

## 動物体の認識・識別をめざした ビジョンプロセッサの構想と試作モデルの開発

久保田 浩明      福井 和広      石川 実      溝口 博

株式会社 東芝 総合研究所

あらまし      実時間で動いている物体の認識・理解を行うことができる次世代に向けてのビジョンプロセッサを開発するために、全画面の一樣処理と注目部分のみの局所処理を融合させるという考えを提案する。画像の内容に応じた画像分割により求められた複数の注視領域に対する並列処理を全画面一樣処理と併せて行い、従来型の画像処理では苦手とされてきた特徴抽出処理の高速化を狙う。本報告では、全画面一樣処理と注視領域処理の並列化を可能とする試作モデルを開発し、画像認識処理に有効であるかを検討する。

### Conception and Development of Pilot Model for Advanced Vision Processor to Recognize and Classify Moving Objects.

Hiroaki Kubota, Kazuhiro Fukui, Minoru Ishikawa and Hiroshi Mizoguchi

Research and Development Center, Toshiba Corporation

1 Komukai Toshiba, Saiwai-Ku, Kawasaki, 210 Japan

Abstract      This paper proposes an idea for image recognition and understanding that overall and local image processing should be executed in parallel. Using this idea, we aim at advanced vision processor that can recognize moving objects in an image and treat moving pictures.

The vision processor executes feature extraction for image recognition by several local processors in parallel. The vision processor has a characteristic special feature that each local processor processes only remarkable region decided by region division in consideration of image features and contents.

We developed a pilot model of the vision processor and realized speed-up of image recognition.

## 1. はじめに

近年、セキュリティに対する重要度の増大や、生産性の更なる向上への要求を背景として、監視や物体認識といった従来型の画像処理装置では苦手であった用途への画像処理応用が期待されるようになってきている。しかし、新しい分野への応用には、速度、機能、コストの面で、期待される水準に達していないのが現状である。

筆者らは、こうした状況を踏まえ、これまでに実用化されてきた画像処理装置の「次の」ハードウェアとして、実時間で動いている物体の認識・識別を目標としたビジョンプロセッサの開発を行っている。本報告では、認識の高度化を念頭においた従来型とは異なる画像処理の考え方について述べ、ついで、この考え方に基づき、その妥当性と実現可能性を確認する目的で試作したパイロットモデルの詳細について述べる。

## 2. ビジョンプロセッサの概要

### 2-1 特徴抽出部の分割処理

画像認識・画像理解を行う場合、一般的に3つの段階を踏まなければならない<sup>[1]</sup>。まず、入力された画像データに対して、雑音除去処理、画像強調処理、2値化等の前処理を行う。次に、処理目的にあった画像特徴を抽出する。最後に、抽出した特徴を用いて解析・判断等の認識処理を行う。

前処理に関しては、アルゴリズムの研究が古くからなされ、ハードウェア的にも局所並列処理、パイプライン処理等の技術により、専用ハードウェアが製品化され、現在ではほぼ完成された技術となっている<sup>[2]</sup>。特徴抽出処理に関しては、個々のアルゴリズムの研究がなされ、それらの処理を集めたLSIチップが開発されている<sup>[3]</sup>。しかし、次段の認識処理に必要な特徴抽出処理が体系づけられていないために、画像レベルから記号レベルへの情報の変換が確実に行われていない現状にある。最後の認識処理については、ニューロコンピュータやAIマシン等の登場により、画像処理の結果を高速に処理できると思われる。

画像認識・理解を実時間でを行うシステムを構築するには、画像特徴抽出のアルゴリズムを整理すると同時に、その処理に適した高速ハードウェアを組み合わせることが必要である。全ての処理を個別に専用ハードウェア化する方式も考えられるが、これを行うのは規

模、コスト等の問題からいって不可能である。筆者らは、特徴抽出処理を並列に行い、個々で行われる処理は柔軟性の大きいソフトウェアに任せ、複数の汎用プロセッサによる分割処理を採用した。ここで問題となるのが、処理の分割方法およびプロセッサ間のデータ通信方法である。

