ダイナミックメモリを用いた実時間対象追跡

日浦 慎作[†,☆] 村瀬 健太郎[†,☆☆] 松山 隆司[†]

本論文では時変な事象を処理の対象とするモジュールやエージェントなどが並列動作する際に有効な通信方法を提案する．従来のマルチエージェントシステムにおいて多用されてきた黒板モデル（共有メモリ）に，時変な値を維持・管理する機能を付加し拡張した機構（ダイナミックメモリアーキテクチャ）を用いることによって次のような利点が得られる．(1) 各モジュールまたはエージェントは完全非同期に動作することができる．(2) 通信遅延や書き込み/読み出し時刻の差による誤差の影響を抑制することができる．(3) 各モジュールは独立に設計することができるため，システム構築が容易である．これらの利点を検証するため，我々は視点固定型パンチルトカメラを用いた物体追跡システムを構築した．このシステムでは画像を処理する「知覚モジュール」と，カメラの方位を制御する「行動モジュール」をそれぞれ独立させ，それらの間の通信をダイナミックメモリにより実装した．これらは知覚により得た対象の方位・速度や，画像処理を最適化するためにカメラの角速度などの情報を互いに交換する相互依存の関係にあるが，ダイナミックメモリの働きによりこれらがまったく独立した処理サイクルで動作しつつ安定に対象が追跡できることが確認された．また異なった対象物体抽出原理に基づく 2 種類の知覚モジュールを製作し，それらを入れ換えることでモジュラリティが高いことも確認した．

Real Time Object Tracking Using Dynamic Memory

Shinsaku Hiura,[†,☆] Kentaro Murase[†,☆☆]
and Takashi Matsuyama[†]

The paper proposes a novel scheme of active vision named *Dynamic Vision*. It is best characterized by rich interactions between visual perception and camera action modules. The richness is twofold: 1) Rich information is exchanged between the modules to realize both stable image processing and adaptive camera control. 2) Rich dynamic interactions between the modules are realized without disturbing their own intrinsic dynamics. To implement a dynamic vision system, we propose the *Dynamic Memory Architecture*, where perception and action modules share what we call the *Dynamic Memory*. It maintains not only *continuous* temporal histories of state variables such as pan-tilt angles of the camera and the target object location but also their predicted values in the future. Perception and action modules are implemented as parallel processes which dynamically read from and write into the memory according to their own individual dynamics. The dynamic memory supports such asynchronous dynamic interactions (i.e. data exchanges between the modules) without making the modules wait for synchronization. A prototype system for real time moving object tracking demonstrated the effectiveness of the proposed idea.

1. はじめに

我々はこれまで，動的に変化するシーンの状況を実時間的に計測し把握するためのシステム構築法である分散協調視覚を提案してきた[1)]．これは従来視覚システムとして提案されてきたアクティブビジョンやセンサフュージョンの概念を拡張・発展させた考え方である．すなわち，計測パラメータを動的に変更しうる多数の能動視覚エージェントをネットワークにより結合し，それらのコミュニケーション・協調により情報統合を行うことで広範囲にわたる複雑なシーンの計測と解釈を行おうとするものである．このシステムでは，個々の能動視覚エージェント（観測ステーション）単体の視覚機能の優劣だけではなく，各エージェント間の通信・協調法がシステム全体の能力に大きく影響する．なぜなら個々のエージェントがシーンから得る情報は部分的かつ不完全であるため，それらを適切に組み合わせ統合することが重要であるからである．またシー

† 京都大学大学院情報学研究科
Graduate School of Informatics, Kyoto University
☆ 現在，大阪大学大学院基礎工学研究科
Presently with Graduate School of Engineering Science, Osaka University
☆☆ 現在，株式会社富士通研究所
Presently with Fujitsu Laboratories Ltd.

ンは動的であり刻々変化するのに対し，各エージェントのリソース☆は有限であるため，実時間処理のためには適切な競合解消や負荷分散を行う必要がある．なおかつシステムのスケーラビリティやフォールトトレラント性を確保するために，個々のエージェントは自律的であり独立性を有することが望ましい．

このように分散協調視覚システムは実時間制約下の動的マルチエージェントシステムという側面を持つため，従来のエージェント研究で提案されているような協調・競合解消方式やオントロジー・通信言語の統一法だけではなく，より実時間性を意識した通信・協調方式が用いられなければならない．この点でこれまでに提案されてきたエージェント間通信方式には様々な問題があり，特に移動対象の追跡など時変な事象を取り扱う際の通信方式として十分なものは存在しなかった．

たとえば，**黒板法**では各エージェントが非同期的に内容を読み書きするため，エージェントの自律性を確保することが容易である半面，時変な対象を取り扱う場合には情報の書き込みから読み出しまでの時間差に起因した誤差が実対象と通信内容の間に生じる．またパケット型通信では受信側の動作としてポーリング法と割込み法があるが，前者では受信側の処理サイクルが読み出し状態に達し，受信バッファからデータを取り出すまでの期間が伝送遅延時間に加算されるため黒板法と同様の問題が存在する．後者の方法ではこの問題を避けることができるが，受信側の処理は送信側に同期しなければならず，エージェントの自律性は失われる．またこれらの方法では負荷や計算能力の差により生じる待ち時間が各エージェントの実効能力を変化させるため，対象の複雑さや処理アルゴリズムが動的に変化するシステムでは非効率的である．

従来のエージェント間通信法を拡張することにより動的なタスクに対応しようとする試みはこれまでにも研究例があるが十分とはいえない．たとえば黒板法の拡張である white-board architecture[2),3)]は，共有メモリ上に必要なデータが揃うまで読み出し側のモジュールを一時停止することで同期を実現する方法であるため，モジュールの非同期性は失われる．また smart buffer[4)]は異なる動的特性を持つ複数のプロセスを動的に整合させるような仮想的な数値を生成するシステムであるが，これも各エージェントの自律性の観点では不十分である．

