

パーティクルフィルタとその実装法

加藤 丈和

和歌山大学
〒640-8510 和歌山市栄谷930
和歌山大学 システム工学部
E-mail : t.kato@ieee.org

あらまし

本稿では、非線形、非ガウス型の時系列フィルタリング法である、パーティクルフィルタについて、特に Isard らの Condensation 法に代表されるコンピュータビジョンにおける対象追跡への応用に焦点を当て、理論と実装法を概説する。時系列フィルタリングに関する基本的な考えから、カルマンフィルタなどの線形、ガウス型のフィルタリング手法、パーティクルフィルタによる非線形、非ガウス型への拡張について説明し、具体的な実装例を紹介する。

キーワード 時系列フィルタリング、パーティクルフィルタ、カルマンフィルタ、Condensation

Introduction of Particle Filter and Its Implementation

Takekazu Kato

Wakayama University

Department of Computer and Communication Science, Wakayama University
930 Sakaedani, Wakayama-shi, 640-8510, JAPAN E-mail : t.kato@ieee.org

Abstract This article introduces the theory and the implementation of Particle Filter that is one of non-linear and non-gaussian filter. This paper explains the basic idea of Filtering of time series, Kalman-Filter that is liner and gaussian filter and Particle Filter. We introduces the implementation of Particle filter.

Keywords Filtering Time Series, Particle Filter, Kalman Filter, Condensation

1 はじめに

時系列フィルタリングは、もともと制御理論の分野で研究されてきた理論で、過去に観測された信号系列から、未来の値を予測したり、ノイズを含む観測値から対象の状態を推定する問題などが取り扱われてきた。パーティクルフィルタは、時系列フィルタリングの一種であり、非線形、非ガウス型の状態空間モデルにたいして、効率よく状態を推定する手法である。近年コンピュータビジョンの分野でも、対象追跡や輪郭追跡などにおいて注目されており、パーティクルフィルタを使った追跡手法が数多く提案されている。

パーティクルフィルタ以前の時系列フィルタ（カルマンフィルタなど）では、観測モデル、システムモデルが線形、ガウス型の分布モデルなど、強い制

約があり、コンピュータビジョンの分野では一旦ほかの手法で検出、追跡した結果に対するノイズ除去や補完、スムージングなど、補助的な利用しかされてこなかったが、パーティクルフィルタでは、扱う状態空間や観測モデルなどにほとんど制約がないため、画像から直接追跡対象の状態を推定する方法として利用されている。

パーティクルフィルタは、1995年に北川らによってモンテカルロフィルタという名前で提案され、これと同時に Gordon らによってもブートストラップフィルタという名前で提案された。これらは独立に提案された手法であるが、ほぼ同一の手法であり、その後パーティクルフィルタという名前に統一された。一方、コンピュータビジョンの分野では、1998年にパーティクルフィルタと同一の枠組みに基づく輪郭の追跡手法として Condensation 法 [1]

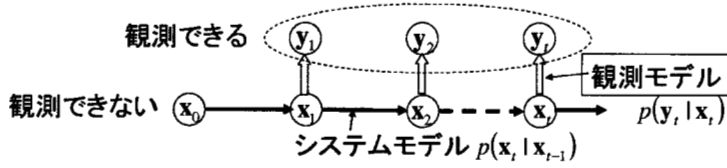


図 1: 状態遷移と観測ベクトル

が提案された。

このようは経緯から、様々な分野で様々な呼び方がされている本手法であるが、提案された経緯から考えると、時系列フィルタリングとして本手法の枠組み自体は、パーティクルフィルタと呼ぶべきであると考え、本稿でもパーティクルフィルタという呼び方を用いるが、モンテカルロフィルタ、ブートストラップフィルタ、Condensation など同一の手法であると考えてよい。

以下、まず 2 章で時系列フィルタを対象追跡に適用する場合について、問題の定式化を行い、3 章では線形、ガウス型分布に対するフィルタ手法であるカルマンフィルタについて簡単に説明する。4 章ではパーティクルフィルタとそれを追跡に応用する方法について説明する。また、5 章では具体的な実装方法について紹介する。

2 時系列フィルタの概要

時系列フィルタは、ノイズを含む観測値から対象の状態を推定する手法である。対象追跡に適用する場合、対象をパラメトリックな状態ベクトルを持つモデルで表現し、画像そのものや画像から得られる特徴量を観測ベクトルとして、対象の位置、姿勢、変形などの状態を推定する。

対象の状態がマルコフ性を満たす場合、時刻 $t, t+1$ の対象の状態ベクトル $\mathbf{x}_t, \mathbf{x}_{t+1}$ と、観測ベクトル \mathbf{y}_t の関係は次式のように定式化される。

$$\mathbf{x}_{t+1} = \mathbf{f}_t(\mathbf{x}_t) + \mathbf{G}_t \mathbf{w}_t \quad (1)$$

$$\mathbf{y}_t = \mathbf{h}_t(\mathbf{x}_t) + \mathbf{v} \quad (2)$$

ここで、 $\mathbf{f}_t(\mathbf{x}_t)$ は、時刻 t と時刻 $t+1$ との状態変化を表わす状態遷移関数、 $\mathbf{h}_t(\mathbf{x}_t)$ は、ある状態のときに得られる観測値の関係を表わす観測関数、 $\mathbf{w}_t, \mathbf{v}_t$ はそれぞれ、ある分布に従うシステムノイズ、観測ノイズである。

