計算機アーキテクチャ 83−14
（1990. 7. 18）

網目結合計算機上での幾何学的問題のための線形時間アルゴリズム

河村明展(+)，石川隆(++)，梅尾博司(+)

(+) 大阪電気通信大学，工学部
(++) （株）松下ソフトリサーチ

本稿では1次元及び2次元網目結合計算機上での幾何学的問題を効率よく解く並列アルゴリズムを提案する．{0，1}で定義された再帰曲線の認識，ならびに平面上の長方形図形集合を圧縮する問題である．再帰曲線の認識では，よく知られているシェルピンスキー曲線とヒルベルト曲線についての認識を行なう．認識に要する時間は，図形の1辺の長さに比例した線形時間である．各プロセッサの記憶容量は有限．長方形図形集合の圧縮問題に関しては，バス付1次元網目結合計算機上で長方形図形集合の1方向圧縮アルゴリズムを提案する．圧縮に要する時間はブロック数に比例した線形時間である．

Linear-Time MCC Algorithms for Some Geometrical Problems

Akinobu Kawamura,(+) Takashi Ishikawa(++) and Hiroshi Umeo(+)

(+) Osaka Electro-Communication University
(++) MATSUSHITA SOFT RESEARCH, INC.

(+)18-8 Hatu-cho, Neyagawa-shi, Osaka, 572, Japan
(++) Twin 21 Bldg., MID Tower, 2-1-61 Chuou-Ku, Osaka, 540, Japan

Abstract

In this paper we develop linear-time algorithms for some geometrical problems on one- and two- dimensional mesh-connected computers. The problems that we consider are recognition of recursive curves and compaction of two-dimensional rectangular figures.

1．　はじめに

網目結合計算機（Mesh-Connected-Computer,MCC と略す）は次の２つの理由から並列計算機の興味深いモデルである．第一にいくつかの基本的な問題に対してＭＣＣ上で効率よく実行できるアルゴリズムが開発されてきたこと，第二に実用化という観点から眺めたとき，ＭＣＣが持つ一様性とモジュラー性はＭＣＣのＶＬＳＩ化にとって非常に都合がよいことである．[1]　網目結合はハイパーキューブ結合と並んで有用なプロセッサ間通信方式である．本稿では１次元及び２次元ＭＣＣ上での幾何学図形問題を効率よく解く並列アルゴリズムを提案する．｛０，１｝で定義された再帰曲線の認識，ならびに平面上の長方形集合を圧縮する問題である．両問題とも線形時間で動作するアルゴリズムである．正確には次の結果を得た．

２次元ＭＣＣは位数ｋのシェルピンスキー曲線及びヒルベルト曲線を入力図形の１辺の長さに比例した時間で認識できる．各プロセッサのメモリ容量は有限状態である．バス付１次元ＭＣＣは２次元２値図形の圧縮を，圧縮するブロック数に比例した時間で認識できる．各プロセッサのメモリ容量はＯ（ｎ）である．

まず２．では２次元ＭＣＣにおける基本操作を説明し，再帰曲線の線形時間認識アルゴリズムを提案する．考察した再帰曲線はシェルピンスキー曲線及びヒルベルト曲線である．認識に要する時間は，図形の１辺の長さに比例した線形時間である．各プロセッサの記憶容量は有限である．

３．ではバス付１次元ＭＣＣで長方形集合の１方向圧縮を提案する．圧縮に要する時間はブロック数に比例した線形時間である．

2．　再帰曲線の線形時間認識アルゴリズム

2．1　２次元ＭＣＣ上での基本操作

２次元ＭＣＣはn^2台の同一プロセッサから構成される並列計算モデルである．図１参照．座標（ｉ，ｊ）に位置するプロセッサを$P_{i,j}$と表わす．プロセッサ$P_{i,j}$は隣接する４つのプロセッサ$P_{i+1,j}$，$P_{i-1,j}$，$P_{i,j+1}$，$P_{i,j-1}$だけと接続されている．各プロセッサは有限個のレジスタから構成される局所記憶を持っている．

