

ビジュアルフィードバックのための
超並列・超高速ビジョンシステムの研究

中 坊 嘉 実



ビジュアルフィードバックのための
超並列・超高速ビジョンシステムの研究

中 坊 嘉 宏

目次

1 序論	1
1.1 はじめに	1
1.2 ビジュアルフィードバック	2
1.3 ビジョンシステム	4
1.4 アクティブビジョン	4
1.5 研究の目的	5
1.6 本論文の構成	6
2 ビジュアルフィードバックのモデル	8
2.1 はじめに	8
2.2 カメラモデル	9
2.3 マニピュレータの機構モデル	9
2.4 ビジュアルサーボ	10
2.4.1 画像特微量	10
2.4.2 ビジュアルサーボの構成	11
2.4.3 システムのダイナミクス	12
2.4.4 サーボレート	12
2.5 アクティブビジョン	13
2.5.1 アクティブビジョンの有用性	13
2.5.2 超高速アクティブビジョン	14
2.5.3 アクティブビジョンによるセンシングのモデル	15
2.5.4 トラッキング可能性の評価	15
2.5.5 未知性を含んだ場合のトラッキング可能性	16
3 ビジョンシステムの構成	17
3.1 はじめに	17
3.2 ビジョンシステムのアーキテクチャ	18
3.2.1 超並列・超高速画像処理	19
3.2.2 画像特微量抽出処理	19

3.2.2.1 画像処理システムと情報量	20
3.2.2.2 1チップによる実現	24
3.2.2.3 完全並列接続による実現	24
3.2.2.4 列並列接続による実現	25
3.2.3 システムコントローラのアーキテクチャ	25
3.2.4 アクティブビジョンのアーキテクチャ	27
3.3 完全並列による超高速ビジョンシステム (SPE-256) の実現	27
3.3.1 概要	27
3.3.2 システム構成	27
3.3.2.1 超並列・超高速画像処理部	28
3.3.2.2 システムコントロール	29
3.3.2.3 アクティブビジョンシステム (AVS-I)	30
3.3.3 実験	31
3.3.3.1 アクチュエータの周波数応答特性	31
3.3.3.2 1ms ターゲットトラッキング	32
3.3.4 考察	34
3.3.4.1 ビジュアルフィードバックの高速性と応用アプリケーション開発	34
3.3.4.2 SPE-256 システムの問題点	36
3.4 列並列による超高速ビジョンシステム (CPV) の実現	37
3.4.1 概要	37
3.4.2 システム構成	38
3.4.2.1 列並列・超高速画像処理部	38
3.4.2.2 コントローラ	42
3.4.2.3 アクティブビジョンシステム (AVS-II)	44
3.4.3 実験	46
3.4.3.1 PD モジュールの感度評価	46
3.4.3.2 多階調フィルタリング画像処理	46
3.4.3.3 ヒストグラムの取得とスレッショルドデインジ	50
3.4.3.4 テンプレートマッチング	52
3.4.3.5 テンプレートマッチングによるトラッキング	54
3.4.3.6 1ms ターゲットトラッキング	57
3.4.3.7 ビジュアルフィードバックの周波数応答解析	59
3.4.4 考察	59
4 ビジュアルインピーダンス	62
4.1 はじめに	62
4.2 ビジュアルインピーダンス	63

4.2.1	概要	63
4.2.2	仮想接触の導入	64
4.2.3	コンプライアンス特性の実現	65
4.2.4	パターンに対するビジュアルインピーダンス	66
4.2.5	マニピュレータへの関節角トルクの計算	67
4.2.6	従来の制御法との統合	68
4.3	実験	69
4.3.1	実験システムの構成	69
4.3.2	仮想接触点抽出のアルゴリズム	71
4.3.3	衝突回避の実験	73
4.3.4	はめあいの実験	74
4.4	考察	74
5	結論	78
	謝辞	80

第 1 章

序論

1.1 はじめに

今日ロボットが社会において果たすべき役割は大きい。特にロボットの活動範囲は、ごく近い将来、我々の日常生活の場へと拡大して行くことが予想される。

これまでのロボットの活動範囲は工場などの生産現場、すなわちロボットのために特別に整備され、人間の立ち入りが禁止された限られた場所が中心であった。しかしより広い分野で、人間の労働の手助や肩代わりをさせたいという要求から、日常生活の場で人間と共存しながら自律的に活動するロボットの登場が期待されている。

一方インターネットに象徴される地球規模の情報ネットワーク、及びパーソナルコンピュータ (PC) の急速な普及と高性能化により、特に専門的な知識や技術を持たない一般の人々が、これらの高度な情報システムを気軽に利用し、その恩恵を受けることが可能になりつつある。このような流れの中で、統合と拡大を続ける情報世界と、現実の世界とを直接結び付ける媒体として、ロボット技術を位置づけることもできる。近年注目を集めているバーチャルリアリティやヒューマノイドロボットの研究に見られるように、実環境のもとで活動するロボットへの期待はますます高まりつつあると言える。

ところがロボットにとって、これまでの整備された工場環境とは異なり、実環境における環境の変動とモデルの未知性が大きな問題となる。すなわちロボットが活動する環境や対象の状態が常に変動し、同時にロボットが環境を認識する際に必要となるモデルについて、あらかじめ正確にはわからない場合が多くなる。さらにロボットの活動に人間が関わってくる場合は、安全性の問題など、これまでのロボット技術とは異なる新しい技術的課題が発生する。よってこのような未知性、不確実性に柔軟に対応し、人間とのインタラクションで生じる危険を事前に察知して回避するような自律的でロバストなロボットシステムを構築することが求められている。

このような問題を解決する方法の一つとして視覚情報の利用が挙げられる。視覚を用いることで対象とロボットの相対的な関係を一度に、かつ非接触で得ることができるようになる。このような観点から、視覚情報の利用は、今後のロボット技術において大きな役割を担うようになると考えられる。

これまでも視覚の利点は古くから認識され、視覚によるロボットの制御が盛んに研究されてきた。これらはビジュアルフィードバックと呼ばれている。特に近年では動的に変化する環境を高速に認識するリアルタイムビジョンや、これに対応して高速なサーボ制御を目指したビジュアルサーボの

研究が行われ、実際のビジョンシステムの開発とロボット制御への応用が試みられている、また受動的に得られる画像情報を利用するだけでなく、カメラを能動的に動かすことで、固定されたカメラ以上の情報を得るアクティブビジョンの研究も注目されている。

しかしこれら従来の研究で問題となるのが、システムのダイナミクスの問題である。実際にビジュアルフィードバックを用いてロボットやカメラを制御する閉ループシステムを構築しようとすると、ビジョンシステムの動作速度が不十分となって、動的な制御の実現は困難であった。これはあとで述べるように、ロボットやアクティブビジョンの制御に対してビジョンシステムの最適な設計がなされていないためであり、具体的言えば、従来用いられてきた CCD カメラのビデオ信号による画像伝送の部分に問題があった。

これに対して石川らは超並列・超高速ビジョンチップ [1] の概念を提案し、実際にデバイスの開発 [2] と応用研究 [3] を行っている、ただし現状の半導体技術では集積化に限界があり、実際のロボットの制御に対して十分な、ビジョンチップシステムの実現には到っていない。

このような背景のもとで、本研究では動的に変化し未知の環境に対応するためのロボットの視覚として、従来のビジョンシステムの限界を打ち破る超並列・超高速ビジョンシステムを実現する、そのために、まずビジュアルフィードバックの研究で従来見落とされてきたダイナミクスの問題について考察し、この考察にもとづいてあるべきビジョンシステムの姿を描く。さらに、これをもとにしたトップダウンの設計を行い、実際にシステムを構築して評価を行う。また視覚によってダイナミクスを直接制御する新しいビジュアルフィードバックの方法を提案し、実際に構築したシステムを用いた実験によって、その有効性を示す。

1.2 ビジュアルフィードバック

ロボットが未知の環境や変化する環境に対応するには、視覚情報の利用が有効になると考えられる。すなわち視覚センサにより対象とロボット、及び周囲の環境の情報を画像として取得し、ここから画像処理によって有用な情報を抽出してロボットの制御を行う方法である。これらはビジュアルフィードバックと呼ばれている。

このようなビジュアルフィードバック制御の重要性は古くから認識され、様々な研究がなされてきた。その最も初期のもとして Shirai ら [4] の研究が挙げられる。この研究ではブロックを対象として画像による線画の抽出を行い、マニピュレータの位置姿勢誤差を修正して物体を操作するビジュアルフィードバック制御を実現している。このような研究の初期の段階では画像処理とマニピュレータの運動制御について、どちらについてもハードウェアの性能が充分とは言えなかった。そのため、その後の研究では動的な問題としての制御と、静的な問題としての認識がそれぞれ分けて考えられるようになった。特に得られた画像をどのように解釈し、有用な情報を抽出するかという人工知能的な対象認識の問題は、現在も主にコンピュータビジョンの分野で研究がなされている。

その後計算機能力の向上から 1 枚の画像をより高速に処理することができるようになり、画像を取り込んでロボットが行動を起こす一連のサイクルが十分短い時間、いわゆるリアルタイムで実行できるようになってきた。この結果、画像処理、軌道計画、アクチュエータ制御それぞれを全て同時並

列に動かす動的なビジュアルフィードバック制御が実現されるようになった [5]、すなわち「見てから動く」ことを繰り返す方法から「見ながら動く」方法への変化である [6]。このようなリアルタイム性に重きを置いた動的なビジュアルフィードバック制御は、これまでのものと区別してビジュアルサーボと呼ばれている [7]。

ビジュアルサーボでは、視覚にもとづくロボットの認識と行動が高速な低次のサーボとして実現されており、環境変動やタスク状況の変化に対する反射的な応答を特徴としている。言い換えれば、コンピュータビジョンのように対象の認識を最終目的として、ある程度時間をかけて画像を解析するのと対照的に、タスク遂行に特化した最低限の特徴量を選んでこれをできる限り高速に画像から抽出し、ロボットの動作に反映させる方法と言える。これによって実現されるフィードバックシステムでは、視覚から抽出した対象とロボットの相対的な関係を誤差情報として、タスクレベルでの閉ループシステム (ECL: Endpoint Closed-Loop system) を実現することが可能となり、モデル化誤差に強いロバストなシステムを実現することができる。このようなビジュアルサーボについての一般的な解説は、例えば Hutchinson ら [8] によりまとめられている。

しかしその一方で、今日実現されているビジュアルサーボシステムにおいて、十分な高速性が達成されているかと言えば、そうではない。ビジュアルフィードバック以外のロボット制御の分野では、マニピュレータの軌道制御における分解加速度法 [9] やコンプライアンスを実現する力制御 [10]、あるいはこれらを混合した仮想内部モデルにもとづく制御 [11] など、より高度で動的な、様々な制御法が提案され、実用化されてきた。その結果センサフィードバックやサーボシステムで制御する対象、あるいは想定するモデルで扱う周波数帯域は、今日のビジュアルサーボで実現されるサーボレートと比較してもさらに高い帯域 [12] が必要になっていると言える。

このようにビジュアルサーボで実現可能なサーボレートとマニピュレータの制御に必要なサーボレートとの差についての影響が指摘されており [13]、この差を埋めるために内挿 [14] や、カルマンフィルタ [15-17]、あるいはその他のモデル [18, 19] にもとづく予測を用いたシステムのパフォーマンスの改善が多数試みられている。しかしこれらの研究は、最初からシステム全体のダイナミクスを考慮して制御系を設計しているわけではない。

これに対して橋本らは、まず非線型制御アプローチによるダイナミクスの定式化 [20] を行い、ついで非線型オブザーバを用いたシステムの状態推定 [21] によりサーボ系におけるダイナミクスの問題に一定の解決を与えている。しかしこの場合でも、対象の状態が精度よく推定できるのは、対象のモデルが充分正確に分かっている場合に限られる。結局ビジョンシステムの高速化なくしては、ビジュアルサーボの高速化の試みも十分な効果をあげられないと言える。これについては持説で述べる。

一方、このようにビジュアルサーボのダイナミクスについて問題を抱えていることで、制御アルゴリズムの面で、ビジュアルフィードバックと他のセンサフィードバックとの融合についても不十分となっている。すなわち視覚から得られる情報と他のセンサから得られる情報について、動作計画 [22] や軌道計画 [23] のレベルで融合した例についてはいくつか見られるものの、これをサーボレベルで融合して動的な制御を実現した例はほとんどみられなかった。

1.3 ビジョンシステム

このようなビジュアルサーボのダイナミクス問題は、主にビジョンシステムのハードウェア性能が不十分であるために生じている。すなわち現状のビジョンシステムではビジュアルサーボに必要なサーボレートが実現できていないことに問題がある。

前にも述べたように、ビジュアルサーボが実現されるようになった鍵とも言える、いわゆるリアルタイムビジョン [24] の実現については、今日の計算機と半導体技術の発展に負うところが大きい。しかしここで問題となるのは、画像圧縮や画像処理に特化したシステムが必ずしもロボットのビジュアルフィードバックシステムとして最適であるとは限らないことである。

現在のロボット研究の分野で用いられているシステムをいくつか見た場合 [24-29]、これらに共通するのは、画像センサから処理システムへ画像を伝送するために、ビデオ信号を用いていることである。これらのシステムではビデオ信号の規格 (NTSC 規格) であるフレームレート (30Hz) あるいはフィールドレート (60Hz) がシステムの画像処理レートの上限となっており、一方で空間解像度については NTSC 規格の解像度 (512×480) に合わせてフレームメモリや処理システムが構成されている。

ところがこのビデオ信号の規格は、いわゆるテレビ、すなわち人間の視覚特性に合わせて決められたものであり、ロボット制御に対して最適なものとは言えない。先に挙げた位置や力のフィードバックの例では、ロボットの動的な制御を実現するためには、一般に 100Hz ~ 1KHz 程度が必要だとされており [12]、実際に研究用として普及しているロボットシステムでも例えばコントローラのオープン化 [30, 31] や、最近では安価な PC を用いた場合でも [32]、ロボットセンサやソフトウェアサーボ処理システムとして実時間の条件を満たすようなシステムの構築が可能である。一方現在開発されている、ビデオ信号を用いたシステムではこの条件を満たすことができない。

これは言い換えれば従来のビジョンシステムの設計が、特に視覚センサの部分でハードウェアの制約にもとづくボトムアップの手法によっているからだと考えられる。実際、先に挙げたビジョンシステムにおいても、1 フレームあたりに多数のマッチング処理や特徴量抽出処理が可能 [25, 26, 29] となっており、プロセッサの処理能力が不足しているわけではない。

これに対して視覚センサまで含めたシステム全体としてトップダウンで最適な設計を行うことにより、ロボットの動的な制御に必要な高速なサーボレートを実現することが可能になると考えられる。

1.4 アクティブビジョン

一方、空間解像度についても同様な指摘ができる。これまでも、カメラを能動的に動かすことで、固定されたカメラ以上の情報を得るアクティブビジョン [33-35] が提案されている。アクティブビジョンの詳細については様々な方法、概念が提案されているが、おおまかにいえば、ビジョンの能動性を利用して、タスクに対して最適な処理とシステムを実現することが目標となっている [33]。しかしここでも、さらに一歩踏み込んで、アクティブビジョンの利用を前提としてビジョンシステムを設計し直すトップダウンのアプローチは見られない。

アクティブビジョンの代表的な方法である注視制御 [34] を例に挙げれば、必要な領域だけを注視することで、ビジョンシステムに要求される空間解像度は従来より低くても充分となる。代わりに、

その分システムの高速化がもし達成されるならば、制御におけるフィードバックレートを向上させ、さらに高速な動きによって高度な能動性を実現することができる。これは同時にビジョンシステムの目的であるロボットの制御に対しても、高速かつ動的な制御のために有効な方法と言える。

しかしこのような、自らの能動的な運動も加えた処理と機構を実現し、これを前提とした設計によるシステム構築のアプローチは、これまでのビジョンシステムでは見られなかった。

1.5 研究の目的

ここまで述べてきたように、従来、視覚によるロボットの制御に対してビジョンシステムの最適な設計がなされていなかったことにより、特に高速性の面で充分な性能が実現されていなかった。その結果、ビジュアルフィードバック制御におけるダイナミクスの問題を解決することが困難となり、さらにこれによって視覚による制御のみならず、ロボットの他のセンサフィードバック制御との融合についても妨げられていたと言える。

以上のような背景から本研究は以下のことを目的として行った。

1. ビジュアルサーボのためのビジョンシステムのトップダウン設計
2. 超並列・超高速ビジョンシステムの構築
3. 新しい動的なビジュアルフィードバック制御法の提案

これを図 1.1 に示す。

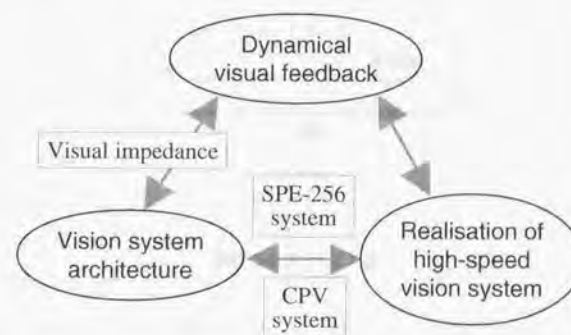


図 1.1 本研究の目的

すなわち、1 によってビジュアルフィードバックに何が必要になるかを明らかにし、ついでこれにもとづくビジョンシステムのアーキテクチャについて考察してトップダウンによる設計を行った。さらに2に挙げた実際のビジョンシステムの実現を目的として、システムの実装方法について検討し、

異なる2つの方法を提案して、それぞれシステムを構築した。最後に3で挙げた新しい制御法を提案し、実際のロボットによる実験を行って提案した方法の有効性を示した。

1.6 本論文の構成

本論文の構成を図1.2に示す。

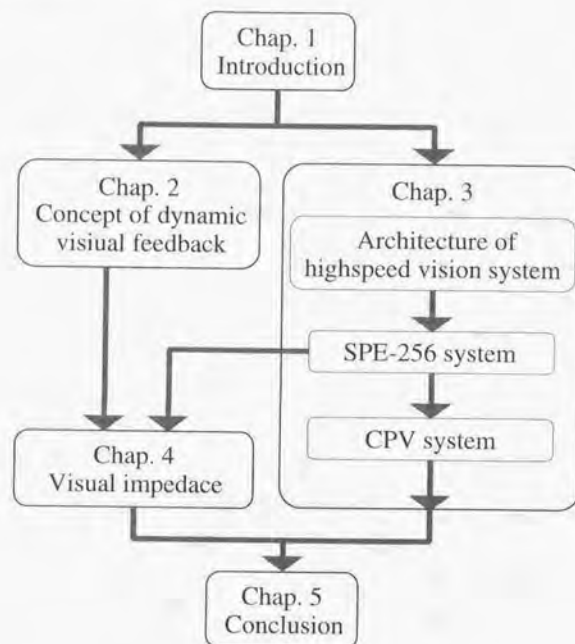


図 1.2 本論文の構成

以降の章では、まず第2章でビジュアルフィードバックの概念と理論について、特にダイナミクスの問題を中心に論じる。

第3章では第2章での議論にもとづいて超高速ビジョンシステムのアーキテクチャについてその基本的な考え方を述べ、これを実際のシステムとして実現するための様々な要因について検討し、トップダウンによるシステム設計をおこなう。さらにこれに基づいて、完全並列のアプローチによる SPE-256 システムと列並列のアプローチによる CPV システムという2つの異なる方法によるシステムを構築し、それぞれ実験と評価を行う。

第4章ではダイナミクスを取り入れた新しい方法としてビジュアルインピーダンスを提案する。ここではその概念と理論について詳しく述べ、第3章で説明した構築したシステムを用いて実験を行い、その有効性を確かめた。

最後の第5章ではこれらの結果をまとめ、今後の展望を述べる。

第2章

ビジュアルフィードバックのモデル

この章では本研究におけるビジュアルサーボのモデル化を行い、従来の研究の中での位置づけを明らかにする。これにもとづいて、従来のビジュアルフィードバックの研究の問題点を明らかにする。

2.1 はじめに

ロボットが未知の環境や変化する環境で活動する場合、視覚情報の利用が大きな効果を生むと考えられる。序論で述べたように、従来からこのような視覚の重要性が認識され、これを用いたロボットの制御について多くの研究が行われてきた。

一般に画像センサから得られる信号は大量かつ冗長で、ビジュアルフィードバックを行うためには何らかの処理が必要になる。従来このような画像処理の問題に関しても様々な研究が行われてきた。一方で、近年盛んに研究されているビジュアルサーボの研究は、これとは逆に、画像処理は、視覚からとにかくタスクに関連づけられた何らかの値(これは特徴量と呼ばれる)を得ることに限定し、これをフィードバックして構成する閉ループ制御系のロバストな性質を利用することで、タスクの実現を目指している。本研究においても対象認識や複雑な画像処理については、問題をなるべく単純化して単純なもののみ考える。

またこれまでのビジュアルサーボに関する研究では、予測や推定を用いることでいたずらに問題が複雑になっていることがある。これに対して本研究では制御についても単純化を目指す。

すなわち、サーボレートの高速化や、高速アクティブビジョンの利用など、センサや能動性を含めた全体で、ビジョンシステムの設計と構築をめざし、その結果ビジュアルサーボの問題を単純化することを目指している。

以下ではビジョンシステムとビジュアルフィードバックをモデル化し、従来のビジュアルサーボの研究を概観して、ダイナミクスとサンプリングレートの問題について考察する。ついで新しい高速アクティブビジョンの提案を行う。

2.2 カメラモデル

最初にビジョンシステムを構成するカメラの撮像モデルを示す。カメラの撮像モデルは一般に透視写像で表現できる [36]。

3次元空間内の点 $P(x, y, z) \in \mathbb{R}^3$ が、画面上の点 $P'(X, Y) \in \mathbb{R}^2$ に像を結ぶとする。レンズに歪みがなく、レンズの主軸が画像面と直交する理想的な状態を仮定すると、このとき画像面の X 軸と Y 軸を3次元空間の x 軸と y 軸と一致させ、また z 軸がレンズの主軸と一致するように座標系をとることができる。

以上の準備のもとで、点 P と点 P' の間には次の関係が成り立つ。

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \frac{f}{z} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.1)$$

となる。ただし、 f はカメラの焦点距離である。

本研究ではビジョンシステムは常に1台だけという状況で議論を進め、ステレオや3次元認識の問題は、本論文では触れない。

このとき対象物は常にカメラから一定の距離にあるものとして、その距離は既知であるとすと、

$$\frac{f}{z} = c \quad (c: \text{const}). \quad (2.2)$$

となる。このように仮定すると、(2.1) 式は次式のように正射影で表わされる。

$$\begin{bmatrix} X \\ Y \end{bmatrix} = c \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.3)$$

これは対象がカメラから充分遠く、かつその場所であり動きが小さいときにより近似となる。

2.3 マニピュレータの機構モデル

一方剛体リンクからなるマニピュレータが m 個の関節を持つとき、関節の自由度に対応して m 個の一般化座標によって表わされる [37]。これをベクトル $q \in \mathbb{R}^m$ で表すことにするとこのときマニピュレータの手先の位置姿勢 $s_r \in \mathbb{R}^6$ は、関節角 q の関数として、

$$s_r = s_r(q) \quad (2.4)$$

と表される。

あとで述べるアクティブビジョンについても全く同様に定式化できる。アクティブビジョンに搭載された機構の位置姿勢を $s_c \in \mathbb{R}^6$ とすると、マニピュレータのモデルと同様に、 n 個の回転関節からなるとして、関節角ベクトル $q' \in \mathbb{R}^n$ を用いて、

$$s_c = s_c(q') \quad (2.5)$$

と表すことができる。

このときマニピュレータの関節 q の微小な変化に対する姿勢 s_r の変化を表わしたものを、マニピュレータのヤコビアン $J_{robot} \in \mathbb{R}^{6 \times m}$ と呼ぶ。

$$J_{robot} = \frac{\partial s_r}{\partial q} \quad (2.6)$$

2.4 ビジュアルサーボ

2.4.1 画像特徴量

ビジュアルサーボでは、サーボを駆動する誤差情報として、画像から抽出した特徴量を利用する。

画像から抽出する特徴量としては様々なものが考えられるが、例えば Jang ら [38] は、(3.1) 式で合わされるような画面の各点での演算 k と、画面全体にわたる積分との組み合わせで、特徴量を表現することを提案している。

$$\xi = \iint_{Image} k(X, Y, I(X, Y)) dX dY \quad (2.7)$$

ただし $I(X, Y)$ はその点での画像の強度を表わす。(3.1) 式における k の取り方によって、領域の面積、中心、モーメント、ある方向への投影、テンプレートマッチングなどが表現できる。

