

パーティクルフィルタを用いた進入検出と物体追跡

松村 遼* 岡村 健史郎**

Object Detection and Tracking Using Particle Filter

Ryo MATSUMURA and Kenshiro OKAMURA

Abstract

Object tracking is an important role in many areas of engineering such as surveillance and computer vision. Over the last few years, particle filters have been proved to be powerful tools for object tracking. To adopt particle filters, the number of tracking objects is very important. In this paper, we propose a method of entrance detection in the specific areas using two-state particle filters. We demonstrate the algorithm on a set of real object targets whose image contains a number of peoples, bicycles and cars.

Key words: Particle Filter, object tracking, multi-target detection

1. まえがき

犯罪防止を目的とした物体検出・追跡システムなどの監視システムへの期待が高まっている。近年、パーティクルフィルタを用いた3次元空間上での物体追跡についての研究が多く報告されている^{[1][2]}。パーティクルフィルタを用いた物体追跡ではパーティクル群によって多数の仮説を保持し、ロバストな物体追跡を可能にしている^[3]。

パーティクルフィルタを用いた物体追跡では、一つの物体ごとにフィルタと呼ばれるパーティクルの集合を一つ与えることによって追跡を行う。そのため監視領域内に存在する物体数を正確に知ることが必要となり、監視領域内への物体の進入と退出を検出することが重要となる。しかしこれまでの研究では監視領域内へ同時に進入してくる物体は常に一体という環境で実験を行っており、複数の物体が同時に進入してくる場合については考察が不十分であった^{[1][2]}。そこで、本稿ではパーティクルフィルタを用いた3次元空間上での物体追跡において、複数物体が同時に進入してくる場合について検討を行った。

本稿では、パーティクルフィルタを検出のみを行う状態と追跡を行う状態に区別する。前者を検出フィルタ、後者を追跡フィルタと呼ぶ。物体が進入してくると考えられる場所に進入物体検出領域を設け、そこに検出フィルタを設置することによって物体の進入検出を行う。この検出フィルタが検出領域内で物体の進入を検出したとき、この物体の追跡を開始するために追跡フィルタとするとともに、別の物体の進入に備えるため新たな検出フィルタを設置する。この時、検出領域内に存在する追跡フィルタの領域を定め、検出フィルタがこの領域を探索しないようにすることで検出フィルタが既に追跡中の物体を重複して検出しないよう工夫した。これにより複数物体の同時進入の検出が可能となった。

本稿では、キャリブレーション済みの固定単眼カメラを用いて撮影した動画を対象にした実験にて、3次元空間上での物体追跡と複数物体の同時進入検出の例を示す。まず、**2.**にてパーティクルフィルタを用いた物体追跡について述べ、**3.**では世界座標と画像座標の対応について述べる。次に、**4.**では物体の進入および退出の検出手法について述べ、**5.**の実験では、監視領域内への進入物体検出実験の結果を示す。**6.**で本稿の結論について述べる。

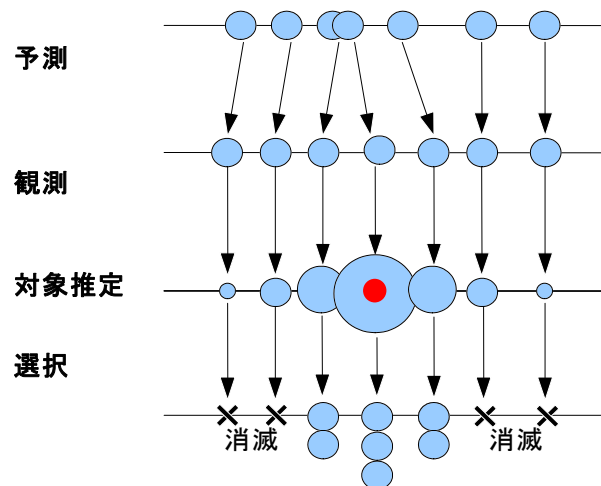


図 1 パーティクルフィルタの概略図

2. パーティクルフィルタを用いた物体追跡

パーティクルフィルタとは多数のパーティクルを用いて、前状態からの予測と現在の観測情報から現在の状態を推定する手法である。図 1 にパーティクルフィルタの概略図を示す。図中の丸はパーティクルを表している。パーティクルフィルタとは図 1 のように「予測」「観測」「対象推定」「選択」という操作を時間ごとに繰り返すことによって物体の追跡を行う。図 1 のようにパーティクルの重みが高ければ復元抽出により仲間を増やし、重みが低ければ消滅する。