処理の分割方法は従来、図2-1に代表される、画像の内容に依存しない画像分割方式が採られてきた。この方法では一つのプロセッサのみで対象を定義できないため、特徴量計測等の処理をするには、隣接した領域を受け持つプロセッサから必要な情報を受け取る手間がかかる。これを防ぐために、各プロセッサに割り当てる分割領域に画像としての意味を持たせた分割を行い、処理の途中においてのプロセッサ間の相互通信を最小限にする方法を探る<sup>[4]</sup>。

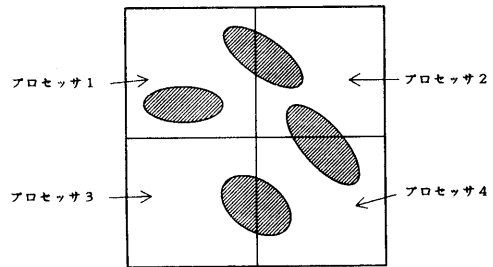


図2-1 従来の画像分割方法

### 2-2 特徴抽出部の並列処理形態

逐次処理向きである特徴抽出部においては、従来の画素並列技術の効果が現われなため、複数プロセッサによる分割処理を行い、並列処理化と処理範囲の限定により、高速性を実現する。その際、画像内容に依存する次のような並列処理形態を採る。

#### a. 四角形領域ごとの並列処理

各プロセッサに割り当てる領域の分割を単純な四角形領域単位とする方法である。列単位、点単位の分割はこれに含まれる。図2-2のように画像中の物体に対して、位置に依存した分布や個数、大きさ、形状などを特徴計測する場合にこの方法が有効である。分割のための前処理も簡単である。

#### b. ラベルごとの並列処理

前処理部において2値化された画像の連結成分ごとに領域分割する方法である。各プロセッサには連結成分を含む長方形領域を割り当てる。図2-3のように割り当てを行う。1つのプロセッサに複数の連

結成分を割当てることもありうる。この並列処理は連結成分単位で計測される特徴抽出に有効である。

c. 注視点、注視領域ごとの並列処理

予め与えられた簡単な図形特徴やモデルを参照して、認識対象となる候補領域を探索し、個々を各プロセッサに割当てる方法である。図2-4のようにある程度の幅と高さを持った領域(丸い形をした部分)を簡単なマッチング処理等により探索し、ピックアップされた領域のみを各プロセッサに割当てる。指定された形の物体あるいは図形の認識、理解に有効である。

