

# Tracklet 特徴量と Mean-Shift クラスタリングによる 歩行者流量推定方式の提案

米司健一<sup>†1</sup> 吉永智明<sup>†1</sup> 松原大輔<sup>†1</sup> 額賀信尾<sup>†1</sup>

監視カメラの映像から歩行者の流量を推定する技術に関し、低演算量でロバストな推定結果が得られる方式を提案する。従来の流量推定方式では、画像中から人の流量をロバストに検出するために、HOG 等の演算量の多い特徴量を用いる必要があった。これに対し提案方式では、演算量の少ない Tracklet 特徴量を、Mean-Shift クラスタリングを用いて歩行者毎の動き情報に変換して解析することで、低演算量かつロバストな流量推定が可能となった。

## Pedestrian Flow Estimation using Tracklet and Mean-Shift Clustering

KENICHI YONEJI<sup>†1</sup> TOMOAKI YOSHINAGA<sup>†1</sup>  
DAISUKE MATSUBARA<sup>†1</sup> NOBUO NUKAGA<sup>†1</sup>

We developed a method of pedestrian flow estimation for surveillance camera system. Previous methods like HOG descriptor need high computational cost for accurate estimation. In this paper, we present a robust method for pedestrian flow estimation of which computational cost is low. Our method uses Mean-Shift clustering to divide tracklets into each pedestrian's information, and it achieves both accurate estimation and low computational cost.

### 1. はじめに

近年、施設の安全性やサービス向上に向けて、監視カメラに映る対象の移動経路や数を推定する技術への期待が高まっている。例えば大型商業施設では、イベント時における警備員の配置計画や、店舗の効果的なレイアウトの策定のために、施設利用者の数や移動経路等の情報の活用が期待されている。

一方で監視カメラは、凶悪犯罪の増加やセキュリティ意識の向上に伴い、様々な場所への設置が進んでいる。設置された監視カメラは防犯用途を想定しており、事件発生時の犯罪捜査などに主に利用される。そのため、事件発生時以外はあまり活用されておらず、事件発生時以外の有効活用が望まれている。

また近年では、ネットワークを通じた IP 網によるシステムの構築により、多数の監視カメラ映像をサーバ上に集約し、リアルタイムに解析を行う事が可能となりつつある。そこで本稿では、防犯用途として設置された監視カメラ映像をサーバ上に集約して解析し、人物の流量をリアルタイムに計測することを目的とした画像処理方法を提案する。これにより、新たにカメラを追加することなく、人物の流量情報をリアルタイムに得る事が可能となり、防犯用途に設置された監視カメラの活用範囲を広げる事ができる。

従来手法では、人物の流量を計測するために、図 1 (a) の様な、遠方から大きい俯角で撮影された映像を用いる場合が多い。



(a)遠方かつ俯角大の映像 (b)低位置かつ俯角小の映像

図 1 監視カメラで撮影された映像の種類

この映像は人物同士の重なりが少なく、流量を推定する為の画像処理に適している。一方で防犯用途として設置された監視カメラは、人物の顔や全身を映す目的で設置されている場合が殆どであり、低い位置から浅い俯角で設置されている。そのため、図 1(b)の様に、撮影された映像は人物同士の重なりが大きく、流量の推定が困難である。また、多数のカメラ映像をサーバに集約して解析処理を行う場合、解析処理は低演算量である方が望ましい。そこで本稿では、Tracklet 特徴量と Mean-Shift クラスタリングを用いることで、人物同士の重なりが大きい場合においても、ロバストかつ低演算量で流量を推定可能な方法を提案する。

本研究では、防犯用途として設置された監視カメラで撮られる映像として、高さ 2.0[m]~5.0[m]程度の位置から撮られた映像を想定する。また流量の計測対象を歩行者に限定する。さらに、カメラの位置や向きは固定とする。

<sup>†1</sup>(株)日立製作所 中央研究所  
Hitachi Ltd., Central Research Laboratory

## 2. 従来手法

人物の流量推定は、映像中から歩行者の人数と移動方向を検出する事により行う。

歩行者の人数と移動方向を検出する手法として、人物検出処理により映像中から歩行者の位置を検出して人数を数え、検出した全ての歩行者を時間方向に追跡処理することで移動方向を得る手法が多数研究されている[1][2][3][4]。人物の検出手法としては、背景差分を用いる手法[5][6]や、HOG 特徴量等により人物検出を行う手法[7][8]など、様々な手法が提案されている。しかし、背景差分を用いる手法は人同士の重なりが弱く、HOG 特徴量を用いる手法は計算速度や検出対象の解像度が課題となる。追跡手法としては、KLT トラッカ等の勾配法を用いた追跡手法[9][10]や、パーティクルフィルタによる追跡手法[11]など多数の方法が提案されている。また Kratz らは混雑状況下における人流の時空間モデルを用いることにより非常に混雑した環境での特定対象の追跡を実現している[12]。しかし一般に追跡処理は、追跡精度や追跡対象数の増加に従い演算コストが増えるといった課題がある。これに対して Benfold らは、HOG 特徴量の計算を GPU(Graphics Processing Unit)を用いて高速化し、KLT トラッカと MCMCDA(Markov-Chain Monte-Carlo Data Association)を用いて追跡を行うことで、重なりが強い流量推定を高速に行う手法を提案している[13]。しかしこの手法では、GPU を用いる事を前提としており、演算速度は速いがシステムにかかる負荷が大きい。

