

Title	移動ロボットのビジョンシステムに関する研究
Author(s)	築山, 俊史
Citation	大阪大学, 1988, 博士論文
Version Type	VoR
URL	https://hdl.handle.net/11094/1387
rights	
Note	

Osaka University Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

Osaka University

移動ロボットのビジョンシステムに関する研究

昭和62年12月

築山俊史

移動ロボットのビジョンシステムに関する研究

昭和62年12月

築山 俊史

通商産業省 電子技術総合研究所

制御部 視覚システム研究室

論文要旨

本論文では自律的な移動ロボットのためのビジョンシステムについて述べる。自律的な移動とは、ロボット自身が環境の変化や自分の位置をなんらかの方法で検知し、その情報をもとに目的地まで移動することである。これはロボットの知能化であるが、このような知能の実現には、

- ① 走行環境の状態や位置の同定のための環境理解、
- ② 最適経路を見つけるための問題解決法、
- ③ 地図情報（ワールドモデル）などの走行環境の知識表現、
- ④ 種々の情報を有機的に統合する管理機構（システム化）

などの課題を解決しなければならない。

移動ロボットの研究は、1960年代の終り、SRIでの研究が始まりといわれているが、研究の流れを見ると、ワールドモデルから最適経路を見つけるような記号処理的な課題は比較的古くから研究されている。それに比べ、走行環境の認識は実際のデータを扱わなければならない、処理装置の制約などから、比較的、その歴史は浅い。走行環境の認識としては、車の移動には障害物の存在とその位置情報が不可欠なことから、距離センサよる障害物の検出法の研究が多く行なわれてきた。しかし、人間の知能と視覚機構の関係を考えると、高度な自律的移動には、TV画像のような情報量の多い2次元データからシーンを大局的に理解して、これをもとに走行に必要な情報を得ることが不可欠と考えられる。TVカメラからの画像データを処理し、対象とするシーンを理解したり、そこに存在する物体を認識することは、コンピュータビジョンなどと呼ばれ、知能ロボットの研究と連携するような形で研究が進んできた。

自律的な移動には、走行環境のTV画像から、2種類の情報を得ることが必要である。1つは進行方向の状態、2つ目はロボットの移動量である。前者は進もうとする方向が走行可能かどうかを、ロボット自身が判断するための

情報である。既に、画像から静止した障害物や走行空間の情報を得る方法が提案されている。しかし、自律的な移動ロボットが有用なのは、人が共存する既存の建物内での種々の作業と考えられる。そのため、走行可能かどうかの判断には人の動きの情報が不可欠である。後者のロボットの移動量とは、指示された目的地とロボットの現在位置の関係をj知るための情報である。正確な移動量の計測には、TVカメラで走行環境内の静止点を定期的に観測し、この位置情報から移動量を計算することが最も有効な方法である。ところで、このような情報をTV画像から得る場合、時系列の画像を処理しなければならないが、扱うデータ量は膨大なものになる。そのため、処理方式の開発や処理の実行を効率よく行う画像処理システムの開発も重要な課題である。本論文では、この3つのテーマ、即ち、①衝突回避のための人の動きの検出法、②画像列からロボットの移動量の推定法、③効率のよい画像処理システムの構成法について考察する。

本論文では、移動ロボットとして車輪のついた作業ロボットあるいは無人の搬送車を考へ、単眼のTV画像データをもとに、建物内のような人工的な環境を移動することを想定している。第1章で、3つのテーマの背景と問題点を概観する。第2章では、画像処理用システムの構成法として、計算機および機能装置をネットワークで相互結合する方式を提案する。また、この考えを具体化した画像処理システム(DIP)について述べる。提案する方式は、任意の装置間で大量の画像データを効率よく転送できるのみならず、各種の機能装置や計算機の追加接続が容易に行える特徴を有する。そのため、広範なロボットビジョンの研究開発および処理の実行が効率よく行なえる。第3章では、ロボットが移動しながら撮像したTV画像から、そのロボットの移動量を推定する方法について提案する。移動量の推定には画像間で特徴点の対応関係と、対応点の位置の情報が必要である。特徴点にはシーン内の物体の頂点を使い、その対応付けにはシーンの中で一般に見られる長い直線をガイドに使う。物体の頂点は入力画像のエッジマップを接合テーブルを使い解析

し、求める。物体の頂点の位置情報はTV画像の逆透視変換により求める。提案する方式では、画像間の見かけの変化が大きい場合でも、対応付けが正しく行なえる。そのため、ロボットがかなり移動しても、その移動量の推定が可能である。第4章では、時系列のTV画像から、人の動きを検出する手法を提案する。人は画像間で見かけの大きさや、形が大きく変化する。そのため、ここでは画像ごとに人を抽出し、その人の実際の位置を計測し、位置の情報から画像間で人の対応付けを行う。3次元運動を扱う動画像処理では、物体の見え隠れの問題があるが、ここでは人の予想される移動速度を使って解決する。簡単なオペレータによる画像処理を示し、また、その処理時間を短くするために専用のハードウェアで画像処理を実行し、実時間処理の可能性を検討する。第5章ではまとめを述べる。

目次

第1章	序論	1
1	はじめに		
2	画像処理システムの構成法		
3	TV画像から移動量の推定		
4	人の動きの検出		
第2章	画像処理システムの構成法	10
1	システムの構成法		
1.1	基本概念		
1.2	モジュール間の関係		
1.3	パラメータ化された制御情報		
2	システムの具体化		
2.1	DIP のモジュール		
2.2	簡易言語		
3	使用例		
第3章	TV画像から移動量の推定	26
1	処理の概要		
2	セグメントの抽出		
3	廊下の境界線の推定		
4	セグメントの解釈		
5	頂点の対応付		
6	実験結果と考察		

第4章	人の動きの検出	49
1	処理の概要		
2	廊下の領域の推定		
3	廊下面と人の領域の分離		
4	人の判定とその位置の推定		
5	人の対応付		
6	実験結果		
7	高速化のための考察		
第5章	まとめ	71
謝辞		73
参考文献		74
付録1	サブルーチンライブラリ概説	80
付録2	DIP 用サブルーチン説明書	82
付録3	DIP を動作させるための制御ルーチンとパラメータ	95
付録4	DIP 制御用マイクロプログラム	98

移動ロボットのビジョンシステムに関する研究

第1章 序論

1 はじめに

最近、マニピュレータ型の作業用ロボット[1]をはじめ、誘導型の搬送ロボット[2]など、工業の分野で多くのロボットが使われている。現在実用化されているロボットの大部分は、プログラムによりその作業内容を変更できる自由度はあるものの、一定の空間内で繰り返し行なう作業に限定されている。しかし、ロボットに2次元の世界を自由に移動できる自律性があれば、作業領域は広範なものとなり、また、作業自体の質的な変化も期待でき、その有用性が一段と増すものと考えられる。自律的な移動とは、ロボット自身が環境の変化や自分の位置をなんらかの方法で検知し、その情報をもとに目的地まで移動することである。これは、とりもなおさず、ロボットの知能化である。ロボットの知能化、特に、自律的な移動の場合、

- ① 走行環境の状態や位置の同定のための環境理解、
- ② 目的地までの最適経路を見つけるための問題解決法、
- ③ 地図情報（ワールドモデル）などの走行環境の知識表現、
- ④ 種々の情報を有機的に統合する管理機構（システム化）

などの課題を解決しなければならない。

自律的な移動ロボットの研究は、1960年代の終り、SRIでの研究[3]が始まりといわれている。この研究では、与えられたワールドモデルをもとに、指定されたゴールまでの最適な経路探索が主なテーマであった。実証実験の

ためにSHAKYとなづけられた移動ロボットが作られ、アルゴリズムの検証が行なわれた。この実験では、ロボットに搭載されたTVカメラと超音波センサを使い、予想される障害物の検出も行なわれた。ワールドモデルをもとにした経路探索は、その後、カリフォルニア大学[4]でも行なわれた。そして、1970年代の終り頃からフランスのLAAS[5]で、また、日立[6]でも経路探索のための問題解決の研究が行なわれている。

1970年代の後半には、JPLで火星探信用ロボットの研究[7]が行なわれた。これは未知の環境を対象としたものであり、距離センサを使い障害物地図を作り、これをもとに走行可能な空間を見つけるものである。環境の3次元情報を得る方法としてレーザレンジファインダ[8]とステレオビジョン[9]が使われた。その後、未知の環境として屋内を対象としたものに、ステレオビジョンによる移動ロボットの研究[10]がある。一般に未知の環境から得た障害物地図は不確実な情報を含んでおり、このような条件のもとでの経路探索の方法[11,12]も研究されている。もし対象となる走行環境が建物内に限定される場合、ロボットが積極的に動き回り、ワールドモデルを作り、これをもとに経路を見つければ、より確実な移動が可能になる。ロボット自身がワールドモデルを作る方法として、超音波センサ[13,14]レーザレンジファインダ[15]、ステレオビジョン[16]による3次元情報を使ったものが提案されている。

1985年頃から、アメリカでStrategic Computing Programと呼ばれるプロジェクトのもとで、陸上を自律走行する車の研究が始まった。実際の地図データベースから車が通行可能な道筋を見つけたり[17]、TV画像から走行に有用な情報を地図情報をもとに抽出するモデルベースの画像処理[18]、TV画像から道路を見つけそれに沿って走行する車の開発[19-21]などが行なわれている。これとは別に、屋内を移動する自律的な車椅子の開発[22]も、最近行なわれている。

移動ロボットの研究の流れを見ると、記号处理的な面が強い課題は比較的古くから研究され、様々な手法が提案されている。それに比べ、走行環境の認識は実際のデータを扱わなければならない、処理装置などの制約から、比較的、研究の歴史は浅い。ロボットの移動には障害物の存在とその位置情報が不可欠なことから、走行環境の認識としては、距離センサ[23-25]による、局所的な障害物の検出法の研究が多く行なわれてきた。しかし、人間の知能と視覚機構の関係を考えると、高度な自律的な移動には、TV画像のような情報量の多い2次元データからシーンを大局的に理解して、これをもとに走行に必要な情報を得ることが不可欠と考えられる。TVカメラからの画像データを処理し、対象とするシーンを理解したり、そこに存在する物体を認識することは、コンピュータビジョンなどと呼ばれ、歴史的には、Robertsの研究[26]に始まる。この研究では単純な物体の濃淡画像から物体の線画を求め、これとモデルとの照合により物体の形状を認識するものであった。その後、物体を表わす2次元の線画の規則性の研究が行なわれ、線画から物体の形状を認識する方法[27,28]が提案された。また、TVカメラで観測される物体表面の明るさ[29]、テクスチャのパターンの見かけの歪[30]や既知図形の歪具合[31]から、それらが描かれている物体の面の傾きを求め、これを手掛りに物体の形状を認識する方法なども研究されている。画像を小領域に分割し、その領域の特徴量とシーンから与えられる知識にもとずいてシーンを解釈する方法[32,33]、時系列の画像を扱う動画像処理[51-53]の研究もある。コンピュータビジョンは、当初、ロボットの知能化のための視覚機能の実現を目指したが、その後、一般的な意味での物体の認識アルゴリズムの研究へ方向が向いた。そのため、処理時間とか、対象となるシーンが実際の問題からかけ離れる傾向にあった。しかし、最近、ロボットの知能化に関心が高まるにつれて、再び、ロボットビジョンを想定した問題が扱われるようになりつつある。移動ロボットのビジョンを想定したものに、画像から道路部分の抽出[18,34]や、室内シーンの解釈[35]がある。

自律的な移動には、走行環境のTV画像から、2種類の情報を得ることが必要である。1つは進行方向の状態、2つ目はロボットの移動量とその方向である。前者は進もうとする方向が走行可能かどうかを、ロボット自身が判断するための情報である。既に、画像から静止した障害物[10]や走行空間の情報[18,34]を得る方法が提案されている。しかし、自律的な移動ロボットが有用なのは、人が共存する既存の建物内での種々の作業と考えられる。そのため、走行可能かどうかの判断には人の動きの情報が不可欠である。後者の移動量とその方向というのは、指示された目的地とロボットの現在位置の関係をj知るための情報であり、定期的に計測しなければならない。正確な移動量とその方向の計測には、走行環境内での不動点をTVカメラで観測し、これを計測の基準に利用することが不可欠である。このような情報を得るには、時系列の画像を処理しなければならないが、扱うデータ量は膨大なものになる。そのため処理方式の開発や処理の実行を効率よく行う画像処理システムの開発も重要な課題である。本論文ではこの3つのテーマ、即ち、①画像列からロボットの移動量の推定法、②衝突回避のための人の動きの検出法、③効率のよい画像処理システムの構成法について考察する。ここでは、移動ロボットとして車輪のついた作業ロボットあるいは無人の搬送車を考え、単眼のTV画像データをもとに、建物内のような人工的な環境を移動することを想定している。この章の後半ではここでとりあげるテーマの背景と問題点を概観する。第2～4章では提案する方式の詳細を、第5章でまとめを述べる。

2 画像処理システムの構成法

ロボットビジョンの実現には、画像演算のみならず、画像データの入力や表示などの機能装置で構成される画像処理システムが必要とされる。しかし、現状では膨大な画像データ処理にシステムが十分に対処しきれておらず、処理速度とシステムの柔軟性の調和がとれたシステムの構成方式の確立が重要な課題になっている。

従来の画像処理システムの構成法について見ると、画像演算用プロセッサを中心に画像データ用メモリと画像データの入力と表示のためのビデオ入出力回路を一体化する方式[36,37]がある。画像演算にはパイプライン式のプロセッサ[38]、専用のハードウェア[39-41]が提案されている。この方式では、画像メモリの共有によるデータ転送の効率化や、演算回路の専用化により、画像処理の高速化を図るものである。しかし、専用回路による演算の種類が限られることと、内蔵の制御用計算機ではプログラム開発環境が十分とはいえず、広範な画像処理を行うのには制約が多い。

他の方法として、汎用計算機のチャンネルや内部バスに各種の機能装置を接続し、システムを構成する方式[42-45]がある。この方式では、計算機を核とする星形のシステム構成になり、接続装置は独立に扱える。そのため、高性能な装置の追加接続が容易に行え、汎用性のある大規模な処理システムの構築がやりやすい。その反面、接続装置間でのデータ転送が、計算機を経由することになり、頻繁な画像データ転送を伴うような処理実験では非常に効率が悪くなる。

画像処理システムの設計に当っては、①汎用性、②拡張性といったいわゆる柔軟性と、③処理速度、の三つの特性の向上に務めることが重要である。ところが、これらの特性の間にはトレードオフがあって、これらの調和をどのような方法で実現するかが、システム構成上の重大な課題である。画像処理で使われる演算は多種多様で、画像を単位とする演算になることが多い。広範な画像処理を高速に行おうとすれば、汎用計算機による処理と演算

ごとの専用プロセッサを併用したり、順次使いわけたりしなければならない。画像処理実験の流れを見ると、画像データの入力、演算、表示を組合せたサイクルの繰り返しになる。そのため、計算機とかプロセッサと他の装置との間で、画像を単位とするデータの受渡しが頻繁におこる。画像が持つデータ量は非常に大きいので、このとき、システム内でのデータ移動量は膨大なものになる。一方、ハードウェアの面からみると、必要とされる計算機や種々の機能装置は互いに異質なため、システムは非均質な構成になる。このようなことから、画像処理の高速化には、処理プログラムに従って、機能の異なる種々の装置間の結合を効率よくスイッチングし、膨大な画像データを高速に受け渡しする機構が重要な役割をはたすと考えられる。またこの機構は機能拡張の面から、機能装置の追加接続のしやすさも考慮されなければならない。

ここで提案するのは、計算機を含めすべての機能装置を一つのネットワークに相互接続し、システムを構成する方式である。装置間でのデータ転送あるいは個々の機能の実行の制御は、このネットワークに流す制御情報をもとに、該当する装置が自律的に行う方式である。これは、従来のような画像演算の効率化を中心にしたもの[37-39]ではなく、計算機を含め各種装置の相互結合に注目したシステム構成法である。この方式ではすべての装置間でのデータ転送は直接シェイクハンドが可能になり、効率のよいデータ転送が実現できる。システムの制御は特定の計算機によらず、また各々の装置の制御機構は独立しているので、専用プロセッサを含め各種装置の追加接続が容易に行える。さらに、現状では、計算機のOSの機能に一長一短があり、ユーザが特定の計算機に依存する割合が多い。このシステム構成法により、異種の計算機が各種の機能装置を共用することが容易に可能となる。

第2章では、ネットワークによるシステム構成法の詳細と、この考えをもとに設計開発した実験システムDIP(Digital Image Processing System)を例にとり、システムの具体化の方法を述べる。

3 TV 画像から移動量の推定

自律的なロボットの移動には、走行環境からの情報をもとに、自分の移動量を推定する機能が要求される。一般に、TV画像から移動量を推定する場合、各画像からシーン内の基準となる特徴点の抽出とその位置の計測、そして、画像間で特徴点の対応関係が必要である。特徴点として既知図形を使えば、各TV画像ごとに、この図形を基準点としたときの車の位置がわかり、移動量の計算が可能である。既知図形に、垂直な平面（無地の壁）に描がかれた長さが既知の菱形図形を使って移動ロボットの位置を求める方法[46]がある。この方式ではTVカメラの光軸は床面に平行で、この図形の中心をさしているものとする。入力TV画像から、この図形の垂直と水平線方向の対角線が検出されその視角が計測される。移動ロボットの位置はこの2つの視角及び既知の対角線の長さから計算できる。このような方法で移動量を求める場合、扱う画像が単純になり、また画像間で特徴点の対応付けが不要になるので、処理時間は短くなる。しかし、一定の走行空間ごとに人工的な特徴点を必要とするため適用範囲に限られる。

シーン内の一般的な特徴点を手掛かりにしてロボットの移動量を求める方法[10]も提案されている。この方式では位置の異なる2点でとった2組のステレオ画像を使う。まず、各観測位置で得られたステレオ画像から、インタレストオペレータにより、シーン内の物体の角に相当する画素（特徴点）が抽出される。ステレオ画像間でこれらの画素の対応付けが行なわれ、三角測量によりそれぞれの座標系での特徴点の位置が計算される。次に、最小二乗法をもちいて、1番目と2番目の座標系の間で、観測された特徴点の位置のずれが最小となるように座標系の変換を行なう。この変換量からロボットの移動量がわかる。この方式は特別な図形を必要としないので適用範囲は広い。しかし、座標系の変換が正しく行われるのは、カメラの動きが比較的小さい場合に限られる。これは走行中頻繁に画像を取りこみ、処理しなければならないことを意味する。

汎用性のためにはシーンで一般に見られるような特徴点を用いるのが望ましい。しかし、特徴点の抽出や位置の計測のための処理にかなりの計算時間を必要とする。そのため走行中の撮像回数はできるだけ少ないのが望ましい。しかし、画像のサンプリング間隔が長くなるにつれて、対象物体の見かけの大きさや位置が大きく変わり、画像間で特徴点の対応付けが困難になる問題が生じる。従来の画像のマッチング手法の多くは画像上での対象物体の見かけの大きさや位置の変化が小さい場合を扱っている[47-49]。見かけの変化が大きくなる場合、対象物体に固有な構造情報を手掛かりに使うのが有効である。しかし、2次元の構造情報たとえば画像から得られた対象物体の線画だけでは、画像間の対応付けにあいまいさがのこる[50]。第3章では、画像間の見かけの変化が大きい場合でも有効なように、シーンの3次元の構造情報を各画像から求め、これを使って画像間の対応付けを行い、カメラの動き（即ちロボットの移動量）を求める方法を提案する。シーンの構造の解析には画像の逆透視変換から得られる距離情報を使う。

4 人の動きの検出

人が歩行しているような環境下で移動ロボットが自律走行する場合、衝突回避のために、人の動きは不可欠な情報である。画像から対象物の動きを検出する場合、サンプリング間隔の短い2枚の画像から、画素ごとの速度ベクトルを抽出し、つぎに物体の運動を仮定して得られる拘束条件を使い、物体ごとに速度ベクトルをグループ化し、物体の速度を求める手法がよく用いられる。代表的なものの一つにオプティカルフローによる方法がある。この方法では明度の勾配と画像間の明度の変化の関係から、画素の速度ベクトルを求める。Horn[51]らは同一物体上での画素の速度ベクトルは滑らかに変化するという条件をもうけ、物体の運動を計算する方法を提案している。オプティカルフローは、表面の明るさの変化が滑らかな物体でないと使えないなど、制約が多く、適用範囲が限られる。

画素の速度ベクトルを求める別の方法として照合法 [52] が提案されている。この方法では、まず、画像中から特徴的な点、たとえば物体の角に相当する部分をさがし、画像間でのこの特徴点の対応をとることにより速度ベクトルを求める。つぎに、物体は剛体であるとして同一速度ベクトルをもつ特徴点を集め、移動物体を抽出する。この方法では、物体を表わすのに十分な数の特徴点を選び出すのが困難である。

差分画像から物体の動きを検出する方法も提案されている。差分画像とは、物体の移動によって生じる画像間の差分を表わす画像である。Nagel [53] らは、差分画像を作る方法として、各画像を小領域に分割し、小領域ごとの明るさの平均や分散を使い、基準画像（たとえば一枚目の画像）の小領域ごとの差分を求める方法を提案している。差分画像から移動物体の輪郭を求める場合、かなりの枚数の差分画像が必要となる。また、この間、物体の形や移動方向が一定でなければならない。

移動ロボットの自律走行への応用を想定した場合、実時間処理を行うことが望ましい。しかし、画像処理にはかなりの時間が必要なため、画像のサンプリング間隔をある程度長くしなければならない。しかし、画像間で対象物の見かけのサイズや形が大きく変わる。特に、対象物が人の場合顕著である。ここでは、これらの条件を満たすため、各画像ごとに人である領域を切り出し、その実際の位置を計算し、その位置情報から人の対応付けをする方法を提案する。第4章でこの方法の詳細を述べるが、先述の照合法のように画素ごとの対応づけでなく、画像ごとに対象物を見つけ、その対応付けをするので、対象物体は変形してもよく、また、比較的長いサンプル間隔の画像でも対応付けが可能である。

第2章 画像処理システムの構成法

1 システムの構成法

1.1 基本概念

前章の第2節で述べたように、画像処理システムで要となるのは、機能の異なる種々の装置をスイッチングし、膨大な画像データを効率よく受け渡しする機構である。この機構をネットワークとパラメータ化された制御情報によりソフト的に実現する方法を提案する。画像処理システムの構築は、図2.1に示すように、計算機を含めすべての機能装置をそれぞれモジュール化し、このネットワークで相互接続することにより行なう。モジュール化というのは、ネットワークとの接続口あるいはネットワークでの通信方式を統一するために、機能装置にインターフェイスを付加することである。モジュールへの制御情報はすべてパラメータ化しておく。計算機がユーザの処理プログラムに従い、パラメータ化された数値データをネットワークに送出すと、該当するモジュールのインターフェイスはこの制御情報を取り込み、解読し、指定された動作を開始する。このとき、計算機はモジュール内の制御シーケンスには関与しない。送出するパラメータの組合せにより、二つのモジュールを結合しデータ転送を行わせたり、あるいは、モジュール内でのデータ処理を実行させる。

画像のデータ量は膨大であるので、モジュール間でやり取りをする時、転送を高速に行われなければならない。このため、このシステム構成法では、16ビットのデータバスと複数の制御信号線からなる共通バスでネットワークを構成する。そして、データ転送時のオーバーヘッドを最少にするために、その転送は、一時的にバスを占有する方式で行なう。

1.2 モジュール間の関係

共通バス方式のネットワークでは、画像データの転送や、制御情報のプ

ロードキャストが効率よく行える。さらに機器の増設がハード的に容易であるなど利点がある。その反面、バスへのアクセス競合やバス上でのデータの衝突が問題となる。これを防ぐために、ネットワークに接続されるモジュールを2つに分類し、その役割や関係を次のように制限する。図2.1に示すように、計算機や手動制御回路をACTIVE型(A型)モジュールとし、それ以外の機器をPASSIVE型(P型)モジュールとする。A型モジュールはネットワークにP型モジュールを動作させるための制御情報を送出することができる。他方、P型モジュールはこの制御情報で指定された動作のみを行い、他のモジュールに制御情報を出すことはできない。モジュール間のデータ転送は、任意の2つのP型モジュール間か、制御情報を出したA型モジュールとP型モジュール間に限定する。2つのモジュールはバスを占有し、両者のシェイクハンドで同期をとりながらデータ転送を行なう。

画像処理実験では、プログラム間の頻繁な通信による計算機の協調とか、複数の計算機による並列処理はほとんどない。そのため、ここでは複数のA型モジュールが、P型モジュールを共用する使いかたに限定する。この場合、A型モジュールがネットワークに同時にアクセスし、衝突する可能性がある。これを防ぐために、ネットワークにアクセス制御回路を設けておく。A型モジュールはここに占有宣言を行ってから、制御情報をネットワークに送出すようにする。この結果、常に1つのA型モジュールの要求に同期して、モジュールでの機能の実行や、モジュール間のデータ転送が行われ、競合をさけることができる。

1.3 パラメータ化された制御情報

ネットワークに接続されているP型モジュールの機能や構造は互いに異なっている。計算機が接続機器の内部シーケンスの制御を直接行うのは、システムの効率や拡張性の面で得策ではない。そのため計算機が必要とするP型モジュールに起動をかけると、あとは指定されたP型モジュールが自律的

に制御を行なう。

ところでP型モジュールの選択や組合せ、あるいは、その動作内容はユーザの要求により変動するものである。またモジュールの制御方式はそれぞれ固有のものである。これらを統一した形で扱えるように、P型モジュールへの制御情報はパラメータ化された数値で引きわたす。すべてのA型モジュールのインターフェイスはパラメータをネットワークに送出する機能を、一方、P型モジュールのインターフェイスには、このパラメータを選択的に取り込み、モジュール内の制御を行う機能を設けておく。

