

階層化並列バス構造マルチマイクロコンピュータ システムの構成と画像認識への応用

Design of a Multi-microcomputer system based on a hierarchical parallel bus structure and its application to image recognition

木村通秀
Michihide KIMURA

亀山充隆
Michtaka KAMEYAMA

樋口龍雄
Tatsuo HIGUCHI

東北大学 工学部
Tohoku University

1. まえがき 計算機の知的情報処理一般への応用では、演算の主体は非数値計算となり、その中心は記号処理に代表される抽象型データ処理となることがよく知られている。その処理中におけるデータフローは一定にならず、データに依存して変化する。さらに、その演算量も膨大なものとなる。

このような目的に対して、現在の汎用計算機では、数値計算に比べて非数値計算能力が十分でないため、処理速度の点で多々問題が存在する。

本稿では、高速かつコンパクトでコストパフォーマンスに優れ、拡張性を備える密結合マルチマイクロコンピュータシステム(図1参照)を提案する。本システムの最大の特徴は、図2(a)に示すように、同一のグループ内に属する共有メモリとCPU間においては複数の並列バスをもち(以後これを中間バスと呼ぶ)、メモリアクセルルートが複数存在する点である。さらに図2(b)に示すように、これらのグループ間のデータ授受を並列共通バス(以後通信バスと呼ぶ)を用いて行うことにより、階層化されている。これらの並列バス競合をアービタを用いて適切に解消することにより、あたかも電話交換機のようなデータのやり取りを行うことができる。したがって、バス競合による並列実行時のスループットの低下を大幅に軽減できる。

このような特質を生かした応用として、CPU間の協調操作が頻繁に発生する知的情報処理への応用が期待できる。本稿では、システム評価用シミュレーションを目的とした画像認識の例として、黒板モデルを用いたプロダクションシステムによる3次元物体認識システムについて述べる。この処理の特徴としては、画像の前処理に近い低いレベルにおいても、できるだけ知識を用いることで認識率を上げようというもので、領域の統合を仮定し、それを検証しながら推論を進めるといふ、ジグソーパズルを解くようなイメージをもつ画像認識を基本としている。

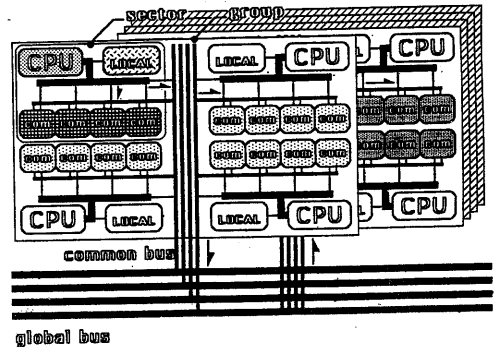


図1. 共通メモリバスシステム

2. システムのアーキテクチャ

本システムのハードウェア上の特長は、次の通りである。

- 1) 汎用のマイクロプロセッサを用いる。
- 2) 共通メモリバス方式に基づく階層化並列バス構造を有する。従って、並列バスによる高速性と、階層化による拡張性をうまく両立させることで、コストパフォーマンスを向上させている。
- 3) 密結合とすることによって、プロセッサ間の大量または頻繁なデータ交換が可能である。

また、ソフトウェア的観点からの特長は、

- 4) 複数のメモリアクセルルートを持つ事で、並列性の効率が上がる。

- 5)メモリ上でアクセス競合を解消することによって、プロセッサ間でのデータが一致する。
- 6)各プロセッサから見たデータのアドレスを一致させることによって、ポインタ操作が容易となる。(ポインタによるデータ渡し、参照など)

本システムの全体を示す。共通メモリバスシステムの構成図を図1に示す。図において、CPU一つ当たりのブロックをセクタと呼び、隣接する4つのセクタでグループを作る。セクタ中において、LOCALと書いてある部分はCPUの専有ハードウェアであり、CPUから出ているバスをCPUバス、comと書かれた共有メモリ直結のバスをlocalバス、グループ中央にある4本の並列バス(中間バス)をcommonバス、グループ間を結ぶ4本の並列バス(通信バス)をglobalバスと呼ぶ。

