

# 情報伝播方式に基づく並列市街地地図認識

押谷 徹<sup>†</sup> 渡邊 豊 英<sup>†</sup>

本稿では、市街地地図画像から道路情報を自動的に抽出する市街地地図認識に対して、処理速度の向上のために並列処理を適用する手法について述べる。データ並列に基づく並列処理では、プロセッサ間での同期処理や分割による部分画像の境界付近の情報損失が問題となる。これらの問題に対しては、処理可能な部分画像を動的にプロセッサへ割り当てること、および部分画像間での情報の交換が有効である。そのために、分割型多層黒板モデルを導入し、情報伝播に基づくパイプライン的な実行制御法を提案する。すなわち、分割型多層黒板上で下位層から上位層への縦方向の情報伝播と、隣接する部分域間での横方向の情報伝播を定義する。これらの情報伝播によって処理手続きの実行に必要な情報が揃った部分域から動的にプロセッサを割り当てる戦略をとる。最後に、本手法による試作システムを並列計算機上で実装し、その有効性を明らかにする。

## Parallel Urban Map Recognition Based on Information Propagation

TORU OSHITANI<sup>†</sup> and TOYOHIDE WATANABE<sup>†</sup>

In this paper, we address a method to apply the parallel processing to urban map recognition, by which road information is automatically extracted from urban map images, for the improvement of processing efficiency. In the parallel processing based on the data parallelization, there are problems about the synchronization among processors and the information loss in the boundaries of partial images. For these problems, the dynamic allocation of executable partial images to processors and the exchange of information among partial images are effective. So, we introduce the multi-layer partitioned blackboard model and the pipeline-like execution control method based on the information propagation. Thus, we define the vertical and horizontal information propagations on the multi-layer partitioned blackboard. When information which is necessary for the execution of procedure is gathered, the data segment is allocated to a processor dynamically. Finally, we implement the prototype system on the parallel computer and make the effectiveness of our approach clear.

### 1. はじめに

市街地地図認識などのパターン認識では、対象となるオブジェクトの抽出、識別処理において、処理速度の向上が重要な課題の1つである。このための手法として、並列処理の適用がある。パターン認識では、隣接するデータ間で局所的に同一処理が繰り返し実行され、複数の異なる処理手続きによってオブジェクトが段階的に抽出、識別されるという特徴があり、並列実行可能な性質が観測される。この特徴に基づいて、異なるデータに対して同一の処理手続きを適用するデータ並列、同一データに対して複数の異なる処理手続きを適用する機能並列による並列処理が可能であるが、並列化の試みの多くは、入力画像を部分画像に分割

し、部分画像ごとに複数のプロセッサで並列に処理するデータ並列に基づくアプローチであった。

画素レベルの処理は隣接する数ピクセルの画素単位の処理であるが、ベクトル・レベル、記号レベルの処理では、局所的な処理によって得られた情報を統合して目的とするオブジェクトを抽出、識別するので大域的な情報が必要となる。したがって、データ並列に基づくアプローチでは、複数の部分画像のデータを参照する必要があり、部分画像間での同期処理が必要となる。しかし、部分画像に対する処理内容や部分画像中のデータのばらつきによって、部分画像ごとの処理時間が必ずしも均等ではなく、データ待ちなど同期処理によるオーバーヘッドが生じ、処理効率に影響を与える。また、画像の分割によって境界付近の情報が失われ、正しくデータが抽出されなくなる。この境界問題によって、誤認識、未抽出など認識結果に影響を与える。一方、機能並列では各処理手続きが一般に互いに

<sup>†</sup> 名古屋大学大学院工学研究科情報工学専攻  
Department of Information Engineering, Graduate  
School of Engineering, Nagoya University

関連しており、処理手続き間の同期処理が必要となる。したがって、パターン認識での並列処理では、(i) 部分画像間や処理手続き間での同期処理問題、(ii) 分割された部分画像間の境界問題を解決することが重要である。(i)の同期処理問題に対しては、処理手続きの実行に必要な処理データが揃うことによって処理可能となった部分画像からプロセッサへ動的に割り当てるのが有効である。また、(ii)の境界問題に対しては、隣接する部分画像間で情報を交換し、その情報に基づいて部分画像を処理することが必要である。そのために、処理データなどの情報の伝播を定義し、それに基づく部分画像のパイプライン的な実行制御法を提案する。すなわち、画素レベルからベクトル・レベル、記号レベルへの上位レベルの処理データの生成を縦方向の情報伝播、隣接する部分画像間での情報の交換を横方向の情報伝播とする。処理手続きの実行に必要な縦方向および横方向からの情報が揃った部分画像からプロセッサへ動的に割り当てて処理する。

本稿では、市街地地図画像から道路情報を自動抽出する市街地地図認識に対して、並列化を適用する手法について述べる。我々が対象とする市街地地図認識のアルゴリズムは、早川らが提案した手法<sup>1)</sup>を用いる。この手法では、二値化、細線化、線分抽出、候補線分抽出、平行線分対同定、交差点角抽出、交差点同定、道路ネットワーク構築という処理手続きからなり、市街地地図画像から道路片と交差点を抽出し、交差点を起点として道路片を接続していくことによって、道路の接続関係を表す道路ネットワークを構築する。この処理過程において、画素レベルの市街地地図画像は線分などのベクトル・レベルから道路片や交差点の記号レベルへと変換され、最終的にグラフ表現された道路ネットワークとなる。このような市街地地図認識の枠組みに対して、(i)、(ii)の問題を解決し、処理速度向上の視点から効率的な並列処理を実現する。このための基本的なアプローチは、情報伝播に基づくパイプライン的な実行制御であり、黑板モデル<sup>2),3)</sup>に基づいた分割型多層黑板モデル<sup>4)~8)</sup>を提案する。そして、本手法による並列市街地地図認識の試作システムを分散メモリ型並列計算機 AP1000 で実装し、様々な評価実験から本手法の有効性を確認する。