そこで本論文では，各エージェントが非同期性を保ちながらも送受信の時間差に起因する誤差を抑制することが可能な，新たな通信方式（**ダイナミックメモリアーキテクチャ**）を提案する．この方式では黒板法と同等の非同期性を有しながら，伝達遅延の影響を吸収することが可能であり，時変な事象を取り扱うマルチエージェントシステムの構築において非常に有効な方式である．

☆ 計算や通信のリソースだけではなく，カメラの方位などの物理的パラメータも各時刻において単一の値しかとりえないため，競合を起こしうる重要なリソースである．

2. 動的能動視覚（Dynamic Vision）

分散協調視覚システムにおいて，パン・チルトやズームなどの撮影パラメータ調整機構を有するエージェント（観測ステーション）は能動視覚エージェントと呼ばれる．このような視覚システムはアクティブビジョンシステム[5),6)]と呼ばれ研究されてきたが，これらのうち大半は**撮影パラメータ**の最適化を主としたものである[7)]（**図 1**）．しかし分散協調視覚システムでは，他の能動視覚エージェントと協調した観測を行う必要性から撮影パラメータの競合が発生することがあり，観測ステーション単体にとって最適な撮影パラメータがつねに設定可能であるとは限らない．そのため，画像解析手法に適した撮影パラメータを決定する方法だけではなく，先に決定された撮影パラメータに合わせ**画像解析パラメータ**を適応させる必要が生じる．

さらに移動対象の追跡など，対象の状態や撮影パラメータが動的に変化する場合，**画像撮影時刻**の決定も重要な要素である．たとえば，移動対象が画面内に収まるように撮影するには，撮影方位と撮影時刻の両者は不可分な要素である．また，2 台のカメラで対象の 3 次元位置を正確に求めるためには，両者の撮影時刻を正確に合わせる必要がある．このように分散協調視覚システムでは，通信の影響や実時間性を無視できないため，逐次処理的なアクティブビジョンの枠組みだけでは十分ではない．

他方，視覚サーボ[8),9)]の研究では実時間性を重視した制御理論などが多数提案されているが，これらの手法では知覚と行動が密に結び付けられており自律性が低い．そのため分散協調視覚システムのように，シー

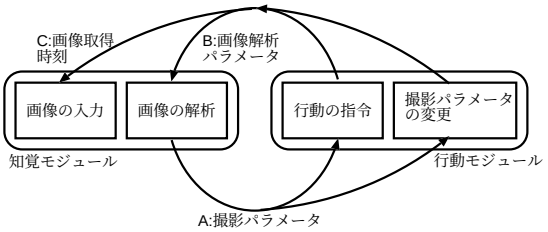

図 1 知覚・行動の各モジュール間のインタラクション
Fig. 1 Interaction between perception and action module.

ンの変化や他の能動視覚エージェントとのインタラクションにより行動や知覚の処理内容が動的に変化しうる状況へ適用することが困難である．

そこで本論文では，このような動的要素の強い能動視覚を**ダイナミックビジョン**（Dynamic Vision）と名付け，従来の研究とは特に区別する．分散協調視覚システムを構成する単位である各々の能動視覚エージェントは，ダイナミックビジョンシステムであると言い換えることができる．ただし本論文はこのダイナミックビジョンシステム単体の構成法について考察したものであり，各々の能動視覚エージェントは通信・協調処理に適した構成であることを重視するが，エージェント間の協調・通信方式そのものを提案するものではない．

これまでの議論で明らかなように，ダイナミックビジョンシステムでは以下のような条件を満たしたシステム構成をとるべきである．

(1) 画像を取得し処理を行う知覚機能部と，撮影パラメータを更新し制御する行動機能部はそれぞれ**独立したモジュール**として構築する．
(2) 各モジュールは**並行かつ非同期**に動作する．
(3) 各モジュール間の通信は統一されたインタフェースを持ち，なおかつ非同期な送受信の時刻差などに起因する誤差を抑制する機能を有する．

なお，他の能動視覚エージェントとの通信を行う通信モジュールや，リソースの競合解消やモジュールの管理を行うモジュールも並列に動作するものとする☆．このような構成をとることにより，シーンの状況や他のエージェントとのインタラクションに対し柔軟な対応が可能な能動視覚エージェントを構築することが可能となる．

これらの条件のうち (1)，(2) については容易に実現が可能であるが，(3) の問題を直接的に表現する概念やこれを明示的に取り扱った研究例，またそれを実現する手法は提案されていない．そこで次に，この機能を実現するアーキテクチャである**ダイナミックメモリアーキテクチャ**を提案する．

3. ダイナミックメモリアーキテクチャ

従来用いられている共有メモリや黒板法などの通信法では，前章の条件，つまり各モジュールが自律性・非同期性を保ちつつ時変な対象に関する情報を高精度に伝達することは不可能である．このような通信法について考察するためには，伝達される値だけではなく，その伝達が行われる時刻を陽に取り扱う必要がある．そこで通信される情報を表現する時刻を**インデックス時刻** t_{idx} と定義し，送信側が情報を発信する時刻 t_w や受信側が情報を受け取る時刻 t_r とは特に区別する．ただし以下では，各モジュールは共通の時刻を共有し，モジュール間でずれはないものとする☆☆．

前章の条件 (1)，(2) より，情報の発信・取得は非同期に行われるため，$t_w \neq t_r$ である．また，あるモジュールがセンサや画像などから情報を得た後，情報を発信するまでの処理や，情報を受信してからその情報を利用するまでの処理には有限の時間を要するため，一般に $t_{idx} \neq t_w$，$t_{idx} \neq t_r$ である．たとえば，時刻 t_{idx} にセンサから得たカメラ方位を時刻 t_w に共有メモリへ書き込む場合，$t_{idx} < t_w$ となる．また画像キャプチャの後に，その画像が撮影された方位を共有メモリから読み出すときは，画像の撮影時刻を t_{idx} とすると $t_{idx} < t_r$ となる．画像処理の結果得られた対象の方位を共有メモリから読み出し，そちらへカメラを向ける処理を行う場合は逆に $t_{idx} > t_r$ となる．このように，t_{idx}，t_w，t_r の大小関係には制約条件はなく，任意の時刻に関する情報がつねに読み書き可能でなくてはならない．しかし，情報の発信側と受信側がそれぞれ指定する t_{idx} が互いに一致する保証はないため，このずれを補償する機構が必要となる．