これに対して、映像中から動き情報を抽出し、抽出した動き情報を解析することで、流量を得る手法が提案されている。例えば村井らは、ST-Patch を用いて動き情報を抽出し、それらを統合することで人物の位置や移動方向を検出している[14]。ST-Patch は、複雑な動きの評価を行う事が可能な特徴量であり、歩行者の移動方向の推定に有効である。しかし、ある一定領域、一定時間の輝度値の変化や勾配を9次元のベクトルに落とし込むため、得られた9次元ベクトルからは対象の細かい移動量の再現が困難である。これに対し近年では、Tracklet と呼ばれる軌跡情報を用いて流量を推定する手法が多数提案されている[15][16]。Tracklet 特徴量は、短時間(画像数フレーム分の時間)の追跡処理を画像中の多数の点に対して行うことで得られる軌跡情報の集合であり、各追跡点の追跡には正確な処理を必要とせず演算コストが低い。また多数の点に対する追跡処理を行うため隠れに強く、ロバストに流量推定できるといった特徴がある。そこで本研究では、防犯用途として設置された監視カメラの様に、対象の隠れが多い環境においては Tracklet 特徴量が有効であると考え、Tracklet 特徴量を用いた流量推定手法を採用する。

Tracklet 特徴量を用いた流量推定手法の従来手法である Zhou らの方式[16]では、各 Tracklet 特徴量が歩行者毎に分



図 2 Tracklet の検出結果の例

離して得られていることを前提としている。そのため、防犯用途として設置された監視カメラの様に、映像中の歩行者同士の重なりが大きく、Tracklet 特徴量が歩行者毎に分離されていない状況には適用できないといった課題があった。そこで報告者は、Mean-Shift クラスタリング[17]を用いて各 Tracklet 特徴量を歩行者毎に分離し、流量推定を行う手法を提案する。

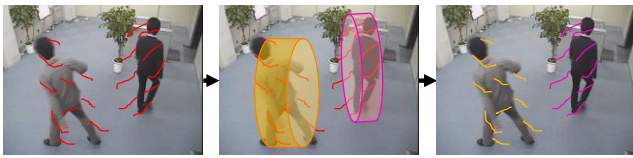
## 3. Mean-Shift を用いた Tracklet 特徴量クラスタリングによる流量推定

本章では提案する流量推定手法について述べる。はじめに Tracklet 特徴量について概要を説明した後、Mean-Shift クラスタリングを用いて Tracklet 特徴量を分離する方法について述べる。

### 3.1 Tracklet 特徴量の概要

Tracklet 特徴量は映像の時空間変化を記述する特徴量であり、人の動作識別などに用いられる [18]。Tracklet 特徴量は、画像中のある特徴点の軌跡ベクトルを抽出し、その軌跡ベクトル周辺の画像変化と、移動方向を記述した特徴量である。本研究では、Tracklet 特徴量を歩行者毎に分離するために、Tracklet 特徴量の軌跡ベクトルを用いて分離する方式を提案する。この軌跡ベクトルを以下では Tracklet と呼ぶ。

Tracklet の算出には、KLT トラッカを用いる。以下その手順について説明する。まず、追跡対象となる特徴点を Tomasi らの方式[9]により検出する。次に新しい画像フレームが入力された際に、新たに追跡しやすい特徴点を選択し、追跡を行う特徴点として追加する。同時に前フレームにおいて選択した特徴点を、Lucas-Kanade 法[10]により追跡処理する。この処理を新しい画像フレームが入力される度に繰り返す。各特徴点の追跡処理は、一定時間経過するか、隠れなどによって追跡の信頼度が低下するまで行う。以上の様に特徴点を次々に追加しながら追跡することで、Tracklet を大量に集めることができる。さらに集まった Tracklet のうち、動きが全くない軌跡や、人の動きとして



(a)Tracklet 抽出 (b)軌跡の判定 (c)クラスタリング  
 図 3 Tracklet のクラスタリングの流れ

は大きすぎる軌跡などを除去することで、歩行者の動きに対応する Tracklet を得ることができる。

以上のようにして検出した Tracklet の例を図 2 に示す。図 2 中の赤い線が Tracklet を表しており、それぞれ長さの違う軌跡を多数集める事ができる。次の 3.2 節では以上の方法により検出した Tracklet を歩行者毎に分離する方法について説明する。