以下、2章ではパターン認識における並列処理の課題とそれに対する解決法と、並列化の概要について述べる。3章では分割型多層黑板モデルとその上での情報伝播に基づくパイプライン的な実行制御法、4章ではシステムの構成法について述べる。そして、5章では並列計算機での実装方法と実験結果を報告し、6章で

まとめと今後の課題を述べる。

## 2. 並列市街地地図認識の概要

### 2.1 パターン認識の並列処理

市街地地図認識のパターン認識の並列化の手法として、市街地地図画像を部分画像に分割し、それらをプロセッサへあらかじめ固定的に割り当て、各プロセッサへ並列に割り当てられた部分画像に対して処理手続きを順次実行するデータ並列がある。これは、パターン認識における処理の局所性に基づくものである。しかし、画素レベルの市街地地図画像から線分などのベクトル・レベル、道路片や交差点、道路ネットワークなどの記号レベルへという段階的な処理において分割された部分画像に対する処理内容や部分画像中のデータのばらつきによって、各部分画像の処理時間に差が生じ、各プロセッサの処理進度が異なる。画素レベルの処理は、隣接する数ピクセルの局所的な処理であるが、ベクトル・レベル、記号レベルの処理では局所的処理によって得られた線分から道路片や交差点を抽出し、それらを統合して道路ネットワークを構築するので大域的な情報が必要となる。したがって、複数の部分画像のデータを参照しなければならないので、各部分画像の処理時間の差からプロセッサ間でのデータ待ちなどの同期処理が必要となる。たとえば、図1のように、プロセッサ1~4へ部分画像a~dをそれぞれ割り当てて処理する。プロセッサ1が交差点角データを生成した後、交差点同定処理を実行するためには、プロセッサ2,3の隣接する部分画像の交差点角データも必要である。したがって、プロセッサ1は、プロセッサ2,3が交差点角抽出処理を終了して、交差点角データを利用できるまで待つ必要がある。同期処理の削減のためには、各プロセッサの処理時間が均等になるように画像を分割することが重要となるが、個々の画像に対して最適な分割方法を決定することは困難

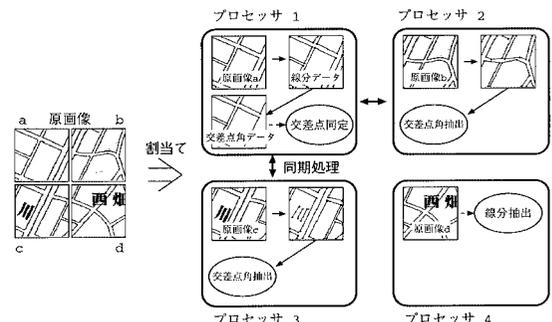


図1 プロセッサ間の同期処理

Fig. 1 Synchronization among processors.



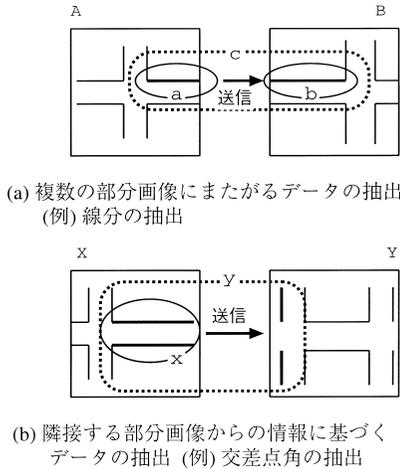


図3 部分画像間での情報の交換

Fig. 3 Exchange of information among partial images.

が存在することを示唆している。したがって、境界付近の情報を隣接する部分画像へ送ることによって、隣接する部分画像では、受け取った情報に基づいてデータを推定し、抽出することが可能となる。図3(b)は、隣接する部分画像からの情報に基づく交差点角抽出の例である。部分画像 X の交差点角抽出処理で用いた線分のうち、境界付近の線分 x から交差点角 y が推定されるので、この線分 x の情報を隣接する部分画像 Y へ送信する。部分画像 Y では、線分 x の情報を用いて交差点角 y を正しく抽出することが可能となる。このように部分画像間で情報を交換することによって、部分画像中のデータを推定することが可能となり、画像の分割によって分断された情報を正しく抽出できる。

以上のような部分画像のプロセッサへの動的な割当て、および部分画像間での情報の交換を実現するために、分割型多層黒板モデルを導入し、情報伝播に基づくパイプライン処理的な実行制御法を提案する。多層構造をなし、各層を部分域へと分割した黒板上で、処理データや部分域の境界付近のデータなどの情報をパイプライン的に上位層の部分域や同一層の隣接する部分域へ伝播する。画素レベルからベクトル・レベル、記号レベルへの上位レベルの処理データの生成による下位層から上位層への情報伝播を縦方向の情報伝播、一方隣接する部分域間での情報の交換を横方向の情報伝播とする。そして、情報伝播に基づいて処理手続きの実行に必要な縦方向および横方向からの情報が揃った部分域から、プロセッサへ動的に割り当てて処理する。

