

## 異種センサ統合に基づく 頑健で高速な動線検出方式

森口有紀江<sup>†</sup> 小西勇介<sup>†</sup> 中尾敏康<sup>†</sup>

本稿では、リアルタイム処理を想定したセンサデータの欠落に頑健な動線検出処理方式について述べる。人物動線検出方式の一つに、人物の移動軌跡と ID を異種センサから取得し、軌跡と ID の組合せを人物動線仮説として生成し、各仮説の尤度を軌跡および ID の検出結果から定量化し、最尤推定することにより軌跡と ID を統合する手法がある。しかし、この手法には、センサデータの欠落が頻発すると動線検出精度が低下する課題がある。また、リアルタイム処理を想定する場合、現時刻から過去一定時間分のセンサデータに対しこれらの処理を都度行うと冗長な演算処理が生じるため、一連の動線検出処理が完了するまでに時間がかかる。そこで我々は、現時刻から過去一定時間内に検出した軌跡および ID を用いて対応付けに有効な情報をより多く活用することにより、センサデータの欠落に頑健な動線検出方式を実現する。更に、提案方式に過去に生成した仮説や尤度等の中間処理結果を記憶する仕組みを設け、新たに入力されたセンサデータに影響する箇所のみを更新することで冗長な演算を解消するという高速化を行った。

### A Robust and Fast Human Trajectory Detection Method based on Sensor Fusion

Yukie Moriguchi, Yusuke Konishi and Toshiyasu Nakao

This paper describes a robust and real-time human trajectory detection method against sensor data discontinuity. One of the human trajectory detection methods detects human trajectories and IDs by heterogeneous sensor, creates combination of trajectory and ID as hypothesis of identified trajectory, quantify the likelihood of identified trajectory hypothesis, and assign ID to trajectory by maximum likelihood estimation method. However, this human trajectory detection method has a problem to reduce identified trajectory detection accuracy if lack of sensor data frequently occurs. Moreover, this method takes a long time to complete the human trajectory detection process because redundant operation arises each time when executed these processing using sensor data history. We use sensor data history for robust human trajectory detection, and add storage for hypothesis and likelihood created in every process stage for fast human trajectory detection.

### 1. はじめに

近年、セキュリティやマーケティングなどの分野で「ID 認証された個人ごとの」「途切れない」人物動線を検出する技術が必要とされている。特に、セキュリティ分野への適用を実現するためには、最新の人物位置をリアルタイムに取得できることが望まれる。例えば、オフィスのセキュリティ分野では、ビルや部屋の出入口に設置された入室管理ゲートを越えた後も人物を連続して追跡し、不審行動を検出した場合は速やかにアラートをあげるといったサービスが必要になる。

このような人物動線を検出する技術として、異種センサから取得した人物の移動軌跡と ID を統合する手法が提案されている<sup>1)</sup>。単一種のセンサだけを使用する手法では、使用するセンサ種ごとに追跡精度やコスト、分解能の面で利点と欠点がある。そのため、異種センサを統合してセンサ種ごとの欠点を補い利点を生かすことにより、単一種のセンサのみを用いる手法に比べて動線検出の性能を改善できると期待されている。

しかし一方で、異種センサから取得した人物の移動軌跡と ID を統合する手法には、センサデータの欠落が頻発すると、軌跡と ID を正しく統合することが困難になる課題と、軌跡と ID の組合せ数が爆発的に増加するためリアルタイム処理が困難になる課題がある。

そこで我々は、センサデータの欠落が頻発する場合において、センサデータの履歴情報を活用することによって人物動線を頑健に検出する研究<sup>2)</sup>、更に過去に生成したセンサデータの組合せの一部を逐次更新し人物動線を高速に検出する技術の研究を進めている。

本稿では、頑健化および高速化を実現する動線検出方式を提案し、カメラとアクティブ RFID を用いた人物動線検出システムに適用し頑健化に関する評価実験を行った結果と、シミュレーションデータを用い高速化に関する評価実験を行った結果について報告する。

### 2. 従来方式の特徴と課題

はじめに、「ID 認証された個人ごとの」人物動線を検出するために、異種センサを用いて移動軌跡と人物 ID を統合する人物動線検出技術の従来方式について述べる。

日高ら<sup>3)</sup>は、カメラで検出した軌跡とアクティブ RFID で検出した ID を統合し動線検出する手法を提案している。本手法では、追跡領域内に設定した ID 検出エリア内における軌跡数と ID 数に応じて、各軌跡に対して各 ID が割当てられる尤度を算出し、

<sup>†</sup>NEC サービスプラットフォーム研究所  
Service Platforms Research Laboratories, NEC Corporation

この尤度を時間方向に統合した値が最大となるような軌跡と ID の組の集合を動線検出結果とする。他にも、カメラと RFID を用いた同様のアプローチとして、中嶋ら<sup>4)</sup>、坂本ら<sup>5)</sup>の手法がある。中嶋らは異種センサ間の位置の類似度、坂本らは移動ベクトルの類似度に基づいて、軌跡と ID 情報の対応付けを行っている。このような従来方式はいずれも、現時刻までに欠落なく検出できた軌跡と ID から算出した尤度に基づいて軌跡と ID を統合している。

