

再帰トーラス結合アーキテクチャを用いた 並列画像解析アルゴリズム (I)

— 並列 Hough 変換アルゴリズムとその性能評価 —

青山 正人

浅田 尚紀

松山 隆司

岡山大学工学部情報工学科

岡山市津島中3-1-1

あらまし 我々は並列画像理解システムの実現を目指して、再帰トーラス結合アーキテクチャ(RTA)に基づくMIMD型分散メモリ並列計算機の開発を行っている。本論文では、画像解析レベルの代表的な処理であるHough変換を、2次元RTA上で効率的に実行するための並列アルゴリズムについて検討を加えた。まず全プロセッサ・エレメント(PE)間の相互ブロードキャストを並列に行うための並列シフトデータ転送手順を示し、次に画像空間およびパラメータ空間の空間分割に基づく並列Hough変換アルゴリズムを4種類提案する。シミュレーションによって各アルゴリズムの性能を定量的に評価した結果、「各PEは全特徴点を保持し、短冊状に割り当てられた部分パラメータ空間に投票する」手法が最も有効であることが明らかになった。

和文キーワード

再帰トーラス結合アーキテクチャ、画像理解、Hough変換、
並列アルゴリズム、ブロードキャスト

Parallel Image Analysis Algorithms on the Recursive Torus Architecture(I)

— Parallel Hough Transformation Algorithms and their Performance Evaluation —

Masahito AOYAMA

Naoki ASADA

Takashi MATSUYAMA

Department of Information Technology,

Faculty of Engineering,

OKAYAMA UNIVERSITY

3-1-1, Tsushima-naka, Okayama

Abstract We have been developing parallel algorithms for image understanding based on the recursive torus architecture (RTA), which we proposed as an abstract MIMD architecture with dynamically reconfigurable torus network. This paper discusses parallel Hough transformation algorithms on the 2-dimensional synchronous RTA. First, we present an effective algorithm for parallel broadcasting among all PEs. Then, we propose four types of parallel Hough transformation algorithms using parallel broadcasting. Simulation results show that the algorithm that “each PE holds all feature points in an image and makes votes in parallel in its corresponding striped region of the parameter space” gives the highest performance.

英文 key words

recursive torus architecture, image understanding, Hough transform, parallel algorithm, broadcast

1 はじめに

画像理解を高速に行うためには、雑音除去やエッジ検出などの画像データを一様に扱う画像処理レベル、線分抽出や領域分割などの幾何学的な構造を持ったグラフィックデータを扱う画像解析レベル、そして知識を用いた推論や認識のためのシンボリックデータを扱う認識・理解レベルのそれぞれの段階において並列処理を実現する必要がある。このように多様なデータ構造とアルゴリズムを柔軟かつ統一的に扱うためのMIMD型分散メモリ並列計算機として、われわれは再帰トーラス結合アーキテクチャ(Recursive Torus Architecture、以下RTAと略す)を提案し、並列画像理解システムの実現を目指している[1]。

RTAは、トーラス結合を構成するPE(Processor Element)間を結ぶ通信線上にスイッチを配置し、動的にPE間の接続関係を変化させることによってトーラス結合の再帰的な分割を可能にした動的結合網であり、トーラス結合の幾何学的次元によって1次元RTA、2次元RTA、…、 n 次元RTAに、さらにスイッチ切り換えの同期方式によって同期式RTA、非同期式RTAに分類することができる。このようにRTAそのものは、汎用のMIMD型分散メモリ並列計算機の抽象アーキテクチャであり、目的に応じて自由に構成形態を選択することが可能である。本研究の目的である並列画像理解システムでは、画像データの2次元構造を扱うのに適した2次元同期式RTAが基本的な構成形態となる。

2次元同期式RTAは、2次元のメッシュおよびトーラス結合を内包しているので、画像データの2次元構造とPEの2次元配置を対応付けることによって、画像処理レベルの一樣処理を並列実行できることは明らかである。一方、画像解析レベルおよび認識・理解レベルの処理は、データ構造やアルゴリズムが複雑であるため、データや処理の並列性は明らかではなく、RTAの特徴を生かした効果的な並列アルゴリズムの開発が、並列画像理解実現のための重要な鍵となっている。そこで本論文では画像解析レベルの処理として、画像から直線や円などの図形要素を抽出するHough変換に着目し、2次元同期式RTAにおける並列Hough変換アルゴリズムを提案し、その有効性を定量的に評価した結果を示す。