我々は以上の考察から，共有メモリを拡張することにより動的能動視覚に適したモジュール間通信法を構築することを考えた．このような共有メモリ（ダイナミックメモリ）は従来の共有メモリを以下の点について拡張したものである（**図 2**）．

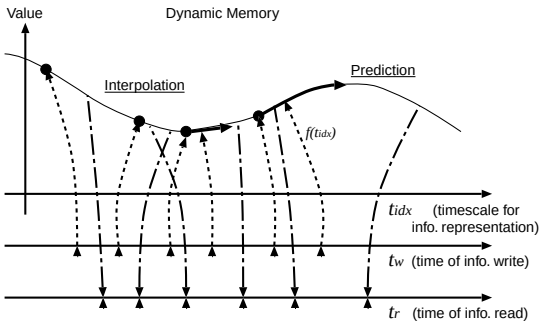

図 2 ダイナミックメモリにおける情報の表現とその操作
Fig. 2 Representation and manipulation of information on dynamic memory.

☆ これらのモジュールは分散協調視覚システムを構築するためには不可欠であるが，能動視覚エージェント単体の問題ではないため，本論文では論じない．

☆☆ 単一の計算機内はもとより，ネットワークを介して接続された計算機間でも，ntp などの技術によりこの仮定は容易に満たされうる．

(1) メモリへは単なる値のみが書き込まれるのではなく，その値を記述するためのインデックス時刻 t_{idx} とともに書き込まれ，この時刻は情報の発信時刻 t_w とは独立である．
(2) 書き込まれる値は瞬時値のみでなく，インデックス時刻 t_{idx} に関する連続関数として書き込むことも可能である．
(3) 値をメモリから読み出す際には，インデックス時刻 t_{idx} を指定する．この時刻は，情報の読み出し時刻 t_r とは独立である．
(4) メモリは関数値の計算や補間（内挿)・予測（外挿）の機能を有し，必要に応じ読み出し時に機能する．
(5) 最新の情報が新たに書き込まれたとき，過去の情報との不連続性を抑制する機構（平滑化機能）を有する．

本論文では，このような通信方式を用いた動的マルチエージェントシステム構築法を**ダイナミックメモリアーキテクチャ**として提案する．しかしこれらの機能は，特殊な専用のメモリ装置として物理的に実装する必要はなく，ソフトウェア的に実現してもよい．連続関数を表現する方法としても，リモートオブジェクトのようなプログラムコード形態で実装する以外に，特定の関数（たとえば多項式）の係数を与えるような方法でもよい．最も重要な点は関数値の計算や補間・予測機能を通信プロトコル内に隠蔽することにより，これらを情報取得側で行う場合に発生する問題を避けることができることである．すなわち，個々のモジュールの独立性を高めることが可能となり，情報の発信側と取得側の整合性を意識する必要がなくなる．たとえば，画像から物体を検出するアルゴリズムが切り替わることにより物体の方位の表現方式が変更された場合には，単に書き込まれる情報を変更すればよく，情報取得側には変更を加える必要がない．また駆動機構の相違などにより運動特性が異なるカメラに対して同一の知覚モジュールを利用することなどが可能である．

ダイナミックメモリによって実行される補間・予測が効果的に機能するかどうかはタスクの種類や処理アルゴリズムなどにより異なると考えられる．幸い，一般に行動モジュールにおける撮影パラメータ（カメラ方位など）制御では，あらかじめカメラの動特性などを同定して制御に利用していることが多く，将来の値の変化を高精度に推定することができる連続関数を容易に提供できることが多い．シーンの状況に関しても，知覚モジュールは対象の予測に利用可能な情報をシーンから得ることが可能な場合があり，このような場合には知覚モジュールにより動的に予測方法を切り替えることなどが可能となるため，行動モジュールに予測機構を組み込むような方法に比べ柔軟性が高いなどといった利点もある．またネットワークを介した複数の能動視覚エージェント間の通信においても，ダイナミックメモリを用いることにより通信遅延の影響を抑制することが可能であると考えられる．

4. 動的能動視覚に基づく動物体の追跡システムの構築

本論文ではダイナミックメモリを用いた動的能動視覚システムの有効性を示すため，視点固定型パンチルトカメラを用いた動物体の実時間追跡システムを構築した．以下ではダイナミックメモリの導入法を中心にシステムの構成を述べる．

4.1 タスクとシステム構成

タスク カメラのパン角・チルト角を制御し，シーン中に侵入した対象を画面の中心にとらえるように対象を追跡する．監視システムにおいて，対象追跡中の画像を監視員が観察することなどを想定し，対象像の重心位置を**なめらかに**追跡するようにカメラを制御する．

カメラ このシステムでは視点固定型パン・チルト・ズームカメラと呼ばれる形式の能動カメラを仮定する．このカメラは，1) パン・チルトの回転軸が直交する，2) レンズの投影中心が回転軸の交点上にある，という条件を満たすカメラであり，カメラのパン・チルトにより視差が発生しないという利点がある[10]．実験では市販の能動カメラである SONY EVI-G20 を用いるが，これは上記の条件をほぼ満たし，視点固定型パン・チルト・ズームカメラとして用いた場合の誤差は実用上無視することができる[10]．