パーティクルフィルタでは物体が存在すると考えられる部分の尤度と重みを大きくすることで物体が存在すると考えられる付近にパーティクルが集中ようになる。本稿では尤度を背景差分により決定する。背景差分とは入力画像と、予め取得しておいた検出対象が存在しない背景画像との差分をとることにより物体を抽出する方法である。

以下でパーティクルフィルタによる 3 次元空間上での追跡アルゴリズムについて説明する。なお、時刻 t での i 番目のパーティクルを $\mathbf{x}_t^{(i)} = (x_t^{(i)}, y_t^{(i)}, z_t^{(i)})$ と表す。また、時刻 t での i 番目のパーティクルの尤度と重みをそれぞれ $\pi_t^{(i)}$, $w_t^{(i)}$ と表し、パーティクルの数を N とする。

a) 予測

時刻 t でのパーティクルの位置を予測する。本稿では予測モデルとして、式(1)で示されるモデルを用いた。 $\mathbf{w}_t^{(i)} \sim \mathbf{N}(0, \sigma^2)$ は 3 次元のガウシアンノイズである。 \mathbf{v}_{t-1} は時刻 $t-1$ での物体速度で、これは以降で述べる「対象推定」によって得られた物体中心位置から式(2)を用いて求める。ここで \mathbf{P}_{t-1} , \mathbf{P}_{t-2} はそれぞれ時刻 $t-1$, $t-2$ での物体中心位置を表す。

$$\mathbf{x}_t^{(i)} = \mathbf{x}_{t-1}^{(i)} + \mathbf{v}_{t-1} + \mathbf{w}_t^{(i)} \quad (1)$$

$$\mathbf{v}_{t-1} = \mathbf{P}_{t-1} - \mathbf{P}_{t-2} \quad (2)$$

b) 観測

パーティクルの尤度と重みを求める。本研究では 3 次元空間上のパーティクルを **3.** で述べる世界座標と画像座標の対応の式によって投影した点 (u, v) の、入力画像と背景画像の背景差分値を尤度とする。入力画像 f_I における投影された点の画素値を $(r_I(u, v), g_I(u, v), b_I(u, v))$, 背景画像 f_B の画素値を $(r_B(u, v), g_B(u, v), b_B(u, v))$ とするとパーティクル i の尤度は次式によって求める。

$$\pi_t^{(i)} = \sqrt{(r_I(u, v) - r_B(u, v))^2 + (g_I(u, v) - g_B(u, v))^2 + (b_I(u, v) - b_B(u, v))^2} \quad (3)$$

また、背景の明度変化によるノイズの影響を除去するため、閾値 α 以下の尤度を持つパーティクルは尤度を 0 とするようにした。そして得られた尤度から重みを決定する。重み $w_t^{(i)}$ は、以下の式(4)のように集団の尤度で正規化したものを用いる。ただし式(4)の分母が 0 になる場合は $w_t^{(i)} = 1/N$ とした。図 1 において「観測」後のパーティクルはその丸の大小により重みの大きさを表している。

$$w_t^{(i)} = \pi_t^{(i)} / \sum_{i=1}^N \pi_t^{(i)} \quad (4)$$

c) 対象推定

以下の式(5)のように重み付き平均により、時刻 t での物体中心位置 \mathbf{P}_t を推定する。図 1 では推定された物体中心位置を赤い点で表している。

$$\mathbf{P}_t = \sum_{i=1}^N w_t^{(i)} \mathbf{x}_t^{(i)} \quad (5)$$

d) 選択

パーティクルの重みに比例した確率でパーティクルを復元抽出し、時刻 t でのパーティクル集合を得る。この時、図 1 に示すように重みの大きいパーティクルは何度も選択され、重みの小さいパーティクルは選択されず消滅する。