P型モジュールの起動は、次の5種類のコマンドを使い、図2.2 に示す手順で行う。

- ① システムの占有要求 (OCP命令)
- ② システムの占有解除 (NOCP 命令)
- ③ システムのステータス情報の読み込み (STS命令)
- ④ パラメータの送出 (PSET 命令)
- ⑤ モジュールの起動 (EXEC 命令)

このコマンドシーケンスと、モジュールインターフェイスの関係は次のようになる。ユーザは OCP命令でアクセス制御回路に占有宣言を行う。もし別のA型モジュールがネットワークにアクセスしていなければ、NOCP命令を出すまで、排他的に使える。次に、PSET命令でパラメータをデータバス(16ビット)に送出する。パラメータは上位8ビットがP型モジュール内のレジスタを示すアドレス情報、下位8ビットがモジュールに対する制御情報になる。A型モジュールは制御信号線の1つを使い、全モジュールにパラメータを送出したことを知らせる。この信号により8ビットのアドレスで指定されモジュールのレジスタはバスからのデータを取り込む。パラメータは1個とはかぎらず、画像メモリモジュールの場合、例えば、入出力モードを決めるパ

ラメータや、画像メモリのアクセス領域を決めるパラメータなどが必要である。次に、Aモジュールから、バスの制御信号線の1つを使って、EXEC命令が出されると、パラメータが設定されているモジュールは動作をはじめ。例えば、モジュール1とモジュール2にそれぞれ入出力するように指定されていれば、両モジュールは、シェイクハンド方式でデータ転送を開始する。A型モジュールが他のP型モジュールとデータ転送を行う場合、A型モジュールのインターフェイス自身にも同様のパラメータを設定する必要がある。指定したモジュールが所定の作業を終了したかどうかは、STS命令でステータス情報を読み込むことにより知る。

2 システムの具体化

2.1 DIPのモジュール

このシステム構成法では、モジュールの組合せで、必要な処理機能を実現する。そのため1つのモジュールが1つあるいは複数の機能を分担することになる。処理機能をどのように分割し、1つのモジュールとするかは、個々のユーザの想定している処理システムに依る。しかし、モジュール化の基準として次のようなものが考えられる。

- ① 既存の機能装置の場合は、これにインタフェースを付加し、1つのモジュールとする。
- ② 新たに機能装置を設計する場合は、実験に必要な機能を1つ以上実現でき、かつ他のモジュールと機能の組合せがやりやすい単位を1つのモジュールとする。

この節では、TV画像データを扱う処理システムとして設計開発したDIP（図2.3に外観を示す。）をとり、モジュールの具体例およびこれをシステム化する方法について述べる。DIPでは最大16台のモジュールが実装でき、各モ

ジュールのパラメータレジスタは最大15個を想定している。このうち4台がAモジュールとしてアクセス制御回路に登録できる。図2.4にDIPに実装されているモジュールを示すが、2台の汎用計算機(PRIME 750とHITAC 8350)と手動制御回路がAモジュールとして登録されている。次に、このモジュールの機能を述べる。

(1) メモリモジュール

3層の画像メモリと3層のフレームメモリが、それぞれ独立した6個のモジュールとして実装されている。メモリモジュールの画素数は256 × 256個である。画像メモリの1画素は8ビット、そして、フレームメモリの1画素は2ビットである。メモリをそれぞれ独立させたのは、モジュールのインターフェスを簡略化するためと、画像単位でデータを扱いやすくするためである。画像メモリのサイズは標準のTV画像の1フィールドの分のデータ量を基準にして決めた。メモリモジュールの主な役割は次の3つである。

- ① TVカメラからの入力画像データのバッファエリア。
- ② 計算機あるいは専用プロセッサの作業エリア。
- ③ ディスプレイ装置の画像データ用エリア。

①のTVカメラからの入力と、②のディスプレイ装置への出力は、データ転送時間に制約があるので、専用のI/Oポートから行う。これ以外のモジュールとのデータ転送はバス経由で行う。ディスプレイ装置は、常時、各メモリモジュールからデータを読み出し、濃淡画像の合成と表示を行っている。しかし、他のモジュールからメモリモジュールにアクセス要求があると、これが優先される。バス経由でメモリモジュールとデータ転送する場合、メモリ上にアクセス領域をパラメータで指定する。

(2) 画像データ入力制御回路

高解像度TVカメラと標準のTVカメラからの画像データを画像メモリに書き込む時の制御を行うモジュールVDIC(Video Digital Input Controller)である。各画像メモリに対応して、3台のVDICが実装されている。高解像度カメラ(高分解能画像入力装置)というのは、撮像管にカルニコン(1.5インチ)を使用し、この映像信号を1024 x 1024 画素(1画素8ビット)に量子化する装置である。このシステムで高解像度カメラを使用したのは、複雑な3次元シーンにも十分対処できるようにするためである。このTVカメラの画像データは1024 x 1024 画素ある。画像メモリの容量は16分の1であるので、TV画像上の入力したい領域の位置と、そのサイズをこのモジュールにパラメータとして与える必要がある。VDICは送出されてくる全画像データから、指定された領域内のデータを256 x 256 個だけ取り出し、対応する画像メモリに書き込む。このTVカメラはVDIC 1 と 2 へ非同期でデータを送出しており、画像メモリ 1 と画像メモリ 2 への書き込みは独立に行える。

標準のTVカメラというのは、NTSCのビデオ信号を出力するカメラのことである。VDIC 3はこの画像信号を量子化(画素数256 x 256、1画素8ビット)し、後続の画像メモリ 3に書き込む。

(3) ディスプレイモジュール

メモリモジュールの画像データから濃淡画像のビデオ信号を合成するモジュールである。画像処理実験では入力画像データの適性検査や、処理プログラムのデバッグのために、原画像と処理結果の対応がつくと問題点の検討がやりやすくなることが多い。このため、画像メモリの原画像に、フレームメモリの図形データを重ね合せて表示できるようになっている。

フレームメモリのデータが、重ね合わせる原画像に直接関係のない図形(文字とかグラフ)ならば、その部分は原画像の濃淡に無関係に一定の明るさで表示するほうが見やすい。逆に原画像に関係のある領域や線画図形なら

ば、その部分は原画像のレベルに一定の明るさが加算されるような表示の方が判りやすい。このディスプレイモジュールでは、読み出されたフレームメモリの画素のごとに、このような表示方式の切り換えを行なう。切り換え情報はフレームメモリの各画素の上位1ビットにもたせてある。ディスプレイモジュールは同一周期で2つのメモリからデータを読み出し、そのつど濃淡レベルを決めるテーブルの参照と、重ね合せの情報を解釈をしながら濃淡TV画像を合成する。

(4) 画像演算モジュール

2次元の画像データ全域にわたる画一的な演算を高速に行うための専用プロセッサである。このモジュールは既存の演算部[54]に画像メモリとバスインターフェイスが付加されたものである。演算部は積和演算、領域番号付などの5種類の演算回路と制御用プロセッサで構成されている。このモジュールでは専用回路による演算以外に、制御用プロセッサのメモリ領域を拡張してマイクロプログラムによる演算[付録1,4]もできるようになっている。現在、表2.1に示す演算がサブモジュール化され、実行可能である。これらの演算を実行する場合、演算ルーチンのコールと演算に必要な情報を、このモジュールのインターフェイスにパラメータとして与えればよい。

このモジュールの4層の画像メモリ(256 x 256画素、1画素8ビット)は主に画像演算部の作業エリアとして使われるが、先述の画像メモリモジュールと同様、他のモジュールから直接アクセスできるようになっている。またこの4層のメモリにもD/A変換器が接続されており、常時画像表示ができるようになっている。

(5) デジタル入出力ポート

このモジュールは他のモジュールが外部機器とデータ通信を行う時、その入出力制御を行う。たとえばTVカメラの電動ズームをプログラムで制御する

ような場合に使われる。このモジュールには出力回路が2系統、入力回路が1系統ある。各ポートは16ビットのデータ線と3本の制御信号線で構成される。外部機器とのデータ転送の同期はシェイクハンド方式で行われる。

2.2 簡易言語

モジュールの選択とその機能の実行は、第1.3節で述べたコマンドシーケンスで行う。モジュールの選択は、送出するパラメータ[付録3]の組合せで行なう。論理的に矛盾がなければ、この組合せは自由に選べる。しかし、使用頻度の高い基本的な機能は、ライブラリ化[付録1,2]してあり、FORTRANで使えるようにしてある。また、これらのサブルーチンをコマンド形式化することにより、対話的な画像処理が容易に可能となる。ところで、モジュール間のデータ転送あるいは画像演算は、画像を単位として行われることが多い。そのため、画像データを1つの変数名で代表し、この変数名間の演算や代入により画像処理が表現できれば、処理の流れとモジュールとの関係や、その意味が明瞭になる。これは通常の数値演算の変数と同じ形式による扱い方である。この形式で画像演算が扱えるように、DIP用にFORTRANプリプロセッサ(DIPFORT)が用意されている。

DIPFORTでは、まず変数名と画像エリアの対応関係を定義しなければならない。使えるのは計算機のメモリ、フレームメモリと画像メモリモジュール、演算モジュール内の画像メモリである。それぞれメモリの種類ごとに#で始まる文で下記のように宣言しなければならない。

① 画像演算モジュールのメモリとの対応付

PMEM = 変数名 1, ..., 変数名 k

② 画像メモリモジュールとの対応付

DMEM = 変数名 1, ..., 変数名 l

③ フレームメモリモジュールとの対応付

FMEM=変数名 1, ..., 変数名 n

④ 計算機内のメモリとの対応付

CMEM=変数名 1, ..., 変数名 n

①～③の変数名には、宣言順序に従って、物理的なメモリ装置の登録番号が割当てられる。このため、他の文でこの変数に値を代入してはいけない。④の変数名はプログラム内で宣言されている配列(256 x 256語、1語16ビット)でなければいけない。FORTRAN プログラムの中に、宣言された変数の四則演算、代入およびファンクションによる式があれば、DIPFORT はこれを演算順序に従って、実行可能なサブルーチンを生成する。この式はIF文の後半にあってもよい。代入というのはメモリ間でのデータ転送を表わす。ここで四則演算およびファンクションはすべて画像演算モジュールで行われる。もし演算結果を一時蓄える作業エリアが必要な場合、①と②で宣言された変数名のうち、その式の中で使われていないものが割り当てられる。生成されるサブルーチンはファンクションと同じ名称で、引数の数が1つ多い。これは出力エリアの引数が最後に加わるためである。四則演算、代入など基本演算は既に用意されている。ユーザが新たにファンクションを定義する場合、画像演算モジュールでの演算プログラムを前述の形式でサブルーチン化しておけばよい。

3 使用例

実際の処理例は第4章の後半で述べる。ここでは、TVカメラから画像データ(256 x 256画素)を入力し、明るさの変化量を求め、その結果を計算機に転送する例について述べる。DIP での処理手順ならびに1画素あたりの処理時間はつぎのようになる。

- ① VDIC 1を使いTVカメラから画像メモリ1 にデータを入力する
($1 \mu s/1$ 画素)。
- ② 画像メモリ1 から画像演算モジュールへデータを転送する
($1 \mu s$)。
- ③ 積和演算回路で画素(i, j) ごとに水平方向の差分演算を行なう
($1 \mu s$)。

$$D_h(i, j) = \sum_{k=-1}^1 [I(i+1, j+k) - I(i-1, j+k)]$$

- ④ 同じく垂直方向の差分の演算を行なう($1 \mu s$)。

$$D_v(i, j) = \sum_{k=-1}^1 [I(i+k, j+1) - I(i+k, j-1)]$$

- ⑤ 画像演算モジュールのマイクロプロセッサで下記の演算を行なう
($3 \mu s$)。

$$D(i, j) = |D_h(i, j)| + |D_v(i, j)|$$

- ⑥ 画像演算モジュールのメモリから計算機にデータを転送する
($4 \sim 6 \mu s$)。

処理時間にはモジュールに対するパラメータ設定の時間は含まれていない。しかし、パラメータの設定には高々数十バイトのデータ転送しか伴わないので、その時間は $5ms$ 位である。計算機への1画素あたりの転送時間は HITAC 8350の場合、 $4 \mu s$ 、そしてPRIME 750は $6 \mu s$ になる。なお原画像および処理画像は各画像メモリに保存されており、常時、表示されている。

DIPFORT を使って、②から⑥までの処理手順を表記するとつぎのようになる。

```
C
C   Example of DIPFORT
C
C   INTEGER*2 IA(256,256)
C
C   #   PMEM= IP1, IP2, IP3, IP4
C   #   DMEM= ID1, ID2, ID3
C   #   CMEM= IA
C
C   IA= ABS(GRDH(ID1))+ ABS(GRDV(ID1))
C
C   STOP
C   END
```

ここでID1 は画像メモリ 1 を、IAは計算機のエリアをそれぞれ表わしている。ABS は絶対値、そしてGRDH、GRDVはそれぞれ水平、垂直方向の差分を求める関数である。演算はIP1 ~IP4 、 ID2 、 ID3 の画像メモリを作業エリアに使って行われる。

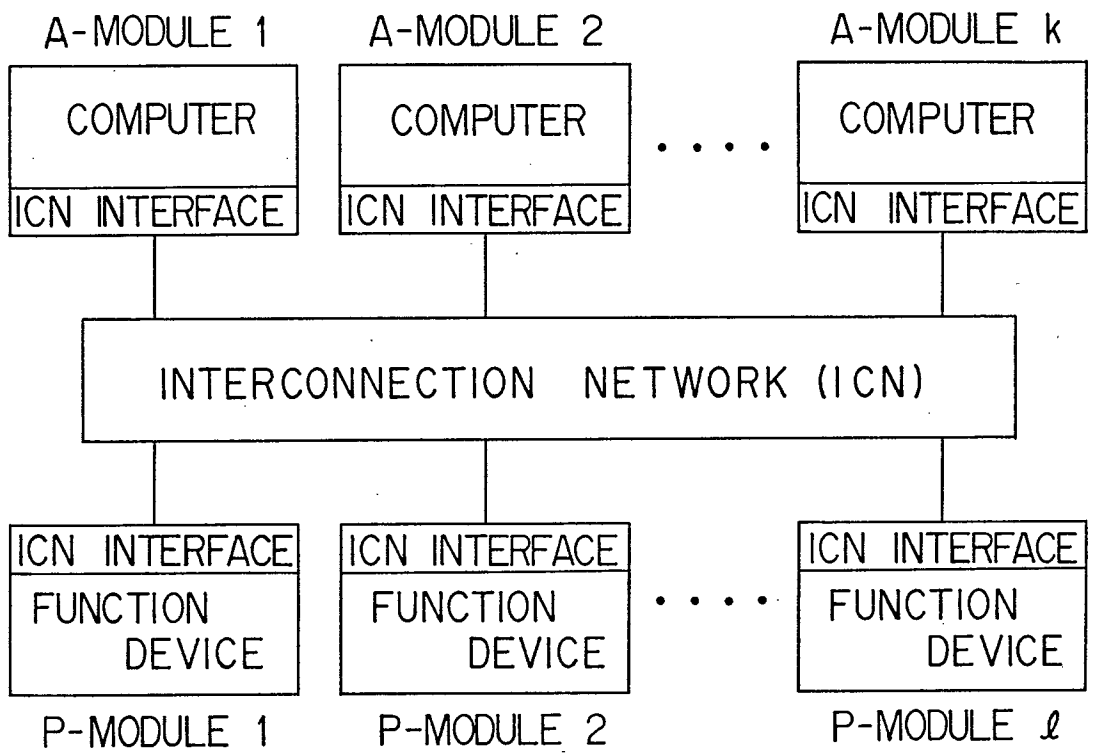


Fig. 2.1 Modular system containing a connection network.

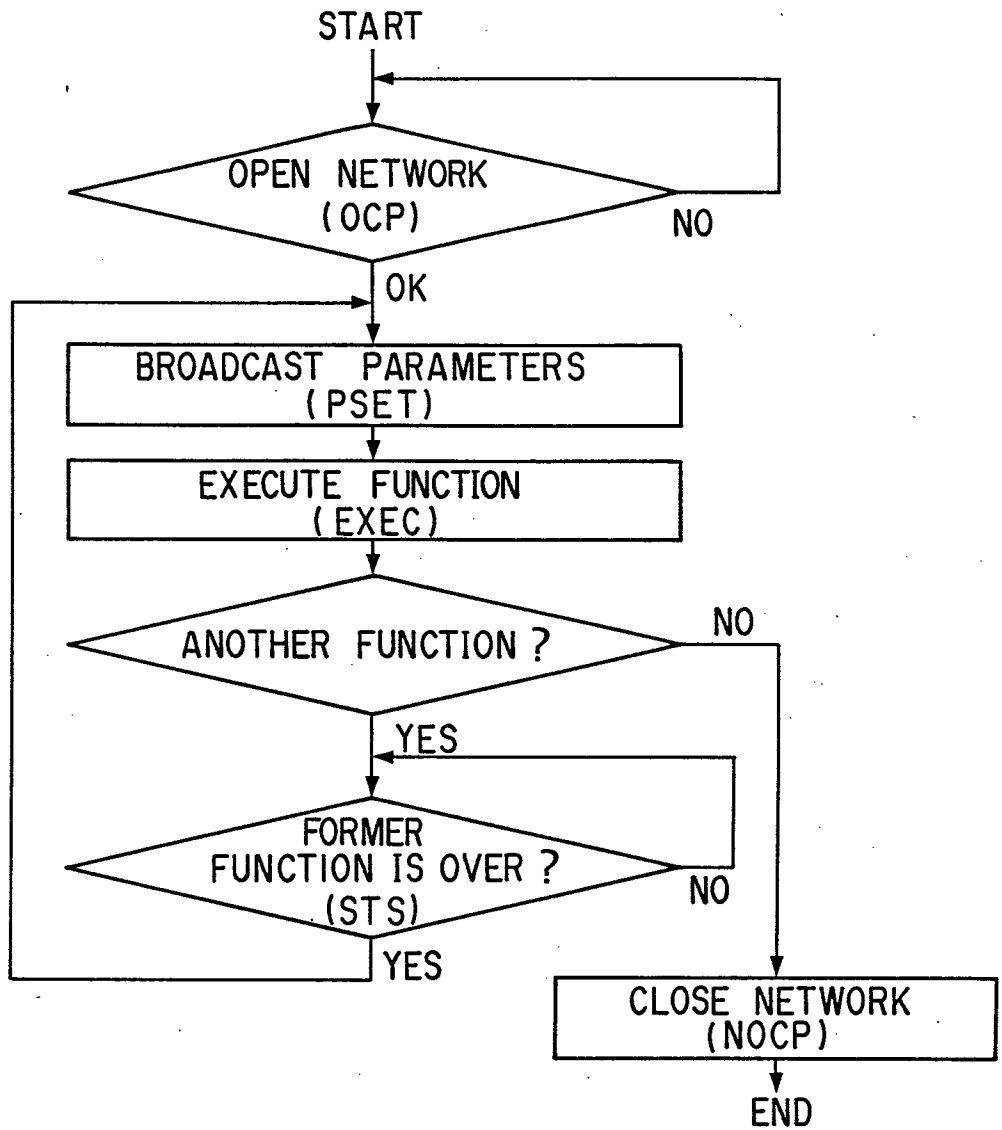


Fig. 2.2 Flow chart of the control sequence.

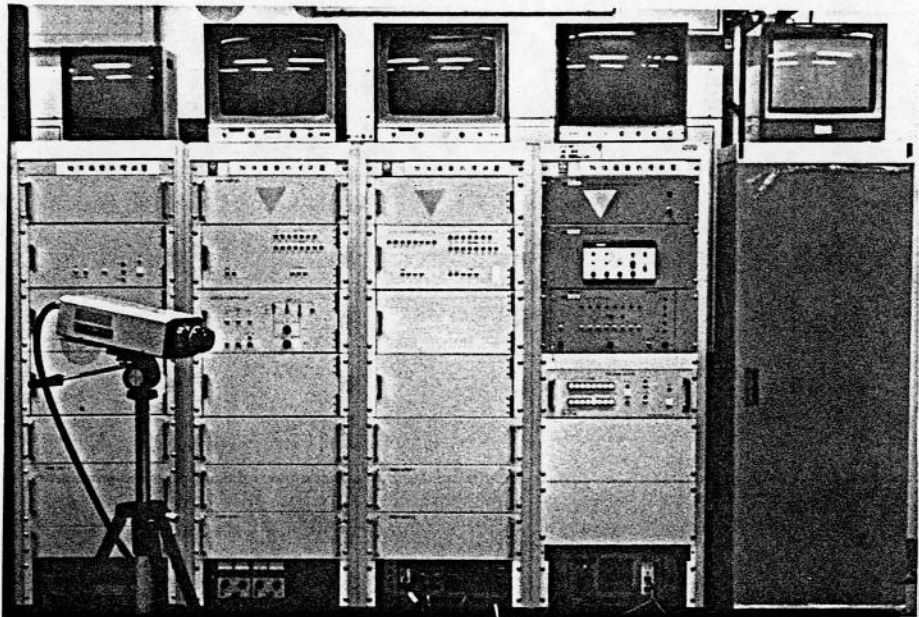


Fig. 2.3 Photograph of the DIP.

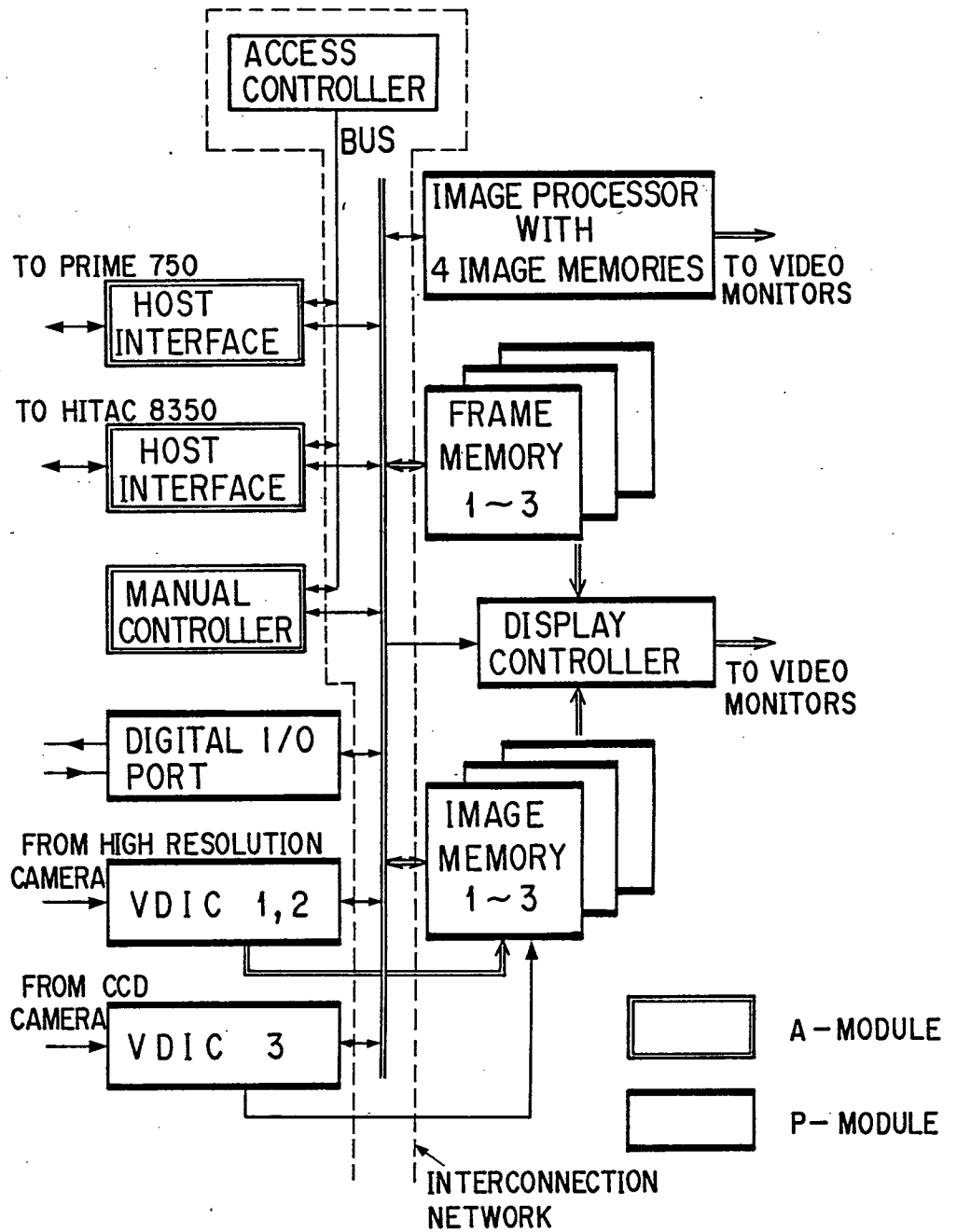


Fig. 2.4 Block diagram of the DIP.

Name of Subroutine	Function
PPPFLT	Filtering
PPPHST	Histogram
PPPDCV	Data conversion
PPPLFL	Logical filtering
PPPLAB	Labeling
PPPSTC	Setting constant
PPPBTS	Bitwise shift
PPPABS	Absolute value
PPPLST	Level shift
PPPADD	Add
PPPSUB	Subtract
PPPMLT	Multiply
PPPDIV	Divide
PPPDCC	Mask-and-convert
PPPOR	Logical OR
PPPAND	Logical AND
PPPXOR	Logical XOR
PPPCMP	Compare-and-convert
PPPRNG	Maximum and minimum
PPPGRA	Gravity
PPPMON	Moment

Table 2.1 List of the subroutines.

