

ダイナミックメモリを用いた実時間対象追跡

富山 忠文 松山 隆司

京都大学大学院情報学研究科 知能情報学専攻

本稿では、独立、並行に動作する対象検出(知覚モジュール)とカメラ制御(行動モジュール)とを統合した実時間対象追跡について述べる。とくに、対象情報、カメラ情報の管理とこれらの情報を利用した予測に基づく処理とに焦点を当てる。ダイナミックメモリでこれらの情報をすべて管理し、モジュール間の非同期な情報交換を実現する。各モジュールは、時刻と情報をダイナミックメモリに要求するだけで、それぞれの処理に必要な情報を入手することができる。知覚モジュールでは、対象の運動情報の予測を利用し、安定に対象を検出する。行動モジュールでは、カメラの運動情報と対象の運動情報との予測を利用して、カメラの視線方向を制御し、対象を追跡する。これにより、安定な実時間対象追跡が実現される。

Real-Time Object Tracking Using the Dynamic Memory

T. Tomiyama T. Matsuyama

Department of Intelligence Science and Technology
Graduate School of Informatics, Kyoto University

Dynamic interaction between visual perception and camera action modules is an essence of the system for object detection and tracking with an active camera. To harmonize these modules without synchronization, we employ a dynamic memory architecture. Dynamic interaction between perception and action modules is realized by the dynamic memory. Our object tracking system using the dynamic memory is characterized by (1) the dynamic data management by the dynamic memory with the Kalman filter for error reduction and prediction, (2) image analysis of a focused local area predicted based on the object trajectory in the dynamic memory, and (3) the prediction based camera control to compensate delays in image processing and physical camera motion.

1 はじめに

ドラマ、映画、スポーツ中継、あるいは、監視システムなどの分野では、移動対象をカメラでとらえ続けるという映像撮影技術が必要である。近年、これを自動化し、その技術を利用して、自動監視システム、知的テレビスタジオ、遠隔講義システムなどを実現しようという研究が盛んに行われている。移動対象をカメラで自動的にとらえ続けるためには、計算機に接続されたアクティブカメラというシステム構成を採用し、撮影された画像から対象を検出し、カメラを制御してそれを追跡しながら撮影し続けなければならない。このように、実時間対象追跡を実現するためには、対象の検出とカメラの制御が必須であり、

しかも、これらを実時間で処理しなければならない。

シーンに侵入した対象を検出するには、対象の多様性を考えると、対象に関する事前知識を必要とせず、静止している対象でも安定して検出できる背景差分法が有効である。背景差分法を適用するためには背景画像が必要であるが、視点固定型カメラをもちいると、全視野の背景モデルを生成することができる[1]ので、カメラの視線方向がわかれば、背景モデルからその視線方向の背景画像を生成することができる。背景差分法によって安定に対象を検出するためには、誤検出をどれだけ抑えるかが問題となるが、そのためには、大きさを考慮した対象の表現法の導入が望ましい。また、対象の運動情報を利用して、対象の画像中での存在領域を予測し、その領域で検出をはか

るなどの工夫が必要となる。

一方、カメラ制御に関しては、対象を追跡するようにカメラを制御することはもちろんのこと、対象を含む映像をユーザに見やすいように提示するために、なめらかにカメラ運動を制御する必要がある。しかし、カメラの視線方向を物理的に動かすには時間を要するので、対象が現在存在する方向にカメラの視線方向を向けようとするだけでは、意図した見せ方で対象を画面中にとらえることができない。時間遅れなく対象を追跡するには、対象やカメラの運動に関して予測を行い、その予測に基づいてカメラを制御することが必要となる。

これまでみてきたように、対象を検出するにはカメラの視線情報が必要であり、対象を追跡するようにカメラを制御するには対象に関する情報が必要である。すなわち、対象検出とカメラ制御には互いに他方の情報が必要である。他方の処理結果を待つてその情報を入手しては、処理のタイミングでの必要な情報が得られず、しかも他方の処理速度に影響され、実時間処理が実現されないという問題が生じる。そこで、対象検出とカメラ制御はそれぞれ、知覚モジュール・行動モジュールとして独立・並行に動作し、互いに他の動作効率を損なわないようにする必要がある。その上で、モジュール間の非同期な情報交換を実現する枠組みの導入が必要となる。ダイナミックビジョン [2], [3] は、このようなシステムのモジュール同士の動的な統合を提供する枠組みであり、そこでは、ダイナミックメモリを介してモジュール間の非同期な情報交換が実現される。