計算機 PentiumII 266 MHz の PC に FreeBSD を実装したホスト計算機の PCI バス上に，超並列画像処理装置 RVS-10G を搭載したものを実験に用いる．RVS-10G は 256 個のプロセッサエレメントを一次元的に結合した SIMD 型並列画像処理装置であり，10^{10} operations/sec. のピーク性能を持ち，ホスト CPU と並列に動作する．知覚モジュールと行動モジュールは UNIX プロセスとして実装され，ダイナミックメモリはこれらのモジュール間の共有メモリクラスとして実装されている．知覚モジュールは RVS-10G を制御し画像処理を行う．また行動モジュールは RS-232C を介して EVI-G20 を制御する．

4.2 知覚モジュールの構成

我々は能動カメラから得た画像からリアルタイムに対象物体の領域を切り出すための画像処理法として，視点固定型パン・チルト・ズームカメラの特性を利用した2種類の手法（背景差分法に基づく方法，オプティカルフロー検出に基づく手法）を導入した．提案手法が高いモジュラリティを持つことを確認するため，実験では同一の行動モジュールに対しこれらのモジュールを組み合わせ，どちらの場合も安定に対象が追跡可能であることを確認した．

4.2.1 背景差分による対象の検出

背景差分法は，あらかじめ対象物体を含まないシーンの像（背景画像）を記憶しておき，入力画像と背景画像の明度差をとることにより対象を抽出する手法である．しかし能動カメラを用いた場合，カメラの方向などにより背景の像は変化するため，我々は視点固定型パン・チルトカメラの特性を利用した背景差分法を提案した[10]．この手法では**図 3** のように，撮影時のカメラ方位と同一の方位の画像を背景画像データベース（Appearance Plane）から生成し背景画像として用いる方法である．視点固定型パン・チルトカメラはカメラの運動に対して視差を生じないため，継目のない単一の Appearance Plane を用いて任意のカメラパラメータに対する背景画像を生成することができる．

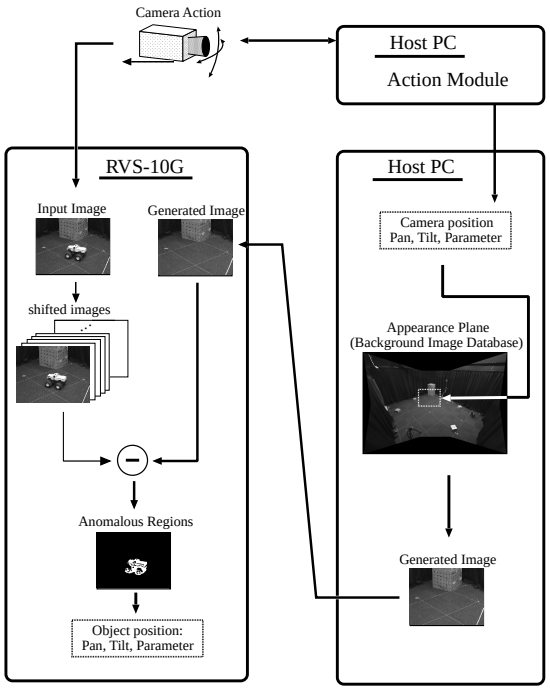

図 3 背景差分法による対象の抽出
Fig. 3 Object detection based on background subtraction.

Appearance Plane を応用した背景差分法が正常に動作するためには，背景画像の生成に用いたカメラ方位が，カメラの撮影時方位と正確に一致する必要がある．従来の研究[10]ではカメラを間欠的に動作させ，静止時に画像を撮影していたが，本研究ではカメラを滑らかに動作させるためカメラを停止させることはできない．カメラ運動中は撮影時刻のずれが方位のずれとして直接現れるため，本研究ではダイナミックメモリを用いてカメラ撮影時刻のカメラ方位を取得することにより誤差の抑制を図る．しかしなおわずかな誤差が残存することが考えられるため，図3のように入力画像を微小に並行移動させた画像を複数枚生成し，背景画像との残差が最も少ないときの対象領域を結果として用いる手法をとった．この並行移動量はわずかであるため，透視変換を並行移動により近似することによる誤差は1画素以下である．この誤差を吸収するため，差分の計算には一方の画像の8近傍を含む9画素と差分をとり，それらの最小値を差分結果として用いた．またモーションブラーの影響に対処するため，背景画像のエッジ強度を用いて画素ごとに閾値を設定し，差分結果の二値化に用いた．

Appearance Plane から背景画像を作成する処理にホスト CPU を，入力画像の並行移動と差分計算に RVS-10G を用いることで並列処理し，15 frames/sec の処理速度を得た．このとき画像は縦横 256 画素で，並行移動した画像は上下左右 8 画素の計 64 枚作成した．

4.2.2 オプティカルフローを用いた対象の検出

カメラの運動にともない撮影された画像から移動対象を検出する方法として，対象と背景の見かけ上の動き（オプティカルフロー）の差を利用する方法が考えられる（**図 4**）．カメラの運動自由度や背景の複雑さによって様々な方法が考えられるが，ここでも視点固定型パン・チルトカメラの性質を利用して簡単化された手法を用いる．すなわち視点固定型パン・チルトカメラではパン・チルトにともなう運動視差が生じないため，カメラの回転角速度と静止背景のオプティカルフローの間に1対1の関係が成立することを利用した対象の抽出を行う．静止背景上の点 Q を画像平面上へ投影した点を (x, y)，カメラの姿勢角を (θ, ϕ) とすると，オプティカルフロー $(\Delta x, \Delta y)$ は次式で表される．ただし，カメラ運動機構はジンバル構造であり，固定部から順にパン機構，チルト機構を経てカメラが搭載されているものと仮定する．