第3章 TV画像から移動量の推定

1 処理の概要

本章では、撮像位置の異なる2枚のTV画像データから、そのカメラの動き（ロボットの移動量）を推定する方式を提案する。撮像位置の違いによる画像の見掛けの変化による影響をなくすために走行環境の構造情報を使う。ロボットの走行環境として建物内の廊下を想定しているが、廊下は、通常、水平な平面（床面）と垂直な平面群（壁、柱、戸、箱状の物体など）でできている。壁と床面との間には長い直線の境界線ができ、また、床面上には垂直な平面群による凸状の頂点が多く見られる。これらはシーンのどの位置からでも観測できるので、この境界線を基準線として、また、凸の頂点を特徴点として使う。提案する方式は同一の基準線と特徴点に相当する物を2枚の画像の中で見つけ、その位置の情報からカメラの動き、即ちロボットの移動量を求めるものである。ここで廊下は天井から十分な照明で一様に照らされており、その床面は比較的反射率が高い平面でできているものとする。また、TVカメラは移動前後で同一側の壁面を向いているものとする。

図3.1にTV画像の入力方式とカメラの座標系を示す。TVカメラは h の高さから見込角 θ でシーンの画像を取る。座標系 (x, y, z) は移動するカメラに固定され、 x と y 軸は床面にあり、カメラの光軸は $y-z$ 平面にあるものとする。図3.2に提案する方式の原理を示す。カメラは時刻 t_1 に O_1 の位置から、シーン内の床面上の基準線 g と特徴点 P を観測し、その位置を求める。カメラはロボットと共に移動し、時刻 t_2 に O_2 の位置から、同じ直線と点を同様に観測する。混乱をさけるために、図3.2では z 軸は省略されている。このときカメラの回転および並進成分と観測された点 P の位置の間に次のような関係がある。

$$[x_1 \ y_1 \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -t_x & -t_y & 1 \end{bmatrix} \begin{bmatrix} \cos(-\phi) & -\sin(-\phi) & 0 \\ \sin(-\phi) & \cos(-\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} = [x_2 \ y_2 \ 1]$$

ここで、 (x_1, y_1) と (x_2, y_2) はそれぞれの座標系での点 P の位置を表わす。

(t_x, t_y) はカメラの並進成分で ϕ は z 軸回りの回転角を表わす。

図3.2 より、カメラの回転角 (ϕ) はそれぞれの座標系からみた直線 g の方向 (α_1, α_2) により決まる。カメラの並進成分 (t_x, t_y) はこの回転角と点 P の座標値から計算できる。この境界線や凸の頂点は床面にあるので位置は画像の逆透視変換から求める。即ち、図3.1 で示された平面上の点 P の位置はその画像アドレス (Q)、カメラの高さ (h)、見込角 (θ) などから計算できる [55] ことを利用する。

処理の手順は次のようになる。①まず画像から画素間の明るさの変化量をもとに、境界線と凸の頂点の稜線の候補になるエッジ部分を抽出し、エッジをセグメント化する。②床面上にあるセグメントで、その距離が廊下の最大幅をこえない物を境界線の候補とする。このなかで長さが一定値以上で、その位置が最も遠方にあるセグメントを境界線とする。③セグメントの位置と接続関係から凸の頂点を見つける。ここでは頂点を三面頂点と仮定する。④頂点を形成するセグメントの長さ、角度などを使い、画像間で同じ形状の頂点の対を作る。境界線は画像間で共通であり、この境界線にそった頂点の順序と頂点間の相対的な距離は、見る方向が変わっても不変である。頂点の対のなかでこの制約条件を満足する頂点の対の組み合わせを作る。この組み合わせが画像間で対応付けされた頂点になる。⑤対応付けされた頂点の対と境界線の位置から前述の方式でカメラの動きを計算する。

第2～4節で画像からシーンの構造情報を抽出する方法、第5節ではこの構造情報をもとに、画像間の対応付けの方法を述べる。第6節では実画像を使った実験結果について考察する。

2 セグメントの抽出

この節では、入力画像からエッジ部分を抽出し、これからS、HとVセグメントを決定する方法について述べる。SとHセグメントは床面と垂直な平面との間の稜線に対応する線分のことで、Sセグメントは画像上で傾いている線分である。Hセグメントは水平な線分を意味する。Vセグメントは垂直な平面間にできる稜線に対応している。照明の仮定から、廊下は天井から十分な照明で一様に照らされている。このとき、床面と同じ反射率をもつ平面がその廊下に垂直に立っていたとすると、床面からある程度の高さまで、その垂直面の明度は床面のそれより小さく観測される。ここでさらに、床面は比較的反射率の高い平面でできているとの仮定から、床面はTV画像の中央下部に位置する明るい領域になり、SとHセグメントは、TV画像中で下方から上方へ暗く、そして大きく変化するエッジ部分である。

ここで画像上の座標系の原点を画像の中心にとり、 i 軸は右方向にそして j 軸は上方向に正とする(図3.1)。画像上で (i, j) の位置にある画素の明るさを $I(i, j)$ とする。エッジを検出するために以下に定義するようなオペレータを使い空間微分をおこない、各画素のエッジの強さと方向を求める。

$$D_h(i, j) = \sum_{k=-1}^1 [I(i+1, j+k) - I(i-1, j+k)]$$

$$D_v(i, j) = \sum_{k=-1}^1 [I(i+k, j+1) - I(i+k, j-1)]$$

ここで、 D_h 、 D_v は水平および垂直方向の差分を表わす。エッジの強さ(G_a)と傾き(G_d)には次のような測度を使う。

$$G_a(i,j) = |D_h(i,j)| + |D_v(i,j)|$$

$$G_d(i,j) = D_h(i,j) / D_v(i,j)$$

S と H セグメントの候補となる画素は、前記の照明の仮定から、明るさの変化分 (G_a) が大きく、傾き (G_d) が一定の範囲内あるエッジである。また明るさの変化方向の条件からエッジの D_v 成分が負になる。そのため、(1)、(2)、(3) 式を満足する画素を S セグメントの候補点とする。

$$|G_a(i,j)| > D_1 \quad (1)$$

$$D_v(i,j) < -D_2 \quad (2)$$

$$D_4 > |G_d(i,j)| > D_3 \quad (3)$$

画素が上記の (1)、(2) と次式(4) を満足すれば、その画素を H セグメントの候補点とする。

$$|G_d(i,j)| < D_3 \quad (4)$$

V セグメントの候補点は(5)、(6) 式を満足する画素とする。

$$|D_v(i,j)| < D_5 \text{ and } |D_h(i,j)| > D_6 \quad (5)$$

$$|G_d(i,j)| > D_4 \quad (6)$$

($D_1 \sim D_6$ は正の数でシーンごとに実験できめる。)

次に、それぞれのセグメントの種類ごとにエッジの候補点となった画素の連結性 (4 連結) を調べ、近接する画素どうしグループ化する。セグメントは廊下を構成している平面間にできる綾線に対応しているので、画素数が少

ないグループ、あるいはエッジの傾き方向(G_d)が分散しているグループはセグメントの候補から削除される。残りの各グループは線状領域になるので、この領域の長軸方向の中心線をそのグループを代表するセグメントとする。このセグメントにラベル付けをし、それぞれの種類ごとのリストに登録する。

ところで明るさの変化をもとにしたセグメントの抽出では、得られたSとHセグメントのグループには床面との稜線だけでなく、物体の上辺にあたる稜線も含んでいる可能性がある。そのため、次にセグメントの廊下での位置を計算して、床面上の稜線である可能性が小さいセグメントをリストから削除する。まず、SとHセグメントは全て床面にあるとみなし、それぞれの線分の両端を延長した直線を考える。そして、TVカメラの位置を原点とする座標系を床面上に考え(図3.1参照)、原点からそれぞれの直線までの距離を計算する。距離は逆透視変換を使い求める。距離がかなりある場合、セグメントは床面ではなく、その上方にある(物体の上辺にあたる稜線)と考えられるから、この距離が一定値(廊下の最大幅)を越えるものをセグメントのリストから除く。図3.3(a)は入力画像の例である。同図(b)は画像から抽出されたセグメントを示している。

3 廊下の境界線の推定

この節では、距離情報を使い、SとHセグメントのグループから廊下の境界線を推定する方法について述べる。廊下の境界線というのは、廊下の柱などの部分的なでっぱりを取り除いたとき、側壁に相当する仮想的な垂直の大きな平面と、床面とが交わってできる稜線のことである。この稜線はTVカメラの視軸方向を決めるための基準線としてだけでなく、次節で述べるセグメントの接続関係から頂点を見つけるときの手掛りに、そして2つの画像間での頂点の対応付けのガイドとして使われる。

境界線はTVカメラの方向により画像中では正か負の傾きを持つ直線、ある

いは水平な直線になる。ここで“正”の傾きというのは画像座標系で直線の傾きが正になるものをさす。廊下の境界線の近辺には柱などの物体と床面によりいろいろな方向に稜線ができるが、境界線にそう方向に多くのセグメントがみられる、そのためSセグメントのグループを傾きの正負により2つに分割し、各グループごとにセグメントの長さを合計し、その値が最大になるグループを境界線の候補とする。次に、このグループに属するセグメントの両端を延長した直線を考える。そして、TVカメラの位置を原点とする座標系を床面に考え（図3.1 参照）、原点からそれぞれの直線までの距離を計算する。距離は逆透視変換を使い求める。このグループ内の直線のうち、互いに平行で、その両者の距離が短いのがあれば、もとの2つのセグメントは同じ直線上にあるとみなす。そして、両者を併合し、1つの直線に置き換える。直線上にあるセグメントの線分の長さの合計が一定値以上で、その距離が最大のものを廊下の境界線とする。

4 セグメントの解釈

廊下は3面頂点世界との仮定から、画像に現れるセグメントの接続点のタイプは図3.4 のようになる[55]。接続点Pの物理的形状は、W型とL型では床面にある凸の頂点、Y型では床面と2つの垂直な平面による凹の頂点になる。T型では稜線の一部が垂直な物体で隠されている。この節ではセグメントの接続関係から接続点のタイプを決める方法について述べる。

セグメントの探索は最も確実なセグメント、即ち廊下の境界線になるセグメントから、隣接しているセグメントを順次見つけていく。隣接セグメントの探索は、図3.5 のような限定された領域で行う。この領域内にセグメントの端が見つかれば、両者は接続しているものとして、それぞれのセグメントを延長して交点を求める。この交点を接続点とする。ところでVセグメントは実際よりかなり短くなることが多い。これは、ここで想定している照明条件のもとでは、観測位置によっては、垂直な稜線をはさむ2つの面の明るさ

に差が生じなくなる部分ができ、エッジが完全な形で抽出できないことによる。そのためVセグメントの探索範囲を上方向に長くとってある。

すべてのセグメントの接続点を調べたあと、形状は次の方法で決定する。Vセグメントの下端に2つのSセグメント（あるいはHとSセグメント）が接続している場合、または2つのSセグメント（あるいはHとSセグメント）だけの接続点は、SやHセグメントの床面上の位置関係から、その形状をW型かY型とする。VセグメントとSセグメント（あるいはHセグメント）が接続している場合、完全なセグメントである保証がないので、これだけから形状の判定はできない。たとえば図3.6に示す例では2通りの可能性がある。ここではVセグメントに別のS（あるいはH）セグメントが接続しているかを調べ、図3.7のように左右に1つずつセグメントがあれば、 P_1 をL型、 P_2 をT型とする。これ以外のVセグメントとの接続点や、他との接続が見つからなかったS（あるいはH）セグメントの端点は孤立点とする。図3.8は、図3.3(b)のセグメントの接続関係を表わす。接続点のマークは次のような意味を持っている。+ : W型、□ : Y型、■ : T型、◇ : L型、▲ : 孤立点、H : 画像の境界点。

5 頂点の対応付け

この節では画像間の頂点の対応付けの方法について述べる。対応付けは各画像から得られた頂点の3次元世界での特徴量を使って行なう。

前節で得られた頂点のうちW型とL型の頂点は凸のため、撮像位置が変っても見え隠れする可能性が少ない。これを対応付けのための特徴点として使う。すべてのセグメントが完全な形で抽出されている保証がないので、孤立点とラベル付けされた点も凸頂点である可能性がある。そのため境界線から一定値以上はなれている孤立点も特徴点として使う。画像間での頂点の対応付けのために次のような特徴量を使う。

- ① 境界線から頂点までの距離。
- ② 頂点を形成する左側のS（あるいはH）セグメントの長さ。
- ③ 頂点を形成する右側のS（あるいはH）セグメントの長さ。
- ④ 2つのセグメントによりできる角度。

ここで左側のセグメントというのは観測者からみて左側にあるものをさす。またセグメントの長さ及び角度は3次元世界での量を割り当てる。ところでL型や孤立点の場合、1つのセグメントが隠されるか消失している。この場合、頂点から境界線までの距離をセグメントの長さとする。また頂点がW型の場合でも、もし境界線に垂直なセグメントが小さな物体に隠され、セグメントの1部が欠けていたら、頂点から境界線までの距離をそのセグメントの長さとする。これは画像間での見る方向が大きく変化しても、W型とL型の間での対応付けが正しく行なわれるようにするためである。

対応付けはそれぞれの画像から頂点を1つずつ選び、それぞれの特徴量の差が一定値以下であれば、2つの頂点は対応付けの候補対にする。候補対から、最終的な対応付けは次のように行なう。ここでは、境界線にそった頂点の順序ならびに頂点間の相対距離はTVカメラの位置が変わっても不変であることを利用する。まず対応点の候補は図3.9に示すような座標系 (u, v) にプロットする。 u 軸は1番目の画像中の境界線に対応し、 v 軸は2番目の画像中の境界線に相当する。それぞれの座標軸の原点は、境界線の画像での左端の点である。 u と v の座標値は頂点をそれぞれの境界線へ垂直に投影した点の原点からの距離である。上記の制約条件から正しい対応付けは、傾き1の直線上に在る。それ故に、この傾きを持つ直線群の内、対応点の数が最も多いものを取り出し、その直線上にある候補対を正しい対応点とする。

6 実験結果と考察

ここで提案した方式を様々なシーンを使って検証した。図3.10(a)～(d)は同じシーンを4つの異なる位置から取ったTV画像である。このシーンは研究室の一隅に箱などの物体を置いて、廊下のシーンをシュミレートしたものである。床面はアイボリ系統のプラスチックタイルでできており、照明は天井の蛍光灯による。TVカメラの高さは約1mでその見込角は約17°である。TV画像は512 × 512画素で各画素は256階調である。画像データはVICOM画像処理プロセッサ[56]で量子化され処理された。

図3.11は図3.10(a)を処理し、セグメントを抽出した結果である。マークの付いた頂点是对応付けの候補点を表わしている。ところで、ここで抽出された頂点や廊下の境界線は、画像のノイズなどの影響により、その画像中での位置が多少ずれている可能性がある。また、撮像場所の床面の状態により、TVカメラのパラメータが多少変動しているかもしれない。その結果誤差が生じて、頂点の対応付けや、移動量の計算に影響することが考えられる。通常、パラメータの変動は稀であり、あったとしても、微小変動である。パラメータの変動が微小ならば、すべての頂点の位置が一樣にずれると考えられる。そのために、頂点間の相対距離の誤差は打ち消され、対応付けへの影響は少ない。しかし、移動量に多少誤差が生じる可能性がある。ノイズなどの影響による画像中での抽出点のずれは一律でないので、各頂点ごとに影響の程度が異なり、歪んだ線画になる。そのため、頂点の対応付けと移動量の両方に影響がでる可能性がある。ここでは、頂点の対応付けに関しては、その評価式の条件の設定で、誤差の影響を最小限にした。カメラの移動量は1対の頂点から決定できる。しかし、上記のような原因で、頂点の位置は誤差を含んでいる可能性がある。そのため、得られたすべての対応点ごとにカメラの移動量を計算し、その平均値を最終的な移動量とした。

図3.12は図3.10(a)と(b)との間での対応がとれた頂点を示している。さらに図3.13は図3.10(a)と(c)との間での対応付け、そして図3.14は図3.10

(a) と (d) との間での対応付けの結果を示している。図3.15は図3.10(a) ~ (d) 間でのカメラの移動量を示している。座標系の原点を図3.10(a) でのTVカメラの位置にとってある。各 y 軸の方向はカメラの視軸方向を示している。これらの結果から、見る方向が大きく変化しても対応付けが有効にできていることがわかる。

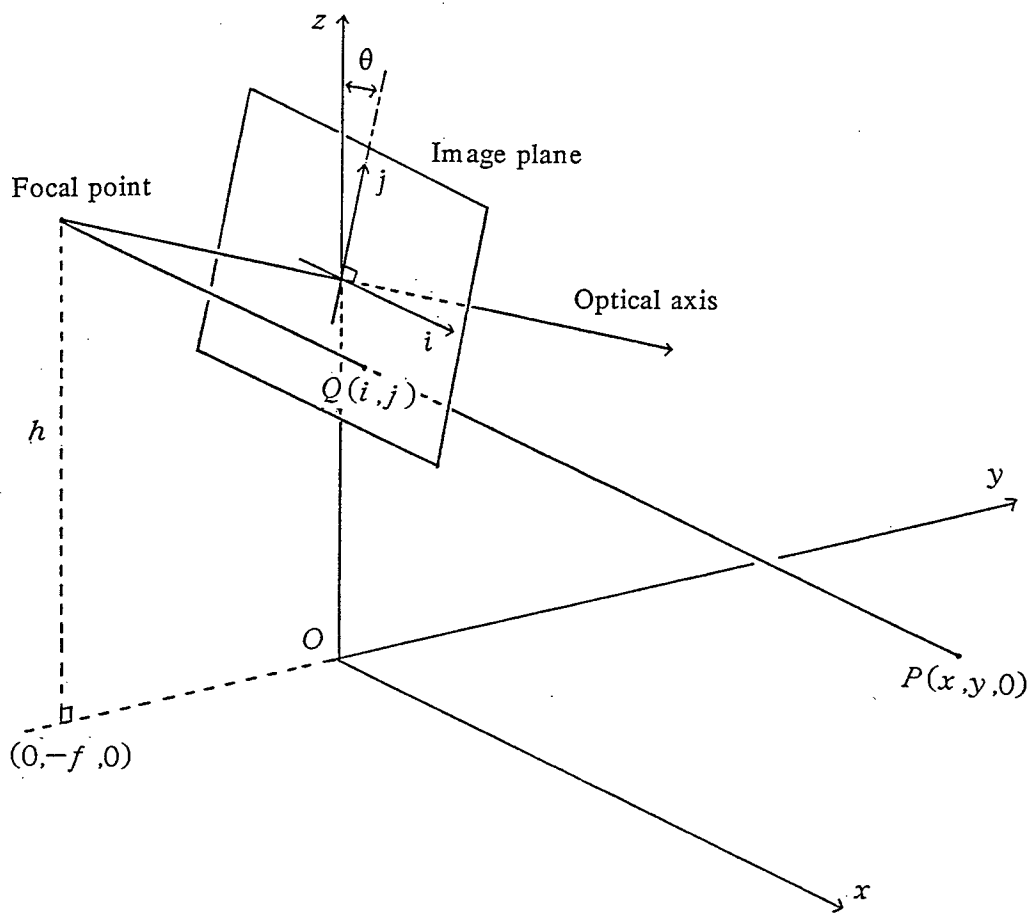


Fig. 3.1 Geometry of imaging a scene by a TV camera.

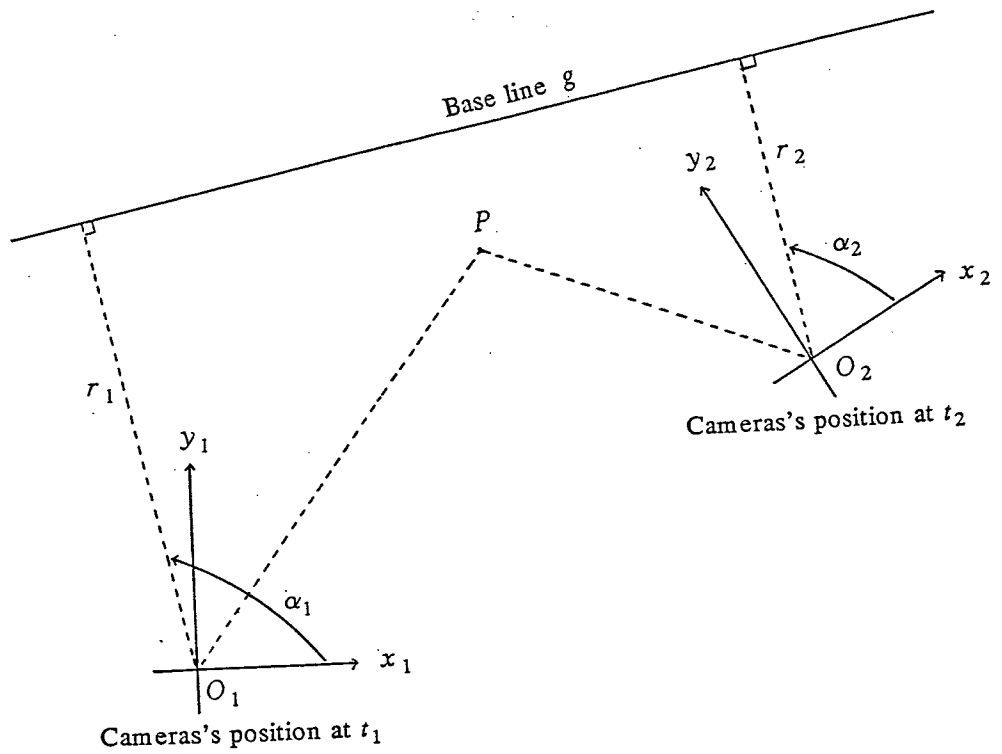
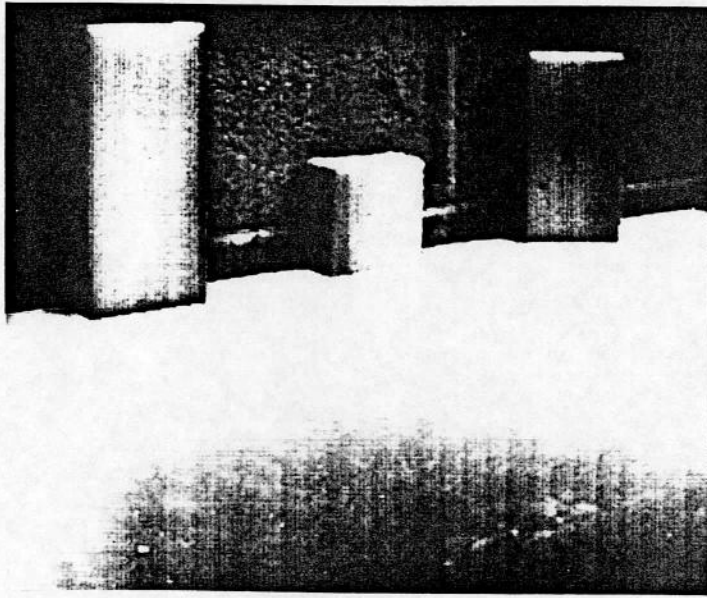
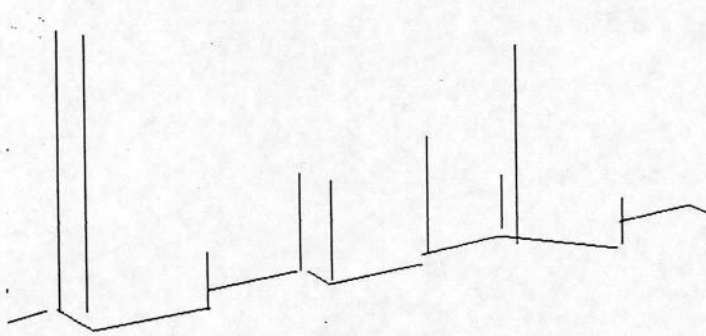


Fig. 3.2 Geometry of the proposed method.



(a) Original image.



(b) Overlay of the extracted segments.

Fig. 3.3 Example of extraction of edge lines.

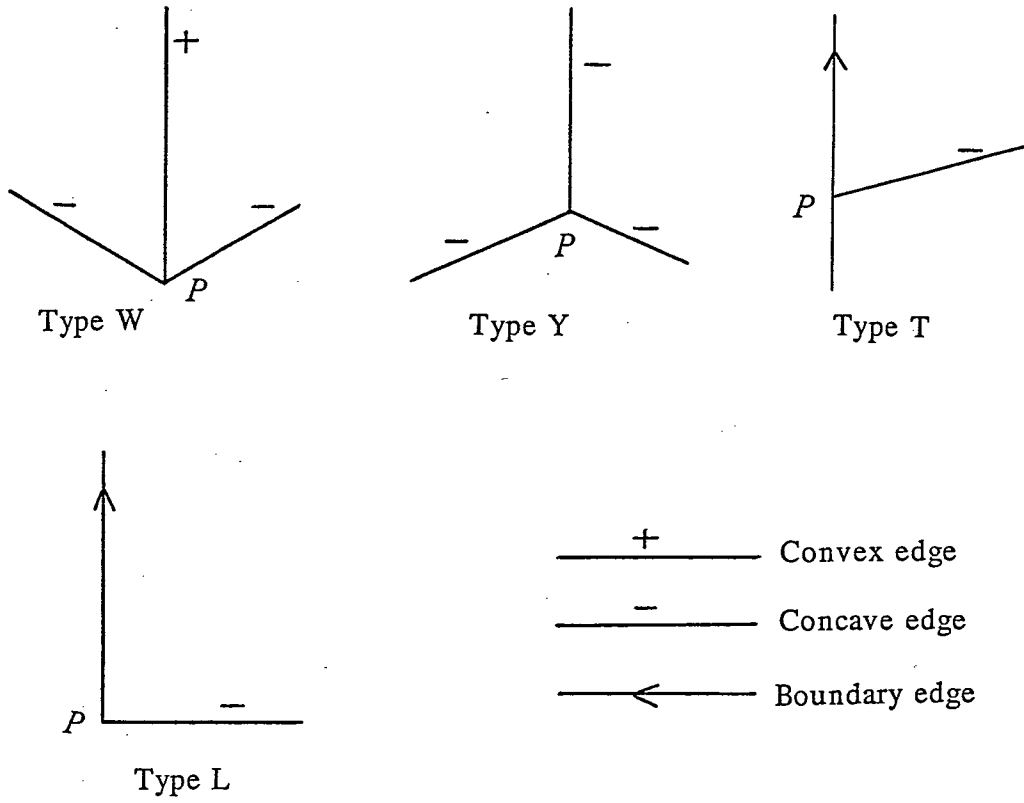


Fig. 3.4 Junction types.

