7J-6

2値画像の一符号化法と演算アルゴリズム

西谷泰昭, 吉田浩幸†, 清水賢資
群馬大学工学部, †沖電気工業(株)

1 はじめに

画像処理, コンピュータグラフィクスなど大量の図形データを計算機で処理するには, 莫大な計算量とメモリ量が必要であり, データの効率的な表現とそのデータに対する効率的な演算が重要となる. 本稿では, Pコードと呼ばれる2値画像表現のデータ構造を提案し, 代表的な2次元画像表現であるquadtreeとの関係, メモリ量, 各種の演算アルゴリズムについて述べる.

2 Pコードの定義と領域の表現

画素はこれ以上分割できない最小の正方形であり, その一辺の長さを1とする. また, 画像は$2^N \times 2^N$の画素からなるものとする. 通常, 画素はx座標、y座標の組で表現されるが, これを1個の整数に符号化する.

定義1 Pコード(pixel-code)は, (x,y)で表わされる画素を次のような関数Pで符号化したものであり, $2(N+1)$ビットの整数である.

$$P(x,y) = \sum_{i=0}^{N} < x >_i 2^{2i} + \sum_{i=0}^{N} < y >_i 2^{2i+1}$$

ここで, $< x >_i$はxの2進数表示の第iビットを表わす. (注: 負の座標を扱うために2^{N+1}を法とする座標系を考えている.)

直観的には, Pコードの偶数ビットがx座標, 奇数ビットがy座標を表わす. 図1(a)に$2^2 \times 2^2$の画像についての各画素のPコードを示す.

領域の表現は, Pコードが整数であり大小関係がつけられることを利用する.

定義2 Pコードの組$[p_1, p_2](p_1 < p_2)$は, Pコードの集合$\{p \mid p_1 \leq p < p_2\}$を表わす. 更に, 偶数個のPコードをソートした列$[p_1, p_2, \ldots, p_{2L}](p_{2i-1} < p_{2i} \leq p_{2i+1})$をPコード列(pcode-sequence)といい, これは, $\bigcup_{i=1}^{L}[p_{2i-1}, p_{2i}]$を表わす. LをPコード列の長さという(図2(b)).

Encoding and operations of binary images
Yasuaki Nishitani, Hiroyuki Yoshida†, Kensuke Shimizu
Gunma University, †Oki Electric Industory

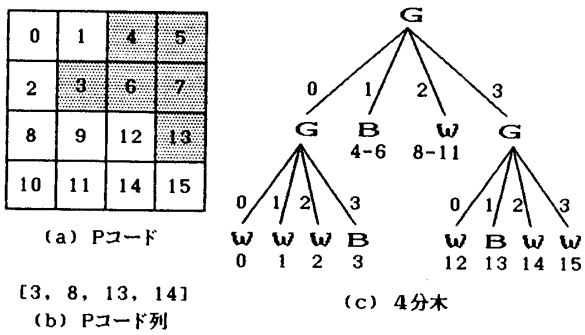

図1 Pコードと4分木

3 Pコード列とquadtree

2次元領域に対して, よく用いられる階層データ構造のひとつに4分木(quadtree)がある[1]. これは, 正方領域(quadrant)の4分割を, quadrantが対象領域に含まれるか(Bノード), 外になる(Wノード)まで再帰的に行ない, それを木構造で表わしたものである(図1(c)).

Pコードを4進数として見ると, 4分木の根から画素へのパスとなっている. (Bノードへのパスはquadcodeと呼ばれる[2].) Pコードとquadrant(4分木のノード)の間には次の性質が成り立つ.

性質1 pをPコードとし, $s \leq smax(p)$とする. このとき, $r = [p, p+4^s]$について次式が成り立つ. このようなrをquadrantという.

$$q \in r \quad iff \quad X(p) \leq X(q) < X(p) + 2^s \\ \& \ Y(p) \leq Y(q) < Y(p) + 2^s$$

ここで, $smax(p)$は, pの下位$2i$ビットがすべて0であるような最大のiであり, $X(p)$はpのx座標, $Y(p)$はpのy座標である.

この性質は, 4分木のノードに対応する正方領域をPコードの組で表現できることを示している. この変換は簡単に実現できる. また, 性質1から, Pコード列の長さLがBノードの個数(n_B)より小さく, Wノードの個数(n_W)+1より小さいことがわかる. すなわち, $L \leq min(n_B, n_W+1) \leq (n_B+n_W+1)/2$である. nを4分木のノード数とすると, $3n = 4(n_B + n_W) - 1$ [3]であるので, Pコード列の長さLとメモリ量M(ビット)について次の性質が成り立つ.