以下、まず2章では2次元同期式RTAの概要と基本的性質について述べた後、RTAにおいてすべてのPE間でデータを並列にブロードキャストする効果的な手法を明らかにする。3章では、 ρ - θ Hough変換を高速化、高精度化した γ - ω Hough変換[2]に基づいて、RTAにおいて直線検出を行うための並列化手法とそのアルゴリズムを4種類提案する。4章では3章で提案した並列Hough変

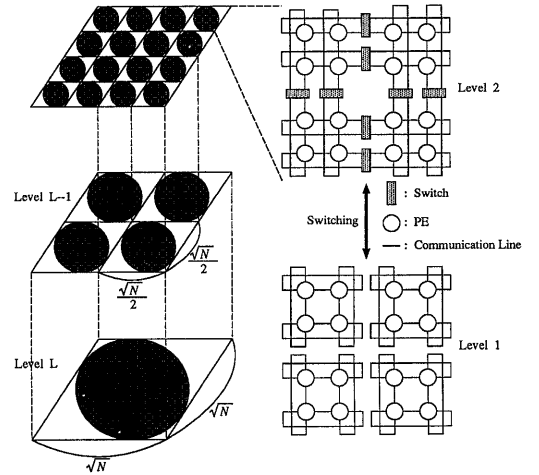


図1: 2次元同期式RTAの概念図

換アルゴリズムの性能を、シミュレーションによって定量的に評価した結果を示し考察を述べる。最後に5章では、本論文のまとめを述べ、今後の課題を示す。

2 再帰トーラス結合アーキテクチャ

2.1 2次元同期式RTAの概要

2次元同期式RTAの構成原理を図1に示す。 $N(=2^{2L}, LはL \geq 2の整数)$ 個のPEで構成された2次元同期式RTAは、 $\sqrt{N}$ 個のPEで構成された1次元同期式RTAを $\sqrt{N}$ 個水平、垂直の2方向に配置した構成に相当し、最大のトーラスである $\sqrt{N} \times \sqrt{N}$ の結合状態をレベルLと呼ぶ。スイッチ切り換えによって、レベルLのトーラスは4個の $(\sqrt{N}/2) \times (\sqrt{N}/2)$ の部分トーラスに分割することができ、この状態をレベルL-1と呼ぶ。以下同様のスイッチ切り換えを再帰的に繰り返すことによって、最終的にレベル1では 2×2 の部分トーラスが $2^{L-1} \times 2^{L-1}$ 個存在することになる。トーラス結合状態のレベル変化は、PE間の通信線上に配置されたスイッチの動的切り換えによって実現されており、あるレベルから任意のレベルへと直ちに遷移することが可能である。このように2次元同期式RTAでは、トーラス構造の再帰的な埋め込みによって動的にPE間の接続関係を変化させることができるので、静的なトーラス結合に比べて通信距離特性が格段に改善され、ソーティングなどの基本的な並列アルゴリズムでは他のEnhanced Meshアーキテクチャと同等の性能を示すことが明らかになっている[1]。

2.2 並列ブロードキャスト

2次元同期式RTAにおいて種々の並列アルゴリズムを開発するためには、組織的なスイッチ切り換えに基づく並列データ転送手順を明らかにしておく必要がある。文献[1]では、各PEが保持しているデータを1個のPEに集める基本並列データ転送手順を示し、それに基づく並列ソーティングなどのアルゴリズムについて報告した。本節では、各PEが保持しているデータを全PEに対して並列にブロードキャストする、すなわち全PE対全PE通信を実現するための並列データ転送手順を示す。まず1次元RTAにおける並列シフトデータ転送手順を定義する。

並列シフトデータ転送手順

N 個のPEで構成された1次元RTAを図2に示すようにリング結合とみなし、各PEは右の通信線にデータを1個送出し、左の通信線からデータを1個受理する転送動作を繰り返す。各PEは、最初に保持していたデータを順に送出し、最後に終了記号を送出した後は、受理したデータをFIFOの順で送出する。各PEにおいて、終了記号を $N-1$ 個ずつ送出かつ受理すれば、全データを受理したことになるので転送動作を終了する。

1次元RTA全体で転送するデータの総数を d 、各PEにおいて初期状態で保持しているデータの個数の最小値を $d_{\min}$ 、1個のデータを転送するのに要する時間を1とすると、並列シフトデータ転送手順に要する通信時間は、 $d + N - (d_{\min} + 1)$ と表され、通信オーダは $O(d + N)$ となる。