すなわち各画素ごとの並列処理により

1. 各画素ごとのセンサ情報 $I(X, Y)$

2. 各画素の座標値 X, Y

にもとづいて $k(X, Y, I(X, Y))$ が計算され、次に演算の結果は

3. 画像全体にわたる積分計算 (総和演算)

によって2次元のパターン情報から特徴量へと変換される。このような処理構造に関する考察にもとづいて、次章では、ビジョンシステムの処理アーキテクチャとして、超並列ビジョンシステムの構成を導く。

このようにして得られる画像特徴量 ξ について、いくつかの種類についてこれを求めて並べたものを、画像特徴量ベクトル ξ と呼ぶ。画像特徴量ベクトル ξ は、マニピュレータと対象物、及びカメラの位置姿勢の関数となる。かりに対象物体が動かないとすると、

$$\xi = \xi(s_c, s_r) \quad (2.8)$$

となる。

ここで、一般にマニピュレータが微小に動いたときの画像特徴量ベクトル ξ の変化を画像ヤコビアン J_{image} と呼ぶ。

$$J_{image} = \frac{\partial \xi}{\partial s_r} \quad (2.9)$$

あとで述べるように、ビジュアルサーボにおいて、画像ヤコビアンが重要な役割を果たす。

一方制御という目的に対して特徴量が満たすべき条件があげられる。この条件については、例えば橋本 [6] に詳しい。

2.4.2 ビジュアルサーボの構成

ビジュアルサーボは次の2つの観点から4つに分類される [6-8]。

1. 制御構造が、ビジュアルサーボと関節角サーボとで階層的かどうか。
(look-and-move vs visual-servo)
2. フィードバック制御における誤差信号が画像特徴空間か、実空間か。
(特徴ベース vs 位置ベース)

それぞれの特徴と、本研究の位置付けは以下になる。

- 1で階層的である場合は、視覚フィードバック情報は目標関節角を出力するだけに使うので、視覚にもとづく動的な制御はできない。ここでは、マニピュレータの固有振動数程度の高速な動きを制御の対象としてこの方法は除外し、視覚情報からロボットの制御量を直接出力するサーボを考える。
- 2の分類では、一般に視覚ベースの方が位置ベースよりもキャリブレーション誤差に強いという特徴を持っている。特徴点から位置への変換が単純な線形変換で計算される場合はほとんど同じ条件となるが、ステレオや、物体のモデルを用いる場合は問題となる。本研究では視覚ベースの方法を用いる。

このようにして視覚ベースで直接サーボを駆動する場合、画像ヤコビアンが重要な役割を果たす。一般のビジュアルサーボの手法において、マニピュレータへの指令値の計算は次のように行われる。ただし、マニピュレータの自由度を表す q の次元 m と、 $J = J_{image} J_{robot}$ のランク $k = \text{rank } J$ との関係により次の2つの場合に分けられる。

- $k = m$ のとき、 J が正則なので逆行列 J^{-1} が存在する。例えば望ましい特徴量の変化 $\dot{\xi}$ が与えられた時、マニピュレータへの速度出力指令は、例えばよく用いられる方法は、

$$\dot{q} = J^{-1} \dot{\xi} \quad (2.10)$$

で与えるものである。ただしこれについての問題は次節で述べる。

- $k \neq m$ のとき、 J が特異となり疑似逆行列を用いる [8]。

ただし、特に $k < m$ のときは、ロボットの自由度に対して画像特徴量が十分でない場合であり、この場合は他のセンサからの情報を統合して制御を行う必要がある。その際に、従来のシステムではロボットの他のセンサと視覚センサのサンプリングレートが大きく異なることが制御系を組む上で問題となる。そのため、このようなビジュアルサーボを含む異種のセンササーボを統合した研究としては、視覚制御の方向と他のセンササーボの方向を完全に直交化したハイブリッド制御と用いたもの [23, 39] や、タスクの遂行状況によりサーボループを切替えるもの [22] などが見られる。

2.4.3 システムのダイナミクス

ビジュアルサーボにおいてより高速な応答を得ようとする、制御系のダイナミクスに対する考慮が重要となる。例えば、橋本らは、ロボットの動特性を考慮した動的システムとしてビジュアルサーボ系を定式化し、タスクレベルでの逆動力学 (task level inverse dynamics) を非線形フィードバックの手法を用いて計算し、制御を行っている [20, 40] ところが、サーボの制御量はサンプリングされた画像に対する画像処理によって得られるので、この画像処理のレートでサーボレートが規定される。

後で述べるように、ロボットの制御では、最も下位のサーボレートとして 1ms 程度が必要とされており、従来のビデオフレームレートで制限されたビジュアルサーボでは不十分となる。これについて、例えば Corke らは CCD カメラを用いたビジュアルサーボ系のダイナミクスについて、画像のサンプリングと伝送のタイミングなどを考慮して詳細な検討を行っている [13]。その結果、サンプリングレートについては最大限高速にした方がよいとしたうえで、その場合でも、

- サンプリングから制御までの遅れが系の性能に大きな影響を与えるので遅れ補償が必要であるが、実際には正確なモデルを得るのが難しく、安定でかつ十分な性能を持たせるのは容易ではないこと。
- 現在の 50Hz 程度のサンプリングレートではロボットのトルク制御を実現するのは困難であり、速度制御のマイナーループが必要となること。

などを指摘している。

また先に挙げた橋本らの研究では [20]、望ましいロボットのサンプリング周期 (1ms) を得るために、カメラのサンプリング周期 (33ms) の間で対象は動かないという仮定を設けて内挿を行っている。

しかし実際に行いたいのはビジュアルサーボで対象の高速な動きを捕らえることである。そこで、対象の動きを予測するためにカルマンフィルタを用いた研究 [15] や、対象の運動をモデル化してオブザーバで推定する研究 [21] がある。あるいはもっと詳細な運動モデルが得られる状況に限れば、予測はかなり有効となる。例えば Rizzi らは juggling において自由空間でのボールの運動モデルを立て、オクルージョンがあるような状況でも、完全な予測をもとにトラッキングを実現している [41]。

しかし、それらの方法では画像のサンプリングレートが充分ではないので、外乱に弱いという、モデル化されていない対象の動きに対して著しく応答が悪くなってしまう [13]。

また実際にはこれ以外の多くの研究では、サーボのサンプリングレートやロボットのダイナミクスについては棚上げされており、制御性能の低い低速なシステムとなっている。

結局、ダイナミクスまで考慮した高速なビジュアルサーボが必要とされているにも関わらず、ビデオレートの壁のために研究と実用化がはばまれていた。よって視覚情報を最大限利用した実時間ロボットサーボを実現するためには、何らかの方法でビデオレートを超える高速な視覚システムを構築する必要があると言える。

2.4.4 サーボレート

一般にデジタルサーボでは、サンプリング間隔を短くすればするほどサーボの安定性と速応性が向上するということが広く言われている。例えば従来のビデオ信号を用いたビジュアルサーボのサン

プリングレートとしては、ビデオ信号をそのまま用いて 30Hz のビデオフレームレートでサーボを構成する場合と、フィールドレートを利用して 60Hz のサンプリングレートで制御を行う場合の 2 通りが考えられる。特に、ビジュアルサーボのように実時間での処理を特徴とする場合、高速に移動する物体に追従するためには 30Hz のビデオフレームレートでは不十分であり、解像度を犠牲にしてもフィールドレートを利用して 60Hz のサンプリングレートで処理を行った方が、より高い性能を実現できることが指摘されている [42]。

ところがロボットの制御レートについて考えると、ロボットの機械系が有する固有振動数 $1/T_m$ に対して、最も下位の関節サーボのサイクルタイム T_a が満たすべき条件として $T_a < T_m/10$ 程度が必要であると言われている [43]、よって一般的なロボットの固有振動数が 20~50Hz 程度であることから、ロボットのサーボのサイクルタイムとしては 1~5ms が必要となる [12]。

例えば、産業用ロボットの Unimation 製 PUMA560 では、関節ごとのサーボは 1 サイクル 0.875ms で計算を行っており、PID 制御に従ってアクチュエータを制御している [44]。このことから、ビデオ信号を用いたシステムでこの条件を満たすためには何らかの補間や予測を用いなければならない。

一方、あとで述べる SPE-256 システムを用いて構築したビジュアルサーボでは、1ms のサイクルタイムで画像を処理することができ、ロボットのサーボとして十分な速度を有することになる。

2.5 アクティブビジョン

2.5.1 アクティブビジョンの有効性

一般に視覚を用いてセンシングを行う場合、ビジョンセンサを能動的に動かすことで、より多くの有用な情報が得られると考えられる。これらは一般にアクティブビジョン [35, 45] と呼ばれている。

ここではビジョンセンサのセンシング領域をセンシング対象に追従させるように能動的に動かすことを考える。

固定されたビジョンセンサには、次のような欠点がある。

- タスクに関係しない無駄な領域が増えることによる、処理とデバイスのコストの増大
- 非常に広い画角を扱うレンズなど光学系の実装の問題
- 特殊なレンズを用いた場合は一般の射影幾何による解析が困難

これに対して、アクティブビジョンが有用な理由として次の点が挙げられる。

- 最初にアクティブビジョンの構成をとれば、センシング領域 (可動範囲) の拡大が容易
- 対象が常に画面の中心にあると仮定されることにより、理論、処理、デバイスの全てについて、近似や単純化が可能
- 運動によって実質的な解像度をあげることが可能

例として第 4 章で述べる実験で用いた 7 軸マニピュレータを取り上げて比較を行うと以下のようになる。

- マニピュレータの手先における絶対誤差を1mmとする。一方マニピュレータの可動範囲は直径1910mmとなる。これを必要な解像度と視野に変換すると、固定カメラでは約2000×2000画素が必要となる。 $f=1\text{m}$ 程度とすると画角は90度程度となり、非常に広角のレンズとなる。この場合光学系の定みの問題が避けられず、これを補正する処理が必要になる。またいわゆる広角レンズを用いた場合は射影幾何による解析は困難になる。
- 一方アクティブビジョンでは制御に必要な手先の周辺を見ればよい。実験では直径60mmの球を用いたので、約64×64画素程度あれば十分と考えられる。すなわち固定カメラの場合の1/400の画素数でよいことになる。画角は5度程度となってレンズにおける近軸光学近似が充分成り立つ。一方モータの可動範囲としては90度程度必要となるが、そのような設計は容易である。

このようにロボットや人間など、広い範囲を動的に運動するような対象をセンシングする状況では、視線を能動的に変化させるアクティブビジョンが有効になる。

これを実現するためには本章で述べるように、ビジョンシステムのアーキテクチャとして、アクティブビジョンアクチュエータを組み込んだモジュール構成にすることが重要である。

2.5.2 超高速アクティブビジョン

例えば高速に運動するロボットに追従し、ビジュアルフィードバック制御を行う場合を考える(図2.1)。これを実現するためにはセンシング対象となるロボットの作業領域、手先の高速な動きに対し

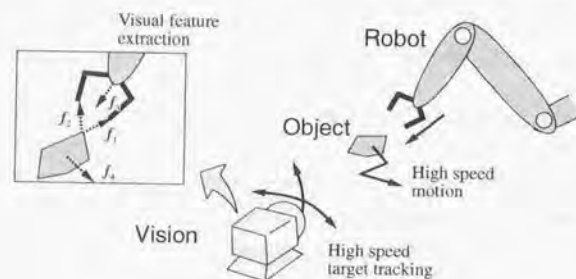


図 2.1 Dynamical active vision for robot control

で、アクティブビジョンも高速に追従しなくてはならない。これに対して従来提案されているアクティブビジョンの手法の多くは静的なものであった[46]。

同時に、ロボットの作業対象が未知物体である場合、センシングのためには対象の予測しがい動きに即応することが必要である。

このようなアクティブビジョンシステムを実現するためには、視覚センサをアクティブに動かす場合のシステムのダイナミクスについて、対象のダイナミクスに基づいて適切に設計を行うことが必

要となる。

そこで、ここでは実世界の対象の高速な動きに対応し、動的なトラッキングを行うアクティブビジョンシステムの設計方法について論じる[47]。

2.5.3 アクティブビジョンによるセンシングのモデル

ここでは簡単のために、センシングする対象は2次元平面上を移動する点とし、その状態が $\mathbf{x} = (x, \dot{x}, y, \dot{y})^T$ で表わされるとする。一般的には、センシングされる画像を領域として扱い、対象の状態変数として3次元実空間での運動を記述すれば良い。

ビジョンシステムの運動を表わす状態変数 θ についても、簡単のため、対象の存在する xy 平面と平行な平面上を移動できるものとして、状態変数が $\theta = (\theta, \dot{\theta}, \phi, \dot{\phi})^T$ で表わされるとする。また θ 軸と x 軸、 ϕ 軸と y 軸の方向が一致するものとする。

この系のダイナミクスは次式で表わせる。

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{D}\mathbf{u}_0 + \mathbf{G}\mathbf{w} \quad (2.11)$$

$$\dot{\theta} = \mathbf{A}\theta + \mathbf{B}\mathbf{u}_c \quad (2.12)$$

$$\xi = h(\mathbf{x}, \theta) + \mathbf{v} \quad (2.13)$$

ただし、 \mathbf{w}, \mathbf{v} は平均が $\bar{\mathbf{w}}, \bar{\mathbf{v}}$ で互いに独立、共分散行列がそれぞれ \mathbf{Q}, \mathbf{R} のノイズである。このように、センシング対象とアクティブビジョンの、両者のダイナミクスを考慮する必要がある。

$h(\mathbf{x}, \theta)$ はカメラによる観測を表し、画像上で $\xi = [\zeta, \eta]^T$ として画像中心からの偏差が観測されるものとする。ここでは簡単に次式で観測が表わされるとする。

$$h(\mathbf{x}, \theta) = \frac{f}{z} [x - \theta, y - \phi]^T, \quad f, z: \text{const} \quad (2.14)$$

ここで、 z は画像と対象との距離、 f はレンズの焦点距離を表わしている。

センシング精度については、画像センサの解像度が求める精度を満たしているかが問題となる。例えば画像のピクセルによる離散化誤差について、画像センサの画素ピッチを d_0 とし、必要な解像度を Δx とする。その結果(2.14)式の f/z について以下の条件が必要である。

$$\frac{f}{z} \geq \Delta x / d_0 = c_0 \quad (2.15)$$

2.5.4 トラッキング可能性の評価

アクティブビジョンでターゲットをセンシングし続けるためのトラッキングについて、これが可能な条件をトラッキング可能性と呼び、考察を行う。

画像センサで問題となるのは、センサの撮像範囲にある限界が存在することである。すなわち、(2.13)式で表わされるセンシングにおいて、 ξ のとりうる値の集合を \mathcal{X} とすると、ある撮像範囲 \mathcal{W} が存在して、

$$\mathcal{X} \subset \mathcal{W} \quad (2.16)$$

がなりたつ必要がある。

例えば、(2.14)式において撮像範囲の限界を ξ_0, η_0 とすると、 x と θ の偏差として許される範囲は、(2.15)式と合わせて、以下ようになる。

$$\begin{aligned} \left| \frac{f}{z} (x - \theta) \right| < \xi_0, \quad \frac{f}{z} \geq c_0 &\implies |x - \theta| \leq \xi_0 / c_0 \\ \left| \frac{f}{z} (y - \phi) \right| < \eta_0, \quad \frac{f}{z} \leq c_0 &\implies |y - \phi| \leq \eta_0 / c_0 \end{aligned} \quad (2.17)$$

(2.17)式で右部のどれかが満たされなくなったときは、トラッキングに失敗したものであり、以後のセンシングは続行できない。

これに対して、対象のダイナミクスが(2.11)式で与えられた場合、内部モデル原理に基づいて制御系を構成し、(2.17)式の拘束条件を満たすように最適サーボ系を構成することができる。

2.5.5 未知性を含んだ場合のトラッキング可能性

ここでは、対象の運動に含まれる未知性や観測に含まれるノイズの影響を考慮してトラッキング可能性を評価する。その方法として次のものを提案する。

(2.11)(2.13)式から、対象の状態 x は直接観測できず、また常にノイズの影響を受けている。このとき、誤差分散 Q, R がわかればカルマンフィルタなどを用いて x の推定値 \hat{x} を推定することができる。さらに、 \hat{x} の推定誤差分散行列 P が計算されるので、ノイズのガウス性を仮定すれば \hat{x} の累積誤差分布関数 $\Phi_P(z)$ が求められる。

そこで(2.17)式に対応して、 x, y に対する累積誤差分布関数からなるベクトルをあらためて $\Phi_P(z)$ とすると、

$$q = \left(\Phi_P\left(\theta - \frac{\xi_0}{f_0}\right) + \Phi_P\left(\theta + \frac{\xi_0}{f_0}\right) \right) \quad (2.18)$$

として、ある瞬間にトラッキングが実現されている確率ベクトル q が計算される。これに対して、トラッキング可能性として次の評価値を考えることができる。

$$J = r \cdot q \quad (2.19)$$

ただし、 r は重みベクトルである。これによって、アクティブセンシングにおいて、ノイズが含まれる系のトラッキング可能性が評価できる。

例えば、システムの画像サンプリングレートが大きくなると、ターゲットやアクティブビジョンの動きに伴うノイズや誤差の影響が大きくなる。その結果、トラッキング可能性は低くなる。

このように視覚を用いたアクティブセンシングでは、アクティブビジョンの動特性とセンシング性能との関係を考慮することが重要であり、この章ではその方法の一つとしてトラッキング可能性の評価を提案した。これを用いることで、センシングを続けるために必要なアクティブビジョンの動特性について評価し、対象の動特性に合わせた高速な動きと高い応答性をもつシステムの設計が可能になる。

第3章

ビジョンシステムの構成

この章では、まずビジュアルサーボを実現するための、ビジョンシステムのアーキテクチャについて考察する。次にその結果にもとづいて、完全並列と列並列という2つの異なる方法によりシステムを構築し、それぞれ性能を評価して、その有効性を示す。

3.1 はじめに

前章で述べたように、ロボットの動的なビジュアルフィードバック制御を実現するためには、センサフィードバックレートが充分であることが必要であった。しかし従来開発されてきたビジョンシステムでは、特にセンサデバイスである CCD カメラとビデオ信号に限界があるという点で、最適なビジョンシステムが実現されているとは言えなかった。

本来システム設計においては、まず与えられるタスクからシステムに必要な性能を導き、次にそこから実際のシステムを構築する手段を検討するという手順を踏むことが重要である。これに対して従来のビジョンシステムでは撮像デバイス、画像処理システム、およびアクティブビジョンのための機構とアクチュエータは、それぞれ別個に最適化されたものであっても、これをひとつのシステムとして組み上げた時に、全体としてタスクに対して最適なものにはなっていない場合があった。

あるいは目標とするタスク、あるいはそれに必要な仕様の明確化が充分なされていなかったために、ある程度の性能が達成されたとしても、それを幅広く応用するには到らない場合もあったと考えられる。

そこで本研究ではまず従来のビジョンシステムのビデオ限界を打ち破る方法として超並列・超高速ビジョンシステムの考え方について説明し、前章で議論したビジュアルサーボとアクティブビジョンの実現を前提として、これに必要な仕様を詳細に検討する。次にこれを実際に構築する方法について、いくつかの方法について考察を加える。また画像処理部分以外の周辺システムについても、そのアーキテクチャや必要な仕様について検討を行う。

最後に以上の結果をふまえて、実際に2つの異なる方式によるビジョンシステムを構築する。その結果、実現された性能を実験によって明らかにして評価を行う。

3.2 ビジョンシステムのアーキテクチャ

本研究ではビジョンシステムのアーキテクチャとして、次の構成要素を想定する。

1. 画像処理システム
2. システムコントローラ
3. アクティブビジョンシステム

1. の画像処理システムでは画像の取得と処理を一体として行うものとする。2. のシステムコントローラでは画像処理結果を取り出し、外部システムやアクティブビジョンシステムとのインタフェース、およびシステム全体の統合と調整を受け持つ。3. のアクティブビジョンシステムは、前章で述べたように、与えられたタスクにもとづいて、主にビジュアルフィードバックによりビジョンシステムの能動的な動きを実現する。

これに対してビジョンシステムの外部制約としてタスク環境とロボットシステムが存在する。

本研究でのロボットシステムでは、ビジュアルフィードバック制御がなされるものとする。これを実現するために、ロボットの制御システムはビジョンシステムからの情報をリアルタイムで受け取って、ロボットの制御を行う。あるいはロボットシステムがアクティブビジョンに対して、タスクの要請にもとづいて指令を与えることもある。タスク環境はビジョンシステムとロボットをとりまく環境であり、第1章で述べたように、動的に変化し未知性の大きい環境を想定している。

以上のビジョンシステム概念を図3.1に示す。

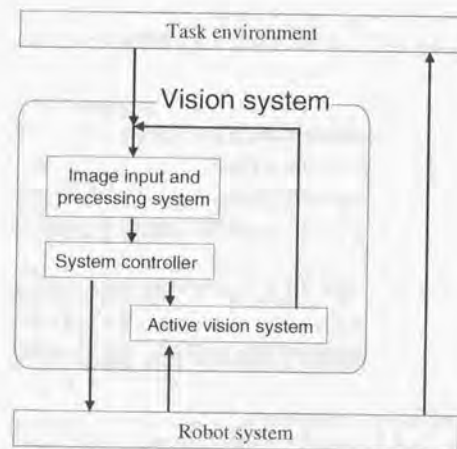


図 3.1 Concept of vision system

このようにビジョンシステムのアーキテクチャを考えると、環境とロボットという制約を考慮した上で、ビジョンシステムの仕様と性能を検討し、これにもとづいてそれぞれの内部システムの仕様と構成を設計すればよいと考えられる。

以下では、それぞれの構成要素のアーキテクチャについて、順に考察していく。

3.2.1 超並列・超高速画像処理

先に述べたように従来の CCD カメラを用いたビジョンシステムでは、画像の取り込みレートがビデオ信号の伝送レートで制限されていることが問題となる。このような問題を解決するためには、IO ボトルネックとなっている画像のシリアル伝送の部分で解消し、同時に処理についても並列処理の手法を導入して、全体としてボトルネックがないように、画像センサと処理システムを構成することが必要である。

さらに、特に処理については、画像を取り込んだ直後に行うフィルタ処理などのいわゆる初期視覚処理では、画像処理における各種の演算が全画素について共通で、かつ近傍どうしの演算が多用されることに注目する。すなわち、センサから取り込んだ画像をそのままの形で処理すること、言い換えれば、それぞれの画素について 2 次元のトポロジを保ったままで、完全並列な処理を実現することが有効である。

このようなアプローチとして超並列・超高速画像処理によるビジョンシステムを提案する。さらに提案するビジョンシステムの具体的な姿を明らかにするために、まず本章の最初に挙げたロボットとタスクの制約について考察し、その後これにもとづく具体的な実装方法について検討する。

3.2.2 画像特徴量抽出処理

次に、ビジュアルフィードバックに必要となる特徴量抽出処理の構造について考察する。

先に 2.4.1 節で述べたように、ビジュアルフィードバックにおける画像特徴量のモデルとして、(3.1) 式の表現が一般に知られている [38]。

$$\xi = \iint_{I_{\text{image}}} f(X, Y, I(X, Y)) dX dY \quad (3.1)$$

ただし X, Y は座標値、 $I(X, Y)$ は各画素のセンサ入力である。演算 f のとり方によって、領域の面積、中心、モーメント、ある方向への投影、テンプレートマッチングなど様々な特徴量が表現される。

この処理の構造をみると、各画素ごとのセンサ情報と座標値にもとづく演算 $f(X, Y, I)$ と、その結果を画像全体にわたって積分計算 (総和演算) するパターンから特徴量への変換からなることがわかる。またノイズの除去やエッジ抽出などの初期視覚処理についても、各画素ごとの近傍演算が多用され、これは演算 f に含めることができる。