ここで、時刻 t までの観測値の集合を $\mathbf{Y}_t = \{\mathbf{y}_1, \dots, \mathbf{y}_t\}$ としたとき、追跡問題は \mathbf{x}_t の事後確率分布 $p(\mathbf{x}_t | \mathbf{Y}_t)$ や、 \mathbf{x}_t の最小分散推定量 $\hat{\mathbf{x}}_t$ を推定する問題として定式化できる。 $\hat{\mathbf{x}}_t$ は \mathbf{x}_t の条件付き期待値

$$\hat{\mathbf{x}}_t = \mathcal{E}(\mathbf{x}_t | \mathbf{Y}_t) = \int \mathbf{x}_t p(\mathbf{x}_t | \mathbf{Y}_t) d\mathbf{x}_t \quad (3)$$

として与えられる。

観測値 \mathbf{y}_t から状態 \mathbf{x}_t の事後確率 $p(\mathbf{x}_t | \mathbf{Y}_t)$ を直接推定することは困難であるため、時系列フィルタリングでは、ベイズの定理と \mathbf{x}_t のマルコフ性から逐次的な推定問題に置き換えて推定する(図 1)。事後確率 $p(\mathbf{x}_t | \mathbf{Y}_t)$ は、ベイズの定理により次式のように事前確率と尤度の積に置き換えられる。

$$\begin{aligned} p(\mathbf{x}_t | \mathbf{Y}_t) &= p(\mathbf{x}_t | \mathbf{y}_t, \mathbf{Y}_{t-1}) = \frac{p(\mathbf{x}_t, \mathbf{y}_t, \mathbf{Y}_{t-1})}{p(\mathbf{y}_t, \mathbf{Y}_{t-1})} \\ &= \frac{p(\mathbf{y}_t | \mathbf{x}_t, \mathbf{Y}_{t-1}) p(\mathbf{x}_t, \mathbf{Y}_{t-1})}{p(\mathbf{y}_t, \mathbf{Y}_{t-1})} \quad (4) \\ &= \frac{p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{Y}_{t-1})}{p(\mathbf{y}_t | \mathbf{Y}_{t-1})} \end{aligned}$$

ここで、 $p(\mathbf{y}_t | \mathbf{x}_t)$ は、ある状態 \mathbf{x}_t のときに、観測値 \mathbf{y}_t を得る確率(尤度)であり、式(2)の観測モデルによって与えられる。また、分母の $p(\mathbf{y}_t | \mathbf{Y}_{t-1})$ は、状態 \mathbf{x}_t とは無関係な項であり、実際の問題では $\int p(\mathbf{x}_t | \mathbf{Y}_t) d\mathbf{x}_t = 1$ となるような正規化定数 k_t と考えることができる。

$p(\mathbf{x}_t | \mathbf{Y}_{t-1})$ は、時刻 t における \mathbf{x}_t の事前確率であり、 \mathbf{x}_t のマルコフ性により、次式のように与えられる。

$$\begin{aligned} p(\mathbf{x}_t | \mathbf{Y}_{t-1}) &= \int p(\mathbf{x}_t, \mathbf{x}_{t-1} | \mathbf{Y}_{t-1}) d\mathbf{x}_{t-1} \\ &= \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{Y}_{t-1}) p(\mathbf{x}_{t-1} | \mathbf{Y}_{t-1}) d\mathbf{x}_{t-1} \quad (5) \\ &= \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} | \mathbf{Y}_{t-1}) d\mathbf{x}_{t-1} \end{aligned}$$

ここで、 $p(\mathbf{x}_{t-1} | \mathbf{Y}_{t-1})$ は時刻 $t-1$ における事後確率であり、 $p(\mathbf{x}_t | \mathbf{x}_{t-1})$ は時刻 $t-1$ と t の間の状態推移確率であり、式(1)によって与えられる。

時系列フィルタリングでは、時刻 $t-1$ までの観測から時刻 t の状態を推定する式(5)を prediction(予測)と呼び、予測と状態推移確率から事後確率を推定する式(4)を filtering(フィルタ)と呼ぶ。時系列フィルタリングとは、予測とフィルタにより各時刻の事後確率 $p(\mathbf{x}_t | \mathbf{Y}_t)$ を求める問題である。

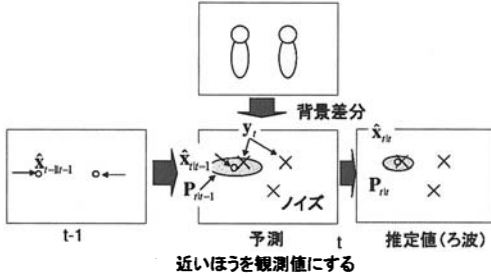


図 2: カルマンフィルタを用いた追跡

3 カルマンフィルタ

3.1 線形システムの推定

時系列フィルタリングにおいて、システムプロセス $f_t(\mathbf{x}_t)$ と観測プロセス $h_t(\mathbf{x}_t)$ が線形であり、かつ、システムノイズ \mathbf{w}_t 、観測ノイズ \mathbf{v}_t が白色のガウス分布である場合、1960年に Kalman [2] が提案したカルマンフィルタと呼ばれる手法で、線形に推定することができる。