この並列シフトデータ転送手順を利用して、2次元RTAにおける並列ブロードキャストを以下の2種類の方法で実現する(図3)。

方法1

2次元RTAを、各PEがアドレス¹に従って、 $(0,0), \dots, (0, \sqrt{N}-1), (1, \sqrt{N}-1), \dots, (1,0), (2,0), \dots, (\sqrt{N}-1,0), (0,0)$ という順序に接続された1次元リング結合とみなして、並列シフトデータ転送手順を適用する。

N 個のPEで構成された2次元RTAにおいて、通信時間は先の並列シフトデータ転送手順と同じ $d + N - (d_{\min} + 1)$ と表され、通信オーダは $O(d + N)$ となる。

方法2

2次元RTAの各行、各列を1次元リング結合とみなして、行→列(列→行)の順にそれぞれ

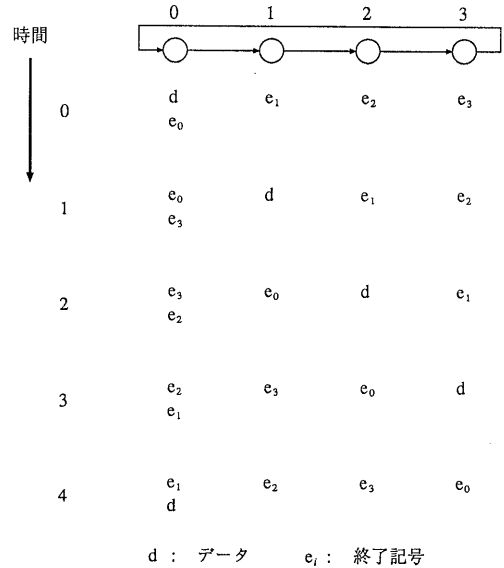
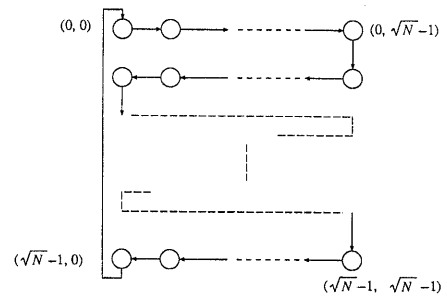
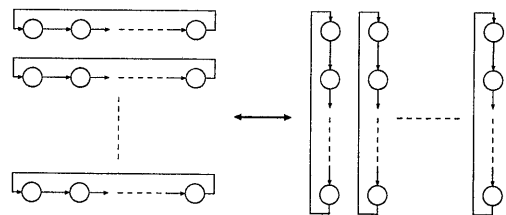


図2: 並列シフトデータ転送手順



方法1



方法2

図3: 2次元RTAにおける並列ブロードキャスト

¹2次元RTAにおいて、上から i 行目($0 \leq i \leq \sqrt{N}-1$)、左から j 列目($0 \leq j \leq \sqrt{N}-1$)の位置にあるPEのアドレスを (i, j) とする。

の方向で並列シフトデータ転送手順を適用する。

行→列(列→行)の順で並列シフトデータ転送手順を適用する場合、 N 個のPEで構成された2次元RTA全体で転送するデータの総数を d 、初期状態で各行(列)に存在するデータの個数の最大値を d_{lmax} 、最小値を d_{lmin} 、 d_{lmax} 個のデータを保持している行(列)に含まれるPEが保持しているデータの個数の最小値を $d_{lmax-min}$ とすると、通信時間は $d+d_{lmax}+2\sqrt{N}-(d_{lmin}+d_{lmax-min}+2)$ と表され、通信オーダは $O(d+\sqrt{N})$ となる。

方法1と方法2のいずれにおいても、通信時間が最良となるのはすべてのPEが同一個数のデータを保持するとき、最悪となるのは1個のPEがすべてのデータを保持するときである。最良通信時間については、方法1と方法2の通信時間の差が $N-2\sqrt{N}+1(>8)$ となるので、方法2の方が有利であることが分かる。一方、最悪通信時間については、方法1と方法2の通信時間の差は $N-d-2\sqrt{N}+1$ となり、PEの数 N とデータの総数 d によって優劣が入れ替わる。実際には、両者の通信時間の差は最良と最悪の場合の間になるので、一般には方法2を適用する方が有利である。したがって、以下では2次元RTAにおける並列ブロードキャストには方法2を用いることとする。