本システムでは共通メモリバスシステムに複数の中間バスを用意することによってCPU間でバス競合が発生するのを避け、並列性を損なわないことによる高速度(図2a参照)と、共通メモリバスを階層化することによる拡張性を合わせ持つ構成(図2b参照)をとっている。これは従来の共有バス方式におけるスループットの低下や、クロスバー方式における拡張性の悪さを補うものである。

本システムのアクセス方式は、プロセッサと共有メモリの位置関係で次の3通りになる。

- 1)同じセクタのメモリ
- 2)同じグループのメモリ
- 3)他のグループのメモリ

1)は、CPUバスから直接localバスをアクセスできるため、最も早く目的のメモリバスをアクセスでき、優先度も高い。

2)は、一度中間バス(commonバス)をアクセスしてから目的のメモリバス(localバス)をアクセスする。この時、中間バスの中で空いているものを選ぶため、同時に他のプロセッサが別のメモリをアクセスしてもバス競合が起こらない。

3)は、最も遅いバスアクセスである。まず、中間バス(commonバス)の空いているものをアクセスし、次に通信バス(globalバス)をアクセスする。この場合は、中間バスが決まれば通信バスが一意に選択されるため、グループ間バス競合は避けられない。次に、目的のグループの中間バスで空いているものをアクセスし、最後に目的のメモリバスをアクセスする。このバスアクセスは、最速速度というよりも、拡張性に重点をおいて設計されている。

こうした特長を生かすために、平均的に一点に負荷が集中しないような工夫を施した。

第一に、バスセクタなどでは(図3参照)固定順位による優先順位のデコードが行われているので、これをCPU毎、バス毎にずらし、公平な

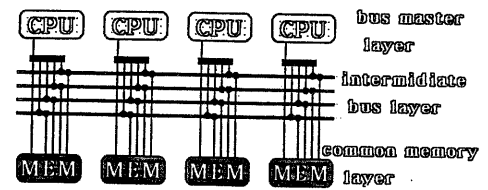


図2(a) 並列バス

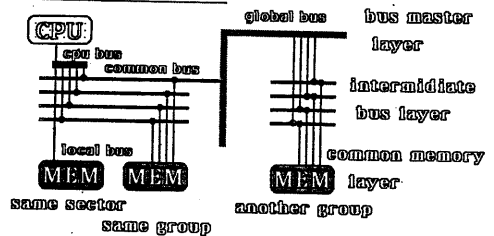


図2(b) 階層化並列バス

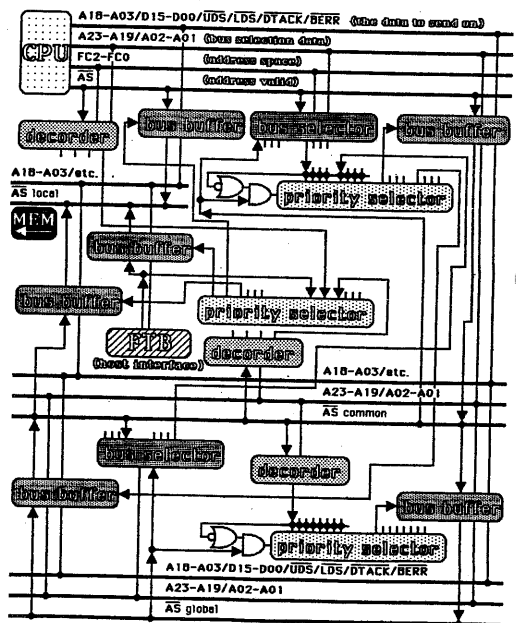


図3 バスアビータ

バスアクセスができるようにした。

第二に、共通メモリバスの数をグループ内のCPU数と同じにすることで、同時に各CPUがメモリアクセスを行ってもバス競合が起こらないようにし、これをメモリ直結のバス(localバス)にも適用した。(図1参照)さらにこのメモリバスのアドレス配置を工夫することで、この特長を生かした。具体的には、アドレスの下位ビットをlocalバスセレクト信号とした。(図4参照)