3.2 カルマンフィルタのアルゴリズム

$f_t(\mathbf{x}_t)$, $h_t(\mathbf{x}_t)$ が線形な場合、式 (4), (5) は状態遷移行列 \mathbf{F}_t と観測行列 \mathbf{H}_t を用いて次式のように書ける。

$$\mathbf{x}_{t+1} = \mathbf{F}_t \mathbf{x}_t + \mathbf{G}_t \mathbf{w}_t \quad (6)$$

$$\mathbf{y}_t = \mathbf{H}_t \mathbf{x}_t + \mathbf{v}_t \quad (7)$$

このとき、事前分布 $p(\mathbf{x}_t | \mathbf{Y}_{t-1})$ 、及び事後分布 $p(\mathbf{x}_t | \mathbf{Y}_t)$ はガウス分布となり、時刻 t における推定値 $\hat{\mathbf{x}}_{t/t} = E\{\mathbf{x}_t | \mathbf{Y}_t\}$ は次のように線形に計算できる。

$$\hat{\mathbf{x}}_{t/t} = \hat{\mathbf{x}}_{t/t-1} + \mathbf{K}_t [\mathbf{y}_t - \mathbf{H}_t \hat{\mathbf{x}}_{t/t-1}] \quad (8)$$

$$\hat{\mathbf{x}}_{t/t-1} = \mathbf{F}_{t-1} \hat{\mathbf{x}}_{t-1/t-1} \quad (9)$$

ここで、 \mathbf{K}_t はカルマンゲインと呼ばれ、次式で与えられる。

$$\mathbf{K}_t = \mathbf{P}_{t/t-1} \mathbf{H}_t^T [\mathbf{H}_t \mathbf{P}_{t/t-1} \mathbf{H}_t^T + \mathbf{R}_t]^{-1} \quad (10)$$

$$\mathbf{P}_{t+1/t} = \mathbf{F}_t \mathbf{P}_{t/t} \mathbf{F}_t^T + \mathbf{G}_t \mathbf{Q}_t \mathbf{G}_t^T \quad (11)$$

$$\mathbf{P}_{t/t} = \mathbf{P}_{t/t-1} - \mathbf{K}_t \mathbf{H}_t \mathbf{P}_{t/t-1} \quad (12)$$

ただし、 $\mathbf{Q}_t, \mathbf{R}_t$ はそれぞれ、 $\mathbf{w}_t, \mathbf{v}_t$ の共分散行列である。

3.3 対象追跡への適用

カルマンフィルタを対象追跡に応用するには、観測モデルが線形であるため、画像そのものを観測値

とすることはできない。そこで、図2に示すように、背景差分などの手法によって検出した対象位置のフレーム間の対応付けやノイズ除去に利用する方法が提案されている。このとき、カルマンフィルタの予測によって、過去の追跡結果から次のフレームの追跡結果 $\hat{\mathbf{x}}_{t/t-1}$ を予測し、その近傍から対象の位置を探索する方法が多数提案されている [3, 4, 5, 6]。これらの手法では、状態ベクトルを対象の位置と速度 (と加速度) を用いて $\mathbf{x}_t = [x_t, y_t, \dot{x}_t, \dot{y}_t]^T$ と表現し、状態遷移行列 F を次式のように与えて対象の動きの予測を行っている。

$$F = \begin{bmatrix} 1 & 0 & \delta t & 0 \\ 0 & 1 & 0 & \delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

また、観測行列 H を次式のように与え、位置のみが観測されるとしている。

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (14)$$

状態遷移行列 F 、観測行列 H の与え方を工夫することで、アフィン変換によるオブジェクトの変形を含む追跡 [5, 7] や、3次元の位置姿勢の追跡 [8, 9] などに拡張する方法も提案されている。また、カルマンフィルタの $f_t(\mathbf{x}_t)$, $h_t(\mathbf{x}_t)$ が線形であるという制限を緩め、これらの関数が滑かであるという制約のもとで、推定値の近傍で線形近似することでカルマンフィルタを適用する、拡張カルマンフィルタを用いた追跡も報告されている。

4 パーティクルフィルタ

4.1 非線形システムの推定

カルマンフィルタは、単純な線形計算で予測が可能であり、計算量も少なく安定という利点を持つ一方で、システムモデル、観測モデルが線形で、システムノイズ、観測ノイズは単峰性のガウシアンのみという強い制約がある。そのためコンピュータビジョンへの応用では、画像の観測過程が複雑であるため時系列フィルタリングによって画像から得られる特徴を用いて直接対象の状態推定を行うことは困難であった。