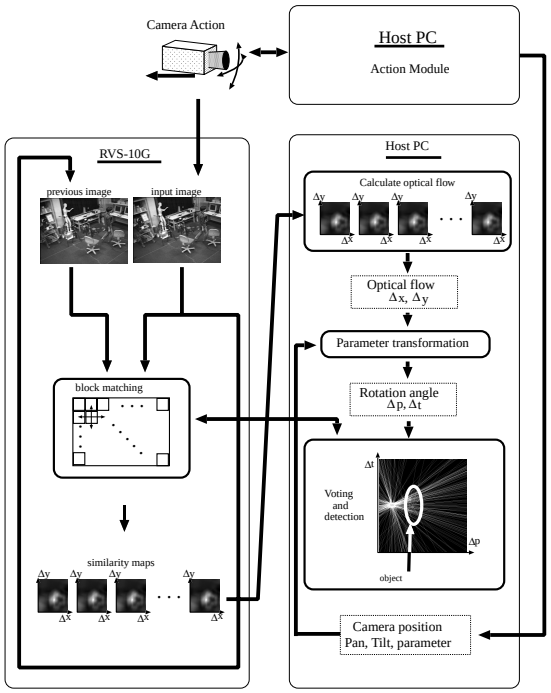

図 4 オプティカルフローによる対象の抽出
Fig. 4 Object detection based on optical flow.

$$\Delta x = \left(\frac{x^2+\rho^2}{\rho}\sin\phi - y\cos\phi\right)\Delta\theta + \frac{xy}{\rho}\Delta\phi$$
$$\Delta y = \left(\frac{xy}{\rho}\sin\phi + x\cos\phi\right)\Delta\theta + \frac{y^2+\rho^2}{\rho}\Delta\phi \quad (1)$$

ここで ρ は焦点距離を表す．この式は，カメラの回転角速度が一定でも投影点 Q の位置 (x, y) やカメラのチルト角の瞬時値 ϕ によりオプティカルフローは様々に変化しうることが分かる．幸い，カメラのチルト角はチルト角速度の積分値であり急激に変化することはないため，この積分やカメラに搭載されたポテンショメータなどから容易に得ることができる．そこでこれらの値が既知であると仮定すると，式 (1) を変形することによりオプティカルフローからカメラ角速度を求める式が得られる．

$$\Delta_{p(\phi)} = \frac{(y^2+\rho^2)\Delta x - xy\Delta y}{(x^2+y^2+\rho^2)(\rho\sin\phi - y\cos\phi)}$$
$$\Delta t(\phi) = \frac{\{(x^2+\rho^2)\sin\phi - y\rho\cos\phi\}\Delta y - x(y\sin\phi + \rho\cos\phi)\Delta x}{(x^2+y^2+\rho^2)(\rho\sin\phi - y\cos\phi)} \quad (2)$$

この式を用いることで，画像の各部から得たオプティカルフローのうち静止背景部分に関する値はすべて実際のカメラの角速度 $(\Delta x, \Delta y)$ へと変換される．また逆に，静止背景とは異なる動きをする部分からはこれらとは異なるパン・チルト値が得られるため，この性質を用いて静止背景領域と移動物体領域の識別を行う．識別の一手法として，行動モジュールとの通信により得た実際のカメラ角速度を画像から得た値と比較する方法が考えられる．しかし実際にはセンサの精度や通信遅延による誤差が含まれる．そこで画像から得たシーンの見掛けの角速度を投票空間へ投票し，ピーク値を検出することで画像から正確なカメラの角速度を得る．物体領域は，このピーク値から外れた投票値を与えた領域として抽出する．なお，カメラの角速度が高速である場合に対応するため，オプティカルフローを求めるための類似画素探索範囲や投票空間は角速度 0 の点を中心に設定するのではなく，行動モジュールから得たおおよそのカメラ角速度を中心とした範囲に設定する．

前後する 2 枚の画像間でブロックマッチングを行い，ブロックの類似度分布を得る処理を高速化するために RVS-10G を用いた．またその類似度分布からオプティカルフローを求め，カメラ角速度への変換・投票と対象領域を抽出する処理にホスト CPU を用いることで並列処理し，15 frames/sec の処理速度を得た．このときブロックサイズは縦横各 16 画素とし，これを縦横 16 画素ずつ並行移動し類似度分布を求めた．アパーチャ問題により類似度分布が点のピークを持たず，線状となる現象を類似度分布のモーメント解析により検出し，投票先を点から直線に切り替える処理も行っている．

4.3 行動モジュールの構成

行動モジュールは対象の方位に関する情報を知覚モジュールから受け取り，カメラをその方位へ制御するタスクを負う．ダイナミックメモリを用いて通信を行うため，各モジュールは完全に自律的に動作することが可能である．そのため行動モジュールは既存の単純な制御アルゴリズムをそのまま利用することが可能であり，本論文では PID 制御を用いている．まず RS232C を経由してカメラへパン・チルト角読み出しコマンドを送出し，その値が得られると，その時刻に関する対象の方位をダイナミックメモリから読み出す．次にそれらの差から PID 制御を用いてカメラ角速度を決定しカメラへ送出する．

行動モジュールの自律性により，制御対象のモデル同定や各パラメータ調整は行動モジュールを単体で用いた実験により行うことが可能である．実験の結果，カメラの回転角速度はパン・チルト方向とも，遅れ $\tau = 70\,\mathrm{ms}$ の無駄時間系と時定数 $T = 50\,\mathrm{ms}$ の一次遅れ系を直列に接続したモデルで良好に近似できることが分かった．初速を v_{p_0}，目標値を v_{p_1} としたとき

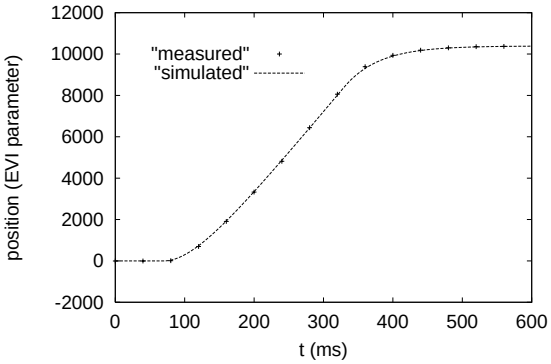

図 5 カメラ運動モデル
Fig. 5 Dynamics model of the camera.

