

能動視覚エージェント群による複数対象の実時間協調追跡

浮 田 宗 伯[†] 松 山 隆 司[†]

能動視覚エージェント群の協調動作による複数対象の実時間追跡システムを提案する。能動視覚エージェント (Active Vision Agent, 略して AVA) とは、ネットワーク接続された計算機とアクティブカメラの組の論理モデルである。各 AVA は、相互に交換される検出対象情報を統合し、その結果に基づいて自身の役割を適応的に変更する。こうした AVA 群の協調動作により、複雑な動的環境下における複数対象の移動を同時に追跡することが可能となる。AVA 間の実時間協調を実現するために、我々は三層構成のインタラクションアーキテクチャを提案した。各層では、並列プロセス群が効果的な協調動作のために必要な情報を交換している。この情報交換の実時間性を保証するために、ダイナミックメモリ³⁾を利用した。また実験により、提案システムが適応的に役割分担を行いながら、複数対象を実時間で追跡可能なことを確認した。

Real-Time Cooperative Multi-Target Tracking by Communicating Active Vision Agents

NORIMICHI UKITA[†] and TAKASHI MATSUYAMA[†]

This paper presents a real-time cooperative multi-target tracking system. This system consists of a group of Active Vision Agents (AVAs), where an AVA is a logical model of a network-connected computer with an active camera. All AVAs cooperatively track their target objects by dynamically interacting with each other. As a result, the system as a whole can track multiple moving objects simultaneously under complicated dynamic situations in the real world. To implement the real-time cooperation among AVAs, we designed a three-layered interaction architecture. In each layer, parallel processes mutually exchange various information for the effective cooperation. To realize the information exchange in real time, we employed the dynamic memory architecture³⁾. Experimental results demonstrate that AVAs cooperatively track their target objects in real-time while adaptively changing their roles.

1. はじめに

様々な実世界ビジョンシステム (監視システム, ITS など) を実現する上で、対象の検出・追跡は最も重要かつ基盤となる技術の一つである。それゆえ、対象追跡に関する研究は数多く行われているが、これらにはシステムの特性・機能や観測する実世界に関して多くの制約が存在する (カメラ数, カメラの機能, 追跡可能対象数など)。従来、多く研究が行われてきたのは単一対象の追跡であった。これに対して近年、複数対象の同時追跡に関する研究が盛んに行われている。しかし、広範囲における複数対象追跡を、分散カメラ群を利用して実現したシステムは、ほとんど報告されていない。中でも、自律的に動作する**複数のアクティブカメラ**を利用した**複数対象**の実時間追跡は、これまで

に実現されていない。従来のシステムにおけるカメラ数, カメラの機能 (固定もしくはアクティブカメラ), 追跡可能対象数など, システムの特性・機能に関する制約は、追跡システムの振る舞いの柔軟性・適応性を損ねてしまい、その結果として追跡システムを適用可能なタスクや環境が制限されてしまう。

そこで我々は、様々なユーザの目的に適用可能、かつ複雑なシーンの動的状況に適応可能なシステムとして、アクティブカメラ群による複数対象の実時間協調追跡システムを提案する。提案システムでは、広範囲にわたる継続的対象追跡のためのカメラ制御と、追跡対象の詳細な情報 (高解像度な対象画像) を得るためのカメラ制御を実現する。この相反する二つのカメラ制御を同時に実現し、かつ各カメラがシーンの動的状況に対して即応的に対処できることを保証するのは非常に困難な問題であり、従来の対象追跡システムでは実現不可能である。

本システムは、自律的に動作する能動視覚エージェ

[†] 京都大学大学院情報学研究科
Graduate School of Informatics, Kyoto University

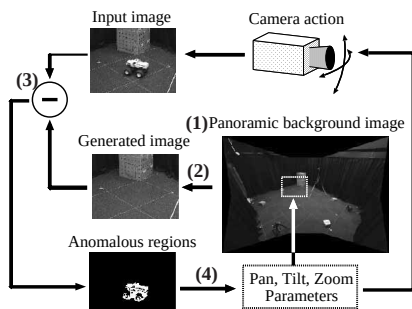


図 1 FV-PTZ カメラを用いた対象検出・追跡
Fig. 1 Object detection and tracking using an FV-PTZ camera.

ント (Active Vision Agent, 略して AVA) の集合によって構成されている。AVA とは、アクティブカメラを備え、相互にネットワーク結合された計算機の論理モデルである。観測空間中に分散配置した AVA 群を利用することにより、追跡対象の 3 次元情報の獲得、広範囲における継続的追跡、などが可能となる。

上記のシステム構成において、目的とする実時間協調追跡を実現するためには、様々な問題を解決する必要がある (例: 能動観測時の対象検出に適したアクティブカメラの開発²⁾, アクティブカメラによる実時間対象追跡の実現³⁾)。本研究では、AVA 間の実時間協調動作の実現に焦点をおいた。

我々は、AVA 間の実時間協調を実現するために三層構成のインタラクションアーキテクチャを提案する。各層では、その層で動作している並列プロセス群の協調動作に必要な対象情報が交換されている。この情報交換を実時間で実現するために、我々はダイナミックメモリ³⁾を利用した。各層における動的インタラクションの結果、システム全体として複雑かつ動的な環境下における複数移動対象の追跡が可能となる。

2. 協調追跡

2.1 AVA の構成と機能

各 AVA は、計算機と視点固定型パン・チルト・ズーム (Fixed-Viewpoint Pan-Tilt-Zoom, 略して FV-PTZ) カメラ²⁾によって構成されている。FV-PTZ カメラとは、任意の視線方向・ズームにおいて常にカメラの投影中心が固定されるように設計されたアクティブカメラである。この FV-PTZ カメラの特性を利用することにより、各 AVA は移動対象を独立に検出・追跡可能である (図 1)。

- (1) 背景のワイドパノラマ画像を生成する: FV-PTZ カメラの特性を利用すると、任意のパン・チルト・ズームで撮影された画像を貼り合せて、一

枚のワイドパノラマ画像を容易に生成できる。

- (2) 画像撮影時のパン・チルト・ズームに対応する背景画像をワイドパノラマ画像から抽出する。
- (3) 生成した背景画像と観測画像との差分を得る。
- (4) 差分画像中に背景以外の物体が検出された場合、検出領域の中から一つの追跡対象を選び、その領域を注視するようにカメラを制御する。

各 AVA は自身のダイナミクスに従い自律的に動作しているため、カメラ間の画像撮影のタイミングは非同期である。一方、文献 4) など多くの分散カメラシステムでは、同期信号発生装置を利用して全カメラが同期撮影を行っている。しかし、動的かつ多様なシーンの状況に対応し、複雑なタスクに応じて各カメラの役割を適応的に変更可能なシステムを実現するためには、各カメラは自律的に動作・観測を行うべきである。

一般に、エージェントには以下の二種類が存在する。**ソフトウェアエージェント**⁵⁾ 物理的な身体を持たない仮想エージェント。各エージェントは、システム内のデータ (検出対象の情報など) と対応づけられている。

実世界エージェント⁶⁾ 自身が制御可能な物理的身体 (アクティブカメラ、移動ロボットなど) を持つエージェント。システム内のエージェントと実世界の中の身体の間には、一対一の対応関係がある。我々は「実世界で動作する知的システムは、環境と相互にインタラクションを行うための身体が必要である」と考える。そこで本論文では、物理的身体を持つ実世界エージェントを単にエージェントと呼ぶ。

複数カメラによる複数対象の同時追跡に関しては、従来もいくつかの研究がなされてきたが、それらのほとんどはソフトウェアエージェントを利用したものである。文献 7) では、(1) ソフトウェアエージェントが、各検出対象に対応するように定義されており、(2) 全てのカメラは、ソフトウェアエージェント群によって共有されている。このような定義では、各ソフトウェアエージェントは、対象追跡を行うために全てのカメラの検出対象情報をチェックする必要がある。さらに本質的な問題として、アクティブカメラを利用する場合、複数のソフトウェアエージェントがそれぞれの追跡対象を注視するために、一つのカメラに対して同時に相反するカメラ制御を要求する可能性がある。

一方、本システムにおいて、エージェント (AVA) は単一のカメラに対応している。よって、全ての AVA は自身の備えるカメラを制御することにより対象を追跡可能である。以上の議論から、対象追跡というタスクに対しては、我々が提案する「エージェントとカメ

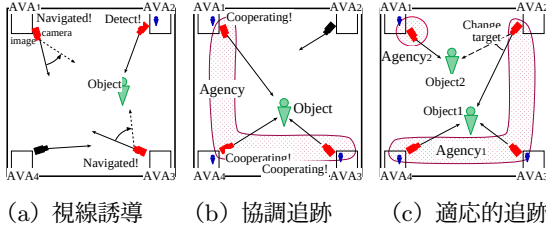


図2 AVA 群による協調追跡（網掛け領域はエージェンシを表し、同じ領域内の AVA は同一エージェンシに属する。）

Fig. 2 Basic scheme for cooperative tracking.