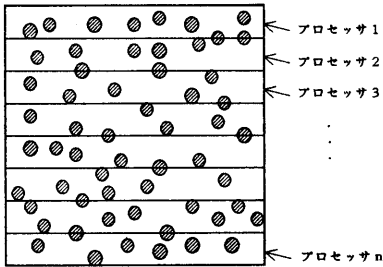


図2-2 四角形領域ごとの並列処理

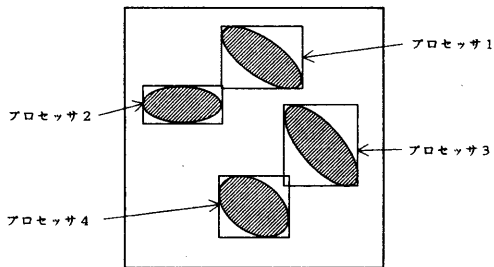


図2-3 連結成分ごとの並列処理

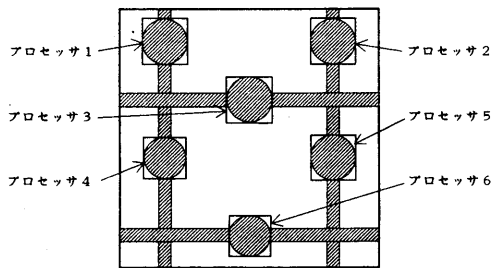


図2-4 注視点、注視領域ごとの並列処理

d. 解像度等を変えた複数画像の同時処理

入力画像に対して拡大、縮小、アフィン変換等を施して得られた解像度、スケールの異なる複数の画像を同時に処理する方法である。例えば、図2-5のようにプロセッサ1では画像全体を粗く処理し、他のプロセッサでは画像中のある部分に絞って細かい処理を行う。この並列処理は一連の認識処理の手順の前半部分、後半部分を分割して複数のプロセッサに振り分ける場合に有効である。また、一枚の画像から大きさ、あるいは、形状の異なる対象を認識する場合にもこの処理方法を使うことができる。

e. 複数オペレーションの同時処理

多数の特徴抽出オペレーションの中で同時に行うことのできるものを複数のプロセッサで分担して同時に処理する方法である。図2-6のように画像中の同じ領域を複数のプロセッサに割当て、各プロセッサは異なる処理を行う。

上記の並列処理形態のうち2つ以上の形態が同時に全プロセッサ内に存在することを許し、より効率的な特徴抽出を可能にする。

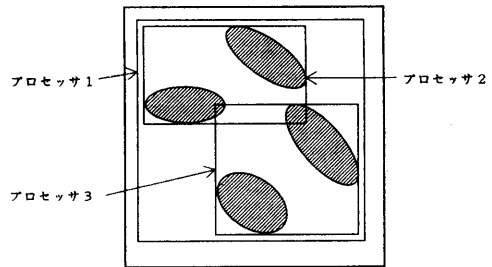


図2-5 解像度を変えた複数画像の同時処理

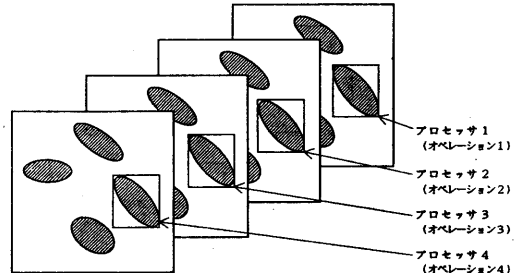


図2-6 複数オペレーションの同時処理

### 2-3 ローカルメモリと画像バス

前節で述べた並列処理を効率的に行うため、各注視領域処理モジュールが個々にローカルメモリを持ち、画像バスを流れてくる画像データのうち、必要な領域のみをメモリに取り込むことができる。画像データの取り込みは画像バス中のクロックおよび同期信号により動作し、処理モジュールのプロセッサの負担なしに行える。注視領域の設定はモジュール内から簡単な操作で行うことができ、瞬時の変更が可能である。このように、ローカルメモリの画像データは全画像に対するウィンドウのように扱うことができる。

画像バスは、注視領域処理モジュールへの画像入力用の入力画像バスと画像出力用の出力画像バスに分け、実時間処理に対応できるよう高速転送する。

### 3. ビジョンプロセッサの構成と機能

このビジョンプロセッサ（VP）による画像処理は、図3-1に示すように、前処理部を受け持つ画素レベル処理装置、特徴抽出処理を受け持つ複数の注視領域処理モジュール（ローカル処理モジュール）に大きく2つに分けて行う。図中、画像入力モジュールでは、カラーカメラ2台までのビデオ入力、画素レベル処理装置との画像データ通信、注視領域処理モジュールへの画像バス出力を行う。画像出力モジュールでは、カラーモニターへの処理結果の表示を行う。各モジュール、処理装置の全体的制御は全てホストコンピュータで行う。