このアルゴリズムを用いた物体の進入検出と追跡は次のように行う。まず、初期設定として予め決めた特定の領域に一樣にパーティクルを発生させ、それぞれの場所の観測のみを行う。これは予測や選択により、パーティクルが移動し検出領域を出たり、特定領域に集中し進入物体を見逃したりすることを防ぐためである。この状態にあるフィルタを特に検出フィルタと呼ぶ。検出フィルタにおいて尤度が一定条件を満足すると、物体が進入したと判断し全ての処理を行う。この状態にあるフィルタを追跡フィルタと呼ぶ。

図 2 にはパーティクルフィルタにより、進入してきた物体に対する追跡を行った結果を示す。(a)に入力画像、(b)に背景画像、(c)に追跡結果画像を示す。この時、パーティクル数は 500 に設定し、画像中の赤い点がこれらのパーティクルを表す。このように追跡を行うと物体にパーティクルが集中する。進入してきた車以外の画像上部にある車は背景画像内に存在したため、背景と判断され追跡対象から除かれている。図 3 には追跡結果の連続画像を示す。画像は毎秒 30 フレームで撮影を行い、撮影開始からのフレーム数を時刻 t とおいた。また、追跡中に現れた四つの物体を A~D で示す。図では四つの物体が連続してそれぞれ同じフィルタにより追跡されていることを示している。