なお並列ブロードキャストを行う目的でRTAにブロードキャスト・パスを付加しても、パスの競合を避けるためにIPE対全PE通信を繰り返すことによって全PE対全PE通信を実現することになる。したがって通信オーダは、並列シフトデータ転送手順と同じ $O(d+N)$ となり、RTAにおける並列ブロードキャストのためには、専用のブロードキャスト・パスを付加する必要のないことが分かる。

本節で示した並列シフトデータ転送手順では、各PEで転送動作の終了を検出するために終了記号を必要としたが、あらかじめRTA全体で転送するデータの総数が全てのPEにわかっているならば、終了記号を使用することなく各PEで終了を検出することが可能である。これを実現するためには、基本並列データ転送手順における総和のアルゴリズムを使用してデータの総数を求め、基本並列データ転送手順を逆順に実行する方法が利用できるため、終了記号を付加しない並列ブロードキャストも実現可能である。

3 並列Hough変換アルゴリズム

3.1 $\gamma-\omega$ Hough変換

$\gamma-\omega$ Hough変換は、 $\rho-\theta$ Hough変換のパラメータ空間において生じる投票数の偏りをなくし、

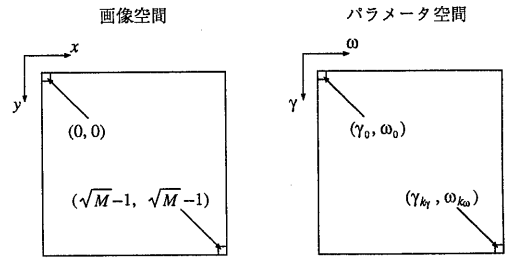


図4: 画像空間、パラメータ空間の座標

デジタル直線の検出精度を高めることを目的として考案されたHough変換であり、変換式に三角関数を含まないという特徴を持っている[2]。この $\gamma-\omega$ Hough変換では、画像空間中の特徴点 (x,y) は $\gamma-\omega$ パラメータ空間において次式で表される2本の区分的直線として投票軌跡を描く。

$$\begin{cases} \gamma = \alpha\omega y + x & \frac{1}{\alpha} \leq \omega < \frac{1}{\alpha} \\ \gamma = (2-\alpha)x + y & \frac{1}{\alpha} \leq \omega < \frac{3}{\alpha} \end{cases}$$

定数 α は、画像空間におけるデジタル直線の検出分解能を決定する係数であり、画像空間の中心点と辺上の画素を結ぶ直線集合において隣接する2直線が区別できるようにするためには、 $\alpha = 2/(\text{画像空間の1辺の画素数})$ に設定する必要がある。

以下では、 $\gamma-\omega$ Hough変換に基づき、2次元RTAを用いた直線検出用の並列Hough変換アルゴリズムについて述べる。ただし本論文では、処理対象画像はすでに特徴点検出が行われているものとし、パラメータ空間への投票が終了するまでのアルゴリズムについて考える。また以下の議論では、2次元RTAは $\sqrt{N} \times \sqrt{N}$ ($N = 2^{2L}$, $L \geq 2$ の整数)個のPEで構成され、画像は $\sqrt{M} \times \sqrt{M}$ ($M = 2^{2K}$, K は $K \geq 2$ の整数、 $L \leq K$)の画素で構成される正方画像であるとする。そして図4に示すように、画像空間の画素の座標は左上を $(0,0)$ 、右下を $(\sqrt{M}-1, \sqrt{M}-1)$ とし、パラメータ空間のセルの座標は左上を (γ_0, ω_0) 、右下を $(\gamma_{k-1}, \omega_{k-1})$ とする。ただし、 k_γ, k_ω はそれぞれ γ, ω パラメータの量子化数を表す。

3.2 並列化手法

初期状態において、2次元RTAは図5に示すように2次元画像空間にマッピングされ、各PEは対応する位置の部分画像データを保持しているものとする。したがってアドレス (i,j) のPEは、左上が $(j\sqrt{\frac{M}{N}}, i\sqrt{\frac{M}{N}})$ 、右下が $((j+1)\sqrt{\frac{M}{N}}-1, (i+1)\sqrt{\frac{M}{N}}-1)$ の座標値で表現される $\sqrt{\frac{M}{N}} \times \sqrt{\frac{M}{N}}$ 画素分の部分画像を局所メモリに保持していることになる。