本稿では次の基本演算を使用する．

a．　与えられた入力図形の一辺の長さｌがある自然数ｋについて$l=2^{k-1}$か否かのチェック，

b．　中点（２分割点）位置の判別，

c．　ブロック・データの平行移動，

d．　ブロック・データの回転．

１辺の長さをｎとすると，上記の各演算の計算時間はＯ（ｎ）時間であることは明かである．

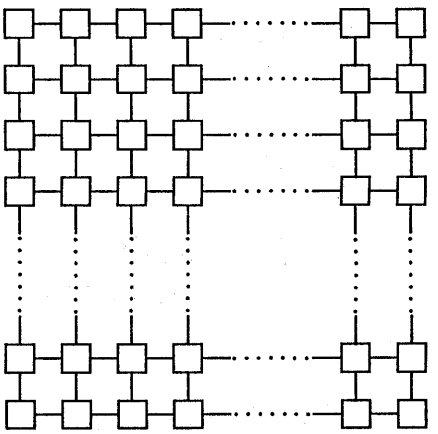

図１　２次元ＭＣＣ

2．2　シェルピンスキー曲線の線形時間認識アルゴリズム

シェルピンスキー曲線は空間充填を行なう閉曲線として最もよく知られている曲線の一つである．同図形の定義は文献[2]による．１点だけからなる図形を位数０とする．位数ｉ－１の折れ線４個を図２のようにつないで位数ｉの折れ線をつくる．さらに位数ｉの折れ線４個を図３のようにつないでできる閉曲線を位数ｉのシェルピンスキー曲線と呼ぶ．

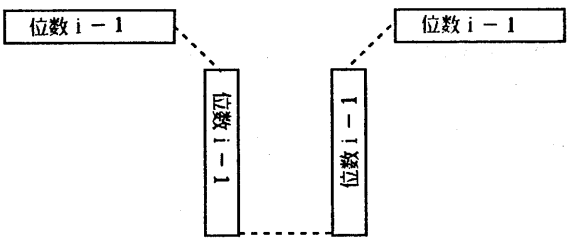

図２　位数ｉの折れ線の構成
文献［２］より

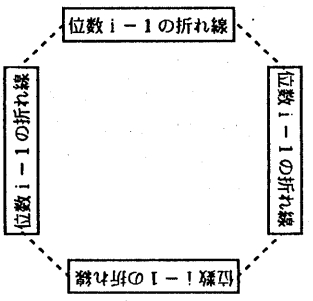

図３位数ｉのシェルピンスキー曲線の構成
文献［２］より

図４に位数４のシェルピンスキー曲線を示す．この手順からわかるように，図２の折れ線は，充填する領域が直角二等辺三角形である．[2]

これらの図形をＭＣＣ上に写像した時，曲線が存在すれば”１”，存在しなければ”０”状態をプロセッサが持つようにする．図５は位数１のシェルピンスキー曲線のＭＣＣ上への写像である．

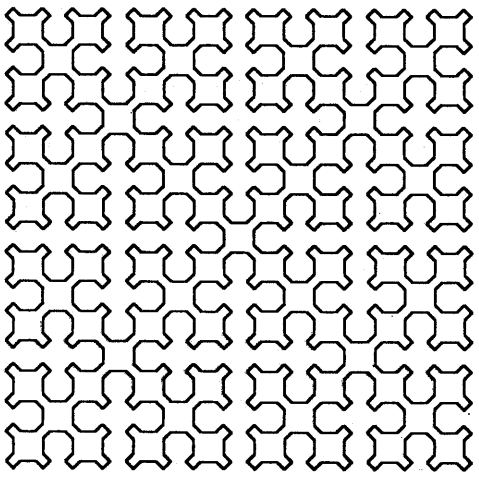

図４　位数４のシェルピンスキー曲線

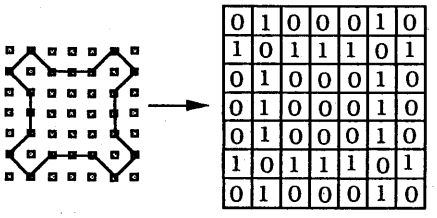

図５　位数１のシェルピンスキー曲線の｛０，１｝表現

上記の手順の逆を行なえばシェルピンスキー曲線の認識が可能である．具体的には以下の動作をする．まず，与えられた図形を対角線で４分割する．分割されたブロックはそれぞれ次のように回転する．右のブロックは左に９０度，左のブロックは右に９０度，下のブロックは１８０度回転する．そして上の三角形に全ブロックを移動し，同一な図形かどうか調べる．次項図６（ａ）参照．

次に，その三角形をさらに４個の３角形に分割する．図６（ｂ）参照．右下の三角形は右に９０度，左下の三角形は左に９０度回転する．一番右の三角形は回転しない．そして一番左の三角形に移動する．それらの各領域における部分曲線が同一な図形であるかを調べる．もし同一ならその三角形をまた４個の三角形に分割する．図６（ｃ）参照．それぞれ前回のように回転し一番左の３角形に移動して，それらが同一な図形かどうか調べる．この動作を図６（ｄ）の三角形の大きさになるまで繰り返し，残った図形がｄの図形と同じならば元の図形はシェルピンスキー曲線であるといえる．