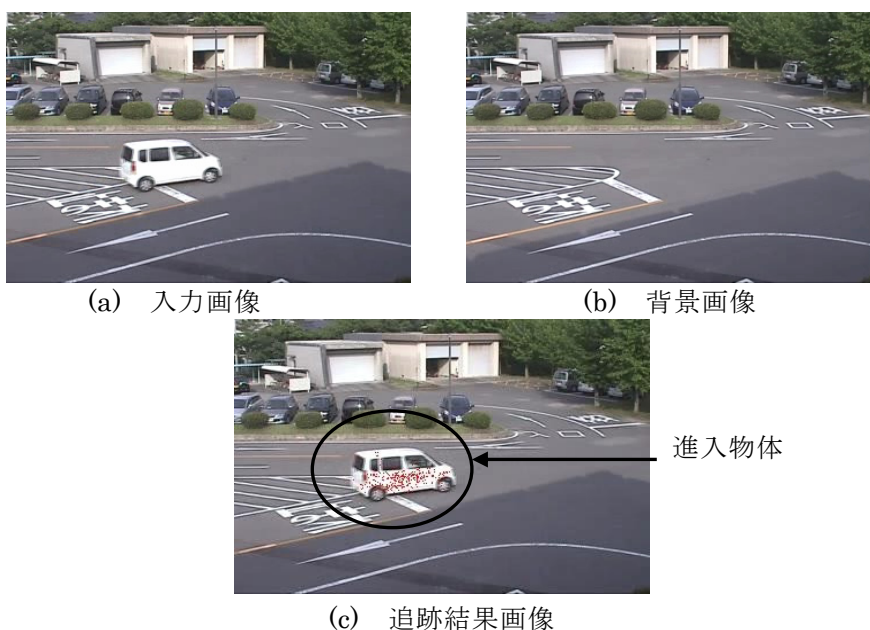


図 2 パーティクルフィルタによる物体追跡

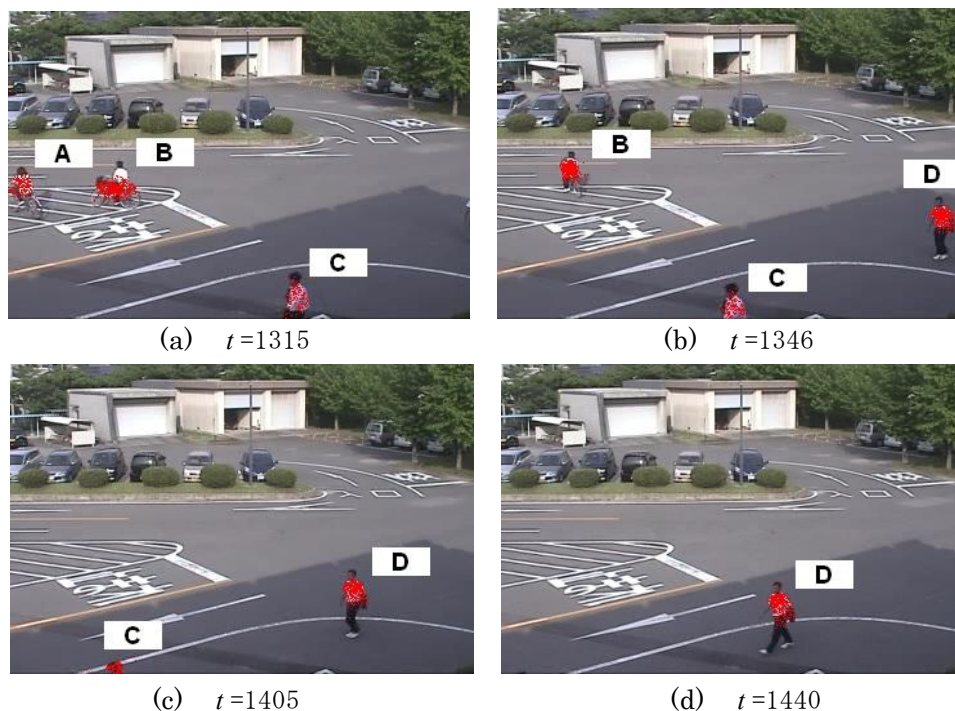


図3 追跡結果連続画像

3. 世界座標と画像座標との対応

3次元空間上で物体追跡を行うには、カメラキャリブレーションを行い、3次元空間である世界座標系と2次元画像平面との対応関係を求める必要がある。世界座標系の点を $P(x, y, z)$ 、 P に対応する画像座標系の点を $p(u, v)$ とすると、 P と p の関係は点 P の同時座標表現 $P = [X, Y, Z, 1]^T$ と点 p の同時座標表現 $p = [u, v, 1]^T$ を用いて次のように表せる^[4]。

$$sp = \mathbf{A}[\mathbf{R} | \mathbf{T}]P \quad (6)$$

ここで、 $[\mathbf{R} | \mathbf{T}]$ はカメラの外部パラメータ行列、 \mathbf{A} は内部パラメータ行列、 s はスケール係数を表す。内部パラメータ行列 \mathbf{A} は

$$\mathbf{A} = \begin{bmatrix} fs_u & fk_s & u_0 \\ 0 & fs_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

と表される。 f は焦点距離、 s_u, s_v は u, v 方向の画素サイズ、 u_0, v_0 は画像平面とカメラ光軸の交点である主点座標、 k_s は画像平面上の u 軸と v 軸の傾きを表すせん断係数である。 \mathbf{R} は 3×3 のカメラの回転行列で、 \mathbf{T} は 3×1 のカメラの平行移動行列であり、それぞれ

