

# パーティクルフィルタと特徴点検出を用いた 物体追跡に関する実験的検討

山西 健太<sup>†</sup> 鈴木 幸司<sup>†</sup>

<sup>†</sup> 室蘭工業大学

## 1 はじめに

動画中の物体追跡はロボットや画像認識などさまざまな応用がなされているが、実用性を高めるためには動画内からリアルタイムで対象を検出、追跡することが必要である。本研究では物体追跡の手法の一つとしてパーティクルフィルタを使用して物体追跡を行う。パーティクルフィルタを使用した物体追跡では、尤度推定を行う手法が重要である。そこで本研究では尤度推定の部分に特徴点検出 (SURF[1]、FAST[2][3]) を組み合わせた場合の有効性に対して処理時間と追跡率の2つの点に着目して実験的に検討した。

## 2 パーティクルフィルタ

パーティクルフィルタは物体の動きを多数のパーティクルの密度で近似することで追跡する手法である。予測、尤度推定、フィルタの3段階の処理を行うことで対象物体の位置を推定する。

### 2.1 予測

各パーティクル毎に前の時刻の状態から現在の状態を予測する。本研究では (1) 式に示す状態遷移モデルを使用した。

$$\vec{x}_t = 2\vec{x}_{t-1} - \vec{x}_{t-2} + \vec{w}_t \quad (1)$$

ここで  $\vec{x}_t$  は時刻  $t$  における各パーティクルの座標、 $\vec{w}_t$  は正規乱数を表す。

### 2.2 尤度推定 [4]

各パーティクルごとに追跡対象と同サイズの領域  $R$  の共分散行列を計算、比較することで尤度を推定する。本研究では (2) 式の特徴ベクトルを使用した。

$$\vec{f} = [x \ y \ I_R \ I_G \ I_B \ I_x \ I_y] \quad (2)$$

ここで  $x$ 、 $y$  はピクセルの座標、 $I_R$ 、 $I_G$ 、 $I_B$  はピクセルのRGB値、 $I_x$ 、 $I_y$  はピクセルの一次勾配である。この特

徴ベクトル  $\vec{f}$  を使用してパーティクルごとに共分散行列  $C_R$  を計算する。追跡対象の共分散行列  $C_0$  と各パーティクルの共分散行列  $C_R$  を用いて (3) 式、(4) 式より尤度  $P_R$  を計算する。

$$\rho(C_0, C_R) = \sqrt{\sum_{i=1}^d \ln^2 \lambda_i(C_0, C_R)} \quad (3)$$

$$P_R = e^{-\alpha \cdot \rho(C_0, C_R)^2} \quad (4)$$

ここで、 $\lambda_i(C_0, C_R)$  は  $C_0$  と  $C_R$  の一般化固有値、 $\alpha$  はパラメータであり、本実験では 0.2 とした。尤度  $P_R$  を正規化することで各パーティクルの重みを設定する。

### 2.3 フィルタ

尤度推定の処理で計算した重みに基づいてパーティクルの位置を定めていく。このときパーティクルの総数は変化させないように設定する。本研究では重みが最大であるパーティクルの領域を追跡対象の位置とした。

## 3 観測手法

本研究では前述のように追跡対象とパーティクルの周囲の領域を比較することで位置の推定を行っている。しかしこの方法では追跡対象の存在しない部分の領域に関しても計算を行う必要がある。そこで本研究では尤度推定の処理に特徴点検出を組み合わせることで処理時間の軽減を試みる。図1のようにパーティクルの周囲の領域内に特徴点が存在する場合にのみ共分散行列を計算することで計算量を軽減させる。