しかしながら、軌跡や ID の検出に失敗しセンサデータの欠落が頻発すると、尤度算出に利用できる情報量が減少し動線検出精度が低下する課題がある。ここで、軌跡の欠落とはオクルージョン等によりカメラによる人物追跡に失敗し軌跡が途切れることを意味し、ID の欠落とは ID 検出エリア内にも関わらず ID が検出されないことを意味する。この課題の具体例を図 1 に示す。図 1 では、追跡領域内を移動する 2 人について検出された軌跡 1~4 と ID1~2 を示している。軌跡および ID には欠落が生じ、現時刻では軌跡 3,4 と ID1,2 だけが検出されている。この場合、現時刻の情報のみでは軌跡 3,4 に ID1,2 のどちらかを割当ててべきか決定できない。

そこで、このようなセンサデータの欠落が生じた場合でも「ID 認証された個人ごとの」「途切れない」人物動線を検出するために、センサデータの履歴情報を利用した異種センサ統合による人物動線検出技術の従来方式が提案されている。

川島ら<sup>6)</sup>は、カメラで検出した軌跡と加速度センサを搭載した携帯端末に付与された ID を統合し動線検出する手法を提案している。本手法では、オクルージョン等により追跡領域内で断片化して検出した一定時間分の軌跡の組合せ候補を生成し、軌跡の組合せ候補から計算した人物の速度と、加速度センサから得られる垂直方向の加速度との関係性に着目し、これらの一致度を尤度として算出し、この尤度が最大となるような軌跡と ID の組の集合を動線検出結果とする。

しかしながら、このようなセンサデータの履歴情報の組合せから軌跡と ID の対応関係を推定する方式では、センサデータの欠落が生じるほどセンサデータの組合せ数が爆発的に増加することになる。単純に全通りの軌跡の組合せおよび軌跡の組合せと ID との組合せを生成することは、リアルタイム処理を想定した動線検出システムに適用する上で、処理時間および計算資源の観点から現実的ではない。したがって、リアルタイム処理を想定した動線検出システムに適用するためには、尤もでないセンサデータの組合せを効率的に枝刈りする仕組みや、少ない計算量で高速にセンサデータの組合せを生成し、尤度を算出し、最尤推定するための仕組みが必要になる。

以上の従来方式の特徴と提案方式が満たすべき特徴を表 1 に示す。本稿では、3 章で履歴情報を用いた頑健な動線検出方式の概要と、不要なセンサデータの組合せを生成しないための工夫について述べ、4 章で過去に生成したセンサデータの組合せ結果を活用することによるセンサデータの組合せ生成処理および尤度算出処理の高速化の工夫について述べる。

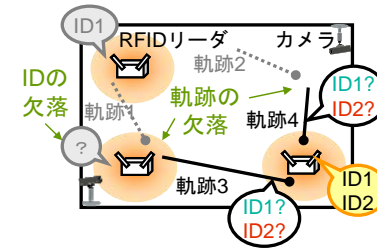


図 1 履歴情報を利用しない動線検出

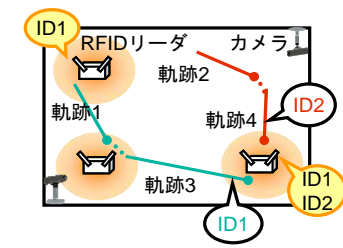


図 2 履歴情報を利用する動線検出

表 1 提案方式と従来方式の特徴

	ID 認証された動線	センサデータの欠落に対する頑健性	センサデータの組合せ生成時の高速性
日高ら <sup>3)</sup> 、中嶋ら <sup>4)</sup> 、坂本ら <sup>5)</sup>	○	×	×
川島ら <sup>6)</sup>	○	○	×
提案方式	○	○	○

### 3. 提案方式

#### 3.1 概要

センサデータの欠落に対して頑健に人物の動線を検出するために、現時刻から過去一定時間に検出した軌跡と ID の履歴情報を利用する人物動線検出方式を提案する。

過去に検出したセンサデータを用いることの有効性を示す具体例を図 2 に示す。図 2 では図 1 と同じセンサデータが検出されているものとする。この場合、過去の時刻では、左上の ID 検出エリアで得られたセンサデータより軌跡 1 に対して ID1 を割当てられる。現時刻では、この情報と右下の ID 検出エリアで得られたセンサデータに基づいて、軌跡 1 と連結する可能性が高い軌跡 3 に ID1 を、軌跡 4 に ID2 をそれぞれ割当てられる。このように、現時刻に検出しているセンサデータのみでは動線を正しく検出できない場合でも、過去に検出したセンサデータも活用することによって、より正確に軌跡と ID を統合し動線を検出することが可能となる。

#### 3.2 構成

提案方式では、入力される軌跡と ID の組合せから動線仮説を生成し、各仮説の尤度を軌跡および ID の検出結果から定量化し最尤推定することにより、軌跡と ID を統合する。ただし、全組合せの動線仮説を生成し最尤推定を行うと計算量が膨大になるため、仮説生成処理を軌跡の組合せ問題と ID の割当て問題に分割し、計算量を低減する(図 3)。具体的には、軌跡の組合せ問題で尤もらしい軌跡の組合せのみ生成し、