ラを一对一対応させる」というエージェントの定義が適していると考ええる。

2.2 協調追跡のための基本タスク

本システムでは、多数の AVA がシーン中に設置されている。以下に、システム全体の基本タスクを示す。

- (1) 初期状態において、各 AVA は独立に観測シーン中の対象を探索する。
- (2) ある AVA が対象を検出すると、この AVA は他の AVA の視線をこの対象へ向ける（図2 (a)）。
- (3) 同一対象を注視する AVA 群は、その対象を協調的に追跡する（図2 (b)）。同一対象を追跡する AVA の集合を**エージェンシ**と呼ぶ。
- (4) 対象の移動に応じて、各 AVA は自身の追跡対象を適応的に変更する（図2 (c)）。
- (5) 追跡対象が観測シーンから出ると、その対象を追跡していた AVA は、その時点のシーンの状況に応じて、再度対象探索を行うか、それとも他のエージェンシに参加するかを決定する。

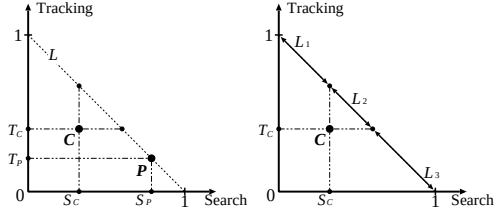
上記の協調追跡を実現するために、我々は以下の問題を解決しなければならない。

複数対象同定 各対象を注視するために、検出した複数の対象をそれぞれ同定する必要がある。

実時間即応処理 シーンのダイナミクス（対象の移動）に対応するために、全ての処理を実時間でを行い、シーンの状況の変化に対して即応的に対処する必要がある。

適応的役割分担 目的とするタスクを実行するため、我々は AVA に対して二段階の役割分担を行う必要がある。(1) 対象の探索と追跡を同時に行うために、対象追跡が行われている間も新しい対象を探索するための AVA を用意しなければならない。(2) 各対象がその対象の注視に適した AVA 群によって追跡されるように、各 AVA に対して追跡対象を割当てなければならない。

我々は、これらの問題を AVA やエージェンシ間の実時間協調動作によって解決する。



(a) 現在状態とタスク制約条件 (b) システムの三状態

図3 システム状態記述グラフ

Fig. 3 System state graph.

3. タスクの仕様記述

対象追跡システムは、まず「シーン中のどこに対象が存在するか」を探索する必要がある。この役割を**サーチ**と呼ぶ。そして、対象を発見すると「検出対象を注視」して情報を収集する。この役割を**トラッキング**と呼ぶ。また、複数の対象が検出された場合、与えられたタスクに対する各対象の重要度に基づいて「情報を必要とする対象を重点的に注視する」という機能も求められる。

本システムでは、ユーザが追跡システムに与えるタスクを設定するために、以下のパラメータ（タスク制約条件、対象重要度、効用度関数）を用いる。

3.1 タスク制約条件

サーチを行う AVA を**フリーランサ**と呼ぶ。フリーランサは、自律的にシーンの広範囲探索を行い、新たな対象の検出を役割とする。また、トラッキングを行う AVA 群は、追跡対象と一対一に対応したエージェンシを生成する。エージェンシに所属する AVA を**メンバ**と呼ぶ。

サーチとトラッキングの組み合わせによるシステムの多様な振る舞いを図3に示す2軸のグラフによって表現する。このグラフを**システム状態記述グラフ**と呼ぶ。横軸、縦軸は、それぞれサーチとトラッキングに従事する AVA の数の割合を表しており、それぞれを**サーチレベル**、**トラッキングレベル**と呼ぶ。

定義1 (サーチレベル、トラッキングレベル)

$$\text{Search-level} = \frac{\text{フリーランサの総数}}{\text{AVA の総数}} \quad (1)$$

$$\text{Tracking-level} = \frac{\text{メンバの総数}}{\text{AVA の総数}} \quad (2)$$

$$0 \leq \text{Search-level} \leq 1 \quad (3)$$

$$0 \leq \text{Tracking-level} \leq 1 \quad (4)$$

システム状態記述グラフ上で、現在状態とタスク制約条件を次のように定義する。

定義2 (現在状態 $P(S_P, T_P)$) ある瞬間における

システム全体のサーチレベル (S_P) とトラッキングレベル (T_P) を表す (図 3 (a) 中の P)。現在状態 P がとりうる値の範囲は、図 3 (a) 中の直線 L 上である。即ち、常に $(S_P + T_P) = 1$ である。

定義 3 (タスク制約条件 $C(S_C, T_C)$) ユーザによりタスクにあわせて設定される定数であり、動作中にシステムが最低限確保すべきサーチレベル (S_C) とトラッキングレベル (T_C) を表す (図 3 (a) 中の C)。但し、 $0 \leq (S_C + T_C) \leq 1$ 。即ち、総 AVA の $(1 - (S_C + T_C)) \times 100[\%]$ の AVA はタスク制約条件に影響されることなく、状況のみに応じて自由にサーチとトラッキングの間で役割を変更可能である。

システムの現在状態とタスク制約条件の関係に対して、以下に示す 3 つの状態が存在する。

サーチ欠如状態 $T_C < T_P$, かつ $S_C > S_P$ 。即ち図 3(b) 中の線分 L_1 上に現在状態 P が存在する時。

タスク充足状態 $T_C \leq T_P$, かつ $S_C \leq S_P$ 。即ち図 3(b) 中の線分 L_2 上に現在状態 P が存在する時。

トラッキング欠如状態 $T_C > T_P$, かつ $S_C < S_P$ 。即ち図 3(b) 中の線分 L_3 上に現在状態 P が存在する時。

システムがタスク充足状態以外の場合、各 AVA は現在状態がタスク制約条件を満たすように、自身の役割をサーチとトラッキングの間で動的に変化させる。

本システムでは、各 AVA は他者との通信時にそれぞれの現在の役割に関する情報を交換しており、その情報に基づいてシステムの現在状態を推定する。

3.2 対象重要度

AVA とエージェントの対象同定能力により判別可能な対象のクラス毎に、対象重要度を指定する。

定義 4 (対象重要度) 各クラスに属する対象の重要度を表す尺度で、0 以上 1 以下の定数である。エージェント P に所属するメンバ数を M_P 、エージェント P の追跡対象の対象重要度を I_P とすると、 $M_P = (AVA \text{ の総数}) \times (I_P / S)$ 。但し、 S は全エージェント $1, \dots, A$ の対象重要度 $I_1, \dots, I_A$ の総和である。

3.3 効用度関数

タスク制約条件と対象重要度の制約の下で、各 AVA は自身の役割を自由に変更可能である。この適応的な役割変更のための指標は、効用度関数としてシステムに与えられる。各 AVA は、タスク制約条件と対象重要度による AVA 数の制約を守りながら、効用度関数の評価値を上昇させるように自身の役割を変更する。本システムにおける効用度関数は、以下のサーチ適性とトラッキング適性の和により表現される。

サーチ適性 フリーランサ f のサーチに対する適性を

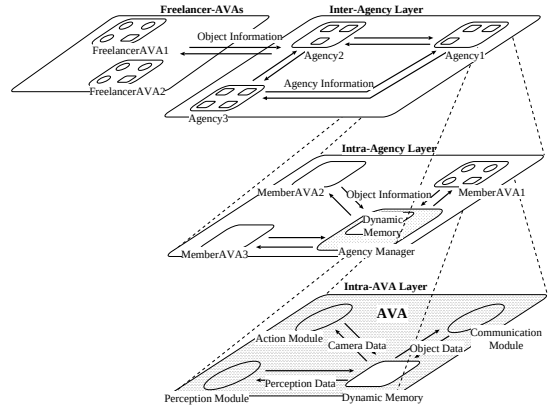


図 4 システム内の 3 つの階層 (最下層: Intra-AVA, 中間層: Intra-agency, 最上層: Inter-agency)

Fig. 4 Three layers in the system (bottom: intra-AVA, middle: intra-agency, top: inter-agency layers).

表す関数 V_{S_f} 。

トラッキング適性 メンバ m が追跡対象 n のトラッキングを行う際の適性を表す関数 $V_{T_m}^n$ 。

この効用度関数は、追跡システムに与えられるタスクに応じて、ユーザが適当に設定可能である^{*}。

4. 三層構成インタラクションによる協調追跡

本システムでは、複数の自律的な処理主体が動的にインタラクションを行いつつ協調的に動作することにより、システム全体が一つの追跡システムとして振る舞う。このように、システムを多数の自律的な処理主体の集合として構成することにより、システム全体の複雑な動作を処理主体間のインタラクションにより表現可能である。その結果、システム全体の設計を各処理主体の設計へと簡単化できる。また、処理主体間の動的なインタラクションによって創発されるシステム全体の状態数・制御フローの分岐数は、処理主体の増加に伴い組合わ式的に増大する。この性質を利用することにより、実世界の多様かつ複雑な状況に対応可能な柔軟なシステムを構築可能である¹⁾。

4.1 システムの階層

対象追跡を行うためには、各カメラによって観測された対象情報を時間的・空間的に対応付ける、即ち対象同定を継続する必要がある。本システムでは、対象同定のために行われる動的インタラクションを、処理主体や交換される情報の種類によって 3 つの層に分けた (図 4)。最下層から順に Intra-AVA 層, Intra-agency 層, Inter-agency 層と呼ばれ、それぞれ一つの AVA,

^{*} 6.1 節中の評価実験において例を示す。

一つのエージェンシ、システム全体に相当している。各層では、交換される情報の種類に応じた対象同定が実行され、その対象同定の成否に応じて、複数対象の協調追跡を実行するためのプロトコルが実行される。

本システムのような実世界システムを実現する上で、画像処理や対象同定の失敗は不可避な問題である。本研究では、協調プロトコルの実行に伴い (1) エージェントやエージェンシ間の情報交換によってシステム全体としての誤りの発生を抑え、(2) システムが誤った振る舞いを起こしてしまった場合でも、後の観測・対象同定の結果に基づいてその誤りを修正し、継続的に対象追跡が行われるシステム設計を目的としている。