Hough変換では、画像空間およびパラメータ空間の分割方法と各PEへの割り当てを工夫することによって、PE間のデータ転送の負荷をいかに少なくするかが処理効率向上の鍵となる。以下で示す並列Hough変換アルゴリズムは、各PEが処理する画像空間およびパラメータ空間の範囲によって、図6に示す4種類の並列化手法に分類することができる。

手法1は画像空間についてのみ分割を行い、パラメータ空間の1セルに対応するバケットをPE間で転送することにより各セルに対する投票を得るものである。それに対して、手法2、3、4は画像空間、パラメータ空間ともに分割を行う手法である。

以下、各手法の特徴とアルゴリズムを述べる。なお、手法2、3、4におけるパラメータ空間の分割の様子を図7に示す。

手法1 PE間でバケットを転送し、各PEは、バケットのセルの値に対応する特徴点が部分画像中に存在すれば、バケットに投票する。

この手法は、パラメータ空間中の1セルが画像空間中の1直線に対応するというHough変換の性質を利用したものである。パラメータ空間中の各セルを表すバケットが、それぞれ画像空間において対応する直線に沿ってPE間で転送される。バケットは (γ, ω) の値とそのセルに対する投票をカウントする部分で構成される。各PEは通過するバケットのセルの値に対応する特徴点を保持している場合に、そのバケットに投票する。ただし、複数のバケットが並列に移動するとき衝突を起こさないようにするため、バケットは画像空間の上から下への転送と、左から右への転送の2回に分けて行うことになる。なお投票結果は、各PEが保持している複数のバケットに残されている。

1. バケットを上から下(左から右)へ転送する場合、予め割り当てられた ω の値と部分画像の上辺(左辺)の画素の座標値 (x, y) から、 γ の値を計算し、パラメータ空間中のセル (γ, ω) に対応するバケットを生成する。その座標値 (x, y) が示す画素に特徴点が存在すればバケットに投票する。
2. バケットは部分画像内およびPE間を (γ, ω) に対応する直線に沿って移動し、各PEは特徴点が存在すればバケットに投票する。なお、画像空間の左右(上下)どちらかの辺上の画素と下辺(右辺)の画素を結ぶ直線に対応するバケットは順次生成される。
3. すべてのバケットが画像空間中で対応する直線を走査すれば処理が終了する。

正正方分割 (すべての手法)

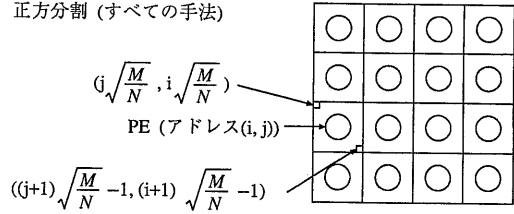
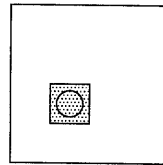
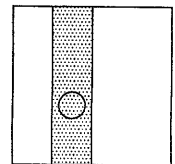
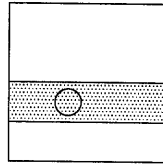


図5: 画像空間のマッピング

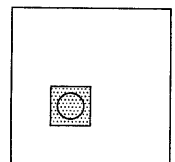
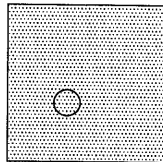
手法1



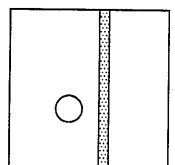
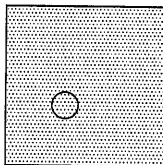
手法2



手法3



手法4



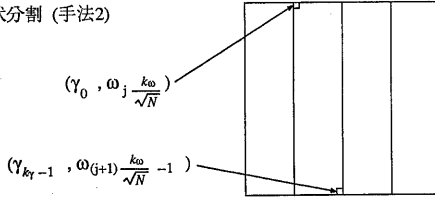
画像空間

パラメータ空間

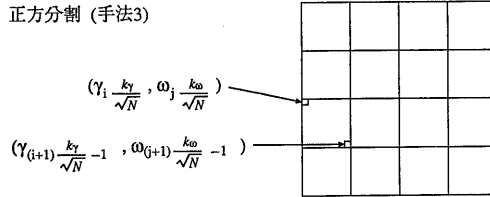
○ : PE ■ : 処理領域

図6: 並列化手法

短冊状分割 (手法2)



正方分割 (手法3)



短冊状分割 (手法4)

