

モバイルエージェントによる  
センサ設置環境の変化に対応した  
人物追跡システムの構築に関する研究

2017年 7月

四元 辰平

# 目次

第1章	序論	1
第2章	人物追跡システムについて	6
2.1	序言	6
2.2	現在における監視システムについて	6
2.2.1	カメラを用いた映像監視システム	7
2.2.2	GPSを用いた監視システム	9
2.2.3	ビーコンを用いた監視システム	10
2.3	センサの配置と人物の追跡	11
2.4	集中型のシステムと分散型のシステム	12
2.5	提案する人物追跡システム	14
2.5.1	対象とするシステム	14
2.5.2	システム構成	17
2.5.3	システムの特徴	19
2.5.4	人物追跡の流れ	21
2.6	人物追跡システムの実装と実験	23
2.6.1	人物追跡システムの実装	23
2.6.2	モバイルエージェントによる人物追跡の実験	29
2.7	結言	34
第3章	センサの検出範囲の変化に対応した隣接関係の導出法	35
3.1	序言	35
3.2	センサの検出範囲と人物の追跡	35
3.2.1	センサの検出範囲の変更における問題点	36
3.3	隣接ノード決定アルゴリズムによる問題解決	37

3.3.1	隣接センサ	37
3.3.2	追跡の流れ	39
3.3.3	隣接関係の計算	41
3.4	評価	45
3.5	結言	47
<b>第4章</b>	<b>隣接関係の局所化手法</b>	<b>48</b>
4.1	序言	48
4.2	各センサが持つ情報の局所化	49
4.3	隣接関係の局所化による計算手法	50
4.4	削減例	51
4.5	検出範囲の変化への対応	54
4.5.1	センサの追加への対応	55
4.5.2	センサの削除への対応	55
4.6	評価	56
4.6.1	局所化による点の削減効果	57
4.6.2	追跡継続性の確認	58
4.7	結言	62
<b>第5章</b>	<b>センサの不確実性に対応した人物追跡手法</b>	<b>63</b>
5.1	序言	63
5.2	隠れ隣接関係の導出による問題解決	65
5.2.1	センサの未検出への対応	65
5.2.2	未検出への対応	66
5.2.3	隠れ隣接関係の導出	70
5.2.4	隠れ隣接関係の反映	72
5.2.5	隣接関係の広がりの抑制	72
5.3	実験	74
5.3.1	仮追跡の動作確認	74
5.3.2	隠れ隣接関係による追跡成功率の変化	76
5.3.3	隠れ隣接関係の削除方式の比較	77
5.3.4	閾値による追跡成功率の比較	78

5.4	センサの検出率が低い場合における考察 . . . . .	83
5.5	誤検出への対応 . . . . .	84
5.6	結言 . . . . .	85
<b>第 6 章</b>	<b>結論</b>	<b>86</b>
6.1	まとめ . . . . .	86
	謝辞	89
	参考文献	91
	公表論文	97

# 図版目次

2.1	提案システム	15
2.2	監視画面イメージ	16
2.3	システム構成の概要	17
2.4	人物を検出し監視画面に表示する仕組み	18
2.5	人物追跡の流れ	22
2.6	人物追跡システムの実装	23
2.7	Bluetooth と Wi-Fi アダプタを装着した Raspberry Pi 2	25
2.8	iBeacon タグ (MB002)	26
2.9	ビーコンタグ (MAMORIO)	27
2.10	LED 点灯回路の回路図	28
2.11	LED 基盤を搭載した Raspberry Pi	28
2.12	簡易実験に用いたマップ	29
2.13	実験に用いたマップ	30
2.14	人物の移動とエージェントの追跡結果（追従を意識して移動した場合）	32
2.15	子エージェントの配布先	33
2.16	人物の移動とエージェントの追跡結果（追従を意識せずに移動した場合）	33
3.1	隣接センサ	38
3.2	エージェントの追跡アルゴリズム	40
3.3	センサの検出範囲と行列 $X, Y$	42
3.4	センサの検出範囲と行列 $X', Y'$	44
3.5	追跡実験の結果	46