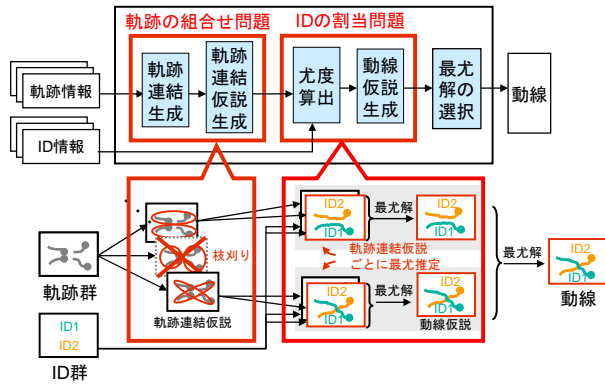


図 3 履歴情報を利用した動線検出方式の概要

生成された軌跡の組合せ毎に尤もらしい ID の割当結果を生成する。このように、各ステップで尤もらしいと判定した組合せおよび割当結果を後段の処理で使用することで処理量を低減する。

以降は、軌跡の組合せ問題と ID の割当て問題の処理の詳細について述べる。

### (1) 軌跡の組合せ問題

軌跡の組合せ問題とは、入力された  $m$  個の軌跡を  $n$  人分 ( $1 \sim m$  人分) に分ける問題である。また、 $m$  個の軌跡を  $n$  人分に分ける 1 つの組合せを軌跡連結仮説、軌跡連結仮説に含まれる  $n$  人それぞれの軌跡集合を軌跡連結と呼ぶ。

ここで、軌跡連結および軌跡連結仮説の生成方法について図 4 を用いて説明する。はじめに、軌跡の検出時刻が古いものから順に並べる。そして、検出時刻が最も古い軌跡から順に 1 つ選択し、選択した軌跡が 1 つ前の軌跡を選択した際に生成した軌跡連結と連結する場合、および連結しない場合の軌跡連結仮説を生成する。ただし、選択中の軌跡を軌跡連結に連結する場合、各軌跡の端点の位置座標と検出時刻より時空間的に連結しうるかどうかを判定し、連結しうる場合のみ連結する。具体的には、検出時刻が重複する軌跡間の連結判定と、検出時刻が重複しない軌跡の連結判定を行う。検出時刻が重複する場合は、各時刻で軌跡間の距離が予め定めた閾値以下であるかどうかを判定し、閾値を越える場合はその軌跡を軌跡連結に連結しない。検出する時刻が重複しない場合は、軌跡間の途切れ距離とおよび途切れ時刻から求めた移動速度が予め定めた閾値以下であるかどうかを判定し、閾値を超える場合はその軌跡を軌跡連結に連結しない。そして、全ての軌跡について上記の処理を行った時点で生成されている軌跡連結仮説が、過去一定時間分の軌跡情報を利用して生成した軌跡連結仮説となる。

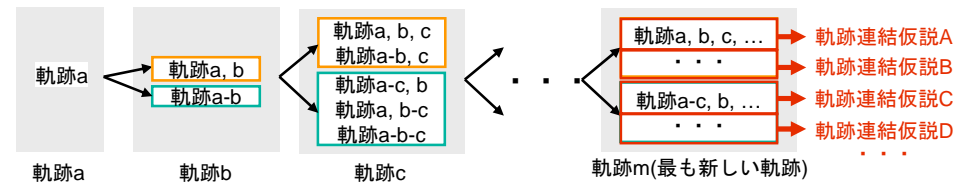


図 4 軌跡連結仮説の生成手順

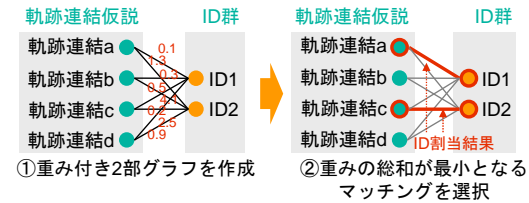


図 5 2部グラフを用いた割当問題

### (2) ID の割当問題

ID の割当問題とは、軌跡連結仮説 ( $p$  個の軌跡連結を含むとする) に対して  $q$  個の ID を割当てる場合に最も尤もらしい割当て方を求める問題である。また、 $p$  個の軌跡連結に  $q$  個の ID を割当てた 1 つの組合せを動線仮説と呼ぶ。

ここで、ID の割当問題を解くために必要な尤度の算出方法と、動線仮説の生成方法について説明する。

#### ● 尤度算出

生成した各軌跡連結仮説に対し、入力情報として得られる過去一定時間に検出した ID 情報を割当てた場合の尤度を算出する。

はじめに、追跡領域内における ID 出力確率分布を定義しておく。ID 出力確率は、実環境で得られた ID の検出特性を基に決定する。追跡領域上を移動する人物の  $x$  座標を  $x$ 、 $y$  座標を  $y$ 、RFID リーダの ID を  $rid$  とし、ある人物の任意の位置座標  $x, y$  についてある RFID リーダから検出される ID 出力確率を  $P_{rid}(x, y)$  と表す。