以下では、まず自律処理主体間の実時間インタラクションのための基本的な機能について紹介し、次に各層でのインタラクションについて述べる。

4.2 ダイナミックメモリ：実時間非同期インタラクションのためのアーキテクチャ

自律処理主体（並列プロセス）間の実時間協調を実現するためには、プロセス群は相互の情報を非同期的かつ即応的に交換しなければならない。こうしたプロセス間インタラクションを実現するためのアーキテクチャとして**ダイナミックメモリ**³⁾が提案されている。ダイナミックメモリは、並列プロセス群が同時に操作可能な共有メモリであり、通常の時系列変数値（例：カメラのパン・チルト角の履歴、追跡対象の移動軌跡）に加えて、未来における予測値も取り扱い可能である。ダイナミックメモリによる動的情報の交換は、移動対象の追跡のような実時間かつ即応的な処理を必要とするシステムの実現に対して非常に効果的である。

一つのダイナミックメモリを共有する各プロセスは、自身が取り扱う状態変数を任意のタイミングでダイナミックメモリに書き込む。書き込まれた情報は、時系列情報として複数プロセス間で共有されるので、全プロセスの時計情報は一致していなければならない。本システムでは、(1) 同一計算機内の全プロセスはその計算機の内部時計を共有し、(2) 全ての計算機の内部時計を同期させることにより、システム内の全プロセスの時計情報を一致させた。

ダイナミックメモリに対する読み込み・書き込み操作は以下のように定義される。

書き込み操作 時刻 t において、プロセスが変数値 v を得ると、 (t, v) の組をダイナミックメモリに書き込む。このような時系列変数値はプロセスのダイナミクスにしたがった不定期なタイミングで時々刻々得られる値であり、ダイナミックメモリに離散時系列値として記録される（図 5 中の白い点群）。

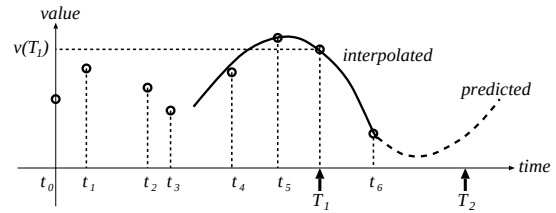


図 5 ダイナミックメモリにおける時系列情報の表現
Fig. 5 Representation of a time varying variable in the dynamic memory.

読み込み操作

時系列補間： 書き込み操作と並列した任意のタイミングで、各プロセスはダイナミックメモリに記録されている値を読み込み可能である。また、各変数は時系列情報として記録されているので、要求時刻における値が記録されていない場合でも、ダイナミックメモリが離散値の補間を行うことにより任意の時刻における値を返すことが可能である（例：図 5 中の T_1 ）。

未来予測： 各プロセスの動作のタイミングによっては、書き込みを行うプロセスがまだ記録していない未来の値を他のプロセスが要求するケースが起こり得る。このような場合、ダイナミックメモリは、記録値の履歴に基づいて要求時刻における予測値を返す（例：図 5 中の T_2 ）。

ダイナミックメモリに記録される変数は、ある事象のダイナミクス（例：アクティブカメラのパン・チルト・ズームパラメータ）を表現している。よって、時系列補間と予測のための関数は、各事象のダイナミクスを適切に表現していなければならない。そこで、文献 3) で述べられているように、各事象のダイナミクスを表す関数を、事前のキャリブレーションによって適切に定義しておく必要がある。

4.3 Intra-AVA 層におけるインタラクション

図 4 中の最下層（Intra-AVA 層）は、AVA の各機能に対応する視覚、行動、通信のモジュール群とダイナミックメモリにより構成されている。各モジュールはダイナミックメモリ介してインタラクションを行い、その結果として一つの AVA の動作が発現する。

視覚： 画像の観測と観測画像中の対象検出を繰り返す。時刻 $t+1$ において対象 $1, \dots, N$ を検出すると、各対象の方位線（カメラと検出対象を結ぶ 3 次元直線） $L^1(t+1), \dots, L^N(t+1)$ を計算し、検出対象情報としてダイナミックメモリに記録する。方位線は、カメラの投影中心の 3 次元座標 (x_c, y_c, z_c) を通る 3 次元ベクトル $x_i x_b + y_i y_b + f z_b$ によって表される。

但し、 (x_i, y_i) は観測画像中における検出対象領域の重心座標、 f はカメラの焦点距離、 x_b, y_b, z_b はそれぞれ 3 次元空間中における画像平面の x 座標、 y 座標、カメラの光軸方向を表す単位ベクトルである。そして、追跡対象を注視するために、ダイナミックメモリに記録されている追跡対象の移動履歴と検出対象情報を比較する。まず、時刻 $t+1$ における追跡対象の方位線 $\bar{L}(t+1)$ を推定し、 $L^1(t+1), \dots, L^N(t+1)$ と比較する。この比較の結果、 $\bar{L}(t+1)$ との角度差が最小である $L^x(t+1)$ を $t+1$ における追跡対象の検出結果と見なす。但し、 $x \in \{1, \dots, N\}$ である。即ち、時刻 $t+1$ における追跡対象の方位線 $\hat{L}(t+1) = L^x(t+1)$ である。方位線 $L^x(t+1)$ は、追跡対象情報としてダイナミックメモリへと記録される。

行動：フリーランサの行動モジュールは、予め定められた軌跡に沿ってカメラの視線方向を制御し、広範囲探索を行う。

メンバの行動モジュールは、追跡対象を注視するようにカメラを制御しなければならない。行動モジュールは、ダイナミックメモリ中に記録されている以下の二種類の対象情報を参照してカメラ制御を行う。

- 視覚モジュールが記録する追跡対象の方位線 $\hat{L}(t)$ 。
- 通信モジュールが記録する追跡対象の 3 次元位置 $\hat{P}(t)$ 。エージェンシは複数のメンバを伴って協調追跡を行う際、追跡対象の 3 次元位置 $\hat{P}(t)$ を計測しており、その値を自身のメンバに送信する（詳細は後述）。

これら二種類の対象情報は、ダイナミックメモリに記録されている $\hat{P}(t)$ の最新時刻 t_{new} と現時刻 now との関係に応じて参照される。 $now < t_{new}$ 、または $(now - t_{new})$ が閾値以下の場合、即ちダイナミックメモリによる時系列補間が可能、または妥当な未来予測結果の獲得が可能である場合には、行動モジュールは $\hat{P}(now)$ を読み込む。それ以外の場合は、追跡対象の方位線 $\hat{L}(now)$ を読み込む。

通信：通信モジュールによりネットワークを介して交換される情報には、検出対象情報 ($\hat{L}(t)$, $\hat{P}(t)$) と後述する協調追跡のための様々なプロトコルに関わるメッセージの二種類が存在する。AVA は、自身の役割に応じて以下の情報を他のプロセスに送信する。

- フリーランサは、自身の検出対象情報をブロードキャストする。このブロードキャストメッセージは、エージェンシによってのみ受信、処理される。
- メンバは、検出対象情報を自身の所属するエージェンシに対してのみ送信する。

ダイナミックメモリ：協調して一つの AVA として

働くために、視覚、行動、通信の各モジュールはそれぞれの持つ時系列情報（即ち、視覚：検出対象の方位線、行動：カメラの視線方向・ズーム値、通信：受信した対象の 3 次元位置）を動的に交換する必要がある。この情報交換は、各 AVA 内のダイナミックメモリを利用することにより、任意のタイミングで非同期に行われる。

4.4 Intra-agency 層におけるインタラクション

4.4.1 対象同定

先に述べたように、同時に同一対象を追跡する AVA 群はエージェンシを組織する。図 4 中の中間層 (Intra-agency 層) は、同一エージェンシに所属するメンバによって構成される。本システムでは、エージェンシと追跡対象は、常に一対一に対応付けられなければならない。この対応関係を動的に生成、かつ継続的に維持するために、同一エージェンシ内のメンバは対象検出結果を交換し、追跡対象の同定を行う。Intra-agency 層では、以下の 2 種類の対象同定を行う必要がある。

4.4.1.1 空間的对象同定

エージェンシのメンバ $1, \dots, M$ が、それぞれ時刻 $t_1, \dots, t_M$ において画像観測を行った場合、各メンバが検出した対象の方位線群 $\{L_1^i(t_1)|i = 1, \dots, N_1\}, \dots, \{L_M^i(t_M)|i = 1, \dots, N_M\}$ の間で、同一対象を表す方位線同士を対応付けなければならない。但し、 $\{L_m^i(t_m)|i = 1, \dots, N_m\}$ は、メンバ m が時刻 t_m において検出した対象群 $1, \dots, N_m$ の方位線の集合である。本システムにおける空間的对象同定は、異なる AVA が検出した対象の方位線間の距離によって成否が決まる。即ち、方位線間の距離が閾値以下であれば同定成功、それ以外は同定失敗と見なす。

但し、各 AVA は自律的に画像撮影を行っているため、空間的对象同定に使用される検出対象情報は、それぞれ異った時刻における対象の方位線を表している。また、ネットワークを介したメッセージ送信は、各 AVA が送信した検出対象情報と対象同定のタイミングのずれを引き起こし、予測不能な問題が生じてしまう可能性がある。よって、各 AVA から送信された対象情報をそのまま AVA 間で比較してしまうと、対象同定の結果が不安定になってしまう。

この問題を解決するために、我々は**仮想同期**によって各方位線の観測タイミングを仮想的に同期させる（詳細は 4.4.2 節にて述べる）。ある時刻 T において同期がとられた方位線群 $\{L_1^i(T)|i = 1, \dots, N_1\}, \dots, \{L_M^i(T)|i = 1, \dots, N_M\}$ を相互に比較することによって、安定な同定結果を得ることが可能となる。また、同定成功した方位線群の交点（また