この結果、ビジョンシステムの画像処理部のアーキテクチャとして次の 3 つの機能が必要になることが自然に明らかになる。

1. 各画素ごとの完全並列演算機能 ($f(X, Y, I)$)、ただし特徴量ごとに演算が可変 (プログラマブル) であることが必要。

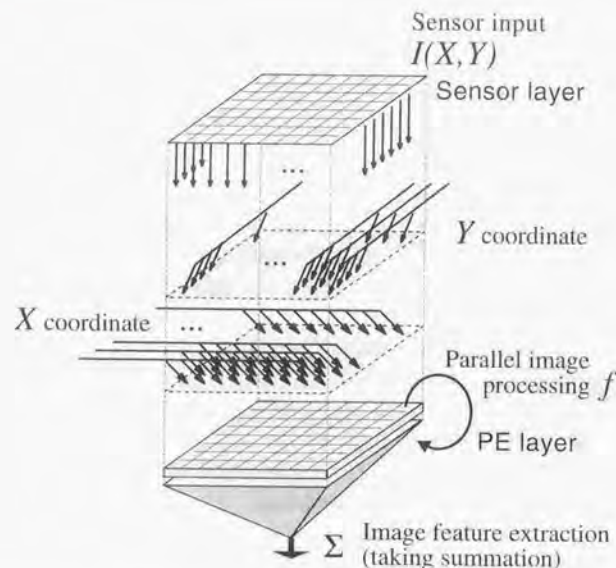


図 3.2 Architecture for image feature extraction

2. 各画素に座標値 (X, Y) を送り込むための、行 / 列並列のブロードキャスト機能、
3. 2次元アレイ状に得られる並列処理結果を特徴量に変換する総和演算機能、

これを Fig.3.2に示す。

このように、画素ごとに完全並列な処理に加えて、行 / 列並列のブロードキャストや総和演算の機能を実現することにより、効率的な特徴抽出の計算が可能になる。

3.2.2.1 画像処理システムと情報量

次に画像の情報量について考える。画像に含まれる情報量については、例えば画像圧縮の分野で研究が盛んである。ただしこれらの研究では画像の情報量を考える出発点として、次の2つを考慮することが多い。1つは既に取り込まれた画像、あるいは規格として定められた画像にどれだけの情報が含まれているかを考えるものであり、もう1つは最終的に人間に提示することを目的として、人間の視覚特性を考慮して、画像にどれだけの情報を含ませればよいかを考えるものである。

いずれにしても、本研究ではビジュアルフィードバックという目的のために、どのような画像情報が必要かを考えなくてはならないので、画像圧縮の分野で用いられるこれら2つの方法は、あまり

役に立たない。そこで、ここでは定性的な議論ではあるものの、本章の最初に示したタスクとロボットの制約条件から仕様を検討を試みる。

まず最も単純に、ビジュアルフィードバックにおける画像センサから取り込まれる連続画像の情報量 I について、センサで取り込んだすべてのデータが等しく均一な情報を持つとすると I は以下のように表される [48]。

$$I = N \times M \times D \times L \quad (3.2)$$

ただし N, M は画素数 (pixels) を表し、 D は画素のグレイレベルのビット数 (bits) を、 L は単位時間あたりのフレーム数 (fps) を表す。ここで (3.2) 式において、右辺 $N \times M$ は空間解像度に対応し、 L は時間解像度に対応していることに注目して、これらの定性的な関係をグラフに表わすと、図 3.3 のようになる。

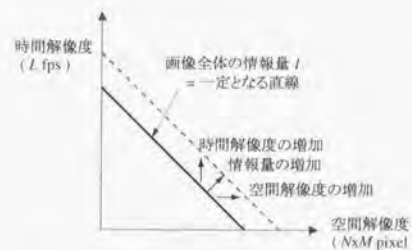


図 3.3 時間と空間の解像度と情報量 (Log scale)

これに対して、先に挙げたタスクとロボットの制約条件から、実際の情報量には偏りがあると考えられる。このような定性的な情報量の傾向をいくつかの根拠と共にグラフに示すと時間解像度については図 3.4 のように、空間解像度については図 3.5 のようになると考えられる。

この結果、一般的な人の生活空間やロボットの一般的なタスク環境でビジュアルフィードバック制御を行うことを前提とすると、時間と空間の解像度に対して、必要な情報量は図 3.6 のように変化すると考えられる。

すなわち、ある一定の低い空間 / 時間解像度しか確保されない場合には、もう一方の時間 / 空間解像度をあげても全体としての情報量は、あまり大きくならない。一方、時間と空間の両方の解像度を上げるにあたって、そのコストにトレードオフの関係がある場合は、最小のコストで最大の情報量を得ることができる。適切な時間と空間の解像度が存在する。

この結果から次節以降で述べる超並列・超高速ビジョンシステムの仕様を決定する際に、最適なビジョンシステムの存在が予想される。

一方、ロボットのダイナミクスを制御するためには、時間解像度についてある一定以上のレートでフィードバックを行う必要がある。これは 2.4.4 節で議論したように、一般に、理論上で導かれるサンプリング定理よりも厳しい条件となることが指摘されており、ロボットの機械系の固有振動数 $1/T_m$

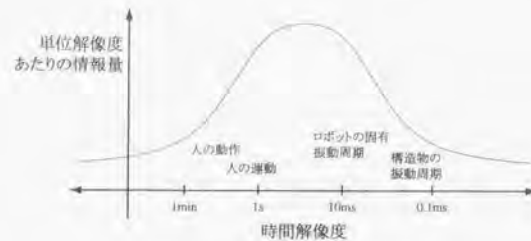


図 3.4 時間解像度に対する情報量

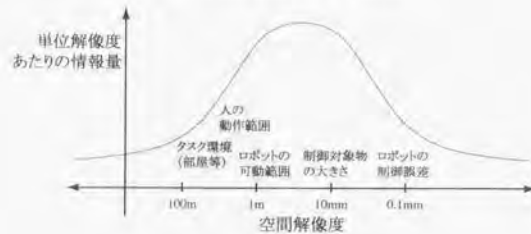


図 3.5 空間解像度に対する情報量

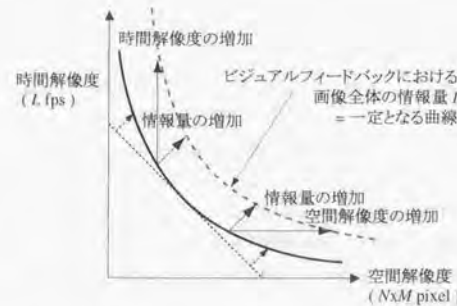


図 3.6 制約条件を考慮した時間と空間の解像度と情報量 (Log scale)

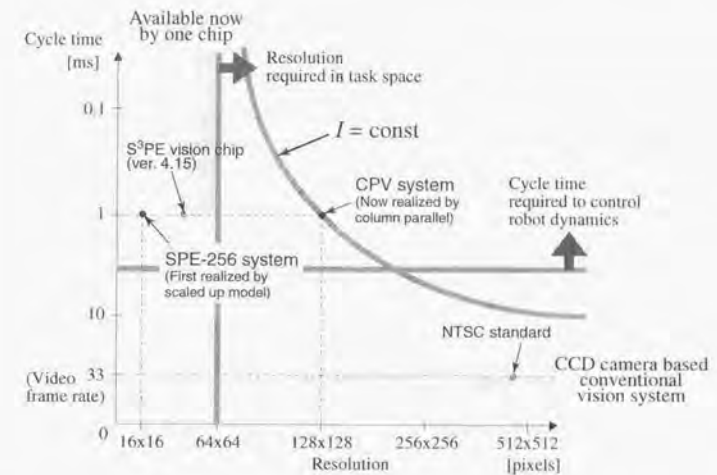


図 3.7 Comparison with conventional image system

に対して、サンプリング周期 $T_s < T_m/10$ を満たすことが必要である。これを第4章で実験に使用するマニピュレータの第5軸の固有振動数 $1/T_m = 26.2\text{Hz}$ に当てはめると、ビジョンシステムのサンポレートは $1/T_s > 262 \approx 300\text{Hz}$ となることが分かる。

同じく2.5.1節で議論したように、空間解像度についても必要な最小分解能と視野の範囲から、対象を認識するために最低限必要な解像度が存在する。例として、第4章での実験の条件では、 64×64 画素程度が必要であることが示されている。

これらをまとめると表3.1のようになる。

設計条件	制約条件	限界値を決める要因	限界値
空間解像度	対象物の認識	視野と解像度の関係	約 64×64 画素
時間解像度	ダイナミクスの制御	サンプリング定理	約 300 fps

表 3.1 時間と空間の解像度についての限界

これらの結果から、以下の節で説明する実際の各アプローチの比較を含めて図3.7に示す。

3.2.2.2 1チップによる実現

石川らはビジョンチップ [1] とそのアーキテクチャ S³PE (Simple and Smart Sensory Processing Elements) を提案し、実際にチップの開発を行っている [2, 49]。これは、図 3.8 に示すように全ての画像センサに対して、超並列のプロセッシングエレメント (PE) を 1 対 1 に直結したもので、これらを 1 チップに納めたものである。これによって画像伝送のボトルネックを解消し、同時に超並列・超高速の視覚処理を実現して、従来のビジョンシステムでは不可能な非常に高速なビジョンシステムを可能にするものである。

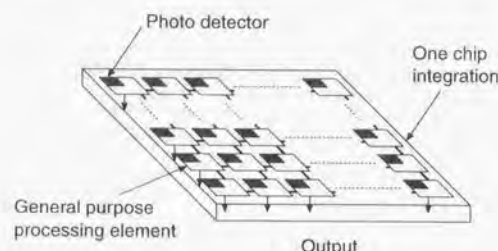


図 3.8 One chip integrated massively parallel vision system

またロボットの制御を考えると、制御システムの必要に応じているような画像処理アルゴリズムを駆使して画像から情報を抽出する必要がある。これについては、ビジョンチップの PE として汎用のプロセッシングエレメントを用いることで実現している。

ただしここで問題となるのは、1 チップ上に多数のセンサと処理エレメントの集積化が必要となる点である。集積化のためにコンパクトな PE が求められる一方で、多様な処理を実現する汎用な処理機能を持つためには、ある程度の PE の規模が必要となり、結局現在の半導体技術では、64×64 画素程度が現実的に可能な限界であると試算されている [2]。すなわちこれ以上の数の画素を集積化するためには、今後の半導体技術の進歩を待たなければならない。

現在 16×16 画素をもつビジョンチップが動作しているが [2]、応用システムとしての動作はまだ実現されていない。

3.2.2.3 完全並列接続による実現

一方、1 チップによるビジョンシステムの構築に先行して、1 チップ化の前段階、スケールアップモデルとして位置づけられる SPE-256 システムを構築した。

このシステムでは完全並列による実現を採用して、従来にない超高速なビジュアルフィードバックを実現することを第一の目標とした。その際、1 チップによるアプローチと異なり、ディスクリートを中心としたシステム構成をとることにより、動作状況を確認しながらの確実なシステム構築が可能

である。

実際に構築したシステムの詳細は本論文の 3.3 節で示す。構築したシステムでは、前節の試算で必要とされていた時間解像度の条件を満たす 1ms のフィードバックを実現しており、超高速なビジュアルフィードバックの実現という当初の目標が達成されている。

またこれをアクティブビジョンに搭載してターゲットトラッキングシステムを構築し [50, 51]、アプリケーションの開発や [52-55]、高速視覚の特徴を活かした新しいアルゴリズムの開発 [56] に利用されている。これらの応用実験により、従来の限界を超える超高速のビジョンシステムが、ロボットなどのビジュアルフィードバック制御において実際に大きな効果をもつことを示すことができた。

一方で、このシステムではディスクリートで完全並列システムを構築するという実装上の制約から、解像度が 16×16 画素、階調が白黒のバイナリのみと少ないものとなっている。この点についてはビジョンシステムに必要とされる仕様を満たしているとはいえなかった。

3.2.2.4 列並列接続による実現

これに対して、新しく提案する列並列接続のアプローチによるシステムは Column Parallel Vision (CPV) system と呼び、列並列の画像データ転送を特徴としている。

すなわちセンサと処理部を 1 チップ化するのではなく、別々のチップ、ボードに実装し、それらの間での画像の伝送を、列並列に、行方向にスキャンして行っている。この方法を採用した結果、完全並列の場合と同様、従来の CCD カメラの限界となっていたシリアル伝送の部分を解消し、通信ボトルネックを回避することができた。また同時に半導体の実装上の自由度が増すことにより、多数の画素をもつシステムの構築が可能になった。

完全並列と比較した場合、列並列の構成では行方向のスキャンが必要となるために、速度に一定の限界がある。しかし最初に明らかにしたビジョンシステムに対する要求仕様を満たす限り、最終的なタスクである動的なビジュアルフィードバックに対しては、最適な構成となる。

実際、CPV システムでは行方向のスキャンと完全並列な処理を組み合わせることで、128×128 画素という今までにない高解像度を実現し、同時にアプリケーションで必要とされる 1kHz の高速なサーボレートを実現することが可能となった。この情報量は先に述べた SPE-256 システムと比較して、画素数が 64 倍の 128×128 画素、解像度がバイナリから 8bit 階調と、単純な計算では 128 倍の情報量となっている。

また画像処理部としてビジョンチップと同様、画素ごとに完全並列な処理を行う S³PE アーキテクチャを用いており、画像の処理については超並列・超高速処理が実現されている。また AD 変換器についても 128 列並列に用意することで、8bit 階調の画像を取り込んで処理することが可能となった。

構築した CPV システムの詳細は 3.4 節で述べる。

3.2.3 システムコントローラのアーキテクチャ

並列画像処理システムを構成する場合、システム全体の動作を制御し、かつ外部システムとのインタフェースを行うコントローラの役割が重要になる [57, 58]。

前章で述べたビジュアルサーボを実現する場合に、ビジョンシステムでの典型的な動作は、例えば次のようになる。

- 画像を取り込む。
- 画素ごとの並列処理により、各種の画像処理を行う。
- 総和演算回路等を用いて、演算結果を特徴量として抽出する。
- 外部ホストシステムに演算結果を出力する。

これらの一連の処理を、要求されるフレームレート（ここでは 1kHz）で行う必要がある。

ここから具体的なコントローラの仕様を考えると、まずビジョンシステム全体の動作を管理するためのプログラム制御と、処理演算を行いながらビジョンチップへインストラクションを供給する機構が必要であり、同時にこのプログラムをストアするためのプログラムメモリが必要となる。

一方データの流れとしては、2.4.1節で示したように、並列処理の処理結果から必要な情報を抽出する総和演算機構と、コントローラからデータを送り込むバスとして、列並列のインタフェースが必要である。

これらの動作をビジョンシステムとして自律して行うことで、外部システムとの通信速度に依存せずに高速にビジョンチップを駆動することができる。図 3.9 にこの概念図を示す。

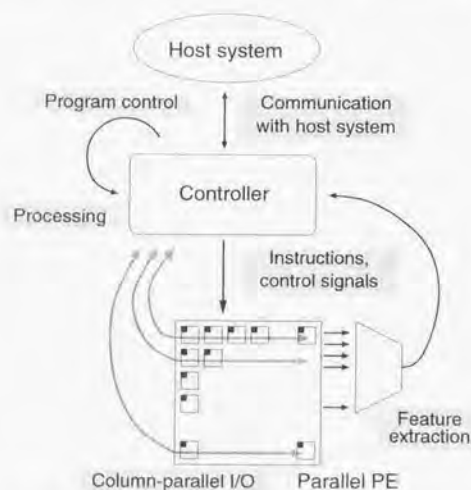


図 3.9 Concept of vision system control architecture

3.4節で述べる CPV システムでは、これらの概念にもとづいたコントローラアーキテクチャ設計がなされており、その詳細は 3.4.2.2 節に示されている。

3.2.4 アクティブビジョンのアーキテクチャ

前章で述べたように、アクティブビジョンにおける注視制御の枠組みから、ターゲットトラッキングを行うことが有効である。その際、高速な視線の追従を実現するために、ビジュアルサーボによりアクチュエータの制御を利用する。

具体的には画面の各軸に平行な pan, tilt 方向のアクチュエータを用いて視線方向の制御を行うことが考えられる。その際的设计指針として、以下の項目が挙げられる。

- ダイナミクスを含めたアクチュエータの制御が必要となるため、アクチュエータの制御において、トルクもしくは速度の低次の指令が可能であることが必要。また制御サイクルも十分高速である必要がある。
- サーボ動作における高い応答を実現するために、機構構造の最適化 (小型化 + モーメントの最小化) と適切な出力のアクチュエータが必要。
- 視線を移動させる移動範囲が想定されるタスクに対して十分であることが必要。
- タスクとのインタラクションを最小にするために、システムの物理的なサイズをコンパクトに設計することが必要。
- アクチュエータの制御処理を行う処理システムには、システムの動特性に応じた高速なシステムが必要である。

これらの指針に基づいて、アクティブビジョンシステムの設計を行う必要がある。

3.3 完全並列による超高速ビジョンシステム (SPE-256) の実現

3.3.1 概要

この節で説明する SPE-256 システムは、これまでに述べた超並列・超高速ビジョンシステムについて、全並列型のシステムとして構築した。

図 3.10 に示すように、このシステムでは並列の画像処理部として SPE-256 システムを用い、16×16 画素のセンサと処理部を、ディスプレイな配線により 1 対 1 に直結することで、センサ情報に対する完全並列な処理を実現している。その結果、1ms という従来にない高速なサイクルタイムでのビジュアルフィードバックを可能にした。

また PD アレイはアクティブビジョンに搭載され、視覚にもとづく視線の高速な制御が可能となっている。そこで、実験では構築したシステムを用いたターゲットトラッキングを行ってその有効性を確認した [50, 51]。

以下では、このビジュアルフィードバックシステムの構成と基本的な性能を示し、このシステムを用いたターゲットトラッキングの実験結果と性能の評価を示す。

3.3.2 システム構成

本システムは、図 3.11 に示すように、画像を入力して並列演算処理を行う SPE-256 システムと、その処理結果に基づいて視線を制御するアクティブビジョンシステムからなり、全体として 1ms

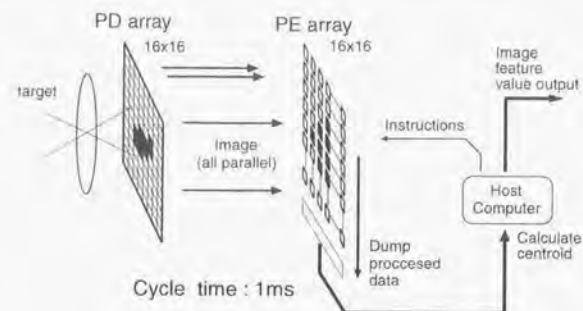


図 3.10 All parallel vision system SPE-256

のサイクルタイムでのビジュアルサーボを構成している。以下の節ではこれらのシステムを順に説明していく。

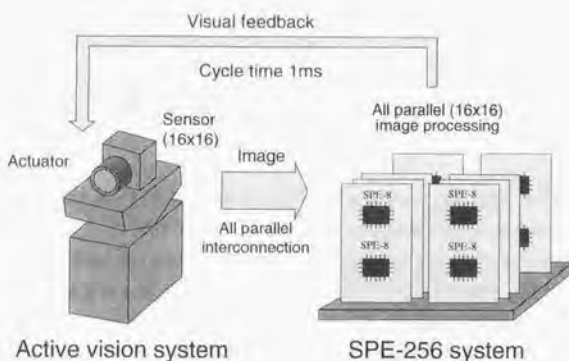


図 3.11 Overview of 1ms target tracking system

3.3.2.1 超並列・超高速画像処理部

並列演算処理部となる SPE は SIMD 型の制御によりビットシリアルに演算が行われる。演算ユニット機構としては、AND、OR、EX-OR、加減算、乗算が可能で、1PE あたり 24bit のレジスタを持っている。また各 PE はメッシュ状に配置され、それぞれ 4 近傍との入出力。センサ入力と処理結果の並列出力が可能である。この SPE の内部構造を図 3.12 に示す。

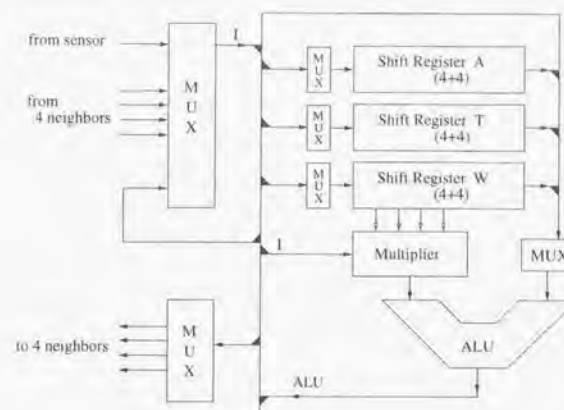


図 3.12 Block diagram of SPE

このアーキテクチャでは集積化のために 1PE あたりのゲート数を大幅に削減し、337 ゲートというコンパクトな回路が実現されており、現在 8 個の PE が集積化された LSI として SPE-8 が試作されている。この SPE-8 を 32 個用いて SPE-256 を試作した。

SPE-256 では $16 \times 16 = 256$ 個の PE をマトリクス状に配置し、これに対応した画像センサには浜松フォトニクス社から一般に市販されている 16×16 素子 PIN フォトダイオード (PD) アレイ S3805 (放射感度 0.5 A/W at 800nm, 応答速度 7ns) を用いている。この PD アレイの出力は全画素並列に取り出すことができるので、それぞれの PD 出力を PE と一対一に結線し、入力画像の各画素に対して完全並列な処理を実現している。また各 PE に送られた PD 出力は適当なスレッショルトレベルにより 2 値化され、それに対して SPE での種々の画像処理を行っている。

SPE では、ALU やレジスタを利用したマイクロプログラミングを行うことにより、得られた画像に対して様々な画像処理を行うことができる。表 3.2 に、一般的な画像処理を行った時の実行時間を示す。

3.3.2.2 システムコントロール

SPE ヘインストラクションを送る方法としては、3.2.3 節で示したように専用のコントローラを用いるべきであるが、今回は最初の試作ということもあり 4 章で示すロボット制御システムで用いられている Texas Instruments 社の並列処理用 DSP, TMS320C40 (256 MOPS, 320 Mbyte/sec, 20 Mbyte/sec の非同期データ転送) をホストとして使い、パラレル IO ポート (PIO) を介して直接インストラクションを送り出している。このため現在のシステムでは PIO の動作速度でサイクルタイムが制限されており、本来 SPE-8 のインストラクションサイクルが、最高 100ns であるのに対して、

表 3.2 Execution time for algorithms on SPE-256

Algorithms	Steps	Execution time	present	max
Edge	2-neighbor	23	287.5 μ s	2.3 μ s
	4-neighbor	33	412.5 μ s	3.3 μ s
Slewwindow	23	287.5 μ s	2.3 μ s	

サイクルタイム 12.5 μ s での動作となっている。

一方 PE での並列処理の結果はパターンとして得られるので、2.4.1節で述べたように、パターンの重心や総和を求めたものを画像特徴量とし、ビジュアルフィードバックに利用する。これまでに、このようなパターン情報の重心や総和を並列かつ高速に求めるものとして、抵抗ネットワーク回路 [59] やデジタル回路によるもの [60, 61] が提案されている。ただし今回はホストコンピュータとして用いた DSP, TMS320C40 をそのまま利用し、逐次処理によって重心を計算することにした。具体的には PE アレイの一边と PIO を接続し、そこから SPE で並列処理した処理結果を DSP へ送り出している。しかし、実際にはこの処理結果の排出のために 1 サイクルごとに 500 μ s 程度の時間を消費しており、システム全体のサイクルタイムである 1ms の半分を占めている。このことからわかるように、集積化ビジョンチップの処理の並列性を活かし、ボトルネックをなくすためには、重心や総和などの出力計算を並列に行う専用機構を実装することが重要である。

3.3.2.3 アクティブビジョンシステム (AVS-I)

アクティブビジョンシステムの可動部には PD アレイとレンズが載せられており、直交するチルト、パンの 2 軸についての回転自由度をもっている。それぞれの軸のアクチュエータにはハーモニックドライブシステムズ社製 DC サーボアクチュエータ RH-8 (チルト)、RH-11 (パン) を用いており、これらのサーボとしてサーボランド社製アナログサーボコントローラ SMCM2-AI (ステップ応答 0.2ms 以下) を用いて速度制御を行っている。従ってこのサーボに適切な速度指令値を与えてやれば、ビジョンシステムの視線方向を自由に変えることができる。