本稿では、独立、並行に動作する対象検出 (知覚モジュール) とカメラ制御 (行動モジュール) とを統合した実時間対象追跡について述べる。とくに、対象情報、カメラ情報の管理とこれらの情報を利用した予測に基づく処理に焦点を当てる。ダイナミックメモリでこれらの情報をすべて管理し、モジュール間の非同期な情報交換を実現する。各モジュールは、時刻と情報をダイナミックメモリに要求するだけで、それぞれの処理に必要な情報を入手することができる。知覚モジュールでは、対象の運動情報に基づいて、画像中の対象の存在領域を予測し、対象を検出する。行動モジュールでは、カメラの視線方向の変化に要する動作時間とその間の対象の運動予測とを用いて、カメラの視線方向¹を制御し、対象を追跡する。これにより、各モジュールは他のモジュールの動作に影響されたり影響を及ぼしたりすること無く、独自のタイミングで処理を実行し、実時間対象追跡が実現される。

¹本稿では、パン方向とテルト方向のみを制御対象とする。

2 ダイナミックメモリを用いた知覚と行動の動的統合

知覚モジュール (対象検出) と行動モジュール (カメラ制御) はそれぞれの処理のために互いに他方の情報を必要とするので、実時間対象追跡を実現するには、各モジュール内の実時間処理と、それぞれの処理を行うために必要とする、実時間での情報交換が必要である。知覚モジュールと行動モジュールが互いに他に影響を及ぼさないよう独立・平行に動作していることを考えると、後者は、知覚モジュールと行動モジュールの非同期な情報交換となっている。このようなシステムに対する枠組みを提供するのがダイナミックビジョン [2], [3] である。ダイナミックビジョンとは、システムを構成する各モジュールが独立、並行に動作する状況下で、非同期な情報交換を実現しながらモジュールを動的に統合する枠組みであり、そこでは、ダイナミックメモリを介してモジュール間の情報交換が実現される (図 1)。

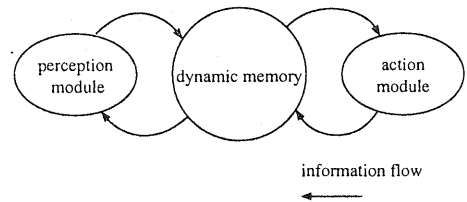


図 1: ダイナミックメモリを介した非同期な情報交換

ダイナミックビジョンの考え方に基づいてシステムを構成することで、独立・平行に動作する知覚モジュールと行動モジュールの非同期な情報交換が実現される。それによって、自分自身が得た情報や他方が得た情報を利用して、予測に基づく処理が可能となり、安定した実時間対象追跡が実現される。たとえば、現在の対象情報に基づいてカメラ制御を行うと追従の時間遅れを生じるが、予測に基づいたカメラ制御を行うことで、追従の時間を遅れをなくすることができる。また、対象の運動情報に関する予測を用いることで、追跡対象以外の検出を排除し、安定に追跡対象を検出することができる。

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との間の非同期な情報交換を司る。[4] では、知覚、行動の各モジュールは、現在の時刻における値をダイナミックメモリに問い合わせ、対象検出やカメラ制御を行っていた。すなわち、将来の値を予測して、対象検出やカメラ制御を行っていなかったため、安定な対象検出や時間遅れのない追従が実現されていなかった。そこで、本稿では、ダイナミックメモリに記述されている情報の予測値を利用し、安定な対象検出、時間遅れのないカメラの視線制御を実現す

る。また、対象検出の安定化をはかるため、大きさを考慮した対象の表現を新たに導入する。さらに、ダイナミックメモリにおいて、すべての情報を管理することにする。すなわち、各モジュールが必要とする情報が、それぞれが保持する情報であっても、ダイナミックメモリからその情報を得て処理を実行することにする。これにより、各モジュールは、予測に基づく処理のために必要な情報を得ることができる(図2)。

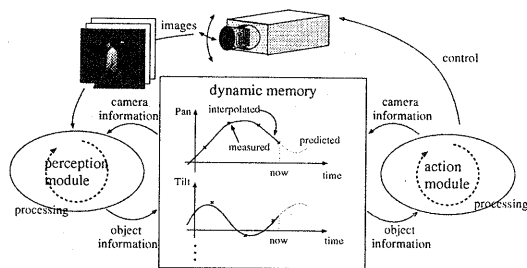


図2: ダイナミックメモリを用いた知覚と行動の動的統合

3 ダイナミックメモリ

知覚モジュールは対象情報を、行動モジュールはカメラ情報を、それぞれのタイミングでダイナミックメモリに書き込み、測定値が得られるたびにそれぞれの情報を随時更新する。ダイナミックメモリは、これらの情報をすべて管理し、各モジュールからの問い合わせにいつでも対応できるように、測定値をもとに補間・予測を行い、各情報を時間に関する連続関数として保持している。各モジュールは、時刻と情報を指定するだけで、ダイナミックメモリからそれぞれの処理に必要な情報を入手することができる。