カルマンフィルタの強い制約をなくして、一般的な状態空間モデルに対して、フィルタリングを適用する方法が1980年代から統計学の分野でさかんに研究されてきたが、その中でも1995年に提案されたパーティクルフィルタは、式(5)をランダムサンプリングによるモンテカルロ近似によって推定する方法であり、高次元の状態空間にたいして効率よく状態推定が行える。この手法では、パラ

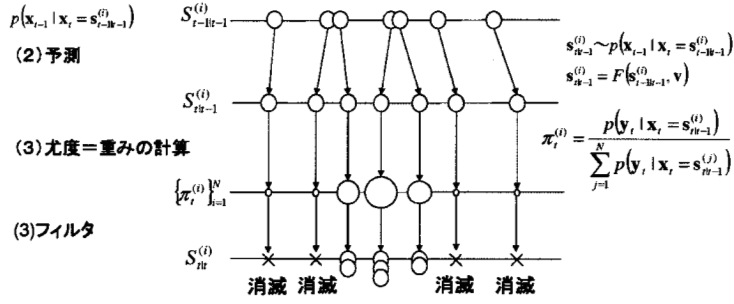


図 3: パーティクルフィルタのアルゴリズム

メトリックな状態ベクトルで表現可能なモデルであれば基本的にはどんな対象でもよく、また観測モデルも尤度関数さえ定義できればよいため、様々な分野で応用されている。

コンピュータビジョンの分野でも、複雑な観測過程を持つ画像の特徴を直接使って状態推定を行えるため、1990年代後半から現在まで、パーティクルフィルタを追跡に応用する研究が多数報告されている。

4.2 基本的なアイデア

パーティクルフィルタの基本的アイデアは、事後分布 $p(\mathbf{x}_t | \mathbf{Y}_t)$ 、及び事前分布 $p(\mathbf{x}_t | \mathbf{Y}_{t-1})$ を、これらの条件付分布に従う多数のサンプル（粒子）を用いて近似するという点である。

事前分布 $p(\mathbf{x}_t | \mathbf{Y}_{t-1})$ に従うサンプル集合を $S_{t|t-1} = \{s_{t|t-1}^{(1)}, \dots, s_{t|t-1}^{(N)}\}$ 、事後分布 $p(\mathbf{x}_t | \mathbf{Y}_t)$ に従うサンプル集合を $S_{t|t} = \{s_{t|t}^{(1)}, \dots, s_{t|t}^{(N)}\}$ とする。このとき、事前分布 $p(\mathbf{x}_t | \mathbf{Y}_{t-1})$ 、事後分布 $p(\mathbf{x}_t | \mathbf{Y}_t)$ は、サンプル集合を用いて以下のように近似される。

$$p(\mathbf{x}_t | \mathbf{Y}_{t-1}) \approx \frac{1}{N} \sum_{i=1}^N \delta(\mathbf{x}_t - s_{t|t-1}^{(i)}) \quad (15)$$

$$p(\mathbf{x}_t | \mathbf{Y}_t) \approx \frac{1}{N} \sum_{i=1}^N \delta(\mathbf{x}_t - s_{t|t}^{(i)})$$

ただし、 $\delta(\cdot)$ はディラックデルタ関数である。

4.3 アルゴリズム

パーティクルフィルタのポイントは、二種類のサンプル集合 $S_{t|t-1}, S_{t|t}$ をそれぞれ事前分布 $p(\mathbf{x}_t | \mathbf{Y}_{t-1})$ 、事後分布 $p(\mathbf{x}_t | \mathbf{Y}_t)$ に従って生成することである。これらの分布は、式(4),(5)に従って推定され、パーティクルフィルタでは、これらの式を、次のような手順でサンプルに適用しながら、

逐次的にサンプル集合を生成する。図3のサンプル集合の生成の概念図を示す。

1. 初期化: $i = 1, \dots, N$ について $s_{0|0}^{(i)} \sim p_0(\mathbf{x})$ を生成する。ただし、 $p_0(\mathbf{x})$ は、あらかじめ与えた初期分布である。 $t := 1$ として以下の手順を実行する。
2. 予測: 各サンプルについて、時刻 t における予測サンプル $s_{t|t-1}^{(i)} \sim p(\mathbf{x}_t | \mathbf{x}_{t-1} = s_{t-1|t-1}^{(i)})$ を次の手順で生成する。

(a) $i = 1, \dots, N$ について、 l 次元の乱数としてシステムノイズ $\mathbf{v}^{(i)} \sim q(\mathbf{v})$ を生成する。ただし、 $q(\cdot)$ はあらかじめ設定したシステムノイズ \mathbf{v} の分布である。

(b) 式(1)のシステムモデル f_t に従って時刻 $t-1$ の各サンプル $s_{t-1|t-1}^{(i)}$ を遷移させて予測サンプルを生成する。

$$s_{t|t-1}^{(i)} = f_t(s_{t-1|t-1}^{(i)}, \mathbf{v}_t^{(i)}) \quad (16)$$

3. 尤度推定: 各予測サンプル $s_{t|t-1}^{(i)}$ について、重み $\pi_t^{(i)}$ を推定する。

$$\pi_t^{(i)} = \frac{p(y_t | \mathbf{x}_t = s_{t|t-1}^{(i)})}{\sum_{i=1}^N p(y_t | \mathbf{x}_t = s_{t|t-1}^{(i)})} \quad (17)$$