1グループのCPU数は、2のべき乗をとるとデコードの効率上がるため、ここではプロセッサの物理的制約を考慮して4とした。これより1

グループ内では、中間バス (commonバス) 数4、メモリ直属バス (localバス) 数 $4 \times 4 = 16$ となり、グループ間では、通信バス (globalバス) 数4 (commonバス数と同じ) とした。グループ数はプロセッサのアドレスバスの容量と、プロセッサ一つ当たりのメモリ容量に左右されるが、ここでは最大7グループ、合計最大28個のプロセッサシステムを構成できる。(図1参照)

ここで、従来行われてきた高速化手法との比較を行う。

まず、[データ] キャッシュや [データ] バイプライン等の手法は、密結合マルチプロセッサシステムにおいては、同時に同じメモリに違うプロセッサが読み込みと書き込みを行った場合にデータの正しい通信ができない。(図5参照) また、特定ネットワークを組むことや [データ] バイプライン等の手法は、データフローが固定できる場合のみ有効となる。(疎結合システムにおけるメッセージパッシング等の手法では、有効かも知れない。) また、SIMD型アレイ構造では、単一のデータ構造をとり、データフローを一樣にできないと実現が難しい。これらの理由から、MIMD方式を用いた。

3. システム構成 具体的なシステム構成を図6に示す。CPUはMC68000を4x(1~7グループ)とし、密結合MIMD型階層化並列バス構造を用いる。

メモリの内訳は、

- 1) 専有メモリ
 - [スーパーバイザ]
 - OS用 supervisor main RAM
 - I/O用 BIOS RAM
 - [ユーザ]
 - メインプログラム用 user program memory
 - ローカルデータ用 local data memory
 - ローカルスタック用 local stack memory
- 2) 共有メモリ
 - [ユーザデータ]
 - 各CPUにつき512KB (64Kx4)

共有メモリをユーザデータ領域にハード的に設定したため、システムの一部の暴走からローカルな環境を保護すると共に、共有メモリバスへの負荷を減らしている。メモリマップを表1に示す。付属回路として、

- 1) バスアービタ、バス競合を解消する。
- 2) 割込み制御部、ソフトウェアの同期制御。
- 3) 入出力制御部、初期設定などに用いる。
- 4) ホストインタフェース、データの高速度転送に用いる。(First Transfer Bus)

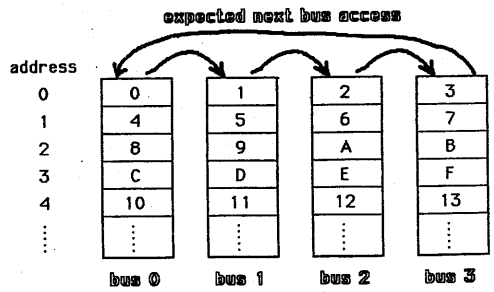


図4 localバスのアドレス配

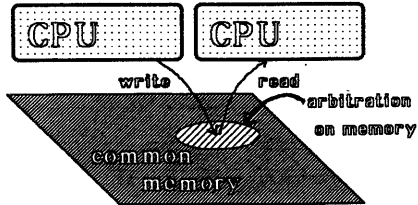
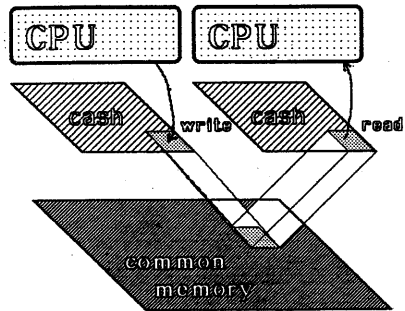


図5 キャッシュを用いた時のデータ誤り(上) 及び本システムでの実現(下)