前述したように、PD で取り込まれた画像は各画素並列に SPE-256 へ送られ、2 値化された上で与えられたプログラムに従って画像処理が行われる。その処理結果は DSP へ送られ、重心を計算して画像特徴量としている。さらに DSP では、その画像特徴量の目標値からの偏差から、PID 制御によりアクチュエータへの指令を計算している。アクチュエータのサーボはこの DSP で計算された指令値に従って速度制御を行っている。

これら一連の処理によって、画像情報をフィードバックして視線を制御するビジュアルサーボを構成しており、そのサイクルタイムは 1ms となっている。SPE-256 を含めた全体システムでの処理の流れを図 3.13 に示す。またシステム全体の写真を図 3.14 に示す。

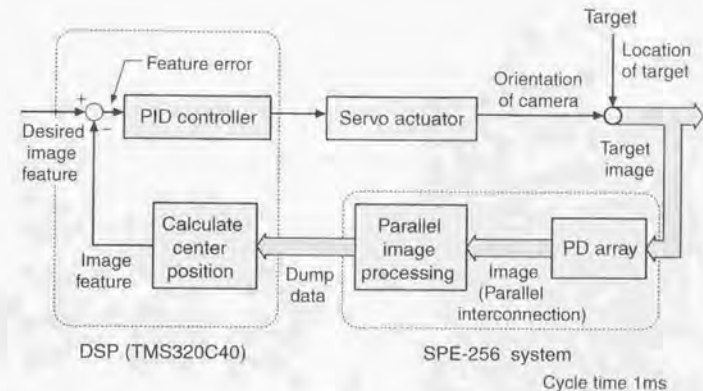


図 3.13 Block diagram of 1ms target tracking system

3.3.3 実験

3.3.3.1 アクチュエータの周波数応答特性

まず、アクティブビジョンシステムの基本的な動特性を評価するために、サーボループからビジュアルフィードバックの部分を除いた開ループでの周波数応答を測定した。この中にはアクチュエータやサーボコントローラ、視線の方向を変えるための機構部分の動特性が含まれている。これを最初に測定しておくことで、ビジュアルフィードバックをサーボループに組み入れた時に、そのシステムの性能がどの部分で制限されているかを評価することができる。

測定結果を図 3.15 に示す。アクチュエータに与えた速度指令に対して、実際に得られた速度のゲインと、位相遅れを示している。破線がパン方向の回転軸、実線がチルト方向の回転軸の応答を表している。ゲインのグラフ中に点線で引いた -3dB の線とグラフの交点から、カットオフ周波数は約 20Hz (時定数: $T_m=50$ ms) であることがわかる。

この測定で得られたパラメータをもとに、ビジュアルサーボを組む際にどの程度のサンプリングレートでフィードバックを行えばよいかを考察する。本論文のアクティブビジョンシステムを 1 次系とみなし、系の時定数 $T_m=50$ ms を基準として評価すると、SPE-256 システムで実現される 1ms のサンプリング間隔では、サンプリングタイム T_s が満たすべき条件 $T_s=1 < T_m/10=50/10$ ms を充分満たしていることが分かる。

一方従来のシステムのようにビデオレートの制限のためにサンプリング間隔が $T_s=33$ ms 以上になるとサンプリングタイム T_s が満たすべき条件 $T_s=33 \times T_m/10=50/10$ ms を満たせなくなり、この場合安定性を確保するために、アクチュエータが本来持っている動作性能を犠牲にし、ゲインを低く設定してサーボを設計せざるを得なくなる。



図 3.14 Picture of 1ms target tracking system

これらのことからわかるように、本システム程度の動作周波数を有するアクティブビジョンシステムに対してビジュアルフィードバックによる制御を行う場合、従来用いられているビデオ信号のサンプリングレートでは不十分であり、速応性を犠牲にして低速なシステムとして設計するか、何らかの予測を用いてフィードバックの遅れを補償するしかない。逆に、本論文のシステムのように、1msのサンプリング間隔でのビジュアルフィードバックが実現された場合、ロボットなどの制御のためのビジュアルサーボとしてそのまま利用できるだけの充分な高速性を有しているということが言える。

3.3.3.2 1ms ターゲットトラッキング

次に、実際にこのビジュアルフィードバックシステムを用いたターゲットトラッキングの実験の結果を示す。この実験では、PD から入力したターゲットの画像を2値化し、得られたパターンをそのまま用いて、その重心を画像特徴量とし、ビジュアルフィードバックを行った。また、通常のターゲットトラッキングでは、移動するターゲットに対して、ターゲットの画像の中心が画面の中央に見えるように視線を変化させるが、この実験の場合、追跡対象となるターゲットを高速ビジョンに見合うほど速く、かつ正確に動かすのは難しい。そこで、等価的に同じ状況を作るために、画面の中でターゲットが見えるべき位置を指定してこれを目標位置とし、ターゲットがその位置にくるように視線を

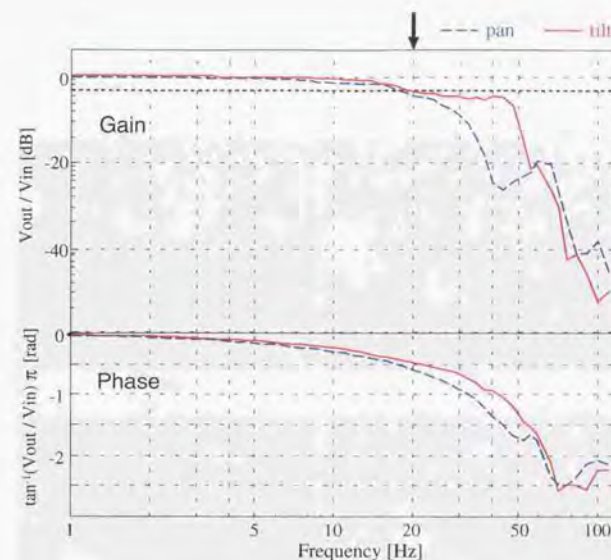


図 3.15 Frequency response of active vision

制御することにした。

実験結果を図 3.16 に示す。目標位置として 5Hz の三角波を与えてこれに追従させた。実線が 1ms のサイクルタイムによるターゲットトラッキング動作を示す。破線は、本システムにおいて動作サイクルをわざと落とし、ビデオレートと同じ 33ms のサイクルタイムでトラッキングを行ったものである。1ms では目標に対して充分追従しているのに対して、33ms では追従性能が著しく悪くなっているのがわかる。

特に 1ms でのトラッキングでは目標に対する遅れは約 20ms であり、センサフィードバックの遅れの影響はほとんどなく、アクチュエータなどの機構部分の動作周波数の限界だけから遅れが生じていると言える。これに対して 33ms のサイクルタイムでは約 70ms の遅れが生じており、オーバーシュートも大きくなっている。これは、アクチュエータの動作周波数の限界だけではなく、センサフィードバックが遅すぎるために、目標の動きの高速な変化に対応できないのだと考えられる。

なおこの実験では 1ms と 33ms の場合について、サイクルタイムのみを変えて測定を行っているが実際に CCD カメラを用いた場合はサイクルタイムと共に、33ms の位相の遅れが発生することに注意を要する。制御系の特性においては、この位相の遅れが大きな悪影響をおよぼすことが Corke ら [13] によって示されている。そのため実際の CCD カメラを用いたシステムではこの実験結果以上

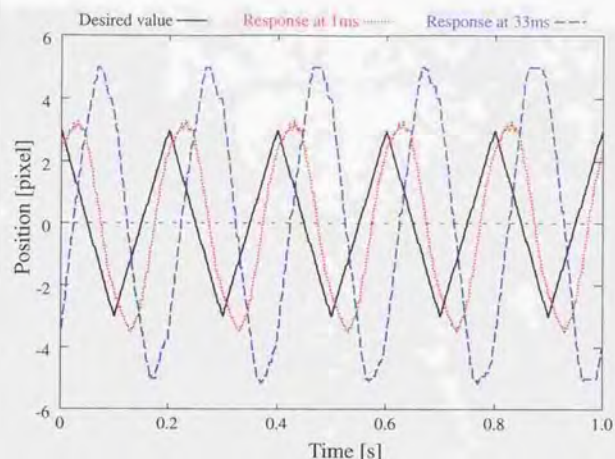


図 3.16 Tracking response to a 5Hz triangular wave (pan)

に応答が悪くなるものと考えられる。

次に実際に不規則に運動する物体に追従できることを確かめるために、白いボールを手でバウンドさせて応答を調べた [50]。実験の結果を図 3.17 に示す。図からわかるように、バウンドする瞬間のボールの不規則な動きにも追従できている。

第1章で述べたように、一般にセンサフィードバックの遅れを補償する手段として、対象の動きを予測するという方法がとられるが、この実験のように、対象の動き自体が予測不可能な動きをする場合には、その方法は有効ではなくなると考えられる。これに対して 1ms という高速なセンサフィードバックが実現された場合には、このような対象の動きに対しても、簡単な制御で十分な追従性能が得られることがわかった。

3.3.4 考察

3.3.4.1 ビジュアルフィードバックの高速性と応用アプリケーション開発

ここでは SPE を用いた完全並列処理によるビジョンシステムを実現し、センサ情報を並列に処理することで 1ms という高速なビジュアルフィードバックを可能にした。また、これを用いてビジュアルフィードバックによるターゲットトラッキングを行い、高速なトラッキングが実現されることを示した。

特に、このシステムでは現在ロボットの姿勢角センサとして用いられているポテンショメータやエンコーダと同等なサンプリングレートで、画像情報に基づいた実時間のサーボループを構成すること

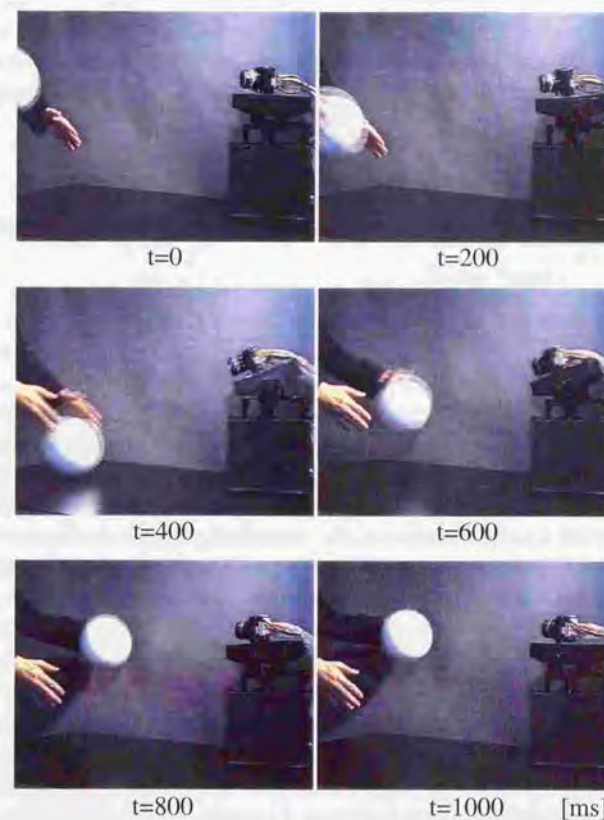


図 3.17 Tracking a bounding ball

ができるようになっており、視覚に基づく制御においても、他のセンサフィードバック制御と同様に、高速なフィードバックの実現が、サーボシステムの制御性能に大きな影響を与えることが示された。逆に言えば、このような高速性の実現により、従来視覚に基づく制御が必要と思われていた、高度な認識処理や運動予測処理が、必ずしも必要ではない可能性が示された。

このような利点を活かした関連研究として、これまでにロボットの制御に用いて動的な環境に実時間で適応できるシステムを構築したり [52-54, 62]、バーチャルリアリティ技術への応用も実現され [55]、高度な運動制御が可能となることが示されている。すなわち、システムの高速度を最大限に利用することで、これまでの視覚システムでは不可能だったダイナミックな制御と環境変動への適応が実現されている。

3.3.4.2 SPE-256 システムの問題点

一方、SPE-256 システムの構築とアプリケーション開発を通して明らかになった課題を挙げると以下のようになる。

まずロボットのなどの制御に必要な画像として、十分な解像度が得られていないことが挙げられる。しかしこれは前節で述べたように、最初の試作システムとして高速フィードバックの有用性を示すためには十分であり、当初から想定されていたことである。これに対する解決方法は既に述べた。

これに対して、ビジョンシステムの構成上、コントローラが実装されていないことは予想以上に問題となった。すなわち、このシステムでのインストラクションの送出と、処理後のパターンデータの取り込みは、DSP のパラレル IO を用いて行っている。そのためパラレル IO の動作速度が約 $4\mu\text{s}$ 、SPE 命令のインストラクションサイクルが $12.5\mu\text{s}$ と、システム全体の動作速度が制限されてしまう。この結果 1ms のフレームレートの中で可能な処理が約 80 インストラクションと制限され、イタレーションを含むような複雑な処理を行ったり、複数の処理を混在させることは困難となった。また、SPE-256 システムがバイナリ画像しか処理できなかったことに対して、多値の画像取得と処理が可能となった場合、PE でのビットシリアル演算が必要となり、その際にこの問題が一層顕著に現れることが指摘できる。

同様な問題として、ビジュアルフィードバックで必須となる特徴量抽出の際のボトルネックが挙げられる。特徴量の抽出についても、32bit 幅のパラレル IO を用いたスキャンによって画像のダンブを行い、さらに DSP による演算を利用しているため、重心検出の特徴量抽出に約 $500\mu\text{s}$ の時間を消費している。これは、現在のシステムで画素数が少ないためにこのような簡易的な方法が可能となっているが、次節で述べる CPV システムのように、ビジュアルフィードバックで必要となる 64×64 画素や 128×128 画素の画素数では、パラレル IO とスキャンを用いる方法では十分なバンド幅が得られないのは明らかである。加えて最初の節で述べたように、特徴量抽出に必要な XY 方向のデータの伝送についても全く実装されていない。

これらのシステム構成上の問題をまとめると、図 3.18 のようになる。

これらの問題については、CPV システムにおける列並列伝送の採用による高解像度の実現と、同時に専用のコントローラと特徴量抽出回路、XY データ伝送機構を導入することにより解決がなされている。これについては次節で詳しく述べる。

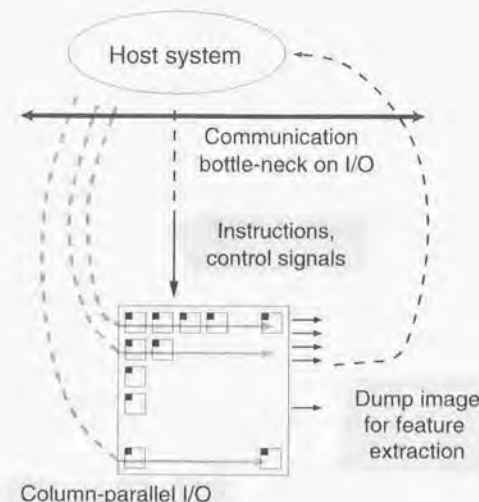


図 3.18 I/O bottle-neck on SPE-256 system

一方アクティブビジョン (AVS-I) の動作速度についても充分とは言えない。これは PD アレイのデバイスのサイズが大きいため、機構系とモータのサイズを小さくすることができないことに原因があり、結果として高速な動作が妨げられていた。すなわちアクティブビジョンシステムにおける高速動作の実現のためには、物理サイズがコンパクト性が重要になることが明らかになった。

これに対して次節で述べる AVS-II ではシステムの小型化が実現されており、機構部分の動特性の改善と、より高速なアクチュエータとの組み合わせによる高速な動作が実現されている。

3.4 列並列による超高速ビジョンシステム (CPV) の実現

3.4.1 概要

次にこの節では Column Parallel Vision (CPV) system と呼ぶ、列並列型の超並列・超高速ビジョンシステムについて説明する。

このアプローチでは SPE-256 システムと同様、センサと処理部をチップ化するのではなく、別々のチップ、ボードに実装している。そのうえで、それらの間での画像の伝送を列並列、行方向のスキャンにより実現しているこの方法を採用した結果、従来の CCD カメラの限界となっていた通信ボトルネックを回避することができ、同時に半導体の実装上の制約をうけることなく、多数の画素をもつシステムの構築が可能になった。

前節で述べた SPE-256 システムでは高速性については 1ms という要求を達成したもの、実装上の制約から、解像度については 16×16 画素、階調が白黒のバイナリのみと十分とは言えなかった。これに対してこの節で述べる CPV システムでは、画素数が 64 倍の 128×128 画素、解像度がバイナリから 8bit 階調と、充分量の情報が利用可能となり、同時にこれまでと同様に 1ms という超高速な処理を実現した。

また前節で視線の制御に用いた Active Vision System (AVS)-I では、ビジョンシステムの速さに対してアクチュエータの応答が充分ではないという問題があった。これに対して、今回高速応答とコンパクト化を実現した新しいシステム AVS-II を構築し、実際に充分な応答が得られることを示すことができた。

そこでこの節では、まず構築したシステムの詳細を説明し、次に一般的な画像処理を適用することにより、本システムにおける視覚処理の性能を確認する [63]。

最後に AVS-II に搭載してターゲットトラッキングの実験を行い、その性能の評価を行った [64]。

3.4.2 システム構成

ここで構築したシステムは、超高速画像処理を行う CPV システムと、実際に視線を移動させるためのアクティブビジョン (AVS-II)、これを制御するホスト DSP ネットワークからなる。

構築したシステムの全体図を図 3.19 に示す。

CPV システムは画像の取り込みと処理を行って画像特徴量を抽出し、これを制御のためのセンサ情報としてホスト DSP に出力する。また内部は、列並列出力 PD アレイ、超並列プロセッシングエレメント (PE) アレイ、コントローラの 3 つのモジュールからなる。

CPV システムの構成を図 3.20 に示す。以下の節では、システム各部の詳細について説明する。

3.4.2.1 列並列・超高速画像処理部

PD アレイ部は受光部と列並列 AD 変換器アレイからなる。受光部の画素数は 128×128 画素、AD 変換器アレイは 128 列並列で 8 ビットの分解能を持っている。出力される画像データは、選択された列ごとに、128 列並列にビットシリアルに PE アレイへ転送される。この結果、各フレームの全画素データは、約 1ms の間に読み出すことができる。

出力された画像データを対応する PE に送るには PD アレイの各列ごとに送られてくるデータを PE アレイ内のシフトレジスタで順次転送している。図 3.21 に PD アレイと PE アレイとの間でのデータの転送方法を示す。

画像処理部は、 128×128 個の PE アレイからなり、画像の各画素に対して完全並列な処理を実現している。PE の内部構成は S³PE アーキテクチャに基づいており、コンパクトな回路と処理の汎用性という相反する要求仕様を満たすために、SIMD 型制御、4 近傍接続、ビットシリアル ALU、24bit ローカルメモリ、メモリマップ I/O などを採用している [49]。PE が実際に行う処理命令は、後述するコントローラから供給される。

また先に述べたように、CPV システムではセンサデータの入力のために、PD アレイとのインタフェースとしてシフトレジスタを搭載している。図 3.21 に示されるように、シフトレジスタを通

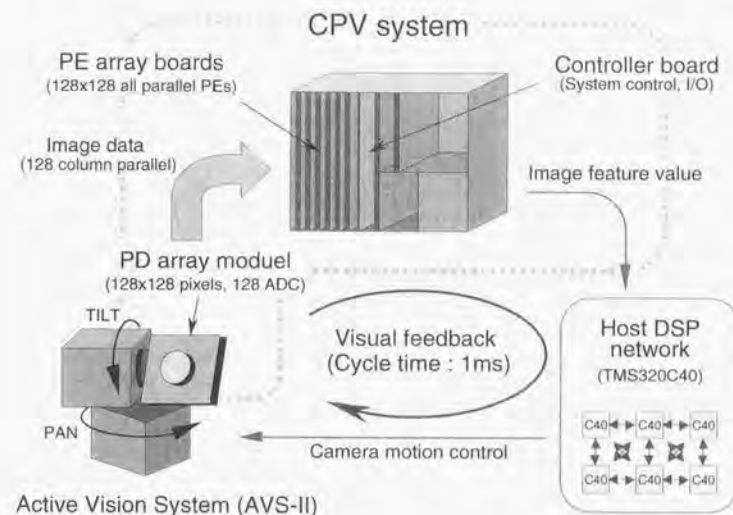


図 3.19 Overview of 1ms vision system

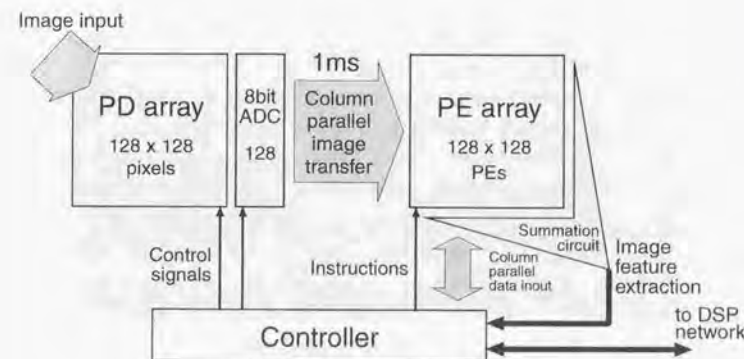


図 3.20 Column Parallel Vision (CPV) system

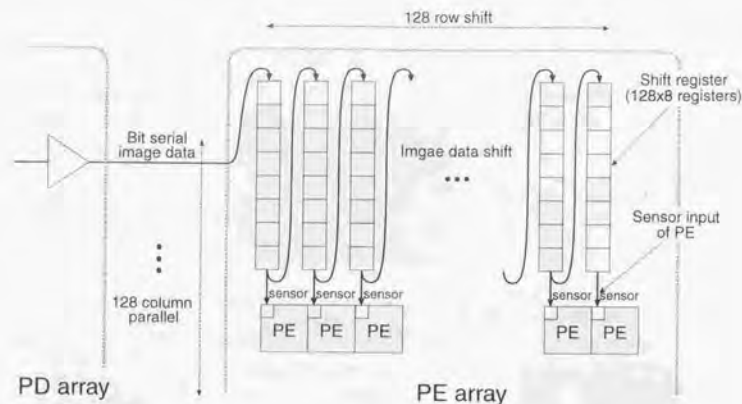


図 3.21 Column parallel data transfer from PD to PE

して送られてくるセンサデータは、PE のセンサ入力から取り込まれる。

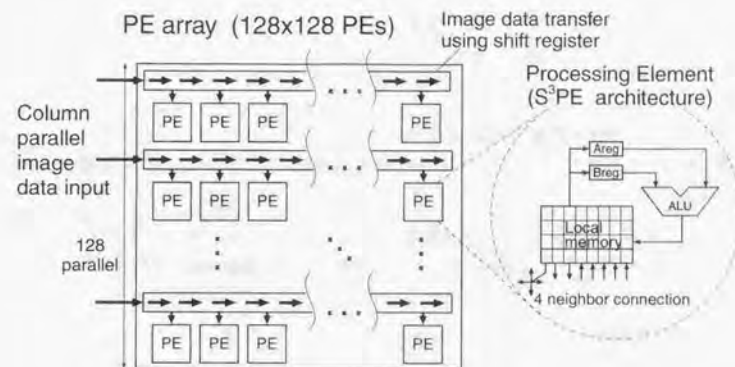
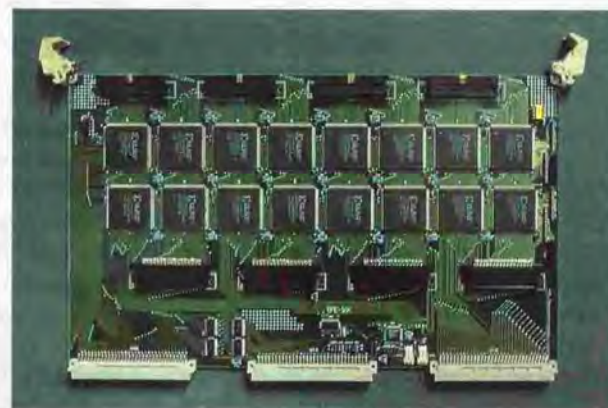
PE アレイの実装は、多数の FPGA (Field Programmable Gate Array) を用いて行われている。具体的には、 16×8 個の PE を内蔵したチップ (XILINX XC4044XL) を 128 個用いており、全体で 8 枚のボードで画像処理部を構成している。