### 3.2 Mean-Shift による Tracklet 特徴量のクラスタリング

Tracklet はある特徴点の時間方向の移動軌跡を表したベクトルである。ここでは、ある時刻  $t$  において、追跡処理を継続している Tracklet を歩行者毎にクラスタリングする方法を提案する。

提案手法では、Tracklet が以下 2 つの特徴を有する事を利用し、Tracklet のクラスタリングを行う。

- (特徴 1) 各特徴点が歩行者の形状に合わせて分布する。
- (特徴 2) 各歩行者から得られる Tracklet はそれぞれ似た軌跡となる。

以上の特徴を用いて、図 3 の様に、ある一定時間の移動軌跡が一定範囲内に収まる Tracklet を同一クラスタとして分類することで、歩行者毎に Tracklet をクラスタリングする。

Tracklet をクラスタリング処理する際には、歩行者の人数が分からない状態であるため、分類するクラスタ数が不明である。そのため、k-means 法[19]の様に、クラスタ数をあらかじめ指定する必要があるクラスタリング方法は適用できない。そこで提案手法では、クラスタ数を自動で決定するクラスタリング方法を用いる。クラスタ数を自動で決定するクラスタリング方法として、Affinity propagation[20]や、Coherent Filtering を用いる方法[21]などがある。これらの方法は精度の高いクラスタリングを実現可能であるが、演算コストが高い。そのため提案手法では、演算量を考慮して、単純な演算によりクラスタリング可能な Mean-Shift クラスタリング[17]を用いる。

各 Tracklet はそれぞれ追跡開始時間が違うため、ベクトルの大きさが異なる。そこで提案手法では一定フレーム数でベクトルを区切ることで、ベクトルの長さを一定に揃えてクラスタリング処理を行う。時刻  $t$  における  $i$  番目の Tracklet の特徴点の座標ベクトルを  $\mathbf{p}_i(t)$  とし、クラスタリングに用いるフレーム数を  $n$  フレームとすると、各 Tracklet から次の式(1)で与えられる特徴点の集合  $\mathbf{X}_i$  を取り出してクラスタリングを行う。

$$\mathbf{X}_i = \{\mathbf{p}_i(t), \mathbf{p}_i(t-1), \mathbf{p}_i(t-2), \dots, \mathbf{p}_i(t-n-1)\} \quad (1)$$

以下ではまず、説明を簡単にするために、2次元空間での Tracklet のクラスタリング方法について説明する。次に、2次元空間でのクラスタリング方法を3次元空間でのクラスタリング方法に拡張する。

#### 3.2.1 2次元空間での Mean-Shift クラスタリング

ここでは、式(1)においてクラスタリングに用いるフレーム数を  $n=1$  とした場合について説明する。 $n=1$  の場合、クラスタリング処理に用いる特徴点のベクトルは、時刻  $t$  における特徴点のみとなり、2次元座標上における点群のクラスタリング問題として考える事ができる。すなわち、Tracklet の数を  $N$  個、各特徴点のベクトルを  $\mathbf{x}_i$  とすると、 $n=1$  の場合の Tracklet のクラスタリングは、 $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N \in \mathbf{R}^2$  の点群のクラスタリング問題となる。また各点  $\mathbf{x}_i$  は式(1)の  $\mathbf{p}_i(t)$  と一致する。

Mean-Shift クラスタリングは、次の式(2)の更新式に従い Mean-Shift 用サンプル点  $\mathbf{y}_i$  の更新を行う。

$$\mathbf{y}_i^{j+1} \leftarrow \frac{\sum_{i=1}^N K_{\mathbf{H}_i}(\mathbf{y}_i^j - \mathbf{x}_i) \mathbf{x}_i}{\sum_{i=1}^N K_{\mathbf{H}_i}(\mathbf{y}_i^j - \mathbf{x}_i)} \quad (2)$$

Mean-Shift 用サンプル点  $\mathbf{y}_i$  は各点  $\mathbf{x}_i$  に対する Mean-Shift 処理の更新結果であり、初期値  $\mathbf{y}_i^0$  は  $\mathbf{y}_i^0 = \mathbf{x}_i$  で与えられる。式(2)を用いて全ての Mean-Shift 用サンプル点  $\mathbf{y}_i$  の収束点を得た後に、収束点が近くにある点同士を NN(Nearest Neighbor)法を用いて同一クラスタとして分類する。

ここで、 $K_{\mathbf{H}_i}(\mathbf{x})$  は Mean-Shift 用カーネル関数であり、次の式(3)で与えられる。

$$K_{\mathbf{H}_i}(\mathbf{x}) = K(\mathbf{H}_i \mathbf{x}) \quad (3)$$

$\mathbf{H}_i \in \mathbf{R}^{2 \times 2}$  はバンド幅行列で、 $\mathbf{H}_i = \text{diag}(1/h_{i,1}, 1/h_{i,2})$  である。また  $K(\mathbf{x})$  は Mean-Shift に用いるカーネル関数である。