$$\mathbf{R} = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \quad (8)$$

$$\mathbf{T} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (9)$$

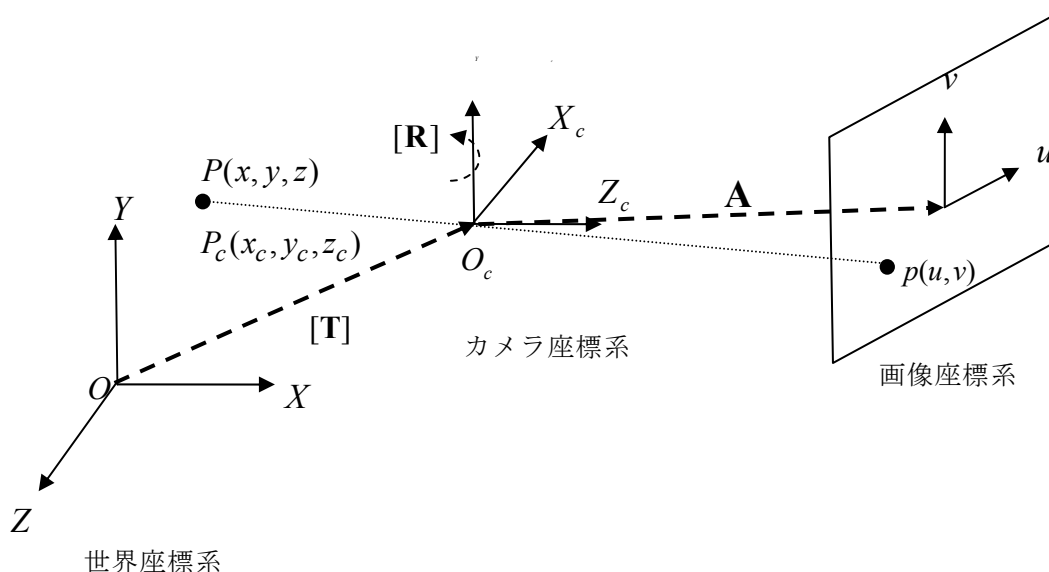


図 4 世界座標系から画像座標系への投影

と表される。図 4 に世界座標系から画像座標系への投影の様子を示す。図 4 のように世界座標系で表される点 $P(x, y, z)$ を回転行列と平行移動行列をあわせた外部パラメータ行列 $[R | T]$ を用いて、カメラ座標系の点 $P_c(x_c, y_c, z_c)$ に変換する。そして内部パラメータ行列 A により画像座標系に投影し点 $p(u, v)$ を得る。

内部パラメータ行列 A と外部パラメータ行列 $[R | T]$ は OpenCV ライブラリを用いて求めた^[4]。OpenCV とは Intel 社による画像処理向けの C++ 用ライブラリである。内部パラメータ行列 A は一マスの一辺が 5 cm の 5×5 マスの白黒チェックパターンを用いて推定し、外部パラメータ行列 $[R | T]$ の推定には事前に計測した 6 点を用いた。

推定した内部パラメータ行列 A と外部パラメータ行列 $[R | T]$ を用いて、式(6)で示される対応式によって図 5 中に示される 3 点 A~C の対応を検証した。検証は次のように行った。まず、3 点の世界座標での座標 (x, y, z) を実測し、この座標を式(6)によって変換し投影点の座標 (u, v) を求める。次に、画像からこれら 3 点の 2 次元上の座標 (u', v') を求め、最後にこれら (u, v) と (u', v') の値を比較することで、式(7)~(9)にあるパラメータの精度を確かめた。表 1 に検証の結果を示す。入力画像のサイズは 360×240 である。この表に示されるように図中 3 点の誤差は最大でも 3 pixel 程度で、許容可能な範囲だといえる。

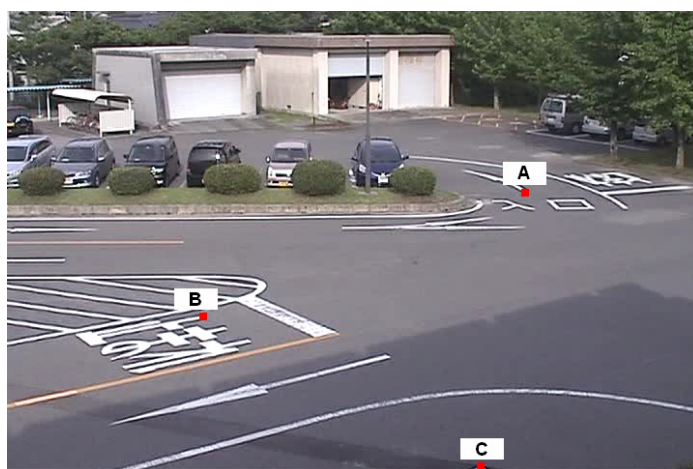


図 5 座標系対応検証点

表 1 変換パラメータ検証結果

	世界座標 (x, y, z) [cm]	式(6)で求めた 投影座標(u, v) [pixel]	画像座標(u', v') [pixel]	誤差($u - u', v - v'$) [pixel]
A	(1810, 460, 0)	(269, 93)	(267, 94)	(+2, -1)
B	(270, -170, 0)	(102, 157)	(101, 157)	(+1, 0)
C	(250, -960, 0)	(241, 236)	(244, 235)	(-3, +1)