ダイナミックメモリで表現したい情報を INFO とすると、INFO に関して、時刻 t_n に更新され次の更新時刻 t_{n+1} までの区間だけ有効な情報 $INFO_n$ を、その有効区間を付加して次のように定義する。

$$INFO_n : [t_n, t_{n+1}) \\ f_n(t)$$

ここで、 $f_n(t)$ は情報を表す関数で、 $[t_n, t_{n+1})$ でのみ有効である。 $n=0$ は、観測をはじめて最初に得られた情報に対応する。また、最新の情報については、有効区間を $[t_n, \infty)$ とした予測値とし、次の更新時まで有効であるとする。

3.1 対象の運動情報の記述

3.1.1 対象の表現法

本稿では、対象を矩形領域で表現する。具体的には、投影中心から矩形の対角線の交点(対象の重心)に向かうベクトルのパン・チルトの方位角とパン方向の方位角での矩形の大きさ、チルト方向の方位角での矩形の大きさによって対象を表現する(図3)。これにより、対象の大きさを考慮して画面中に対象を適切にとらえるカメラ制御が可能となる。また、矩形領域の一部に注目することで、対象の特定の部分に注目したカメラ制御も可能となる。その反面、矩形領域の切り出しに伴う誤差が対象の表現に影響を与えるという問題があり、誤差を伴う対象の表現の下で、信頼性の高い対象の表現をいかに補間、予測するかが大きな課題となる。

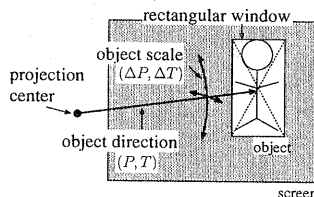


図3: 対象の表現法

3.1.2 対象の運動モデルとカルマンフィルタを用いた推定・予測

知覚モジュールでの処理時間は微少であること、視点固定型カメラでは、投影中心からの奥行き方向の情報に考慮する必要がない²ことを踏まえ、知覚モジュールの処理時間内では、対象の運動はカメラの投影中心を中心とする等角速度運動であると仮定する。

対象を検出して得られる測定値は、離散的な時間間隔でダイナミックメモリに書き込まれることに加え、得られる個々の測定値には誤差が含まれているのでこれを前提にし、いかに信頼性の高い予測値を求めるかが重要となる。本稿では、カルマンフィルタ [5], [6] を用いてこの予測を行うことにする。

[処理周期が一定の場合]

考えている有効区間における対象の重心のパン・チルトの方位角を $P_{obj_n}(t)$, $T_{obj_n}(t)$, 対象の大きさのパン・チルトの方位角を $\Delta P_{obj_n}(t)$, $\Delta T_{obj_n}(t)$ とし、それぞれの時間変化率を $\dot{P}_{obj_n}(t)$, $\dot{T}_{obj_n}(t)$,

²本稿では、対象を矩形領域として表現しているため、厳密には、奥行き方向の情報を考慮しなければならない。しかし、ここではズーム制御を扱っていないため、奥行き方向の情報を考慮しなくても差し支えない。

$\Delta \dot{P}_{obj_n}(t), \Delta \dot{T}_{obj_n}(t)$ とする. このとき, $x_n = (P_{obj_n}(t), T_{obj_n}(t), \Delta P_{obj_n}(t), \Delta T_{obj_n}(t), \dot{P}_{obj_n}(t), \dot{T}_{obj_n}(t), \Delta \dot{P}_{obj_n}(t), \Delta \dot{T}_{obj_n}(t))^T$ とおくと, 対象運動は, 次のように表される.

$$x_{n+1} = Fx_n + G\omega \quad (1)$$

ただし, $F = \begin{pmatrix} I_4 & \Delta t \cdot I_4 \\ 0_4 & I_4 \end{pmatrix}$, $G = \begin{pmatrix} 0_4 \\ I_4 \end{pmatrix}$, $\omega = (\omega_1, \omega_2, \omega_3, \omega_4)^T$ (各成分は平均 0, 分散 σ_i^2 の白色雑音) であり, I_4 は 4×4 の単位行列, 0_4 は 4×4 の零行列を表す. また, Δt は知覚モジュールの処理周期 (前後フレームの時間差) を表す. 一方, 誤差を $v = (v_1, v_2, v_3, v_4)^T$ (各成分は平均 0, 分散 $\sigma_{v_i}^2$ の白色雑音) とすると, 測定値 $y_n = (P'_{obj_n}(t), T'_{obj_n}(t), \Delta P'_{obj_n}(t), \Delta T'_{obj_n}(t))^T$ は,

$$y_n = Hx_n + v \quad (2)$$

と表される. ここに, $H = \begin{pmatrix} I_4 & 0_4 \end{pmatrix}$ である.