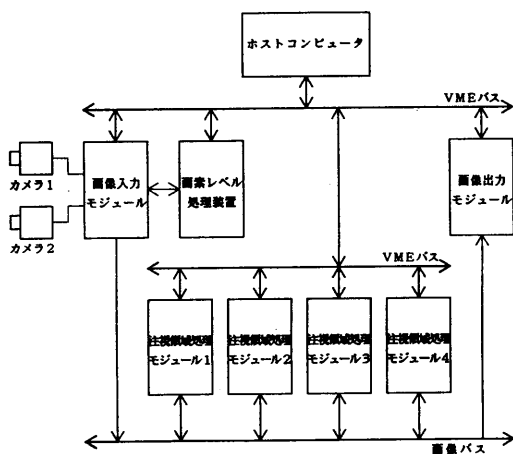


図3-1 ビジョンプロセッサのハードウェア構成

この章では、前章で述べたビジョンプロセッサの構想に基づいて開発された試作装置について、各モジュールごとに説明する。

### 3-1 ホストコンピュータ

ホストコンピュータとして、本システムは東芝のワークステーションAS-3160Mを用いる。画像認識の実行時には、各モジュールのコントロールを行う。また、ソフトウェアの開発装置である。

### 3-2 画素レベル処理装置

画素レベル処理装置として、本システムは東芝の汎用画像処理装置TOSPIX-iを用いる<sup>(5)</sup>。この装置では、主に画像処理の前処理を行う。具体的には、雑音除去処理、画像強調処理、2値化処理、ラベリング処理等を組み合わせて、注視領域を決定する。512画素×512画素の画像1枚の処理を、約33ミリ秒で行うことができる。

### 3-3 注視領域処理モジュール

様々な並列処理形態を実現するために、注視領域処理モジュールは、図3-2に示すように、ローカルプロセッサおよびウィンドウコントローラ、ウィンドウメモリ、アドレスデコーダ、バスI/Fにより構成される。

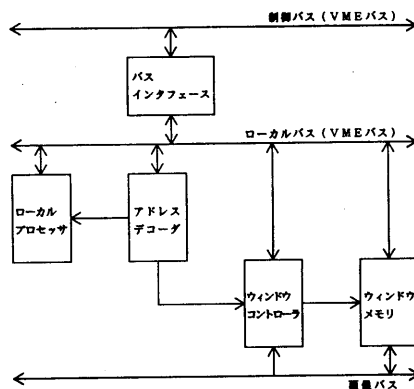


図3-2 注視領域処理モジュールのハードウェア構成

ローカルプロセッサにはモトローラ68020(20MHz)を用いている。このプロセッサで実行する画像処理のプログラムは、ホストコンピュータで開発、コンパイルを行う。プログラムの転送は、バスI/Fを通して行われる。

ウィンドウコントローラは、指定された注視領域を画像バス中のクロック信号によりウィンドウとし

て発生させ、ついでウィンドウのメモリ上のアドレスを発生させる。あらかじめ入力画像バスセレクトにおいて選択された入力用の画像バスから画像データを読み込み、ウィンドウメモリに書き込む。

ウィンドウメモリは64KBであり、最大256画素×256画素の画像を確保できる。画像バスからの画像取り込みに関しては画像バスのデータ速度で行える。したがって、1画面最大1/60秒で取り込みが終了する。

取り込んだウィンドウメモリ中の画像データは、常に出力画像バスに出力することが可能であり、随時モニタすることができる。このため、複数の処理モジュールで画像データが衝突しないようにアービタを備えている。

アドレス発生器には、間引き取り込み、間引き出力の機能、および、画像データを2次元アドレスで書き込む機能を設けている。

ローカルプロセッサは、ウィンドウメモリに画像データが書き込まれると、メモリ上の画像データに対して特徴抽出処理を開始する。プロセッサの処理は限定された小領域だけを行う。画像以外の処理結果はバスI/Fを通してホストコンピュータへ送られ、認識処理等が施される。

画像バスに対する読み書きは、プロセッサの負担なしに行われ、また、プロセッサの処理範囲は小領域に限定されているので、与えられた処理を短時間で終了することが可能である。