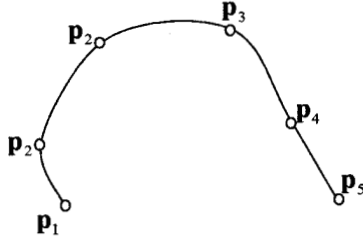
$p(y_t | \mathbf{x}_t = s_{t|t-1}^{(i)})$ は、状態 \mathbf{x}_t が $s_{t|t-1}^{(i)}$ であったときに、観測 y_t を得る確率（尤度）であり、式(2)の観測モデル h_t に従って推定する。

4. フィルタ: $S_{t|t-1} = \{s_{t|t-1}^{(1)}, \dots, s_{t|t-1}^{(N)}\}$ からそれぞれ $s_{t|t-1}^{(i)}$ を重み $\phi_t^{(i)}$ に比例する割合で N 個復元抽出し、 $S_{t|t} = \{s_{t|t}^{(1)}, \dots, s_{t|t}^{(N)}\}$ とする。

このとき、 $S_{t|t}$ は、重み $\phi_t^{(i)}$ の確率に従ってランダムに抽出してもよく、また重み $\phi_t^{(i)}$ が

スプラインモデル

$$\mathbf{x} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_M\} \quad 2M\text{次元}$$



楕円モデル

$$\mathbf{x} = (x, y, \alpha, \beta, \theta) \quad 5\text{次元}$$

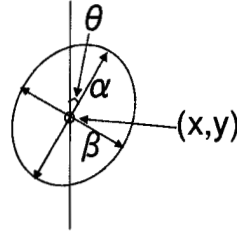


図 4: 動的輪郭モデルと楕円モデル

大きい順に $\mathbf{s}_{t|t-1}^{(i)}$ を $[\phi_t^{(i)} N]$ 回ずつ, N 個になるまで順に選択してもよい.

$t := t + 1$ として, 2 の予測へ.

また, ある時刻 t における推定値 $\hat{\mathbf{x}}_t$ は, 次式のように尤度による重み付き平均によって期待値として求めることができる.

$$\hat{\mathbf{x}}_t = \mathcal{E}(\mathbf{x}_t) \approx \frac{1}{N} \sum_{n=1}^N \mathbf{s}_{t|t}^{(n)} = \sum_{n=1}^N \pi_t^n \mathbf{s}_{t|t-1}^{(n)} \quad (18)$$

5 パーティクルフィルタを用いた追跡の実装

パーティクルフィルタは, 観測モデルとして非線形の任意のモデルを適用できるため, 画像の観測過程が複雑なコンピュータビジョンの問題に適している. そのため近年, 主に対象追跡にパーティクルフィルタを適用した手法が数多く提案されている.

パーティクルフィルタを実際の問題に適用するには, 式 (16) のシステムモデル (状態推移関数) と, 式 (17) 観測モデル (尤度関数) を定義する必要がある. 本説では, パーティクルフィルタを対象追跡に適用する場合について, 状態ベクトルとシステムモデル, 及び観測モデルの実装方法の具体例についていくつか紹介する.

5.1 対象のモデル

パーティクルフィルタでは, パラメトリックな状態ベクトルで表現することができれば, 基本的にはどんなモデルでも適用が可能である. 対象追跡の問題において最も単純なモデルは, 画像上での対象の位置を追跡する場合であり, このとき, 状態ベクトルは画像上の位置 $\mathbf{x}_t = (x_t, y_t)$ となる.

また, 位置だけでなく対象の姿勢やスケールを状態ベクトルとして推定することもできる. Isard ら [1] は人物の頭部を楕円のモデルとし, 楕円の中心位置 x_t, y_t , 長軸の長さ α_t , 短軸の長さ β_t , 長軸の回転角度 θ_t の 5 次元の状態ベクトル $\mathbf{x}_t = (x_t, y_t, \alpha_t, \beta_t, \theta_t)$ を用いている (図??右).

ほかにも, 動的輪郭モデルを用いた輪郭追跡に適用している研究も報告されている. 図??左に示すように, 輪郭を M 個の制御点 $\mathbf{p}_{t,1}, \dots, \mathbf{p}_{t,M}$ によって定義されるスプラインなどの曲線とすると, $2M$ 次元の状態ベクトル $\mathbf{x}_t = (\mathbf{p}_{t,1}, \dots, \mathbf{p}_{t,M})$ となる.

5.2 システムモデル (状態推移関数) の実装

ランダムウォーク

最も単純なシステムモデルは, ランダムウォークである. 状態ベクトルが画像上の座標 $\mathbf{x}_t = [x_t, y_t]^T$ の場合, システムノイズを共分散 Σ_2 の二次元正規分布とすると, f_t は次のように定義される.

$$\begin{aligned} \mathbf{v}_t &\sim N(\mathbf{0}, \Sigma_2) \\ \mathbf{x}_t &= f_t(\mathbf{x}_{t-1}, \mathbf{v}_t^{(i)}) = \mathbf{x}_{t-1} + \mathbf{v}_t^{(i)} \end{aligned} \quad (19)$$

$\mathbf{v}_t \sim N(\mathbf{0}, \Sigma)$ は, 平均 $\mathbf{0}$ で共分散が Σ_2 の二次元正規分布に従って, \mathbf{v}_t をランダムに生成することを意味する.