対象の運動と測定値が式 (1), (2) のように表されるとき, カルマンフィルタによって推定値 \hat{x}_n , 予測値 \bar{x}_{n+1} は以下のように計算される.

$$\hat{x}_n = \bar{x}_n + K_n(y_n - H\bar{x}_n) \quad (3)$$

$$\bar{x}_{n+1} = F\hat{x}_n \quad (4)$$

ここで, K_n は, 測定値 y_n に対して \bar{x}_n の修正を行う利得行列である.

[処理周期が一定でない場合]

上の議論では, 知覚モジュールの処理周期 (Δt) は一定であるとしている. しかし, 処理周期は, 処理内容やシーンの状況に応じて変化するので, Δt はフレームごとに変わると考える方が現実に即している.

カルマンフィルタを用いて, 処理周期が一定でない場合における情報の更新法を示す. ダイナミックメモリでは, 時刻 t_n に更新された対象の運動情報 OBJ_n は, 次のように表現される.

$$\begin{aligned} \text{OBJ}_n : [t_n, \infty) \\ (P_{obj_n}(t), T_{obj_n}(t), \Delta P_{obj_n}(t), \Delta T_{obj_n}(t), \\ \dot{P}_{obj_n}(t), \dot{T}_{obj_n}(t), \Delta \dot{P}_{obj_n}(t), \Delta \dot{T}_{obj_n}(t)) \\ = \bar{x}_{n+1}(t) \end{aligned} \quad (5)$$

知覚モジュールによって, y_{n+1} が時刻 t_{n+1} で測定されたとする (図 4). このとき, $\Delta t_{n+1} = t_{n+1} - t_n$ とおくと, 式 (4), (5) から次式を得る³.

$$\bar{x}_{n+1}(t_{n+1}) = F(\Delta t_{n+1})\bar{x}_{n+1}(t_n)$$

³式 (5) において, $\bar{x}_{n+1}(t)$ は時刻 t_n 以降の予測を表しているが, $t = t_n$ の時はその時刻の推定値 \hat{x}_n に等しく, $\bar{x}_{n+1}(t_n) = \hat{x}_n$ となる.

推定値 \hat{x}_{n+1} は, 式 (3) に基づいて次のように求められる.

$$\hat{x}_{n+1} = \bar{x}_{n+1}(t_{n+1}) + K_{n+1}(y_{n+1} - H\bar{x}_{n+1}(t_{n+1}))$$

以上の計算から, 対象の運動情報 ($t_{n+1} \leq t$) は次のように更新される.

$$\begin{aligned} \text{OBJ}_{n+1} : [t_{n+1}, \infty) \\ \bar{x}_{n+2}(t) = F(\Delta t)\hat{x}_{n+1} \\ (\Delta t = t - t_{n+1}) \end{aligned}$$

また, 時刻 t ($t_n \leq t < t_{n+1}$) においては, $t = t_{n+1}$ での連続性を考慮して, 以下のように線形補間で修正する.

$$\begin{aligned} \text{OBJ}_n : [t_n, t_{n+1}) \\ \bar{x}_{n+1}(t) = \frac{\hat{x}_{n+1} - \hat{x}_n}{t_{n+1} - t_n}(t - t_n) + \hat{x}_n \end{aligned}$$

3.2 カメラの運動情報の記述

カメラ運動速度の変更と現在のカメラ方位の取得の二つの命令を交互に送信し, 視点固定型カメラを制御する. すなわち, ダイナミックメモリ内では, カメラの視線方向とその速度を用いてカメラの運動情報を表現する.

3.2.1 カメラの運動モデル

カメラを実際に制御するには一定の時間が必要である. 命令を送信してもそれを受理されるまでには一定の時間を要する. 加えて, カメラの視線方向を物理的に変更するにも一定の時間を要する. これらの時間は, カメラの運動特性によって定まるため, その特性をあらかじめ測定しておくことは可能である. [4] では, カメラのこれらの運動特性を無駄時間と一次遅れモデルによって近似し, その特性をうまく把握している. 本稿でも, このモデルに従ってカメラの運動特性を近似する.