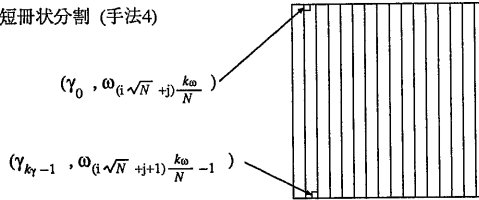


図7: パラメータ空間の分割

手法2 各PEは同じ行に属する全ての特徴点を保持し、割り当てられた短冊状部分パラメータ空間に投票する。

この手法では、各PEは同じ行に属するPE間で特徴点を並列ブロードキャストし、带状の部分画像に含まれる特徴点を処理する。すなわち、アドレス(i, j)のPEは左上が(0, i√M)、右下が(√M-1, (i+1)√M-1)、で表現される√M×√M画素分の带状の部分画像空間を受け持つことになる。一方、パラメータ空間は短冊状に分割され、同じ列に属するPEには同じ短冊領域が割り当てられる。すなわち、アドレス(i, j)のPEは左上が(γ₀, ωⱼ * kω / √N)、右下が(γ_{kγ-1}, ω_{(j+1) * kω / √N} - 1)で表現されるkγ × kω / √Nセル分の短冊状部分パラメータ空間を受け持つことになる。各PEで部分パラメータ空間に投票後、投票を受けたセルを各列ごとに並列ブロードキャストし、同一セルへの投票を足し合わせることで全特徴点からの投票結果が得られる。

1. 各PEは部分画像から特徴点を抽出し、同じ行に属するPEに並列シフトデータ転送手順でブロードキャストする。
2. 各PEは、短冊状部分パラメータ空間のωの値と特徴点の座標値(x, y)から、γの値を

計算し部分パラメータ空間に投票する。

3. 各PEは短冊状部分パラメータ空間から投票を受けたセルを検出し、同じ列に属するPEに並列シフトデータ転送手順でブロードキャストする。
4. 各PEは同一セルに対する投票値を足し合わせる。

手法3 各PEは全特徴点を保持し、割り当てられた正方部分パラメータ空間に投票する。

この手法では、全PE間で特徴点を並列ブロードキャストし、各PEは画像空間中のすべての特徴点を処理する。一方、パラメータ空間はすべてのPEで独立となるように正方分割し、2次元RTAにマッピングする。すなわち、アドレス(i, j)のPEは左上が(γᵢ * kγ / √N, ωⱼ * kω / √N)、右下が(γ_{(i+1) * kγ / √N} - 1, ω_{(j+1) * kω / √N} - 1)で表現されるkγ / √N × kω / √Nセル分の正方部分パラメータ空間を受け持つことになる。各PEは割り当てられた正方部分パラメータ空間に対する投票結果を保持している。

1. 各PEは部分画像から特徴点を抽出し、並列シフトデータ転送手順でブロードキャストする。
2. 各PEは、正方部分パラメータ空間のωの値と特徴点の座標値(x, y)からγの値を計算し、その(γ, ω)が正方部分パラメータ空間の範囲内であれば投票を行う。
3. すべての特徴点を処理すれば終了する。

手法4 各PEは全特徴点を保持し、割り当てられた短冊状部分パラメータ空間に投票する。

この手法は、手法3と比べてパラメータ空間の分割方法が異なり、パラメータ空間を短冊状分割し各PEに割り当てる。すなわち、アドレス(i, j)のPEは、N > kωのとき、上が(γ₀, ωᵢ * kω / √N)、下が(γ_{kγ-1}, ωᵢ * kω / √N)で表現されるkγセル分の短冊状部分パラメータ空間を保持し、i√N + j > kωとなるアドレス(i, j)のPEは処理を行わない。逆に、N ≤ kωのとき、左上が(γ₀, ω_{(i * √N + j) * kω / N})、右下が(γ_{kγ-1}, ω_{(i * √N + j+1) * kω / N} - 1)で表現されるkγ × kω / Nセル分の短冊状部分パラメータ空間を保持する。

1. 各PEは部分画像から特徴点を抽出し、並列シフトデータ転送手順でブロードキャストする。
2. 各PEは短冊状部分パラメータ空間のωの値と特徴点の座標値(x, y)からγの値を計算し、部分パラメータ空間に投票する。
3. すべての特徴点を処理すれば終了する。

4 性能評価

4種類の並列Hough変換アルゴリズムのそれぞれについて、2次元RTA上で実行した場合の特性をシミュレーションによって求め、比較検討を行った。シミュレーションには、以下のパラメータを用いた。

