった．

5．むすび

文書画像データを抄録，検索，蓄積するために必要なデータの符号化にQTREEを利用する方法について述べた．特にQTREEを操作するための基本関数と，データ格納と復元に関する方法を提案し，実験例を示した．

QTREEは構造化されたデータなので文書画像の基本的

な処理を比較的簡単に実現できる可能性がある．現在画像の転送の効率を上げるために，種々のアルゴリズムやデータ構造が提案されている．しかし文書画像のように，その内容を抄録，検索するためのものは比較的少ない．今後，文書画像データベースシステムを構築していく予定である．

参 考 文 献

1）大久保：領域塗り手続きを用いた図形の構造化，宇部高専研究報告，38号，平成4年3月

2）飯沢：文書画像データベース，情報処理33巻5号，1992年5月

3）吉田，稲垣　他3：知識ベースに基づいた図書目録カードの理解，情報処理31巻12号，1990年12月

4）美濃，池田：視覚的特徴に基づく図面からのシンボル候補の抽出と分類，情報処理33巻9号，1992年

5）孫，鈴木，他2名：文字構造情報に基づく高精度な文字切り出し処理を用いた文書認識，情報処理33巻9号，1992年

（平成4年9月10日受理）