どんな位数のシェルピンスキー曲線でもこの動作で認識することが可能である．

２．３　ヒルベルト曲線の認識アルゴリズム

ヒルベルト曲線は空間充填をおこなう閉曲線として最もよく知られている曲線の一つである．１点だけからなる図形を位数０とする．位数ｉ－１の図形を図７の様につないで出来る開曲線を位数ｉのヒルベルト曲線という．図８に位数４のヒルベルト曲線を示す．

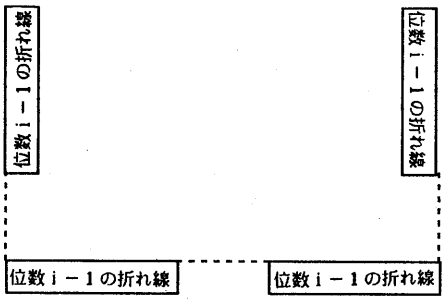

図７　位数ｉのヒルベルト曲線の構成

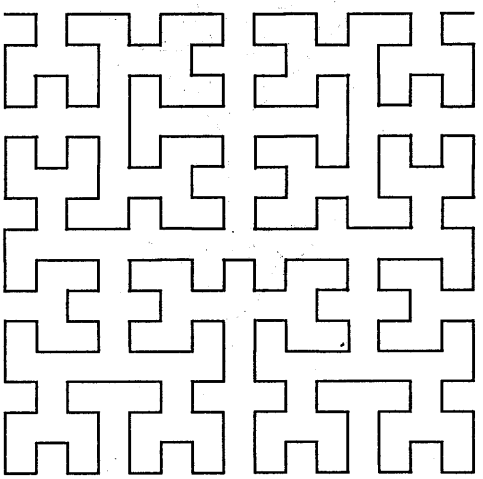

図８　位数４のヒルベルト曲線

この手順の逆を行なえばヒルベルト曲線の認識が出来る．具体的には以下の動作をする．

まず，与えられた図形を図９（ａ）の様に４個に分割する．左上のブロックを右に９０度，右上のブロックを左に９０度回転させる．右下のブロックは回転させない．そして左下のブロックに移動する．それらが合同な図形かどうか調べる．この動作を図９（ｃ）の大きさになるまで繰り返す．残った図形がｃの図形と同じならば元の図形はヒルベルト曲線であると言える．

２．４　時間計算量

両アルゴリズムの時間計算量について述べる．両アルゴリズムとも，同じ動作の繰り返しとなっている．１辺の長さをｎとすると１回の動作はＯ（ｎ）であるのは明解である．この動作をｌｏｇ　ｎ回繰り返される．しかし，１辺の長さは毎回１／２になるので全体の時間計算量はＣ（ｎ＋ｎ／２＋ｎ／４＋・・・・）＝Ｏ（ｎ）となる．但しＣは定数．