本稿では実世界である 3 次元空間上にパーティクルを分布し、投影される画像でのパーティクルの尤度を求める。3 次元空間上で追跡を行うため、カメラから遠く離れた画像に小さく写る物体でも追跡が可能であると考えられる。

4. 物体の進入退出検出

物体の追跡は一つの物体ごとにフィルタと呼ばれるパーティクルの集合を一つ与えることによって行う。そのため、物体の進入や退出を精度よく検出する必要がある。そこで文献[1]をもとに複数物体の同時進入に対応させるよう以下のように進入物体検出方法を考える。

図 6 のように物体が進入してくる領域に進入物体検出領域を設け、この領域内にパーティクルを発生させ、進入物体検出用の検出フィルタを設置する。文献[1]では追跡フィルタで表される物体が検出領域から移動し十分離れたところで新たな検出フィルタを設置していた。これでは検出領域内に同時に進入してくる物体を検出することはできない。そこで本稿では進入物体を検出し追跡を開始すると同時に別の物体に備えて新たな検出フィルタを設置するようにした。この時、検出領域内に存在する追跡中物体の領域を定めることで、すでに追跡中の物体を重複して検出しないようにした。これにより、同じ物体が何度も検出されることを防ぎ、一つのフィルタで一つの物体を追跡することができる。また複数物体の進入にも対応することができる。**2.** で述べたようにこの検出フィルタは「観測」のみを行う。検出フィルタは 0 より大きな値の尤度を持つパーティクルの数が閾値 β を超えると物体が進入したとして、追跡フィルタとなり **2.** で述べた a) ~ d) までの処理を行う。

図 7 に進入物体検出の一連の様子を示す。黄色い楕円は物体を表し、図中の水色の丸はパーティクルを表す。図 7(a)では A, B の二つの物体が同時に進入してきている。このとき、検出領域内において物体 A を検出し、検出フィルタが追跡フィルタとなったとき、追跡フィルタの領域 R をそのパーティクルの分散から決定する(図 7(b))。同時に、別の物体 B の進入に備えるため新たな検出フィルタを設置する(図 7(c))。この検出フィルタは、領域 R 内にあるパーティクルの尤度は 0 とすることで、追跡中の物体 A を検出せず、物体 B のみを検出する(図 7(d))。



図 6 進入物体検出領域

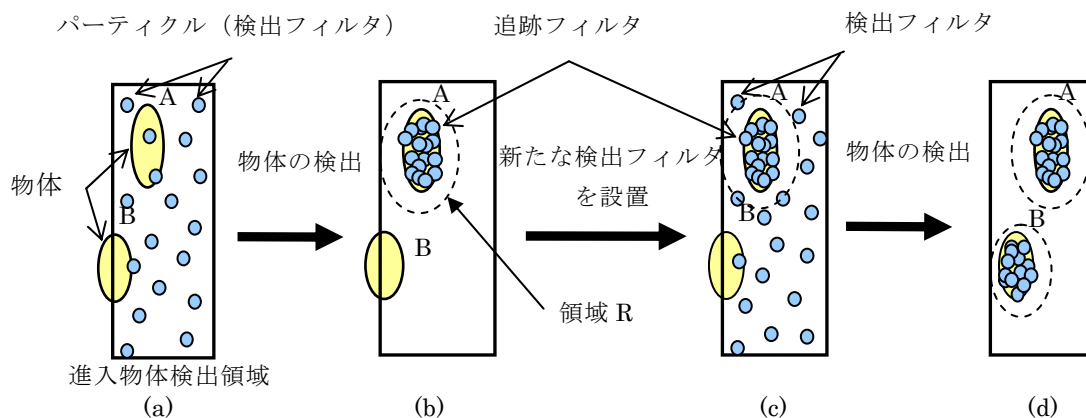


図7 進入物体の検出

追跡フィルタは0より大きな尤度を持つパーティクルの数が閾値 γ を下回ると物体が退出したとして、この追跡フィルタを削除する。以上をまとめると次のアルゴリズムとなる。なお時刻 t での k 番目のフィルタを $F_t^{(k)}$ とし、フィルタの数を K と表す。また、パーティクルの x, y, z 方向の重み付き分散がそれぞれ $\sigma_x^2, \sigma_y^2, \sigma_z^2$ であるとき、追跡フィルタ k の領域 $R_t^{(k)}$ を x, y, z 方向の半径がそれぞれ $2\sigma_x, 2\sigma_y, 2\sigma_z$ である楕円で表す。