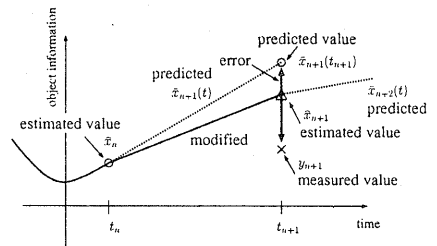


図 4: 対象情報の更新方法

無駄時間と一次遅れモデルによると、パン方向、チルト方向に一定速度 (V_{p0}, V_{t0}) で運動しているカメラに、時刻 $t = 0$ で速度指定コマンド (V_p, V_t) を与えたあとのカメラのパン方向の速度 $v_p(t)$ は、 τ を無駄時間、 T を時定数として、

$$v_p(t) = \begin{cases} V_{p0} & (0 \leq t < \tau) \\ (V_p - V_{p0}) \left(1 - e^{-\frac{t-\tau}{T}}\right) + V_{p0} & (\tau \leq t) \end{cases} \quad (6)$$

となる。チルト方向についても同様である。初期方位 (P_0, T_0) が与えられたときは、上式を積分すれば各時刻におけるパン・チルトの方位が得られる。

3.2.2 カメラの運動情報の表現

運動情報にはパン・チルトの方位とその角速度がある。カメラのパン、チルトの方位を P_{cam_n}, T_{cam_n} 、パン・チルト方向の角速度を v_{p_n}, v_{t_n} とすると、カメラの運動情報 CAM_n は次のように記述される。

$$CAM_n : [t_n, \infty) \\ P_{cam_n}(t), T_{cam_n}(t); v_{p_n}(t), v_{t_n}(t)$$

カメラに制御命令が送信されると、カメラの運動モデルに基づいた予測と、カメラ方位の実測値によるモデルの修正で CAM を更新する [4]。更新の詳細は [4] に譲るが、以下、その概要を簡単に説明する。

現在のカメラの運動情報 CAM_n に対して、時刻 t_{n+1} に速度指定命令 ($V_{p_{n+1}}, V_{t_{n+1}}$) をカメラに送信したとすると、 CAM_n の有効区間が $[t_n, t_{n+1})$ に変更され、無駄時間と一次遅れモデルを用いて $CAM_{n+1} : [t_n + 1, \infty)$ が定義される。 $P_{cam_{n+1}}(t)$ と $v_{p_{n+1}}(t)$ は、 $P_{cam_n}(t_{n+1})$ と $v_{p_n}(t_{n+1})$ 、および、式 (6) を用いて定義される。 $T_{cam_{n+1}}(t)$ と $v_{t_{n+1}}(t)$ も同様である。

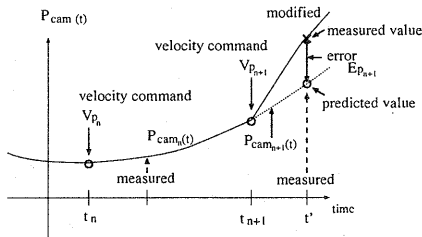


図 5: カメラ情報の修正更新法

次に、方位取得命令によって時刻 $t' (t_{n+1} \leq t')$ におけるカメラのパン、チルト方位の実測値 P', T' が得られたとする。このとき、パン方向のカメラ方位の運動モデルに基づく値の実測値からの誤差 $E_{p_{n+1}} = P_{cam_{n+1}}(t') - P'$ ($t_{n+1} \leq t'$) をもとめ、この値をも

とに、カメラ運動速度の連続性も考慮して、カメラの運動モデルを次のように修正する。

$$CAM_{n+1} : [t_{n+1}, \infty) \\ P_{cam_{n+1}}(t) = P_{cam_{n+1}}(t) + \\ E_{p_{n+1}} \left(1 - \exp\left(-\frac{t - t_{n+1}}{a(t' - t_{n+1})}\right)\right) \\ v_{p_{n+1}}(t) = v_{p_{n+1}}(t)$$

ただし、 $a(t' - t_{n+1})$ は時定数を表す。チルト方向についても同様である。

4 知覚モジュール

知覚モジュールは、カメラの入力画像から背景差分を用いることで対象を検出し、対象の方位、大きさをダイナミックメモリに書き込む。本節では、ダイナミックメモリにおいて予測される、カメラの運動情報、対象の運動情報を用いることで、画像中の対象の存在領域を予測し、安定に対象を検出する手法を述べる。

4.1 背景差分法

本稿では、知覚モジュールにおける画像処理を高速に行うために画像処理ボード RVS-10G を用いる。

まず、対象の存在しない時のシーンを任意の視線方向で撮影し、全視野の背景モデルを生成しておく [1]。各時刻の入力画像に対して、その画像が撮影された時の視線方向に全視野の背景モデルを再投影することで、その画像に対応する背景画像を生成する。この生成された背景画像と撮影された画像との差分をとることで、対象を検出することができる (図 6)。

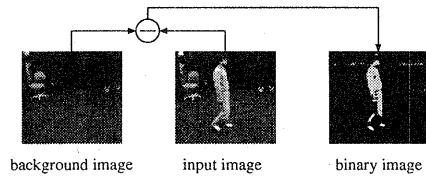


図 6: 背景差分の基本原則

しかし、実時間対象追跡では、カメラの視線方向を変化させながらシーンの撮影を行うため、撮影した時点の正確なカメラの視線方向が得られないという問題が生じる。この問題に対して、入力画像を平行移動させて背景差分を行うことで、入力画像の方位のずれを取り除いたり、入力画像の平行移動に伴う歪みを除去するために、平行移動させた入力画像と背

景が画像の8近傍との差分をとるといった方法が提案されている[4].