バスI/Fはホストコンピュータとの通信を受け持つ。

### 3-4 画像入力モジュール

図3-3に画像入力モジュールの構成を示す。このモジュールは、TVカメラ、画素レベル処理装置、ホストコンピュータから入力された画像を画像バスに出力するものである。ステレオ視およびカラー動画像に対応した構成となっている。次のような機能をもっている。

ステレオ視に対応できるように、カラーTVカメラ(R, G, B信号)2台まで接続できるようにA/D変換器を備えている。各8ビットのデジタル信号に変換する。

デジタル化された1組のR, G, B信号は、まず、各4ビットにデータ圧縮され、計12ビットの信号をカラーlookupアップテーブル(LUT)のアドレスとしてデータを参照し、指定される色の抽出を行

う。このカラーLUTはステレオ視対応のため、2組用意している。この機能により実時間レベルでのカラー識別が可能である。

カメラから入力されるインタレースの信号をノンインタレースに変換するため、画像メモリは2枚1組の構成で、R, G, B信号用の3組用意されている。1組の画像メモリの動作は、書き込みを行っているあいだに、もう一方の画像メモリから画像バスに出力する。これを交互に繰り返すことにより、画像バスへの連続的な動画像出力を可能にしている。書き込み/読み出しの繰り返しを止め、連続画像中の一画面を保持するフリーズ機能もあり、1枚の画像を固定して画像バスに流すことが可能である。画像データの出力速度は60nS/画素である。この速度は、TVカメラ1フレームの入力時間に対して2フレームの画像を転送できるものである。この転送時間の短縮により、注視領域処理モジュールが画像データを取り込む際のオーバーヘッドを削減している。

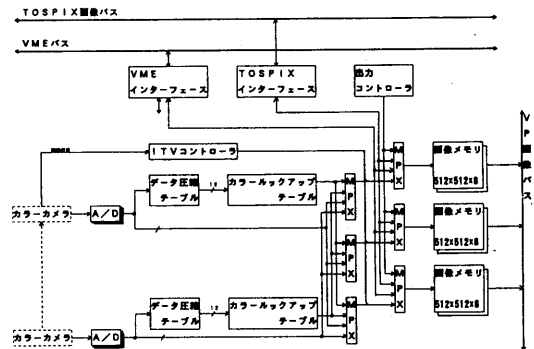


図3-3 画像入力モジュールのハードウェア構成

### 3-5 画像出力モジュール

画像バスに流れているデータを画像モニタに表示できる信号に変換するモジュールである。このモジュールの構成を図3-4に示す。R, G, Bの3本の入力画像バスに対応して、マルチプレキサ(MPX)および1対のラインバッファを3組備え、その他D/A変換器、タイミングコントロール、VMEインターフェースで構成される。

MPXでは、入力画像バスの3本、出力画像バスの3本、計6本から画像データを選択する。この選択は入力バスよりも出力バスの信号を優先し、さらにVMEインターフェースからの指示によりR, G, Bから1つを選択する。

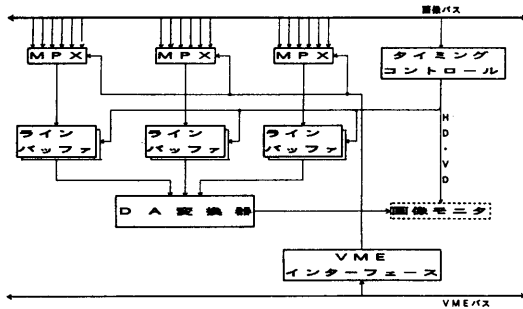


図3-4 画像出力モジュールのハードウェア構成

ラインバッファは、MPXによって選択された信号を交互に書き込み／読み出しを行い、リアルタイム出力を可能にする。

D/A変換器では、ラインバッファから出力されるデジタル信号をアナログ信号に変換し、画像モニタで表示可能な信号にする。これらのタイミングの制御および画像モニタの制御信号の生成は、タイミングコントロールで行う。