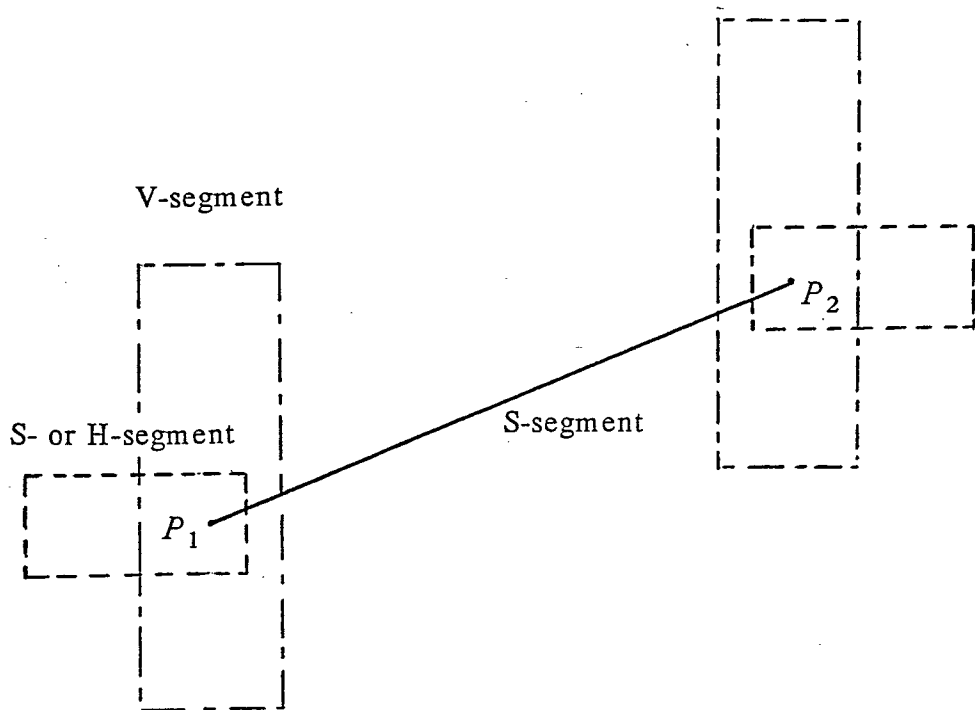


Fig. 3.5 Searching areas for junction points.

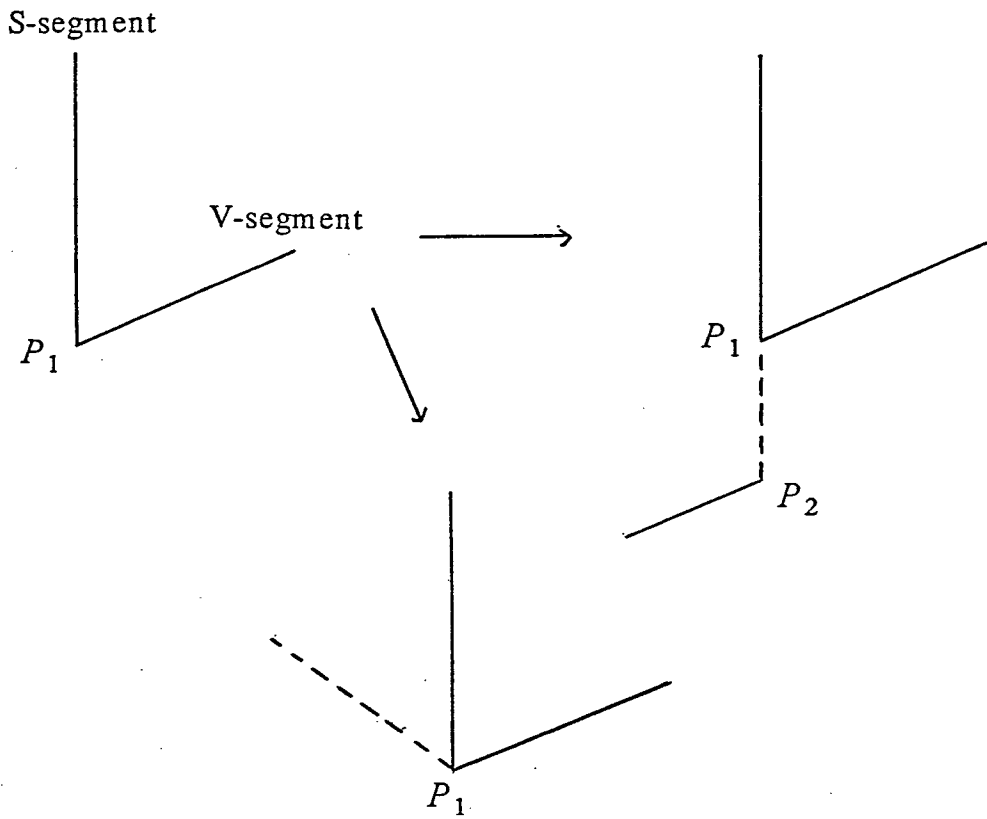


Fig. 3.6 Ambiguity of junction types.

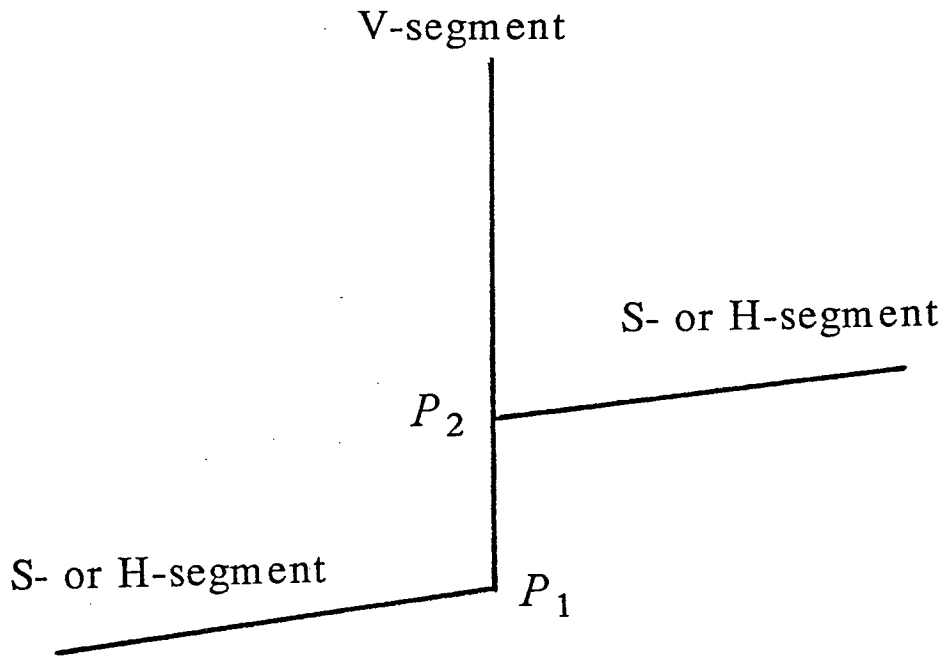


Fig. 3.7 V-segment with two junction points.

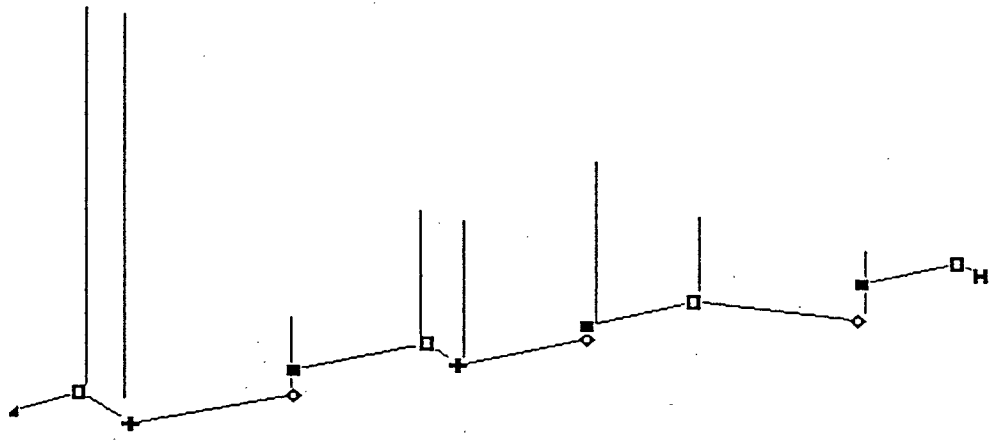


Fig. 3.8 Vertices in Fig. 3.3(b).

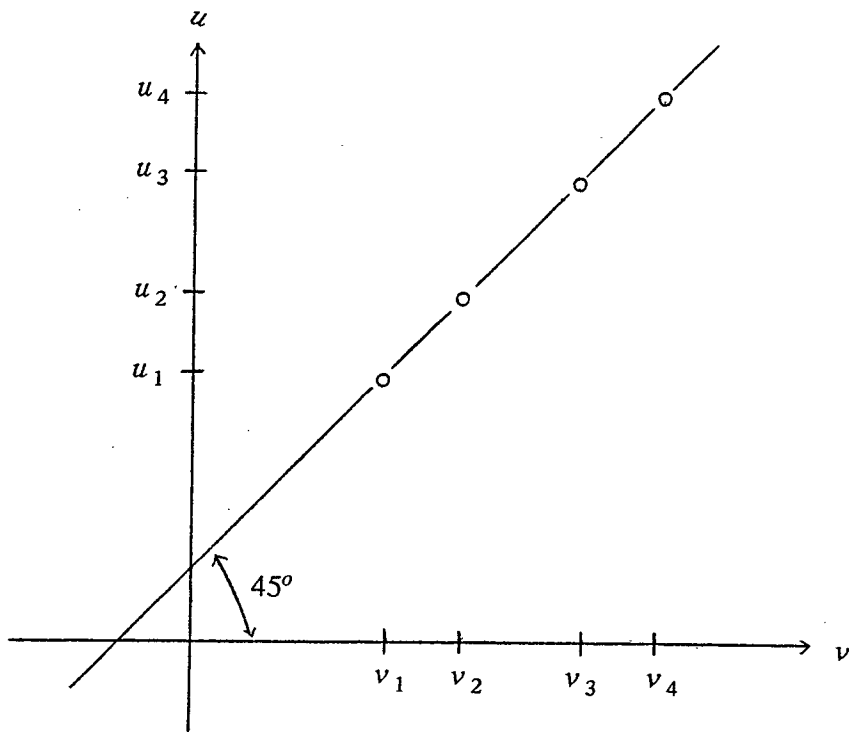
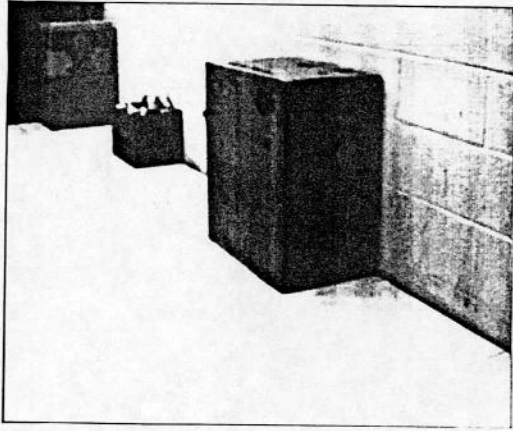
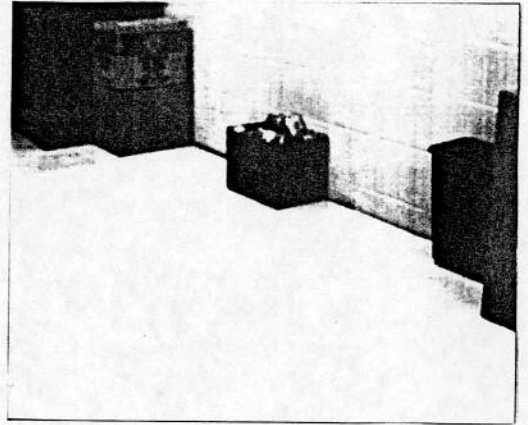


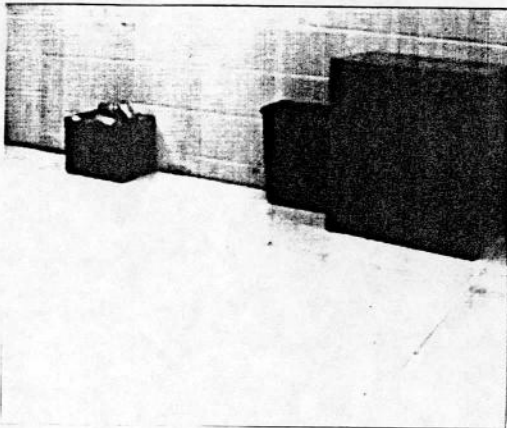
Fig. 3.9 Mapping matching pairs.



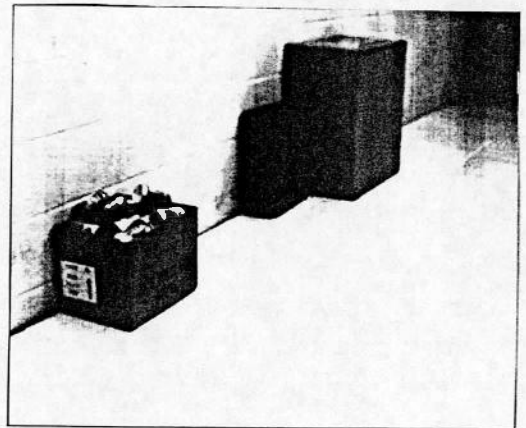
(a) First image.



(b) Second image.



(c) Third image.



(d) Fourth image.

Fig. 3.10 Four different images of the same scene.

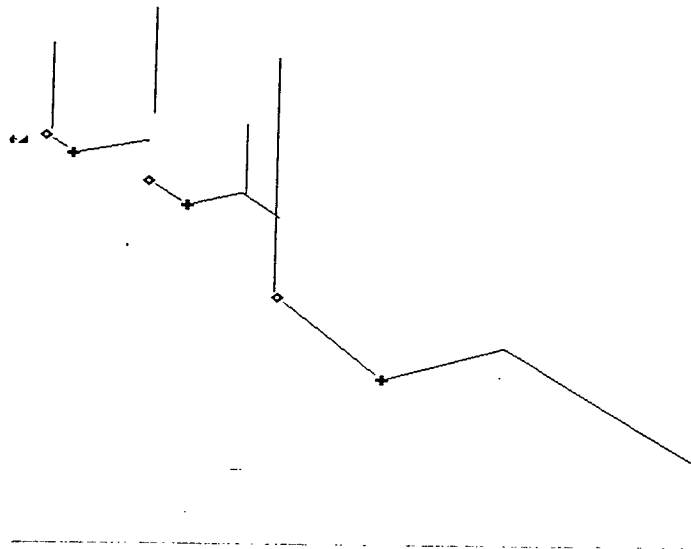


Fig. 3.11 Vertices in Fig. 3.10(a) for matching.

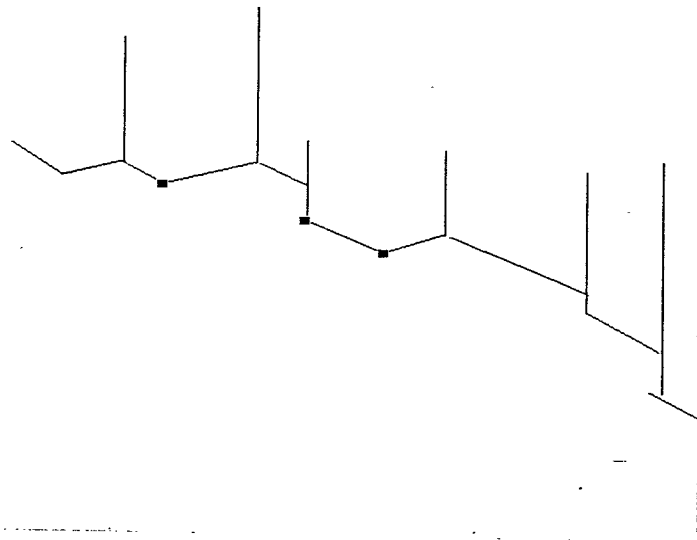


Fig. 3.12 Matched vertices in Fig. 3.10(b).

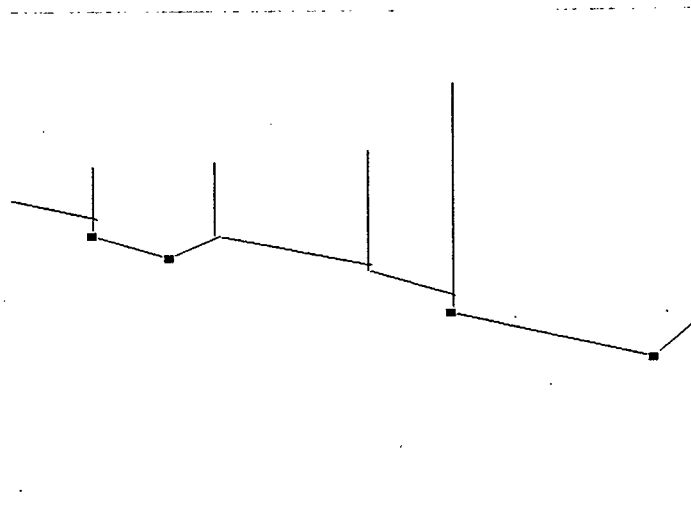


Fig. 3.13 Matched vertices in Fig. 3.10(c).

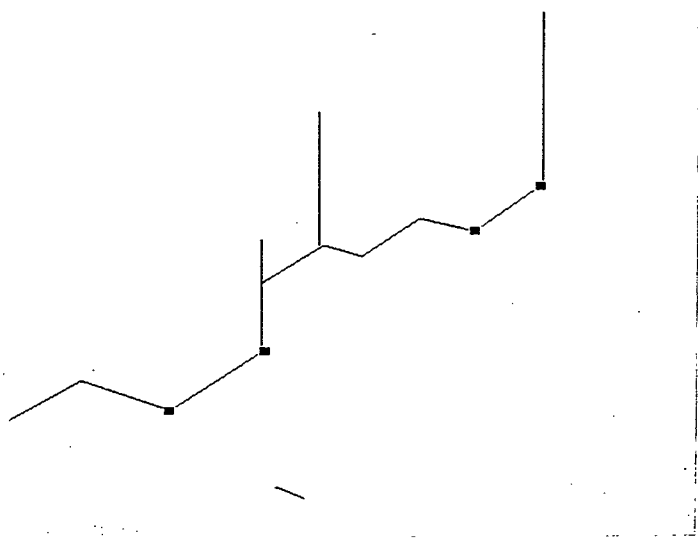


Fig. 3.14 Matched vertices in Fig. 3.10(d).

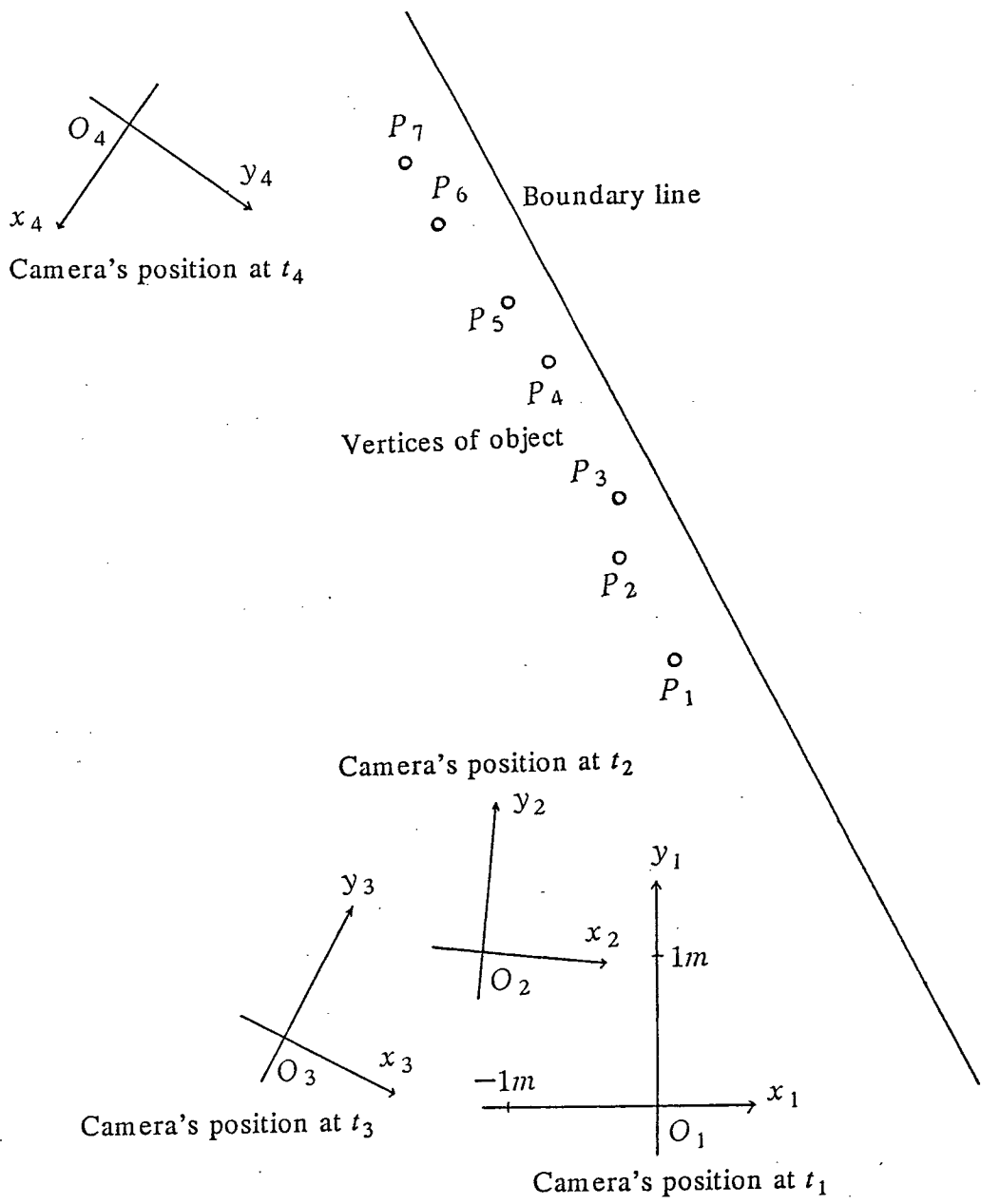


Fig. 3.15 Camera movements in the scene of Fig. 3.10(a)-(d).