次に、軌跡連結仮説から得られる ID 出力確率を取得する。この際に、軌跡連結仮説の内挿および外挿による補間処理を行う。検出時間が重複しない軌跡間を内挿する場合は、軌跡の消失点から出現点を直線補間し、重複する軌跡間を内挿する場合は、重複する各時刻における軌跡間の midpoint を位置座標として補間する。また外挿する場合は、軌跡の出現点又は消失点に停留していたものとして補間する。補間処理を行った軌跡連結仮説の ID を  $tracks$  とし、ある時刻  $t$  における軌跡連結仮説の位置座標  $x$  を

$x_{tracks,t}, y$ , 位置座標  $y$  を  $y_{tracks,t}$  と表す. なお, 時刻  $t$  において軌跡連結仮説  $tracks$  が検出されていない場合は計算から除外する.

次に, ID 検出結果を  $S_{rid,tid,t}$  と表す. 時刻  $t$  に RFID リーダ  $rid$  でタグ  $tid$  が検出されている場合は 1 の値をとり, 検出されていない場合は 0 の値をとる.

以上より, 時刻  $t=s$  から時刻  $t=e$  までに検出した軌跡連結仮説  $tracks$  と  $IDtid$  が紐づく尤度は, 式(1)で算出する. なお, 式(1)では環境中に設置した RFID リーダの集合を  $R$  としている.

$$L_{tracks,tid} = \sum_{t=s}^e \sum_{R=rid} \log(S_{rid,tid,t} \times P_{rid}(x_{tracks,t}, y_{tracks,t})) \quad (1)$$

● 動線仮説生成

軌跡連結仮説に対する ID の割当問題を, 軌跡連結群と ID 群の頂点集合から成る重み付き 2 部グラフの最小重み最大マッチング問題に置き換えて解く (図 5). 2 部グラフにおける各辺の重みは, 式(1)で算出した尤度に基づいて算出したスコアを用いる. マッチング問題を解いた結果として得られる軌跡連結と ID の対応関係が, ある軌跡連結仮説と ID 群から得られる最尤解となる. このとき, ID を割当てられた軌跡連結は, その ID を持つ人物から検出された移動軌跡とみなし, ID を割当てられなかった軌跡連結は ID を持たない移動体(unknown)から検出された移動軌跡とみなす. そして, この最尤解を動線仮説とする. このように, 軌跡連結仮説毎に最尤推定を行い, 最尤解のみを動線仮説とすることで, 生成される動線仮説数を削減し計算量を低減する.

4. 提案方式の高速化

3 章で述べた提案方式では, 入力情報として現時刻から過去一定時間分の軌跡情報および ID 情報を取得し, これらの全データを用いて軌跡連結の生成, 軌跡連結仮説の生成, 尤度算出, 動線仮説の生成, 最尤解の決定を行う. この提案方式をリアルタイム処理を想定した動線検出システムに適用する場合, 一連の動線検出処理を一定時間 (1 秒など) ごとに逐次実行することになる. しかし, ある時刻  $t$  における入力情報と前回の動線検出時 (時刻  $t-1$ ) における入力情報との差分が僅かである場合あるいは差分が無い場合に全入力情報を対象に動線検出処理を行うと, 前回生成した軌跡連結, 軌跡連結仮説, 尤度と部分的に重複する情報を生成することになる. そのため, 一連の処理を終えるまでに不要な時間がかかる問題がある.

そこで, これらの冗長な演算処理を無くすために, 一連の動線検出処理過程で生成される軌跡連結, 軌跡連結仮説, 尤度等の中間処理結果を記憶・更新する仕組みを追加する (図 6). 入力情報として前回の動線検出処理時刻から現時刻までに検出された

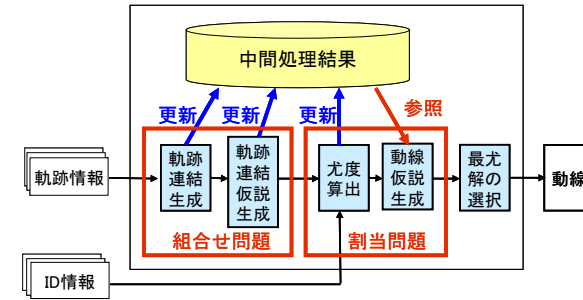


図 6 逐次更新による高速化の構成

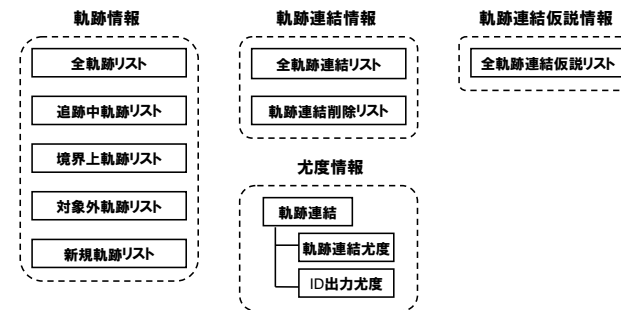


図 7 逐次更新する中間処理情報

表 2 軌跡情報の分類

分類	該当条件
追跡中軌跡	軌跡の検出開始時刻から現時刻まで途切れなく検出している軌跡が該当する.
境界上軌跡	軌跡の検出開始時刻が定義した時間窓よりも古くなった軌跡が該当する.
対象外軌跡	軌跡の検出終了時刻が定義した時間窓よりも古くなった軌跡, かつ, この軌跡が連結しうる軌跡の検出開始時刻が, 定義した時間窓よりも古くなった場合に該当する.
新規軌跡	前回の動線検出処理では検出されておらず, 現時刻の動線検出処理で初めて検出された軌跡が該当する.