進入物体の検出と追跡の終了アルゴリズム

(1) 初期設定

物体が進入してくると考えられる領域に検出フィルタを設置し、その個数を K とする。

(2) 追跡および退出処理

フィルタ $F_t^{(k)} (k=1, \dots, K)$ のなかで追跡フィルタに対してのみ2. で述べた(1)～(4)までの処理を適用する。ただし「観測」の後に退出処理を行い、0より大きな尤度を持つパーティクルの数が閾値 γ を下回る追跡フィルタは物体が退出したとして削除し、 K の値を1減じる。

(3) 進入検出処理

フィルタ $F_t^{(k)} (k=1, \dots, K)$ のなかで検出フィルタに対してのみ「観測」を適用する。「観測」は追跡フィルタの領域 $R_t^{(k)}$ の外にあるパーティクルのみ行う。0より大きな尤度を持つパーティクルの数が閾値 β を超えた検出フィルタは物体が進入したとして追跡フィルタへ移行し「対象推定」「選択」を適用する。

(4) 新たな検出フィルタの設置

(3)の処理において、追跡フィルタへ移行した検出フィルタが存在した場合、新たな検出フィルタを設置し、 K の値を1増やす。

(5) (2)へ戻る

5. 実験

固定したカメラで撮影された単眼動画像に対し、本手法を用いて、進入物体の検出実験を行った。検出対象は人物、車、自転車、バイクの四種類である。

5.1 実験条件

実験に用いたコンピュータの OS は Fedora 6 で、CPU は Pentium4 の 2.60GHz である。動画は毎秒 30 フレームの取り込みを行った約 10 分の長さである。この動画の特徴として、多くの人物が制服を着用しているため同じ色調の物体が多いということ、制服の色が黒や濃紺で地面のアスファルトと区別が難しいこと、さらに学校の下校時の時間帯を撮影したため人物が集団で行動するパターンが多いということ、などが挙げられる。表 2 に実験で用いたパラメータを示す。

「進入検出・退出の処理」で用いた追跡開始閾値 β と追跡終了閾値 γ は、追跡中の物体がノイズ等の影響で退出したと誤って判断されないようにするため、 $\beta > \gamma$ となるように設定する。本実験ではパーティクル数 N に対して $\beta = N/10 + 5$, $\gamma = N/10 - 5$ とした。進入物体検出領域は図 6 中に示す位置で、表 3 で示される大きさに設置する。

表 2 実験パラメータ

画像サイズ [pixel]	360×240
パーティクル数 N	500
式(1)で用いたガウシアンノイズの標準偏差 σ [cm]	30
「観測」で用いた尤度値判定の閾値 α	25
追跡開始閾値 β	55
追跡終了閾値 γ	45

表 3 進入物体検出領域

	Δx [m]	Δy [m]	Δz [m]
右側の検出領域	0.15	15	1.7
左側の検出領域	0.15	47	1.7

実験 1 ではカメラから遠く離れた場所に進入してきたため画像に小さく写る物体でも検出ができるかを確認する実験を行い、実験 2 では本稿で提案した複数物体検出手法で検出が行えることを確認するための実験を行った。

5.2 進入物体検出実験

5.2.1 実験 1 カメラから遠く離れた場所への進入

図 8 は 3 次元空間上でカメラから遠く離れた場所にバイクが進入してきた画像である。(a)に入力画像を示し、(b)に結果画像、(c)にバイクを拡大した画像、(d)に結果を拡大した画像を示す。(b)、(d)の画像中の赤い点はパーティクルを表す。このようにカメラから離れた場所に存在し、画像上に小さく写る物体でも検出し追跡することができた。これは 3 次元空間上でカメラから離れた場所にある検出領域内のパーティクルが画像上に投影される場合、画像上でのパーティクル同士の間隔が密になり、その結果小さく写る物体が進入してきても閾値 α 以上のパーティクル数が十分多くなり追跡開始閾値 β を超え検出されたためである。