Mean-Shift クラスタリングではバンド幅行列  $\mathbf{H}$  とカーネル関数  $K_{\mathbf{H}_i}(\mathbf{x})$  を適切に設定する必要がある。ここでは、3.2 節で説明した特徴 1 に着目することで、バンド幅行列  $\mathbf{H}$  とカーネル関数  $K_{\mathbf{H}_i}(\mathbf{x})$  を決定する。

特徴 1 から、各特徴点  $\mathbf{x}_i$  は図 4 の様に、歩行者が存在する領域に均等に分布していると仮定する事ができる。この仮定から、各特徴点は、人の形状の重心位置を中心とし、人の形状に合わせた形で分布していると言える。そのため、バンド幅を人の形状に合うように適切に設定し Rectangular カーネルを用いて重心位置を計算すれば、特徴点を人毎にクラスタリングする事ができる。しかし、図 4 に示す様に、画像上での人の大きさが対象の奥行によって変わるため、位置ごとに適切なバンド幅が変動する点が課題となる。そこで提案手法では、Mean-Shift 用サンプル点  $\mathbf{y}_i$  から適切なバンド幅行列を計算する関数  $B(\mathbf{y})$  を用いて、サンプル点ごとにバンド幅を変えながら Mean-Shift を行うことで、こ

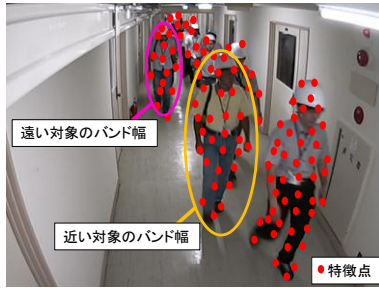


図 4 特徴点の分布例と適切なバンド幅

の課題を解決した。以上を踏まえて、提案手法では次の式(4)のカーネル関数を用いて Mean-Shift を行う。

$$K_{\mathbf{H}}(\mathbf{x}) = K(\mathbf{H}\mathbf{x}) = K(B(\mathbf{y}_i)\mathbf{x}) = \begin{cases} 1 & \text{if } (|B(\mathbf{y}_i)\mathbf{x}| < 1), \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

バンド幅行列算出用関数  $B(\mathbf{y})$  については、次の 3.2.2 にて詳しく説明する。

### 3.2.2 バンド幅行列算出用関数 $B(\mathbf{y})$ の設定

関数  $B(\mathbf{y})$  は、Mean-Shift 用サンプル点  $\mathbf{y}$  を入力とし、バンド幅行列  $\mathbf{H}$  を出力する関数である。以下ではその設定方法について説明する。

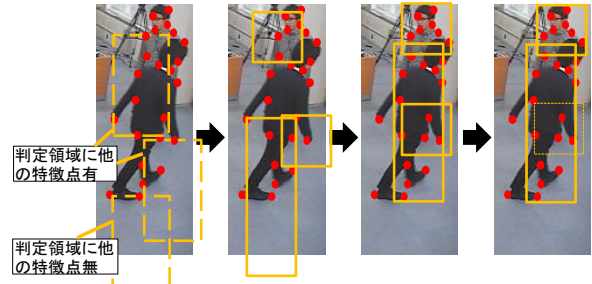
まず、 $\mathbf{y}$  が人の中心位置に存在するとして、地面から高さ 80[cm](日本人の成人平均身長約半分の高さ)の位置の点であると仮定する。次に、地面の位置情報と、カメラの外部パラメータ及び内部パラメータを用いて、カメラ座標上の点  $\mathbf{y}$  の位置を実空間座標に変換する。次に実空間座標上で人を半径 40[cm]高さ 165[cm]の円柱と仮定して、その円柱をカメラ座標に投影する。次に投影された円柱に外接する長方形を求める。最後に求めた長方形の幅を  $h_1$ 、高さを  $h_2$  と設定する。以上により、Mean-Shift 用サンプル点  $\mathbf{y}$  の奥行に従いバンド幅を決定する。

以上の様にバンド幅を決定しながら Mean-Shift を行うと、最終的な収束点では図 4 に示す様な人の形状に合わせたバンド幅となる。そのため、人の形状の重心位置にサンプル点  $\mathbf{y}$  を収束させる事が可能となり、歩行者毎に Tracklet をクラスタリングする事ができる。

以上の方法において、バンド幅の算出を全ての特徴点に対して毎時間行くと演算コストが高くなる。そこで、予め画像中の全画素位置に対して  $B(\mathbf{y})$  を計算し、全ての画素位置に対する  $\mathbf{H}$  の対応表を作っておくことで、演算コストを低減する事ができる。

### 3.2.3 オクルージョンへの対応

3.2.2 では人の形状に合うようにバンド幅を決定する方法について述べた。しかし、オクルージョンがある場合には全身の形状に合うバンド幅を用いると誤差が大きくなる。そこで、オクルージョンを考慮して、バンド幅の縦幅  $h_2$  を