4.1	マップにおける局所化の例	51
4.2	センサの追加による点の変化	56
4.3	センサの追加による点の変化	57
4.4	シミュレーション環境のマップ	61
4.5	追跡実験の結果	61
5.1	センサの未検出のイメージ	64
5.2	見逃しの発生	64
5.3	未検出への対応	67
5.4	統合のために追加する処理	69
5.5	隠れ隣接関係の導出	71
5.6	シミュレーションマップ	74
5.7	シミュレーション結果	75
5.8	追跡成功率	76
5.9	隣接センサ数の平均	77
5.10	複数回の実験結果（追跡成功率）	78
5.11	追跡成功率（センサ検出率 20~50%）	79
5.12	隣接センサ数の平均（センサ検出率 20~50%）	79
5.13	追跡成功率（センサ検出率 90%）	80
5.14	隣接センサ数の平均（センサ検出率 90%）	80
5.15	追跡成功率（削除方式による比較）	81
5.16	隣接センサ数の平均（削除方式による比較）	81
5.17	追跡成功率（閾値による比較）	82
5.18	隣接センサ数の平均（閾値による比較）	82
5.19	仮追跡エージェントが複数生成される例	83

# 表目次

2.1	Raspberry Pi 2 Model B のスペック . . . . .	25
2.2	Raspberry Pi 3 Model B のスペック . . . . .	25
2.3	iBeacon タグ (MB002) のスペック . . . . .	26
2.4	ビーコンタグ (MAMORIO) のスペック . . . . .	27
2.5	簡易実験における監視ノードの隣接関係 . . . . .	29
2.6	監視ノードの隣接関係 . . . . .	31
2.7	簡易実験における追跡対象人物の移動方法 . . . . .	31
4.1	マップ全体の点の数 . . . . .	58
4.2	局所化したときの点の数 . . . . .	59
4.3	測定に用いた計算機 . . . . .	59
4.4	隣接関係の計算にかかる時間 . . . . .	60
5.1	センサの検出結果と現実の事象の分析結果 . . . . .	66

# 第1章 序論

IoT (Internet of Things) 時代の到来により, 自動車や電車などの乗り物, テレビや冷蔵庫などの家電, 医療機器やドアなどの設備に至るあらゆるモノがネットワークを介してつながりつつある。IoTの主役はセンサである。あらゆる場所に配置されたセンサにより数値化されたデータが収集され, ネットワークを介してシステムと連携することであらゆる場所の状況を収集しその情報に基づいた判断を行いサービスを提供することが可能になりつつある。

センサを活用しネットワークにつながったシステムとしてまず考えられるのが, 離れた位置にある人や物体の状況把握とその制御を行うシステムである。例えば, スマートフォンを所持したユーザがドアに近づくとドアを開錠する製品<sup>1)</sup>や, 小児が下校する際に校門を出ると保護者に通知を行うようなシステム<sup>2)</sup>が既に実用化されている。またバスなどの乗り物がネットワークにつながることで, バスの現在位置などのリアルタイムな情報を得ることのできるシステム<sup>3)4)</sup>も実用化されている。

人物の所在を把握するシステムは昔から存在する。最も普及しているものとして映像監視システムがある。これは離れた場所に存在するカメラの映像を一カ所で監視員が監視することのできるシステムである。古くは有線のアナログカメラから, 現在はIPネットワークを介して映像を受信し離れた場所の映像を見ることが出来る。映像監視システムは複数の場所を同時に監視することができるため有用なシステムであるが, カメラに映った映像から監視対象の人物を判断しその行動を把握するのは監視員の役目である。人物を追跡してその行動を把握するために, 監視員は人物の撮影されているカメラを切り替えて追いかける必要がある。この作業の負担は非常に大きく, カメラの数と追跡対象の人物の人数が多くなるほど人間では対応できなくなる。

そこで, 人物を自動で検出しその移動経路を自動で収集する, 人物追跡システムが研究されている。人物の行動を把握することは犯罪者の追跡や, こどもや社



会的弱者のような保護すべき人物の所在を把握するだけでなく、人物の行動分析からマーケティングや医療への活用<sup>5)</sup>が期待される。このため、カメラやセンサが検出した人物の認識から、移動方向や移動速度からの移動経路推定といった、様々な技術の研究がなされている。

人物追跡システムにおいて使用するセンサは、世の中に広く利用されている人物を監視するシステムで使用されるセンサと違いはない。主に用いられるセンサは、カメラ、GPS、近年はビーコンなどである。カメラを用いる人物追跡システムとして、画像から人物を識別して人物の位置を追跡するシステムが実用化されている。画像処理による人物認識は、機械学習やディープラーニング技術の登場により近年目覚しく発達している分野である。しかし、画像の処理には膨大な処理コストを必要とし、識別精度においても完全なシステムはまだ登場していない。GPSを用いた追跡は、見守りサービス<sup>6)7)8)9)10)</sup>や捜査機関での捜査活動にすでに用いられている技術である。GPS受信機を持たせることで、人物がいかなる場所においてもおおよそその位置を示すことができるため多く用いられている。しかし、その精度は大きくばらつきがあり、屋外であることが条件となる。近年はビーコンを用いて屋内での追跡を行うシステム<sup>11)12)</sup>が実用化されている。ビーコンは電波の発信機であり、発信側を壁に取り付けてGPSと同様に壁からの電波を受信しておおよそその位置を把握するものと<sup>12)</sup>、発信機をタグとして人物に持たせ、周りに設置した受信機で発信機の位置を把握するシステム<sup>13)</sup>とがある。GPSとビーコンについては、タグや発信機、スマートフォンなどのデバイスを所持する人物を追跡することが前提であり、何も所持しない人物を追跡することは不可能である。