は、方位線群からの距離が最も小さい最近点)を各対象の3次元位置 $\{P_i(T)|i=1,\dots,N\}$ と見なす。

4.4.1.2 時間的对象同定

空間的对象同定の結果からは、画像観測の瞬間にくつの対象が存在し、それぞれシーン中のどこに存在するかという情報のみしか得られない。また、エージェントにとって必要な対象情報は、自身の追跡対象情報のみであり、他の対象情報は破棄されるべきである。よって、追跡対象の注視を続けるために、追跡対象の移動軌跡と空間的对象同定により得られた対象情報の間で同定を行う必要がある。

空間的对象同定が成功して、複数対象の3次元位置 $\{P_i(T)|i=1,\dots,N\}$ が推定できた場合、追跡対象の3次元位置の履歴と $\{P_i(T)|i=1,\dots,N\}$ を比較し、時刻 T における追跡対象の3次元位置 $P_x(T)$ を決定する。但し、 $x \in \{1,\dots,N\}$ である。また、空間的对象同定において $P_x(T)$ と交差する方位線を検出したメンバは、時間的对象同定が成功、即ち正しく追跡対象を検出していると見なす。

一方、空間的对象同定が失敗して、検出対象の3次元位置が全く求まらなかった場合、各メンバが検出した対象群の方位線と追跡対象の3次元位置の履歴を比較して、追跡対象に対応する方位線を決定する。

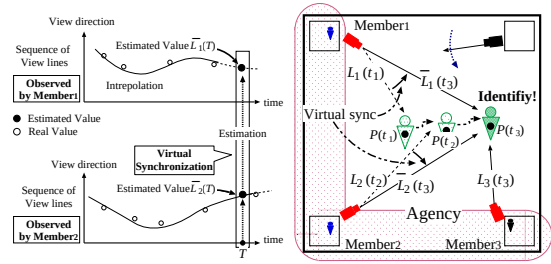
4.4.2 仮想同期

4.4.2.1 空間的对象同定のための仮想同期

各AVAによる非同期な画像観測と予測不能なネットワーク遅延は、空間的对象同定の結果を不安定ににしてしまう。過去の分散自律カメラ群による対象追跡に関する研究では、こうした問題に対して、以下のように対処している。

- 文献7)8)では、各カメラから集められている最新の検出対象情報が同時刻の観測情報であると仮定している。
- 文献9)では、各対象情報に観測時刻を表すタイムスタンプが含まれる。異なるカメラの検出対象情報間で、観測時刻の差が閾値以下の場合、これらの対象情報は同時に観測された情報と見なされる。しかし、これらの近似的手法は、対象が複数検出される複雑な環境下や、ネットワークの負荷によって破綻してしまう。

この問題を解決するために、本システムでは、Intra-AVA層と同様にIntra-agency層にもダイナミックメモリを用意し、メンバの検出結果を時系列データとして記録する。即ち、ダイナミックメモリを使えば、記録した時系列データから任意の時刻 t における値を推定できるため、非同期に観測された複数の時系列



(a) 仮想同期値の読み出し (b) 空間的对象同定

図6 メンバAVA間の空間的对象同定のための仮想同期
Fig. 6 Virtual synchronization for spatial object identification among member-AVAs in the same agency.

データから同一時刻の値を求め、それらの値を比較すれば良い。この機能を仮想同期と呼ぶ。

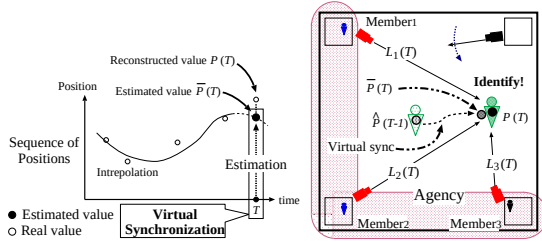
例えば図6(a)の例では、メンバ1, 2の対象検出結果(図中の白点)が共通のダイナミックメモリに記録される。時刻 T において空間的对象同定を行うには、 $t = T$ としてダイナミックメモリからメンバ1, 2の方位線データ ($\bar{L}_1(T)$ と $\bar{L}_2(T)$) を読み出せば良い。この時刻指定読み出しによって、同一時刻の対象方位線が得られ、それらを用いて対象同定を行うことにより、安定な空間的对象同定が可能となる。

具体的な空間的对象同定は、以下のように行われる。まず、エージェントの形成(4.4.3節で述べる)と同時にエージェンシマネージャが生成される。エージェンシマネージャは、エージェンシを代表する自律主体として動作する独立プロセスであり、(1)エージェンシ毎に用意されるIntra-agency層のダイナミックメモリの管理、(2)AVAや他のエージェンシ間の情報交換や対象同定、(3)エージェンシの動的な再編成を行う。エージェンシマネージャは、各メンバから検出対象情報 $\{L_m^i(t_m)|i=1,\dots,N_m\}$ を受信すると、その値をダイナミックメモリに記録する。空間的对象同定を行う際、エージェンシマネージャは全メンバの観測結果を仮想同期した結果 $\{\bar{L}_1^i(T)|i=1,\dots,N_1\}, \dots, \{\bar{L}_M^i(T)|i=1,\dots,N_M\}$ を相互に比較する。比較の結果、 $\bar{L}_p^i(T)$ と $\bar{L}_q^j(T)$ 間の距離が閾値以下の場合、メンバ p の検出対象 i とメンバ q の検出対象 j は同一対象 x に対応すると見なす。また、 $L_p^i(T)$ と $L_q^j(T)$ の交点 $P_x(T)$ を時刻 T における対象 x の3次元位置と見なす。

図6(b)の例では、時刻 t_3 にあわせてメンバ1, 2, 3の検出結果を仮想同期した値 $\bar{L}_1(t_3)$, $\bar{L}_2(t_3)$, $L_3(t_3)$ を比較し、安定な対象同定が実現できている。

4.4.2.2 時間的对象同定のための仮想同期

時間的对象同定のためには、時刻 T における検出対象情報と追跡対象情報の履歴との比較が必要になる。



(a) 仮想同期値の読み出し (b) 時間的対象同定

図 7 Intra-agency 層における時間的対象同定のための仮想同期
Fig. 7 Virtual synchronization for temporal object identification.

しかし、時刻 T における検出対象情報と履歴中の T とは異なる時刻における対象情報をそのまま比較すると異なる時刻における情報の比較になってしまい、対象同定結果は不安定になってしまう。この問題も、エージェンシマネージャが追跡対象の 3 次元位置 $\hat{P}(t)$ の履歴をダイナミックメモリに記録しておき、仮想同期を行うことにより解決できる。

空間的対象同定成功時 ダイナミックメモリに記録された追跡対象の 3 次元位置の時系列データから、時刻 T における追跡対象の 3 次元位置 $\bar{P}(T)$ を推定し、 $\{P_i(T)|i = 1, \dots, N\}$ と比較する。 $\bar{P}(T)$ との距離が最小の点 $P_x(T)$ を選び、その距離が閾値以下の場合、 $P_x(T)$ を時刻 T における追跡対象の 3 次元位置と見なす。

空間的対象同定失敗時 各メンバが検出した対象の方位線群 $\{L_x^i(t_1)|i = 1, \dots, N_1\}$, $\dots$, $\{L_M^i(t_M)|i = 1, \dots, N_M\}$ と追跡対象の移動軌跡を比較する。検出対象の方位線 $L_x^y(t_x)$ と追跡対象の移動軌跡を比較する場合、ダイナミックメモリから時刻 t_x における追跡対象の 3 次元位置 $\bar{P}(t_x)$ を推定し、 $\bar{P}(t_x)$ と $L_x^y(t_x)$ との間の距離を計算する。また、 $L_x^y(t_x)$ を追跡対象と同定した場合、 $\bar{P}(t_x)$ を $L_x^y(t_x)$ に投影した点を時刻 t_x における追跡対象の 3 次元位置 $\hat{P}(t_x)$ と見なす。但し、 $x \in \{1, \dots, M\}$, $y \in \{1, \dots, N_x\}$ である。このように、仮想同期によって時間的対象同定も安定化できる。

図 7 は、空間的対象同定が成功した場合の例を示している。追跡対象の 3 次元位置の履歴から、エージェンシマネージャは時刻 T における追跡対象の 3 次元位置を推定可能である (図 7 (a))。図 7 (b) では、空間的対象同定により時刻 T における検出対象の 3 次元位置 $P(T)$ が復元されている。この場合、エージェンシマネージャは同時刻における追跡対象の 3 次元位

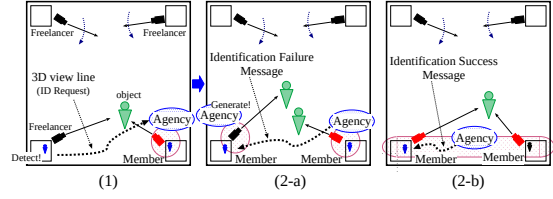


図 8 Agency Formation
Fig. 8 Agency formation.