画像空間 … 512×512 (画素)
 パラメータ空間 … 1024×1024 (セル)
 全画素数に対する特徴点数の比率 = 14 (%)
 データ1個の転送時間: 代入演算時間 = 10:1

この条件の下で、2次元RTAを構成するPEの個数を4×4から128×128まで変化させ、PE数の増加にともなう並列Hough変換アルゴリズムの処理速度の向上比率を求めた。2次元RTAを構成するPE数($\sqrt{N} \times \sqrt{N}$)を横軸、単一PEでHough変換を行った処理時間を基準にして求めた各アルゴリズムの速度向上比(対数目盛)を縦軸として描いたグラフを図8(特徴点が均一に分布している場合)および図9(特徴点が偏って分布している場合)に示す。なおデータの転送時間と代入演算時間の比率は、32ビットプロセッサと32ビット幅のデータ通信線によってRTAを構成した場合を想定し、32ビットデータ1個の転送および代入に要する標準的な時間比率として10:1を設定した。以下、各手法の特性について考察を加える。

手法1は、各PE間で特徴点のブロードキャストを必要としないかわりに、特徴点の有無にかかわらず画像空間中に存在可能なすべてのデジタル直線上を、対応するカウンタが走査する必要があるため、PE数が少ないと処理効率が極端に低下するという性質を持つ。PE数が増加するにしたがって並列化の効果が現れるが、512×512個の画素数に対してPE数が64×64個でようやく他の手法と同等の性能を示し、MIMD型よりはむしろSIMD型の並列計算機に適したアルゴリズムと考えられる[3]。しかしこの手法は、全画素数に対する特徴点数の比率が極端に高い場合や特徴点分布に大きな偏りがある場合にも、処理効率の低下がほとんどないという特徴を持つ。

手法2、3、4はいずれも最初に特徴点のブロードキャストを行うが、手法2では各PEが帯状画像領域の特徴点を保持するのに対して、手法3、4では各PEが全ての特徴点を保持する点が異なる。したがって手法2では、PE数が増加するにしたがって特徴点分布の偏りが分散されることになり、PEと画素が1対1に対応するまで処理速度が上昇を続ける。このことは手法1と同様に、SIMD型の並列計算機に適したアルゴリズムと考えられる[3]。しかし逆にPE数が少ない場合は、特徴点分布の偏りによって一部のPEに負荷が集中することになるので、手法3、4に比べて性能が低下す

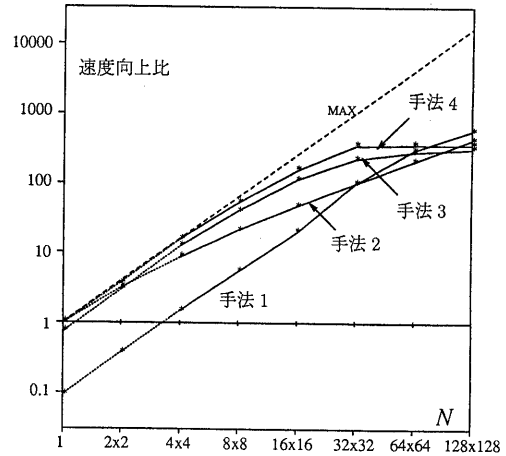


図8: 特徴点が均一に分布している場合

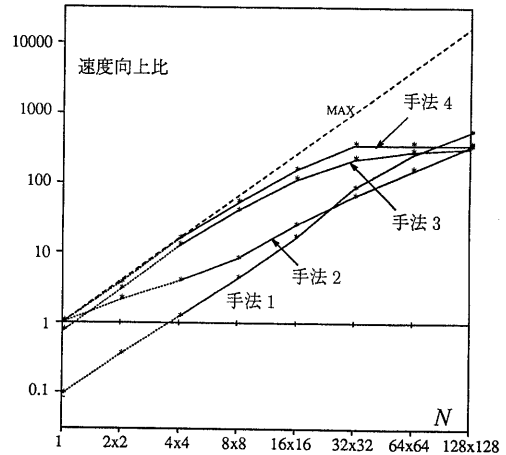


図9: 特徴点が偏って分布している場合

る傾向がある。また手法2では、パラメータ空間が $\sqrt{N}$ 個に分割されているのに対し、手法3では N 個、手法4では N 個または k_w 個($k_w < N$ の場合)に分割されているため、並列化の効果は手法3、4の方が高い。