ランダムウォークのシステムモデルは, 対象の動きが比較的小さい場合や, 複雑な動きの場合に用いられる. また, 姿勢を含めた追跡や, 動的輪郭モデルに適用することもできる.

線形予測モデル

ランダムウォークでは, 対象の動きに関わらず, 前のフレームの対象位置を中心とした分布によってサンプリングを行うため, 速い動きをする対象を追跡する場合に効率の良いサンプリングを行うことができない. そこで, カルマンフィルタで 3.3

節で述べたカルマンフィルタを用いた追跡と同様に、等速直線運動を仮定した、線形な運動予測モデルによるシステムモデルを使うことも多い。

対象の二次元位置の追跡を行う場合、対象位置 x_t, y_t に加えて、対象の速度 \dot{x}_t, \dot{y}_t を状態ベクトルに加えて $\mathbf{x}_t = [x_t, y_t, \dot{x}_t, \dot{y}_t]^T$ とする。

$$\begin{aligned} \mathbf{v}_t &\sim N(\mathbf{0}, \Sigma_A) \\ \mathbf{x}_t &= f_t(\mathbf{x}_{t-1}, \mathbf{v}_t^{(i)}) = F\mathbf{x}_{t-1} + \mathbf{v}_t^{(i)} \\ F &= \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (20)$$

線形予測モデルも、角速度を状態ベクトルに加えて姿勢を含めた追跡などにも適用することが可能である。また、さらに加速度を状態ベクトルに加え、定加速度運動を仮定したモデルに拡張することも可能である。

線形予測モデルでは、対象の動きは予測モデルにあてはまっていれば、速い動きなどに対して効率的なサンプリングが可能であるが、一方でモデルに合わない動き（ボールが床でバウンドするなど）にたいしては追跡に失敗しやすくなるという問題もある。

このほかにも、ボールが定加速度運動をしている状態と、バウンドしている状態などといった離散的な状態変数を状態ベクトルに加えて、それぞれの状態に適した予測モデルを適用する方法も提案されている。

5.3 観測モデル（尤度関数）の実装

観測モデルについては、あるサンプル $\mathbf{s}_{t|t-1}^{(i)}$ を状態パラメータとしたとき、対象のモデルが入力画像にどれだけマッチしているかを評価すればよく、追跡対象によって様々な観測モデルが提案されている。その中でも代表的な例として、Isard らの Condensation[1] で用いられている輪郭を用いた評価と、その複数カメラへの拡張を紹介する。

5.3.1 エッジを用いた評価

エッジを用いた評価では、サンプル $\mathbf{s}_{t|t-1}^{(i)}$ の状態の対象モデルの輪郭と、入力画像中のエッジがどれだけマッチしているかを評価する。

輪郭上のある点 z_j を通り法線方向の直線上に M 個のエッジ点が存在し、それぞれ輪郭から v_m ($m = 1, \dots, M$) だけ離れているものとする。このとき、この点に関する尤度を次のように定義する。

$$p(\mathbf{y}_t | \mathbf{z}_j) \propto 1 + \frac{1}{\sqrt{2\pi\sigma\alpha}} \sum_{m=1}^M \exp\left(-\frac{v_m^2}{2\sigma^2}\right) \quad (21)$$

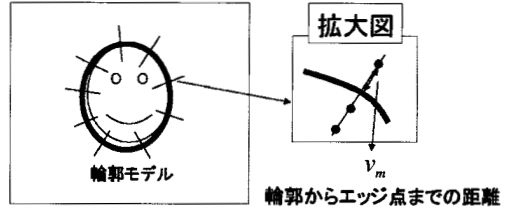


図 5: エッジを用いた評価

ただし、 α, σ は定数であり、それぞれ対象以外のエッジの多さ、対象のエッジと輪郭との距離の標準偏差に応じて適切に設定する。

輪郭上の O 個の点で評価する場合、全体の尤度は次のように計算できる。

$$p(\mathbf{y}_t | \mathbf{x}_t = \mathbf{s}_{t|t-1}^{(i)}) = \prod_{j=1}^O p(\mathbf{y}_t | \mathbf{z}_j) \quad (22)$$

5.3.2 複数カメラによる評価

Isard らの輪郭による評価を、複数カメラを用いた 3 次元追跡に適用した方法 [10, 11, 12] も提案されている。これら手法では、対象を 3 次元空間中の立体（回転楕円体など）とし、状態ベクトルを 3 次元空間中の座標 $\mathbf{X}_t = (X_t, X_t, Z_t)$ として、各サンプルの状態のモデルを各カメラの画像へ投影したものをを用いて、各カメラの画像を用いて評価し、それらを統合する方法である。