4.2 運動予測に基づく対象検出

従来の対象検出[4]では、入力画像全体に対して一様に背景差分を行っていた。そして、得られる変動領域の重心⁴を対象として検出していた。しかし、画像処理によって得たい情報は、入力画像中の対象に関する部分領域の情報である。そこで、対象の運動予測に基づいて、入力画像中の特定の領域に対して画像処理を行うことによって、対象検出の安定化をはかる。運動予測に基づく対象検出とは、入力画像中の対象の存在領域を予測して、その領域内で動的に背景差分の2値化の閾値を決定する手法である。動的に閾値を決定することで、対象と背景の明度差に適した2値化が可能になる。

知覚モジュールは、ダイナミックメモリから入力画像を撮影した時刻 $t = t_c$ のカメラの視線方向のパン・チルトの値 $(P_{cam}(t_c), T_{cam}(t_c))$ とその時の対象の方位 $(P_{obj}(t_c), T_{obj}(t_c))$ 、大きさ $(\Delta P_{obj}(t_c), \Delta T_{obj}(t_c))$ を得る。「ある投影中心に対して、3次元空間中の点は、法線ベクトルの方向と大きさが既知である平面に一意に投影される」という性質を利用すると、これらの量から対象が存在すると予測される領域を入力画像中の矩形領域として表すことができる(図7)。

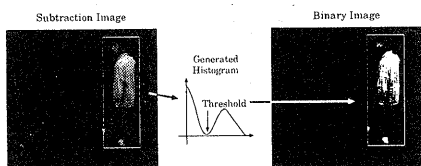


図7: 対象領域の予測とヒストグラムによる2値化

対象が存在すると予測される矩形領域に対して背景差分を行い、そこで得られた差分画像のヒストグラムを作成する。そのヒストグラムに対して判別分析を適用することで最適な閾値を求める。判別分析とは、クラス間の分離度を最大にするという基準の下で、最適な閾値を求める手法である。ここでは、差分画像中の対象が含まれていると予測される領域において、対象領域と背景領域とを分離するための2値化を行う際の閾値を判別分析を用いて決定する。

5 行動モジュール

行動モジュールは、最初に検出した対象を画面の中央にとらえるように速度指定コマンドを用いて、カ

⁴これは必ずしも実際の追跡対象の重心とは一致しない。

メラ制御を行い、その値をダイナミックメモリに書き込む。また、カメラの方位角を測定してダイナミックメモリに書き込む。

[4]では、現在の対象方位とカメラの視線方向の差をなくすように速度を制御していた。この制御法で得られる速度は、対象方位とカメラの視線方向が大きくずれていると高速になり、カメラの視線方向が対象方位に近づくにつれて抑制される。しかし、カメラの視線方向を変化させるには一定の時間がかかるので、実際にカメラが指定速度になった時点では、対象方位とカメラの視線方向にはずれが生じる。したがって、対象が運動している場合は、カメラの視線方向が対象方位に対して時間遅れを生じる。このため、対象の運動に対して時間遅れを生じた追従となっていた。この問題を解決するためには、カメラの動作時間と、その間の対象の運動の2つを考慮にいたれたカメラ制御が必要である。ここでは、対象の運動予測とカメラの運動予測を用いた制御法を用いる。

時刻⁵ $t_b = t_n + \tau + \Delta t$ での対象方位のパン・チルトの値 $(P_{obj}(t_b), T_{obj}(t_b))$ と時刻 $t_a = t_n + \tau$ でのカメラの視線方向 $(P_{cam}(t_a), T_{cam}(t_a))$ は、ダイナミックメモリにおいて予測されている。したがって、カメラに送信する速度指定コマンドを (V_{p_n}, V_{t_n}) とすると、式(6)を積分して得られる各時刻における方位を参考にして、次式が得られる。

$$P_{cam}(t_b) = P_{cam}(t_a) + \Delta t V_{p_n} + T \left(\exp \left(-\frac{\Delta t}{T} \right) - 1 \right) (V_{p_n} - v_p(t_a))$$

いま、 $t = t_b$ において対象方位とカメラの視線方向が一致するようにカメラを制御することにする、

$$P_{obj}(t_b) = P_{cam}(t_b)$$

が成り立つ。この2式を、 V_{p_n} について解くと、

$$V_{p_n} = v_p(t_a) + \frac{P_{obj}(t_b) - P_{cam}(t_a) - v_p(t_a)\Delta t}{\Delta t + T \left(\exp \left(-\frac{\Delta t}{T} \right) - 1 \right)}$$