### 3. 分割型多層黒板モデル

情報伝播に基づくパイプライン処理的な実行制御を

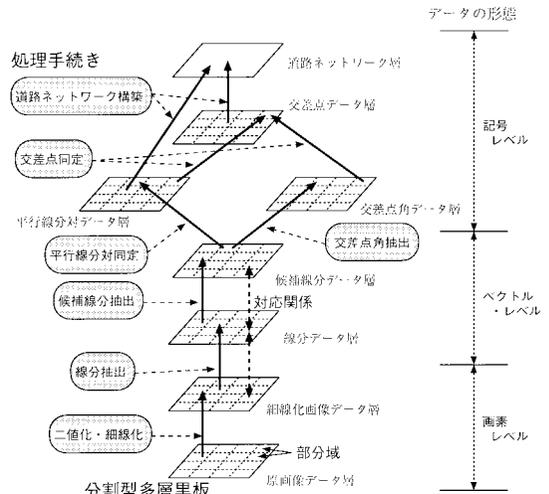


図4 市街地地図認識での分割型多層黒板モデル

Fig. 4 Multi-layer partitioned blackboard model in urban map recognition.

実現するために、黒板モデルを改良した分割型多層黒板モデルを提案する。黒板モデルは黒板を介した複数の処理手続き間での分散協調処理の枠組みを提供するが、実際のシステム構築において、処理手続きの実行制御に関しての規定がない<sup>(11),12)</sup>。したがって、以下の点が重要となる。

- 画素レベル、ベクトル・レベル、記号レベルの異なる形態のデータを保持、管理するための黒板の構造
- 黒板上での情報伝播法
- 情報伝播に基づく処理データと処理手続きの実行制御機構

分割型多層黒板モデルを図4に示す。分割型多層黒板モデルは、分割型多層黒板と処理手続きからなる。分割型多層黒板は様々な形態のデータを保持するために多層構造をなし、また各層は部分域に分割される。一方、処理手続きは分割型多層黒板の各層に対応付けられ、層間に位置する。そして、下位層のデータを参照し、上位層へ処理結果を書き込む。市街地地図認識では、二値化・細線化、線分抽出、候補線分抽出、平行線分対同定、交差点角抽出、交差点同定、道路ネットワーク構築からなる。

#### 3.1 分割型多層黒板

市街地地図認識では、各処理手続きによって処理データは画素レベルからベクトル・レベル、記号レベルへと段階的に変換される。したがって、各処理データを独立に保持し、利用するために多層構造とし、各層が1つの処理データを保持する。図4のように下位層が

ら原画像と細線化画像の画素レベルのデータ，線分と候補線分のベクトル・レベルのデータ，平行線対，交差点角，交差点および道路ネットワークの記号レベルのデータのように 8 層の黑板からなる．また，黑板上の処理データをプロセッサで処理するために，黑板の各層を部分域へ分割する．そして，黑板上の各処理データは部分域で管理し，この部分域がプロセッサへ割り当てられる処理単位となる．

黑板の層間に位置する処理手続きは，下位層の処理データを参照し，上位層の処理データを生成し，上位層へと書き込む．この処理手続きの実行による下位層・上位層の部分域間での処理データの流れを実現するために，黑板の層間で部分域ごとに対応関係を作る．処理手続きが参照する下位層の部分域と処理結果を書き込む上位層の部分域を対応付けて管理し，処理手続きは部分域から得られた処理結果を上位層の対応関係にある部分域へ書き込む．

### 3.2 情報の伝播

分割型多層黑板モデルでは，処理手続きによる下位層から上位層への処理データの流れや部分域間での情報交換による情報の流れが存在する．分割型多層黑板上での縦方向と横方向の情報に対して 2 種類の情報伝播を定義する．

分割型多層黑板の層間に位置する処理手続きは，下位層の部分域の処理データや隣接する部分域からの情報を用いて上位層の処理データを生成し，対応関係にある上位層の部分域へ書き込む．このような下位層から上位層への処理データの流れを縦方向の情報伝播とする．

一方，横方向の情報伝播は同一層の隣接する部分域間での情報交換である．図 3 のように隣接する部分域にデータが続いている場合や，部分域の境界付近にデータが存在する場合，その情報を隣接する部分域へ送信する．

以上のような縦方向と横方向の情報伝播を図 5 に示す．縦方向の情報伝播は  $z$  軸に沿って下位層から上位層へ行われる．一方，横方向の情報伝播は  $xy$  平面上で  $x, y$  軸方向へ行われる．以下，第  $k$  層の  $(i, j)$  番目の部分域を  $(i, j, k)$  と表現する．情報伝播の開始点は最下位層の左上の部分域  $(0, 0, 0)$  とする．この部分域を基準として情報伝播は行われる．

部分域  $(i, j, k)$  における情報伝播は以下のとおりである．

- 縦方向の情報伝播：処理手続きの実行によって得

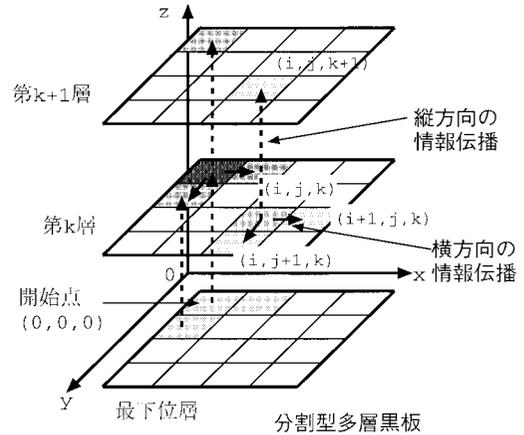


図 5 情報の伝播

Fig. 5 Information propagation.