置 $\bar{P}(T)$ を推定し、 $P(T)$ と $\bar{P}(T)$ を比較して時間的対象同定を行う。

以上の対象同定の結果に応じて、以下の 3 種類の協調プロトコルで定義されるインタラクションが実行される。

4.4.3 Agency Formation プロトコル

フリーランサによるエージェンシの生成、フリーランサのエージェンシへの参加を定義するプロトコル。システムの初期状態において、全 AVA はフリーランサとして独立に対象探索を行う。フリーランサが新たに対象を検出すると、既存の全エージェンシに対して各エージェンシの追跡対象と新たに検出された対象との間の対象同定を依頼する (図 8, 1)。本システムにおいて、こうした AVA とエージェンシ間の対象同定は、エージェンシの持つ追跡対象情報の種類によって以下の二通りに分類される。

追跡対象の 3 次元位置が既知の場合 追跡対象の 3 次元位置と新検出対象を表す方位線間の距離が閾値以下の場合、二つの対象情報は同定される。

追跡対象の 3 次元位置が未知の場合 追跡対象を表す方位線と新検出対象を表す方位線間の距離が閾値以下の場合、二つの対象情報は同定される。この同定の成否に応じて、フリーランサは以下のように振る舞う。

同定が成功したエージェンシが存在しない場合 検出対象を追跡するエージェンシマネージャを起動し、そのエージェンシに加わる (図 8, 2-a.)。

同定が成功したエージェンシが存在する場合 同定が成功したエージェンシに加わる (図 8, 2-b.)。

但し、システムの現在状態とタスク制約条件の間関係によっては、Agency Formation による新たなエージェンシの生成やフリーランサのエージェンシへの参加は行われない；サーチに必要な数のフリーランサが存在しない場合、フリーランサ数を減らすことが許されない。

また、対象の移動に伴い「どの AVA がフリーランサであり、どの AVA がメンバであるべきか」という

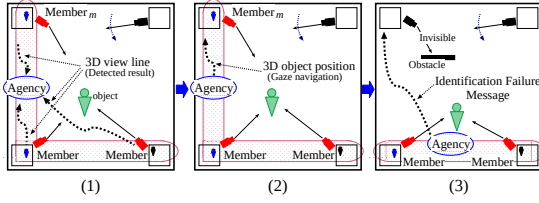


図 9 Agency Maintenance
Fig. 9 Agency Maintenance.

状況は変化する．こうした動的状況に対応するため，エージェンシマネージャ P は，フリーランサ f がエージェンシ P の追跡対象を検出している状況下で以下の基準が満たされた場合，フリーランサ f と自身のメンバ m を交換する．

$$(V_{S_f}/V_{T_f^P}) < W_F(V_{S_m}/V_{T_m^P}) \quad (5)$$

但し， V_{S_i} , $V_{T_i^P}$ は，それぞれ AVA_i のサーチ適性とトラッキング適性（3.3 節参照）， W_F は重み定数である．定数 W_F が大きいほど，フリーランサとメンバの交換が反射的に実行される．

4.4.4 Agency Maintenance プロトコル

エージェンシによる協調追跡と，メンバの脱退・エージェンシの消滅を含めたエージェンシの管理を定義するプロトコル．

エージェンシ生成後，エージェンシマネージャは追跡対象の同定と 3 次元位置計算のため，メンバが検出した対象の空間的・時間的対象同定を繰り返す（図 9, 1.）．メンバ m の時間的対象同定が失敗した場合（メンバ m が追跡対象を見失ったり，別対象を追跡している場合），エージェンシマネージャは追跡対象の 3 次元位置をメンバ m に送信し，その視線方向を追跡対象に向けて誘導する（図 9, 2.）．更に，この同定失敗が一定時間以上続いた場合，メンバ m をエージェンシから除く（図 9, 3.）．そして，全メンバがエージェンシから放出されると，エージェンシマネージャは消滅し，エージェンシが解散する．

また，システムの現在状態とタスク制約条件の間の関係に応じて，エージェンシマネージャは自身のメンバ数を調整する；サーチに必要なフリーランサ数が確保されていない場合，エージェンシマネージャはメンバを放出してフリーランサ数を増加させる．メンバ放出の際には，エージェンシマネージャが各メンバのサーチ適性とトラッキング適性を比較し，サーチ適性の割合（（サーチ適性）/（トラッキング適性））が最も大きいメンバを放出する．但し，（1）複数のエージェンシマネージャが同時にメンバを放出してしまうことに

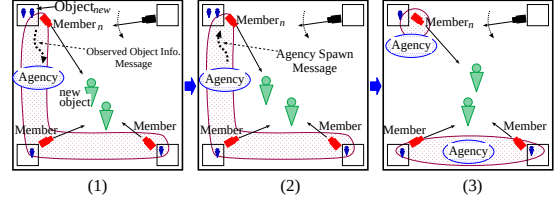


図 10 Agency Spawning
Fig. 10 Agency spawning.

よって，フリーランサ数が過剰に増加することの無いように，また（2）追跡対象の対象重要度が小さいエージェンシがメンバを放出しやすいように，エージェンシマネージャ P は，以下の基準を満たした場合にのみ自身のメンバを放出する．

$$W_M(t_P - t_S)(S_C - S_P) > (I_P/S) \quad (6)$$

但し， S_C , S_P はシステム状態記述グラフ中のサーチレベル， t_P はその瞬間の時刻， t_S は $(S_C > S_P)$ になった時刻， W_M は重み定数， I_i はエージェンシ i の追跡対象の対象重要度， S は全エージェンシの追跡対象の対象重要度の総和である．定数 W_M が大きいほど各エージェンシがメンバ放出するタイミングが早まり，システムの振る舞いが反射的になる．

4.4.5 Agency Spawning プロトコル

既存のエージェンシから新たなエージェンシを生成する方法を定義するプロトコル．

時間的対象同定の結果，メンバ n が検出した対象方位線の一つとエージェンシの追跡対象との同定に失敗した場合，即ちメンバ n の検出対象群中に追跡対象以外の対象 new が存在した場合（図 10, 1.），エージェンシマネージャは他の全エージェンシに対して，対象 new と各エージェンシが追跡している対象との間の同定を依頼する．同定依頼を受けたエージェンシは，同定結果を依頼元のエージェンシマネージャに返信する．この対象同定が全て失敗した場合，即ち対象 new に対応するエージェンシが存在しないと判定された場合，エージェンシマネージャはメンバ n に新たなエージェンシの生成（図 10, 2.）と，そのエージェンシへの移籍（図 10, 3.）を命令するメッセージを送る．

4.5 Inter-agency 層におけるインタラクション

Inter-agency 層は，システム内に存在する全てのエージェンシによって構成されている．複数対象追跡において，対象の多様な行動に適応して追跡を継続するためには，対象の移動やメンバの能力を考慮にいれ，エージェンシ間でメンバを交換する必要がある．こうした動的なエージェンシの再編成を効率的に実行するため，エージェンシ間で追跡対象情報（3 次元位置，

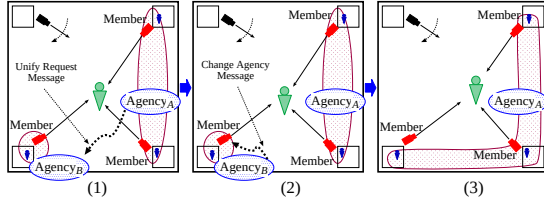


図 11 Agency Unification
Fig. 11 Agency unification.

対象重要度)とメンバ情報(所属メンバ数, 各メンバのカメラ位置・視線方向情報)が相互に交換される。これらの情報を**エージェンシ情報**と呼ぶ。

エージェンシマネージャは, 時間的対象同定により新たな追跡対象情報を得ると, エージェンシ情報を全てのエージェンシに対して送信する。このエージェンシ情報を受信したエージェンシマネージャ i は, 受信した対象と自身の追跡対象の3次元位置を比較して空間的対象同定を行う。この3次元位置の比較による対象同定は, 各3次元位置の観測時刻が異なると不安定になってしまう。しかし, この問題も Intra-agency 層における対象同定と同様, 仮想同期により解決できる。即ち, エージェンシマネージャ i は, ダイナミックメモリに記録している自身の追跡対象の3次元位置の履歴を参照して, 受信した対象と自身の追跡対象の3次元位置を仮想同期することが可能である。

上記の対象同定の結果に応じて, 以下の2種類の協調プロトコルで定義されるインタラクションが実行される。

4.5.1 Agency Unification プロトコル

エージェンシ間の対象同定が成功した場合に実行され, 同一対象を追跡していると判断されたエージェンシ同士の統合を定義するプロトコル。

本システムでは, 常に追跡対象とエージェンシが一对一に対応付けられていなければならない。しかし, この対応関係は対象検出・同定の失敗によって崩される可能性がある。こうして生じた対応関係のミスマッチに対処するプロトコルとして, Agency Unification が必要である。Agency Unification が発生するケースとして, 以下の例が挙げられる。

- AVA による対象検出の失敗により単一対象を複数対象として検出してしまった場合, その対象に対して複数のエージェンシが生成されてしまう。これらのエージェンシは, 後の観測とエージェンシ間の対象同定の結果, Agency Unification によって統合される。
- エージェンシ間の対象識別が失敗した結果, 複数

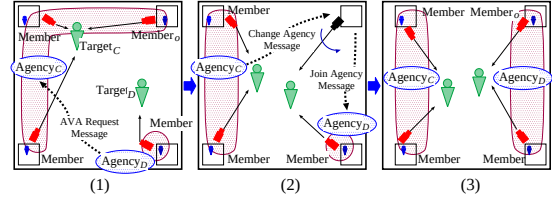


図 12 Agency Restructuring
Fig. 12 Agency restructuring.