軌跡情報および ID 情報を用い, 記憶している中間処理結果のうち, 新たに検出したセンサーデータ又は過去一定時間より古くなったセンサーデータに関連する箇所のみ更新すれば, 前述の冗長な演算処理が無くなり動線検出処理を高速化できる.

中間処理結果を効率的に逐次更新するために, 中間処理結果の状態を分類し管理す

る必要がある。本稿では、動線検出処理過程で生成される中間処理結果を図 7 のように分類した。以降では、各動線検出処理において、どのように中間処理結果を利用し更新するかを説明する。

### (1) 軌跡の組合せ問題

#### ● 軌跡連結の更新

現時刻から過去一定時間内に検出した全軌跡のうち、表 2 で示した分類に該当する軌跡を各々選択し記憶しておく。

表 2 の追跡中軌跡を含む軌跡連結について軌跡連結判定を行い、連結不可と判定された軌跡連結を図 7 の軌跡連結削除リストに追加する。

表 2 の対象外軌跡を含む軌跡連結に含まれる軌跡が 1 つの場合は、図 7 の全軌跡連結リストから軌跡連結を削除し、図 7 の軌跡連結削除リストに軌跡連結を追加し、対象外軌跡を含む軌跡連結に含まれる軌跡が複数存在する場合は、軌跡連結から対象外軌跡のみ削除し軌跡連結を更新する。

#### ● 軌跡連結仮説の更新

図 7 の軌跡連結削除リストに含まれる各軌跡連結を含む軌跡連結仮説を削除する。

図 7 の新規軌跡リストに含まれる各軌跡を検出開始時刻順に並べ替え、検出開始時刻の古い新規軌跡から順に、図 7 の全軌跡連結仮説リストに含まれる各軌跡連結仮説に追加し、軌跡連結仮説を更新する。

### (2) ID の割当問題

#### ● 尤度の更新

前回の動線検出処理時刻から現時刻の間に新たに取得した ID 情報を用い、軌跡連結ごとに、ID 情報の検出時刻における ID 出力尤度を対応付け尤度を更新する。また、軌跡連結に対応付けて記憶された ID 出力尤度のうち、過去一定時間外となった ID 出力確率を削除し尤度を更新する。

## 5. 評価

### 5.1 コンビニ実験による動線検出性能評価

提案手法の有効性を確認するため、カメラとアクティブ RFID を用いた人物動線検出システムに適用し、動線検出性能に関して評価を行った。

#### (1) 実験

図 8 のようなコンビニを想定した環境にカメラ 9 台、アクティブ RFID リーダ 3 台を設置し、一部の人物にアクティブ RFID タグを携帯させた。追跡領域全域でカメラ映像から軌跡を、図 9 に示した 4 箇所 ID 検出エリアで ID をそれぞれ検出できるように実験環境を構築し、表 3 に示した人数・行動の異なる 7 つのシナリオについてデータ収集実験を行った。また、各シナリオで発生した軌跡と ID の欠落回数は表 3 に



図 8 実験環境

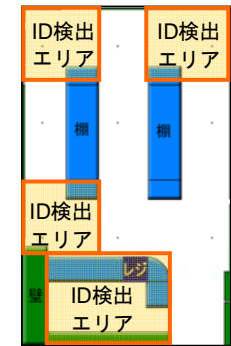


図 9 追跡領域の地図

表 3 シナリオ別の登場人数とセンサデータの欠落回数

シナリオ	RFID タグ 所持人数	RFID タグ 不所持人数	軌跡の 欠落回数	ID の 欠落回数
S1	2 人	6 人	14 回	1 回
S2	2 人	6 人	15 回	1 回
S3	1 人	3 人	10 回	0 回
S4	2 人	5 人	11 回	0 回
S5	1 人	5 人	10 回	2 回
S6	1 人	5 人	13 回	1 回
S7	1 人	5 人	12 回	1 回

示した通りである。なお、本実験で用いたシナリオの時間長は各シナリオ 60~120 秒程度であった。

#### (2) 評価結果

ここでは、センサデータの欠落が生じた入力データに対して、ID 情報を利用することによる軌跡連結精度への影響と、履歴情報を利用することによる ID 割当精度への影響について評価する。そのため、動線検出処理を行う際に、ID 情報の使用状況（使用/不使用）と履歴長の使用状況（最新の軌跡のみ、履歴 10 秒、履歴 20 秒、履歴 40 秒、履歴 60 秒、履歴 300 秒）を変えて比較評価した。なお、ID 情報を使用しない場合は ID 出力尤度を算出できないため、途切れた軌跡間の途切れ距離および途切れ時間に基づいて算出する軌跡連結尤度を用い、動線検出処理を行った。