それぞれのセンサの特徴を生かした監視システムが世の中には多数存在するが、人物を追跡するシステムでは、個人を特定しその人物の位置を正しく把握することが重要になる。一般的にセンサを用いたシステムで位置の精度を向上するためには、センサ自体の精度を向上するか、センサをより密に配置する必要がある。いずれもコストの増大につながる。センサ自体の精度向上は技術的な課題を解決するためのコストが大きく、性能や感度の良いセンサは高価である。感度の向上以外にも時間的に信号の発信周期などを細かくすることで精度の向上は期待できるが、その分消費電力が増大するため性能とエネルギーのトレードオフとなる。センサをより密に配置する場合は、GPSであれば衛星を増やす、カメラであれば全フロアを切れ目なく撮影する、ビーコンであれば電波を受信するセンサを多数

配置することにより精度向上が見込まれるが、装置の台数が増えることでシステムとしてのコストは大きく増加する。監視システムや人物追跡システムの運用においては、コストを抑えるためにより少ないセンサで確実に人物の位置を把握できるようにセンサを配置する工夫が行われている。例えば、カメラで撮影する場合は出入口や改札のような人が確実に通る場所が映るようにカメラを設置するといった工夫である。そのため、監視システムや人物追跡システムでは、センサは離散的に配置されていることが多い。

センサが離散的に配置された環境での人物追跡では、センサがどの位置に配置され、人物がどの順番でセンサに検出されたかを把握できることが重要である。そのためには全てのセンサの位置をシステムが把握し、どの人物がセンサで検出されても特定の人物と識別できる必要がある。情報を中央のサーバで集中して管理するシステムでは、この対策としてはその膨大な情報から一致する特徴データを検索する手法が一般的である。この場合、1台のサーバあるいはクラウドが全てのセンサの情報と追跡対象の人物の情報を管理し、センサから送られてくる全ての情報を処理して人物を識別する。人物がいつどのセンサに検出されるか分からないため、各センサは検出した人物の情報をサーバに常に問い合わせる必要がある。このため、多数のセンサが配置され多数の人物が追跡対象となるほど、通信やサーバでの処理コストが増大する。

本研究では、様々なシステムの規模に対応可能な、複数の人物を同時に追跡できる人物追跡システムの構築について提案する。システムの適用対象としては、複数の建物や組織を跨って追跡できることを考慮する。そのため、中央のサーバで全ての情報を集中管理するシステムではなく、分散構成とする。分散構成のメリットは、計算機の負荷を抑えられることと、任意に拡張できることである。分散構成を前提とすることで、全ての人物を全てのエリアを対象に監視するための膨大な処理コストを現実的な範囲に抑制することができる。

そのため提案するシステムでは、人物を検出するためのセンサと一体化した小型計算機をフロアに設置し、それぞれの小型計算機で人物の識別処理を行う。このセンサと一体化した小型計算機をノードと定義する。小型計算機にはシングルボードコンピュータである Raspberry Pi<sup>14)</sup> を使用することで低コストなシステムの構築を目指している。また、単一のサーバに人物の情報を送信せず、追跡対象の人物の周辺の局所化された範囲に存在するノード間で人物を追跡するために、モバ

イルエージェント技術<sup>15)16)17)</sup>を適用する。モバイルエージェントはデータを持った自律的に動作するプログラムがネットワークに接続されたコンピュータ間を移動しながら処理を行う技術である。個々の人物に対応したエージェントがその人物の特徴データを持って、次に人物が出現すると予測される場所に移動し、そこに存在するセンサが接続されたコンピュータ上でエージェントが持つ特徴データと同じ特徴を持つ人物の監視を行う。人物を検出すると更にその先に人物が出現すると予測される場所に移動して人物を監視する。モバイルエージェント自身はセンサの種類に依存せず、個人の特徴データを保持し同じ特徴データを検出したノードの近辺に展開して監視を行う。このようにモバイルエージェント技術を適用することで、低コストなハードウェアを組み合わせ、センサの種類に依存しない柔軟性に優れたシステムを構築できる。