られた処理データは上位層の第  $k+1$  層と対応関係にある部分域  $(i, j, k+1)$  に書き込まれる．

- 横方向の情報伝播：横方向の情報伝播は開始点  $(0, 0, 0)$  から  $x, y$  軸の正方向へ向かって行われるので，隣接する部分域への情報は同一層の部分域  $(i+1, j, k)$  および  $(i, j+1, k)$  へ送信される．ただし，黑板の最下位層の原画像データ層では，処理内容が二値化・細線化処理であるので，隣接する部分域間で数ピクセル重複して分割する手法を用いる．したがって，最下位層では，隣接する部分画像からの情報は必要ないので，情報伝播は縦方向のみである．

### 3.3 情報伝播に基づく実行制御

同期処理によるオーバーヘッドを減らすために，処理手続きの実行に必要なデータが揃った部分域からプロセッサに動的に割り当てられる．処理手続きの実行に必要なデータは，下位層の対応関係にある部分域からの処理データ，および隣接する部分域からの情報である．したがって，部分域  $(i, j, k)$  の処理可能条件は図 6 のように，

- 下位層の第  $k-1$  層と対応関係にある部分域  $(i, j, k-1)$  からの処理データの書込み
- 同一層の隣接する部分域  $(i-1, j, k)$  および  $(i, j-1, k)$  からの情報の受信

である．ただし，最下位層の原画像データ層では，横方向の情報伝播がないので，市街地地図画像が入力されるとすべての部分域が処理可能となる．

2 つの処理可能条件を満たした部分域は，プロセッサで処理するためにタスクとなる．タスクは処理手続き，処理データおよび隣接する部分域からの情報からなる．タスクはプロセッサが利用可能になると，タスク・スケジューリング方法に従ってプロセッサに割り

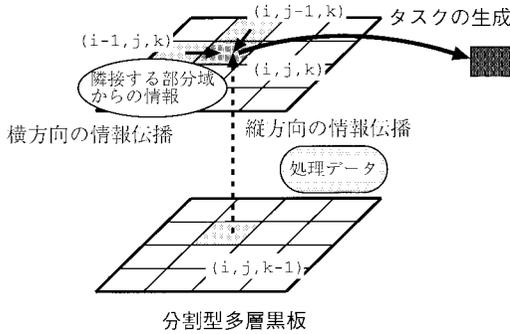


図6 タスクの生成  
Fig. 6 Generation of task.

当てられる。プロセッサは処理手続きを実行し、隣接する部分域からの情報から推定されるデータを探索、抽出し、また処理データを基に上位層の処理データを生成する。タスクの実行後、得られた処理結果や部分域の境界付近の情報を他の部分域へ伝播させる。

#### 4. システム構成

分割型多層黒板モデルにおいて、情報伝播に基づくパイプライン的な実行制御を実現するためには、分割型多層黒板上での情報伝播を監視し、処理可能となっている部分域と処理手続きを適切にプロセッサに割り当てる必要がある。これらの機能を制御システムが実現する。

##### 4.1 制御システム

分割型多層黒板モデルでは、処理手続きの実行に必要な縦方向と横方向からの情報伝播による情報が揃った部分域は処理可能となり、タスクとしてプロセッサで処理される。したがって、分割型多層黒板上の変化に応じてタスクを生成する機構が必要である。また、各プロセッサで並列に実行されている複数のタスクからの情報伝播によって、同時に複数のタスクが新たに生成される。したがって、プロセッサへ割り当てられるまでのタスクの管理や、様々な種類のタスクの中から次に実行するタスクを決定するための実行スケジューリング、およびプロセッサの管理が必要となる。

制御システムは図7のように、黒板モニタ、スケジューリング・キュー、スケジューラから構成される。これらの構成要素の役割は以下のとおりである。

- 黒板モニタ

黒板モニタは黒板上での情報の伝播を監視し、部分域が処理可能条件を満たしたとき、タスクを生成する。そして、タスクの実行管理のために、生成されたタスクをスケジューリング・キューに登録する。

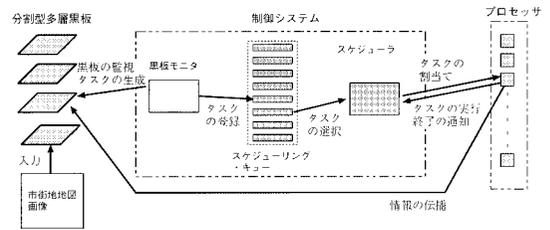


図7 制御システム

Fig. 7 Configuration of control system.