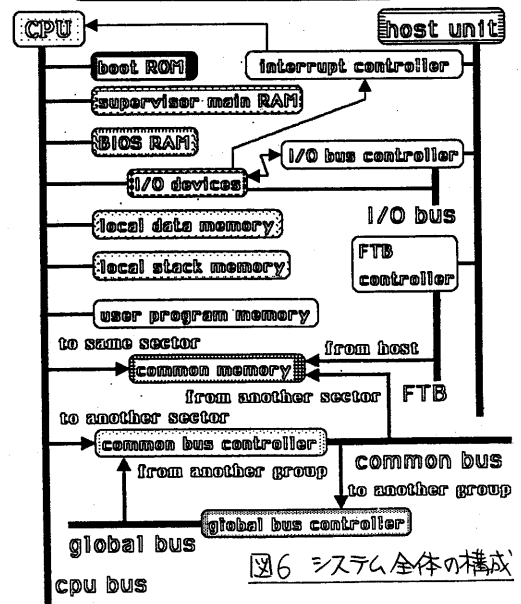


図6 システム全体の構成

共有メモリのソフトウェア的特徴として、どのCPUから見てもデータのアドレスが一致することがある。(図7参照)これにより、ポインタによるデータ渡しが可能で、共有メモリ全域のデータを直接利用可能となる。これから、膨大な相互データの参照が効率的となり、高級言語、例えばC言語などでよく使用されるポインタ操作に適合する。

第二の特徴に、一つのCPU下のメモリバスを1グループ当たりのCPU数(4)と同じにし、そのメモリバスのセレクトにアドレス信号の下位(2)ビットを使用することによって、次のアドレスは隣のバスをアクセスするようにし、優先度の高いCPUのバス独占を防ぎ、パイプライン的に複数のCPUが同一バスをアクセスできるようにした。(図4参照)これは優先順位が固定のデコードを行っているためである。

共通バスのバスアービタを図3に示す。CPUのアドレス確定信号であるASを用いて、非同期にメモリをアクセスする。アクセス方法を述べると、

1) 同じセクタのメモリ

CPU>デコーダ(CPU)>優先順位付きセクタ(local)>バスバッファ(CPU>local)>MEM1

2) 同じグループのメモリ

CPU>バスセクタ(CPU>common)>優先順位付きセクタ(common)>バスバッファ(CPU>common)>デコーダ(common)>優先順位付きセクタ(local)>バスバッファ(common>local)>MEMc

3) 他のグループのメモリ

CPU>バスセクタ(CPU>common)>優先順位付きセクタ(common)>バスバッファ(CPU>common)>デコーダ(common)>優先順位付きセクタ(global)>バスバッファ(global>common)>デコーダ(common)>優先順位付きセクタ(local)>バスバッファ(common>local)>MEMg

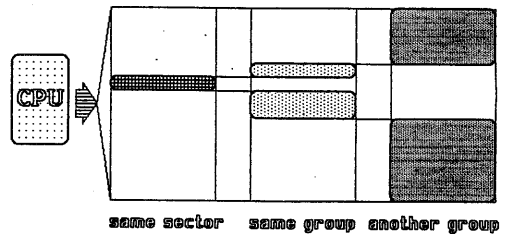


図7 CPUから見た時の共通メモリアドレス

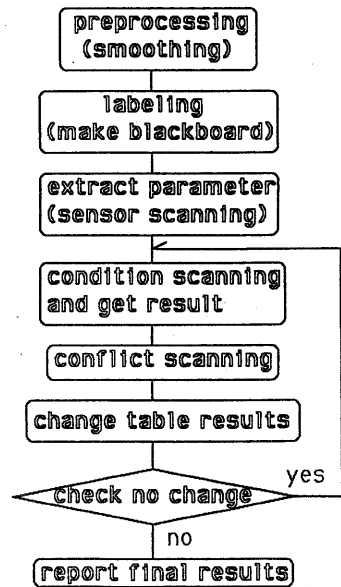


図8 画像認識システムフローチャート

address	user program space	user data space
\$0	user program main area	ex. group0 sector1 common memory same group
\$4		
\$400		
\$1000		
\$40000		
\$80000	extension area for user program	common memory same sector
\$100000		common memory same group
\$200000		common memory another group
\$com_end		unmounted
\$800000		common memory space
\$840000		local data area
\$E00000		local stack area
\$F08000		temporary area
\$FFF000		
\$FFFFFF		

address	supervisor program space	supervisor data space
\$0	reset vector	exception vector
\$4	boot loader	
\$400	supervisor main area (include OS)	
\$1000		
\$40000	extension area for supervisor main	
\$80000		
\$100000		
\$200000		
\$com_end		
\$800000	window for user program	window for local user data area
\$840000		
\$E00000	BIOS reserved	I/O reserved
\$F08000		
\$FFF000		
\$FFFFFF		