次の定理を得る．

［定理１］

２次元ＭＣＣは異数ｋのシェルピンスキー曲線及びヒルベルト曲線を入力図形の１辺の長さに比例した時間で認識できる．各プロセッサのメモリ容量は有限状態である．

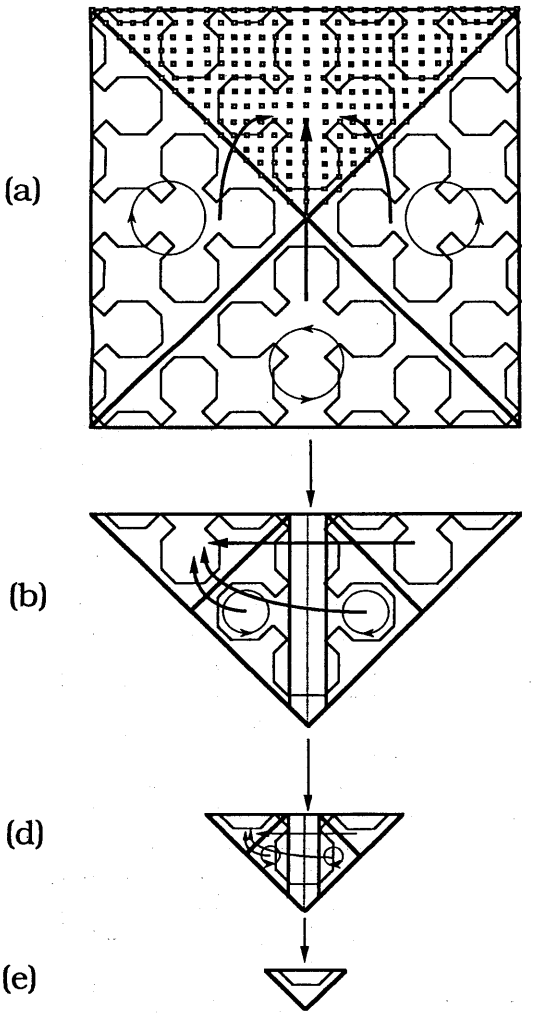

図６　再帰的４分割にもとづいた
シェルピンスキー曲線の認識過程

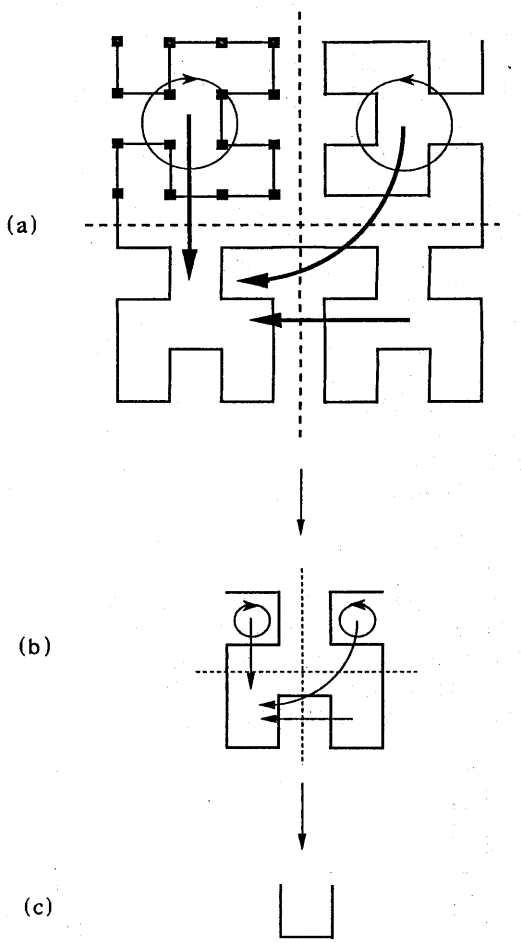

図９　再帰的４分割にもとづいた
ヒルベルト曲線の認識過程

３．　２次元２値図形の圧縮アルゴリズム

３．１　２次元２値図形の圧縮アルゴリズム

このアルゴリズムは図１０の左の図を右の図に変換するアルゴリズムである．この変換は１つ１つの長方形を上下方向にはまったく動かさず、左方向に詰めるものである．上下方向にも動かせば，もっと少スペースになるが複雑になるので今回は行なわない．

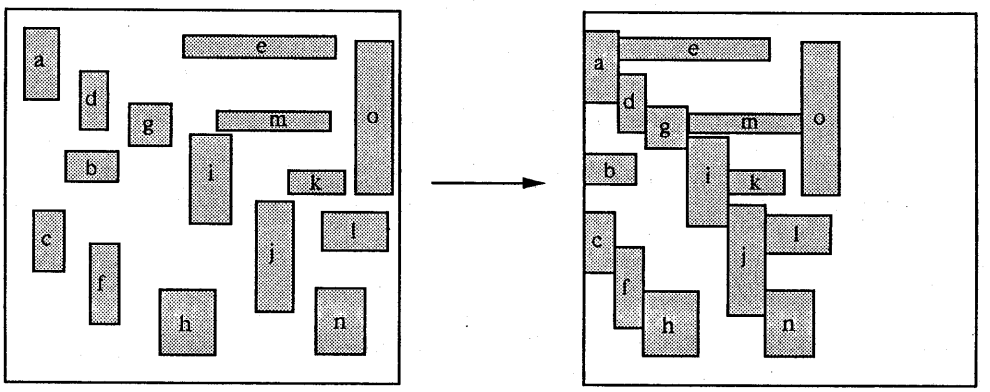

図１０　２次元２値図形の１方向圧縮

本アルゴリズムではバス付き１次元ＭＣＣを使う．図１１参照．バスを持っているので１ステップで全プロセッサにデータを送ることが可能である．入力は0/1画像を与えるのではなく，各長方形の対角の座標を各プロセッサに与える．ブロックの左上と右下の座標，（x_1，y_1），（x_2，y_2）をプロセッサは持つ．図１２参照．