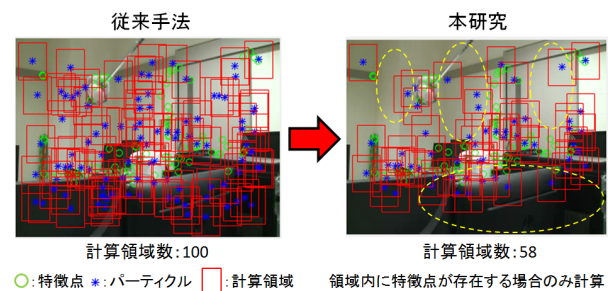


図1: 計算量の軽減

## Experimental Study on Object Tracking Using Particle Filter and Feature Points

Kenta Yamanishi<sup>†</sup>, Yukinori Suzuki<sup>†</sup>

<sup>†</sup> Muroran Institute of Technology

## 4 実験

### 4.1 実験方法

動画に対してパーティクルフィルタを適用して追跡を行い、特徴点検出が物体追跡に有効であるか検討する。有効であるかどうかの指標として処理時間と追跡率の2点を使用した。本研究ではサイズ  $320 \times 240$  [pixels]、フレーム数 297 [frames]、フレームレート 29 [fps] の avi 形式の動画を使用、パーティクル数 100 とし、MATLAB を使用して実験を行った。追跡対象は図 2 に示す矩形領域とし、撮影者が任意に動かした規則性のない動きをする物体である。特徴点検出をしない場合、SURF、FAST を組み合わせた場合について実験を行い比較した。また特徴ベクトルの次元数の違いの影響についても実験を行った。(2) 式に示した 7 次元の特徴ベクトルを使用した場合と  $x$ 、 $y$ 、 $I_R$ 、 $I_G$ 、 $I_B$  の 5 次元の特徴ベクトルを使用した場合について実験した。

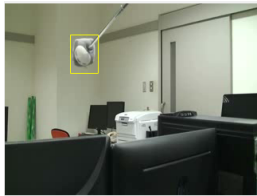


図 2: 追跡対象

### 4.2 実験結果・考察

処理時間について図 3 に示す。特徴点検出方法で比較すると、FAST、SURF、特徴点検出をしない場合の順に処理時間が短い結果となった。次元数が異なる場合でも同様の順に処理時間が短くなっていることがわかる。特徴ベクトルの次元数を 5 個から 7 個に増やした場合、処理時間は増えている。しかしフレーム数のことを考慮するとあまり大きな違いが出てはるわけではないため、追跡率によっては特徴ベクトルの次元数を増やすことも物体追跡に有効であるといえる。

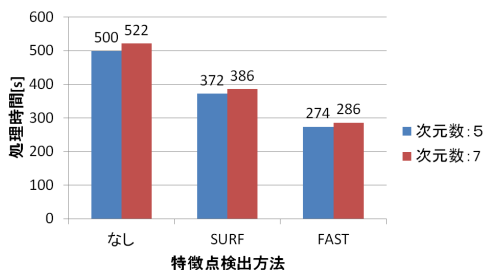


図 3: 処理時間

追跡率については目視で追跡できているか確認した。297 フレームの動画を 8 フレームごとに確認し追跡対象の位置が正しく表示できているか判定、追跡できている割合を求めた。図 4 に追跡率を示す。

特徴点検出をしない場合と SURF を組み合わせた場合は次元数 5 のときは 83% と 89%、次元数 7 のときは 72% と 75% の割合で追跡出来ていた。一方で、FAST を組み合わせた場合は次元数 5 のとき 48%、次元数 7 のとき 54% しか追跡できていない結果となった。これは FAST で特徴点検出を行った時に追跡対象に特徴点が発見されていないフレームが多いことが原因であると思われる。

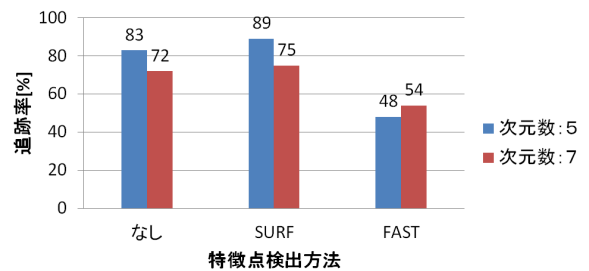


図 4: 追跡率

## 5 まとめ

物体追跡に特徴点検出を組み合わせることは有効であるといえる結果となった。しかし追跡対象に特徴点が発見されるように適切な特徴点検出方法を定める必要がある。特徴ベクトルに関しても使用する値を適切に選択する必要がある結果となった。処理時間に関してはリアルタイムで処理は出来ていない。そのため計算コストの高い共分散行列の計算に対して処理時間の短縮を行う手法について検討する必要がある。

## 参考文献

- [1] H. Bay, A. Ess, T. Tuytelaars and L. Van Gool, Speeded-Up Robust Features(SURF) CVIU, vol. 110, No. 3, pp.346-359, 2008.
- [2] E. Rosten and T. Drummond, Fusing Points and Lines for High Performance Tracking, Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, vol. 2, pp.1508-1515, 2005.
- [3] E. Rosten and T. Drummond, Machine learning for high-speed corner detection, Computer Vision-ECCV, pp.430-443, 2006.
- [4] Y. Su, Q. Zhao, L. Zhao and D. Gu, Abrupt motion tracking using a visual saliency embedded particle filter, Pattern Recognition, vol. 47, pp.1826-1834, 2014.