これら画像処理部と、あとに述べるコントローラの開発に際しては、Cadence 社の Verilog HDL を用いた、ハードウェア記述言語による機能設計とシミュレーションを行い、XILINX 社の FPGA 開発 CAD、Foundation によりインプリメントを行っている。ボードについては、OrCAD 社 (現在 OrCAD 社は Cadence 社と合併している) の回路設計 CAD OrCAD Capture、基板 CAD の OrCAD Layout を用いて設計を行った。

図 3.22 に PE アレイの構成と S^3 PE アーキテクチャの構成を示す。また図 3.23 に作成した PE ボードの写真を示す。ボード上に 16 個の FPGA が搭載されているのが分かる。

このように、システムのメインのデバイスとして使用した FPGA では、何度でも機能の再書き込みが可能となっており、設計上のバグや、仕様変更にもなつて再設計を行い、設計と性能評価についてのインタレーショを行うことが可能となった。その結果、これまでにないアーキテクチャに基づく新しいシステムを構築する際に、完成度の高い試作機の実現が可能となった。一方、システムで主に使用している FPGA チップ (XILINX XL シリーズ) については、その後継互換チップが既にリリースされ、数世代の刷新が行われている。これらのチップを使用すれば、現在のシステムにはほとんど手を加えることなく、動作速度などについて数段の性能向上が期待できる。

現在のシステムでは、全体の動作速度として処理の 1 ステップが 330ns で、ほぼ全ての機能がエラーなく実行可能なことを確認している。

図 3.22 S^3 PE architecture with column parallel data transfer図 3.23 PE board (including 16×128 PE)

3.4.2.2 コントローラ

並列画像処理システムを構成する場合、システム全体の動作を制御し、かつ外部システムとのインタフェースを行うコントローラの役割が重要になる。

本節の最初に挙げたビジョンシステムの目的に対応して、このコントローラで実現された機能は以下になる。

- ・ PE アレイからの画像特徴量の抽出、演算：

専用総和演算回路により、PE アレイでの処理結果を高速に抽出し、さらに必要な各種演算を行う。
- ・ PE アレイとの行・列並列データ入出力機能：

PE アレイに対して、X,Y 座標のブロードキャスト、近傍接続を用いてのデータの入出力を行う。
- ・ S³PE の SIMD 制御：

PE アレイに対してインストラクションを送出し、SIMD でのプログラム制御を実現する。
- ・ ユーザプログラムによるプログラム制御：

メインメモリにダウンロードしたユーザプログラムにより、システムの全動作のプログラム制御を実現する。
- ・ 外部システムとのインタフェース：

共有メモリ方式により、外部ホストシステムとのデータ入出力を必要なレートで実行する。
- ・ 各モジュール間の同期管理、制御：

システムを構成する各モジュールの制御を行い、モジュール間での同期を調整する。

コントローラの構成を図 3.24 に示す。

システムのインストラクションセットアーキテクチャの大きな特徴として、全インストラクションを SPE 命令とコントローラ命令の 2 種に分けていることが挙げられる。これはビジョンシステムとして並列画像処理 PE とコントローラの動作が、密接に連携して行われることが理由であり、SPE 命令によって PE アレイにおける SIMD 並列画像処理を行う間はコントローラではプログラム制御のみを行って演算処理を行わず、逆にコントローラでの処理を行う間は、PE アレイの SIMD 命令として先に述べた NOP レジスタの命令が送られる。現在までのところ、このような構成をとることによる処理の冗長さの問題は、特に顕在化していない。

一方コントローラのコアアーキテクチャとしては、図に示されるように、システムバスとして 32bit のデータバスを持ち、ALU での演算として、加減算、キャリーボローつき加減算、ビットごとの論理演算と左右シフトが可能である。また、プログラム制御機能として、無条件分岐、Zero、Carry、Overflow、Sign の 4 種と、これらの真と偽の両方の条件での分岐が可能となっている。レジスタには A,B ふたつのオペランドレジスタ、条件コードレジスタ (CC)、インストラクションレジスタ (IR)、プログラムカウンタ (PC)、メモリアドレスカウンタ (MAR)、またコントローラが処理を行う間に SPE が行うべき命令を格納する NOP レジスタ (NOP)、自動アクリメントを行うタイマレジスタ (TIMER)

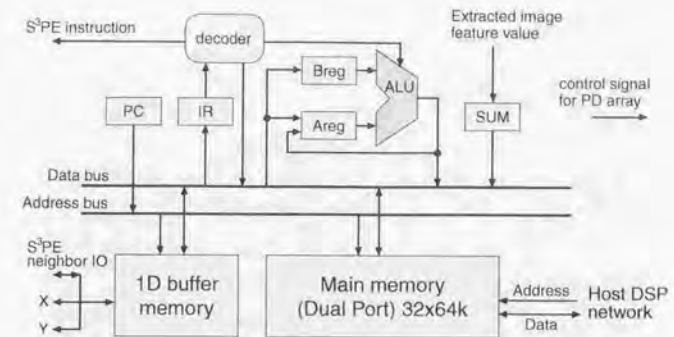


図 3.24 Architecture of the controller

が実装され、現在はまだ動作確認がなされていないものの、将来の拡張として、各種の割り込みについても仕様が規定されている。

メインメモリは、その全てが上で述べた DSP ホスト側との共有メモリとなっており、非常に柔軟なシステム制御が可能となっている。具体的なデバイスとしては IDT 社の大容量高速 Dual Port SRAM, IDT 7027L20 (16×32K bit, アクセスタイム 20ns) を 4 個使用している。またホスト側との通信データ線幅を極力押さえるために、Dual Port メモリにおけるコントローラ側のデータアクセスが 32×64K bit と 32bit ワードなのに対して、DSP 側からは 16×128K bit と、16bit ワードでのアクセスを行う構成となっている。これによって、ホスト DSP と CPV システムの物理的な通信、IO の制約が緩和され、ビジョンシステムのモジュールとしての自立性を高める要因となっている。

またビジュアルフィードバックのためのビジョンシステムに特有となる、画像特徴量抽出のための総和演算回路については、階層構造を用いてビットシリアルに遅れなく総和演算を行う専用回路 [60] を実装しており、先に述べた SPE の PE を実装した FPGA、及びこれを搭載した PE ボード上の CPLD (XILINX 社 XC9536L)、またこれらの 8 枚の PE ボードを統括するコントローラ FPGA にそれぞれ、分散して加算演算モジュールが実装されている。最終的な出力は、メインメモリとは別の拡張メモリアドレス (Sumout) 上にマップされ、通常のメモリアクセスと同様にして、総和演算結果を取り出すことができる。

一方、XY 方向の 1 次元データアクセスを実現する 1 次元メモリについては、図 3.25 に示ように 1 次元メモリと呼ぶバッファメモリを用いてコントローラの 32bit 幅のデータバスと 128bit 並列の PE アレイの入出力とを接続している。コントローラ側からのアクセスは、総和演算器と同様に一定のメモリ空間にマップされ、一方で専用のコントローラ命令 (select1D) により、PE アレイからアクセスするメモリのアドレスを指定することができ、並列処理 PE 側で、それぞれの X, Y, left, neighbor などの機能レジスタにアクセスすることで、これらのアドレスへの入出力を実現している。

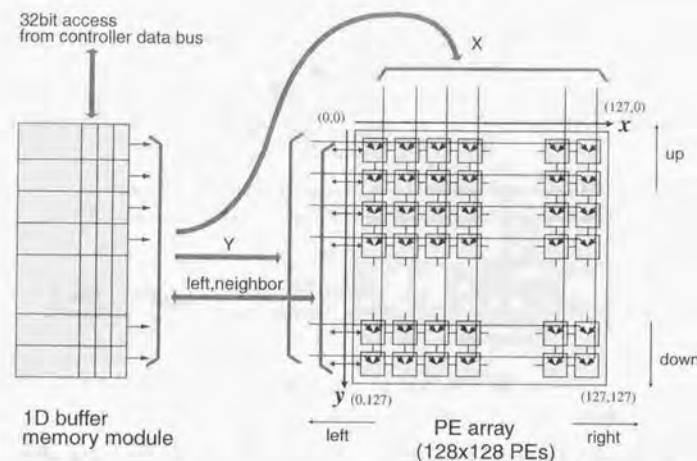


図 3.25 Row and column parallel data inout and broadcast

最後の AD 変換器を内蔵する PD モジュールとの同期通信制御については、全て完全にハードウェアでの隠蔽がなされ、PD のための各種のクロック、制御信号が自動で生成されている。その際に、ユーザからはシステムクロックと PD クロックとの比を示すクロック倍率設定 (clkdev)、AD 変換器の階調値 (depth)、PD アレイの読み出し開始セクション (secstrt)、読み出し終了セクション (secend) 等がして可能で、これらを調整することで、可能な PD クロックスピード (= センサフレームレート) やグレーレベル、読み出し領域について、任意に帰ることができる。これらの設定値についても全て拡張メモリアドレス上にマップされており、また、SPE 命令の専用のオプション指定により、プログラムでベルでの PD との同期をとることができる。

以上のコントローラと 1 次元メモリは合わせて FPGA (XILINX XC4044XL) 2 個を用いて 1 ボード上に実装されている。図 3.26 に作成したボードの写真を示す。コントローラのチップ、システム、ボードの開発は、PE 部と同様にハードウェア記述言語と各社 CAD を用いて行った。

以上のような構成により、画像を取り込んでアプリケーションに応じた処理を行い、画像特徴量を抽出して外部システム (ホスト DSP) に出力するというビジュアルフィードバックの一連の処理が、目標のサイクルタイム (1ms) に対してボトルネックなく実現されている。

3.4.2.3 アクティブビジョンシステム (AVS-II)

AVS-II は視線を高速移動させるためのアクチュエータ部である。可動部には前述の PD アレイが搭載されており、直交するパン、チルトの 2 軸の自由度を持っている。各軸には AC サーボモータを



図 3.26 Controller board

モータ	AC サーボモータ (ダイレクトドライブ)
制御自由度	2 自由度 (チルト・パン)
モータ出力	100W × 2
可動範囲	280(チルト), 340(パン)
外寸	D150, W150, H170 mm
重量	3.5Kg

表 3.3 Specifications of AVS-II hardware

使用し、ギヤを用いず、ダイレクトドライブで使用している。表 3.3 に AVS-II のハードウェア仕様を示す。

全体として、高出力、コンパクト化を実現しており、3.3.2.3 節で述べたアクティブビジョンシステムと比較して高い応答性能が実現されている。

アクティブビジョンの運動制御はホスト DSP 上で行っている。ホストでは TI 社製、並列処理用 DSP (TMS320C40) を使用した DSP ネットワークを構成しており、IO と処理の負荷を分散して、1ms のサイクルタイムに対してボトルネックのないシステムを実現している [54]。

具体的な制御方式としては、ビジュアルフィードバックによるトルク制御を行っており、視線の高速かつ動的な制御を実現している。AVS-II の制御ブロック図を図 3.27 に示す。

また、以上で説明したシステムの写真を図 3.28 に示す。

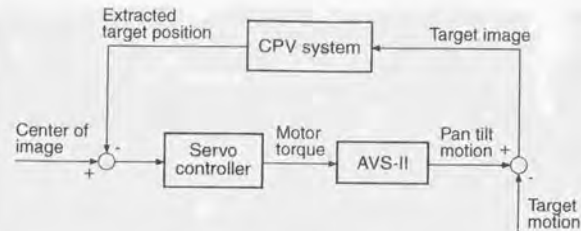


図 3.27 Block diagram of AVS-II control

3.4.3 実験

3.4.3.1 PD モジュールの感度評価

PD モジュールの感度を評価するために、各フレームレートでのグレースケール画像を比較した。

データは階調を最小 (3bit) にした場合と、最大 (8bit) にした場合の両方の画像を測定した。PD モジュールに供給するクロックについては、8bit 階調での動作時に 3bit 動作時のちょうど 2 倍のクロックを供給することになる。すなわち、1.28ms/frame で動作させたとき、8bit 動作時は 1MHz、3bit 動作時は 500KHz のクロック周波数となる。

照明については、フレームレートを高にしたときに室内光で感度が十分でなくなるため、別に照明を用意した。照明ありの条件と、照明なしの条件の比較は、各ビット階調について、12.8ms/frame で動作させたときの両者の画像で比較できる。なお 8bit 階調で 1.28ms/frame の動作時には、強い照明をあてた場合でも十分なセンサ出力が得られなかった。

以上の感度評価の結果画像を図 3.29、図 3.30 に示す。

この結果から、1.28ms での 3bit 階調画像を見ると、取得された画像の感度はあまり十分とは言えないが、高解像度でのパターンの識別と利用は可能であると言える。また 8bit 階調での動作では、数 ms 程度のフレームレートで、かなりの階調を含む画像の取得が可能であることがわかる。また数 10ms 程度のフレームレートでは、室内光のもとでの画像の取得が容易に可能である。

これをふまえて、以下の実験では高速動作については、3bit 階調での 1.28ms 動作を利用し、多階調画像処理では十分な階調を得るために、フレームレートを落として実験を行っている。

またこの実験を行った環境について、室内光のもとで写真に撮影したものを、図 3.31 に示す。

3.4.3.2 多階調フィルタリング画像処理

本システムの処理能力を確かめるために、取り込んだ多階調画像に対して、一般的に用いられている画像処理を適用した。

最初に多階調のフィルタリング処理の実験結果を示す。実験は、多階調 (8bit, 256 階調) の画像に対して S³PE で並列処理を行い、処理結果をダンプして表示している。この実験では画像処理の結果

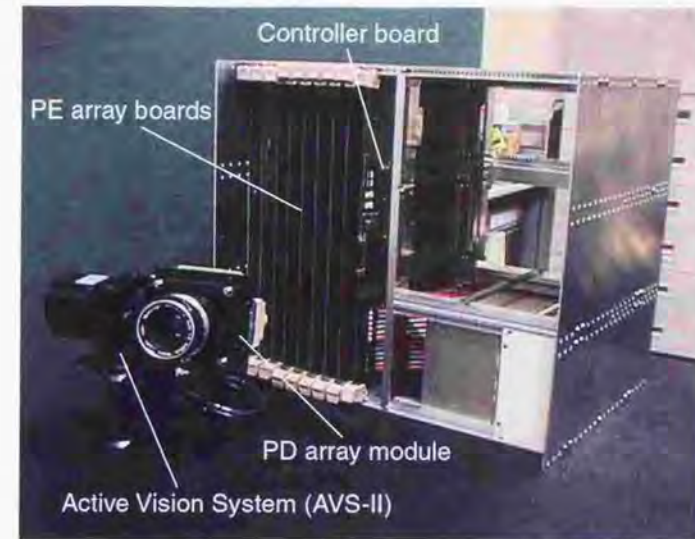


図 3.28 CPV system with AVS-II

を評価しやすくするために、PD でノイズの少ない画像が得られるように画像の取り込み速度を落としており、12.8ms/frame で画像の取り込みを行っている。

画像処理は全て完全並列に行われるため、非常に少ないステップ数で処理が完了する。実験を行った画像処理について、8ビット階調の画像に対するステップ数と、現システムでの最高速度 (3Msteps/sec) で実行した場合の処理時間を表 3.4 に示す。

実験の各処理の処理内容は以下の通りである。ただし、 $I(x, y)$ は座標 (x, y) での画素での入出力の輝度値を表わしている。また各処理中で、データは随時 8bit 長に値を丸めている。

- 2 近傍はかし処理：

$$I(x, y) = \{I(x-1, y) + 2I(x, y) + I(x, y-1)\} / 4$$

- 7 近傍はかし処理：

repeat 7 times:

$$I(x, y) = \{I(x-1, y) + 2I(x, y) + I(x, y-1)\} / 4$$

- 2 近傍エンボス処理：

$$I(x, y) = \{2I(x, y) - I(x-1, y) - I(x, y-1) + I_{const}\} / 4$$

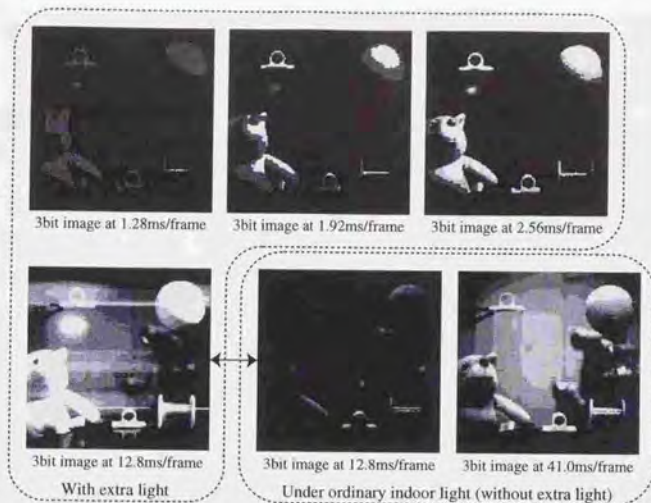


図 3.29 3bit gray level images at different gray levels and framerates

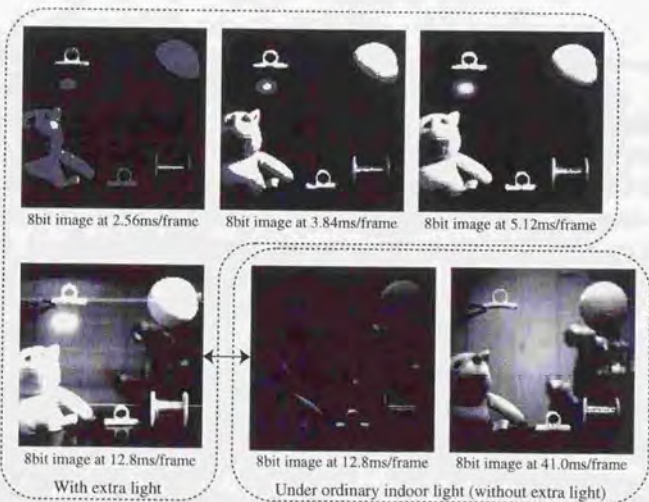


図 3.30 8bit gray level images at different gray levels and framerates



図 3.31 Environment of experiments of pattern matching

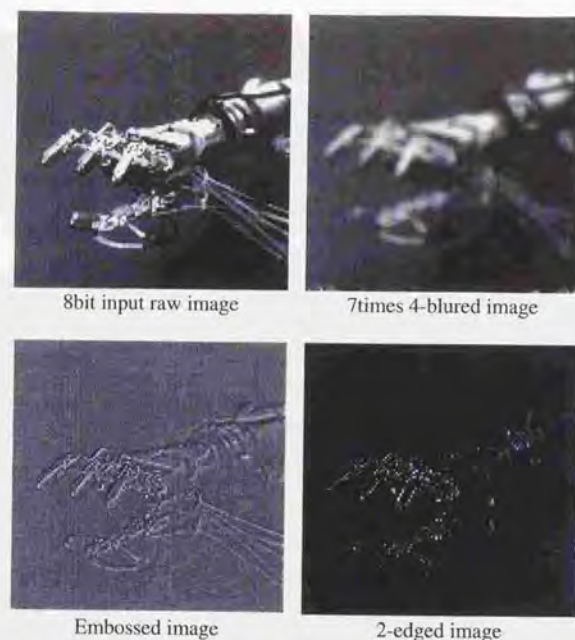


図 3.32 Results of image processing executed by CPV system

- 2 近傍エッジ処理:

$$I(x, y) = \{|I(x, y) - I(x-1, y)| + |I(x, y) - I(x, y-1)|\} / 2$$

ただし $I_{const} = 2 \times 256 = 512$.

また図 3.32 に実際に取り込んだ画像と処理した結果を示す。

3.4.3.3 ヒストグラムの取得とスレッシュホルディング

多値画像に対する認識処理において、2 値化のためのスレッシュホルディングと、これに先立つヒストグラムの取得は重要な役割を担う。このヒストグラムの計算と、スレッシュホルディングの処理を CPV システムで行った結果を以下に示す。

ヒストグラム、及び累積ヒストグラムの計算は、全階調値に対してスキャンを行い、該当するがその数を総和演算器でカウントする方法で計算している。集計結果を、図 3.34 に示す。

それぞれのグラフから、ヒストグラムと累積ヒストグラムの各集計値が正しく求められているこ

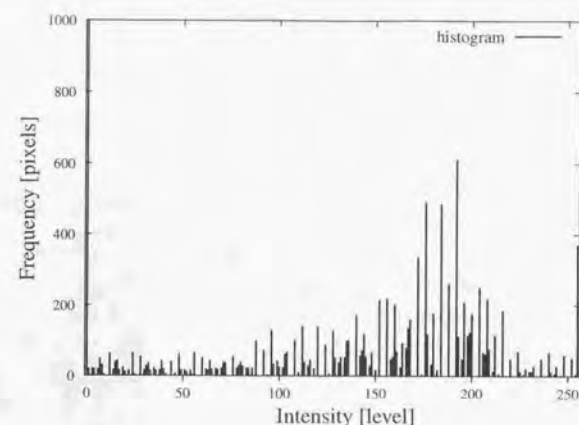


図 3.33 Histogram of 8bit gray scale image (calculated by CPV system)

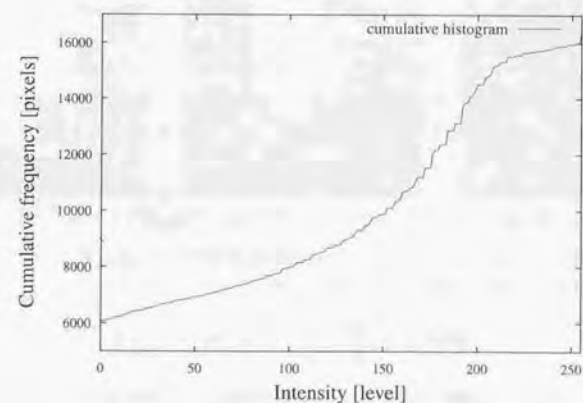


図 3.34 Cumulative histogram of 8bit gray scale image (calculated by CPV system)

画像処理例	ステップ数	処理時間
センサ情報取り込み	8	26.6 μ s
2 近傍ぼかし処理	58	19.3 μ s
7 近傍ぼかし処理	406	135.3 μ s
2 近傍エンボス処理	59	19.6 μ s
2 近傍エッジ処理	59	19.6 μ s

表 3.4 Image processing time in CPV system

とが分かる。ただし、ヒストグラムのグラフからは、階調値の分布が、まだ理想的にはばらついているとは言えず、センサ取得の部分でなんらかの問題があることが示されている。

一方、現在の方法では全階調値に対して、一つづつ画素の頻度数を集計しているため、実時間で処理は困難であるが、今後、次に述べるスレッショルディングを目的とする場合は、スレッショルドレベルに対して何らかの仮定を設けてトラッキングを行うことにより、2 値化の処理において、動的に環境に適応する実時間処理も可能である。

またここで集計した値にもとづいて、累積ヒストグラムの上位 50%, 25%, 5% の画素をそれぞれ 1 とし、残りの下位の画素を 0 とするスレッショルディングを行った結果を図?? に示す。このように、CPV システムでは環境に応じた自律的な処理によって、各アプリケーションに対して適切なスレッショルドレベルを選択し、2 値化を行うことが可能である。

3.4.3.4 テンプレートマッチング

次にテンプレートマッチングの実験を行った。最初の実験では、固定されて 1 枚だけ取得された画像のなかで、画像中に含まれるある特定のパターンに対して、全画面での探索と評価を行った。その目的としては、実際に画像中から抜き出されたテンプレート自身とのマッチングや、類似のパターンとのマッチングが可能かどうかを確かめることが挙げられる。

まず 8bit 階調の画像とテンプレートパターンとのマッチング結果を示す。マッチングの計算方法としては、最も簡単なものの一つとしてよく用いられている、現画像とテンプレートとの差分の絶対値の総和で評価している。すなわち、座標 (x, y) 点での画素値を $I(x, y)$ 、テンプレートを $J(x, y)$ 、テンプレート領域を $N \times M$ とすると、以下の式になる。

$$Eg(u, v) = \sum_{N, M} |I(x+u, y+v) - J(x, y)| \quad (3.3)$$

この結果、マッチした点は周囲に対して十分小さな谷を形成し、自分自身とマッチしたときは、値が 0 となる。

上式にしたがって、S³PE にてテンプレートと画像の差分と絶対値の取得を計算し、総和演算器にて総和の演算を行いながら、テンプレートを移動させて、全画面分スキャンした。実験において取