あるサンプル $\mathbf{s}_t^{(i)} = (X_t^{(i)}, Y_t^{(i)}, Z_t^{(i)})$ を、カメラ j のカメラ座標に投影した状態を $\mathbf{s}_{j,t}^{(i)} = (x_{j,t}^{(i)}, y_{j,t}^{(i)})$ とする。このとき各カメラにおける尤度 $p(\mathbf{Y}_{j,t} | \mathbf{x}_{j,t} = \mathbf{s}_{j,t}^{(i)})$ が定義されているとすると、元の 3 次元のサンプル $\mathbf{S}_t^{(i)}$ の尤度は次式のように与えられる。

$$p(\mathbf{Y}_t | \mathbf{X}_t = \mathbf{S}_t^{(i)}) = \prod_{j=1}^M p(\mathbf{Y}_{j,t} | \mathbf{x}_{j,t} = \mathbf{s}_{j,t}^{(i)}) \quad (23)$$

5.3.3 その他の尤度関数

その他にも、対象の見えを直接用いて、テンプレートマッチングによる相関などの類似性を尤度として用いる手法 [13] や、AdaBoost などの識別機の出力を尤度として用いる手法 [14, 15] などが提案されている。

5.4 複数対象への拡張

モンテカルロフィルタリングを応用した追跡手法では、対象のパラメータをノンパラメトリック

な分布として表現できる反面、複数対象を扱う場合には、単一対象に対する分布か、複数対象に対する分布かを区別することが困難であるという問題点が生じる。この問題を解決し、複数対象に対応させる手法が提案されている [16, 17, 18, 19]。Isard ら [16] は、対象の個数を状態変数に含め、対象の数が違う状態の尤度推定法を提案し、個数が未知の複数対象追跡を実現している。また、Tweed と Calway [19] は、複雑なシーンで 30 羽までの鳥の群の追跡を行っている。

5.5 計算量の削減

モンテカルロフィルタリングの欠点は、詳細かつ安定な追跡を実現するためには、多数のサンプルを生成する必要がある、速度が低下してしまう点である。このような欠点を解決するために、効率的なサンプリングの手法が提案されている。Isard ら [20] は、あらかじめ肌色検出によって求めた顔や手らしい領域に重点的にサンプリングを行う Importance sampling を提案している。また、MacCormick ら [21] は、密度の粗いサンプリングと詳細なサンプリングの 2 回のサンプリングによって、効率の良く推定する Partitioned Sampling と呼ばれる手法を提案している。

6 まとめ

本稿では、時系列フィルタの概念と、線形システム、ガウス型分布に対するカルマンフィルタから、非線形システム、非ガウス型分布への拡張としてのパーティクルフィルタについて説明し、またそれを対象追跡に適用するばあいの実装方法について紹介した。

パーティクルフィルタは、非線形システム、非ガウス分布に適用可能で高次元の状態空間を効率よく推定することができる時系列フィルタの一種であるが、パラメトリックに表現される対象であればどんな対象でも扱うことができ、また、観測モデルについてもある状態に対する尤度関数さえ定義できればよいといった特長を持つため、観測として複雑な生成過程を持つ画像を扱うコンピュータビジョンの問題にも適しており、近年注目の集まっている手法である。

Mean-Shift [22] や Lucas Kanade [23, 24] の特徴点追跡手法などの、類似度の勾配に基づく追跡手法と比べると、図 6(a) に示すように、尤度分布のピーク周辺に極大値が存在する場合、勾配に基づく手法では、最適解に収束するのは難しいが、パーティクルフィルタではランダムサンプリングによって確率的に推定するため、このような場合でも正しく推定することが容易である。一方で、図 6(b) に示すように、尤度分布が鋭いピークを持ち、ピーク位置の周辺で急激に尤度が下がるような場合、サ

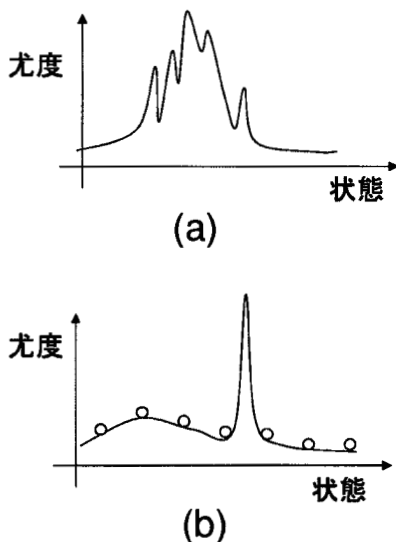


図 6: 尤度分布

ンプルの密度が低いと正しくピーク位置を推定できない場合が存在する。

このようなことから、パーティクルフィルタは、どんな追跡問題にたいしても比較的適用がたやすい手法であるが、尤度関数の設計によっては、効率よく推定できない場合も存在し、この尤度関数の設計をどうするか、パーティクルフィルタの適用において最も重要なポイントであるといえる。

なお、パーティクルフィルタの実装例としては、Isard らによる The Condensation Algorithm のホームページ [25] からダウンロードできる、simple cond や、OpenCV ライブラリ [26] に含まれる cvCondensation などが有名である。cvCondensation の使用例は、[27] のホームページなどが参考にできる。

参考文献

- [1] Isard, M. and Blake, A.: CONDENSATION – Conditional Density Propagation for Visual Tracking, *International Journal of Computer Vision*, Vol. 29, No. 1, pp. 5–29 (1998).
- [2] Kalman, R.: A new approach to linear filtering and prediction problems, *Trans. ASME – Journal of Basic Engineering*, pp. 35–45 (1960).
- [3] Bar-Shalom, Y. and Fortmann, T. E.: *Tracking and Data Association*, Academic Press (1988).