今回は、ID 情報の利用による軌跡連結性能への影響を評価するため、軌跡連結時間率および軌跡連結回数率を評価した。また、履歴情報の利用による ID 割当性能への影響を評価するため、ID 割当の適合率・再現率を評価した。

● 軌跡連結時間率

軌跡連結時間率とは、ある人物の動線について、どれだけの時間正しく検出できているかを示す指標である。軌跡連結時間率は人物の実際の移動軌跡に対する正解した移動軌跡の割合として算出した。算出した軌跡連結時間率について、入力に使用したセンサデータの履歴長別にまとめた結果を表 4 に示す。

● 軌跡連結回数率

軌跡連結回数率とは、ある人物の軌跡断片について、どれだけの回数を正しく連結できたかを示す指標である。軌跡連結回数率は軌跡の途切れ回数に対する正しく連結できた回数として算出した。算出した軌跡連結回数率について、入力に使用したセンサデータの履歴長別にまとめた結果を表 5 に示す。

● ID 割当の適合率・再現率

ID 割当の適合率・再現率とは、ある人物の軌跡断片について、どれだけ正しい ID を割当られているかを示す指標である。ID 割当の適合率を検出結果中の全 ID 付軌跡数に対する正解した ID 付軌跡数の割合、再現率を正解データ中の全 ID 付軌跡数に対する正解した ID 付軌跡数の割合として算出した。算出した適合率・再現率について、入力に使用したセンサデータの履歴長別にまとめた結果を表 6 に示す。なお、ID 不使用の場合は ID 割当の適合率・再現率は算出できないため、ID を使用する場合の結果のみを示している。

(3) 考察

● ID 情報を利用することによる軌跡連結精度への影響

表 4, 表 5 の結果より、ID 情報を使用した場合の方が ID 情報を使用しない場合に比べ、正しく軌跡を連結できることが分かる。また、軌跡連結時間率の方が軌跡連結回数率よりも良好な結果が得られていることから、連結に失敗した軌跡の多くは検出時間がごく短い軌跡であることが分かる。したがって、今回の実験結果からは、ID 情報を用いることによって、検出時間の短い軌跡を取りこぼさずに連結できる効果が得られることが分かる。

ところで、表 4 の結果では、ID 情報を使用する場合と使用しない場合の性能差はごく僅かである。このことから、実環境で検出される軌跡は、軌跡間の距離と時間情報を用いて連結できる場合が多いと推察できる。例えば、人物同士のすれ違いによって軌跡が途切れた場合には、軌跡間の距離と時間情報だけでは人物がすれ違ったのか U ターンしたのかを判別できないため、軌跡を正しく連結することは難しい。一方で、人物が 1 人で移動している状況で軌跡が途切れた場合は、軌跡間の距離と時間情報が近いものを選択し、途切れた軌跡を連結することは容易である。今後は、軌跡が途切れた際の状況を考慮して性能評価を行い、その結果を基に ID 情報を利用することによる軌跡連結精度への影響について再度考察する必要がある。

また一方で、今回の実験では、入力情報として与えた ID 検知情報がごく僅かであ

表 4 軌跡連結時間率

	最新の軌跡のみ	過去 10 秒	過去 20 秒	過去 40 秒	過去 60 秒	過去 300 秒 (全履歴)
ID 使用	73.60%	71.49%	75.29%	79.86%	85.55%	85.58%
ID 不使用	66.21%	67.92%	72.11%	78.53%	81.87%	83.13%

表 5 軌跡連結回数率

	最新の軌跡のみ	過去 10 秒	過去 20 秒	過去 40 秒	過去 60 秒	過去 300 秒 (全履歴)
ID 使用	3.53%	42.35%	57.65%	64.71%	67.06%	65.88%
ID 不使用	0%	30.59%	32.94%	49.41%	48.24%	55.29%

表 6 ID 割当の適合率・再現率

	最新の軌跡のみ	過去 10 秒	過去 20 秒	過去 40 秒	過去 60 秒	過去 300 秒 (全履歴)
適合率	90.00%	65.96%	76.19%	89.19%	92.50%	91.18%
再現率	71.05%	75.61%	78.05%	80.49%	90.24%	75.61%

ったため、ID 情報を利用することによる効果が得られにくい可能性も考えられる。今回使用した ID 検知情報は、再現率を犠牲にして適合率を高くするというチューニングを行っている。このチューニングを選択した理由は、ID の検出漏れよりも誤検出が起こる方がセンサデータの統合時に悪影響を及ぼすと判断したためである。今後は、ID 情報の適合率と再現率のバランスを変えた入力データを複数パターン用意して同様に性能評価を行い、その結果を元に ID 情報を利用することによる軌跡連結効果について再度考察する必要がある。

● 履歴情報を利用することによる ID 割当精度への影響

表 6 より、入力に使用する履歴情報の時間長を増やすことで ID の割当精度が向上する傾向を確認できる。しかし、履歴長が 60 秒の場合と 300 秒の場合とを比較すると、精度の低下がみられた。これらの結果から、より多くの履歴情報を利用することによって動線検出性能は向上していることが分かる。