対象を単一対象と見なししてしまった場合(追跡対象同士が近接したせいで, 全 AVA が単一対象領域しか検出できなかった場合など), エージェンシと検出対象間の一对一関係を守るため, Agency Unification によって一時的にエージェンシが統合される。後の観測により各対象が識別されると, Agency Formation, または Agency Spawning により, エージェンシが再度生成される。

図 11 の例では, エージェンシ B の追跡対象との間の対象同定に成功したエージェンシマネージャ A が, 統合要求メッセージをエージェンシマネージャ B に送信している(図 11, 1.)。このメッセージを受信したエージェンシマネージャ B は, 自身の全メンバに対してエージェンシ A への移籍を命令し(図 11, 2.), 自身は消滅する。この結果, 二つのエージェンシが統合する(図 11, 3.)。

4.5.2 Agency Restructuring プロトコル

各エージェンシの追跡対象情報とメンバ情報を考慮にいれ, 各対象の注視に適したメンバの再編成を定義するプロトコル。エージェンシ間の対象同定が失敗した場合に実行される。

Agency Restructuring は以下の条件に基づいて実行される。

- 各エージェンシのメンバ数は, その追跡対象の対象重要度によって決定される。
- 対象重要度によって決定される数の制約の下で, 各エージェンシには追跡対象の注視に適した AVA がメンバとして振り分けられる。この振り分けは, 効用関数のトラッキング適性を参照して実行される。すなわち, メンバ $1, \dots, M$ はトラッキング適性 $V_{T_1}^{n_1}, \dots, V_{T_M}^{n_M}$ の総和が最大となるように, それぞれの追跡対象 n_m を割り当てられる。但し, M はシステム内のメンバの総数, $n_m \in \{1, \dots, T\}$, T は追跡対象の総数。

実行例を図 12 に示す。エージェンシマネージャ D が, エージェンシ C の追跡対象との間の対象同定に失敗すると, 上記の条件に基づいてエージェンシ C からメンバを要求するか否かを決定する。要求を行う場

合、メンバ要求メッセージをエージェンシマネージャ C に送信する (図 11, 1.)。この要求を受けたエージェンシマネージャ C は、エージェンシ D の追跡対象の注視に適したメンバを選択し、そのメンバに対してエージェンシ D への移籍する命令するメッセージを送信する。このメッセージを受信したメンバ (図中の Membero) は、エージェンシ D へ移籍する (図 11, 2.)。その結果、二つのエージェンシが再編成される (図 11, 3.)。

4.5.3 フリーランサとの通信

エージェンシマネージャは以下の機能を実現するために、マネージャ間だけでなくフリーランサとの間でも情報交換を行う必要がある。

エージェンシと追跡対象間の一対一対応関係の維持

フリーランサが新たに対象を検出した際、その対象を追跡しているエージェンシの存在を確認するために、検出対象情報を全てのエージェンシマネージャに対して送信する必要がある。

システムの現在状態の確認 フリーランサ、またはエージェンシマネージャが、新たにフリーランサへと役割変更した AVA からのメッセージを受信すると、システム現在状態のサーチレベルを上昇させる。同様に、エージェンシマネージャからのメッセージを受信すると、そのエージェンシのメンバの数に応じて、システム現在状態のトラッキングレベルを上昇させる。

本システムでは、AVA の動的な役割変更に対応して上記のメッセージ交換を保証するために、フリーランサとエージェンシマネージャ間の情報交換はブロードキャストにより実装した。

5. システムの性能と動作の健全性

5.1 節において、理論的な追跡可能対象数の上限について述べ、5.2, 5.3 節において、システムが観測 (対象同定) 結果に基づいて内部状態を更新し続けることにより、適切な協調注視状態に回復不可能な状態遷移を行わない、すなわちシステム設計が健全であることを示す。

5.1 追跡可能な対象数と追跡安定性

システムの対象追跡に関する能力は、様々な要素に依存する (カメラや追跡対象の数、カメラの性能、カメラ位置とシーン構造の関係)。これらの要素は「システム設計に依存する理論的なシステムの性能限界」と「画像処理の失敗や対象の遮蔽などに起因される予測不能な問題」に大別される。実世界システムにおいて、後者の問題は不可避なものであるため、実世界の

あらゆる状況において、システムが全ての検出対象を継続的に追跡可能であることを保証するのは困難である。そこで本節では、理論的なシステムの性能限界に関して既存の他手法と比較するために、提案システムにおける対象とカメラの数の間の関係について述べ、同時に追跡可能な対象数の上限について論じる。

本システムでは、エージェンシと追跡対象は一対一に対応付けられる。また、各エージェンシには最低でも一つのメンバが所属していなければならない。更に、追跡対象の 3 次元情報を復元するためには、最低でも二つ以上のメンバからの検出対象情報が必要となる。追跡対象の 3 次元情報の利用は、エージェンシによる協調追跡の安定性・有効性を大きく向上させる。

- 3 次元方位線の比較に比べて、3 次元位置の比較による対象同定のほうが安定な結果が得られる。
- 障害物による遮蔽によってメンバが追跡対象を見失った場合でも、追跡対象の 3 次元位置を知らせることにより、対象方向の注視を可能とさせる。
- 追跡対象と AVA の備えるカメラの 3 次元位置を比較することにより、対象の注視に適した AVA を選択することが可能となる。

上の議論に基づいて、本システムの対象追跡能力を対象数 (n_t) と AVA 数 (n_a) の関係により定義する。

Case 1 $n_t \leq \lceil \frac{n_a}{2} \rceil$: 復元された 3 次元情報に基づいて、全ての対象を継続的に注視可能である。

Case 2 $\lceil \frac{n_a}{2} \rceil < n_t \leq n_a$: 全ての対象を追跡可能であるが、最低でも $(n_t - \lceil \frac{n_a}{2} \rceil)$ の対象は単一 AVA によって注視される。

Case 3 $n_a < n_t$: $(n_t - n_a)$ 以上の対象が追跡不可能である。

一方文献 7) では、システム全体で共有されるソフトウェアエージェントが各対象と対応付けられるため、理論的な追跡対象数の上限は存在しない。そこで、本システムにおいても追跡可能な対象数の制限を緩和するためには、以下に示すシステム設計が考えられる。

3 次元情報復元のためのブロードキャスト メンバが検出対象情報をブロードキャストし、各エージェンシがその情報に基づいて対象同定を行うことにより、Case 2 における追跡可能対象数の制限が緩和される可能性がある。しかし、メンバによる検出対象情報の送信は、全ての観測時刻において発生するため、ブロードキャストによる情報交換はネットワークの負荷を大きく上昇させてしまう。

空のエージェンシのためのアクティブカメラ制御 メンバによる検出対象情報のブロードキャストは、メンバを持たないエージェンシによる対象同定を

表 1 対象同定の結果実行される協調追跡プロトコル

Table 1 Cooperative-tracking protocols activated depending on the result of object identification.

受信対象情報	対象同定の結果	
	成功	失敗
フリーランサが送信した検出対象の方位線	Agency Formation	Agency Formation
メンバが送信した追跡対象の方位線	Agency Maintenance	Agency Maintenance, Agency Spawning
メンバが送信した非追跡対象の方位線	Agency Maintenance	Agency Spawning
エージェンシが送信した追跡対象の 3 次元位置	Agency Unification	Agency Restructuring

可能とする。この結果、理論的には Case 3 における追跡可能対象数の上限がなくなる。しかし、メンバを持たないエージェンシは、アクティブカメラの制御ができないため、視線方向制御による継続的な対象注視が保証されず、またズーム制御による高解像度な対象画像の獲得が不可能である。本論文では、2.1 節で述べたカメラとエージェント間の一対一関係の利点を備えたまま上記の問題を避けるために、(1) エージェンシは、自身のメンバが送信する検出対象情報のみから追跡対象の 3 次元情報を復元する、(2) 各エージェンシには最低でも一つのメンバが所属する、という条件の下で協調追跡を実現した。

5.2 協調プロトコルの必要十分性

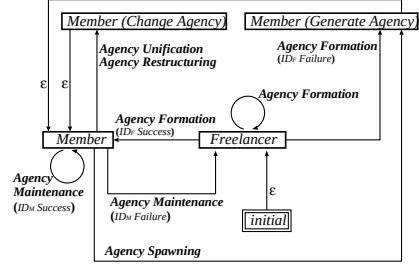
提案システムでは、全ての実世界の動的状況は、対象同定の結果によって表され、全ての対象同定は、他者から対象情報を受信したエージェンシマネージャによって実行される。したがって、受信対象情報の種類と実行された対象同定の結果の関係を確認することにより、協調プロトコルの種類の必要十分性を示すことができる。本システムでは、表 1 が示すように、全ての対象同定の結果に応じて過不足無く適切な協調プロトコルが設計されている。

5.3 状態遷移の健全性

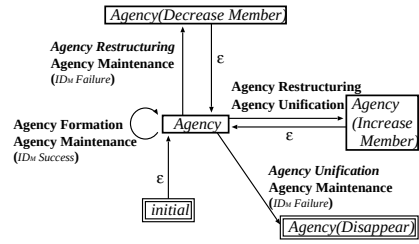
Intra-AVA 層では、視覚・行動・通信モジュールが相互に情報を交換しながら、一つの AVA として機能している。モジュール間の情報交換はダイナミックメモリを介して行われるため、各モジュールは情報を交換する相手の状態に依存してデッドロックを起こす事なく、自身のタスクを継続できることが保証されている。

Intra-agency 層では、エージェンシマネージャが、同じエージェンシに所属するメンバから検出対象情報を集めて対象同定を繰り返すことにより、協調追跡が行われている。この層における情報交換も、エージェンシマネージャの持つダイナミックメモリを介して行われるため、各メンバは独自のダイナミクスにしたがって継続的に機能することができる。図 13 (a) に AVA の状態遷移図を示す。

Inter-agency 層では、エージェンシ間の対象同定が



(a) AVA の状態遷移



(b) エージェンシの状態遷移

図 13 AVA とエージェンシの状態遷移モデル。ボールド体、斜体で表記されたプロトコルは、それぞれ自身の実行した対象同定、他エージェンシから送信された対象同定結果に応じて実行されるプロトコルを示す。また、 ϵ は自動的に実行される状態遷移を表す。ID_F、ID_M は、それぞれエージェンシがフリーランサ、メンバとの間で実行する対象同定を表す。

Fig. 13 State transition networks of the AVA and agency.