(a)特徴点存在判定 (b)バンド幅設定 (c)Mean-Shift計算 (d)重なり統合

図 5 オクルージョンを考慮した Mean-Shift

変える必要がある。

図 5 はオクルージョンを考慮してバンド幅の縦幅  $h_2$  を決定し、Mean-Shift を行う処理の概要図である。まず、各特徴点に対して、特徴点から下の一定の範囲を判定領域とし、この判定領域に他の特徴点が存在するかどうかを判定する(図 5 (a))。特徴点が存在しない場合には、オクルージョンが無い領域の特徴点である可能性が高い為、3.2.2 の方法で縦幅  $h_2$  を設定する。一方で、判定領域に特徴点が存在する場合には、オクルージョンがある可能性があるため、3.2.2 で用いた円柱のうち、頭部に相当する部分のみを用いて縦幅  $h_2$  を設定する(図 5 (b))。以上の様に決定したバンド幅を用いて、Mean-Shift クラスタリングを行う(図 5 (c))。最後に収束した結果のうち、バンド幅の重なりが大きいサンプル点同士を統合する(図 5 (d))。以上により、オクルージョンを考慮した Mean-Shift クラスタリングを実現できる。

### 3.2.4 3次元空間における Mean-Shift クラスタリング

ここでは、3.2.1 において説明した 2次元座標上におけるクラスタリングを 3次元空間におけるクラスタリングに拡張する方法について説明する。

まず、各点のベクトル  $\mathbf{x}_i$  は次の式(5)となる。

$$\mathbf{x}_i = (\mathbf{p}_i(t)^T, \mathbf{p}_i(t-1)^T, \mathbf{p}_i(t-2)^T, \dots, \mathbf{p}_i(t-n-1)^T)^T \quad (5)$$

また、Mean-Shift 用サンプル点  $\mathbf{y}_i$  を次の式(6)とする。

$$\mathbf{y}_i = (\mathbf{m}_i(t)^T, \mathbf{m}_i(t-1)^T, \mathbf{m}_i(t-2)^T, \dots, \mathbf{m}_i(t-n-1)^T)^T \quad (6)$$

Mean-Shift 用サンプル点の更新処理は式(2)を用いる。

3次元空間における Mean-Shift では、3.2節で述べた特徴 2 を考慮してバンド幅を時間方向に拡張し、図 6 の様な筒状の範囲を囲むバンド幅を用いる。各時間のバンド幅の算出には、3.2.2 において説明した関数  $B(\mathbf{y})$  を用いる。またカーネル関数も同様に時間方向に拡張し、次の式(7)を用いる。

$$K_{\mathbf{H}}(\mathbf{x}) = \begin{cases} 1 & \text{if } (\forall x \in \mathbf{X}_i, |B(\mathbf{m}_i(t_x))\mathbf{p}(t_x)| < 1), \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

ここで  $t_x$  は各  $x$  における時刻  $t$  を意味する。

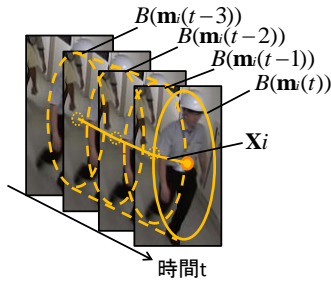


図 6 3次元空間 Mean-Shift クラスタリング用バンド幅

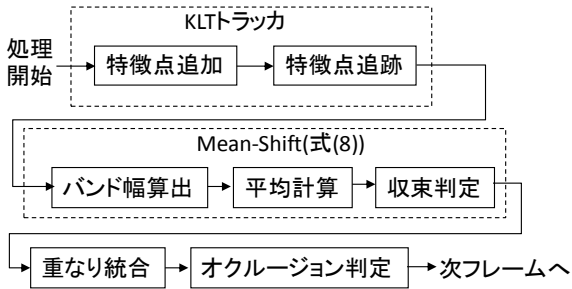


図 7 提案手法の処理フロー

以上により 3次元空間における Tracklet のクラスタリングが可能となる。3次元空間において Tracklet をクラスタリングすることで、対象の移動軌跡を考慮したクラスタリングを行う事ができる。そのため、2次元空間でのクラスタリングと比べ、人同士の重なりに対してよりロバストなクラスタリングが可能となる。

#### 4. Mean-Shift クラスタリング効率化

Mean-Shift クラスタリングは、全ての Mean-Shift 用サンプル点が収束するまで処理を繰り返すため、計算コストが高い。そこで提案手法では、Mean-Shift の計算を時間方向に引き継ぎながら更新することで、演算を効率化した。以下ではその効率化方法について説明する。

ここではまず、説明を簡単にするために 2次元空間での Mean-Shift クラスタリングの演算効率化手法について説明する。提案する効率化手法では、以下の更新式(8)を用いて、Mean-Shift の計算を行う。

$$y_i^t \leftarrow \frac{\sum_{i=1}^N K_{\text{Hi}}(y_i^{t-1} - \mathbf{p}_i(t-1)) \mathbf{p}_i(t)}{\sum_{i=1}^N K_{\text{Hi}}(y_i^{t-1} - \mathbf{p}_i(t-1))} \quad (8)$$