- スケジューリング・キュー

タスクはプロセッサが利用可能となるまで待つ必要がある。スケジューリング・キューは生成されたタスクを保持する。黒板の層ごとにタスクの種類が異なるので、黒板の層ごとにそれぞれ独立したスケジューリング・キューを用意する。そして、タスクは生成された黒板の層に対応するスケジューリング・キューに登録される。

- スケジューラ

プロセッサがタスクの実行を終えると、そのプロセッサへ新たなタスクを割り当てて処理することができる。したがって、プロセッサはタスクの実行後、スケジューラに実行終了を通知する。スケジューラはプロセッサが利用可能となると、タスク・スケジューリング方法に従ってスケジューリング・キューからタスクを取り出し、プロセッサに割り当てる。

##### 4.2 処理の流れ

入力された市街地地図画像が黒板の最下位層へ書き込まれると、この層の部分域は黒板モニタによってタスクとしてスケジューリング・キューに登録される。また、タスクの実行によって発生する情報伝播を黒板モニタは監視し、処理可能条件を満たした部分域をタスクとしてスケジューリング・キューに順次登録する。一方、スケジューラはプロセッサが利用可能となると、スケジューリング・キューからタスクを取り出し、プロセッサに割り当てる。プロセッサが、このタスクを実行することによって、新たに情報伝播が生じ、部分域が処理可能となる。この処理は黒板上に変化がなくなるまで繰り返される。

黒板モニタによるタスクの生成と、スケジューラによるタスクのプロセッサへの割当てによって、最下位層から上位層への縦方向の情報伝播や隣接する部分域への横方向の情報伝播が生じる。このように情報が黒板上を伝播することによって、入力された市街地地図画像から平行線対や交差点などの中間データが段階的に求められ、最終的に道路ネットワークが構築さ

れる。

#### 4.3 タスク・スケジューリング

複数のプロセッサで並列に実行されているタスクによって、情報伝播が生じ、その結果、複数のタスクが新たに生成される。これらのタスクは、プロセッサが利用可能となるまで実行を待つ必要がある。黒板の層ごとに処理手続きが対応付けられ、その処理内容はそれぞれ異なる。したがって、黒板の層ごとに異なる種類のタスクが生成されるので、効率良く並列処理を行うためには、次に実行するタスクを適切に選択する必要がある。

まず、黒板の層ごとに異なる種類のタスクを保持するために、黒板の層ごとにそれぞれ独立したスケジューリング・キューを用意する。タスクは、それが生成された層に対応するスケジューリング・キューに登録される。そして、スケジューラは実行するタスクを選択するとき、スケジューリング・キューを決定し、その先頭からタスクを取り出す。

したがって、タスク・スケジューリングでは、タスクを取り出すスケジューリング・キューを決定する必要がある。そのために、スケジューリング・キューに優先度を設定し、優先度とスケジューリング・キュー中の未処理タスク数に基づいてタスク・スケジューリングを行う。すなわち、優先度の高いスケジューリング・キューを選択し、優先度が同じ場合、未処理タスク数の多いスケジューリング・キューを選択する。

効率良く処理を進めるためには、処理データなどの情報をスムーズに伝播させる必要がある。特に、下位層から上位層の情報伝播では、下位層の処理が進まないと上位層の処理を開始することはできない。したがって、下位層の優先度を高く設定する。しかし、下位層のタスクを優先的に実行することによって、上位層の情報伝播が停滞してしまう可能性がある。そのために、プロセッサで処理されるタスク数に応じて優先度を下げることによって、上位層のタスクも実行できるようにする。

### 5. 試作システムでの実験と評価

市街地地図画像から道路情報を抽出する並列市街地地図認識の試作システムを分散メモリ型並列計算機 AP1000 で実装した。この並列計算機は 1 台のホスト計算機（ホスト）と 64 台のセル計算機（セル）からなる。

#### 5.1 試作システムの構成

本試作システムでは、ホストは制御システムとして、タスクの生成やプロセッサへの割当てなど、タスクの

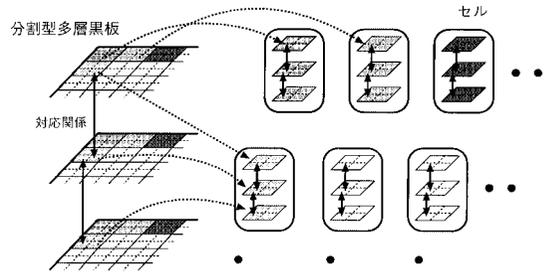


図 8 セルによる分割型多層黒板のデータ管理

Fig. 8 Data management of multi-layer partitioned blackboard by cell computers.

管理、実行制御を行う。一方、セルはプロセッサとしてタスクを実行する。また、タスクは黒板にアクセスする必要があるため、セルが黒板を管理する。

AP1000 は各セルが独立なメモリを持つ分散メモリ型並列計算機であるので、共有メモリである分割型多層黒板を分散メモリ上で仮想的に実現する。まず、黒板上の処理データを各セルのメモリへ分割する。分割型多層黒板は多層構造をなし、各層は部分域に分割され、各部分域は層間に対応関係を持っている。したがって、図 8 のように、部分域ごとに各セルが黒板上の処理データを管理する。このとき、層間に対応関係を保持するために、対応関係にある部分域をすべて同一のセルで管理する。