8bit input raw image



Thresholded at 50% intensity



Thresholded at 25% intensity



Thresholded at 5% intensity

図 3.35 Result of thresholding 8bit gray scale image

得られた現画像、抜き出されたテンプレートパターン、マッチング結果を画像平面上に輝度値で表わしたものを、図 3.36 に並べて示す。

また、最後のマッチング結果をより見やすいように 3D グラフにしたものが、図 3.37 である。ただしマッチした点の位置をわかりやすく、評価値を示す z 軸を反転させている。これより自分自身とは正確にマッチして値が 0 となり、また画面の下右よりの場所に、類似の形状とマッチして、反転した軸に対して鋭いピークを示しているのが分かる。

次にバイナリ画像とテンプレートパターンとのマッチング結果を示す。バイナリのマッチング計算には、多値画像で用いた差分のかわりに、より簡便でかつ同時に正規化がなされる論理演算 \oplus が適している。

先と同様に座標 (x, y) 点での画素値を $I(x, y)$ 、テンプレートを $J(x, y)$ 、(ただしどちらも 2 値の論理値)、テンプレート領域を $N \times M$ とすると、以下の式になる。

$$Eb(u, v) = \sum_{N, M} |I(x+u, y+v) \oplus J(x, y)| \quad (3.4)$$

この結果、マッチした点は周囲に対して十分大きなピークを形成し、自分自身とマッチしたときは、値がテンプレートの面積、 $N \times M$ に一致する。

上式にしたがって、 S^3PE にてテンプレートと画像の差分と絶対値の取得を計算し、総和演算器にて総和の演算を行いながら、テンプレートを移動させて、全画面分のスキャンを行った。

取得された現画像、2 値化された画像、抜き出されたテンプレートパターン、マッチング結果を画像平面上に輝度値で表わしたものを順に示したのが、図 3.36 である。

また、最後のマッチング結果をより見やすいように 3D グラフにしたものが、図 3.37 である。これより自分自身とは正確にマッチして鋭いピークを生じている。ただしそのときの値は、データの保存領域を節約するために $1/4$ 倍としているので、 $N = 23, M = 12$ に対して、 $N \times M/4 = 23 \times 12/4 = 135$ となっている。また画面の多階調でのマッチングと同様に、右下よりの場所にある、類似の形状とマッチして鋭いピークを示しているのが分かる。

以上により、多値画像、バイナリ画像の両方について、十分に精度でマッチングが可能であることが示された。

3.4.3.5 テンプレートマッチングによるトラッキング

次に、上記のテンプレートマッチングを用いて、画面内でのターゲットトラッキングを行った。1ms の高速なフレームレートを得るために、画像は 3bit で取得したが、マッチングの処理についてはフレーム時間内での処理に十分余裕があることを確かめるために、多階調 (8bit 階調) でのマッチングを行った。

前節のテンプレートマッチングの実験では、全画面のスキャンを行うために実時間での処理はできないが、本論文の最初に述べたように、認識処理を高速なフレームレートで行うことにより、対象の動きが非常に小さくなり、スキャンの必要性がほとんどなくなる。

今回の実験では、現在テンプレートがある位置から、1 画素分だけ周囲を探索し、最もマッチした点を次のテンプレートの位置とした。これは、石井ら [56] による、セルフウィンドウ法と全く同様

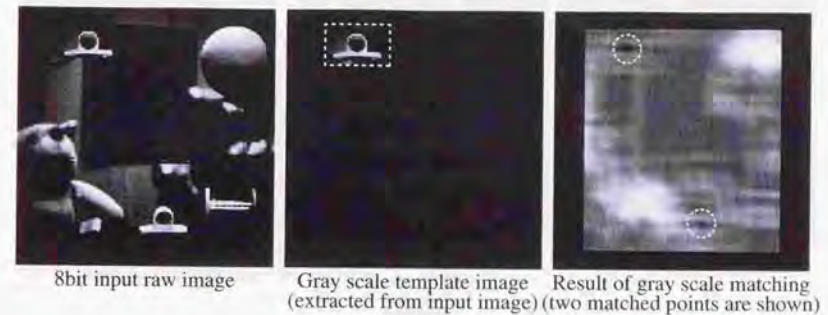


図 3.36 8bit pattern matching executed by CPV system

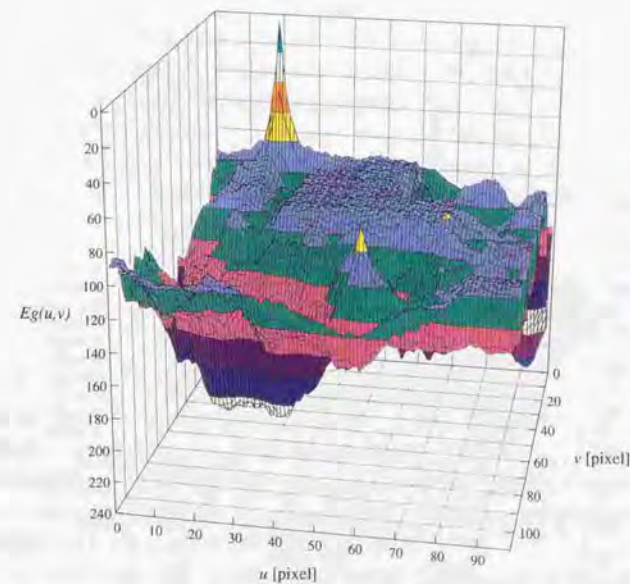


図 3.37 Result of 8bit pattern matching

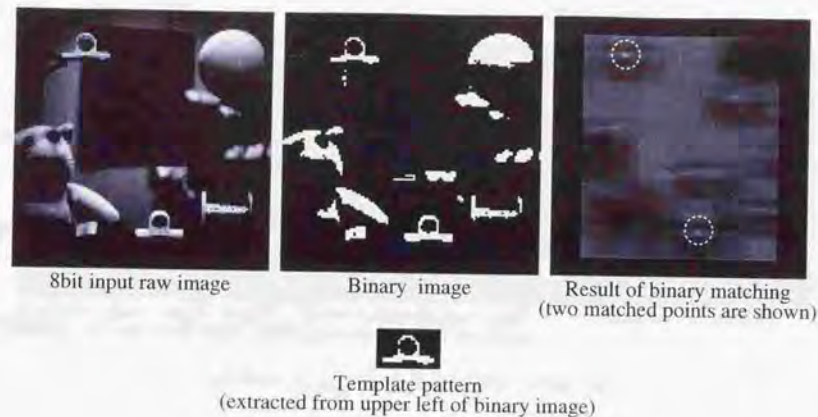


図 3.38 Binary pattern matching executed by CPV system

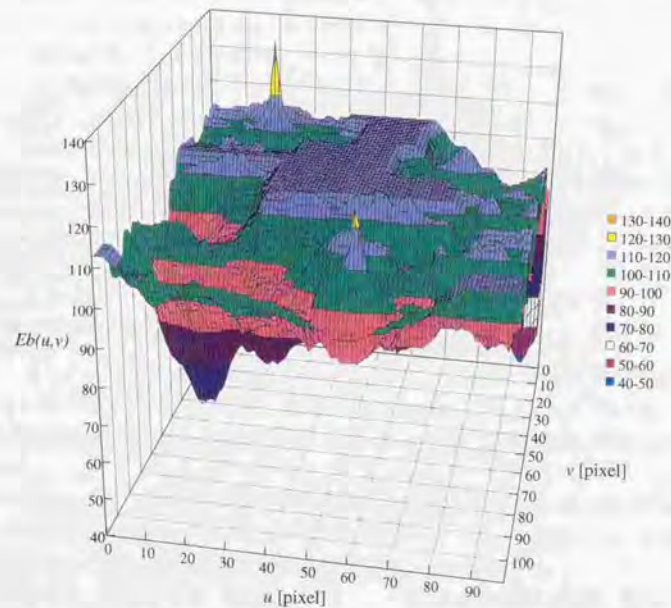


図 3.39 Results of binary pattern matching

処理内容	ステップ数	処理時間
センサ情報取り込み	8	2.7 μ s
テンプレートの移動 $\times 20(\max)$	360	120.0 μ s
差分 $\times 9$ (for scan)	144	48.0 μ s
絶対値 $\times 9$	72	24.0 μ s
評価値の集計 $\times 9$	207	69.0 μ s
その他 $\times 9$	180	60.0 μ s
合計ステップ数	971	323.7 μ s

表 3.5 Image processing time for template matching

の方法である。すなわち視覚処理のフレームレートが十分高速になることにより、複雑な処理や運動の予測が必要なくなり、1単位分の移動だけを考慮すればよくなるという考え方である。

実際の処理の結果とトラッキング結果を示す。図 3.40には、実際にトラッキング中の処理中の画像をダンプしたものを示す。テンプレートの位置だけ全ビットが反転して強調されている。

またトラッキング結果の軌道は、図 3.41、図 3.42に示す。図 3.41では、XY 画面上に軌道を示し、フレームレートの 1.28ms ごと軌跡を四角い点で示している。これより、X,Y の両方向成分を含む運動について、共に 1 画素以上の点の急激な移動がなく十分トラッキングできていることが分かる。

このことは X 方向の軌道、Y 方向の軌道をそれぞれ時間軸方向にプロットした、図 3.42で、さらに明らかに示される。すなわち、いずれの方向に対する高速な移動でも 1.28ms のフレームレートに対しては十分ゆっくり移動し、フレームレート内での移動距離は常に確実に 1 以内に収まっている。

この結果、高速かつ予測不可能な不規則な動きに対しても、テンプレートマッチングの認識処理を行いながら、十分追従可能となっている。

また、フレームサイクルごとに、テンプレートマッチングによるトラッキングの各処理に必要な処理時間を表 3.5 に示す。

3.4.3.6 1ms ターゲットトラッキング

最後にアクティブビジョン装置を含めた全体システムの、ビジュアルサーボシステムとしての性能を評価するために、ターゲットトラッキングの実験を行った。

本実験ではシステムの高速度性が充分であることを示すため、1ms のサイクルタイムでのビジュアルフィードバックの実現を目標とした。具体的にはターゲットトラッキングを行うのに十分な 3bit 階調の画像データを 1.28ms で PE アレイに転送している。

フレームごとの画像処理としては、前処理としてノイズ除去のためのフィルタ処理と 2 値化を行っている。また対象認識のアルゴリズムとして、画像の高速フィードバックの特徴を利用するセルフウィンドウ法 [56] を用いている。また最終的に出力する画像特徴量として、認識したトラッキング対象のパターンの重心位置を抽出している。



図 3.40 Captured image at tracking

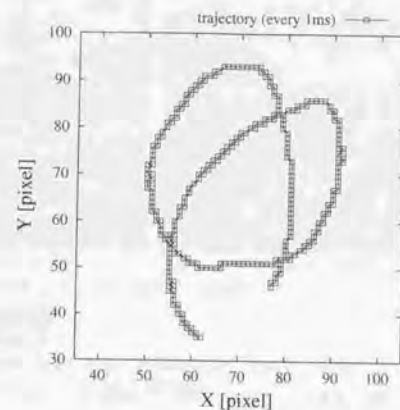


図 3.41 Trajectory of tracking by template matching

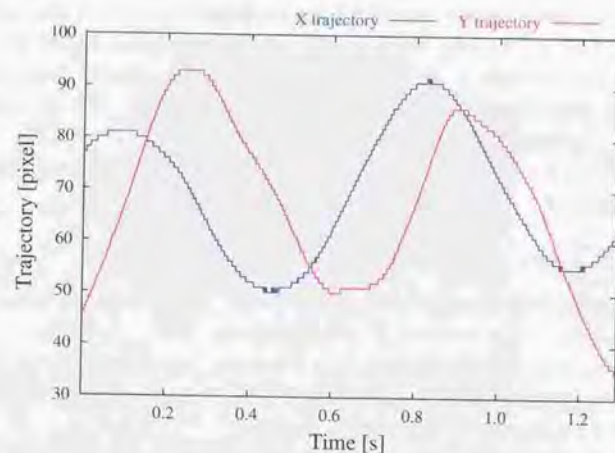


図 3.42 Trajectories of each X and Y of tracking

処理内容	ステップ数	処理時間
センサ情報取り込み	3	1.0 μ s
フィルタ処理	41	13.6 μ s
セルフウィンドウ法	6	2.0 μ s
重心位置計算	126	42.0 μ s
合計ステップ数	176	58.6 μ s

表 3.6 Processing time for target tracking

それぞれの処理にかかるステップ数と時間を図 3.6 に示す。

また実際にターゲットトラッキングを行った結果を連続写真で図 3.43 に示す。手に持った棒の先の白い球をターゲットとしてトラッキングを行っている。ターゲットが不規則、かつ非常に高速に動いているにもかかわらず、充分追従できているのがわかる。

3.4.3.7 ビジュアルフィードバックの周波数応答解析

次にトラッキング動作の応答性能を確かめるために、ビジュアルフィードバック制御時の周波数応答を測定した。

実験は固定したターゲットに対して画像面上でサイン波状の目標軌道を与え、これに追従させて応答を記録した。さらにこれを異なる周波数で行って応答を調べた。制御アルゴリズムとしては各軸ごとに固定パラメータの位相進み制御を行っている。

測定結果を図 3.44 に示す。

パンとチルトの両軸とも 10Hz 以上の目標の高速な変化に追従できていることがわかる。これによって、本システムが、従来にない非常に高速な応答をもつビジュアルサーボを実現していることがわかる。

3.4.4 考察

今回構築した CPV システムについて、CCD カメラを用いた従来のビジョンシステム、1 チップ集積化を目指したビジョンチップシステム、SPE-256 システムに対する比較を行い、列並列の画像伝送と S³PE アーキテクチャに基づく完全並列処理を組み合わせることで、これまでにない高解像度で、ビジュアルサーボなどのアプリケーションに必要とされる 1ms の高速性が実現されることを述べた。また構築したシステムの詳細を説明し、目的とするアプリケーションを実現するための様々な機能を備えていること示した。

また、構築した実際のシステムを用いて実験を行った。一つは汎用の画像処理を適用し、多階調の画像に対する処理が超並列・超高速に実行されることを確かめた。また画像センサをアクティブビジョンに搭載してターゲットトラッキングを行い、高い応答性能を持つ新しいアクチュエータ (AVS-II) と組み合わせることで、これまでにない高速なトラッキング性能が得られたことを示した。

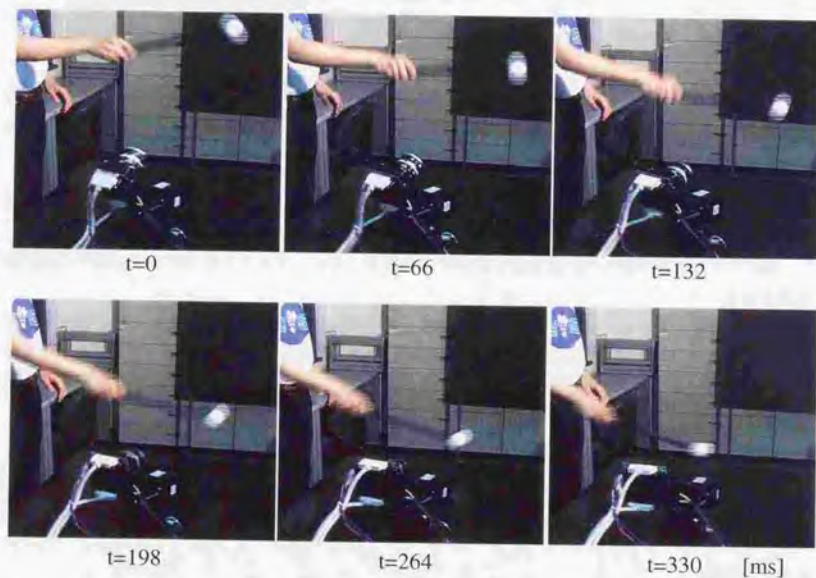


図 3.43 Experiments of 1ms target tracking

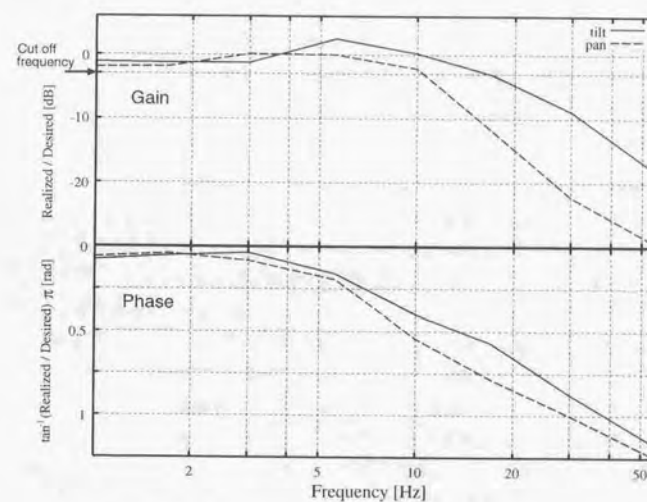


図 3.44 Frequency response of target tracking

第4章

ビジュアルインピーダンス

この章では動的なビジュアルフィードバック制御の新しい方法として、ビジュアルインピーダンスを提案する。最初にその概念と理論を示し、その後実験により有効性を示す。

4.1 はじめに

不確定で変化する環境でロボットを動作させるためには、静的な状態のシーケンスとしてタスクを与えるだけでなく、動的な状態そのものをタスクとして与える場合がある [65, 66]。その典型的な例としては、マニピュレータの仮想コンプライアンス制御 [10, 11, 67] が挙げられる。この方法では、接触作業を伴うタスクなどでモデルの不確定性の影響を軽減するために、マニピュレータの従うべきコンプライアンス特性を仮想的に設定し、制御によりこれを望ましい動特性を実現している。

これに対して、2.4.3節でも述べたように、従来のビジュアルフィードバック制御ではシステムのダイナミクスについてほとんど考慮されていなかった。例えば Castano and Hutchinson [23] はビジュアルコンプライアンスという概念でビジュアルサーボと位置制御のハイブリッド制御を提案しており、細田 [39] らはさらに未知環境で適応的に非干渉化しながら視覚と力のハイブリッド制御を行う方法を提案している。しかしこれらの方法は視覚と位置、力について別々にサーボを適用するものであり、ダイナミクスについては全く考慮されていない。

これは与えられた目標軌道などに対して、システムの過渡応答についての配慮がなされていないという問題だけではない。先に挙げた仮想コンプライアンス制御に見られるように、ダイナミクスを含む形で与えられるタスクについてもほとんど例がなかった。特に画像特徴量をそのまま制御量として利用するビジュアルサーボでは、タスクとして画像特徴量が満たすべき動力学方程式を与えることが考えることができる。ところがこれまでそのような制御法を提案した例はなかった。

一方、辻 [68] らは非接触型インピーダンスを提案し、接触を伴わない状況での仮想インピーダンス制御によりマニピュレータのダイナミクスの制御を実現している。しかしこの方法ではセンサとして PSD を用いており、広義の意味では視覚を用いたものとも言えるが、画像情報で最も重要となるパターンの情報を利用することができず、すなわちフィードバックする特徴量として対象の重心位置の情報しか利用することができず、結果として適用範囲は衝突の回避などに限られている [69]。

このように、従来、視覚情報を利用したマニピュレータのダイナミクスの制御がなされてこなかっ

たのは、ひとつにはビジョンシステムが適切に設計されていなかったことに原因があると考えられる。今後は環境の動的な変化や、環境とロボットとの動的なインタラクションを視覚により認識し、ダイナミクスを直接指定するようなタスクを、ビジュアルフィードバックにより実現できるようにするべきである。

またもう一点重要なのは、他のセンサ制御システムとの融合である。例えばビジョンシステムのダイナミクスを論じた橋本などの研究でも、他のセンサシステムとの融合は考えられていない。

そこでこの章では、ビジュアルフィードバック制御において、タスクそのものにダイナミクスを含む例として、ビジュアルインピーダンスを提案する。

以下の節では、ビジュアルインピーダンスの概念と詳細を説明する。その後、SPE-256 システムとロボットマニピュレータを用いて実験を行ったので、その結果を示す [52, 53, 70, 71]。

4.2 ビジュアルインピーダンス

本節では提案するビジュアルインピーダンスについて、その目的と概念を述べる。さらにこれをマニピュレータの制御に応用するための具体的な方法を示す。

4.2.1 概要

これまでに力センサやトルクセンサを利用してマニピュレータに仮想的なコンプライアンスを持たせる制御法が提案されている [67]。例えば接触作業を伴うタスクをロボットに行わせることを考える。この場合、対象の形状モデルやロボットモーションの少しの誤差が、目標とするタスクに対して重大なエラーを引き起こすことがある。すなわち環境の不確定性や環境との力学的なインタラクションについて不連続性や強い非線型性がある場合、目標軌道だけで与えてマニピュレータを制御するのはほとんど不可能だと考えられる。

これに対して仮想コンプライアンス制御では、環境との力学的なインタラクションを積極的に利用する。すなわち力センサやトルクセンサを用いたタスクレベルのセンサフィードバックを用いることで、不確定で変化する環境に対してもそれに応じたならい動作を実現することができる。具体的にはマニピュレータの目標軌道を与えるだけでなく、動的な状態、すなわち仮想的なコンプライアンス特性を動力学方程式の形で目標として与え、制御によって望ましい動特性を実現する。これによって環境に適応した柔軟な動作を実現するものである。

同様な観点から、ここでは視覚にもとづく仮想的な接触を提案し、これに基づくビジュアルインピーダンス制御を提案する。

この方法ではまず視覚で観測された物体の像に対してリアルタイムで仮想表面を設定する。さらにその仮想表面の内部にあらかじめ適切なコンプライアンス特性を与えておく。その上でマニピュレータが設定された仮想表面に触れると、与えられたコンプライアンス特性に従ってマニピュレータに加える仮想的な反力を計算し、あたかも実際に接触したかのようなマニピュレータの動作を制御により実現する。この様子を図 4.1 に示す。

ビジュアルインピーダンスを用いることで、実際にマニピュレータが対象に接触しなくても、あ

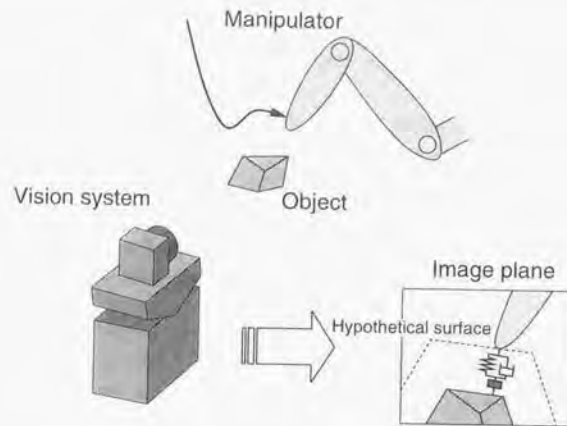


図 4.1 Visually realized hypothetical contact

るいは接触する前に、実際の接触と同様な動きを実現することができる。またそのコンプライアンス特性を適切に設定することで、仮想接触の際の応答が任意に設計可能となる。

すなわち接触作業における仮想コンプライアンス制御と同様な、環境に適応した柔軟な動作が、視覚により実現される。

4.2.2 仮想接触の導入

ここで、前節で述べた仮想接触について、具体的な定義を与える。

まず画像から対象物体の輪郭を抽出し、得られた輪郭線に含まれる画像上の点の集合を $C_o \subset \mathbb{R}^2$ とする。次に、画像上の、ある点 $P = [X, Y]^T$ に対して、輪郭線上の任意の点 $r_o \in C_o$ のうち、 P との距離がもっとも小さいものを選び、その距離を $\varphi(P)$ とする。すなわち、

$$\varphi(P) = \min(|P - r_o|) \quad (\forall r_o \in C_o) \quad (4.1)$$

である。ただし点 P は画像上で対象物体の外部にあるとする。このとき $\varphi(P)$ は、画像上で物体の表面からの距離を値に持つポテンシャル場となる。

そこで、この距離が一定の点の集合、すなわち次式で表わされる点からなる曲線を仮想表面 C_v と呼ぶことにする。

$$C_v = \{P_v \mid \varphi(P_v) = c\} \quad (c: \text{const}) \quad (4.2)$$

このように設定した仮想表面に対して、その内部にマニピュレータが進入したとき、仮想接触が生じるとする。

すなわち、まず画像上でマニピュレータの輪郭線 $C_r \subset \mathbb{R}^2$ に含まれる任意の点 $r_r \in C_r$ のうち、物体表面との距離が最も小さくなる点を仮想接触点 P_t とする。すなわち、