またカーネル関数は式(4)を用いる。式(8)は 1フレーム前の Mean-Shift 用サンプル点  $y_i^{t-1}$  の計算結果と、1フレーム前の特徴点の位置  $\mathbf{p}_i(t-1)$  を基にカーネル関数を計算し、得られた値を現時刻の特徴点の位置  $\mathbf{p}_i(t)$  の重みとしてかけ合わせて平均を計算することで、現時刻の Mean-Shift 用サンプル点  $y_i^t$  を得る。次に  $y_i^t$  が近くにある点同士を NN 法等によりクラスタリングする。以上を繰り返すことで、時間方



図 8 評価用映像



図 9 評価対象領域

表 1 映像撮影条件

シーン名	設置高さ [m]	俯角[°]	画素数 [pix]	フレームレート [fps]
シーン 1	2.2	20	640x480	10
シーン 2	1.9	11	640x480	10

向に結果を引き継ぎながら収束計算を行う事が可能となる。これにより 1フレーム当たりの演算回数を削減できる。

以上の提案手法による演算効率化を 3次元空間における Mean-Shift クラスタリングに適用する場合にも、同様の方法によって式(7)を式(8)の形式に拡張することで、時間方向に結果を引き継ぎながら収束計算を行う事が可能となる。

以上 3章及び 4章にて述べた提案手法により、少ない演算量でロバストに歩行者の流量を推定する事が可能となる。以上で説明した提案手法における 1フレーム分の処理フローを図 7 にまとめる。

#### 5. 評価実験

提案手法を用いて、歩行者の流量の推定実験を行い、推定精度と演算コストを評価する。精度の評価実験では、歩行者の流量として人数の推定結果の正解率を評価する。また、演算コストの評価として、提案手法の処理時間の計測結果を示す。

##### 5.1 実験方法

評価実験では、Tracklet のクラスタリング用のフレーム数  $n$  を 3 として実験を行った。また精度の評価には 4章において高速化した推定方式を用いた。

実験には図 8 に示す 2つのシーンを用いた。どちらも固定の監視カメラで撮影された映像である。各カメラの撮影

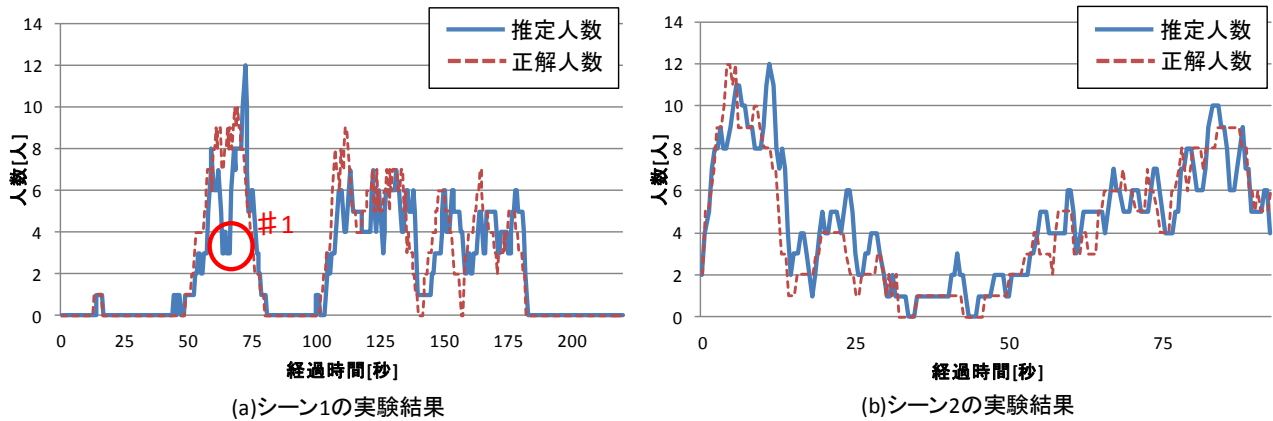


図 10 実験結果

表 2 一定時間毎の人数推定精度

シーン名	時間[秒]	推定人数 [人]	正解人数 [人]	推定精度 [%]
シーン 1	0~40	5	6	83
	40~80	294	343	85
	80~120	155	198	78
	120~160	334	337	99
	160~200	183	183	100
	全体	963	1057	91
シーン 2	0~40	370	320	84
	40~80	320	295	92
	80~120	176	187	94
	全体	865	801	92