#### 4. ビジョンシステムとしての環境の開発

ビジョンプロセッサの試作装置開発にあたり、ビジョンシステムとしてのソフトウェア環境の開発、整備は重要な課題である。ハードウェアを効率よく動かし、最大限の機能を発揮するために次のようなソフトウェアを開発した。

##### 4-1 ライブラリ関数

ハードウェアを直接動かすデバイスドライバは、各モジュール、処理装置ごとにライブラリ管理している。これらは後述するユーザプログラムの開発において、ライブラリ関数として使用できる。

##### 4-2 プログラムの開発

ビジョンプロセッサを動かすには、ホストで実行するメインプログラムと注視領域処理モジュールで実行するローカルプログラムが必要である。いずれもホストで開発を行う。ローカルプログラムは、実行前にダウンロードしておく必要がある。

メインプログラムでは、画素レベル処理装置の制御、各注視領域処理モジュールの制御、処理結果による認識処理を行う。プログラムはC言語で開発を行う。プログラム中でライブラリ関数として、各デバイスドライバを使用できる。

ローカルプログラムでは、指定された領域の画像

処理、および、一部の認識処理、記号処理を行う。プログラムはC言語で開発を行い、注視領域処理モジュール用のライブラリ関数を使用できる。

プログラムの実行は各ローカルのプログラムがロードされ、実行された状態で、メインプログラムを動かすことにより行われる。

#### 5. ビジョンプロセッサの動作実験

##### a. 注視領域決定実験

本システムにおいて、注視領域の決定は、画素レベル処理装置あるいは注視領域処理モジュールで行う。注視領域処理モジュールで注視領域を決定するには、間引き取り込みにより全画面をウィンドウメモリに取り込んで画像全体を処理し、注目する部分を抽出する。本実験では、画素レベル処理装置を用いて注視領域を決定する例について示す。

まず、入力された画像は、TOSPIX-iに転送される。フィルタリング、2値化、ラベリング、各連結成分の外接長方形および面積の計測が順に行われる。TOSPIX-iでは、この処理をパイプライン処理で約33ミリ秒で行える。各成分の外接長方形と面積がホストコンピュータに転送され、注視領域の決定が終了する。

写真5-1は、決定された注視領域の様子をモニター表示した例である。このような注視領域の決定は、画像入力から約40ミリ秒で行える。

TOSPIX内での注視領域の決定処理の手順は、常に上記のとおり行なわれるわけではなく、領域強調、エッジ強調、テンプレートマッチング等の処理を組み合わせて行なわれる。

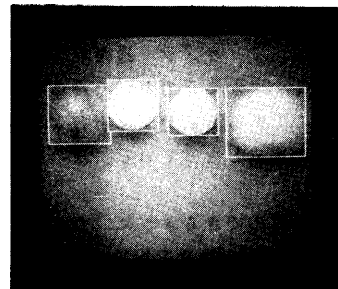


写真5-1 決定された複数の注視領域

##### b. 動物体追跡実験

前述の注視領域決定処理を踏まえ、各注視領域処理モジュールは与えられた領域（写真5-1の白枠内）についてのみ処理する。

決定された注視領域の情報がホストコンピュータから伝えられると、その情報をレジスタにセットした後、画像データの取り込みを行う。取り込みには、最大16ミリ秒要する。

特徴抽出処理はこの画像メモリ内のデータに対して行う。ここでは、連結成分の大きさの確認、および、重心の計測を行い、物体の位置を2次元画像中に求める。処理にかかる時間は注視領域の大きさによって多少異なるが、この実験の場合、15~20ミリ秒で終了した。

この実験により、前記の注視領域決定処理と合わせて、カラーTVカメラから入力された画像に対し、画像中に存在する物体を探索し、2次元座標を計測するという動物体追跡が可能であることが確認された。現在のシステムでは、最大4つまでの物体を同時に追跡することができ、約80ミリ秒ごとに2次元座標が得られる。