第4章 人の動きの検出

1 処理の概要

本章では、一定の時間間隔で入力されたTV画像データから、人の動きを検出する手法を提案する。ロボットの走行環境として建物内の廊下を想定し、人がこの廊下に沿って、ある速度以下でなめらかに移動しているものとする。第1章4節で述べたように、移動ロボットのビジョンシステムは撮像時間間隔が比較的長い事例を扱うことを要求される。サンプル間隔が長くなるにつれて、画像間で人の見掛けのサイズや形が大きく変化し、対応付けが困難になる。しかし、人の膝のあたりは幅がほぼ一定の円柱状とみなせるので、比較的变化は小さいと考えられる。そのため、ここではTVカメラを低い位置に設置して、人の膝のあたりに注目した画像を使う。人の歩行は幅が一定の直立した円柱状の物体が平面上を移動している世界と考へ、人の動きは時間とともに変化する平面上でのこの物体の位置の軌跡で記述する。

歩行シーンは、廊下に固定した、高さ h と見込角 θ のTVカメラ(図4.1参照)から、一定の時間間隔で入力される。入力画像の処理は図4.2に示す手順で行う。まず最初、TV画像中のどこに廊下の領域があるか調べておく。次に、画素の明るさやその分散から廊下の領域の中で人である可能性が高い部分を見つける。ここで廊下は、第3章と同様、天井から十分な照明で一様に照らされ、床面は比較的反射率の高い平面であると仮定する。そして、この領域が円柱状の物体によるものか否かを判定し、もしそうであれば、人とみなし、廊下での位置を計算する。廊下での位置は画像の逆透視変換から求める。即ち、平面上の点 P (図4.1参照)の位置は、その画像アドレス(Q)、カメラの高さ(h)、見込角(θ)などから計算できる[55]ことを利用する。最後に、位置の情報と人の動きに関する拘束条件をもとに、1つ前の画像で見つかった人との対応付けを行う。この対応付けが人の動きを表わす。

第2～4節で画像から人を抽出する方法、第5節では画像間での人の対応

付けの方法を述べる。第6節では実画像を使った実験結果について考察する。第7節では画像演算の専用ハードウェア(DIP)による高速化、ならびに実時間の処理可能性を検討する。

2 廊下の領域の推定

この節では、対象とする歩行シーンの画像から、廊下の領域を推定する方法について述べる。廊下の領域というのは、人がいない状態の廊下で、柱などの部分的なでっぱりを取り除いたとき、側壁に相当する仮想的な垂直面で囲まれる床面のことである。

第3章と同様の照明の仮定から、廊下の床面はTV画像の中央下部に位置する明るい領域になり、側壁との境は第3章で述べた廊下の境界線と同じものになる。そのため、廊下の境界線を第3章の第1と第2節で述べた方法で画像から推定し、これをもとに廊下の領域を決める。ところで、第3章ではTVカメラは1つの側壁の方を向いていると仮定し、廊下の境界線を求めた。ここでは、TVカメラは両側の壁を見ていることがあるので、第3章の第3節で述べた処理の手順を、次のように少し変更する。Sセグメントを傾きにより正負の2つわけ、それぞれのグループごとに廊下の境界線の条件を満足する直線を見つける。このようにして得られた直線が1つならば、これを廊下の境界線とする。もし2本あれば、これらの直線の床面上での傾きを計算する。そして、この2つの直線が平行で、その距離が廊下の幅と一致すれば、これを廊下の2つの向いあう境界線とする。もし一致しなければ、廊下の境界線は長い直線と考えられるので、直線上のセグメントの長さの合計が大きいものを廊下の境界線とする。

この境界線で囲まれる領域が廊下の領域になるが、人の動きは、この廊下領域の奥行を制限した範囲内（これを有効廊下領域と呼ぶ）で考える。これは遠方にいる人は小さくなりすぎて、正しく抽出できないことによる。

図4.3 は歩行シーンの画像の例で、図中の白い部分が抽出された有効廊下領域である。

3 廊下面と人の領域の分離

この節では、画像中の有効廊下領域の中を、床面とそれ以外の領域（人である可能性が高い領域）とに分離する方法について述べる。

前節の廊下のシーンの仮定により、TV画像の中で一様に明るい領域は廊下の床面とみなすことができる。実際の床面の明るさは、照明の正反射光や人の影などにより変動するが、一定値以上の明るさになる。さらに人が洋服のしわなどにより部分的に明るくなることがある。しかし、その明るさは床面の明るさの分布の暗い部分に含まれ、また、この部分の明るさの分散は床面のそれより大きいと考えられる。そのため、画素の平均明るさと明るさの分散により、床面とそれ以外の領域に区別する。図4.1 に示すような画像座標系をとり、 (i, j) の位置にある画素の明るさを $I(i, j)$ とする。平均明るさ (M) と、明るさの分散 (D) を次のように定義する。

$$M(i, j) = \sum_{k=-1}^1 \sum_{l=-1}^1 I(i+k, j+l) \quad (1)$$

$$D(i, j) = |D_h(i, j)| + |D_v(i, j)| \quad (2)$$

ここで D_h 、 D_v は第3章の第2節で述べた水平および垂直方向の差分オペレータと同じものである。(3) あるいは (4) 式を満足する画素を床面とみなす。

$$M(i, j) \geq I_1 \quad (3)$$

$$I_1 > M(i, j) \geq I_2 \text{ and } D(i, j) > D_1 \quad (4)$$

ここで I_1, I_2, D_1 はあらかじめの実験で決めておく。

処理の手順としては、入力画像中の有効領域に対して前述の画像演算を施し床面を抽出する。しかし、抽出された床面は、ところどころ、汚れや影により小さな穴があいて、虫食いになることがある。この場合、その穴の面積

が小さければ床面とみなし、穴をうめる。この結果、有効領域から床面を除いた部分が人である可能性の高い領域になる。図4.4(a)は入力画像で、同図(b)は得られた床面である。床面の中ほどの黒い領域は前述の方法で取り除けなかったものである。

4 人の判定とその位置の推定

この節では床面以外の有効廊下領域から人の部分を切出し、その位置を決定する方法を述べる。人や物は廊下での位置により、TV画像上（有効廊下領域内）でのみかけの大きさが変わる。そのため、画像上の領域を以下のように変換を行ない、みかけの大きさに不変な領域で人かどうかの判定を行なう。

床面に平行な平面上に X-Y座標を考える。この座標の原点をTVカメラの位置に取り、X軸を廊下を横切る方向に、そしてY軸を廊下にそった方向にとる。有効廊下領域ならびに前節で得られた床面に相当する画素について、TVカメラから距離を計算し、この平面に写像する。これを領域地図と呼ぶ。TVカメラの視軸がY軸に一致していない場合は、第2節で求めた廊下の境界線より、そのずれを補正して写像する。領域地図上で、有効廊下領域と床面の領域の差に相当する部分が写像された人の領域になる。図4.5は図4.4の領域地図で、白い部分が床面、そして、黒い部分が人に相当する部分である。

人はサイズが一定の円柱状の物体と考えているので、領域地図上ではこれを写像したパターンになる。しかし、足の位置により図4.6(a),(b)に示すようなパターンの違いがでる。また、TV画像上で人がほかの人に一部分隠れている場合は、(c)のように途中からふくらむ。図4.7は人のパターンの例であるが、この図形の点 P_1 の近傍は廊下に接している足のつま先（人がTVカメラに向かって歩行する場合）に相当し、また、点 P_2 、 P_3 の付近はすねの部分になる。点 P_2 、 P_3 は床面より高い位置にあるので、実際の寸法より大きくなるが、人によらず幅 W_1 は一定範囲内に入る。またTVカメラは人より低い位置に

あるので、 P_5 、 P_6 は領域地図の上端に達する。このパターンの形状とサイズから、次のようにして人かどうかの判断をして、その位置を決める。図4.7と図4.8を使って説明する。

- ① 領域地図の下方から上方へ水平に走査し、領域の底点 P_1 を見つける。
- ② この領域の上方が有効廊下領域の上端に達しているか調べる。達していれば、点 P_1 から Y 軸にそって、一定の距離内で領域の幅 $W1$ が最初に $L1$ ($0.3m$) となる2点 P_2 、 P_3 を見つける。廊下での人の位置(P_4)を $[(X_2+X_3)/2, Y_1]$ とする。
- ③ この領域に別の人が部分的に隠れているかどうかを調べるために、図4.8に示すような直線 g 、 h を点 P_7 、 P_8 からそれぞれ引く。点 P_7 、 P_8 は点 P_1 を通り X 軸に平行な直線と、 Y 軸に平行でそれぞれ点 P_2 、 P_3 を通る2直線との交点である。直線 g 、 h の傾きは、人を幅 $W1$ の円柱とみなし、この円柱を領域地図に写像したときの外郭線の傾きと同じである。それぞれの直線から領域のはみだしている部分の長さ($W2$)を Y 軸にそって調べる。 $W2$ が $L2$ ($0.1m$) 以下であれば、人は隠れていないものとみなし、この領域を削除する。しかし、もし点 P_9 で $W2 > L2$ ならば、ここに別の人が隠れているものとみなし、このふくらんだ部分を残しておく。そのため点 P_9 から直線 g に平行な直線を引き、点 P_7 、 P_8 、 P_9 、 P_{10} 、 P_{11} で囲まれる領域のみを削除する。
- ④ ①にもどる。

5 人の対応付け

この節では時間的に連続する二つの領域地図から、人の対応関係を見つける方法について述べる。一番目の領域地図上の k 番目の人の位置を (X_{1k}, Y_{1k}) と表わし、2番目の領域地図上で、これに対応する点の座標を (X_{2k}, Y_{2k}) と表わす。

人の最大移動速度を V とし、二つの画像のサンプル時間間隔を Δt とすると、 k 番目の人がこのサンプル時間内に移動できる距離は $V \cdot \Delta t$ 以下である。まず最初、この条件で k 番目の人の2番目の領域地図での対応点を見つける。もし対応点が1つでない場合は、評価式(5)が全体として、最小となるような対応点の組合せを選ぶ。

$$E = \sum_{k=1}^n \sqrt{w \times (X_{1k} - X_{2k})^2 + (Y_{1k} - Y_{2k})^2} \quad (5)$$

ここでは人は廊下にそって動く傾向があるという仮定から、 $w > 1$ とする。

しかし、人が別の人にかくれたり、有効領域から出入して、対応点が見つからない場合があるので、領域地図を図4.9のようにA、B、Cの三つのゾーンに分割し、以下の方法で対応付けを行う。ただし、人の動きの仮定から、人はかならずAもしくはCゾーンを経由して、廊下の領域に出入するものとする。まず最初、1番目の領域地図のBゾーンにいる人と、2番目の領域地図の全域の人との間で、前述の条件にあう対応点を見つける。次に、1番目の領域地図のAとCのゾーンの人と2番目の領域地図でまだ対応がっていない人との間で対応を見つける。対応点が見つからないものについては次のように解釈する。

- ① 1番目の領域地図ではBゾーンにだけ人がいた場合：1番目のシーンにいた人は画像のサンプル間隔の間に廊下の領域から出ることができないから、2番目のシーンでは人の陰に隠れたとする。逆に、2番目

の領域地図のBゾーンにいた人で、1番目の領域地図に対応点が見つからない時は、1番目のシーンでは別の人の後にいたとする。また2番目のAとCのゾーンの人には、この時点で廊下の領域に入って来たとする。

- ② 1番目の領域地図のAとCゾーンにも人がいた場合：対応の取れない人がいれば、①と同じように取り扱おう。ただし、1番目の領域地図のAとCのゾーンの対応の取れない人は、次の画像のサンプルまでに廊下領域から出ていったとする。

6 実験結果

実際の歩行シーンを使って、このアルゴリズムの有効性を検証した。図4.10は連続した4つのTV画像の例である。この画像は、当所の廊下にTVカメラを設置し、普通の速さで人が歩行するシーンを、0.7秒間隔で入力したものである。TV画像は256 x 256画素で、各画素は256階調である。床面はアイボリー系のPタイルでできており、照明は天井の蛍光灯による。その床面の平均照度は200ルクスであった。TVカメラの高さは約74cmで、その見込角は10度であった。廊下の有効領域はTVカメラの高さと見込角から、奥行は2.7mから6mとした。また横方向は廊下の幅と同じ1.4mであった。図4.11は、この連続するTV画像から検出された人の動きである。TV画像で左側の人、また真中にいる人は、最初、有効領域外にいたので検出されていない。

平均明るさと明るさの分散による画像演算だけでは、白っぽい服装の場合、床面と区別がつかないかもしれない。この事例を検証するために、白いズボンをはいた人の歩行シーン(図4.12(a))について実験を行ったが、このような悪条件でも、これらの画像演算だけで十分有効であることがわかった(図4.12(b))。

ところで、スカート姿の女性(図4.13(a))の場合、領域地図では、足もとからスカートの部分にかけてふくらみがあられ、別の人が見えているように見える。しかし、スカートによる人の幅は、ほかの洋服姿の場合より少し大きくなるが(図4.13(b))、図4.8で示した2直線 g 、 h よりはみだすことはほとんどなかった。

7 高速化のための考察

TV画像の入力および画像処理は、第2章で述べた画像処理システム(DIP)を使った。第2、3節で述べた画像演算の大部分は画像演算モジュールで、それ以外は計算機で処理を行なった。第3節での床面と人の領域を分離する部分の画像演算モジュール内の処理は以下ようになる。画像演算モジュール内の4層の画像メモリをM1~M4とする。まず、TVカメラから画像データはM1に転送される。第3節の(1)式と、(2)式の絶対値の中味は 3×3 の窓による積和演算回路で計算する。まず、(1)式の演算を行い結果をM2に出力する。あらかじめ廊下の有効領域以外のマスクパターンを計算機で作っておき、これをM3に転送し、演算モジュールのマイクロプロセッサによる処理でM2をマスクし、M4に結果を出力する。次に、M1を入力画像として、(2)式の絶対値の部分を順次積和演算し、M2、M3に出力する。そして、M4の画像でマスクされている部分では0をM1に出力し、そうでない部分の画素については、第3節の(3)式を満たせば、1をM1に出力し、もし(4)式の前半を満たせば、M2とM3のデータから(2)式の絶対値と和を計算し、チェックする。もし条件を満たせば1、さもなければ0を出力する。この部分はマイクロプロセッサによって行うが、すべてを1回の画像データ走査によって行うことができる。さらに、この処理で得られた領域の中で、ノイズによる微小領域あるいは虫くいの部分は、領域番号付とヒストグラムの演算回路を使い消去する。

積和演算は1画像あたり65ms要す。また床面の判定は場合によって処理時

間異なるが、1 画像あたり、200ms となっている。全体では画像の転送、パラメータの設定、さらにノイズによる微小領域の消去も含まれるため 2 秒弱となった。これは計算機に比べ約百倍の速さである。人の位置の計算および人の対応付けは計算機で行なうが、約 5 秒ほどかかる。このため入力 TV 画像は一度デスクに蓄えてから処理を行っている。複数個の画像演算器をカスケード接続できるようなハードウェアの開発や、将来の素子の高速化により、0.7 秒以内にすべて処理できることが予想される。

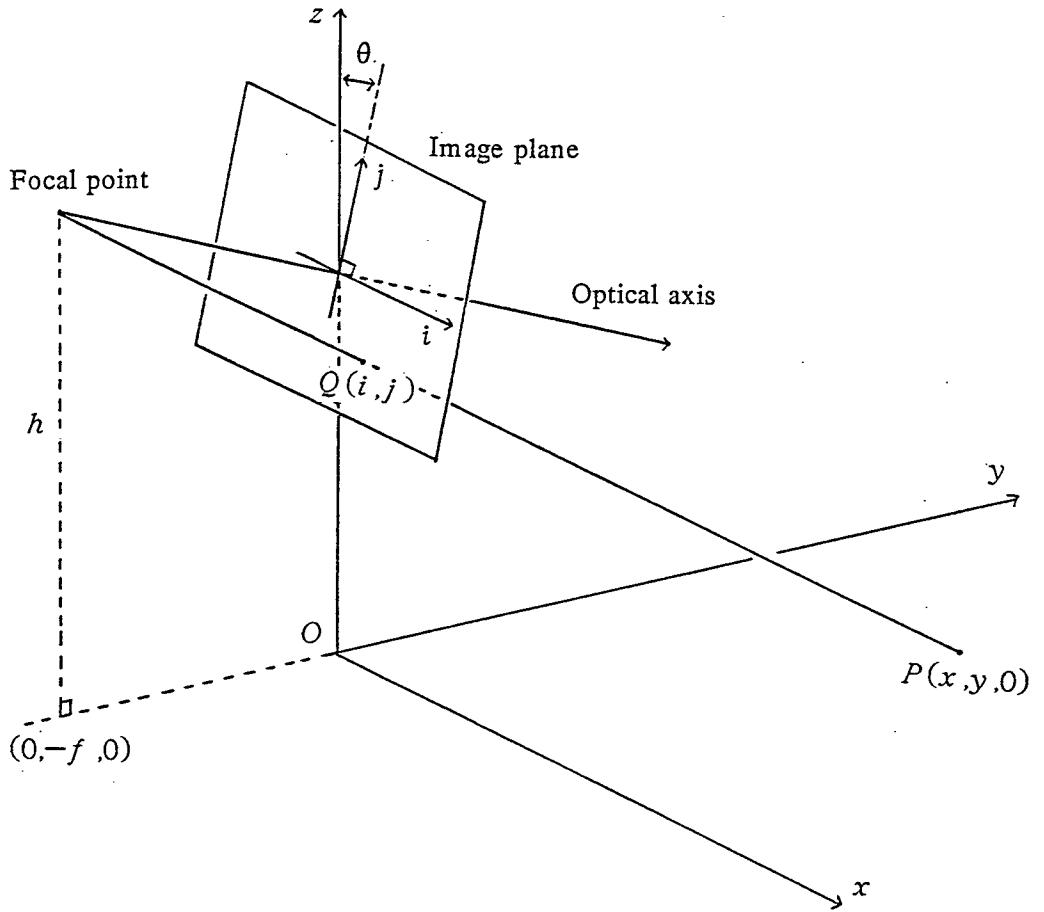


Fig. 4.1 Geometry of imaging a scene by a TV camera.

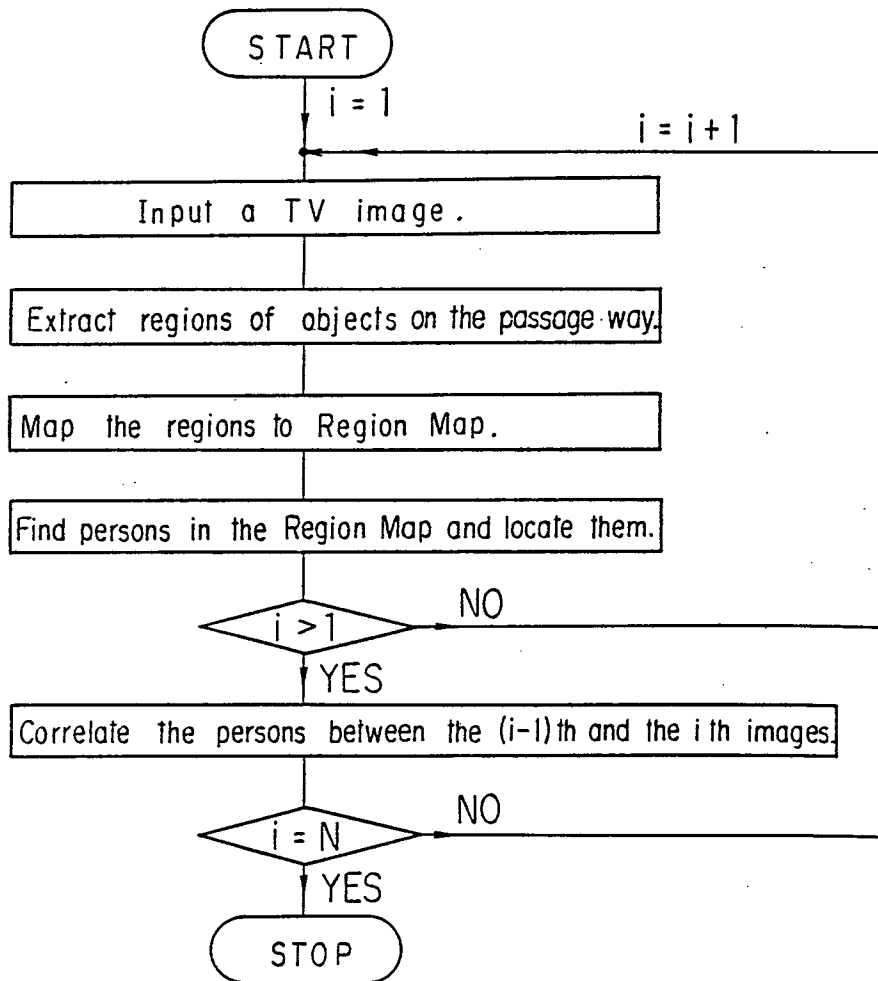


Fig. 4.2 Flow chart of the method.

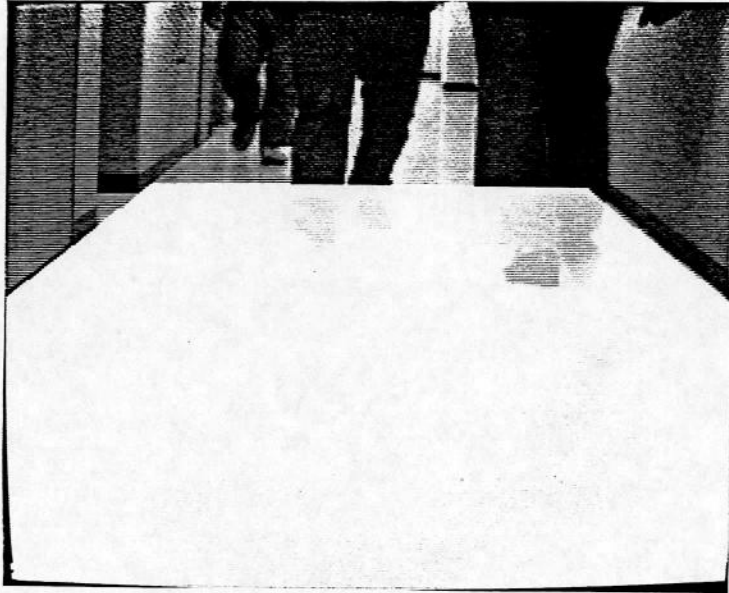
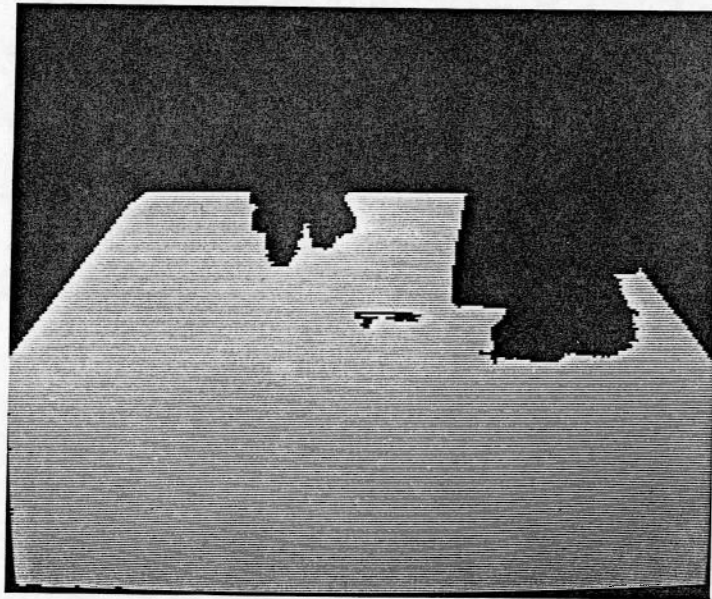


Fig. 4.3 Search area for persons.



(a) Input image.



(b) Passageway region of the image (a).

Fig. 4.4 Extraction of a passageway region.

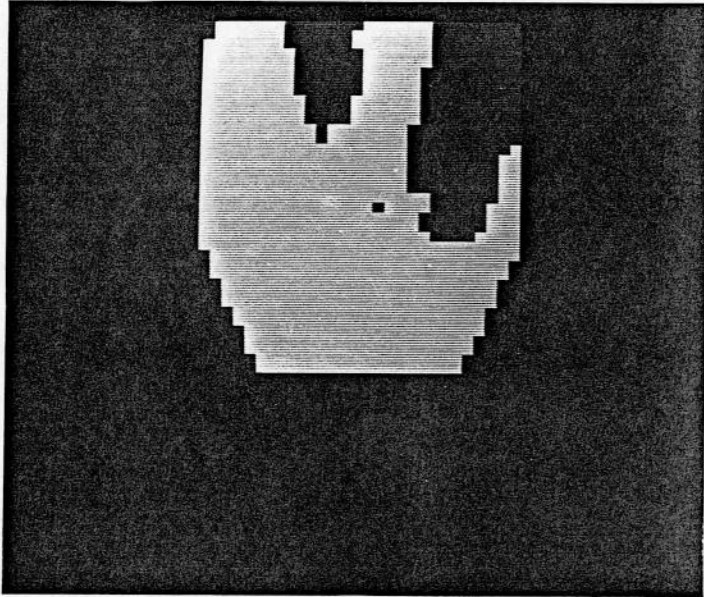


Fig. 4.5 Region map of Fig. 4.4.

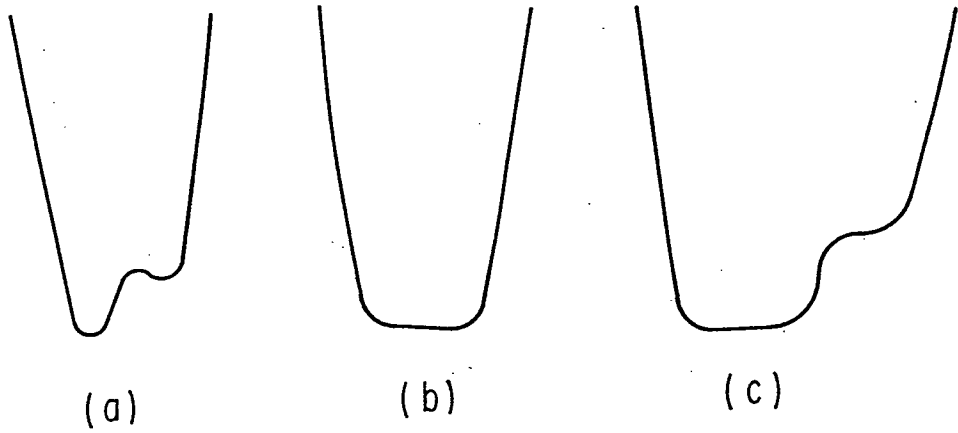


Fig. 4.6 Patterns of persons on a region map.

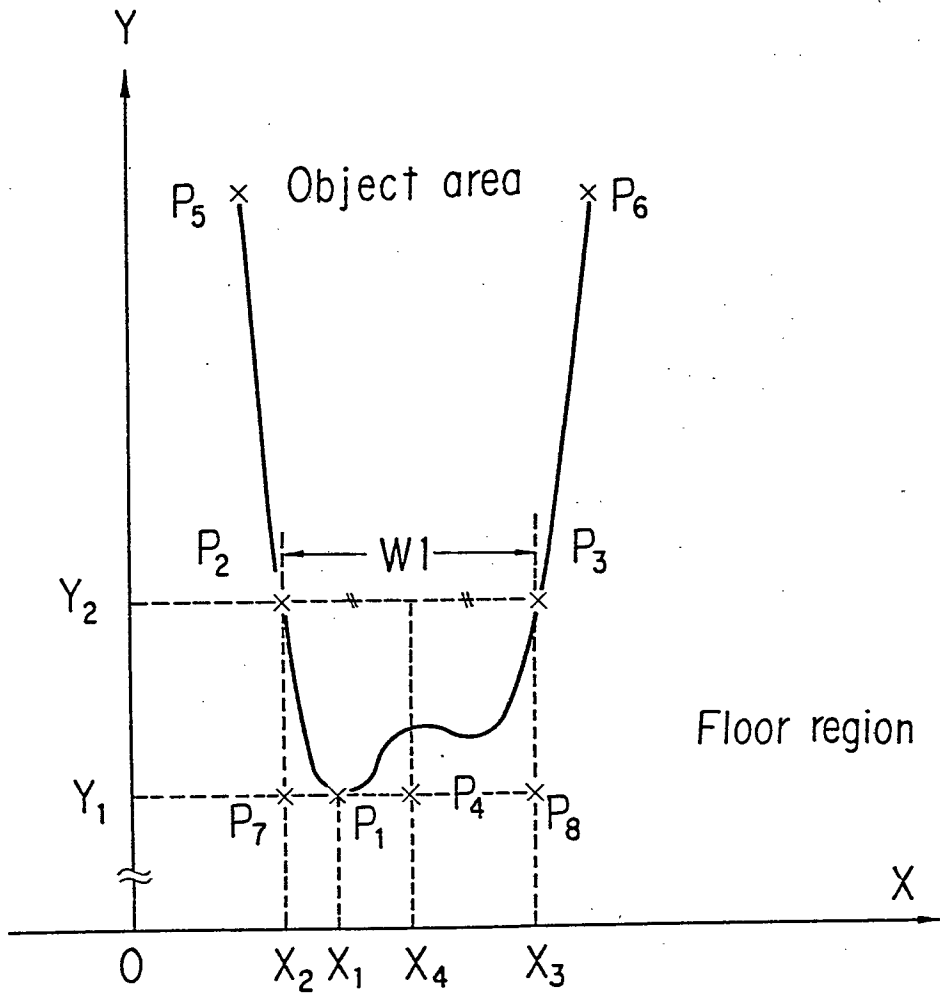


Fig. 4.7 Simplified example of a region map.