のカメラのパン方向の角速度 $v_p(t)$ のステップ応答は以下のようになる．チルト方向についても同様である．

$$v_p(t) = \begin{cases} v_{p_0} & (0 \le t < \tau) \\ (v_{p_1} - v_{p_0})(1 - \mathrm{e}^{-\frac{(t-\tau)}{T}}) + v_{p_0} & (\tau \le t) \end{cases} \tag{3}$$

この数式を用いたシミュレーションにより求めたカメラ方位と，実際のカメラ方位とを比較した例を**図 5** に示す．縦軸はカメラ方位角に比例した値でカメラに固有の値である（以下，実験結果も同じ）．良好な近似が実現されていることが分かる．また，これを制御するための PID 制御の各パラメータは，系が無駄時間系を含むため，実験的に決定した．

5. ダイナミックメモリの実装

ダイナミックメモリは知覚モジュールと行動モジュールの間の双方向情報伝達のために用いられる．知覚モジュールから行動モジュールへは，知覚により得た対象の方位情報が伝達されカメラの制御に用いられる．また逆に，行動モジュールから知覚モジュールへはカメラの運動情報が伝達され，知覚モジュールにおける画像処理の最適化のために用いられる．

3 章で述べたように，ダイナミックメモリからは任意の時刻に関する情報が読み出し可能でなくてはならない．またこれらは必要に応じて更新・変更されるため，今回我々は情報の内部的表現として区分的な時間関数を用いる方法をとった．すなわち情報は以下のような形式で表現される．

$$\begin{aligned} INFO: \quad & f_0(t) \quad \text{for} \ \ t_0 \le t < t_1 \\ & \vdots \\ & f_n(t) \quad \text{for} \ \ t_n \le t < t_{n+1} \end{aligned} \tag{4}$$

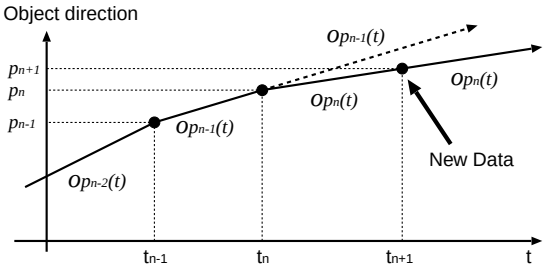

図 6 対象運動情報の表現と更新
Fig. 6 Representation and revision of object motion.

$$f_{n+1}(t) \quad \text{for} \ \ t_{n+1} \le t < \infty$$

ここで関数が定義される区間 $[t_n, t_{n+1})$ や関数 $f_n(t)$ の係数は共有メモリに格納され，情報の入出力にはダイナミックメモリクラスライブラリが用いられる．このライブラリにより情報読み出し時に時刻 t_{idx} に関する関数値の計算などが行われる．以下では対象とカメラの運動情報について，それらの具体的な表現方法とその管理方法について述べる．

5.1 対象の運動情報の表現と管理

4.2.1 項および 4.2.2 項で述べた方法では，対象の見掛けの方位のみがカメラから画像が取り込まれた時点の瞬時値として得られる．一方行動モジュールでは任意の時刻に関する対象の方位が必要となるため，ダイナミックメモリ内において補間関数の生成を行う．そのためには対象の運動をモデル化する必要があるが，このタスクでは物理法則に従って運動するとは限らない人物などの追跡を行うため，高度なモデルによる推定は無意味である．そこで補間には単純な線形近似を用いる．

図 6 に示すように，対象方位の予測・補間関数 $o_{p_n}(t)$ は対象の方位が得られた時刻を境界とした各区間 $[t_n, t_{n+1})$ において，その区間の両端の対象方位 p_n，p_{n+1} を結ぶ直線により表現する．

$$OBJ_{p_n}: o_{p_n}(t) = \frac{p_{n+1} - p_n}{t_{n+1} - t_n}(t - t_n) + p_n \quad \text{for} \quad t_n \le t < t_{n+1} \tag{5}$$

ただし，図 6 に示すように，最新の対象方位が得られた時刻 t_{n+1} よりも後については，最後の区間の直線を延長して外挿する．これを対象の方位角・仰角の双方について同様に行う．

5.2 カメラ運動情報の表現と管理

知覚モジュールにおける対象検出（背景差分による方法，オプティカルフローによる方法）の安定度は，それぞれ画像撮影時のカメラ方位やカメラ角速度の精度に大きく影響されるため，カメラの運動情報は高精度に伝達されなければならない．そこでこれらの値を

4つの区分的な関数，$v_{p_n}(t)$，$v_{t_n}(t)$（パン・チルトのカメラ角速度）および $c_{p_n}(t)$，$c_{t_n}(t)$（カメラ方位）を用いて表現し，ダイナミックメモリを用いて伝達する．幸いカメラは計算機により制御されている対象であり，その動特性は 4.3 節で示したように高精度なモデル化が可能であるため，このモデルを用いることでカメラ運動の高精度な補間・予測が可能である．

行動モジュールはカメラに対し以下の2つの操作を行っており，これらの情報を基にカメラ運動モデルの更新を行う．

(1) カメラへ角速度を設定する操作
(2) 現在のカメラ方位を取得する操作

ただし実験で用いるカメラの特性から，これらの操作は同時には行うことができないため，これらの操作は交互に行われることになる．

まず (1) については，カメラへ設定された角速度を式 (3) の v_{p_1} へ代入することで，カメラの角速度を予測・補間する関数 $v_{p_{n+1}}(t)$ を得ることができる．ここで操作 (1) が行われた時刻を t_n と置く．

$$SPEED_{p_n} : v_{p_n}(t) = v_p(t - t_n) \quad \text{for} \quad t_n \le t < \infty \qquad (6)$$