次に、タスクが各セルに分散して管理されている黒板の部分域へアクセスするために、黒板マネージャを用いる。黒板マネージャは各セルが管理する黒板の部分域の情報を保持する。そして、タスクからの黒板への参照、書込みのアクセス要求に対して、実際に各セルの黒板にアクセスする。すなわち、タスクは黒板へのアクセスが必要となった場合、まず黒板マネージャへアクセス要求を出す。黒板マネージャはアクセス要求のあった黒板の部分域を管理しているセルを調べる。自セルが管理している場合、黒板マネージャはアクセス要求を実行する。一方、他セルが管理している場合、そのセルの黒板マネージャへアクセス要求を出すことによって黒板へのアクセスを実現する。これを図 9 に示す。セル A のタスクが黒板へアクセスするとき、セル A の黒板マネージャへアクセス要求を出す。黒板マネージャは要求された部分域を管理しているセルを調べる。セル A 自身が管理している場合、黒板マネージャが黒板へアクセスする。一方、他のセルであるセル B が管理している場合、セル B の黒板マネージャへアクセス要求を出す。セル B の黒板マネージャは黒板へのアクセスを代行する。すなわち、参照の場合、

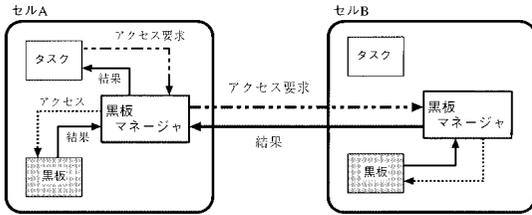


図 9 黒板マネージャを介した黒板へのアクセス  
Fig. 9 Access to blackboard through blackboard manager.

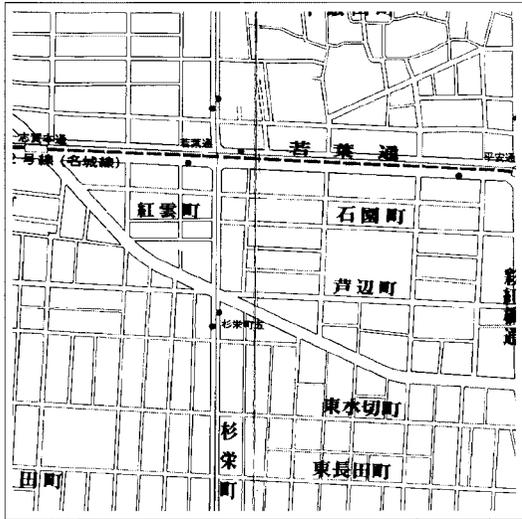


図 10 市街地地図画像の例  
Fig. 10 Example of urban map image.

要求されたデータをセル A の黒板マネージャへ渡し、書込みの場合、渡されたデータを黒板に書き込む。このように、黒板マネージャを介することによって、タスクは黒板上のデータがどのセルで管理されているかを意識することなく、黒板にアクセス可能である。

5.2 実験と評価

並列計算機 AP1000 上で実装した試作システムを用いて評価実験を行った。本実験では、800 × 800 画素の市街地地図画像を 10 枚用いた。また、各画像を 100 × 100 画素の部分画像に分割した。

実験で用いた市街地地図画像の例を図 10 に示す。図 11 は西嶋ら<sup>13)</sup>のプログラムを 1 台のセルを用いて実行した逐次処理によって図 10 から得られた道路ネットワークである。また、図 12 は試作システムによって図 10 から得られた道路ネットワークである。

西嶋らのプログラムは、ボトムアップ処理でまず初期道路ネットワークを抽出し、その道路ネットワークに対してトップダウン処理とボトムアップ処理による協調的推定法を適用しているが、ここでは初期道路ネットワークを抽出するモジュールのみを用いた。

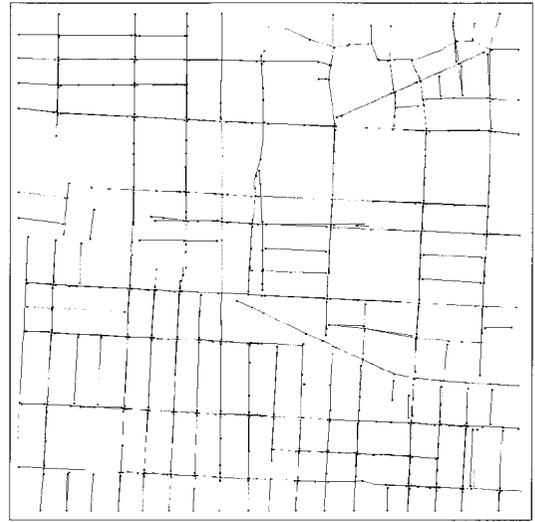


図 11 逐次処理によって得られた道路ネットワーク  
Fig. 11 Road network composed by sequential processing.

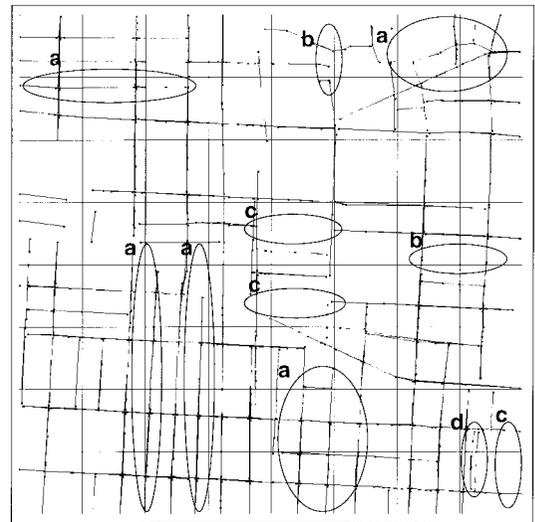


図 12 並列処理によって得られた道路ネットワーク  
Fig. 12 Road network composed by parallel processing.