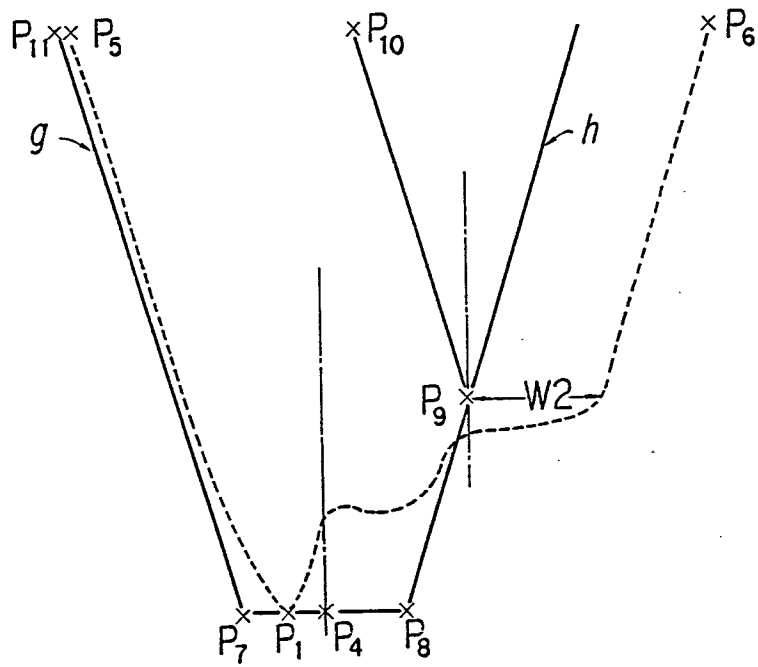


Fig. 4.8 Mask for finding occlusion.

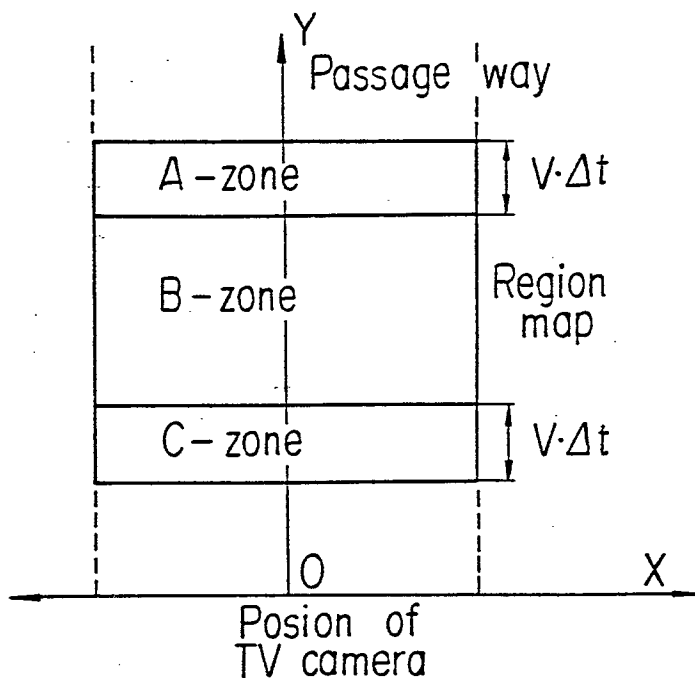


Fig. 4.9 Three zones of a region map.



(a) First image.



(b) Second image.



(c) Third image.



(d) Fourth image.

Fig. 4.10 Example of four consecutive images.

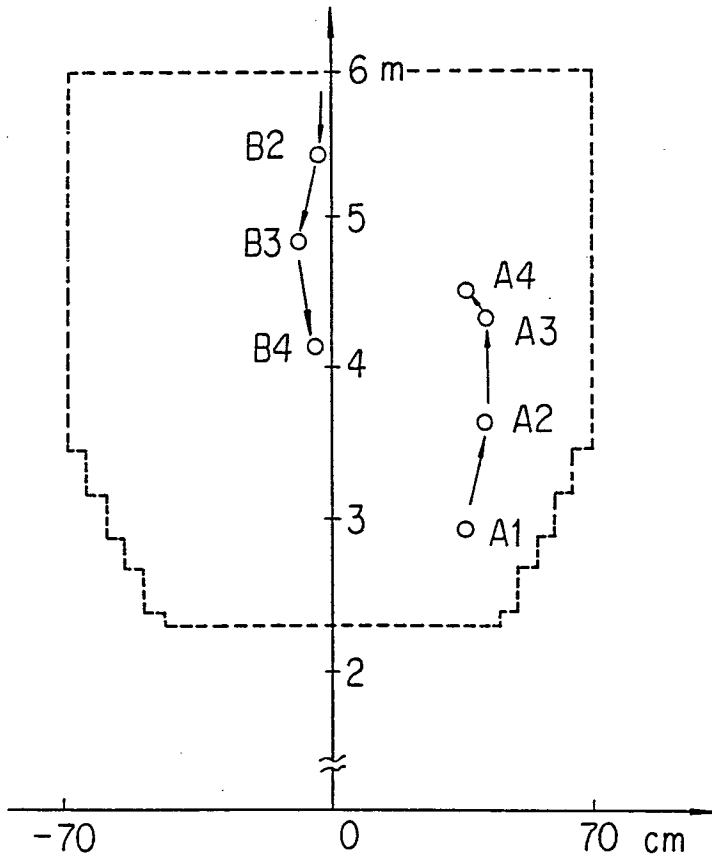
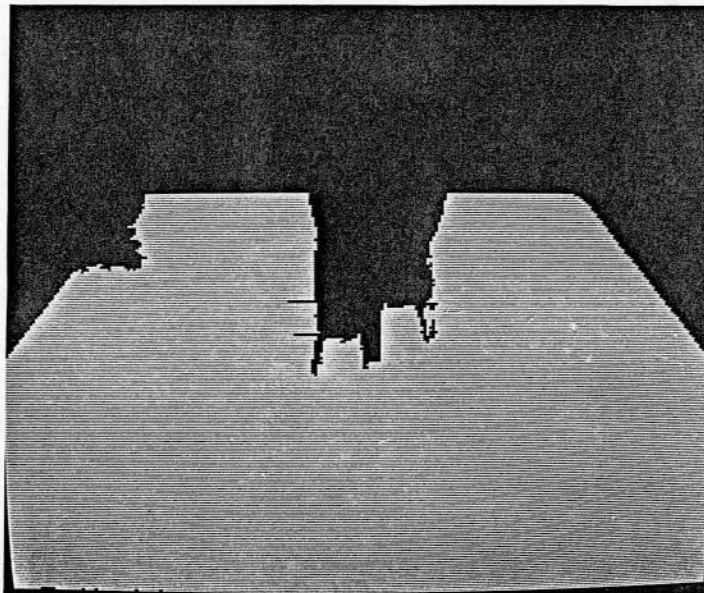


Fig. 4.11 Movements of persons in Fig. 4.10.



(a) Input image.

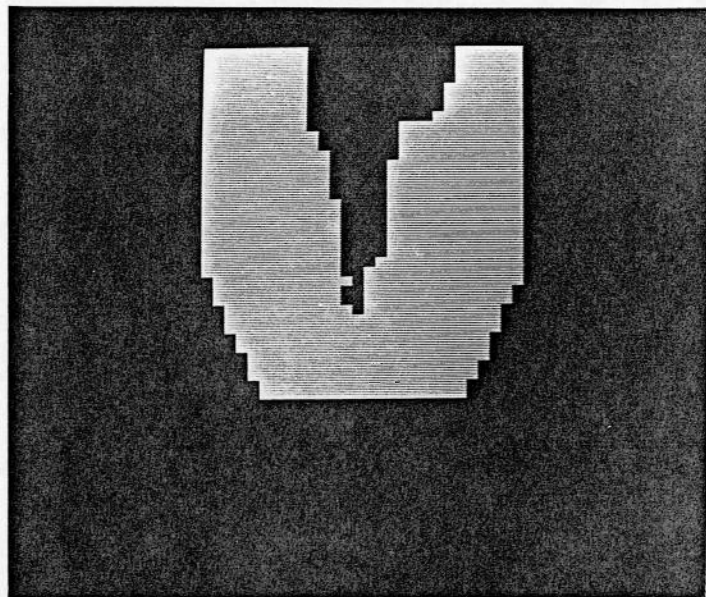


(b) Extracted passageway region.

Fig. 4.12 Example of a person wearing white clothes.



(a) Input image.



(b) Region map of the image (a).

Fig. 4.13 Example of a lady wearing a skirt.

移動ロボットの自律性には、外部状況が大局的に判断できるようなビジョンシステムが必要である。本研究は、このような観点に立ち、単眼のTV画像による移動用ビジョンシステムの実現をめざしたものである。特に、本論文では、移動用ビジョンシステムとしてとりわけ重要と考えられる、人との衝突回避のための動画像処理、画像列からのロボットの移動量の推定法、そして、時系列の画像を扱うための処理システムの諸問題を取り上げた。従来、移動ロボットの研究は、特殊な距離センサを使い、静止した世界を対象としていたが、本研究では人が移動する動的な世界を扱えること、また、シーン内で見られるの一般的な特徴を手掛りにして移動できることを示した。本研究では、短時間にシーンの詳細なデータがとりこめる利点などから、センサとして汎用的なTVカメラ使っているが、反面、データ処理に時間がかかることと、単眼視では距離情報が一部欠落する問題がある。これに対しては、画像処理システムの構成を工夫することにより処理時間が短縮できることを示した。また、単眼視でもロボットの移動に必要な2次元平面内での位置情報が十分得られることを示した。これが本論文の全般的な成果である。この成果は、今後、人と共存しながら、違和感なく移動し、作業するロボットの実現に貢献するものと考えられる。

本論文では3つのテーマを取り上げたが、各テーマごとに総括すると以下のようなになる。第2章では、計算機および機能装置をネットワークで相互結合する画像処理用システムの構成法を提案した。また、この考えに基づいて実験システム(DIP)を設計開発し、この方式の有効性を示した。提案した方式は、各種の機能装置や計算機の接続が容易に行え、任意の装置間で大量の画像データを効率よく転送できる機構に特徴がある。提案した方式は、ロボットビジョン用の処理システムだけでなく、様々な画像処理システムの設計に応用が可能である。

第3章では、ロボットが移動しながら撮像したTV画像から、そのロボットの移動量を推定する方法について提案した。移動量の推定には画像間の対応関係と、対応点の位置の情報が必要である。画像間の特徴点にはシーン内の物体の頂点を使い、シーンの中で一般に見られる長い直線をガイドにして、その対応付けを行なった。物体の頂点は入力画像のエッジマップを接合テーブルを使い解析し、求めた。物体の頂点の位置情報はTV画像の逆透視変換により求めた。

提案した方式では、画像間の見かけの変化が大きい場合でも、対応付けが正しく行なえる。そのため、ロボットがかなり移動しても、その移動量の推定が可能である。対応付けにはシーンの3次元の構造情報を使っている。そのため、この対応付けの方法はロボットが移動しながら走行空間のモデルを作成する場合にも応用ができる。

第4章では、時系列のTV画像から、人の動きを検出する手法を提案した。人は画像間で見掛けの大きさや形が大きく変化する可能性があるので、まず画像ごとに人を抽出した。次に、その人の実際の位置を計測し、位置の情報から画像間で人の対応付けを行なった。そして、実際の廊下のシーンを使って、その有効性を示した。画像から廊下領域を抽出する方法として、簡単なオペレータによる画像処理を示し、また、その処理時間を短かくするために専用のハードウェアで画像処理を実行し、実時間処理の可能性を示した。3次元運動を扱う動画像処理では、物体の見え隠れの問題があるが、ここでは人の予想される移動速度を使って解決した。

今後の課題として、人とTVカメラが共に移動する問題がある。この場合も、ここで提案した方法で人を各画像からまず見つける。しかし、人の位置はワールド座標系で表し、この座標系で対応付けを考えることにより可能と思われる。

謝 辞

本論文は、筆者が電子技術総合研究所において行った一連の研究をまとめたものである。本研究の機会を与えてくださった若松清司・前制御部長、淵一博・元パターン情報部長、西野博二・元パターン情報部長に感謝いたします。

白井良明・制御部長、杉江昇・名古屋大学工学部教授、大島正毅・視覚システム研究室長、諏訪基・人間機械システム研究室長からは、常に変らぬ御支援と御指導をいただきました。ここに感謝いたします。

本論文の第2章の研究では、諏訪基・人間機械システム研究室長と白井良明・制御部長から適切な助言をいただきました。第3章は、筆者が米国イリノイ州立大学に客員研究員として滞在中に行った研究が基になっている。御支援と有益な御意見をいただいたT. S. Huang教授に感謝いたします。第4章の研究を進めるにあたっては、白井良明・制御部長から適切な助言をいただきました。

本論文をまとめるにあたっては、大阪大学基礎工学部の辻三郎教授、鈴木良次教授、井口征士教授から有益な御意見と御指導を賜りました。ここに深く感謝いたします。

以上の外にも多くの方々のお陰によって、本研究を行うことができたと思っております。ここに改めて関係各位に感謝の意を表します。

参 考 文 献

- [1] 戸田, 佐々木 "産業用ロボットの機構と設計 -PUMAメカニズムの紹介-"
日本ロボット学会誌, vol.1, no.2, pp.30-35, 1983.
- [2] 篠塚, 関 "FAにおける無人搬送と貯蔵の自動化" 計測自動制御学会誌,
vol.22, no.11, pp.931-936, 1983.
- [3] N.J.Nilsson "A Mobile Automation: an Application of Artificial
Intelligence Techniques" Proc. 1st IJCAI, pp.509-520, 1969.
- [4] L.S.Coles, A.M.Robb, et al. "Decision Analysis for an
Experimental Robot with Unreliable Sensors" Proc. 4th IJCAI,
pp.749-757, 1975.
- [5] G.Giralt, R.Sobek, et al. "A Multi-Level Planning and Navigation
System for a Mobile Robot; a First Approach to HILARE"
Proc. 6th IJCAI, pp.335-337, 1979.
- [6] Y.Ichikawa, N.Ozaki "A Heuristic Planner an Executive for Mobile
Robot Control" IEEE Tran. SMC, vol.15, no.4, pp.558-563, 1985.
- [7] A.M.Thompson "The Navigation System of the JPL Robot"
Proc. 5th IJCAI, pp.749-757, 1977.
- [8] R.A.Lewis, A.R.Johnston "A Scanning Laser Rangefinder for a
Robotic Vehicle" Proc. 5th IJCAI, pp.762-768, 1977.
- [9] Y.Yakimovsky, R.Cunningham "A System for Extracting Three
-Dimensional Measurements from a Stereo Pair of TV Cameras"
Computer Graphics and Image Processing, vol.7, no.2, pp.195-210,
1978.
- [10] H.P.Moravec "Rober Visual Obstacle Avoidance" Proc. 7th IJCAI,
pp.785-790, 1981.

- [11] C.E.Thorpe "Path Relaxation: Path Planning for a Mobile Robot"
Proc. AAAI-84, pp.318-321, 1984.
- [12] R.Chattergy "Some Heuristics for the Navigation of a Robot"
Int. Journal Robotic Res., vol.4, no.1, pp.59-66, 1985.
- [13] H.P.Moravec, A.Elfes "High Resolution Maps from Wide Angle
Sonar" Proc. IEEE Int. Conf. on Robotics and Automation,
pp.116-121, 1985.
- [14] J.L.Crowley "Navigation for an Intelligent Mobile Robot"
IEEE Journal RA, vol.1, no.1, pp.31-41, 1985.
- [15] M.P.Turchan, A.KC.Wong "Sensor-Based Geometric World Model
Acquisition for Autonomous Vehicles: a Graph Synthesis Approach"
Proc. System Man and Cybernetics Conference, 1985.
- [16] S.Tsuji, J.Yu Zheng, et al. "Stereo Vision of a Mobile Robot:
World Constraints for Image Matching and Interpretation"
Proc. IEEE Int. Conf. on Robotics and Automation, pp.1594-1599,
1986.
- [17] A.M.Parodi, J.J.Nitao, et al. "An Intelligent System for an
Autonomous Vehicle" Proc. IEEE Int. Conf. on Robotics and
Automation, pp.1657-1663, 1986.
- [18] D.T.Lawton, T.S.Levit, et al. "Terrain Models for an Autonomous
Land Vehicle" Proc. IEEE Int. Conf. on Robotics and Automation,
pp.2043-2051, 1986.
- [19] R.Wallace, K.Matsuzaki, et al. "Progress in Robot Road-Following"
Proc. IEEE Int. Conf. on Robotics and Automation, pp.1615-1621,
1986.

- [20] F.P.Andresen, L.S.Davis, et al. "Visual Algorithms for Autonomous Navigation" Proc. IEEE Int. Conf. on Robotics and Automation, pp.856-861, 1985.
- [21] A.M.Waxman, J.Le Moigne, et al "Visual Navigation of Roadways" Proc. IEEE Int. Conf. on Robotics and Automation, pp.862-867, 1985.
- [22] R.L.Madarasz, L.C.Heiny, et al "The Design of an Autonomous Vehicle for the Disable" IEEE Journal RA, vol.2, no.3, pp.117-126, 1986.
- [23] 油田, 金山, 飯島 "移動ロボットのための超音波センサ" バイオメカニズム学会誌, vol.5, no.3, pp.18-24, 1981.
- [24] R.A.Jarvis "A Laser Time-of-Flight Range Scanner for Robotic Vision" IEEE Tran. PAMI, vol.5, no.5, pp.505-512, 1983.
- [25] 安江, 白井 "物体認識のための両眼立体視" 電総研彙報, vol.37, no.12, pp.1101-1119, 1973.
- [26] L.G.Roberts "Machine Perception of Three-Dimensional Solids " Optical and Electro-Optical Information Processing (MIT Press), pp.159-197, 1965.
- [27] D.A.Huffman "Impossible Objects as Nonsense Sentences " Machine Intelligence (Edinburgh Univ. Press), vol.6, pp.295-323, 1971.
- [28] M.B.Clowes "On Seeing Things" Artificial Intelligence, vol.2, no.2, pp.79-116, 1971.
- [29] B.K.P.Horn "Understanding Image Intensities" Artificial Intelligence, vol.8, no.2, pp.201-231, 1977.
- [30] J.Kender "Shape from Texture; an Aggregation Transform That Maps a Class of Textures into Surface Orientation" Proc. 6th IJCAI, pp.457-480, 1979.

- [31] T. Kanade "Recovery of the Three-Dimensional Shape of an Object from a Single View" *Artificial Intelligence*, vol.17, pp.409-460, 1981.
- [32] Y. Yakimovsky, J.A. Feldman "A Semantic-Based Decision Theory Region Analyzer" *Proc. 3rd IJCAI*, pp.580-588, 1973.
- [33] J.M. Tenenbaum, H.G. Barrow "Experimentals in Interpretation -Guided Segmentation" *Artificial Intelligence*, vol.8, no.3, pp.241-274, 1977.
- [34] M. Deering, C. Collins "Real-Time Natural Scene Analysis for a Blind Prosthesis" *Proc. 7th IJCAI*, pp.704-709, 1981.
- [35] A. Robert, S. Vincent "A 3D Perception System for the Mobile Robot HILARE" *Proc. IEEE Int. Conf. on Robotics and Automation*, pp.1105-1111, 1986.
- [36] 木戸出, 麻田, 他 "インテリジェント機能を有する画像メモリ装置" *電子通信学会技術報告 EC79-43*, pp.13-20, 1979.
- [37] 小林, 奥山, 他 "汎用画像認識解析装置 HIDICシリーズ" *映像情報*, vol.16, no.10, pp.19-25, 1984.
- [38] 溝口, 天満 "TIP での画像処理" *情報処理学会研究会資料 CV21-4*, 1982.
- [39] 麻田, 篠田, 他 "二次元フーリエ変換機能などをもった並列画像処理装置の開発" *電子通信学会技術報告 IE78-12*, 1978.
- [40] 松本 "イメージ・パイプライン・プロセッサの特徴とその応用" *映像情報*, vol.16, no.12, pp.19-24, 1984.
- [41] 福島 "画像処理 LSI-Image Signal Processor のアーキテクチャ" *電子通信学会論文誌*, vol.J66c, no.12, pp.959-966, 1983.
- [42] 池田, 西原, 他 "筑波大学学術情報センターにおける画像処理システム" *情報処理学会研究会資料 CV3-5*, 1979.

- [43]八村, 長屋, 他 "国立民族学博物館における対話型マルチユーザ画像処理システム" 情報処理学会研究資料 CV10-4, 1981.
- [44]中村, 弓削 "濃淡画像の前処理機能をもつグラフィックターミナル" 情報処理学会論文誌, vol.23, no.4, pp.428-434, 1982.
- [45]周藤, 梅村, 他 "汎用医用画像処理開発システム(GMIPS) について" 情報処理学会研究会資料 CV20-1, 1982.
- [46]I.Fukui "TV Image processing to Determine the Position of Robot Vehicle" Pattern Recognition, vol.14, no.1-6, pp.101-109, 1981.
- [47]R. Nevatia "Depth Measurement by Motion Stereo" Computer Graphics and Image Processing, vol.5, no.2, pp.203-214, 1976.
- [48]S.T.Barnard, W.B.Thompson "Disparity Analysis of Images" IEEE Tran. PAMI, vol.2, no.4, pp.333-340, 1980.
- [49]M.J.Hannah "Bootstrap Stereo" Proc. Image Understanding Workshop (DARPA), pp201-208, 1980.
- [50]C.Thorpe, S.Shafer "Correspondence in Line Drawing of Multiple Views of Objects" Proc. 8th IJCAI, 1983.
- [51]B.K.P.Horn, B.G.Schunk "Determining Optical Flow" Proc. Image Understanding Workshop (DARPA), pp.144-156, 1981.
- [52]L.Dreschler, H.H.Nagel "On the Frame-to-Frame Conespondence between Grayvalue Characteristics in the Images of Moving Objects" German Workshop on Artificial Inteligence (Springer-Verlag), vol.81, pp.18-21, 1981.
- [53]R.Jain, H.H.Nagel "On the Analysis of Accumilative Difference Pictures from Image Sequence of Real World Scenes" IEEE Tran. PAMI, vol.1, no.2, pp.206-214, 1979.

[54] 森, 篠田, 他 "並列画像演算装置の試作"

電子通信学会技術報告 PRL76-86, 1977.

[55] R.O.Duda, P.E.Hart "Pattern Classification and Scene Analysis"

John Wiley & Sons, 1973.

[56] W.K.Pratt "System Architecture of Vicom Digital Image Processor"

Proc. SPIE, vol.301, pp.78-82, 1981.