行われ、その結果に応じて協調プロトコルで定められたメッセージが交換される。本システムでは、異なるエージェンシ間で矛盾したプロトコルが同時に実行されるのを防ぐために、(1) 各エージェンシは、同時に一つのエージェンシとしかプロトコル実行のためのインタラクションを行わない、(2) メッセージ遅延、動的なエージェンシの生成・消滅などの予測不能な事象に対処するために、エージェンシ間のメッセージ待機にはタイムアウトが設定されている。図 13 (b) にエージェンシの状態遷移図を示す。

6. 実験

提案システムによる協調追跡の有効性を確認するための実験を行った。実験では、10 台の AVA を利用した。各 AVA は、アクティブカメラ (SONY EVI-

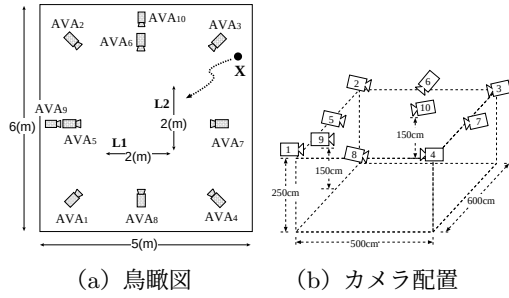


図 14 実験環境
Fig. 14 Experimental environment.

G20) を備えネットワーク結合された PC (PentiumIII 600MHz×2) により構成されている。視覚・行動・通信モジュール群とダイナミックメモリは、スレッドにより実装した。以上のリソースの下、各 AVA は平均 11[枚/秒] で画像撮影と対象検出を繰り返すことができた。また、全ての PC の内部時計は Network Time Protocol¹⁰⁾ によって同期が取られている。

図 14 は、実験環境におけるカメラ配置を示している。全てのカメラの外部パラメータ (3 次元位置と視線方向) は予め較正済である。

6.1 効用度関数の設定

実験では、効用度関数を以下のように設定した。

サーチ適性 フリーランサ f が観測可能なシーン中の領域面積を W_f とすると、フリーランサ f のサーチ適性 V_{S_f} は $V_{S_f} = \alpha_S \times W_f$ で与えられる。但し、 α_S は V_{S_f} が下記のトラッキング適性の値とバランスがとれるように設定された定数である。

トラッキング適性 メンバ m のカメラと追跡対象 n までの距離を D_m^n 、カメラ視野の中心線から対象 n を表す方位線までの角度を A_m^n とすると、追跡対象 n の注視に対するメンバ m のトラッキング適性 $V_{T_m}^n$ は $V_{T_m}^n = \frac{1}{D_m^n} \times \frac{1}{A_m^n}$ で与えられる。

6.2 仮想同期による対象同定の性能評価

対象同定を行う際に仮想同期を行うシステムと行わないシステムを利用した実験を行った。本実験では、非同期観測という条件だけでなく、ネットワークの負荷が大きい際の AVA 間の対象同定にも仮想同期が有効であることを評価するために、実験中にシステムの動作とは関係ないブロードキャスト通信を送り続けた。

システムは、二台の移動ロボットを追跡した。二台のロボットは、それぞれ図 14 (a) 中の L1, L2 上を 50[cm/秒] の速度で往復し続けた。

図 15 (a) は、ブロードキャスト通信のメッセージサイズの変化に応じたネットワークの負荷状況を表している。図 15 (b) は、復元されたロボットの 3 次元位

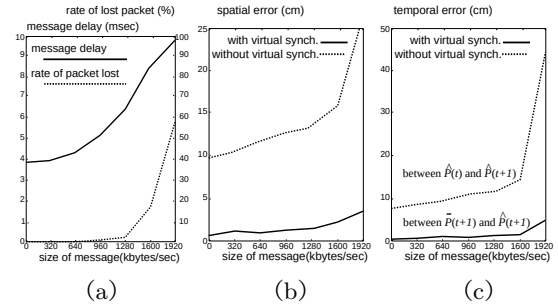


図 15 仮想同期の性能評価. (a) 通信遅延時間 (実線) と通信落ちの比率 (破線). (b) 空間的対象同定の誤差. (c) 時間的対象同定の誤差
Fig. 15 Performance evaluation of the virtual synchronization.

置と各メンバが検出した方位線との間の平均距離、即ち空間的対象同定の誤差を示している。図 15 (c) は、追跡対象の同定のために行われる時刻 t と $t+1$ において復元されたロボットの 3 次元位置間の平均距離、即ち時間的対象同定の誤差を示している。

実験結果から、特にネットワーク負荷が高い状況下において、仮想同期が有効であることが確認できた。

6.3 協調プロトコルの動作確認

協調プロトコルの有効性を確認するための実験を行った。この実験では、システムは二人の対象を追跡した。まず対象 1 が図 14 (a) 中の X で示される地点から観測シーン中に入り、次に対象 2 が同様に室内へと入った。その後、両対象は自由にシーン中を移動した。また、下記のパラメータをタスクとして与えた。

タスク制約条件 サーチレベルは 0.1, トラッキングレベルは 0.9。

対象重要度 全対象の対象重要度は 1.0。

図 16 上部の画像群は、各 AVA によって観測された画像系列の一部である。各画像のサイズは 320×240[pixel] である。図中の同じ行の画像群は、同一 AVA によって観測された画像であり、同じ列の画像群は、ほぼ同じ時刻において観測されている。また、黒線と白線で囲まれた領域は、それぞれ対象 1, 対象 2 の検出結果を示している。

図 16 下部の画像群は、同じ列の観測画像が撮影された時刻におけるエージェンシ構成を示している。図中の白円はフリーランサを、黒と灰色の円はそれぞれエージェンシ 1 とエージェンシ 2 に所属するメンバを示している。また、黒と灰色の四角が、それぞれ対象 1, 対象 2 を示している。

システムは、以下のように動作した。

a: 初期状態において、各 AVA は独立に対象探索を

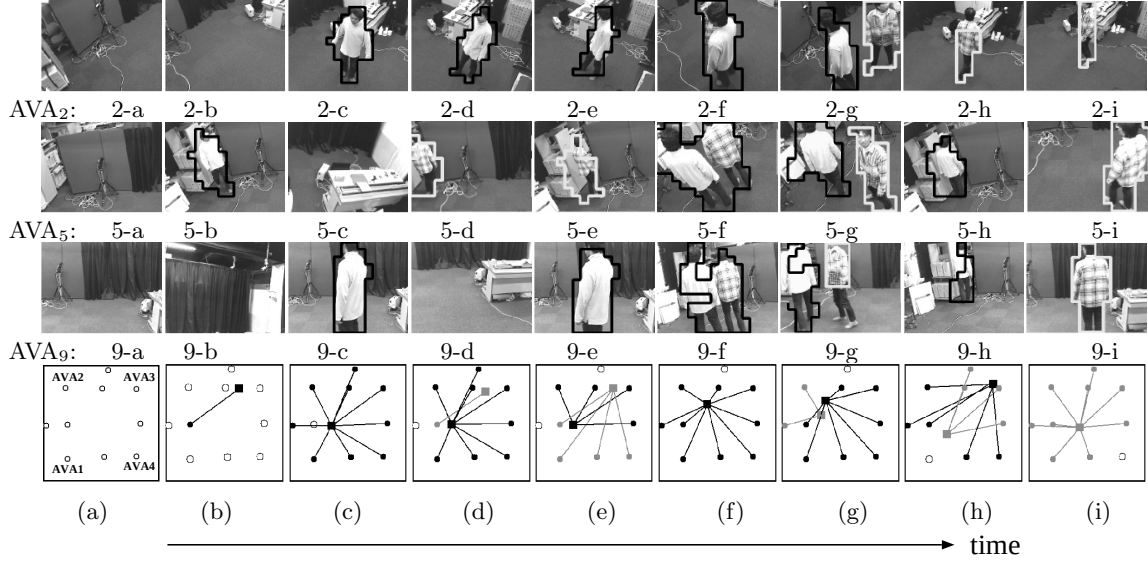


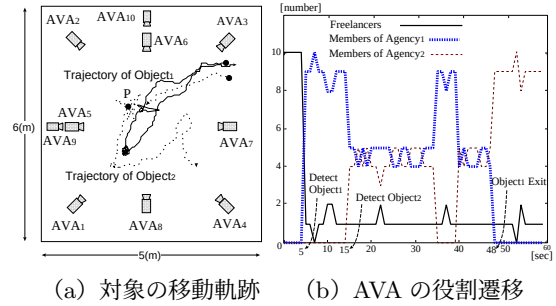
図 16 上部：観測画像系列。下部：エージェンシ構成と各 AVA の役割。

Fig. 16 Upper: Partial image sequences. Lower: The role of each AVA and the agency organization.

- 行った。
- b: AVA₅ が対象 1 を検出し (図 16, 5-b), エージェンシ 1 が生成された。
 - c: この瞬間にサーチを割当てられている AVA₅ を除いた全 AVA は, 対象 1 を追跡するエージェンシ 1 のメンバとして動作している。
 - d: AVA₅ が対象 2 を検出し (図 16, 5-d), エージェンシ 2 が生成された。
 - e: Agency Restructuring によってエージェンシ 1, 2 の間でメンバが交換された。
 - f: 対象 1, 2 が近接し, 両対象を画像上で識別可能な AVA がなくなったため, Agency Unification により一時的にエージェンシが統合された。
 - g: 対象間の距離が開くと, Agency Spawning によりエージェンシ 2 が再生成された。
 - h: 対象 1 が観測シーンから出ていった。
 - i: エージェンシ 1 の消滅後, AVA₄ を除いた全ての AVA がエージェンシ 2 のメンバとなり協調追跡を継続した。

図 17 (a) は, 復元された追跡対象の移動軌跡を示している。また, 図 17 (b) に示すグラフは, フリーランサ, 及び各エージェンシに所属するメンバ数の遷移を表している。グラフ中の凹凸は, AVA が対象の移動に応じて役割を変更している瞬間を示している。

以上の結果から, AVA やエージェンシ間の動的インタラクションによって, システム全体として複数対



(a) 対象の移動軌跡 (b) AVA の役割遷移

図 17 実験結果。

Fig. 17 Experimental results.