表1 システムメモリマップ (a) 実行時(ユーザ) (b) OS (スーパーバイザ)

以上の素子は、PALなどのゲートアレイによって容易に実現できる。

4. 画像認識への応用

4.1 プロダクションシステム 応用分野として、並列性が比較的高く、協調動作が必要で、大量のデータを扱う分野に向くことから、プロダクションシステムを用いた画像処理システムへの応用を考察している。ここでは、並列性の高い処理が可能な黒板モデル³⁾を用いて実現する方法を取り上げる。

プロダクションシステムの基本構成は、

- 1) プロダクション・ルール (PM)
- 2) データベース (WM)
- 3) インタプリタ (PSI)
- 4) センサ (sensor)
- 5) モータ (motor)

からなるが、次の方針でこれを実現する。

1) ルールの評価・実行は各プロセッサが並列に受け持つ。

2) データベースに黒板モデルを用いる。

3) インタプリタはあまり複雑にせず、上位ルール(メタルール)での競合解消を行う。

4) センサについては、一般的な特徴抽出を除いてルールに依存したものが多く、ルールの評価段階で作用させることにし、各ルールはその結果だけを使う。一般的なものは、前処理で黒板の初期設定を行う時に評価してしまうものとする。

5) モータについては、今回は扱わない。

各プロセッサはそれぞれのルールにしたがって並列に各項目を評価し、結果を黒板に書き込む。矛盾の評価は、結果がある程度出た時に、上位のルール(メタルール)でチェックする。その時、矛盾の解消にプロセッサ間で協調操作が必要となる。

プロダクションシステムの適合する分野としては、分岐が頻繁で、コントロールが次の行動決定にあまり重きをおかず、それまでの全体から判断する分野に向いているとされる。

本システムとの適合性について、

1) 基本的に並列処理向きである。特に、ルールの評価および特徴抽出の並列実行が可能である。

2) 競合発生時の解消やルールの選択などの、プロセッサ間の相互データ交換や参照などの協調動作に向いている。

3) 特徴ルール(矛盾に関するものや特殊な操作を扱う)を導入することにより、並列性が向上する。基本的には競合集合を作らず、並列に評価を進め、矛盾が発生してからそれを解消する。矛盾は比較的上位のレベルのルールで検出され、協調操作でそれを取り除く。

4) 黒板モデルは結果の授受に用いられ、領域単位でテーブルを持ち、これを基本として統合を繰り返す、目標の領域を抽出し認識する。(以下、線素を基本として目標物を検定・抽出する方法と対比して、面素認識と呼ぶ)

4.2 処理内容 提案する処理システムの特長には次の点が上げられる。

1) 画像の前処理に近い低いレベルからできるだけ知識を適用することで、領域統合の際に副次的な情報が得られる。これからより複雑なシーン記述が得られ、従来の入力画像と理想的モデルとの橋渡しができる。

2) 画像データを十分利用することによって、認識率を向上できる。

3) より一般性のある形状との照合を可能とする。

4) この処理の目的は、画像中の3次元物体を認識することであるが、途中で得られた結果を再構成してシーンの記述を生成すれば、より柔軟性のある画像理解システムへと発展できる。

なお参考までに、ここでのシミュレーション開発環境について述べる。言語は移植性を考慮してC言語を用いている。画像入力装置からのデータは256x256点モノクローム16階調である。このデータは、ハード的な露光階調補正が行われている。

図8に示すフローチャートの細部を説明する。

1) 前処理(エッジを保ったスムージング)⁴⁾

主としてフィルタリングを行う。ある程度のまとまった基本面素をつくるのが目的でありエッジの特徴を残した基本領域を作ることが目的である。これは9種類のマスクでの分散を求め、最も少ないマスクの平均値を取るというものであるが、詳しい処理内容については、文献³⁾、⁴⁾を参照のこと。