#### c. カラー識別実験

カラーLUTを用いて、カラーカメラ入力時における物体のカラー識別を行った。このカラー識別は、3次元座標軸を考え、この軸上に各々R、G、Bの成分を配置するようなデータをLUTに書き込む。LUTに入力される信号はR、G、B各4ビット、出力8ビットであり、最大256色の識別を行える。

今回、物体として風船を例にとり、風船の色の識別を行ったが、風船の色むら、証明の方向、光源そのもの色、反射等を考慮に入れて、同時に7色までの識別が確認された。写真5-2は4色の風船をカラーTVカメラで撮影したところである。この画像をカラーLUTを通すことにより写真5-3のように4つの風船の色を識別できる。

このカラーLUTの処理をソフトウェアで行わせるとすると、プロセッサにかなりの負担をかけることになる。AS-3160で処理したところ、約5秒を要した。VPでは、2チャンネル同時のカラー識別を、約33ミリ秒で行える。

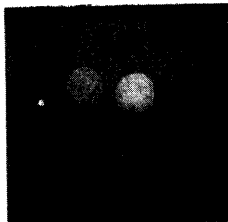


写真5-2 TVカメラ入力画像

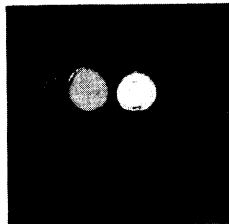


写真5-3 カラー識別結果

#### d. ステレオ視による3次元位置計測実験

この実験では、画像入力モジュールが2台のカラーカメラをサポートしていることを利用し、ステレオ視による動物体の3次元位置を計測する。

カメラからの入力はカラーLUTを通し、カラー識別を行う。これにより2台のカメラで取り込んだ画像間での物体の対応を正確に行っておく。

まず、カメラ1からの画像をTOSPIX-iに転送する。注視領域が決定されると、次にカメラ2からの画像をTOSPIX-iに転送する。これと同時にカメラ1からの画像の注視領域を処理モジュールに送り、物体の2次元座標値を計測する。その後、カメラ2からの画像の注視領域の決定が終了すると、同様に処理モジュールにおいて、2次元座標値を計測する。この時、一つの処理モジュールにおいて、入力された2枚の画像で同一の物体を処理できるように、ホストで制御する。最後に、1組の2次元座標により三角測量を行い、物体の3次元位置を算出する。

以上の手順により、ビジョンプロセッサでの物体の3次元位置の計測が終了する。処理時間は1回の3次元計測で約180ミリ秒であった。

この実験では1回の物体位置計測に2度TOSPIX-iを起動して、TOSPIX-VP間で2往復の画像転送を行っている。この間、処理モジュールは処理を停止しているため、並列性に無駄があった。そこで、TOSPIXでの処理を1台のカメラからの画像のみとして、もう1台のカメラからの画像は処理モジュールに直接転送し、画像全体を大まかに処理して、注視領域の決定および重心計算を行った。これにより、並列性が高められ、1回の3次元計測で約100ミリ秒と全体の速度をあげることができた。秒10枚程度の3次元計測が可能となり、リアルタイム制御が可能な速度が得られた。

この処理を従来の汎用画像処理装置のみを使用しに行った場合、画像中の物体1個を追跡するのに、300ミリ秒を要した。ビジョンプロセッサにおいては、2個の物体を同時に追跡することができ、5倍以上の速度を確認できた。

## 6. 考察

前章において、試作したパイロットモデルの動作実験について報告した。その実験の中で気付いた点について、今後の課題を含めて考察する。

#### a. 画像入力にかかる時間

VPでは、カメラ入力の後、画像メモリに一画面を取り込んだ後、各モジュール、処理装置に画像転送している。カメラが30Hzインタレースで画像を取り込んでいるのを、画像転送する際に60Hzノンインタレースに変換しているためである。