象を継続的に注視できていることが確認できた。

7. ま と め

本論文では, 分散アクティブカメラ群を利用した複数対象の実時間協調追跡システムを提案した。提案システムは, 以下の特徴を持つ。

- 複数の自律的な処理主体が動的にインタラクションを行うことにより, システム全体が一つの追跡システムとして振る舞う。
- 協調追跡を行うための動的インタラクションの種類によって, システムを 3 つの階層に分ける。

Intra-AVA 視覚, 行動, 通信モジュールがインタラクションを行い, 一つの AVA として振る舞う。

Intra-agency 同一エージェンシに属する AVA

群が、特定の対象を追跡するために対象情報を交換する。

Inter-agency 対象の動きに合わせて適応的にエージェンシを再編成するために、エージェンシ群がエージェンシ情報を交換する。

- ダイナミックメモリにより、各層におけるインタラクションを同期に伴う待ち時間無しで実行できるため、高い実時間性・即応性を備える。

これらの特徴を利用して、実世界の複雑、かつ多様な状況に適応可能システムが構成できることを示した。

本論文では、実時間対象追跡を応用例として三層構成アーキテクチャを提案したが、この構成は自律エージェンシ群によって構成される様々な協調システムに対して適用可能である。

Intra-AVA 多様かつ複雑な振る舞いを即応的に実現するためには、知的エージェンシはタスク遂行のために必要な複数の機能モジュールによって構成され、このモジュール群の相互作用によって一つの処理主体としての行動を発現すべきである。

Intra-agency 同じ目的を持つエージェンシ群は、エージェンシを構成して密に情報交換を行いつつ、協調行動すべきである。また、同じエージェンシに所属するエージェンシ群が、一つの目的を遂行するために整合的に振る舞うには、エージェンシマネージャのような意思主体の存在が有効である。

Inter-agency システム全体として効率的にタスクを実行するためには、各エージェンシ（マネージャ）は必要に応じて交渉を行い、その結果に基づいて自身の振る舞い・役割を適応的に決定しなければならない。

また、各自律主体間の協調行動・役割分担を決定つける協調プロトコル（Agency Formation, Maintenance, Spawning, Unification, Restructuring）は、追跡以外のタスクにおいても必要となる。しかし、交換する情報の種類やエージェンシを構成する基準、すなわち各協調プロトコルが実行される基準（例えば、あるシーン状況において、各エージェンシがどのエージェンシに所属するべきかの判定基準）はシステムのタスクに応じて設計する必要がある。

今後は、コンピュータビジョンの分野における様々な知見も考慮にいれ、以下の見地からシステムの機能を拡張していく予定である。

追跡可能な対象数： システム全体でより多くの対象を追跡するためには、5.1 節で述べたように (1) メンバによる検出対象情報のマルチキャスト、(2) メンバを持たない空のエージェンシによる対象注視、を実

現すれば良い。具体的には、特定のエージェンシにメンバとして所属しながら、必要に応じて他のエージェンシへも対象情報を送信する AVA の役割（サポータと呼ぶ）の追加によって実現可能であると考ええる。

最適カメラ配置： タスクに応じた最適なカメラ配置に関する研究¹¹⁾ が行われている。本システムにおいても、シーン中の障害物情報などを自動的に獲得し¹²⁾、その結果に基づいてカメラ配置を変更することにより、シーンの構造と対象の移動の両方を考慮にいたった効率的な協調追跡が実現可能となる。

視野に重なりを持たないカメラ配置： 広範囲における継続的な対象追跡のためには、視野に重なりを持たないカメラ間における対象同定が必要となる。この対象同定を実現するためには、対象の見えに基づいた同定¹³⁾ や、対象の移動経路の距離・経過時間に関する制約を利用する手法¹⁴⁾ などが利用可能である。

詳細画像の選択的詳細観測： タスクに応じて、必要となる詳細な対象画像の種類は異なる。例えば、対象の詳細な 3D モデルを獲得する場合¹⁵⁾、対象の未撮影部位の詳細画像を観測するカメラ制御が必要となる。また、個人認識を行うためには、顔を含んだ頭部を集中的に観測しなければならない⁸⁾。

信頼性の高い対象検出・同定法： 本論文では、各 AVA が観測画像中から対象を検出し、時系列的に同じ対象を対応付ける画像処理部分に関しては焦点を置かず、単純な背景差分による対象検出と検出座標間の近接性評価による対応付けのみが行われている。そこで、これらの処理を頑健に行うために文献 16) などの確率的な対象検出・追跡法を適用することが考えられる。

また、本システムでは、複数カメラの検出対象間や追跡対象の時系列的な対応付けに関しても、各検出結果ごとに対象位置の近接性のみを評価している。この手法では、狭い空間中に多数の対象が集まった場合、追跡対象数が 5.1 節で示した「システムが安定に追跡可能な対象数」以下であっても、対応付けの失敗が頻繁に起こりえる。こうした問題に対しても、多体軌跡推定の研究などで従来行われてきた data association¹⁷⁾ などの適用により、更に頑健な対象同定が実現できる。

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 96P0050 1）、科学研究費補助金特定領域研究（13224051）の補助を受けて行った。

参 考 文 献

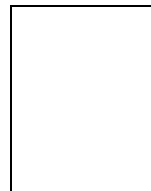
- 1) T. Matsuyama: Cooperative Distributed Vision - Dynamic Integration of Visual Perception, Action and Communication -, *Proc. of*

- Image Understanding Workshop*, pp.365–384 (1998).
- 2) 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).
 - 3) 日浦慎作, 村瀬健太郎, 松山隆司: ダイナミックメモリを用いた実時間対象追跡, 情報処理学会論文誌, Vol.41, No.11, pp.3082–3091 (2000).
 - 4) E. Borovikov and L. S. Davis: A Distributed System for Real-time Volume Reconstruction, *Proc. of IEEE Workshop on Computer Architecture for Machine Perception*, pp.183–189 (2000).
 - 5) N. Yoshida and T. Fuki: Target Tracking Using Tuple-Space-Based Mobile Agents, *Proc. of 19th IASTED ICAI*, pp.389–393 (2001).
 - 6) H. Ishiguro: Distributed Vision System: A Perceptual Information Infrastructure for Robot Navigation, in *Proc. of IJCAI-97*, Vol.1, pp.36–41 (1997).
 - 7) 中澤篤志, 加藤博一, 日浦慎作, 井口征士: 分散視覚エージェントを用いた複数人物追跡システム, 情報処理学会論文誌, Vol. 42, No. 11, pp.2699–2710 (2001).
 - 8) 加藤丈和, 向川康博, 尺長健: 安定な顔認識のための分散協調登録, 電子情報通信学会論文誌, Vol.J84-D-II, No.3, pp.500–508 (2001).
 - 9) G. P. Stein: Tracking from Multiple View Points: Self-calibration of Space and Time, *Proc. of Computer Vision and Pattern Recognition*, Vol. I, pp.521–527 (1999).
 - 10) D. L. Mills: Internet time synchronization: the network time protocol, *IEEE Transaction of Communications*, 39(10), pp.1482–1493 (1991).
 - 11) K. A. Tarabani, P. K. Allen, and R. Y. Tsai: A survey of sensor planning in computer vision, *IEEE Transaction on Robotics and Automation*, 11(1), pp.86–104 (1995).
 - 12) 浮田宗伯, 松山隆司: 移動対象の協調的追跡のための観測可能領域モデル生成・更新法, 情報処理学会論文誌, Vol. 42, No. 7, pp.1902–1913 (2001).
 - 13) I. Haritaoglu, D. Harwood and L.S. Davis: An Appearance-based Body Model for Multiple People Tracking, *Proc. of International Conference on Pattern Recognition 2000*, Vol. 4, pp.184–187 (2000).
 - 14) 和田俊和, 田村牧也, 松山隆司: 広域分散監視シ

- ステムにおける分散協調型対象同定法, 画像の認識・理解シンポジウム, Vol.1, pp.103–108 (1996).
- 15) ウ小軍, 和田俊和, 東海彰吾, 松山隆司: 平面間透視投影を用いた並列視体積交差法, 情報処理学会論文誌 CVIM, Vol. 42, No. SIG 6, pp.33–42 (2001).
 - 16) M. Isard and A. Blake: Condensation - conditional density propagation for visual tracking, *International Journal of Computer Vision*, 29(1), pp.5–28 (1998).
 - 17) I. J. Cox: A review of statistical data association techniques for motion correspondence, *International Journal of Computer Vision*, 10(1), pp.53–66 (1993).

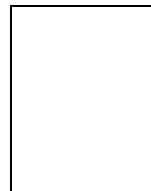
(平成 ? 年 ? 月 ? 日受付)

(平成 ? 年 ? 月 ? 日採録)



浮田 宗伯 (正会員)

1996 年岡山大学工学部卒業。2001 年京都大学大学院博士後期課程修了。同年奈良先端科学技術大学院大学情報科学研究科助手。2002 年科学技術振興事業団さきがけ研究 21 (「情報基盤と利用環境」領域) 研究員兼任。コンピュータビジョン, 分散協調処理に関する研究に従事。1999 年電子情報通信学会論文賞。



松山 隆司 (正会員)

1976 年京大大学院修士課程修了。京大助手, 東北大学助教授, 岡山大教授を経て 1995 年より京大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。2002 年学術情報メディアセンター長, 京都大学評議員。工博。画像理解, 人工知能, 分散協調視覚の研究に従事。最近は「人間と共生する情報システム」の実現に興味を持っている。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, 情報処理学会前理事。