図 13 は横方向の情報伝播のない並列処理<sup>4)~6)</sup>によって得られた道路ネットワークである。ただし、図 12、図 13 の格子は部分画像の境界を表す。

本システムの評価実験の結果を表 1 に示す。この結果は 10 枚の市街地地図画像の平均値である。また、並列 1 は本手法による並列処理、並列 2 は横方向の情報伝播のない並列処理を表す。スピードアップは逐次処理に対する並列処理の処理速度比を表す。また、認識率は市街地地図画像中の道路片のうち、正しく抽出された道路片の割合であり、誤認識率は誤抽出された道路片の割合である。

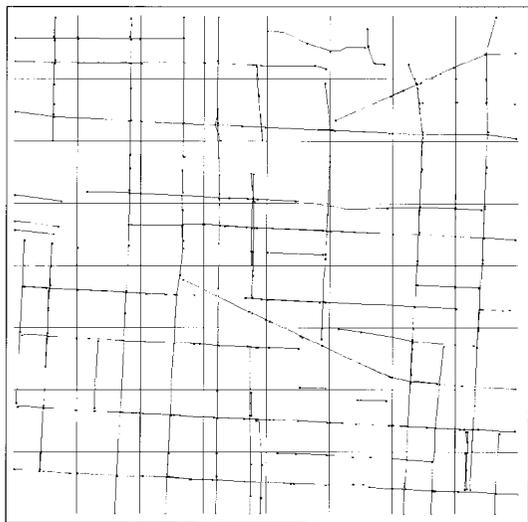


図 13 横方向の情報伝播のない並列処理によって得られた道路ネットワーク

Fig. 13 Road network composed by parallel processing without horizontal propagation.

表 1 処理性能の比較

Table 1 Comparison of processing performance.

	処理時間	スピードアップ	認識率	誤認識率
逐次	536.65 秒	—	80.62%	0.78%
並列 1	25.05 秒	21.42	73.94%	0.85%
並列 2	20.43 秒	26.27	65.38%	0.69%

並列化によって約 21.4 倍の処理速度の向上が得られた。また図 12 a のように、横方向の情報伝播がない場合の図 13 で抽出されなかった境界付近の交差点や道路片が正しく抽出されている。しかし、認識率は逐次処理に比べて多少劣っている。図 12 では、b のように道路片が部分画像の境界と重なることによって正しく抽出されなかったり、境界付近の交差点が正しく抽出されず、その結果、道路ネットワークを構成することができなかった。このように、境界付近の情報の抽出はまだ不十分であるので、改善する必要がある。また図 12 c のように、横方向の情報伝播のない場合 (図 13) で抽出されている道路片が抽出されなかった。一方、誤認識率は逐次処理と並列処理ではほぼ同等であった。誤抽出の例として図 12 の d がある。これは平行線対の幅の間隔で複数の線分が並んでおり、局所的に見るとどれが道路を表す平行線分の対であるかを正しく認識することができなかったからである。これは逐次処理 (図 11) でも見られた。

次に、セル数の変更による台数効果を調べた。これを図 14 に示す。この図から、セル数の増加に対して

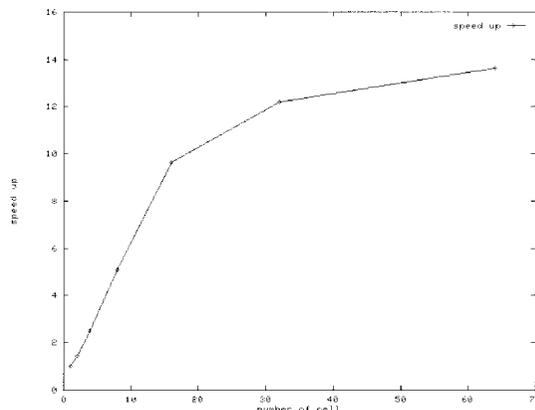


図 14 セル数による処理速度

Fig. 14 Processing speed by number of cell computers.

処理速度の向上率は低下していることが分かる。また横方向の情報伝播のない並列処理に比べて処理速度も低下している。これは、部分域が処理可能となる条件が、下位層からの処理データと隣接する部分域からの情報であり、これらの情報が揃わないと部分域を処理することができず、その結果、データ待ちの部分域が多くなったからであると考えられる。現在の実装では、情報伝播の開始点が最下位層の左上の部分域であり、情報の伝播方向は右下への一方向である。よって、ある部分域でデータ待ちが起こると、情報の伝播が停滞してしまう。その結果、遊んでいるセル数の割合が増加し、処理速度の向上があまり大きくなかったと考えられる。したがって、情報伝播の方法を改善する必要がある。

## 6. おわりに

本稿では、市街地地図認識に対して並列処理を適用する手法について述べた。市街地地図認識などの並列パターン認識において問題となるプロセッサ間の同期処理によるオーバーヘッドや画像分割による境界付近の情報損失に対して、分割型多層黒板モデルを導入し、情報伝播に基づくパイプライン的な実行制御法を提案した。同期処理問題に対しては、処理可能な部分域と処理手続きを動的にプロセッサへ割り当てること、部分画像間の境界問題に対しては、隣接する部分域間での情報交換が有効であることを示した。本モデルでは、黒板の下位層から上位層への対応関係にある部分域間での縦方向の情報伝播と、同一層で隣接する部分域間での横方向の情報伝播を定義し、縦方向と横方向からの情報伝播によって処理手続きの実行に必要な処理データが揃った部分域から動的にプロセッサを割り当てる。そして、本手法に基づく試作システムを並列