表 3 処理時間の計測結果

シーン名	平均処理時間[msec]	
	演算効率化前(3章)	演算効率化後(4章)
シーン 1	62.65	28.2
シーン 2	87.69	30.8

線は推定人数を表す。シーン 1(図 10 (a))及びシーン 2(図 10 (b))のどちらの結果においても、推定人数と正解人数の推移が大よそ一致していることが分かる。図 10 (a)の丸印(#1)では推定人数が正解人数から離れているが、このシーンは歩行者が一斉に急停止するシーンであり、背景と歩行者の切り分けが正しくできず精度が落ちていることが原因であると考えられる。

表 2 は、一定時間毎の人数推定精度を評価した結果である。この結果から、各時間において約 80[%]の精度で人数を推定できている事が分かる。またシーン全体の合計で評価すると、両シーンともに 90[%]以上の推定精度であった。

次に 1 フレームの処理にかかる計算時間の測定結果を表 3 に示す。この結果から、高速化前は 60~80[msec]の処理時間であったのに対して、高速化後はどちらも約 30[msec]の処理時間であった。

図 11 に各シーンの歩行者検出結果の一例を示す。図 11 中の赤い点は各フレームにおいて検出された Tracklet の座標位置である。また、各 Mean-Shift 用サンプル点の収束点に対するバンド幅を矩形で示す。この結果から、クラスタリング結果がおよそ人物の重心位置に収束している事が分かる。

以上の実験から、提案方式では約 90[%]の精度で歩行者の流量を推定でき、1 フレームあたり約 30[msec]で処理可能であることが確認できた。

条件を表 1 に示す。これらのシーンに対して、図 9 に示す領域を対象領域とし、提案手法を用いて歩行者の流量を推定し、実際的人数(正解人数)の推移と提案手法による推定結果(推定人数)を比較した。この対象領域内では地面の高さは変化しないものとし、処理を行った。また推定人数と正解人数の比較は、500[msec]毎に行った。

提案手法の処理時間の計測には、一般的なデスクトップ PC(CPU: Intel(R) Core(TM) i7 2.67[GHz], メモリ: 3.0[GB])を使用した。また各シーンの全フレーム分の処理時間から、1 フレームあたりの平均処理時間を測定した。この際、提案手法の高速化前(3章)と高速化後(4章)の処理時間を比較した。

## 5.2 実験結果

4 章にて説明した高速化後の提案手法を用いて流量推定を行い、精度を評価した結果を図 10 及び表 2 に示す。図 10 のグラフは時間の経過(横軸)に対する人数(縦軸)の変化をプロットしたグラフである。点線が正解人数を表し、実