が得られる。チルト方向の速度 V_{t_n} も同様に求めることができる。カメラに対して、ここで得られた (V_{p_n}, V_{t_n}) を速度変更コマンドとして送信することでカメラを制御する。これにより、対象運動に遅れずにカメラの視線方向を制御することが実現される。

行動モジュールの周期 Δt は、あらかじめ定められた一定値ではなく、モジュール内での処理に依存するので、前もって測定しておくことはできない。そこで、行動モジュールの過去 m 回の平均周期を計算し、それを Δt の値として予測する。

$$\Delta t = \sum_{i=n-m}^n (t_i - t_{i-1})$$

⁵ τ は無駄時間、 Δt は行動モジュールの周期。

6 実験と考察

ここでは、試作システムを設計し、前章までに示した手法の有効性を検証する。試作システムにおいて用いた、ハードウェア構成は次の通りである。

カメラ SONY 製 EVI-G20

計算機 PC (CPU: Pentium II 266MHz, OS: FreeBSD)

画像処理ボード RVS-10G

6.1 対象運動の予測, 推定に関する実験

カメラから 2m から 3m 程度離れたところを歩く一人の人物を一台のカメラを用いて追跡する。カメラからの入力画像はサイズが縦横 240 × 256 画素の濃淡画像である。

本実験では、式 (1), (2) のカルマンフィルタを用いるときの測定値に含まれる誤差 v と、状態モデルの変化を表す ω の分散を、パン方向に対しては $(\sigma_i^2/\sigma_{v_i}^2) = 10^{-11}$ ($i = 1, 3$), チルト方向に対しては $(\sigma_i^2/\sigma_{v_i}^2) = 10^{-13}$ ($i = 2, 4$) とした。以上の設定において、実験した結果を図 9 と、図 10 に示す。

図 9 において、横軸は時間を表し、縦軸はカメラパラメータ⁶を表している。また、(a), (b) が対象方位、(c), (d) が対象の大きさを表している。各図において、実線は予測値、× が測定値を表している。図 9 (d) において、時間が 1 秒のところまで測定値が非常に小さい値になっている。これは、背景差分の結果、対象領域が分離されているために本来の対象の大きさが正しく測定できなかったためである。しかし、この場合でも、カルマンフィルタによって適切な予測が行われていることがわかる。図 9 から、対象の運動予測が正しく行えていることがわかる。

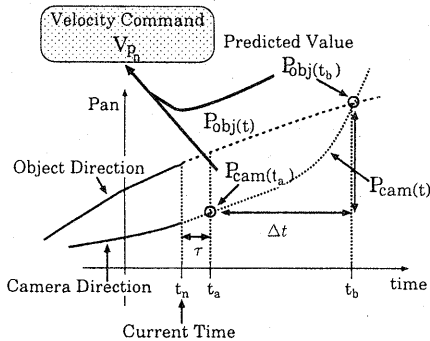


図 8: 予測に基づくカメラ制御

⁶カメラパラメータ y と角度 x (度) は, $y = 519.26667x$ を満たす。

図 10 は、対象追跡を行っている時の入力画像 (上段) と背景差分後の 2 値化画像 (下段) を約 1 秒間隔で連続的に並べたものである。また、入力画像中における矩形領域は、予測された対象の存在領域を示している。対象の運動を等角速度運動と仮定しているので、対象の動きが急激に変化するときは対象領域の予測の精度は悪くなるが、予測領域が対象から全くはずれるということはなく、正確に予測されていることがわかる。

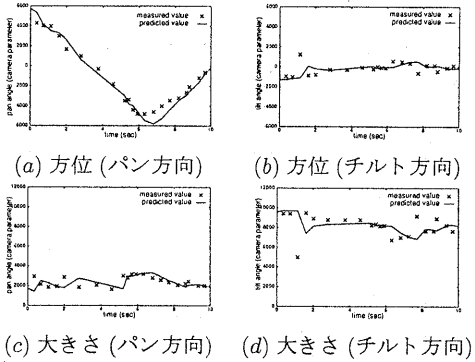


図 9: 対象方位, 大きさの測定値と予測値

6.2 予測に基づくカメラ制御に関する実験

対象の運動予測を用いないカメラ制御と、運動予測に基づくカメラ制御を 6.1 節と同様の条件において対象追跡の実験を行うことにより、カメラ制御の実時間性を考察した。ここでは、対象の運動予測を用いるカメラ制御として [4] で利用されている PID 制御を用いた。