ノンインタレースで画像を入力できるカメラを組み込み、入力におけるスキャンで、処理モジュールにデータを取り込める設計が必要である。

#### b. 画像転送にかかる時間

処理モジュールにおいては、画像バスからのデータ取り込みが完了してから、処理を開始する。画像の転送には、最大30ミリ秒を要する。全体の認識処理が100ミリ秒以内に終了していると仮定すると、処理を遅くする原因として、かなり大きなウエイトをしめているといえる。

これは、TOSPIX-iがVPの画像バスと直結されていないために、この間で改めて画像転送しなければならないからである。画素レベル処理を各処理モジュールに代用させる、あるいは、ビジョンプロセッサの画像バス仕様を取り入れた専用の画素レベル処理装置を組み込む等の改善策がある。

#### c. モジュール間のデータ通信

ひとつのローカル処理モジュールの処理結果を別の処理モジュールで使うときは、一度ホストにデータを渡して、再度データを受け取る必要があった。試作装置では、プロセッサ数が4であったので、速度への影響は無視できたが、今後、プロセッサの数を増やした場合、通信手順等の問題が発生してくると思われる。処理モジュールから処理モジュールへの画像転送についても考えなければならない。

#### d. ローカル処理モジュールの機能強化

本報告では、処理モジュールは注視領域内のデータに対して、特徴抽出処理を行うのが原則であった。しかし、より複雑な認識処理を行う場合、ローカルにおいて全画面を対象とした画素レベル処理を受け持つことがありうる。ローカル処理モジュール上に局所並列処理機能を付加する、プロセッサ自体の性能を向上させる等の改良が必要である。

#### e. ソフトウェア

ハードウェアを効率よく動かすためのソフトウェアの開発も重要な課題となる。とくに、画像特有の並列処理を実現するためのソフトウェアの開発、物

体認識・理解に必要なアルゴリズムの研究をはじめ、残される課題は多い。

前章の3次元計測の例を、従来の汎用画像処理装置を使用し、画像処理装置で不可能な処理は、全てホストコンピュータによるソフトウェアで行った場合に比べて、試作モデルでは約100倍の速度で処理できたことになる。今後、前記の課題を解決することにより、実時間で画像認識を行えるビジョンプロセッサを開発できると考える。

#### 7. おわりに

本報告では、画像の認識・理解を目的とするビジョンプロセッサの構想と、その試作モデルについて述べた。全画面一様処理と注目部分に対する局所的な処理の融合という考え方を提案し、複数の注目部分に対する処理が画像の内容に応じ種々の形で並列化できるというアイデアを示した。そして、この考え方に基づくビジョンプロセッサの構成案とその実現法について述べ、それに基づいてパイロットモデルを試作して実現可能性を示した。

本文中では、この試作したビジョンプロセッサを用いて行った動作実験について簡単に触れた。さらに、この試作ハードウェアの実時間性を実証するために、多関節ダイレクトドライブロボットと組合せ、落ちてくる風船をたたき続ける“お手玉”動作をロボットに行わせる実験を行った。

今後は、画像特有の並列処理を簡便に記述できるようなソフトウェアシステムの構築と、それを用いての動画像処理、動物体認識の研究、応用ソフトウェア開発を進めていく必要がある。

#### 参考文献

- [1]長尾真, “画像認識論”, コロナ社, 1983.
- [2]坂上, 木戸出, “イメージプロセッサの最近の動向” 信学誌, Vol.67, No.1, Jan., 1984.
- [3]奥山, 小林, 武長, 金崎, 有賀, 浅田, “ISP-II用周辺LSI(特徴抽出プロセッサ)”, 昭62信学全大, 1246, March, 1987.
- [4]久保田, 溝口, 麻田, “画像認識システムに向けてのアプローチ”, 第37回情処全大, pp.1500, Sep. 1988.
- [5]押田, 所, “高速画像処理プロセッサTOSPIX-iの特長とその応用例”, 映像情報, pp.37-42, Dec., 1987.