## 6. おわりに

本稿では、防犯用途として設置された監視カメラの映像から、歩行者の流量を推定することを目的とし、Tracklet

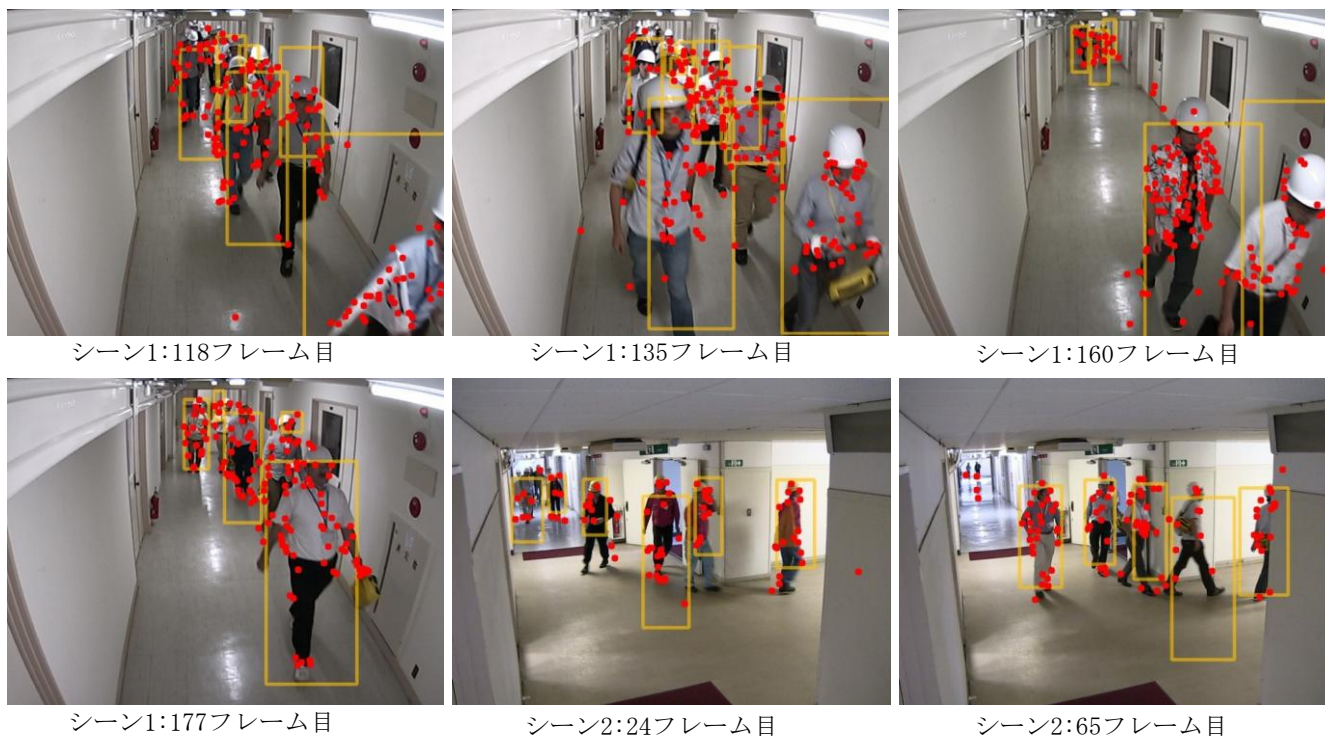


図 11 歩行者の検出結果

特徴量と Mean-Shift クラスタリングを組み合わせた流量推定手法を提案した。提案手法では、演算量の少ない Tracklet 特徴量を Mean-Shift クラスタリングを用いて歩行者毎にクラスタリングすることで、歩行者の流量を推定する。この際、Mean-Shift クラスタリングに用いるバンド幅を Mean-Shift ベクトルの位置及びオクルージョン状態により適切に設定しながら更新を行うことで、対象の奥行にばらつきがある場合や、オクルージョンがある場合においてもロバストに流量を推定する事が可能となった。また Mean-Shift クラスタリングの更新を時間方向に引き継ぎながら処理することで、演算効率の高い処理を実現した。

今後は歩行者が急停止するなど、動きの少ないシーンに対する精度向上検討を行う。また、今回は地面が平面である事を仮定したが、地面が複雑な形状をしている場合についての検討及び評価を今後行う予定である。

### 参考文献

- 1) Hyodo, Y. et al.: Pedestrian Tracking across Panning Camera Network, International Journal of ITS Research, Vol.7 (2009).
- 2) Heili, A. et al.: Detection-based Multihuman Tracking Using a CRF Model, ICCV Workshop (2011).
- 3) Gong, H. et al.: Multi-hypothesis Motion Planning for Visual Object Tracking, ICCV (2011).
- 4) Chen, C. and Odobez, J.: We are not Contortionists: Coupled Adaptive Learning for Head and Body Orientation Estimation in Surveillance Video, CVPR (2012).
- 5) Stauffer, C. et al.: Learning Patterns of Activity Using Real-Time Tracking, Trans. PAMI, Vol.22, No.8 (2000).
- 6) Hofmann, M. et al.: Background Segmentation with Feedback: The

- Pixel-Based Adaptive Segmenter, CVPRW (2012).
- 7) Dalal, N. and Triggs, B.: Histograms of Oriented Gradients for Human Detection, CVPR (2005).
- 8) Prisacariu, A. V. and Reid, I.: fastHOG - a real-time GPU implementation of HOG, Technical Report 2310/09, University of Oxford (2009).
- 9) Shi, J. and Tomasi, C.: Good Features to Track, CVPR (1994).
- 10) Lucas, B. D. and Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision, IJCAI (1981).
- 11) Isard, M. and Blake, A.: CONDENSATION—Conditional Density Propagation for Visual Tracking, International Journal of Computer Vision, Vol.29, No.1 (1998).
- 12) Kratz, L. and Nishino, K.: Tracking Pedestrians Using Local Spatio-Temporal Motion Patterns in Extremely Crowded Scenes, Trans. PAMI, Vol.34, No.5 (2012).
- 13) Benfold, B. and Reid, I.: Stable Multi-Target Tracking in Real-Time Surveillance Video, CVPR (2011).
- 14) 村井泰裕, 藤吉弘亘, 金出武雄: Space-Time Patch を用いた物体の移動方向識別とセグメンテーション, MIRU (2007).
- 15) Ge, W. and Collins, R. T.: Multi-target Data Association by Tracklets with Unsupervised Parameter Estimation, BMVC (2008).
- 16) Zhou, B. et al.: Random Field Topic Model for Semantic Region Analysis in Crowded Scene from Tracklets, CVPR (2011).
- 17) Comaniciu, D. and Meer, P.: Mean Shift: A Robust Approach Toward Feature Space Analysis, Trans. PAMI, Vol.24, No.5 (2002).
- 18) Raptis, M. and Soatto, S.: Tracklet Descriptors for Action Modeling and Video Analysis, ECCV (2010).
- 19) Kanungo, T. et al.: An Efficient k-Means Clustering Algorithm: Analysis and Implementation, Trans. PAMI, Vol.24, No.7 (2002).
- 20) Frey, B. J. and Dueck, D.: Clustering by passing messages between data points, Science, Vol. 315, pp. 972–976 (2007).
- 21) Zhou, B. et al.: Coherent Filtering: Detecting Coherent Motions from Crowd Clutters, ECCV (2012).