付録1 サブルーチンライブラリ概説

利用頻度の高いDIPでの機能はフォートランで使えるように、ライブラリ化されている。現在、用意されているサブルーチンは次の8グループの43個である。

- ① DIP全体の制御に関するサブルーチン
- ② デジタルデータ入出力の制御用サブルーチン
- ③ TV画像の入出力に関するサブルーチン
- ④ ディスプレイの制御用サブルーチン
- ⑤ フレームメモリの読み書きに関するサブルーチン
- ⑥ 画像メモリの読み書きに関するサブルーチン
- ⑦ 画像演算モジュール(PPPと略す。)での演算Iに関するサブルーチン
- ⑧ PPPでの画像演算IIに関するサブルーチン

このライブラリを使い、DIPを動作させる手順は次のようになる。

- ① DIPOPNをコールしDIPの占有宣言をする。
- ② PPPの基本演算IIあるいはユーザの作成した演算を使うときはマイクログラムをPPPプログラムメモリにロードする。
- ③ 画像処理用なサブルーチンをコールする。
連続して次のサブルーチン実行する時は、前回のジョブが終了しているかどうかをDIPSTSを使って調べる。
- ④ 最後にDIPCSLにより占有の解除をする。

個々のサブルーチンの詳細は付録2で説明する。

これらのサブルーチンは共通の I/Oルーチンをコールし、パラメータに相当する数値や画像データをDIP への転送のために引き渡す。このI/O ルーチンは計算機ごとに用意しなければならない。付録3ではPRIME 用I/O ルーチンとDIP のモジュールで定義されているパラメータについて説明する。付録ではPRIME 400 の名称を使っているが、これは、PRIME 用のDIP のソフトウェアは、最初、 PRIME 400で開発され、その後、計算機の入替でPRIME 750 になったことによる。

ところで、PPP 内はマイクロプログラム制御になっており、そのプログラム領域は4Kワード(40 ビット/ワード)である。このうち1KワードはROM 化されており、⑦のグループの演算に関する制御プログラムが入っている。残り3KワードはRAM で、ユーザはこの領域を使ってPPP の画像演算プログラムを作ることができる。現在、この一部分に⑧のグループに関する制御プログラムが入っている。制御マイクロプログラムは、通常のアセンブラに似た言語とそのクロスアセンブラがPRIME に用意されているので、これを使い作成する。また、このマイクロプログラムをPPP にロードしたり、これをコールするサブルーチンも用意されている。この部分の詳細は付録4で述べる。

付録 2 DIP 用サブルーチン説明書

§1 はじめに

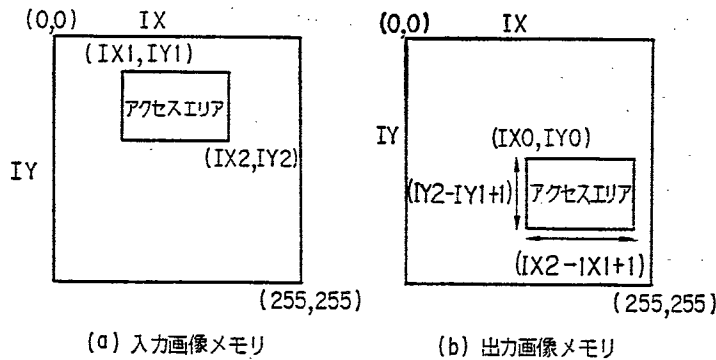
ここでは付表1に示される DIP 用サブルーチンについて説明する。これらのサブルーチンは PRIME-400 のフォートランで DIP を動作させる時に使用するもので、LIB>DIPOML にライブラリ形式で登録

されている。DIPOML 内のサブルーチンは、さらに、PRIME 400 が提供するサブルーチンを呼んでいるので、リンケージをとる時は、EXCPW, VAPPLB, FORTRAN 用ライブラリの3つのファイルが必要である。さらに実行時には、ユーザのプログラムとをロジカルに接続するためのアサインコマンドが必要となる。この手順を示すと下記のようになる。

付表 1 DIPOML に登録されているサブルーチン

参照番号	サブルーチン名	機 能
1-1	DIPOPN	占有宣言
1-2	DIPCLS	占有解除
1-3	DIPRST	初期化
1-4	DIPSTS	ステータスの読み込み
1-5	DIPDSW	データ SW の値の読み込み
2-1	DIPIN	デジタル入力
2-2	DIPOUT	デジタル出力
3-1	DIPVDI	TV カメラ → DIP 画像メモリ
4-1	DIPWDT	画像メモリ用データ変換テーブル書替え
4-2	DIPWFR	フレームメモリ用データ変換テーブル書替え
5-1	DIPWFM	計算機 → フレームメモリ
5-2	DIPRFM	フレームメモリ → PPP 計算機
5-3	PPPFXT	フレームメモリ → PPP 画像メモリ
5-4	PPPDIF	PPP 画像メモリ → フレームメモリ
6-1	DIPWIM	計算機 → DIP 画像メモリ
6-2	DIPRIM	DIP 画像メモリ → 計算機
6-3	PPPWIM	計算機 → PPP 画像メモリ
6-4	PPPRIM	PPP 画像メモリ → 計算機
6-5	DIPPIM	DIP 画像メモリ → PPP 画像メモリ
6-6	PPPDIM	PPP 画像メモリ → DIP 画像メモリ
6-7	DIPMTM	DIP 画像メモリ → DIP 画像メモリ
6-8	PPPMTM	PPP 画像メモリ → PPP 画像メモリ
7-1	PPPFLT	積和演算
7-2	PPPHST	ヒストグラム
7-3	PPPDCV	データ変換
7-4	PPPLFL	論理フィルター
7-5	PPPLAB	領域番号付
8-1	PPPSTC	定数の設定
8-2	PPPBTS	ビットシフト
8-3	PPPABS	絶対値
8-4	PPPLST	レベルシフト
8-5	PPPADD	加 算
8-6	PPPSUB	減 算
8-7	PPPMLT	乗 算
8-8	PPPDIV	除 算
8-9	PPPDCC	データ比較と変換
8-10	PPPOR	論 理 和
8-11	PPPAND	論 理 積
8-12	PPPKOR	排他論理和
8-13	PPPCMP	大小比較・変換
8-14	PPPRNG	領域の最大、最小座標値
8-15	PPPGRA	重心の計算
8-16	PPPMON	モーメントの計算

【注意】
5-3, 8-1~8-16
のサブルーチン
は PPP の RAM
領域のマイクロ
プログラムをコ
ールする。



付図 1 アクセス領域の指定

OK, SEG

$\#_VL_ \#SYSUT 2$ } ロードモジュール用の
ファイル定義

$\$_L\bar{O}_ \#_B_SYSIPT$ } メインプログラムのオブ
ジェクトファイルの
ロード

$\$_LI_DIPOML$
 $\$_LI_EXCPW$
 $\$_LI_VAPPLB$
 $\$_LI$ } DIP 用ライブラリの
ロード

$\$_SAVE$

$\$_Q$

OK, AS_MG } DIP のアサイン命令

OK, SEG_#SYSUT 2 } ロードモジュールの実
行

§ 2 共通性のある引数の説明

次節で各サブルーチンの使いかたを説明するが、引数のうち、共通性のあるものはここでまとめて述べる。

1) 画像メモリ層の指定

DIP 本体には 3 層の画像メモリと 3 層のフレームメモリ、PPP 内には 4 層の画像メモリがある。これらのメモリを使用するサブルーチンでは、メモリ層を指定する引数があり、メモリ層に対応した整数値を与えなければならない。

IM : } DIP 本体の画像メモリを表わす引数で、
IM1 : } 1~3 の整数でメモリ層を指定する。
IM2 : }
IFM : } フレームを表わす引数で、1~3 の整数
で指定する。

IRM : } PPP 内の画像メモリ層を表わす引数で、
1~4 の整数値が許される。ただし、
IRM1 : } IRM, IRM1, IRM2 は入力画像メモリ
IRM2 : } で、IWM は出力画像メモリを表わす。
IWM : } 同一のサブルーチン内では、入力と出力
の引数の値は違ったものでなければなら
ない。

2) アクセス領域の指定

画像メモリ（あるいはフレームメモリ）へのアクセスは付図 1 に示すような矩形形状の領域単位で行う。この領域の位置とサイズを指定するために次のような引数がある。

IX1 : } 0~255 の整数値でアクセス領域の始点
IY1 : } 座標を指定する。

IX2 : } 0~255 の整数値でアクセス領域の終点
IY2 : } 座標を指定する。

IX0 : } 0~255 の整数値で出力画像メモリのア
IY0 : } クセス領域の始点座標を指定する。

画像メモリとこれらの引数の関係は付図 1 のようになる。同図(a)は 1 つの画像メモリに計算機からアクセスする場合、または、画像メモリ間でデータを転送する場合の入力画像メモリとの関係を表わし、同図(b)は出力画像メモリのアクセス領域を示す。但し、画像メモリヘデータを出力する場合でも、(IX0, IY0) が指定できないサブルーチンがあるが、この時は IX0=IX1, IY0=IY1 となる。

3) エラー情報

計算機から DIP パラメータあるいはデータの転送が正常終了かいないかを知ることができる。

IFB : エラー情報を示す引数で正常終了ならば 0 を返す。異常終了ならば、すでに転送が完了したワード数 +1 の値を返す。

異常終了の場合、ユーザの端末に下記のようなエラーメッセージが表示される。

“EXCDIP(*i*)ERROR! STS(3)=*j*

STS(1)=*k*”

i=P パラメータ転送時のエラー

=R データリードの時のエラー

=W データライトの時のエラー

j= エラーコード

k= 完了転送ワード数

4) 引数の型

画像データのデータエリア (IA) あるいはデータテーブル用のデータエリア (IDT) はそのローケーション情報がサブルーチンに引き渡されるので型式は任意でよい。しかし、これ以外の、例えば画像メモリのアドレスを示す引数、モード指定の引数、エラー情報の引数などは整数4バイト長の引数でなければならない。

§ 3 サブルーチンの仕様

1.1 DIPOP

〈機能〉

DIP の占有制御回路へ占有宣言を行う。

〈Calling Sequence〉

CALL DIPOP (ISB)

〈引数の説明〉

ISB : 占有情報を返す。

〈使用上の注意〉

ISB=0 ならば占有でき、以後 DIP の使用が可能である。ISB≠0 ならば占有できず。このとき ISB の下位8ビットは DIP の内部ステータスを示す。そのステータス情報は DIPSTS で得られるものと同じである。

1.2 DIPCLS

〈機能〉

DIP 装置の占有解除を行う。

〈Calling Sequence〉

CALL DIPCLS

1.3 DIPRST

〈機能〉

DIP 装置のシーケンスの初期化を行う。

〈Calling Sequence〉

CALL DIPRST

1.4 DIPSTS

〈機能〉

DIP の内部ステータスを見る。

〈Calling Sequence〉

CALL DIPSTS (ISB)

〈引数の説明〉

ISB : ステータス情報

〈使用上の注意〉

①ステータスは下位8ビットにあり、ビットが立っていると次のような意味をもつ。

2⁷ :

2⁸ : PPP ERROR

2⁹ : PPP EOM

2⁴ : PPP 電源 ON

2³ :

2² : DIP バス busy

2¹ : 占有 OK

2⁰ : DIP の電源 ON

②ステータスは DIP 占有に関係なく見ることができ。

1.5 DIPDSW

〈機能〉

DIP 装置のパネル上の8ビットのデータスイッチの値を読み込む。

〈Calling Sequence〉

CALL DIPDSW (IOB)

〈引数の説明〉

IOB : データスイッチの値 (0~255)

2.1 DIPIN

〈機能〉

デジタル入力回路よりデータ (16ビット) をレコードサイズで指定された個数入力し、計算機に転送する。

〈Calling Sequence〉

CALL DIPIN (IA, N, IFB)

〈引数の説明〉

IA : データを格納するエリア

N : レコードサイズ

〈使用上の注意〉

①time out 機能を抑制しているので、途中でとぎれると、計算機は hung up する。

②指定できるサイズ (N) は 1~255 で、データエリアは N*2 バイト必要であ。

2.2 DIPOUT

<機能>

計算機からのデータ (16 ビット) をレコードサイズで指定された個数だけデジタル出力回路より出力する。

<Calling Sequence>

CALL DIPOUT (IA, N, ICH, IFB)

<引数の説明>

IA : データ格納エリア
N : レコードサイズの指定
ICH : 出力チャネルの指定 (1 or 2)

<使用上の注意>

DIPIN と同じ。

3.1 DIPVDI

<機能>

TV カメラからの画像データを本体の画像メモリに書込む。

<Calling Sequence>

CALL DIPVDI (IM, IW, IX, IY, IFB)

<引数の説明>

IM : 画像メモリ層の指定
IW : ウィンドウサイズの指定
IX : }
IY : } ウィンドウ始点アドレス

<使用上の注意>

- ①IM=1, 2 の場合, 高分解能画像入力装置からのデータがそれぞれの画像メモリに入力される。TV 画像は 1024*1024 画素あるので, ウィンドウにより, 入力する部分を指定の必要がある。ウィンドウサイズは,

IW=1 ならば 1/16 サイズ
2 ならば 1/4 サイズ
3 ならば 1/1 サイズとなる。

TV 画面上のウィンドウアドレス (X, Y) は,

$X=8*IX$
 $Y=4*IY$

となる。但し $0 \leq IX \leq 127$, $0 \leq IY \leq 255$ である。

- ②IM=3 ならば, 標準の TV カメラからの画像データを画像メモリに書込む。この場合, 他の引数 (IW, IX, IY) は意味をもたない。

4.1 DIPWDT

<機能>

ディスプレイ装置内にある画像メモリ用のデータ

変換テーブルを書替える。

<Calling Sequence>

CALL DIPWDT (IM, IDT, IFB)

<引数の説明>

IM : データ変換テーブルの指定
IDT : 変換データの格納エリア名 (512 バイト)

<使用上の注意>

- ①各画像メモリにデータ変換テーブルがそれぞれ対応しており, IM は 1~3 の整数値で指定する。
②変換テーブルは 256 ワード (1 ワード 16 ビット) で, 各ワードの下位 8 ビットが変換データになる。画像メモリの各画素は, その数値 +1 のアドレスにある変換データを濃淡値として, ディスプレイされる。

4.2 DIPWFR

<機能>

ディスプレイ装置内にあるフレームメモリ用のデータ変換テーブル, 輝度レジスタ, ビデオ信号切換レジスタにデータを転送する。

<Calling Sequence>

CALL DIPWFR (IDT, IFB)

<引数の説明>

IDT : データ格納エリア (28 バイト)

<使用上の注意>

- ①IDT の各ワード (16 ビット) の意味は付図 2 のようになる。但し, 各ワードの下位 8 ビットが有効データである。
②FIR 0~3 はフレームメモリの画素 (2 ビット) をビデオ信号にする時の濃淡値 (8 ビット) を決めるレジスタである。画素の値 (0~3) により,

IDT(1)	FIR 0	}	フレームメモリ1の データ変換テーブル
	1		
	2		
	3		
IDT(5)	FIR 0	}	フレームメモリ2の データ変換テーブル
	1		
	2		
	3		
IDT(9)	FIR 0	}	フレームメモリ3の データ変換テーブル
	1		
	2		
	3		
IDT(13)	I REG		輝度レジスタ
IDT(14)	S REG		ビデオ切り換えレジスタ

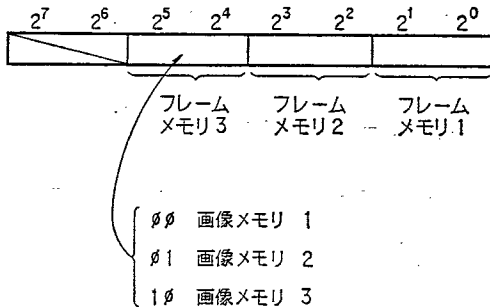
付図 2

FIR 0~3 が参照されるが、画素の値により表示されるモードも下記のように変わる。

画素の値	混合回路からの出力
0	FIR 0 のレベル+画像メモリのレベル
1	FIR 1 のレベル+画像メモリのレベル
2	FIR 2 のレベルのみ
3	FIR 3 のレベルのみ

③IREG は全画面を一定の濃淡レベル（8ビット）で表示する時に使用する。IREG の出力は画像メモリの出力と並行して混合回路に入っており、手動スイッチにより切換えることができる。

④画像メモリ 1, 2, 3 は混合回路でフレームメモリ 1, 2, 3 と混合され、ビデオ信号が合成される。SREG は、プログラムで画像メモリとフレームメモリの組合せを決めるためのスイッチングレジスタで、付図 3 のようにフレームメモリに対応したビット位置に画像メモリを示すビットパターンを指定すればよい。



5.1 DIPWFM

<機能>

計算機のデータを DIP のフレームメモリに転送する。

<Calling Sequence>

CALL DIPWFM (IFM, IX 1, IY 1, IX 2, IY 2, IA, MODE, IFB)

<引数の説明>

MODE: データ形式の指定

<使用上の注意>

DIPRFM と同じ

5.2 DIPRFM

<機能>

DIP のフレームメモリのデータを計算機に転送

する。

<Calling Sequence>

CALL DIPRFM (IFM, IX 1, IY 1, IX 2, IY 2, IA, MODE, IFB)

<引数の説明>

MODE: データ形式の指定

<使用上の注意>

①データ形式

フレームメモリの 1 データは 2 ビットである。1 ワード (16 ビット) の下位 2 ビットに 1 データ、あるいは、8 データ分がパックされた 2 種類のデータ形式が扱える。

②転送モード

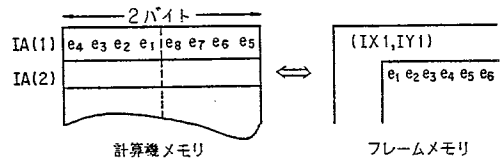
フレームメモリのディスプレイ表示のブランキング期間に読書きするモードと、ディスプレイを停止して優先的に読書きする 2 種類のモードがある。

③引数の与え方

モード指定	データ形式	転送モード
MODE=0	1 ワードに 8 データ	優先的に
1	"	ディスプレイのブランキング期間のみ
2	1 ワードに 1 データ	優先的に
3	"	ディスプレイのブランキング期間のみ

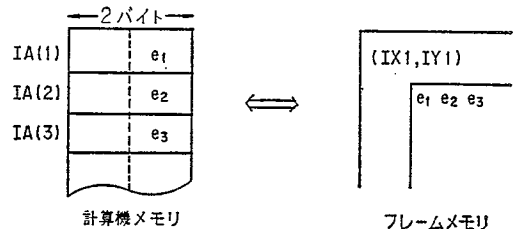
④フレームメモリのアドレスとモードの関係

a) MODE=0, 1 の場合は付図 4 のようになる。



ただしアドレスは下記のような制限がある。

IX1=8*k IY1=l (0 ≤ k ≤ 31)



$$IX2=8*k+7 \quad IY2=n \quad (0 \leq l \leq n \leq 255)$$

b) MODE=2, 3の場合は付図5のようになる。

5.3 PPPFXT

<機能>

フレームメモリ (IFM) のデータを PPP 内の画像メモリ (IWM) に転送する。

<Calling Sequence>

CALL PPPFXT (IFM, IWM, IX1, IY1, IX2, IY2, IFB)

<使用上の注意>

PPP メモリの各画素 (8ビット) は上位6ビットに0が、下位2ビットにフレームメモリのデータが入る。

5.4 PPPDIF

<機能>

PPP 内の画像メモリ (IRM) のデータをフレームメモリ (IFM) に転送する。

<Calling Sequence>

CALL PPPDIF (IRM, IFM, IX1, IY1, IX2, IY2, IFB)

<使用上の注意>

PPP 内の画像メモリの各画素 (8ビット) の下位2ビットがフレームメモリに転送される。

6.1 DIPWIM

<機能>

計算機からデータを DIP の画像メモリ (IM) に転送する。

<Calling Sequence>

CALL DIPWIM (IM, IX1, IY1, IX2, IY2, IA, MODE, IFB)

<引数の説明>

IA : データエリア名

MODE : データ形式と転送モードの指定

<使用上の注意>

DIPRIM と同じ

6.2 DIPRIM

<機能>

DIP 画像メモリ (IM) から計算機へ画像データを転送する。

<Calling Sequence>

CALL DIPRIM (IM, IX1, IY1, IX1, IY2,

IA, MODE, IFB)

<引数の説明>

IA : データエリア名

MODE : データ形式と転送モードの指定

<使用上の注意>

①データ形式

画像メモリの1画素は8ビットである。1ワード (16ビット) の下位8ビットに1画素と、2画素がパックされた、2種類のデータ形式が扱える。

②転送モード

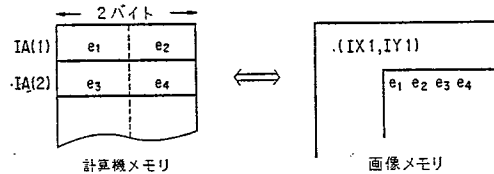
画像メモリのディスプレイ表示のブランキング期間に読書きするモードと、ディスプレイを停止して優先的に読書きする2種類のモードがある。

③引数の与えかた

モード指定	データ形式	転送モード
MEDO=0	1ワードに2データ	優先的に
1	"	ディスプレイのブランキング期間のみ
2	1ワードに1データ	優先的に
3	"	ディスプレイのブランキング期間のみ

③画像メモリと計算機メモリの対応

a) MODE=0, 1の場合は付図6のようになる。



付図6

アドレスには次のような制限がつく。

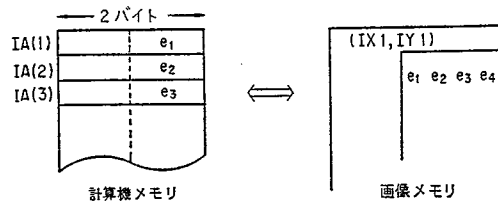
$$IX1=2*k \quad (0 \leq k \leq 127)$$

$$IX2=2*k+1$$

$$IY1=l \quad (0 \leq l \leq n \leq 255)$$

$$IY2=n$$

b) MODE=2, 3の場合は付図7のようになる。



付図7

6.3 PPPWIM

〈機能〉

計算機のデータを PPP 内の画像メモリに転送する。

〈Calling Sequence〉

CALL PPPWIM (IWM, IX1, IY1, IX2, IY2, IA, MODE, IFB)

〈引数の説明〉

IA : データが格納されているエリア名

MODE : データ形式の指定

〈使用上の注意〉

PPPRIM と同じ

6.4 PPPRIM

〈機能〉

PPP 内の画像メモリから計算機へデータを転送する。

〈Calling Sequence〉

CALL PPPRIM (IRM, IX1, IY1, IX2, IY2, IA, MODE, IFB)

〈引数の説明〉

IA : データが格納されるエリア名

MODE : データ形式の指定

〈使用上の注意〉

①データ形式の指定

画像メモリの1画素は8ビットである。

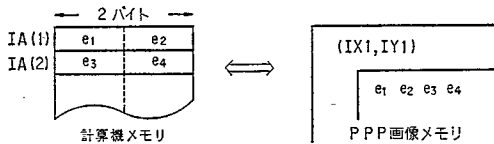
MODE=0 ならば1ワード(16ビット)に2画素

MODE=2 ならば1ワードの下位8ビットに1画素が入っているものとしてあつかう。

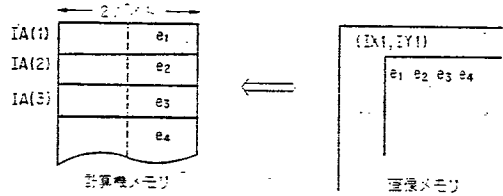
②画像メモリのアドレスとデータ形式の関係

a) MODE=0 の場合は付図8のようになる。

b) MODE=2 の場合は付図9のようになる。



付図 8



付図 9

6.5 DIPPIM

〈機能〉

DIP の画像メモリ (IM) から PPP の画像メモリにデータを転送する。

〈Calling Sequence〉

CALL DIPPIM (IM, IWM, IX1, IY1, IX2, IY2, IFB)

6.6 PPPDIM

〈機能〉

PPP 内の画像メモリ (IRM) から DIP の画像メモリ (IM) へデータを転送する。

〈Calling Sequence〉

CALL PPPDIM (IRM, IM, IX1, IY1, IX2, IY2, IFB)

6.7 DIPMTM

〈機能〉

DIP の画像メモリ間のデータ転送を行う。

〈Calling Sequence〉

CALL DIPMTM (IM1, IM2, IX1, IY1, IX2, IY2, IFB)

〈使用上の注意〉

データは、IM1 から IM2 の画像メモリに転送される。

6.8 PPPMTM

〈機能〉

PPP 内の画像メモリ間のデータ転送を行う。

〈Calling Sequence〉

CALL PPPMTM (IRM, IWM, IX1, IY1, IX2, IY2, IXO, IYO, IFB)

7.1 PPPFLT

〈機能〉

入力画像メモリのデータを荷重テーブルの重みに

よる積和演算を行い、その結果を出力画像メモリに書きこむ。

〈Calling Sequence〉

CALL PPPFLT (IRM, IWM, IX1, IY1, IX2, IY2, IXO, IYO, IWT, MODE, IFB)

〈引数の説明〉

IWT : 荷重テーブル用データエリア

MODE : 演算モード指定 (24バイト)

MODE(1)=KX } 荷重テーブルのサイズ
 MODE(2)=KY }
 MODE(3)=IT 入力データの整型の指定
 MODE(4)=BIAS 正規化のためのバイアス値
 MODE(5)=NORM 正規化のためのシスト量
 MODE(6)=IS 正規化の時のオーバーフロー処理

〈使用上の注意〉

①荷重テーブルは、横方向を KX、縦方向を KY で指定される 2次元のテーブルで、その最大値は 8*8 である。

②IWT は IWT (KX, KY) で指定される 2次元配列で、各ワードのサイズは 16ビットである。各ワードの下位 8ビットに荷重データが格納される。各荷重データは 8ビット 2の補数とみなされる。

③入力データの数型の指定方法

IT=0 : 入力データは 2の補数として扱う。

IT=1 : 入力データは 8ビットの正の整数表示とみなす (演算器内では 2の補数に変換される)。

④正規化のための引数の指定

正規化は $F_0 = (F + \text{BIAS}) / 2^n$ で行う。

BIAS : 24ビット整数

NORM : 表にあるように n を指定するコード

NORM	n	NORM	n
X'F'	0	X'7'	7
X'E'	1	X'6'	8
X'D'	2	X'5'	9
X'8	3	X'0'	10
X'9	4	X'1'	11
X'A'	5	X'2'	12
X'B'	6	X'3'	13

⑤オーバーフロー処理

IS=0 : 下位 8ビットをそのまま出力する

IS=1 : 正、負の最大値を出力する。(X'7F', X'80')

7.2 PPPHST

〈機能〉

画像メモリ内の指定された領域の濃度頻度分布をテーブルメモリに作成する。

〈Calling Sequence〉

CALL PPPHST (IRM, IX1, IY1, IX2, IY2, MODE, IDT, IFB)

〈引数の説明〉

MODE : テーブルメモリの初期化の指定

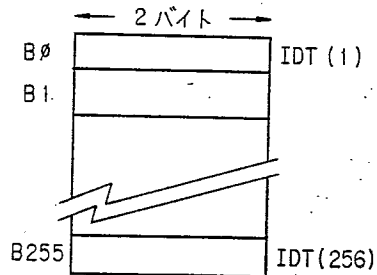
IDT : テーブルメモリ (512バイト)

〈使用上の注意〉

①MODE=0 ならばテーブルメモリを初期化 (オール 0) する。

MODE=1 ならば現在の値に頻度を加算していく。

②各画素 (Bi) の頻度 (16ビット) は付図 10 の形式でテーブルメモリに格納される。オーバーフローした時は、 $2^{16}-1$ の値が入っている。



付図 10

7.3 PPPDCV

〈機能〉

入力画像メモリ of データをデータ変換テーブルにもとづいて変換し、他の画像メモリに出力する。

〈Calling Sequence〉

CALL PPPDCV (IRM, IWM, IX1, IY1, IX2, IY2, IXO, IYO, IDT, IF)