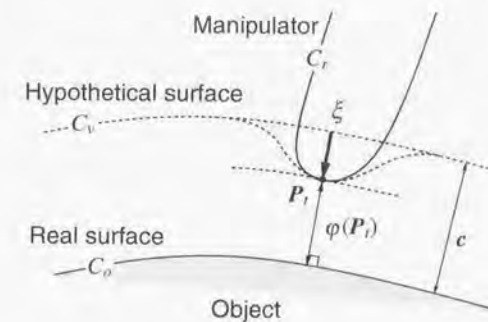
$$\varphi(P_t) = \min(\varphi(r_r)) \quad (\forall r_r \in C_r) \quad (4.3)$$

とする。さらに仮想接触の状態を表わし、マニピュレータが仮想表面に押し込んだときの押し込み量と方向とを示すベクトルを仮想接触ベクトル ξ と呼ぶと、これは次式のように定義することができる。

$$\xi = \begin{cases} \nabla \varphi(P_t) (c - \varphi(P_t)) & \text{where } c \geq \varphi(P_t) \\ \mathbf{0} & \text{where } c < \varphi(P_t) \end{cases} \quad (4.4)$$

ただし $c < \varphi(P_t)$ のときには、仮想接触は生じていないとして、 $\xi = \mathbf{0}$ とした。また $|\nabla \varphi| = 1$ を用いた。

図 4.2 にマニピュレータと対象物体との仮想接触の様子を示す。またマニピュレータと対象物体の輪郭線上で接線が定義できる場合には仮想接触ベクトル ξ は両方の接線に直交する。

図 4.2 Hypothetical contact vector ξ

ここで得られる仮想接触点 P_t と仮想接触ベクトル ξ はいわゆる画像特徴量であり、次節以降で説明するように、本方法ではこれをそのまま用いて制御を行っている。そのため ξ の抽出はリアルタイムで行う必要があるが、その具体的なアルゴリズムについては 4.3.2 節で述べる。

4.2.3 コンプライアンス特性の実現

次に、前節で定義された仮想接触ベクトル ξ に対して、仮想表面内部で、あらかじめ (4.5) 式で表わされるインピーダンス特性が設計されているとする。

$$M\ddot{\xi} + D\dot{\xi} + K\xi = -f_h \quad (4.5)$$

ただし

$$\mathbf{f}_h = \begin{bmatrix} f_x \\ f_y \end{bmatrix} \quad (\in \mathbb{R}^2) \quad (4.6)$$

は画像空間で定義される2次元のベクトルで、画像の中での仮想的な力を表す。 M, D, K は 2×2 の行列で、それぞれ慣性、粘性、剛性行列に対応している。

一般に、(4.5)式は外力 \mathbf{f}_h が作用するときの ξ の運動を表す運動方程式で、例えば機械的インピーダンスをモデル化したものである。しかしここでの外力 \mathbf{f}_h は実際には存在しない仮想的なものである。

逆に、提案する方法では ξ の運動が(4.5)式で表されるインピーダンス特性に一致するように、マニピュレータを制御することを考える。そのためには仮想接触でマニピュレータが ξ だけ押し込んだ結果、仮想接触点 P_i から \mathbf{f}_h なる反力がマニピュレータに加えられると考え、あたかも実際にそのような力に加えられたかのように、画像上で振る舞うように、マニピュレータを制御すればよい。

加えてマニピュレータが剛体リンクからなると仮定すると、画像上でのコンプライアンス中心を定めることで、マニピュレータから見た外力 \mathbf{F}_h によって生じる力のモーメントを計算することができる。その場合、コンプライアンス中心は実現したいタスクに応じて任意に定めることができる。

ここでコンプライアンス中心を P_{cp} とすると、 P_{cp} に加わる仮想的な力のモーメントは次式で表される。

$$N_z = \mathbf{f}_h \times (\mathbf{P}_t - \mathbf{P}_{cp}) \quad (4.7)$$

ただし $N_z \in \mathbb{R}$ は画像面に垂直な軸についての力のモーメントを表している。

以上のことから、(4.5)式で計算される反力 \mathbf{f}_h と(4.7)式による力のモーメント N_z を一つのベクトルにまとめると次式のようにになる。

$$\mathbf{F}_h = \begin{bmatrix} \mathbf{f}_h \\ N_z \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \\ N_z \end{bmatrix} \quad (4.8)$$

仮想接触に伴う望ましいコンプライアンス特性を実現するためには、(4.5)式のインピーダンスパラメータ M, D, K と(4.7)式における剛体リンクのコンプライアンス中心 P_{cp} を適切に設定すればよい。これによってマニピュレータに加えられる仮想的な力とモーメント \mathbf{F}_h が計算され、これを実現するような制御が行われる。

4.2.4 パターンに対するビジュアルインピーダンス

続いてここまでで述べた仮想接触に伴うインピーダンス特性の考え方を、画像パターンに対して拡張する方法を示す。すなわち対象物体の画像パターンとマニピュレータの画像パターンとのインタラクションを考える。

最初にパターン同士の仮想接触を定義する。これは画像上で適当な領域分割を行って、それぞれの領域からの効果がマニピュレータに加えられると考えればよい。例えば具体的な領域分割の方法としては、画面上で仮想接触の状態にある領域が複数の孤立した領域に分けられる場合が考えられる。

さらにマニピュレータに加えられる効果を計算するためには、分割されたそれぞれの領域で計算した結果をそのまま足し合わせればよい。すなわち得られる力とモーメントを \mathbf{F}_{vi} とすると、画面全体からの効果を合わせた力とモーメント \mathbf{F}_h は、簡単に

$$\mathbf{F}_h = \sum_i \mathbf{F}_{vi} \quad (4.9)$$

と計算できる。

例として、図4.3に、はめあいのタスクを実現するためのビジュアルインピーダンスを示す。従来のはめあい作業では RCC (Remote Compliance Center) 機構を用いて、はめあいに適したコンプライアンスの中心を実現するツールが用いられている [10]。これと同様に複数の領域での仮想接触からそれぞれ求まる外力を足し合わせ、コンプライアンス中心に加わる力とモーメントを決定している。その結果、はめあいのタスクに適した RCC 特性を実現することができる。

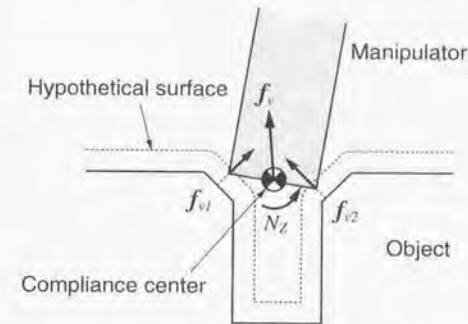


図 4.3 Visual impedance for fitting

4.2.5 マニピュレータへの関節角トルクの計算

前節で計算された仮想的な力とモーメント \mathbf{F}_h から、実際にマニピュレータに加える関節角トルクを導く。これによって仮想接触に伴う対象物体とマニピュレータとのインタラクションが、最終的にマニピュレータの関節角トルクの形で表現され、実際のマニピュレータの制御に適用できるようになる。

ここではマニピュレータはトルク制御で制御されているとする。画像上で観測されるマニピュレータの位置と姿勢を $\mathbf{s}_r \in \mathbb{R}^3$ とする。

$$\mathbf{s}_r = \begin{bmatrix} X_r \\ Y_r \\ \theta_r \end{bmatrix} \quad (4.10)$$

ここで l 次元タスク空間での位置姿勢を $p \in \mathbb{R}^l$ とし、マニピュレータの自由度を m とし、このときのアームの関節角ベクトルを $q \in \mathbb{R}^m$ とすると、次式のヤコビアン $J \in \mathbb{R}^{l \times m}$ を定義することができる。

$$\begin{aligned} J &= \frac{\partial s_r}{\partial q} \\ &= \frac{\partial s_r}{\partial p} \frac{\partial p}{\partial q} \\ & (= J_{\text{image}} J_{\text{robot}}) \end{aligned} \quad (4.11)$$

ただし (4.11) 式の 2 行目、右辺第 1 項は画像ヤコビアン $J_{\text{image}} \in \mathbb{R}^{3 \times l}$ 、第 2 項はマニピュレータのヤコビアン $J_{\text{robot}} \in \mathbb{R}^{l \times m}$ である。これらのヤコビアンは、2 章で述べたように、適当な撮像モデルとマニピュレータのキネマティクスから計算される。

最後に、求めたヤコビアンを用いて、マニピュレータの各関節に与えるトルク $\tau_h \in \mathbb{R}^m$ を次式で計算する。

$$\tau_h = J^T F_h \quad (4.12)$$

これによって仮想接触で加えられる外力 F_h がマニピュレータの関節角トルク τ_h に変換される。

ここで提案する方法では画像から抽出された仮想接触点 P_t と仮想接触ベクトル ξ からマニピュレータへの関節角トルク指令値 τ_h を計算している。これは、4.2.2 節でも述べたように、画像特徴量からマニピュレータの制御指令値を直接計算する特徴量ベースのビジュアルサーボ (2.4 節参照) の構成になっている。

4.2.6 従来の制御法との統合

以上によって計算された τ_h を用いて、実際のマニピュレータの制御器の設計を行う。

τ_h はみかけの上では仮想接触に伴うマニピュレータへの外力として、言い換えるとマニピュレータの動きに伴う受動的な力としてはたらく。よってビジュアルインピーダンスの制御器としては、あらかじめ定められたマニピュレータの目標軌道への制御器、あるいは仮想コンプライアンス特性を実現する制御器で計算される関節角トルク τ_d に対して、ビジュアルインピーダンスによる制御トルク τ_h をそのまま足し合わせるだけとなる。これを次式に示す。

$$\tau = \tau_d + \tau_h \quad (4.13)$$

このブロックダイアグラムを図 4.4 に示す。図に示されるように、マニピュレータのタスク遂行動作に対して、インピーダンス特性を規範としたビジュアルフィードバックをつけ加えることで、環境に適応したコンプライアンス特性を実現している。

なお前節でも述べたように、本方法は特徴量ベースのビジュアルサーボの構成になっている。さらにその制御器はビジュアルフィードバックと他のセンサによるフィードバックを両方同時に用いてマニピュレータのダイナミクスを制御する構成となっている。このため 2.4.4 節で述べたようなサーボレートの問題が、ビジュアルサーボ単体を構成する場合に比べてより一層重要になる。

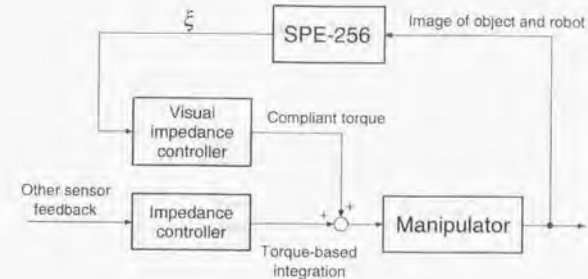


図 4.4 Blockdiagram of visual impedance control

すなわちビジュアルサーボのフィードバックレートが十分でない場合には、そのままでは他のセンサフィードバックのレートと整合させることができず、サーボレートの違いを補償するための機構が必要になる。その結果、システムの安定性と性能を両立させるのは困難だと考えられる。

これに対して本研究の実験システムでは、次節で示すように、3.3 節で説明した SPE-256 システムを用いたビジュアルフィードバックシステムを構築している。そのため動特性の制御に十分な 1ms でのビジュアルフィードバックが実現され、ビジュアルインピーダンスによるフィードバックトルクと他のセンサからの制御トルクは、図 4.4 に示されるようにそのまま単純に足し合わせるだけとなる。全体としてビジュアルサーボと従来のロボット制御との自然な統合が実現されていると言える。

4.3 実験

次に前節で提案したビジュアルインピーダンスの有効性を確かめるために、実際のマニピュレータを用いて原理的な実験を行った。本節では最初にその実験システムの構成、実験で用いた仮想接触点検出のアルゴリズムについて述べる。その後、実際に行った実験の詳細とその結果を示す。

4.3.1 実験システムの構成

実験システムの全体構成を図 4.5 に示す。実験システムは、図 4.5 にも示されているように、ビジョンシステムとして 3.3 節で説明した SPE-256 システムを、マニピュレータとして 7 軸マニピュレータを用いている [54, 62]。ただしマニピュレータについては後述する種々の仮定を用いることで制御を簡便化し、7 軸ある回転関節のうち先端に近い第 5 軸と第 6 軸の 2 軸のみを使用し、他の軸は固定したままである。これは本実験の目的がビジュアルインピーダンスの有効性を原理的に確認するところであり、制御システムの構成を最小限にしたためである。

制御処理については、マニピュレータの制御と SPE-256 システムの制御に複数の並列処理用 DSP を用いて並列分散処理構造をとっている。これはセンサ情報の取得からアクチュエータに指令を送る

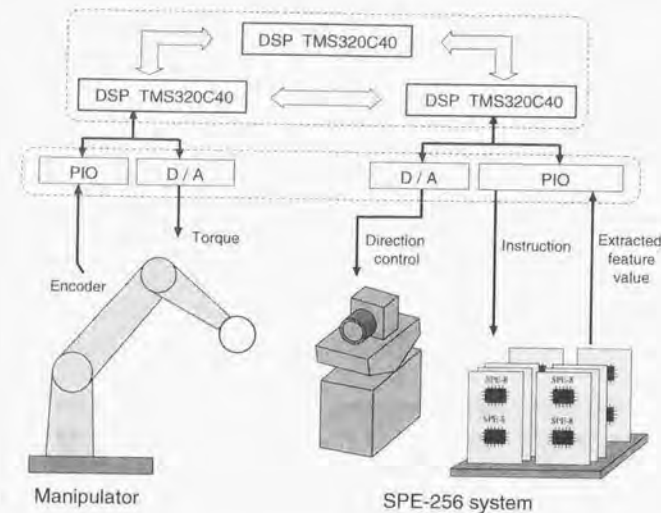


図 4.5 Overview of robot system for experiments of visual impedance

までのフィードバックサイクルタイムを 1ms 以内に押さえることを目標に、全体としてボトルネックのない制御システムを実現するためである。

実験ではマニピュレータの先に半径 3cm 程度の白い球を取り付け、これをマニピュレータの手先として SPE-256 システムで視覚認識を行っている。さらにアクティブビジョンはマニピュレータの制御とは独立に、手先のトラッキングを行っている。3.3 節で説明したようにアクティブビジョンのトラッキングはマニピュレータの運動と比較して充分高速に行われている。そのため実験ではマニピュレータの手先は常に画面の中心に捕らえられていると考えてよい。

マニピュレータのコンプライアンス中心 (4.2.4 節参照) は、簡単のため、球の中心にとった。球による仮想接触では仮想接触ベクトル ξ の方向は常に球の中心方向の逆を向くので、結果として (4.8) 式において $N_{\xi} = 0$ が常になりたつ。

またマニピュレータとビジョンシステムの配置について図 4.6 に示す。まずマニピュレータで使用する 2 つの回転関節軸について、それぞれ直交するように配置し、次にこれらの回転軸と画像面の X 軸、Y 軸の方向についてもそれぞれ直交するようにビジョンを配置した。このとき各軸の回転による手先の運動方向と画像面の X, Y 軸方向が一致するため、(4.11) 式におけるヤコビアン J は対角行列となる。さらに実験では可動範囲は充分小さいとして、この時のヤコビアンを用いて定数で近似した。

このようにヤコビアンが常に対角の定数行列となり、前述したように $N_{\xi} = 0$ であるため、結果

として画面上で位置と姿勢に対応した 3 自由度のうち、姿勢に対応した 1 自由度は制御する必要がなくなり、位置に対応した 2 自由度のみを制御すればよいことになる。

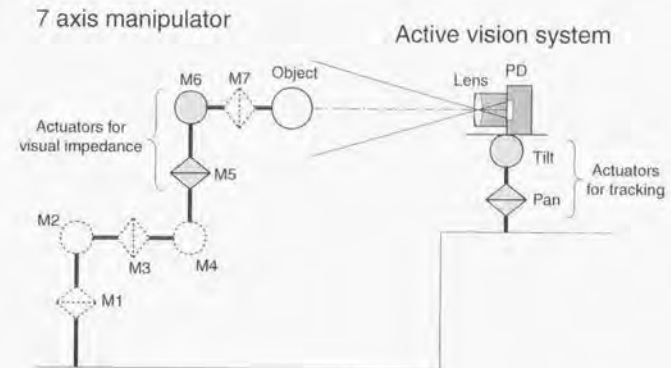


図 4.6 Experimental setup for visual impedance

4.3.2 仮想接触点抽出のアルゴリズム

ここでは SPE-256 システムを用いた仮想接触点抽出のアルゴリズムを示す。画像上の 2 つのパターンに対する複数領域の仮想接触点検出について、システムの超並列性を利用したアルゴリズムを用いる。

ただし複数物体の領域分割については、石井らにより提案されているセルフウィンドウ法 [56] を用いた。この方法のうち 2 値パターンに対する 2 物体のトラッキングの方法により、マニピュレータのパターンと対象物体のパターンの 2 つの領域への分割を行っている。

分割された 2 つのパターン間の最短距離を求めるアルゴリズムには、モルフォロジ (morphology) [72] と呼ばれる信号処理の手法を利用する。手順は以下の通りとなる。制御のサイクルタイムごとに、step0 から step7 までを繰り返す。ただしあらかじめ抽出されているマニピュレータのパターンに含まれる点の集合を A 、対象物体の点を B としている。

- step0. $j = 0$
- step1. $A(0) = A$, $i = 0$
- step2. $A(i)$ を Dilation し、 $A(i+1)$ とする。 B との共通部分 $B \cap A(i+1)$ があるか調べる。
- step3. 共通部分が見つければ、「接触あり」としてその座標を $x(A_j)$ とし、step4へ。そうでなければ step5へ。

- step4. $j = j + 1$. これ以降, $x(A_j)$ 近傍は接触点としては無視する. (step2 において, $x(A_j)$ に対応したパターンをあらたに用意し, これを Dilation しつつマスクをかければよい.)
- step5. $i = i + 1$. 仮想表面までの距離に対応して, $i > i_{max}$ なら処理は終り. そうでなければ step2 にもどる.
- step6. B についても, 同様に, step1 から step5 を繰り返す. それぞれの j について $x(A_j)$ に対応する点 $x(B_j)$ が見つかるはずである.
- step7. それぞれの j について $x_j = x(A_j) - x(B_j)$ を計算する.

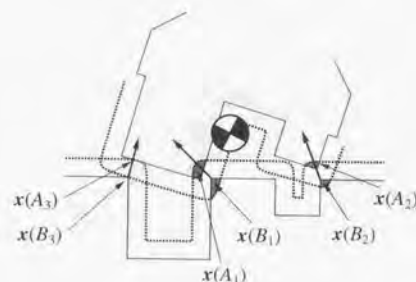


図 4.7 Pattern fitting using visual impedance

求めた x_j が仮想接触ベクトルである. このベクトルを使ってロボットをインピーダンス制御する. すなわちコンプライアンスセンタを決め, それぞれのベクトルを力とモーメントに分解し, ロボットへ与える外力とする.

この方法の利点として, 3点以上の接触も可能なので, 複雑な形状のもののどうしのはめ合いができるようになることがあげられる. また, ロボットと部品との関係としてヤコビアンだけがわかればよく, 正確なハンドリングを必要とせずにはめ合いが可能になる.

具体的な実装については, 衝突回避の実験では仮想接触点は1点のみとの仮定を設け, 提案したアルゴリズムを単純化して1msのサイクルタイムを実現している. 一方はめあいの実験では2点での仮想接触を考慮する必要がある. 提案するアルゴリズムについて1msで動作するように実装することができなかった. これは3.3.2.2節で述べたように, SPE-256システムでシステムの制御に汎用のパラレルIOを用いているため, IOの速度でインストラクションサイクルが制限されているのが原因である. これに対してCPVシステムではシステムの制御のためにコントローラを実装しこの問題は解決されている.

4.3.3 衝突回避の実験

ここでは, ロボットが高速に運動している場合でも仮想接触により衝突が回避され, 同時にインピーダンスパラメータを適切に設定することで振動などの動的な応答が制御できることを示す.

実験では, ロボットの手先に取りつけた球を一定トルクで運動させ, 進行方向の障害物となる壁面との仮想接触を行った. その実験結果を図4.8に示す.

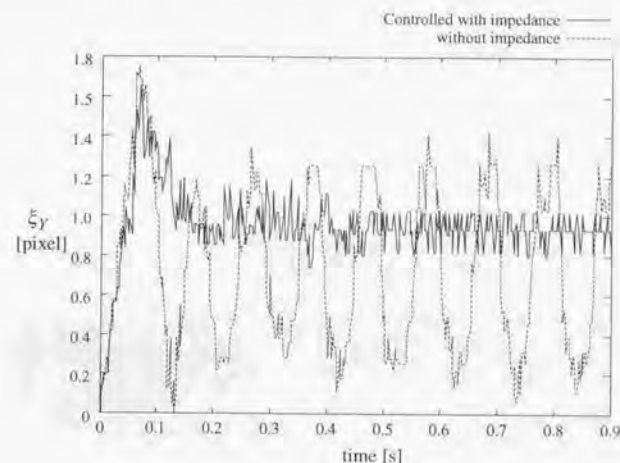


図 4.8 Respons of experiment of virtual contact

グラフはロボット運動に伴う手先の球の重心座標の軌跡を示しており, $\xi_y = 1.0$ に釣り合いの位置がある. また, グラフの実線は, ビジュアルインピーダンス制御を適用してインピーダンスパラメータとして粘性項を加えたもの, 破線は, 単なる比例制御で回避を行ったものである.

単なる比例制御の場合には, たとえ衝突を回避したとしても振動が生じているのに対して, ビジュアルインピーダンスを適用した場合には, 適切な動特性が実現されて振動が抑えられているのがわかる.

次に, 対象の予測できない運動に対しても応答できることを確認するために, 静止した手先に対して人間が手で対象物体を近づけて, 衝突回避の実験を行った. 具体的にはマニピュレータの手先につけた球とはほぼ同じ大きさの球を棒の先に付け, 実験者が手で球をマニピュレータの手先に近づけて応答を調べた. 実験結果を連続写真で図4.9に示す.

静止したマニピュレータの手先に対して, 人間が手で対象物体を近づけると, 衝突回避動作が実現されている.