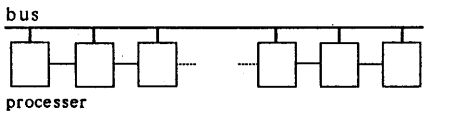

図１１　バス付き１次元ＭＣＣ

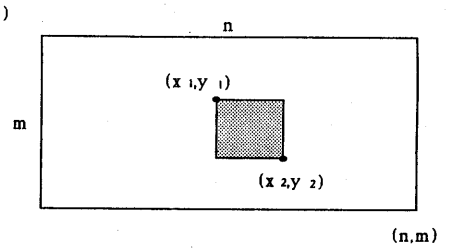

図１２　座標値によるブロックパターンの表現

分かりやすくするために図１０を使い具体的に説明する．各ブロックの左辺から左へ光線を出すとイメージすると，その光線が通過するブロックが存在する場合がある．例えば，ｉが光線を出せばｇ，ｂ，ｃを通過する．

この光線が通過するブロックを求めるために次の動作をする．各プロセッサが順番にブロック名と対角座標をバスに送信する．受け取ったプロセッサは以下の条件を満足するかどうか判断し，満足すればそのブロック名を書き込む．

（a1，b1），（a2，b2）＝　送信されてきた座標
（x1，y1），（x2，y2）＝　プロセッサが持っているブロックの座標

とすると

$a_2 < x_1$　and
$(y_2 < b_1 < y_1$　or　$y_2 < b_2 < y_1)$

上の動作を全プロセッサにわたって行なえば図１３が得られる．ｉのプロセッサはｂ，ｇ，ｃのデータを持ち，ｊのプロセッサはｉ，ｃ，ｆ，ｈのデータを持つ．次にこのデータより考える．

a	
b	
c	
d	a
e	a
f	c
g	d
h	f
i	b, g, c
j	i, c, f, h
k	i, b
l	j, c, f
m	g, d
n	j, h, f
o	e, m, k, i, g, d, a, b

図１３　プロセッサが持つデータ

1．データが何もないもの，つまり上の条件を満足するブロックがなかったものはバスに自分のブロック名とブロック長（x2－x1）を送信する．
2．バスを通じてそのデータを受け取った全プロセッサはそのブロック名を含んでいないか判別する．もし含んでいれば受け取ったブロック長をxレジスタに書き込み，そのブロック名を削除する．
3．ブロック名を削除されてグラフに何もなくなったものは，自分のブロック名とブロック長＋xレジスタを送信する．
4．バスを通じてそのデータを受け取ったプロセッサはそのブロック名を含んでいないか判別する．もし含んでいればそのブロック名を削除し，送信されてきた（ブロック長＋xレジスタ）の値が自分のxレジスタより大きければxレジスタに（ブロック長＋xレジスタ）を書き込む．
5．3と4の繰り返し　（n－1回）

この動作が終われば xレジスタの値から移動先を求める．最初に与えられた座標が（j_1，k_1），（j_2，k_2）で，xレジスタの値がx_1とすると以下の座標が圧縮したときの新座標となる

（x_1，k_1），（$x_1+j_2-j_1$，k_2）

図10のiのブロックはgとbとcに当たる可能性があるが最終的にはgのブロックの右辺が一番右にあるのでgのブロックと隣接することになる．このことより，ブロック長＋xレジスタ，すなわちブロックの右辺の座標が大きい方と隣接することになる．この判別を4の動作で行なっている．

3．3　時間計算量

1～4の動作はすべてO（1）である．3，4の動作が行なわれる度に1つのブロックの座標が決定するので，n回の繰り返しで全ブロックの座標が求められる．

よって，このアルゴリズム全体はO（n）である．

次の定理を得る．

[定理2]

バス付1次元MCCは2次元2値図形の圧縮を，圧縮するブロック数に比例した時間で認識できる．各プロセッサのメモリ容量はO（n）である．

謝辞

本研究の一部は，文部省科学研究費総合研究（A）（代表：東北大学・西関隆夫教授，No.02302047）より補助を受けている．記して謝意を表す．

参考文献

[1] S．アクル著 ”並列ソーティングアルゴリズム” 啓学出版 (1988)

[2] 大野義夫，大山公一 ”コンピューティングの玉手箱” 共立出版，bit vol20 No2 p116-117(1988)

[3] Niklaus Wirth著 ”アルゴリズム＋データ構造＝プログラム” 科学技術出版社 (1976)

[4] Beyer, W, T "Recognition of topological invariants by iterative arrays" Project,Mac Tech. Rep. TR-66,MIT,Cambridge,Mass (1969)

[5] AMT DAP series technical overview, (1989)

[6] Hoshino, T "An invitation to the world of PAX",Computer, May,pp.68-79, (1986)

[7] Ishii,M.,Sato,H.,Murakami,K.,Ikesaka,M.,and Ishihata,H. "Cellular array processer CAP and applications", Proc.of Intern. Conf. on Systolic Arrays, pp.535-544,(1988)