長所として、エッジがぼけないため、領域分割に適していることが上げられる。

処理中の欠点として、繰り返処理を行うと、ノイズの安定している側へ境界がシフトすることがあげられる。このため、あまり処理を繰り返すと、境界が曲がってしまうことがあった。このため、ここでは1回の処理にとどめている。

2) ラベル付け ラベル伝播と付け替えによる方法で、同一明度の連続領域に一つのラベルを与えた。このラベルを基本領域として統合を繰り返す、目的の領域を認識する。(面素認識)新しい領域ができると、ラベルと共に黒板上のテーブルが動的に付加され、基本的な特徴量の評価が行われる。ここでの特徴量とは、領域の位置、面積、明るさ、明度差などの基本的な項目に絞っており、領域統合仮定の基礎データとして用いられる。

3)特徴抽出 これは一般にルールと密接な関係を持つことが多く、ほとんどの特徴量はただ一つのルールに関与するか、又はそのルールの結果および評価における中間副産物(直線の抽出リストなど)を使用することがほとんどである。このため、評価関数のほとんどは特定の特徴抽出ルーチンを呼んでおり、ルールと切り離して考えられない。中間副産物である情報を使用するルールでは、それ以前に既に当該ルールの評価が終わっていないなければならないので、待たされる場合が出てくることになる。ここでは特徴量が評価されるだけでなく、他のルールが使用する副次データを黒板上の領域テーブルに付け加えていく。特徴量として、一般領域では隣接領域リスト、多角形領域では直線リスト、矩形領域や直方体領域では頂点座標リストなどがあげられる。

4)ルール適用 各プロセッサにルールと領域を与えて、評価させる。結果は黒板中の領域テーブルを通じて情報が交換される。ルールはあらかじめ各プロセッサにすべて渡されており、実行中どのルールをどの領域に作用させるかを受け取って、評価にはいる。矛盾を発見したら、その旨を黒板を通じて知らせ、直ちに解消にはいる。これには、領域統合仮定を立てるプロセッサが担当し、矛盾を解消するためのルールを起動する。この矛盾は実際のシステムを形成するルール集合およびモデルと現実の入力画像データとのズレが出るものと考えられ(以下、システム誤差、モデル誤差と呼ぶ)、ルールおよびデータと密接な関係があるため、一般的に取り扱うことができない。

現段階では、前処理段階でエッジを越えて2つの領域が融合してしまった場合を考えている。これを判断するためには、矩形領域のルールで判断するか、直方体領域で判断するのがよいと思われる。

具体的なルールについて述べる。(距離に用いられている単位はピクセルである。)

1)低いレベルのルール(明度によるもの)

2つの隣接領域の明度差が2以下の時、領域統合仮定を認める。(図9.1参照)

2)多角形領域に関するもの(直線によるもの)

領域中の境界線を走査し、直線成分を抽出し、その割合を調べる。(図9.2参照)

3)矩形領域に関するもの(平行線によるもの)

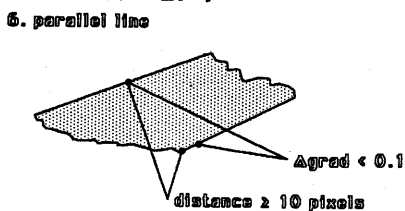
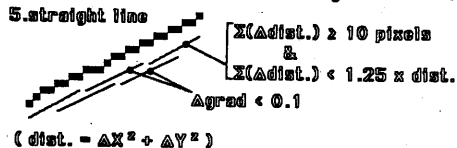
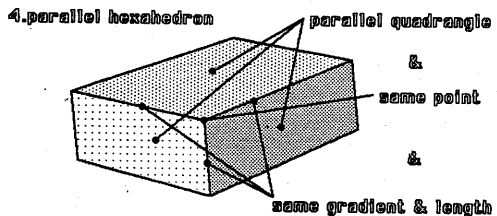
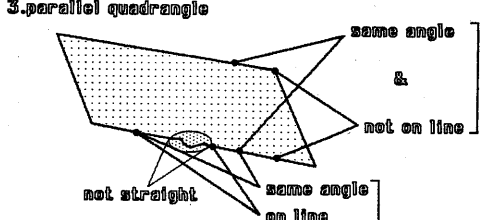
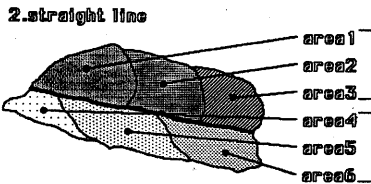
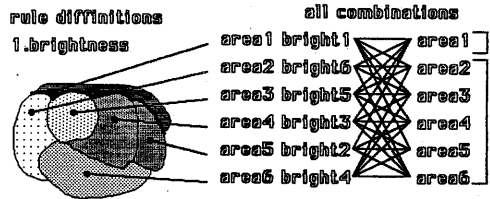
多角形領域において、平行線を調べ、矩形になっているかを調べる。(図9.3参照)