4.3.4 はめあいの実験

ここではビジュアルインピーダンスを用いたタスクの例として、はめあい動作を取りあげる。先に述べたように、はめあい動作では RCC と呼ばれる環境に適応したならい動作が必要とされる。この実験では、視覚から得られるパターンとしての環境の情報を利用し、これをリアルタイムにフィードバックしてロボットの RCC 特性と同様な動特性を実現することができる。

実験では、ロボットの手先の球をすきまにはめこむタスクを行った。与えた目標軌道は実際のすきまとずれており、ビジュアルフィードバックなしで、ロボットが与えられた軌道どおりに運動するだけではタスクが遂行できない。これに対してビジュアルインピーダンスを適用した結果を示す。

マニピュレータの手先の軌跡をグラフにしたものを図 4.10 に示す。グラフ中の円と長方形が、手先に取り付けられた球とはめ込む先の隙間に対応している。

グラフはロボットの手先の球の軌跡を 45ms ごとにサンプリングした点で示している。すきまの両側とロボットとの仮想接触が行われ、タスク遂行に適したならい動作が行われている。

また実験の様子を連続写真にしたものを図 4.11 に示す。

4.4 考察

この章ではビジュアルサーボの枠組みを拡張し、最初から望ましいダイナミクスをタスクとして与える動的なビジュアルフィードバック制御の方法を提案した。具体的には仮想接触の概念の導入により、視覚情報から、実際の接触時における力覚と同様な効果を発生させ、これによってロボットの動的な制御を実現している。

同時にこの方法では仮想接触力を介して、関節角トルクのレベルでビジュアルフィードバックと他のセンサフィードバックとの融合を実現している。そのため、ここで示した位置フィードバック以外にも、力センサからのセンサフィードバックや、あるいは他の視覚センサからのフィードバックを同じマニピュレータ上で融合する方法への拡張も容易である。

このように SPE-256 システムによって実現された超高速のビジュアルフィードバックにより、従来のシステムではできなかった視覚によるダイナミクスの制御が実現されるようになった。

一方、課題としては、はめあいの実験でプログラムの高速化が十分達成できなかったことがあげられる。これは先に説明した CPV システムではプログラム制御機能について格段に向上し、また、より高解像度の画像が利用可能となったことで、今回以上に複雑な視覚パターン情報を用いた高度な制御が可能になると予想される。

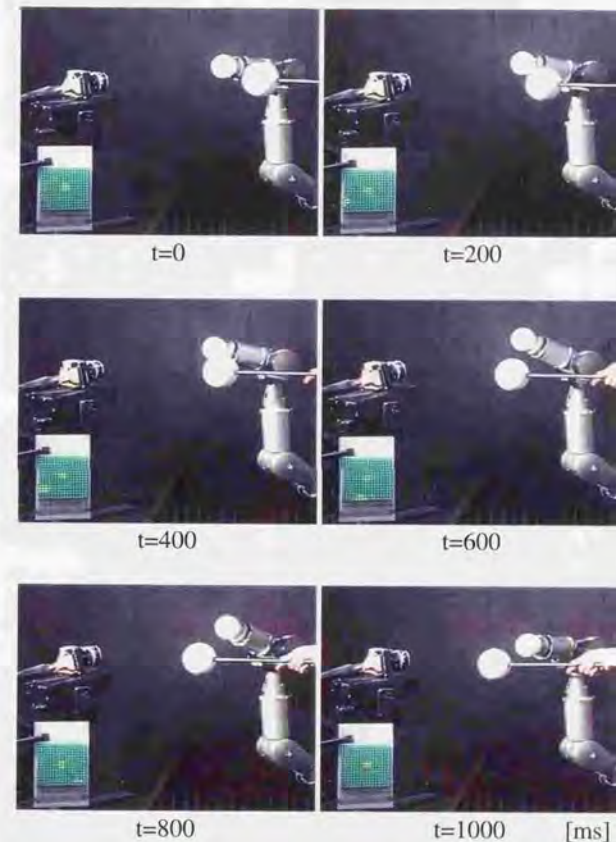


図 4.9 Experiment of collision avoidance

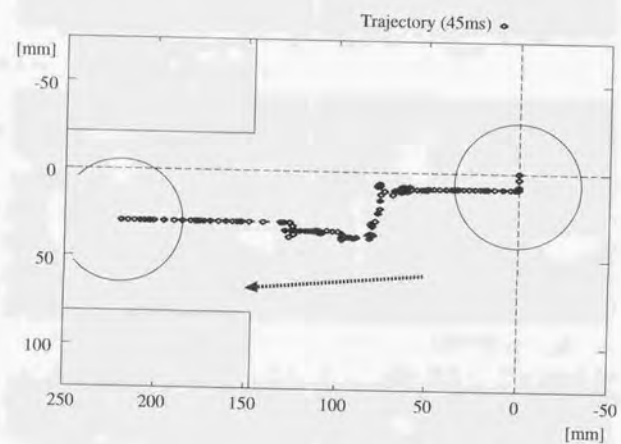


図 4.10 Experiment of fitting task

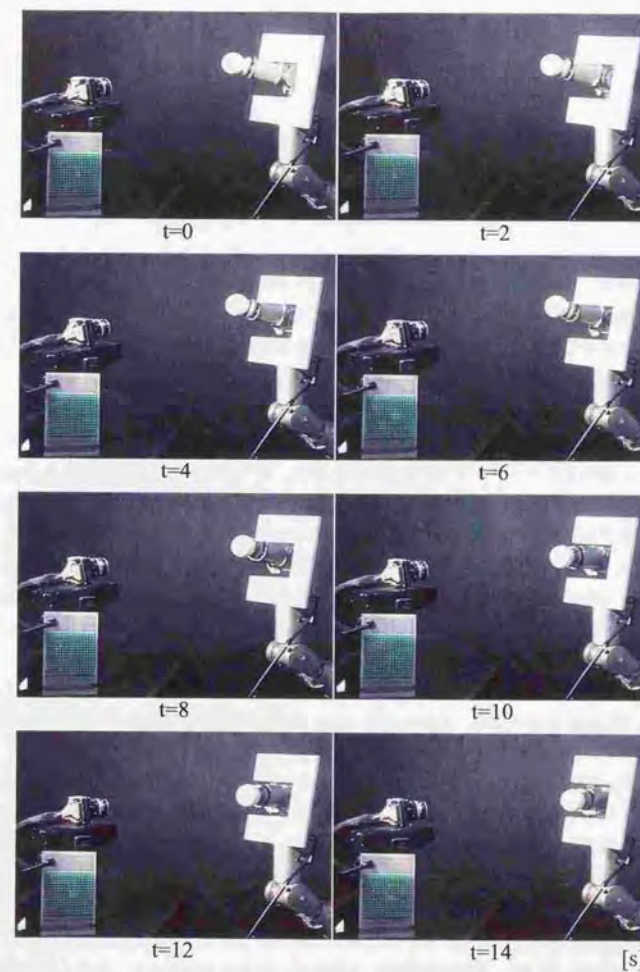


図 4.11 Fitting task

第5章

結論

本論文ではビジョンシステムとビジュアルフィードバック制御について、サンプリングレートに代表される動的な問題の重要性を指摘し、この観点からビジュアルフィードバックとビジョンシステムの構成を総合的に検討し直すことにより、ビジョンシステムの最適な設計と実際の構築を行った。また動的なビジュアルフィードバックの手法を提案して、構築したビジョンシステムにインプリメントし、提案する手法とシステムの有効性を確かめた。

具体的には、第2章で、従来のビジュアルフィードバックの手法におけるサンプリングレートの問題を指摘し、動的なビジュアルサーボの実現のために、ビジョンシステムに求められる性能について考察した。またアクティブビジョンについても、動的な制御の概念を導入して、アクティブビジョンの動特性がビジョンシステムの設計に大きく関わることを示した。

第3章では、前の章の議論にもとづいてビジョンシステムの設計を行い、これに従って、完全並列と列並列の2つの異なる方法により実際にシステムの構築を行った。構築したシステムはそれぞれアクティブビジョンに搭載され、ターゲットトラッキングの実験を行って、システムとしての性能を評価した。その結果、それぞれのシステムで目標とした仕様を達成していることが示された。

これによって従来のビジョンシステムの限界を打ち破るものとして、超並列・超高速ビジョンの重要性を明らかにすることができた。これはビジョンシステムを構成する全ての要素について、それぞれの要求仕様を明らかにし、トップダウンの設計を行った結果である。

最後に、第4章では、新しいビジュアルインピーダンスの概念を提案し、その理論の詳細を示した。この方法では、これまでビジュアルフィードバックで見られなかったタスクレベルでの望ましいダイナミクスを与える制御法について、視覚においてもこれが有効となることを提案し、実際に衝突回避とはめあいという具体的な例に適用して、実験によりその有効性を示した。

以上のように、本研究で従来のビジョンシステムの問題点が明らかになり、同時にこれを用いたビジュアルフィードバックが、ダイナミクスに関して不十分であったことが示された。今後、視覚によるロボットの制御において、本研究で構築した超並列・超高速のビジョンシステムが重要な役割を果たすようになると考えている。

本論文ではその目的としてロボットを取り上げてきたが、この研究の主旨は特にロボットに限ったものではない。序論で述べたように、今日の機械システムは多機能化、自律化の方向へと向かって

いる。すなわち本研究で考察したビジュアルフィードバック制御あるいはビジョンシステムは、今後ロボットを含む様々な自動機械システムが人間と共存し、活躍するために必要となるものと思われる。

謝辞

指導教官である石川正俊先生には、研究内容の全般から、研究を進めるにあたって考え方とポイント、心構え、実際の進め方について、そして研究者としての社会的な態度と責任について、何度も同じ誤りを繰り返す自分に、いつも丁寧に教えを頂きました。誠にありがとうございます。また、なによりこのようなすばらしい研究をする機会を与えて下さったことに、深く感謝致します。

藤村貞夫先生には、修士の頃から研究に対する緻密な考え方と真摯な態度の重要性について、厳しい指導を頂きました。お言葉を頂く度に、いつも反省しきりでした。心からお礼申し上げます。

本論文の副査を受けて頂いた館璋先生、安藤繁先生、稲葉雅幸先生には、まとまらない論文と要領を得ない発表にも、予備審査の時から親切に、かつ示唆に富む貴重な指摘を頂きました。深く感謝致します。

共同研究の相手としてご協力頂いた浜松ホトニクス、中央研究所の豊田様、向坂様、中村様、また半導体事業部の水野様、山川様には、皆様のおかげでこのような貴重なデバイスを利用することができ、これなくしては自分の博士課程の研究はあり得なかったと思っています。これにお返しできる成果が挙げられたかどうか、ご判断を仰ぎたいと思います。心から深くお礼申し上げます。

助手の石井さんには、いつも研究の指南を頂き、研究意欲の源泉として、また想像力をかきたてるトリガーとしても、影に表に、大きな影響をうけました。いつも頼りきりでご迷惑をお掛けしてばかりでしたが、この機会にお詫びさせて頂き、また心から感謝させて頂きます。

藤村研の助手の喜安さんには、毎日の細々とした面倒なこと、計算機のこと、常に親身になって話を聞いて頂き、様々にお世話になりました。どうもありがとうございました。

研究室のメンバーでも、特にこの研究に近い分野のグループの方々には、かなりの部分で直接的にこの研究を助けて頂きました。特に、石井さんはもとより、DSP 関連、計算機関連、ロボット制御で並木さん、ビジョンシステムのアーキテクチャで小室君、コントローラのコアアーキテクチャ、アセンブラでは鏡君に頼ってきました。加えて、成瀬さん、Birloni Ferenc 君、奥君、川又君の、研究員、院生の皆様にはいつも新鮮かつ示唆に富む議論を頂きました。またなにより、楽しく研究生活が送れたのは全く皆様のおかげです。深く感謝致します。

また秘書の坂本さん、宮澤さん、吉武さん、玉井さん、勝山さん、また計数教務室の皆様には、何度もご迷惑をお掛けしながらも、最後までお世話を頂きました。心からお礼申し上げます。

最後に、勝手気ままに過ごした大学院での生活を、理解と愛情をもって援助してくれた私の両親にも、この場を借りてお礼させて頂きたいと思います。どうもありがとうございました。

参考文献

- [1] 石川正俊、超高速・超並列ワシチップビジョンとその応用。日本ロボット学会誌, Vol. 13, No. 3, pp. 31-34, 4 1995.
- [2] M. Ishikawa, K. Ogawa, T. Komuro, and I. Ishii. A CMOS Vision Chip with SIMD Processing Element Array for 1ms Image Processing. In *Dig. Tech. Papers of 1999 IEEE Int. Solid-State Circuits Conf. (ISSCC'99)*, pp. 206-207, 1999.
- [3] M. Ishikawa, T. Komuro, Y. Nakabo, A. Namiki, and I. Ishii. Vision Chip with General Purpose Processing Elements and Its Application. In *Proc. Int. Symp. on Future of Intellectual Integrated Electronics*, pp. 169-174, 1999.
- [4] Y. Shirai and H. Inoue. Guiding a Robot by Visual Feedback in Assembly Tasks. *Pattern Recognition*, Vol. 5, pp. 99-108, 1973.
- [5] Lee E. Weiss, Arthur C. Sanderson, and Charles P. Newman. Dynamic Sensor-Based Control of Robots with Visual Feedback. *IEEE J. Robotics and Automation*, Vol. RA-3, No. 5, pp. 404-417, 1987.
- [6] 橋本浩一、視覚フィードバック制御-静から動へ。システム / 制御 / 情報, Vol. 38, No. 12, pp. 659-665, 1994.
- [7] K. Hashimoto et al. *Visual Servoing*. World Scientific, 1993.
- [8] S. Hutchinson, G. D. Hager, and P. I. Corke. A Tutorial on Visual Servo Control. *IEEE Trans. Robotics and Automation*, Vol. 12, No. 5, pp. 651-670, 1996.
- [9] John Y. S. Luh, Michael W. Walker, and Richard P. C. Paul. Resolved Acceleration Control of Mechanical Manipulators. *IEEE Trans. Automatic Control*, Vol. 25, No. 3, pp. 468-474, 1980.
- [10] 杉本浩一、ロボットアームの力フィードバック制御。計測と制御, Vol. 25, No. 1, pp. 45-50, 1986.
- [11] 小管一弘, 古田勝久, 横山. ロボットの仮想内部モデル追従制御系-メカニカルインピーダンス制御への応用。計測と制御, Vol. 21, No. 1, pp. 55-62, 1988.

- [12] ロボット学会編. ロボット工学ハンドブック, V.2.2 ソフトウェアサーボの実現, pp. 532-534. コロナ社, 1990.
- [13] P. I. Corke and M. C. Good. Dynamic Effects in Visual Closed-Loop Systems. *IEEE Trans. Robotics and Automation*, Vol. 12, No. 5, pp. 671-683, 1996.
- [14] Joun T. Feddema and Owen R. Mitchell. Vision-Guided Servoing with Feature-Based Trajectory Generation. *IEEE Trans. Robotics and Automation*, Vol. 5, No. 5, pp. 691-700, 1989.
- [15] William J. Wilson. *Visual Servoing*, chapter Visual Servo Control of Robots Using Kalman Filter Estimates of Robot Pose Relative to Work-Pieces, pp. 71-104. World Scientific, 1993.
- [16] F. Chaumette and A. Santos. Tracking a Moving Object by Visual Servoing. In *Automatic Control: Proc. IFAC 12th Triennial World Congress*, pp. 643-647, 1993.
- [17] N. P. Papanikolopoulos, P. K. Khosla, and T. Kanade. Visual Tracking of a Moving Target by a Camera on a Robot: A Combination of Control and Vision. *IEEE Trans. Robotics and Automation*, Vol. 9, No. 1, pp. 14-35, 1993.
- [18] Peter K. Allen and Billibon Yoshimi. Automated Tracking and Grasping of a Moving Object with a Robotic Hand-Eye System. *IEEE Trans. Robotics and Automation*, Vol. 9, No. 2, pp. 152-165, 1993.
- [19] A. J. Koivo and Nasser Houshang. Real-Time Vision Feedback for Servoing Robotic Manipulator With Self-Tuning Controller. *IEEE Trans. Systems, Man and Cybernetics*, Vol. 21, No. 1, pp. 134-142, 1991.
- [20] 橋本浩一, 井上貴博, 木村英紀. ビジュアルサーボイング-非線形制御アプローチ. 日本ロボット学会誌, Vol. 13, No. 2, pp. 263-269, 1995.
- [21] 橋本浩一, 木村英紀. ビジュアルサーボイング-非線形オブザーバアプローチ. 日本ロボット学会誌, Vol. 13, No. 7, pp. 986-993, 1995.
- [22] B. J. Nelson and P. K. Khosla. Force and Vision Resolvability for Dexterous Manipulation. *IEEE Trans. Robotics and Automation*, Vol. 12, No. 5, pp. 714-731, 1996.
- [23] A. Castano and S. Hutchinson. Visual Compliance: Task-Directed Visual Servo Control. *IEEE Trans. Robotics and Automation*, Vol. 10, No. 3, pp. 334-342, 1994.
- [24] 井上博允. リアルタイムビジョンとその応用. 日本ロボット学会誌, Vol. 13, No. 3, pp. 306-310, 1995.

- [25] 金出武雄, 木村茂. ビデオレート・ステレオマシン. 日本ロボット学会誌, Vol. 13, No. 3, pp. 322-326, 1995.
- [26] 稲葉雅幸. 局所相関プロセッサを用いたロボットビジョン. 日本ロボット学会誌, Vol. 13, No. 3, pp. 327-330, 1995.
- [27] 内山隆, 森田俊彦, 沢崎直之, 浅田和徳, 佐藤雅彦, 井上博允, 稲葉雅幸. 動き追跡処理システム「トラッキングビジョン」. 日本ロボット学会誌, Vol. 16, No. 1, pp. 52-53, 1998.
- [28] 佐々木茂, 塩原守人. 動画処理プロセッサ ISHTAR によるリアルタイム・オブティカル・フロー検出. 日本ロボット学会誌, Vol. 13, No. 3, pp. 331-334, 1995.
- [29] 藤田善弘, 山下信行, 岡崎信一郎. リアルタイムビジョンのための並列処理 LSI IMAP とそのシステム. 日本ロボット学会誌, Vol. 13, No. 3, pp. 339-342, 1995.
- [30] 大道武生, 川内直人. オープンロボットコントローラ. 日本ロボット学会誌, Vol. 14, No. 6, pp. 10-13, 1996.
- [31] 小林 尚登他. ロボット制御の実験. 計測自動制御学会, 1997.
- [32] 加賀見聡. ロボット研究のための PC/AT 互換機上のリアルタイム OS. 日本ロボット学会誌, Vol. 16, No. 8, pp. 1036-1041, 1998.
- [33] 久野義徳. アクティブビジョン. 人工知能学会誌, Vol. 10, No. 4, pp. 493-499, 1995.
- [34] 石黒浩. 注視に基づくロボットの視覚. 人工知能学会誌, Vol. 10, No. 4, pp. 500-506, 1995.
- [35] Andrew Blake and Alan Yuille. *Active Vision*. MIT Press, 1992.
- [36] 出口光一郎. 画像と空間, 2 投影, pp. 7-38. 昭晃堂, 1991.
- [37] 有本卓. ロボットの力学と制御, 3. ロボットの運動方程式, pp. 64-99. 朝倉書店, 1990.
- [38] W. Jang and Z. Bien. Feature-based Visual Servoing of an Eye-In-Hand Robot with Improved Tracking Performance. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 2254-2260, 1991.
- [39] 細田耕, 五十嵐司, 浅田稔. 未知環境内で動作するロボットのビジュアルサーボ / カサーボハイブリッド制御系. 日本ロボット学会誌, Vol. 15, No. 4, pp. 642-647, 1997.
- [40] 橋本浩一, 海老根巧, 木村英紀. ビジュアルサーボイング-最適制御によるアプローチ. 日本ロボット学会誌, Vol. 12, No. 5, pp. 129-134, 1994.
- [41] Alfred A. Rizzi and Daniel E. Koditschek. An Active Visual Estimator for Dexterous Manipulation. *IEEE Trans. Robotics and Automation*, Vol. 12, No. 5, pp. 697-713, 1996.

- [42] Peter I. Corke. *Visual Servoing*, chapter Visual control of robot manipulators-A review, pp. 1-31, World Scientific, 1993.
- [43] R. P. Paul. *Robot Manipulators*. MIT Press, 1987.
- [44] J. J. Craig. *Introduction to ROBOTICS - mechanics and control*. Addison-Wesley, 1989.
- [45] Henrik I. Christensen, Kevin W. Bowyer, and Horst Bunke. *Active Robot Vision*. World Scientific, 1993.
- [46] 向井利春, 石川正俊. 複数センサによる予測誤差を用いたアクティブセンシング. 日本ロボット学会誌, Vol. 12, No. 5, pp. 715-721, 1994.
- [47] 中坊嘉宏, 石井抱, 石川正俊. 高速視覚フィードバックを用いたアクティブセンシング. ROBOMECH'98 講演論文集, pp. 2BII3-5, 1998.
- [48] 画像電子学会編. 画像電子ハンドブック, 1. 画像情報論, pp. 1-10. コロナ社, 1993.
- [49] T. Komuro, I. Ishii, and M. Ishikawa. Vision Chip Architecture Using General-Purpose Processing Elements for 1ms Vision System. In *Proc. IEEE Int. Work. on Computer Architecture for Machine Perception*, pp. 276-279, 1997.
- [50] Y. Nakabo, I. Ishii, and M. Ishikawa. High Speed Target Tracking Using 1ms Visual Feedback System. In *Video Proc. IEEE Int. Conf. on Robotics and Automation*, 1996.
- [51] 中坊嘉宏, 石井抱, 石川正俊. 超並列・超高速ビジョンを用いた1msターゲットトラッキングシステム. 日本ロボット学会誌, Vol. 15, No. 3, pp. 417-421, 1997.
- [52] Y. Nakabo and M. Ishikawa. Visual Impedance using 1ms Visual Feedback System. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 2333-2338, 1998.
- [53] 中坊嘉宏, 石川正俊. ビジュアルインピーダンス. 日本ロボット学会誌. 投稿中.
- [54] A. Namiki, Y. Nakabo, I. Ishii, and M. Ishikawa. High Speed Grasping Using Visual and Force Feedback. In *Proc. IEEE Int. Conf. on Robotics and Automation*, 1999.
- [55] 大脇崇史, 中坊嘉宏, 並木明夫, 石井抱, 石川正俊. 視覚モダリティ変換を用いたリアルタイム実現仮想接触システム. 電子情報通信学会論文誌, Vol. J81-D-II, No. 5, pp. 918-924, 1998.
- [56] 石井抱, 石川正俊. 1ms ビジュアルフィードバックシステムのための高速対象追跡アルゴリズム. 日本ロボット学会誌, Vol. 17, No. 2, pp. 195-201, 1999.
- [57] 鏡伸吾, 中坊嘉宏, 小室孝, 石井抱, 石川正俊. 1ms ビジョンチップシステムの制御アーキテクチャ. 電子情報通信学会技術報告, Vol. ICD99-8, CPSY99-8, ETS99-8, pp. 15-20, 1999.

- [58] 村田達也, 松内良介, 中坊嘉宏, 石井抱, 石川正俊. 超並列・超高速ビジョンチップシステムの制御アーキテクチャ. 電子情報通信学会技術報告, Vol. PRMU97-150, pp. 161-168, 1997.
- [59] 向井利春, 石川正俊. 並列ビジョンのための2次元座標変換回路. 電子情報通信学会技術報告, Vol. PRU92-72, pp. 111-116, 1992.
- [60] 坂口隆明, 小室孝, 石井抱, 石川正俊. ビジョンチップのためのモーメント出力回路. SICE '96 予稿集, pp. 829-830, 1996.
- [61] 小室孝, 石井抱, 中坊嘉宏, 石川正俊. デジタルビジョンチップのためのモーメント抽出アーキテクチャ. 電子情報通信学会技術報告, Vol. PRMU99-51, pp. 17-22, 1999.
- [62] 並木明夫, 中坊嘉宏, 石井抱, 石川正俊. 1ms 感覚運動統合システムにおける高速並列分散処理. ROBOMECH'99 講演論文集, pp. 2P2-49-058, 1999.
- [63] 中坊嘉宏, 石井抱, 石川正俊, 豊田春義, 水野誠一郎. 列並列 S³PE アーキテクチャによる超高速ビジョンシステム (CPV-I). ROBOMECH'99 講演論文集, pp. 1P1-65-096, 1999.
- [64] 中坊嘉宏, 石川正俊, 豊田春義, 水野誠一郎. 列並列・超高速ビジョンシステム (CPV-I) を用いたアクティブビジョン. 第17回 ロボット学会学術講演会予稿集, p. 2B23, 1999.
- [65] 谷江和雄, 林原靖男. 機械の柔らかい制御. 計測と制御, Vol. 34, No. 4, pp. 274-279, 1995.
- [66] 小管一弘. 力制御法の分類と制御システムの設計法. 日本ロボット学会誌, Vol. 9, No. 6, pp. 751-758, 1991.
- [67] N. Hogan. Impedance Control: An Approach to Manipulation, Part I, II, III. *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 107, No. 1, pp. 1-24, 1985.
- [68] 辻敏夫, 畑木道生, 赤松寛優, 金子真. マニピュレータの非接触型インピーダンス制御. 日本ロボット学会誌, Vol. 15, No. 4, pp. 616-623, 1997.
- [69] 赤松寛優, 辻敏夫, 金子真. 視覚情報を利用したマニピュレータの非接触型インピーダンス制御. 第14回 ロボット学会学術講演会予稿集, pp. 623-624, 1996.
- [70] 中坊嘉宏, 石井抱, 石川正俊. ビジュアルインピーダンスを用いたロボットの制御. ROBOMECH'96 講演論文集, pp. 999-1002, 1996.
- [71] 中坊嘉宏, 石川正俊. 超並列・超高速視覚システムを用いたはめ合い動作の実現. 第14回 ロボット学会学術講演会予稿集, pp. 999-1002, 1996.
- [72] Jean Serra. *Image Analysis and Mathematical Morphology*. Academic Press, 1982.