ただし式 (3) 中の初速 v_{p_0} については，ダイナミックメモリ中の前回の予測関数 $v_{p_{n-1}}(t)$ を用い，$v_{p_0} = v_{p_{n-1}}(t_n)$ として求める．またカメラの方位はこの式の積分により以下のように得られる（**図 7**）．

$$\hat{c}_{p_n}(t) = c_{p_{n-1}}(t_n) + \int_{t_n}^{t} v_{p_n}(t)dt \qquad (7)$$

この式でも同様にカメラの初期方位は前回の予測関数の値 $c_{p_{n-1}}(t_n)$ を用いることにより，補間関数の連続性が失われることを防いでいる．

しかしこの方法では，カメラ角速度については十分高精度であるものの，カメラ方位については積分による誤差の蓄積が懸念される．そこで次に，(2) の情報を用いてカメラ方位のみの修正を行う．ここで (2) の情報が得られた時刻を t'_n とすると，カメラ方位は時定数 $T_n = a(t'_n - t_n)$ の一次遅れ系を用いて以下のように修正することとする．a は誤差修正の早さを決定するパラメータである．

$$DIR_{p_n} : c_{p_n}(t) = \hat{c}_{p_n}(t) + E_p \cdot (1 - e^{-\frac{t-t_n}{T_n}}) \quad \text{for} \quad t_n \le t < \infty \qquad (8)$$

ここで E_p は時刻 t'_n における推定カメラ方位 $\hat{c}_p(t'_n)$ と (2) により得たカメラ方位との差である（図 7）．この方法では時刻 t'_n において誤差 0 とはならないが，時間の経過にともない誤差をなめらかに減少させることが可能であり，誤差の蓄積の影響を排除することができる．なお以上の議論はチルト値 $v_{t_n}(t)$，$c_{t_n}(t)$ についても同様である．

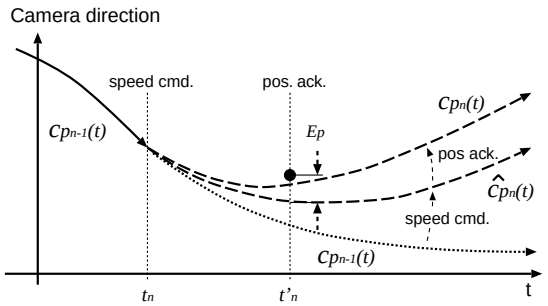

図 7 カメラの運動情報の更新ルール
Fig. 7 Rules of revising camera motion information.

6. 実　　験

4.1 節で述べた機器を用いて人物の動きを追跡する実験を行った．カメラは一般オフィス環境の壁面高さ約 2.5 m に設置した．まず知覚モジュールとして，4.2.1 項で述べた背景差分法を行動モジュールと組み合わせ動作させた結果について示す．**図 8** は，動作中の入力画像と検出された対象領域を 20 frame おきに抜き出したものである．検出結果の画像中の灰色の小矩形は，対象領域の重心を表す．対象の安定に検出されていることが分かる．

図 9 は，知覚・行動の各モジュールがダイナミックメモリへアクセスした時刻を示したものである．それぞれは完全に非同期に動作しており，行動モジュールの処理サイクルが知覚モジュールのそれに比べ長いことから読み書きのタイミングがずれている様子が分かる．

図 10 は，カメラ方位 (a) および対象の位置情報 (b) について，ダイナミックメモリへ書き込んだ値と読み

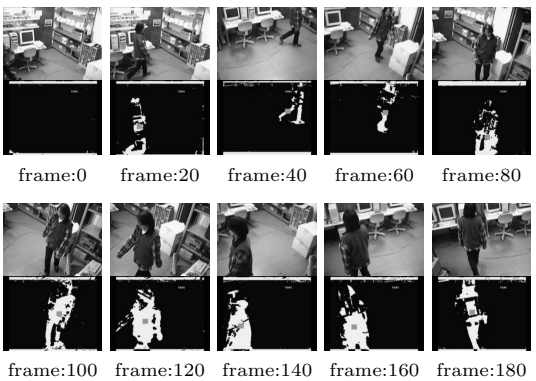

図 8 背景差分法による追跡結果
Fig. 8 Tracking result based on background subtraction.

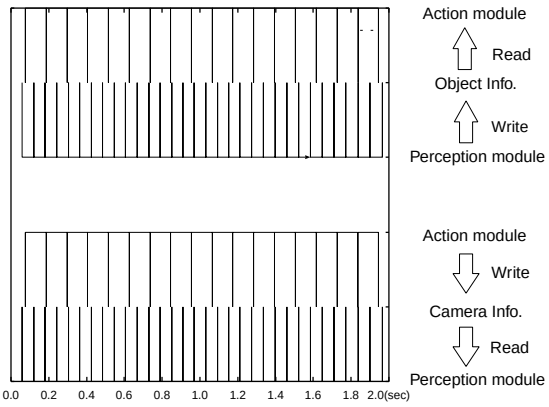

図 9 知覚，行動のダイナミックメモリへのアクセスタイミング
Fig. 9 Access timing to dynamic memory by perseption and action module.

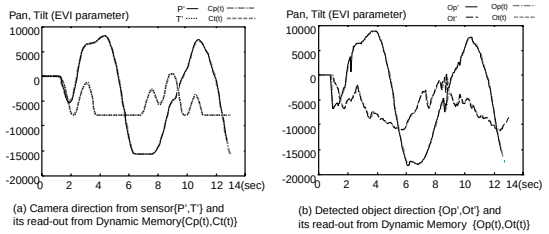

図 10 情報の時間に関する整合性
Fig. 10 Consistency of information towards time axis.