しかし一方で、履歴情報の時間長を増やせば増やすほど、履歴情報に軌跡情報および ID 情報の誤検出が含まれる確率も高まることになる。提案する動線検出方式では入力情報の欠落に対するロバスト性を向上してきたが、入力情報の誤りに対するロバスト性を向上する工夫は取り入れていない。ゆえに、履歴情報の時間長を増やすことで履歴情報にセンサデータの誤検出が含まれる可能性が高まり、かえって動線検出精度が低下する場合がある。今後は、センサデータの誤検出に対するロバスト性を向上するための方式改良に取り組む必要がある。

表 7 シミュレーションデータ作成時の条件

追跡領域	6.5m×8.3m (コンビニ実験環境と同様)
登場人物	10人 (全員 ID タグ所持)
シナリオ時間長	120秒

表 8 動線検出処理の条件

履歴時間長	60秒
軌跡連結仮説の最大性整数	10000個
動線検出処理頻度	1秒間隔

## 5.2 シミュレーションデータを用いた動線検出処理速度の評価

逐次更新を行う動線検出方法と逐次更新を行わない動線検出方法とを比較し高速化の効果を計測するため、シミュレーションデータを用いた評価を行った。

### (1) 実験

5.1 節で述べたコンビニ実験環境を模して、表 7 に示した条件に従ってシミュレーションデータを作成した。シミュレーションデータとは、時刻ごとの軌跡の位置情報を csv 形式で記載したテキストファイルと、時刻ごとの ID 検知情報を csv 形式で記載したテキストファイルを意味する。なお、軌跡の途切れや ID 情報の欠落頻度に関しては、コンビニ実験環境で得られた傾向と同様になるように設定した。

作成したシミュレーションデータを用い、表 8 に示した条件に従って動線検出処理を行った。

### (2) 評価結果

ここでは、逐次更新処理を適用することによる動線検出処理時間への影響について評価する。そのため、前述のシミュレーションデータを用い、逐次更新なしの動線検出方式と逐次更新ありの動線検出方式とでそれぞれ動線検出処理時間を計測した。また、動線検出処理全体で要した処理時間と、動線検出処理に含まれる軌跡連結・軌跡連結仮説生成処理および尤度算出処理で要した処理時間をそれぞれ計測した。

#### ● 動線検出処理全体の処理時間

動線検出処理全体の処理時間の比較結果を図 10 に示す。縦軸を処理時間(msec)、横軸を軌跡連結仮説数とし、1 回の動線検出処理において生成された軌跡連結仮説数と動線検出処理全体の処理時間との関係を表している。シナリオ開始時刻から 1 秒おきに過去 60 秒分の履歴情報を用いて動線検出しており、図 10 にプロットされている 1 つの点は、1 秒おきに行った 1 回の動線検出処理に要した時間を表している。図 10 より、逐次更新なしの場合と比較して、処理時間を約 20 倍以上高速化する結果を得た。

#### ● 軌跡連結・軌跡連結仮説生成処理時間

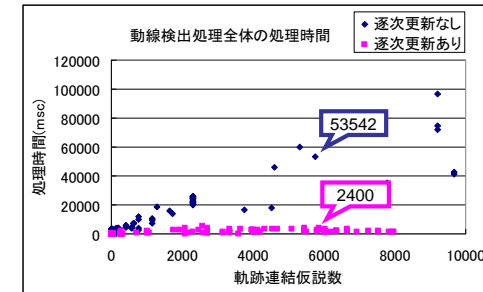


図 10 動線検出処理全体の処理時間

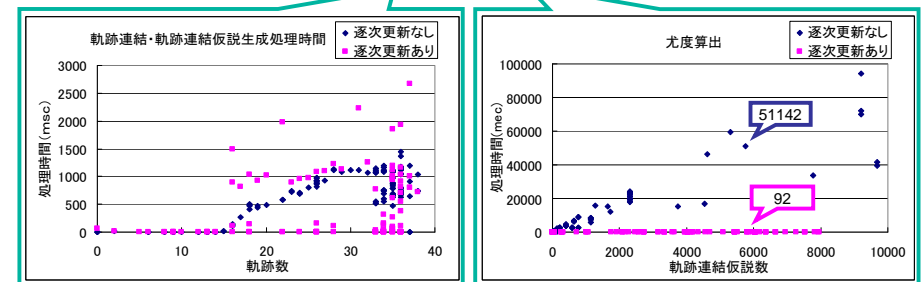
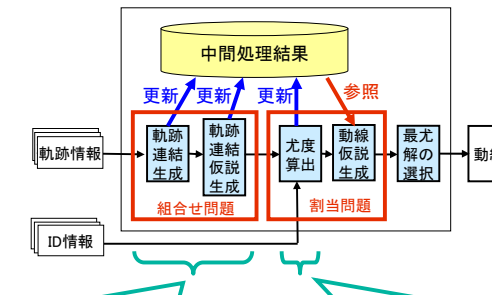


図 11 軌跡連結・軌跡連結仮説生成処理時間 (左) と尤度算出処理時間 (右)