5.2.2 実験 2 複数物体の進入

図 9 に複数の物体が同時に進入してきた場合の結果を示す。図において、撮影開始からのフレーム数を時刻 t とおき、追跡フィルタの領域を赤い楕円で表示した。 $t=1934$ において 2 台の自転車が右より同時に進入してきている。この時、先に上側の自転車が検出され追跡フィルタとなり、その後 $t=1938$ において新たに設置した検出フィルタにより下側の自転車が検出された。**4.** で述べたように物体が検出されると別の物体の進入に備えるため新たな検出フィルタを設置し、さらに追跡中の物体を重複して検出しないようにしているためこのように複数物体が同時に進入してきた場合でも検出することができた。

しかし、図 10 のように複数の物体が近接した状態で進入してきた場合は一つの物体として検出された。これは複数物体がオクルージョンを起こし、つながった一つの大きな物体として検出されたためであると考えられる。この結果から図 9 のように、複数物体がオクルージョンを起こさないようある程度距離が離れていれば、検出が成功するということがわかる。

バイク



(a) 入力画像



(b) 検出結果画像



(c) 入力拡大画像



(d) 検出結果拡大画像

図 8 カメラから遠く離れた場所の進入物体検出結果



図 9 複数物体の同時進入検出結果



図 10 物体が近接した状態で進入してきた場合の検出結果

進入物体検出実験の結果, 表 4 に示すように進入物体の総数は 60 体でそのうち 46 体が検出に成功し, 検出成功率は 76% となった。検出が失敗となった例には, 図 11 のように物体が進入検出領域をかすめるようにして進入してきて検出されないものがあった。これは進入検出領域をかすめるようにして進入してきた結果, 0 より大きな尤度を持つパーティクルの数が追跡開始閾値 β を超えなかったためだと考えられる。

表 4 検出成功率

物体数	60
検出成功物体数	46
検出成功率 (%)	76



図 11 検出失敗例

6. まとめ

本稿では、パーティクルフィルタを用いた3次元空間上での物体追跡における進入物体検出について検討を行った。まずキャリブレーション済みの固定単眼カメラを用いて撮影した動画像を対象に物体追跡を行った。この結果、カメラから遠く離れた画像に小さく写る物体でも検出および追跡が可能であることがわかった。

次に、複数物体の同時進入検出実験を行った。進入物体の検出は物体が進入してくると考えられる場所に進入物体検出領域を設け、そこに検出フィルタを設置することによって物体の検出を行った。検出フィルタが領域内で物体の進入を検出したとき、別の物体の進入に備えるため新たな検出フィルタを設置するようにした。この時、検出領域内に存在する追跡中物体の領域を定めることで、新たな検出フィルタがすでに追跡中の物体を重複して検出しないよう工夫した。これにより物体同士が相互にオクルージョンを起こさないようある程度距離が離れていれば、複数物体の同時進入の検出が可能ということがわかった。

今後の展望として、より多数の物体が行き交う環境で実験を行い、物体の大きさや進入速度などに対する検出精度や追跡精度を検討していく必要がある。

参考文献

- [1] 村北卓也, 石黒浩, “Sampled-Volume-based Data Association Filter による遮へいロバストな実時間複数人物追跡”, 電子情報通信学会論文誌 D, Vol. J90-D, No.10, pp. 2868-2883, 2007
- [2] 松元郁佑, 加藤丈和, 和田俊和, 上田博唯, “複数カメラを用いた CONDENSATION による複数人物頭部の実時間検出・追跡”, 信学技報 PRMU2005-277, Vol.105, No.674, pp. 121-128, 2006
- [3] M.Isard and A.Blake, “CONDENSATION – Conditional Density Propagation for Visual Tracking,” International Journal of Computer Vision, Volume 29, Number 1, pp.5-28,1998.
- [4] 奈良先端科学技術大学院大学 OpenCV プログラミングブック制作チーム, “OpenCV プログラミングブック”, 毎日コミュニケーションズ, 2007