- [4] Wang, H., Goh, W. L., Chua, C. S. and Sim, C. T.: Real-time object tracking, in *21st International conference on industrial electronics*, pp. 1366–1370 (1995).
- [5] Reid, I. D. and Murray, D. W.: Active tracking of foveated feature clusters using affine structure, *International Journal of Computer Vision*, Vol. 18, pp. 41–60 (1996).
- [6] Ziliani, F. and Moscheni, F.: Kalman filtering motion prediction for recursive spatio-temporal segmentation and object tracking, in *Workshop on Image Analysis for Multimedia Interactive Services* (1997).
- [7] Manku, G. S., Jain, P., Aggarwal, A., Kumar, L. and Banerjee, S.: Object Tracking using Affine Structure for Point Correspondences, in *Computer Vision and Pattern Recognition (CVPR'97)*, pp. 704–708 (1997).
- [8] Jung, S. K. and Wahn, K. Y.: 3-D tracking and motion estimation using hierarchical Kalman filter, in *Vision, Image and Signal Processing*, pp. 293–298 (1997).
- [9] Jebara, T., Russell, K. and Pentland, A.: Mixtures of eigenfeatures for real-time structure from texture, in *6th International Conference on Computer Vision*, pp. 128–135 (1998).
- [10] 鈴木達也, 岩崎慎介, 小林貴訓, 佐藤洋一, 杉本晃宏: 環境モデルの導入による人物追跡の安定化, *電子情報通信学会論文誌 DII*, Vol. J88-DII, No. 8, pp. 1592–1600 (2005).
- [11] Wang, Y., Wu, J. and Kassim, A.: Particle Filter for Visual Tracking Using Multiple Cameras, in *IAPR Conferent on Machine Vision Applications*, Vol. 1, pp. 298–301 (2005).
- [12] 松元郁佑, 加藤丈和, 和田俊和: 複数カメラを用いた Condensation によるオクルージョンにロバストな人物追跡, 画像の認識・理解シンポジウム (MIRU2006), pp. 291–506 (2006).
- [13] 松原康晴, 尺長健: 疎テンプレートマッチングとその実時間追跡への応用, *電子情報通信学会論文誌*, Vol. 46, No. SIG CVIM11, pp. 60–71 (2005).
- [14] 小林貴訓, 杉村大輔, 平澤宏祐, 鈴木直彦, 佐藤洋一, 杉本晃宏: カスケード型識別機を用いたパーティクルフィルタによる人物三次元追跡, 画像の認識・理解シンポジウム (MIRU2006), pp. 222–228 (2006).
- [15] Okumura, K., Taleghani, A., Freitas, N. and Gool, J. V.: A Boosted Particle Filter: Multitarget Detection and Tracking, in *European Conference on Computer Vision*, pp. 28–39 (2004).
- [16] Isard, M. and MacCormick, J.: BraMBLe: A bayesian multiple blob tracker, in *International Conference on Computer Vision (ICCV2001)*, Vol. 2, pp. 34–41 (2001).
- [17] MacCormick, J. and Blake, A.: A probabilistic exclusion principle for tracking multiple objects, *International Journal of Computer Vision*, Vol. 39, No. 1, pp. 57–71 (2000).
- [18] Tao, H., Sawhney, H. and Kumar, R.: A sampling algorithm for tracking multiple objects, in *IEEE Workshop on Vision Algorithms*, pp. 53–68 (1999).
- [19] Tweed, D. and Calway, A.: Tracking Many Objects Using Subordinated CONDENSATION, in *Proceedings of the British Machine Vision Conference*, pp. 283–292 (2002).
- [20] Isard, M. and Blake, A.: ICondensation: Unifying low-level and high-level tracking in a stochastic framework, in *5th European Conference Computer Vision (ECCV1998)*, pp. 893–908 (1998).
- [21] MacCormick, J. and Isard, M.: Partitioned sampling, articulated objects, and interface-quality hand tracking, in *European Conference on Computer Vision*, pp. 3–19 (2000).
- [22] Comaniciu, D., Ramesh, V. and Meer, P.: Kernel-based object tracking, pp. 564–577 (2003).
- [23] Lucas, B. and Kanade, T.: An iterative image registration technique with an application to stereo vision, in *IJCAI* (1981).
- [24] Tomasi, C. and Kanade, T.: Detection and tracking of point features, Technical Report CMU-CS-91-132, Carnegie Mellon University (1991).
- [25] The Condensation Algorithm, http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/ISARD1/condensation.html.
- [26] The OpenCV Library, <http://sourceforge.net/projects/opencvlibrary/>.
- [27] cvCondensation sample, <http://nautilus.cs.miyazaki-u.ac.jp/yoshi/pukiwiki/index.php?Files/condensation.cpp>.