性質2 $L \leq (3n+5)/8,\ M \leq (N+1)(3n+5)/2.$

4分木の他の実現法のメモリ量は，次のとおり．ポインタを用いて木を実現したポインタ型4分木は$(8N+9)n$，子の情報を親に持たせることにより葉を削除した改良ポインタ型4分木[3]は$(2N+3)n$，深さ優先の順序でノードを並べたDF表現は$2n$，DF表現にアクセス効率化のためのポインタを加えたものは$(2N+4)n$である．ここで，ポインタは4^{N+1}個のノードを区別するために$2(N+1)$ビットとしている．

4 集合演算

画素が領域に属するかどうかを探索する問題(member)，領域の補集合，共通集合，和集合を求める問題について考える．これらの問題については，Pコード列がソートされた列であることを利用する．member問題については，2分探索法により$O(\log L)$で探索が可能である．また，共通集合，和集合については，2つのソートされた列をマージすることにより，$O(L_1+L_2)$のアルゴリズムを作ることができる(L, L_1, L_2はPコード列の長さ)．補集合については，列の先頭と終端に0，4^Nを挿入あるいは削除することにより，$O(1)$のアルゴリズムが作成できる．

図2に，共通集合の計算時間についての実験結果を，Pコードと改良されたポインタ型4分木について示す(Sun3/60，C言語)．横軸は生成された共通集合の大きさで，4分木のノード数に換算したものである．全体としては，係数が小さいPコードの方が優れている．なお，理論的には，これらのアゴリズムの計算量は，$O(L)=O$(4分木のノード数)であるため，4分木の他の実現法についてのものと(補集合以外は)オーダとして変わらない．

5 Pコードの演算

領域の移動，隣接関係，外周の計算などは，4分木では効率の良いアルゴリズムを作ることが難しい．これは，これらのアルゴリズムが，x座標，y座標の算術演算や比較を基礎としているためである．そこで，Pコードをx(y)座標に変換せずに，Pコード上で算術演算，比較を行なうことを考える．$P_X(x)=P(x,0)$，$P_Y(y)=P(0,y)$とすると，次の式が成り立つ．

$$P(x,y)=P_X(x)\vee P_Y(y)$$
$$P_X(X(p))=p\wedge(01)^*,\ P_Y(Y(p))=p\wedge(10)^*$$
$$P_X(x_1+x_2)=(P_X(x_1)+(P_X(x_2)\vee(10)^*))\wedge(01)^*$$
$$P_Y(y_1+y_2)=(P_Y(y_1)+(P_Y(y_2)\vee(01)^*))\wedge(10)^*$$

ここで，$\wedge, \vee$はビット毎の論理演算であり，$(01)^*$は0101...01なる2進数を表わす．

上記の式により，移動などの計算において最も重要なベクトルの加算を以下のようにPコード上で行なうことができる．

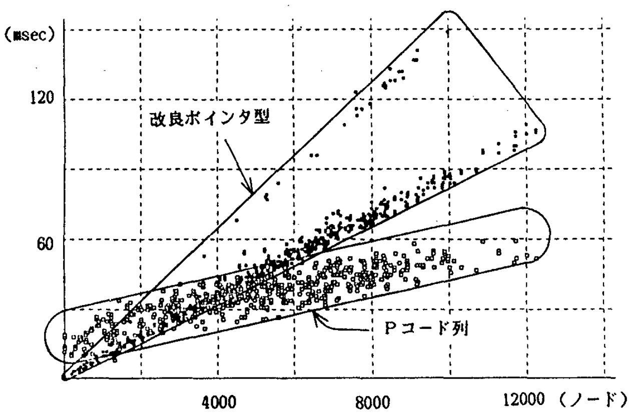

図2　共通集合の計算時間

$$\begin{aligned}P(X(p_1)+X(p_2),Y(p_1)+Y(p_2)) = \\ (p_1\wedge(01)^*+p_2\vee(10)^*)\wedge(01)^* \\ \vee\ (p_1\wedge(10)^*+p_2\vee(01)^*)\wedge(10)^*\end{aligned}$$

また，比較についても次のような関係が成立するので，x座標についての性質をPコード上の演算で調べることが可能になる．

$$x_1=x_2\quad iff\quad P_X(x_1)=P_X(x_2)$$

ここでは例として，2つのquadrant $r_1=[p_1,q_1]$，$r_2=[p_2,q_2]$が隣接しているかどうかを調べる問題を考える．quadrant $r=[p,q]$の1辺の長さを$size_X(r)$で表わすことにする．簡単のため，$X(p_1)\leq X(p_2)$ (Pコード上でチェック可能)とし，x座標についてのみ考える．このとき，隣接の条件は$X(p_2)=X(p_1)+size_X(r_1)$である．これをPコードの条件にすると以下のようになり，Pコード上のビット演算，算術演算で調べることが可能である．

$$\begin{aligned}P_X(X(p_2)) &= P_X(size_X(r_1)+X(p_1))\\ p_2\wedge(01)^* &= ((q_1-p_1)+q_1\vee(10)^*)\wedge(01)^*\end{aligned}$$

6 おわりに

2値画像の符号化Pコードを提案し，そのメモリ量，集合演算がこれまでの4分木の実現法よりも優れていることを示した．また，Pコード上での算術演算がビット演算，算術演算で可能であることを示し，その応用についても触れた．Pコードの利点は，算術演算が可能であることであり，この性質を利用して，多くの問題が見通しよく解決できるであろう．また，3次元への拡張も自明であり，画素数の多い3次元物体の表現にはPコードは有用であると考える．

参考文献

[1] H. Samet, The quadtree and related hierarchical data structure, Comput. Surv., 16, 2, (1984).

[2] S. X. Li and M. H. Loew, The quadcode and its arithmetic, CACM, 30, 7, (1987).

[3] 大川原，清水，Region Quadtreeのデータ量削減と集合演算アルゴリズム，第36回情処全大，4B-9，(昭63).