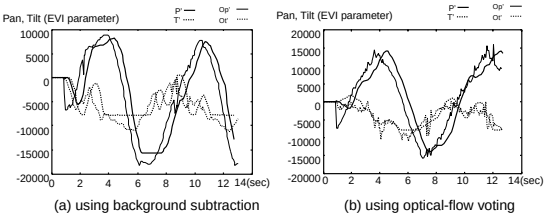

図 11 追跡結果
Fig. 11 Tracking result.

出した値を比較したものである．グラフはほぼ完全に重なっており，大きなずれは看取されない．またカメラの運動は滑らかであり，対象をスムーズに追跡していることが分かる．

最後に追跡の結果として，対象が検出された方位とカメラ方位を比較したものを**図 11** に示す．知覚モジュールとして背景差分を用いたものが (a) であり，オプティカルフローを用いたものが (b) である☆．PID 制御が持つ特性によりカメラ方位がやや遅れてはいるが，どちらの場合も対象をなめらかに追跡していることが分かる．振幅のピーク付近で対象の方位とカメラ方位が大きくずれているのは，カメラの物理的なパン・チルト角の最大値を逸脱して対象が移動したためである．

☆ 同時に実験を行うことが不可能であるため，対象の動きはこれらの実験間では異なる．

7. おわりに

我々は，時変な事象を処理の対象とするモジュールやエージェントなどが並行動作する際に有効な通信方法である，ダイナミックメモリアーキテクチャを提案した．このアーキテクチャでは，非同期に動作するモジュールどうしが通信するときに問題となる，情報の送信と受信の時刻ずれや通信遅延に起因した誤差を抑制することが可能である．また，実時間的に動作するアクティブビジョンシステムを構築する際に上記のような問題が生じることから，我々はこのようなシステムを特にダイナミックビジョンシステムと呼び，これにダイナミックメモリアーキテクチャを組み合わせた．ダイナミックビジョンシステムの例として実時間的かつ滑らかに対象の方位を追跡するシステムを 2 種類の画像解析法を用いて構築し，これによりモジュール単体でのテストやモジュールの差し替えが可能であり，システムの構築や修正が容易に行えることを確認した．

謝辞 本研究は，日本学術振興会未来開拓学術研究推進事業（JSPS-RFTF 96P00501）の補助を受けて行った．

参考文献

1) Matsuyama, T.: Cooperative Distributed Vision – Dynamic Integration of Visual Perception, Action, and Communication, *Proc. Image Understanding Workshop*, pp.365–384 (1998).
2) Shafer, S.A., Stentz, A., and Thorpe, C.E.: An Architecture for Sensor Fusion in a Mobile Robot, *Proc. IEEE Conf. on Robotics and Automation*, pp.2002–2011 (1986).
3) Thorpe, C., Herbert, M.H., Kanade, T. and Shafer, S.A.: Vision and Navigation for the Carnegie-Mellon Navlab, *IEEE Trans. PAMI*, Vol.10, No.3, pp.362–373 (1988).
4) Little, J.J. and Kam, J.: A Smart Buffer for Tracking Using Motion Data, *Proc. Computer Architecture for Machine Perception*, pp.257–266 (1993).
5) Aloimonos, Y. (Ed.): Special Issue on Purposive, Qualitative, Active Vision, *CVGIP: Image Understanding*, Vol.56, No.1 (1992).
6) Aloimonos, Y. (Ed.): *Active Perception*, Lawrence Erlbaum Associates Publisher (1993).

7) 石黒 浩：能動視覚とその応用，コンピュータビジョン：技術評論と将来展望，松山，久野，井宮（編著），pp.219–229, 新技術コミュニケーションズ (1998).
8) Weiss, L.E., Sanderson, A.C. and Neuman, C.P.: Dynamic Sensor-Based Control of Robots with Visual Feedback, *IEEE Trans. RA*, Vol.3, No.5, pp.404–417 (1987).
9) Brown, C.M.: Gaze Control with Interactions and Delays, *IEEE Trans. SMC*, Vol.20, No.1, pp.518–527 (1990).
10) 和田俊和，浮田宗伯，松山隆司：視点固定型パン・チルト・ズームカメラとその応用，電子情報通信学会論文誌，Vol.J81-D-II, No.6, pp.1182–1193 (1998).
11) 村瀬健太郎，日浦慎作，松山隆司：実時間対象検出・追跡のための知覚と行動の動的統合，情報処理学会研究報告，99-CVIM-115, pp.149–156 (1999).

(平成 12 年 3 月 6 日受付)
(平成 12 年 9 月 7 日採録)

日浦 慎作（正会員）

1972 年生．1993 年大阪大学基礎工学部制御工学科飛び級中退，1997 年同大学大学院博士課程短期修了．同年京都大学リサーチアソシエイト，1999 年大阪大学大学院基礎工学研究科助手．三次元動画像計測・処理の研究に従事．工学博士．1993 年電気関係学会関西支部連合大会奨励賞，2000 年画像センシングシンポジウム優秀論文賞受賞．電子情報通信学会会員．

村瀬健太郎（正会員）

1997 年京都大学工学部電気系学科卒業，1999 年同大学大学院工学研究科電子通信工学専攻修士課程修了．同年富士通研究所入社．画像認識システムの研究開発に従事．

松山 隆司（正会員）

1976 年京都大学大学院修士課程修了．京都大学助手，東北大学助教授，岡山大学教授を経て 1995 年より京都大学大学院電子通信工学専攻教授．現在同大学院情報学研究科知能情報学専攻教授．工学博士．画像理解，人工知能，分散協調視覚の研究に従事．1980 年情報処理学会創立 20 周年記念論文賞，1990 年人工知能学会論文賞，1993 年情報処理学会論文賞，1994 年電子情報通信学会論文賞，1995 年第 5 回国際コンピュータビジョン会議 Marr Prize，1996 年国際パターン認識連合 Fellow，1999 年電子情報通信学会論文賞，2000 年画像センシングシンポジウム優秀論文賞．人工知能学会評議員，情報処理学会理事，同学会フロンティア領域委員会委員長．