4)3次元物体に関するもの(直方体によるもの)

隣接する3つの矩形領域が、直方体かどうかを調べる。(図9.4参照)

各ルール中で使われている定義に関しては、

5)直線 線分を互いに50%重なり合う長さ10



$$(distance = |ax + by + c| + [\sqrt{a^2 + b^2}])$$

図9 ルール

以上の線素に分けた後、各々の線素の実際の長さと同端の距離の比が1.25:1以内で、一つ前の線素との角度差が0.1ラジアン以内である。(図9.5参照)

6)平行線 傾きが0.1ラジアン以内の2直線で、片方の直線と、もう一方の線分の両端の距離が10以内。(図9.6参照)

これを評価する関数を呼び、結果を黒板上に表示する。

5) 結果の抽出 これは、実際には黒板の表示になっており、結果をディスプレイしたり、記号で出力したりする部分である。これを発展させることによって、画像理解システムへと進むことができるであろう。Waltzのアルゴリズム5)は、この後の段階で付加するのがよいと思われる。

5. むすび 本マルチマイクロコンピュータシステムの効用により、従来処理量が膨大であり、現実的な画像認識がほとんど不可能であるような分野にも適用可能となった。

現在、シミュレーションを進めており、今後報告する予定である。

最後に、本アーキテクチャの開発のご指導をいただいた、本学情報工学科伊藤教授に深謝する。

参考文献

- 1) 木村、伊藤、亀山、樋口 : 階層型並列バス構造をもつ密結合マルチマイクロコンピュータシステムの構成、昭和60年度電気関係学会東北支部連合大会、1C8、pp.85、(1985)。
- 2) 木村、伊藤、亀山、樋口 : 知的情報処理用密結合マルチマイクロコンピュータシステムの構成、昭和61年度電子通信学会総合全国大会、S14-3、pp.6-315~316(1986)。
- 3) 松山、長尾 : 航空写真の構造解析、情報処理、Vol.21、No.5、pp.468-480(May 1980)。
- 4) Nagao.M. and Matsuyama.T. : Edge Preserving Smoothing, Computer Graphics and Image Processing, Vol.9, No.4, pp.394-407(1979)。
- 5) David Waltz : "Understanding Line Drawings of Scenes with Shadows", in The Psychology of Computer Vision, edited by Patric Henry Winston, McGraw-Hill Book Company, New York, 1975.

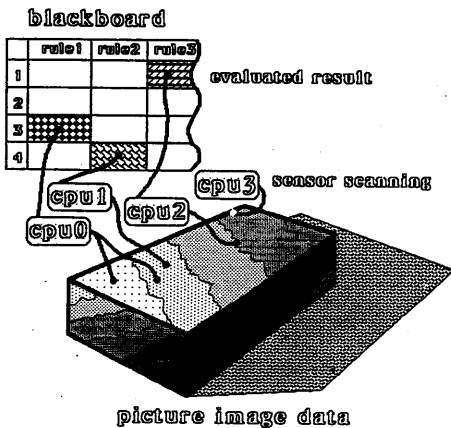


図10 三次元物体認識のイメージ