軌跡連結の更新および軌跡連結仮説の更新処理を合わせた処理時間を比較した結果を図 11 (左) に示す。図の見方に関しては、横軸を過去一定時間内に検出した軌跡数としている点以外は図 10 と同様であるため、説明を省略する。図 11 (左) より、逐次更新ありの場合となしの場合とで、処理時間にほぼ変化がない結果を得た。

### ● 尤度算出処理時間

ID 出力尤度算出処理時間の比較結果を図 11 (右) に示す。図の見方に関しては、図 10 と同様であるため説明を省略する。図 11 (右) より、逐次更新なしの場合と比較して、処理時間を約 500 倍以上高速化する結果を得た。

### (3) 考察

動線検出処理全体でみると、処理時間を約 20 倍以上高速化する結果が得られたことから、逐次更新処理による高速化の効果は大きいことが分かる。これは、尤度算出に要する処理時間を大幅に短縮できたことが大きな要因である。

軌跡連結・軌跡連結仮説生成処理時間に関しては、4 章で述べたように軌跡連結や軌跡連結仮説に関する中間処理結果を複数の状態に分類し、状態ごとに異なる更新処理を行う必要があるため、軌跡連結仮説の更新処理が煩雑になっている。そのため、毎回一から仮説を作成する逐次更新なしの場合と比較して、処理を簡略化できなかったと考えられる。従って、過去に生成した処理結果を効果的に活用するには、更なる改良が必要である。

ID 出力尤度算出時間に関しては、時間窓よりも古くなった ID 出力確率を削除し、新しく検出された ID から得られる ID 出力尤度のみを追加すればよく、更新処理が単純であるため、過去に生成した処理結果を効果的に活用できていることが分かる。

また、本研究では中間処理結果の逐次更新による高速化について取り組んできたため、中間処理結果を参照して利用する動線仮説生成処理に関しては、逐次更新なしの場合と同様の処理を行っている。しかし、現状では動線検出処理全体のうちの約 50% 以上を動線仮説生成処理に要しており、今後更なる改良が必要だと考えられる。今後の改善方針として、例えば、ID 割当処理を行うべき仮説を効率的に選択し、ID 割当の処理回数を削減することで高速化するといったことが考えられる。

## 6. おわりに

本稿では、リアルタイム処理を想定したセンサデータの欠落に対して頑健な人物動線を検出する提案方式を提案した。提案方式では、過去一定時間分の軌跡と ID の組合せを人物動線の仮説として生成し、各仮説の尤度を軌跡および ID の検出結果から定量化し最尤推定することにより軌跡と ID を統合する。また、高速化するため、動線検出処理過程で生成される中間処理結果を記憶する仕組みを追加し、新たに入力されたセンサデータにより変化を受ける箇所のみを更新することによって冗長な演算処理を無くした。

センサデータの履歴活用による動線検出処理の性能評価を行うため、提案方式を実装し、カメラとアクティブ RFID を用いた人物動線検出システムに適用した結果、ID 情報を利用することで軌跡連結精度が向上すること、履歴長を長くすることで ID 割当精度が向上することを確認した。しかし一方で、履歴長を長くし過ぎることによっ

て履歴情報に誤検出したデータが含まれる可能性が高まり、却って動線検出精度が低下する場合もあることが明らかになった。

また、逐次更新による動線検出処理の高速化の評価を行うため、シミュレーションデータを用いた評価を行った。逐次更新ありの場合となしの場合とで処理時間を比較評価した結果、逐次更新ありの方が約 20 倍高速化できていることを確認した。しかし、処理時間全体のうち、本稿で高速化の改良を加えなかった動線仮説生成処理に要する処理時間は、動線検出処理全体の 50% 以上を占めていることが明らかになった。

今後は、センサデータの欠落や誤りに対してよりロバストな尤度算出方式を検討し動線検出精度を改善すると共に高速化の改良を行い、更なる性能改善を目指す。

## 参考文献

- 1) Mori, Suemasu, Noguchi, Sato, "Multiple People Tracking by Integrating Distributed Floor Pressure Sensors and RFID System", IEEE International Conference on System Man and Cybernetics (2004).
- 2) 森口, 小西, 中尾, "異種センサ統合によるセンサデータの欠落に頑健な人物動線検出方式", 情報処理学会全国大会講演論文集, Vol.72nd, No.3, pp.3.89-3.90, 2010.
- 3) 日高, 石寺, "アクティブ RFID リーダとマルチカメラを用いた追跡手法", 情報科学技術フォーラム講演論文集, Vol.7, No.3, pp.161-162 (2008).
- 4) 中嶋, 伊藤, 新田, 馬場口, "RFID タグとカメラ映像を用いたサーベイランス映像中の人物同定", 電子情報通信学会技術研究報告. PRMU, パターン認識・メディア理解, Vol.107, No.281, pp.81-86 (2007).
- 5) 坂本, 佐野, 久保, 岡本, "ステレオカメラと RFID-Tag を併用した個人位置同定手法", 情報処理学会全国大会講演論文集, Vol.68th, No.2, pp.359-360 (2006).
- 6) 川島, 新野, 田島, 新谷, 芳賀, 金田, "複数センサ情報の統合によるロバストな人物動線検出手法の提案", 情報処理学会研究報告, Vol.2009-UBI-22, No.3, 2009.