手法3、4はともに各PEが全ての特徴点を保持するので、パラメータ空間への投票処理は各PEに分散され、特徴点分布の偏りの影響を受けないという特徴を持っている。この両者を比較すると、手法3では特徴点に対応するパラメータ空間上の軌跡を計算しても、PEが受け持つ正方形部分パラメータ空間の範囲に入らない場合は投票が行われない。しかし手法4では、各PEは短冊状部分パラメータ空間を受け持っているため、必ず投票を行うことになる。一方、手法4ではパラメータ空間の w 方向の分割数 k_w より多くのPEが存

在する場合、パラメータ空間を n_w 以上に分割できないため並列化の効果は頭打ちとなる。したがって図8、図9ではPE数が w 方向の分割数に一致する 32×32 個よりPE数が増加しても処理速度の向上は見られない。手法3ではPE数の増加にともない、パラメータ空間を細かく分割し並列度を高めることができるが、各PEが受け持つ正方形部分パラメータ空間が小さくなることで逆に投票効率は低下する。その結果、図8、9のいずれにおいても、手法3は手法4より処理効率が低くなっている。

以上の考察から、2次元RTAにおいてHough変換を高速に行う並列アルゴリズムとしては、手法4が最も有効であると考えられる。すなわち、 $\sqrt{M} \times \sqrt{M}$ 画素の画像空間に対して、シミュレーションと同様に $2\sqrt{M} \times 2\sqrt{M}$ セルのパラメータ空間を用いるとすると、手法4では $\sqrt{2\sqrt{M}} \times \sqrt{2\sqrt{M}}$ 個のPEで最高の性能が得られることになる。 512×512 画素の画像に対しては、2次元RTAのPE構成は 32×32 個で十分であり、約320倍のスピードアップが実現できることになる。この $\sqrt{M}$ と $\sqrt{2\sqrt{M}}$ の関係は、MIMD型分散メモリ並列計算機として構成することを考えている2次元RTAの設計思想と一致し、このことから手法4が2次元RTAの並列Hough変換アルゴリズムとして適していることが分かる。

5 おわりに

本論文では、MIMD型分散メモリ並列計算機アーキテクチャである2次元RTAを用いて並列画像理解を実現するための一連の研究の中で、画像解析レベルの処理としてHough変換に着目し、その並列アルゴリズムを提案し有効性を示した。

まず、各PEから全PEへ並列にブロードキャストするための並列シフトデータ転送手順を明らかにし、その通信オーダがブロードキャスト・バスを利用した場合と同等であることを示した。次に、2次元RTAにおける4種類の並列Hough変換アルゴリズムを示し、シミュレーションによってその性能を定量的に評価した。その結果、2次元RTAとしては「各PEは全特徴点を保持し、割り当てられた罫状部分パラメータ空間に投票する」手法が最も有効であることを確認した。この並列アルゴリズムは、2次元トラス結合アーキテクチャ上で実行できるため、2次元トラス結合を含む2次元RTAにおいて容易に実現することが可能である。本論文によって、パラメータ空間への投票までのアルゴリズムについては明らかになったので、今後はパラメータ空間の極大値検出や線分の抽出などの並列アルゴリズムを開発する必要がある。

また、並列画像理解の実現へ向けたRTAの課

題として、

1. 画像解析レベルの処理として、quadtreeの分散表現や並列ラベリング・アルゴリズムを開発する必要がある。
2. RTAの特徴である動的なスイッチ切り換えを実現するための、スイッチ制御ハードウェアの設計と制御アルゴリズムの開発を行う必要がある。
3. 認識・理解レベルの処理に必要なシンボリックデータの分散表現と並列処理について検討する必要がある。

などを検討中であり、詳細は稿を改めて報告する。

謝辞

2次元RTAにおける並列Hough変換アルゴリズムの基礎的検討に協力いただいた岡山大学工学部情報工学科の清山博文君に謝意を表す。

参考文献

- [1] 松山隆司, 青山正人: 再帰トラス結合アーキテクチャ, 情報処理学会論文誌, Vol.33 No.2, pp.212-222, 1992
- [2] 和田俊和, 藤井高広, 松山隆司: γ - ω 変換—可変標準化による ρ - θ パラメータ空間のひずみの除去と投票軌跡の直線化—, 電子情報通信学会論文誌, Vol.J75-D-II No.1, pp.21-30, 1992
- [3] M.Maresca, et al.: Parallel Hough Transform Algorithms on Polymorphic Torus Architecture, Multicomputer Vision, pp.9-21, Academic Press, 1988