〈引数の説明〉

IDT : 変換テーブル (512バイト)

〈使用上の注意〉

①変換テーブルのサイズを 256 ワードとすると、各

ワード (16 ビット) の下位 8 ビットが変換データになる。

7.4 PPPLFL

<機能>

入力画像メモリの各画素をその近傍 (3×3) の値にもとづいて変換し、他の画像メモリに出力する。

<Calling Sequence>

CALL PPPLFL (IRM, IWM, IX1, IY1, IX2, IY2, IXO, IYO, IDT, IFB)

<引数の説明>

IDT : 変換テーブル (1024 バイト)

<使用上の説明>

- ①入力データは画像メモリの各データ (8 ビット) の下位 1 ビットである。
- ②3×3のマトリックスは付図 11 のように重みづけがされており、そのビットパターン (2^k が注目している画素) により 0~511 までの 1 つの数値を割りつける。この数値を変換テーブルの参照アドレスとし、その変換テーブルの内容を出力する。

2^8	2^7	2^6
2^5	2^4	2^3
2^2	2^1	2^0

付図 11 3×3のマトリックスの重みづけ

- ③交換テーブル (512 ワード) の各ワード (16 ビット) の下位 8 ビットが変換データである。

7.5 PPPLAB

<機能>

2 値化された入力画像中の連結した領域を見つけ、各領域にそれぞれ番号を割りつける。

<Calling Sequence>

CALL PPPLAB (IRM, IWM, IX1, IY1, IX2, IY2, IXO, IYO, ICON, IOV, N, IFB)

<引数の説明>

ICON : 連結モード指定

IOV : オーパフローした時の処置方法の指定

N : 演算結果

<使用上の注意>

- ①入力画像の背景になるものに 0 を入れ、領域に相当するものに 1 を入れなければならない。

- ②ICON=0 ならば 4 連結領域を見つける。

ICON=1 ならば 8 連結領域を見つける。

- ③初回の領域付のとき IOV=0 とし、オーバーフローなどで再度領域付をやりなおすときは IOV=1 とすること。

- ④領域付演算結果は次の 3 通りがある。

$2 \leq N \leq 255$ ならば 正常に終了し、領域番号の最大値が N である。

N=256 ならば領域番号付のテーブルがオーバーフローした。

IOV=1 と指定し、再度領域付を行う必要あり。

$N \geq 1000$ ならば領域付番号が 255 を越え、これ以上領域付が不可能である。

N は領域番号付されずに残った画素の最小 Y アドレスに 1000 が加算されたものである。

- ⑤入力画像の番号付された領域は、その数値に入れかわっており、原画像は保存されない。

- ⑥入力画像エリアの始点は (0, IY1), 終点は (255, IY2) でなければならない。

8.1 PPPSTC

<目的・機能>

PPP 内の画像メモリに定数データを書込む。

<Calling Sequence>

CALL PPPSTC (IWM, IX1, IY1, IX2, IY2, IN, IFB)

<引数の説明>

IN : 書込みたい定数

<使用上の注意>

IN は 0~255 の整数値である。

8.2 PPPBTS

<目的・機能>

入力画像メモリの各データの値を左あるいは右に 1 ビットシフトし、出力画像メモリに出力する。

<Calling Sequence>

CALL PPPBTS (IRM, IWM, IX1, IY1, IX2, IY2, IXO, IYO, MODE, IFB)

<引数の説明>

MODE : シフト方向の指定

<使用上の注意>

MODE=0 ならば右に 1 ビットシフトする。

MODE=1 ならば左に 1 ビットシフトする。

8.3 PPPABS

<機能>

入力画像のメモリのデータ (M1) を2の補数表示とみなし、この絶対値 (M2) を出力画像メモリに出力する。

<Calling Sequence>

CALL PPPABS (IRM, IWM, IX1, IY1, IX2, IY2, IXO, IYO, MODE, IFB)

<引数の説明>

MODE: 絶対値のとりかたの指定

<使用上の注意>

MODE=0 ならば $M2=|M1|$ となる。

MODE=1 ならば $M2=|M1|*2$ となる。

8.4 PPPLST

<機能>

入力画像メモリのデータ (M1) に定数 "N" を加算し出力画像メモリに計算結果 (M2) を出力する。

<Calling Sequence>

CALL PPPLST (IRM, IWM, IX1, IY1, IX2, IY2, IXO, IYO, MODE, IN, IFB)

<引数の説明>

MODE: 入力, 出力画像データの数型指定

IN : 加算する定数

<使用上の注意>

①IN は8ビットの2の補数

②MODE=0ならば入力は正の整数として扱い、結果も正の整数にする。

MODE=1ならば入力は2の補数として扱い、結果も2の補数にする。

MODE=2ならば入力は正の整数として扱い、結果は2の補数にする。

8.5 PPPADD

<機能>

2つの画像メモリ間で各データの加算を行い、結果を他の画像メモリに出力する。

<Calling Sequence>

CALL PPPADD (IRM1, IRM2, IWM, IX1, IY2, IX2, IY2, IXO, IYO, MODE, IFB)

<引数の説明>

MODE: データの数型指定

<使用上の注意>

①入力画素を M1, M2, 出力画素を M3 としたと

き,

MODE=0 ならば

$$M3(8\text{ビット})=M1(8\text{ビット})+M2(8\text{ビット})$$

MODE=1 ならば

$$M3(2'S)=M1(2'S)+M2(2'S)$$

MODE=2 ならば

$$M3(8\text{ビット})=M1(2'S)+M2(2'S)$$

となる。

②オーバーフローしたときはその数型の最大値, あるいは最小値が出力される。

8.6 PPPSUB

<機能>

2つの画像メモリ間で各データの減算を行い、その結果を他の画像メモリに出力する。

<Calling Sequence>

CALL PPPSUB (IRM1, IRM2, IWM, IX1, IY1, IX2, IY2, IXO, IYO, MODE, IFB)

<引数の説明>

MODE: データの数型指定

<使用上の注意>

①IRM1 の画素を M1, IRM2 の画素を M2, IWM の画素を M3 とすると,

MODE=0 ならば

$$M3(8\text{ビット})=M1(8\text{ビット})-M2(8\text{ビット})$$

MODE=1 ならば

$$M3(2'S)=M1(2'S)-M2(2'S)$$

MODE=2 ならば

$$M3(2'S)=M1(8\text{ビット})-M2(8\text{ビット})$$

となる。

②オーバーフローしたときはその数型の最大値, あるいは最小値が出力される。

8.7 PPPMLT

<機能>

2つの入力画像メモリの間で、各データの乗算を行い、出力画像メモリに結果を出力する。

<Calling Sequence>

CALL PPPMLT (IRM1, IRM2, IWM, IX1, IY1, IX2, IY2, IXO, IYO, IBIAS, ISFT, IFB)

<引数の説明>

IBIAS: バイアス値

ISFT: シフト量

<使用上の注意>

IRM1 と IRM2 の各画素の値を M1, M2 とすると、演算結果 M3 は、

$$M3 = (M1 * M2 + IBIAS) / 2^{ISFT}$$

になる。数型はすべて8ビット整数で、オーバーフローしたときは下位8ビットが出力される。

8.8 PPPDIV

<機能>

2層の入力画像メモリ間の各データで除算を行い、出力画像メモリに結果を出力する。

<Calling Sequence>

CALL PPPDIV (IRM1, IRM2, IWM, IX1, IY1, IX2, IY2, IXO, IYO, IFB)

<使用上の注意>

IRM1 の画素を M1, IRM2 の画素を M2 とすると演算結果 M3 は

$$M3 = M1 / M2$$

となる。M2=0 のときは M3=0、また、は **小数点以下** 切りすてる。

8.9 PPPDCC

<機能>

入力画像メモリの各データを定数と比較し、その大小により、指定された定数あるいは別の画像メモリの対応するデータを出力画像メモリに書込む。

<Calling Sequence>

CALL PPPDCC (IRM1, IRM2, IWM, IX1, IY1, IX2, IY2, IXO, IYO, MODE, IN1, IN2, IFB)

<引数の説明>

MODE : 変換方式の指定

IN1 : 比較用定数

IN2 : データ変換用定数

<使用上の注意>

IRM1 の画素を M1, IRM2 の画素を M2 とすると、出力画素 M3 は以下ようになる。

MODE=0 のとき

M1=IN1 ならば M3=IN2, そして、

M1≠IN1 ならば M3=M2 である。

MODE=1 のとき

M1=IN1 ならば M3=M2,

M1≠IN1 ならば M3=IN2 である。

8.10 PPPOR

<機能>

2の入力画像メモリ間で各データの論理和をとり、他の画像メモリに結果を出力する。

<Calling Sequence>

CALL PPPOR (IRM1, IRM2, IWM, IX1, IY1, IX2, IY2, IXO, IYO, IFB)

<使用上の注意>

2つの入力画像メモリの各データを M1, M2 とすると、演算結果 M3 は下表ようになる。

M1	M2	0
0	0	0
0	1~255	1
1~255	0	1
1~255	1~255	1

8.11 PPPAND

<機能>

2つの入力画像メモリ間で各データの論理積をとり、結果を出力画像メモリに出力する。

<Calling Sequence>

CALL PPPAND (IRM1, IRM2, IWM, IX1, IY1, IX2, IY2, IXO, IYO, IFB)

<使用上の注意>

2つの入力画像メモリの各データ M1, M2 とすると演算結果 M3 は下表ようになる。

M1	M2	M3
0	0	0
0	1~255	0
1~255	0	0
1~255	1~255	1

8.12 PPPXOR

<機能>

2つの入力画像メモリ間で、各データの論理排他和を演算し、他の画像メモリに結果を出力する。

<Calling Sequence>

CALL PPPXOR (IRM1, IRM2, IWM, IX1, IY1, IX2, IY2, IXO, IYO, IFB)

<使用上の注意>

2つの入力画像メモリの各データを M1, M2 とすると、演算結果 M3 は下表ようになる。

M1	M2	M3
0	0	0
0	1~255	1
1~255	0	1
1~255	1~255	0

8.13 PPPCMP

〈機能〉

2つの入力画像メモリ間で各データの大きさを比較し、その結果にもとづいて定数を出力画像メモリに出力する。

〈Calling Sequence〉

CALL PPPCMP (IRM1, IRM2, IWM, IX1, IY1, IX2, IY2, IXO, IYO, MODE, IN1, IN2, IN3, IFB)

〈引数の説明〉

MODE: 入力データの数型指定

IN1 : }
IN2 : } 出力定数
IN3 : }

〈使用上の注意〉

①IRM1の画素の値をM1, IRM2の画素の値をM2とすると、IWMに出力されるM3は下記のようなになる。

M1 > M2 ならば M3 = IN1

M1 = M2 ならば M3 = IN2

M1 < M2 ならば M3 = IN3

②MODE=0のときM1, M2は正の整数,

MODE=1のときM1, M2は2の補数として取扱う。

8.14 PPPRNG

〈機能〉

入力画像メモリ中の番号付された領域のXY座標の最大値, 最小値を求め、テーブルメモリにその座標値を格納する。

〈Calling Sequence〉

CALL PPPRNG (IRM, IX1, IY1, IX2, IY2, IDT, IFB)

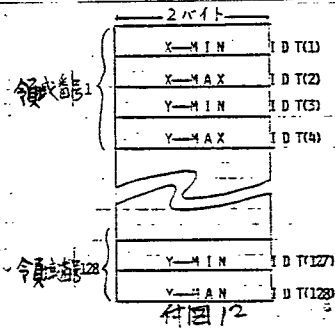
〈引数の説明〉

IDT: データテーブルメモリ (1024 バイト)

〈使用上の注意〉

①領域とみなされる画素の値は1~128でこれ以外の数値のついた画素は背景となる。

②座標値は付図12のようにテーブルメモリに格納される。



付図12

8.15 PPPGRA

〈機能〉

入力画像メモリ中の番号付された領域の重心を求め、その座標値をテーブルメモリに格納する。

〈Calling Sequence〉

CALL PPPGRA (IRM, IX1, IY1, IX2, IY2, IDT, IFB)

〈引数の説明〉

IDT: テーブルメモリ (512 バイト)

〈使用上の注意〉

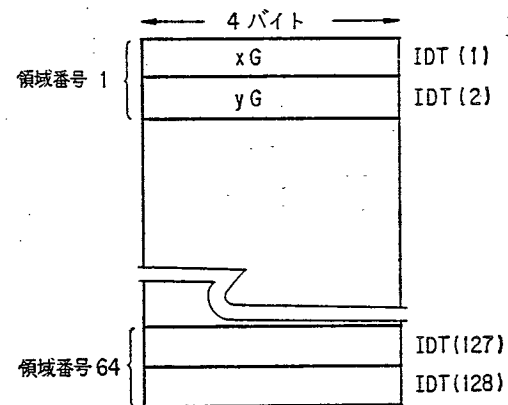
①画素値が1~64までは領域, これ以外の数値は背景とみなされる。

②同一番号の領域の各画素の座標を(x, y).....(x_n, y_n)とすると、重心(x_G, y_G)は下式のように定義される。

$$x_G = x_1 + x_2 + \dots + x_n$$

$$y_G = y_1 + y_2 + \dots + y_n$$

③重心座標値は付図13のようにデータテーブルに格納される。



付図13

8.16 PPPMON

〈機能〉

入力画像メモリ中の番号付された領域のモーメン

トを求めテーブルメモリに計算結果を格納する。

<Calling Sequence>

CALL PPPMON (IRM, IX1, IY1, IX2, IY2,
IDT, IFB)

<引数の説明>

IDT: テーブルメモリ (512 バイト)

<使用上の注意>

①画素値が 1~64 までは領域、これ以外の数値は、
背景として扱われる。

②同一番号領域の各画素の座標を $(x_1, y_1) \dots (x_m,$
 $y_m)$ とすると、モーメントの定義は、

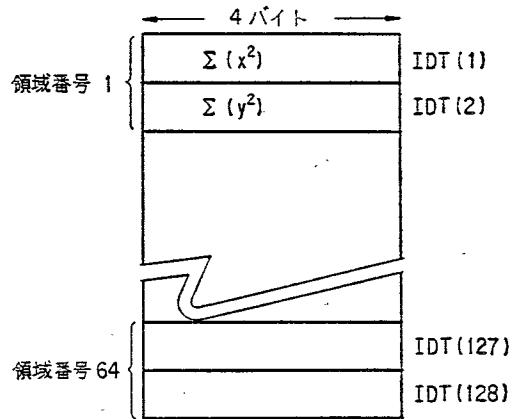
$$\Sigma(x^2) = x_1^2 + x_2^2 + \dots + x_m^2$$

$$\Sigma(y^2) = y_1^2 + y_2^2 + \dots + y_m^2$$

である。

③モーメントの計算結果は付図 14 の仕様でテーブ

ルメモリに格納される。



付図 14

付録3 DIP を動作させるための制御ルーチンとパラメータ

§ 1 DIP 用制御ルーチン

T\$MG

〈目的, 機能〉

PRIME 400 (P-400) から DIP への PP 信号の発生とデータの入出力を管理する。

〈Calling Sequence〉

CALL T\$MG (MG, N, IPP, IA, NW, ISTS, ITIME)

〈引数の説明〉

- MG : デバイス指定 (MG=0)
- N : GPIB (P-400 内の I/O インターフェース) への制御コマンド
- IPP : DIP の P-400 インターフェースに対する制御コマンド
- IA : データ格納エリア
- NW : データ数
- ISTS : GPIB のステータス情報格納エリア
- ITIME : タイムアウトの時間設定

〈使用上の注意〉

- ①各引数は INTEGER*2 で取扱われる。データ数の指定は語単位で行う。
- ②引数 N の与え方とその意味
 - N=1: Go
 - 2: Read
 - 3: Write
 - 5: Read Status
 - 6: PP 信号セットのみ
 - 7: GPIB の初期化
- ③引数 IPP の与え方 (オクタル表示)
 - IPP=000: PPCLR
 - 144: OCP
 - 142: NOCP
 - 146: WEXEC, EXEC
 - 126: REXEC
 - 141: PSET
 - 130: STS
 - 147: RST
- ④ステータス (ISTS の内容)
 - ISTS (1): エラーコード

ISTS (2): 終了フラグ (0 なら終了)

ISTS (3): 転送数

ISTS (4): コントローラステータス

- ⑤タイマー (ITIME) はオクタルで指定
 - 1 単位 40MS で最大 777 まで可能, また 0 ならばタイマーは働かない。
- ⑥P-400 メモリと DIP 内部バスのビット対応
 - 1 ワード (2 バイト) の先頭 1 バイトが DIP の内部バスの上位 1 バイトに対応する。
- ⑦引数 N, IPP の関係

N	IPP(オクタル表示)
2 (Read)	126 (REXEC) 130 (STS)
3 (Write)	146 (WEXEC) 141 (PSET)
6 (PP 信号)	000 (PPCLR) 144 (OCP) 142 (NOCP) 146 (EXEC) 147 (RST)

- ⑧PSET, WEXEC, REXC, STS のあとには PPCLR を出すこと。
- ⑨PPCLR, OCP, NOCP, RST は Go 信号 (N=1) で実行される。
- ⑩T\$MG による 1 つのコマンドの終了は ISTS (2) を見ることにより判定できる。

§ 2 各モジュールへの制御情報

DIP の各モジュールへのパラメータは 1 ワード (2 バイト) 単位になっており, PSET 命令で送出される。各ワードの上位 1 バイトはパラメータの行先を示すレジスタコードで, 下位 1 バイトが制御情報の構成になっている。

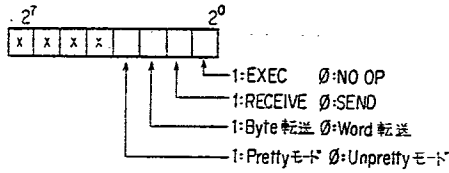
各モジュール内のレジスタコードは付表 1 のようになる。制御情報の意味は次のようになる。

付表 1 レジスタコード一覧表

画像メモリ	1	モードレジスタ (X'21')	X 1 アドレスレジスタ (X'22')	Y 1 アドレスレジスタ (X'23')	X 2 アドレスレジスタ (X'24')	Y 2 アドレスレジスタ (X'25')
"	2	" (X'31')	" (X'32')	" (X'33')	" (X'34')	" (X'35')
"	3	" (X'B1')	" (X'B2')	" (X'B3')	" (X'B4')	" (X'B5')
フレームメモリ	1	" (X'91')	" (X'92')	" (X'93')	" (X'94')	" (X'95')
"	2	" (X'D1')	" (X'D2')	" (X'D3')	" (X'D4')	" (X'D5')
"	3	" (X'E1')	" (X'E2')	" (X'E3')	" (X'E4')	" (X'E5')
画像データ制御回路	1	モードレジスタ (X'11')	X アドレスレジスタ (X'12')	Y アドレスレジスタ (X'13')		【注意】 X' 'は16進表示
"	2	" (X'81')	" (X'82')	" (X'83')		
"	3	" (X'C1')				
ディスプレイ制御回路		モードレジスタ (X'A1')				
データ入出力		モードレジスタ (X'41')	カウンターレジスタ (X'42')			
H-8350 インターフェース		モードレジスタ (X'51')				
P-400 インターフェース		モードレジスタ (X'F1')				
PPP インターフェース		モードレジスタ (X'61')	カウンターレジスタ 1 (X'62')	カウンターレジスタ 2 (X'63')	カウンターレジスタ 3 (X'64')	

a) 画像メモリ 1, 2, 3 及びフレームメモリ 1, 2, 3 関係

* モードレジスタ

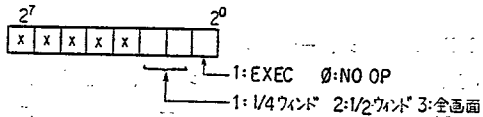


* X1, Y1, X2, Y2 アドレスレジスタ

画像メモリの R/W 領域を指定するアドレスで 0~255 までの数値。

b) 画像データ制御回路 1, 2 関係

* モードレジスタ

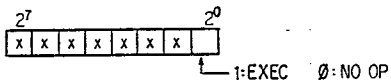


* X, Y アドレスレジスタ

TV 画像の入力領域を指定するアドレスで 0~255 までの数値。

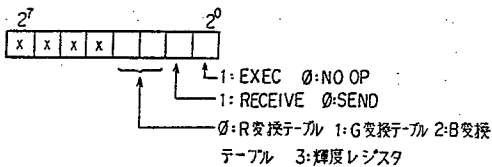
c) 画像データ制御回路 3

* モードレジスタ



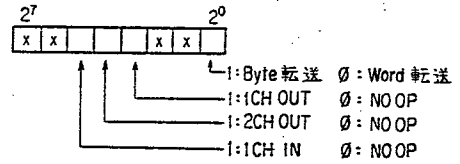
d) ディスプレイ制御回路

* モードレジスタ



e) デジタル入出力

* モードレジスタ

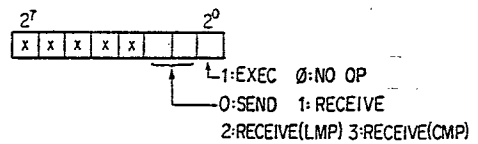


* カウンターレジスタ

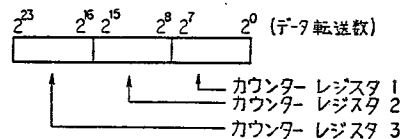
データ転送数を指定するカウンタで 1~255 までの数値。

f) 並列画像演算装置 (PPP) インターフェース

* モードレジスタ

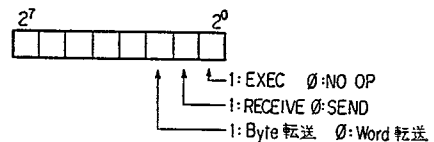


* カウンターレジスタ 1, 2, 3



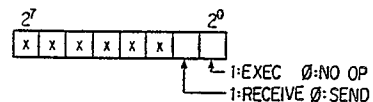
g) H-8350 インターフェース

* モードレジスタ



h) P-400 インターフェース

* モードレジスタ



付録 4 PPP 制御用マイクロプログラム

§ 1 クロスアセンブラ

PPP 制御用のアセンブラ及びこれをマイクロプログラムに翻訳、編集するクロスアセンブラは(株)構造計画作成の「並列演算クロスアセンブラ使用手引書」に説明されている。これはもともと HITAC 8350 のフォートランで動作する仕様になっているが、一部改造し、PRIME 400 で使えるようになっている。PRIME 400 で作成されるファイルは付図 1 のようになる。そして使用方法は次のようになる。

- ①PRIME 400 で <DIP> に login する。
- ②PPPAS に、アセンブラのソースプログラムを作成する。(1行 80 文字以内)
- ③SEG_#KKL で実行される。(端末に ***STOP ができるとコンパイルが完了)
- ④PPPOBJ にオブジェクトプログラムが出力されるが、PPPOBJ は 1レコード (20736 bytes) のファイルで、この構成は COMPMP で扱う PGM, WRK と共通である。
- ⑤PPPLIST に、ソースリストとシンボリックレファレンスリストが出力されるが、これを LP に出力するときは SPOOL_#PPPLIST_#-FTN のコマンドを使う。
- ⑥中間 File として

PPPSTAT, PPPEXP, PPSYMP が自動的に作成される。

§ 2 マイクロプログラムのロード及びコール

2.1 PPPLMP

<機能>

PPP にマイクロプログラムを転送する。

<Calling Sequence>

CALL PPPLMP (IA, ISAD, NB, IFB)

<引数の説明>

IA : マイクロプログラムが格納されているエリア名

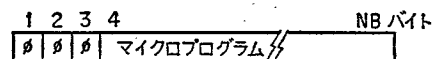
ISAD : 書きこみ先頭アドレス。

NB : 転送バイト数

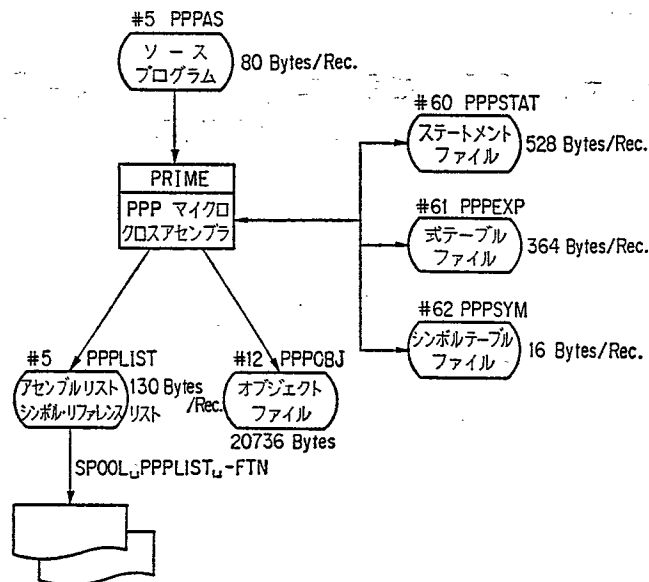
IFB : エラー情報

<使用上の注意>

- ①クロスアセンブラの出力は下図のように、先頭バイトが "0" で、あとマイクロプログラムになっている。



- ②転送バイト数及び格納エリア (IA) のサイズは $NB=3+(5 \times \text{マイクロプログラムステップ数})$



付図 1 クロスアセンブラのファイル構成

である。

③許される書きこみアドレス (ISAD) は 0~4095 である。

④エラー情報は DIP 内のモジュールへのパラメータ転送が正常に終了したか否かの情報を返す。もし、正常ならば IFB=0 である。

マイクロプログラムが正常に PPP へ転送されたかどうかは、マイクロプログラムをコールし実行してみないとわからない。

2.2 PPPEXM

〈機能〉

PPP のマイクロプログラムをコールし、PPP の制御をそのルーチンに引き渡す。

〈Calling Sequence〉

CALL PPPEXM (ISAD, IFB)

〈引数の説明〉

ISAD : コールするマイクロプログラムの先頭アドレス

IFB : エラー情報