図 11 に結果を示す。ここで、横軸は時間を表し、縦軸は対象方位とカメラの視線方向の差をカメラパラメータの値で表現している。また、点線は PID 制御に基づくカメラ制御を示し、実線は予測に基づくカメラ制御を示している。この図において、対象方位とカメラの視線方向の差が零であるということは、カメラの視線方向が対象方位を向いていることを意味しており、負の値は、対象の運動に対してカメラの視線方向が遅れていることを意味している。

図から明らかなように、PID 制御を用いたカメラ制御では、対象方位に対して時間遅れを生じている。また、予測に基づくカメラ制御では、3 秒付近で、大きく対象方位からずれている。これは、対象の運動予測に用いているカルマンフィルタが、最初の数回は測定値から、最適な推定値を出せないことによるものである。しかし、その後は、対象方位に時間遅れなくカメラ制御が行われている。今回の実験では、PID 制御においては、約 0.5 秒の時間遅れがあり、予測を用いたカメラ制御の方は、はじめのぶれを除くと 0.05

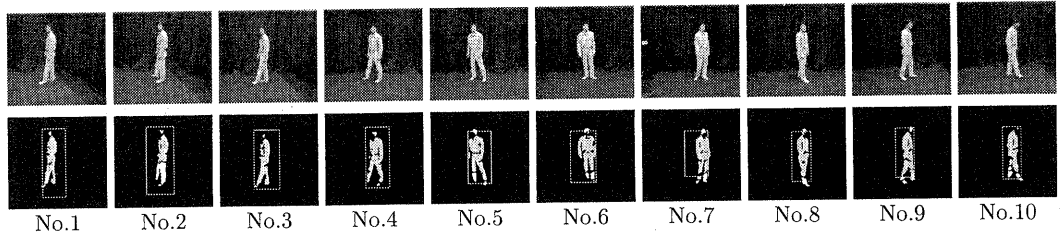


図 10: 運動予測に基づく対象検出 (上段から入力画像, 2 値化画像)

秒程度であった。これより、運動予測に基づくカメラ制御を用いることで、時間遅れが約 90% 改善されたことがわかる。

制御を用いて実現することが挙げられる。また、一般的にシーンには複数の対象が存在するので、対象の同定を実現することも必要である。

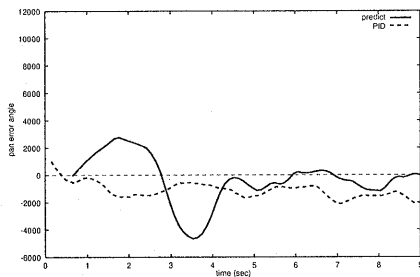


図 11: 予測に基づくカメラ制御の追従性

7 おわりに

本稿では、独立、並行に動作する知覚モジュール(対象検出)と行動モジュール(カメラ制御)による実時間対象追跡における情報管理という問題に焦点を当て、各モジュールがその情報をできるだけ利用し、予測に基づいた処理を行う実時間対象追跡について述べた。

ここでは、追跡対象の情報やカメラの運動情報はすべて、ダイナミックメモリによって管理される。ダイナミックメモリは、各モジュールからの問い合わせにいつでも対応できるように、各情報を時間に関する連続関数として保持している。各モジュールは、時刻と情報をダイナミックメモリに要求するだけで、それぞれの処理に必要な情報を入手することができる。

知覚モジュールでは、対象の運動情報に基づいて、画像中の対象の存在領域を予測し、対象を検出する。一方、行動モジュールでは、カメラの視線方向の変化に要する動作時間とその間の対象の運動予測とを用いて、カメラの視線方向を制御し、対象を追跡する。これにより、安定な実時間対象追跡が実現される。

今後の課題としては、画像中に占める対象の存在領域を一定に保つなど適切な対象の見せ方をズーム

謝辞

本稿を執筆するにあたり、京都大学情報学研究科杉本晃宏講師に適切な御助言を頂いたことに深く感謝する。本研究を行うにあたり、日本学術振興会未来開拓学術研究推進事業 (JSPS-RFTF 96P00501) の補助を受けた。

参考文献

- [1] 和田, 浮田, 松山: “Appearance Sphere —パン・チルト・ズームカメラのための背景モデル—”, *Proc. of MIRU'96*, Vol.2, pp.103-108, 1996.
- [2] T. Matsuyama: “Cooperative Distributed Vision — Dynamic Integration of Visual Perception, Action, and Communication —”, *Proc. of Image Understanding Workshop*, pp. 365-384, Monterey, CA, 1998.
- [3] T. Matsuyama: “Dynamic Memory: Architecture for Real Time Integration of Visual Perception, Camera Action, and Network Communication”, *Proc. of CVPR*, 2000, to appear.
- [4] 村瀬健太郎: “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.
- [5] 加藤寛一郎: “最適制御入門”, 東京大学出版会, 1987.
- [6] 中野, 西山: “パソコンで解くカルマンフィルタ”, 丸善株式会社, 1993.