計算機で実装し、本手法の有効性を確認した。

今後の課題として、まず処理性能の向上があげられる。逐次処理と同等の認識率を得るために、不十分であった境界付近の情報の抽出を改善する必要がある。また、情報伝播方法を改善し、処理速度を向上する必要もある。さらに、並列処理においても逐次処理においても完全な道路ネットワークは得られていない。市街地地図画像から道路要素を抽出し、それらを統合して道路ネットワークを求めるボトムアップ処理だけでは、完全な道路情報を得ることは困難である。そのために、逐次処理では、仮説の生成・検証<sup>14)</sup>や未抽出道路の協調的推定<sup>13)</sup>などのトップダウン処理、協調処理によって、道路ネットワークの修正や未抽出道路の推定による道路ネットワークを精練する手法が提案されている。黑板モデルでは、解を求めるボトムアップ処理と解を精練するトップダウン処理を状況に応じて使い分けることが可能である。したがって、より精度の高い道路情報を得るために、道路ネットワークの推定や修正などを行うトップダウン処理や協調処理も含めた並列処理を実現する必要がある。

謝辞 日頃からご指導、ご教示いただく中京大学大学院情報科学研究科・福村晃夫教授、本学大学院工学研究科・稲垣康善教授、鳥脇純一郎教授に深謝するとともに、研究室の皆様へ感謝します。

### 参 考 文 献

- 1) 早川卓哉, 渡邊豊英, 吉田雄二, 川口邦雄: 市街地地図からの道路ネットワーク情報の抽出, 電子情報通信学会論文誌(D-II), Vol.J74-D-II, No.6, pp.757-765 (1991).
- 2) 石田 亨, 片桐恭弘, 桑原和宏: 分散人工知能, コロナ社 (1996).
- 3) Barr, A., Cohen, P.R., Feigenbaum, E.A. (編), 淵 一博 (監訳): 人工知能ハンドブック第IV巻, 共立出版 (1993).
- 4) 押谷 徹, 渡邊豊英: 市街地地図認識の並列処理, 電子情報通信学会技術研究報告, Vol.PRMU 98, No.70, pp.93-100 (1998).
- 5) 押谷 徹, 渡邊豊英: 分割型多層黑板モデルを用いた市街地地図認識の並列処理, 並列処理シンポジウム JSP'98, pp.271-278 (1998).
- 6) Oshitani, T. and Watanabe, T.: Parallel Map Recognition Based on Multi-layer Partitioned Blackboard Model, *Proc. ICPR'98*, Vol.II, pp.1604-1606 (1998).
- 7) 押谷 徹, 渡邊豊英: パイプライン制御による市街地地図認識の並列処理, 電子情報通信学会技術研究報告, Vol.PRMU 99, No.47, pp.1-8 (1999).

- 8) Oshitani, T. and Watanabe, T.: Parallel Map Recognition by Pipeline Control, *Proc. ISHPC'99*, pp.336-343 (1999).
- 9) Virginie, M.P. and Serge, M.: Data Allocation Strategies for Parallel Image Processing Algorithms, *Int'l Jour. of Pattern Recognition and Artificial Intelligence*, Vol.9, No.4, pp.615-634 (1995).
- 10) 長谷川純一, 興水大和, 中山 晶, 横井茂樹: 画像処理の基本技法 技術入門編, 技術評論社 (1986).
- 11) Wang, L., Asakura, K. and Watanabe, T.: A Parallel Map Recognizer, Based on Blackboard Model, *Proc. VSMM'96*, pp.313-316 (1996).
- 12) 角 保志: 画像理解システムにおける機能分散的協調処理, 人工知能学会誌, Vol.9, No.5, pp.631-636 (1994).
- 13) 西嶋征和, 渡邊豊英: 道路情報自動抽出における未抽出道路の協調的推定法, 情報処理学会論文誌, Vol.38, No.10, pp.1956-1965 (1997).
- 14) 早川卓哉, 渡邊豊英, 杉江 昇: 仮説の生成・検証パラダイムに基づいた市街地地図からの道路情報の抽出, 情報処理学会論文誌, Vol.35, No.1, pp.62-78 (1994).

(平成 11 年 8 月 31 日受付)

(平成 12 年 2 月 4 日採録)



押谷 徹 (学生会員)

1998年名古屋大学工学部情報工学科卒業。2000年同大学院工学研究科情報工学専攻博士前期課程修了。同年富士通(株)入社、現在に至る。在学中、市街地地図認識の並列処理の研究に従事。



渡邊 豊英 (正会員)

1948年生。1972年京都大学理学部卒業。1974年同大学院工学研究科数理工学専攻修士課程修了。1975年同博士課程中途退学。同年京都大学大型計算機センター助手。1987年名古屋大学工学部情報工学科助教授。現在、同大学院工学研究科情報工学専攻教授。京都大学工学博士。統合化環境, 分散協調環境, データベース環境, データベースの高度インタフェース, 知的CAI, 文書理解, 地図理解に興味を持つ。電子情報通信学会, 日本ソフトウェア科学会, 人工知能学会, ACM, IEEE Computer Society, AAAI 各会員。