このようなシステムを実現するためには、追跡対象の人物の周辺のノードに人物の特徴情報を受け渡して対象の人物を監視する仕組みが必要になる。そのため、あるノードが人物を検出した場合に次に人物を検出する可能性のあるノードを把握する必要がある。本稿ではこのノード間のつながりの関係を隣接関係として定義し、隣接関係からつながりのあるノードにモバイルエージェントを派遣し人物を追跡する仕組みを提案する。隣接関係はフロア図などの既知の情報から、センサの設置場所と通路のつながり、センサの検出範囲により計算し求めることができる。しかし、現実の人物追跡システムの運用では、カメラのパン・チルト・ズームのような撮影範囲の変化や、センサの故障や復帰、センサの増設などといったセンサの設置環境の変化によりモバイルエージェントを派遣すべきノード、つまり隣接関係に影響を及ぼす。また、センサには精度があり、必ずしも設置されているセンサが常に確実に人物を検出できるとは限らない。

そこで本稿では現実の人物追跡のシステムの運用において想定される、センサの設置環境の変化に対応した人物追跡システムを提案する。まず初めに、カメラのパン・チルト・ズームのようなセンサの検出範囲の変化に対応した隣接関係を計算する手法を提案する。センサや通路の分岐といった位置の情報にセンサの検出範囲を示す情報を加え、センサの検出範囲の変化や跨りに対応可能な隣接関係の計算式を提案する。次に、センサの故障や復帰、センサの増設に対応し隣接関係を再計算する手法を提案する。隣接関係の再計算においては、各ノードが持つノード間の隣接関係の情報を再計算が発生するたびに更新することが必要になる。

変化が発生するたびに全てのノードの隣接関係の情報を更新することは非効率であるため、再計算の影響を受ける可能性のあるノードに対してのみ情報を更新する、隣接関係の情報の局所化と更新手法を提案する。また、センサが人物を検出できないことは、人物追跡システムにおいては人物の見逃しにつながる。本稿では、センサが人物を確実に検出できないことをセンサの不確実性と定義する。このセンサの不確実性がフロア図やセンサの検出範囲といった設備の情報から次に人物を検出するノードの予測に影響を及ぼす。この問題を解決するために、設備の情報から計算した隣接関係を、センサが実際に人物を検出した情報により更新し、センサの不確実性に対応した追跡手法を提案する。

本稿は以下のように構成されている。まず2章において、提案する人物追跡システムの概要を述べる。3章ではセンサの検出範囲の変化に応じてノードの隣接関係を決定するアルゴリズムについて説明する。4章では分散システムに適用するにあたって各ノードが多く情報を管理しないで済むような局所化手法について述べる。5章ではセンサの不確実性に対応した追跡手法について述べ、最後に6章で本稿をまとめる。

## 第2章 人物追跡システムについて

### 2.1 序言

本稿における人物追跡システムとは、各場所に設置されたセンサが人物を検出し、その検出に基づいて人物の位置と検出時刻を記録し、その移動経路を可視化できるシステムである。ここでは、従来の監視システムによる追跡の問題を明確にし、その問題を解決するためにモバイルエージェントを用いた分散システム構成を検討する。モバイルエージェントを利用して各場所での監視と人物の特徴情報の受け渡しを行うことにより、システムの拡張性と頑健性を兼ね備えたシステム構成となる。

以降、2.2節では、現在の監視システムやその問題点、それぞれのセンサから人物の特徴情報を抽出する方法について述べる。2.3節でセンサの配置と追跡手法について考察し、2.4節では人物を監視し追跡するシステムにおいて、集中型のシステムと分散型のシステムについて言及し、本稿において分散型の構成を採用した理由を説明する。2.5節では、提案する人物追跡システムの構成や特徴、人物追跡の流れについて説明する。2.6節では、小型計算機を用いて構築した追跡処理の概要と動作実験について説明する。最後に2.7節でまとめを述べる。

### 2.2 現在における監視システムについて

従来、人物を監視するシステムと言えば主にカメラによる映像監視システムを指していた。近年はGPSやビーコンといった新しい技術が出てきたことで、他の方法でもより確実に人物の位置を監視することが可能になりつつある。また、ターゲットを追跡するためには、ターゲットと同一であることを判定する必要がある。本稿ではターゲットと同一であることを判定する行為を同定すると言い、ターゲットが同一であることを判定する処理を同定処理と言う。ターゲットの同定には、複

数のセンサから取得した情報から同じターゲットであるかどうかを判断する必要がある。

次節に代表的な人物の位置を監視するシステムとそれぞれのシステムが用いるセンサにおける特徴情報の抽出方法について述べる。

### 2.2.1 カメラを用いた映像監視システム

監視カメラの映像を記録・閲覧する映像監視システムは、社会に広く普及している。映像監視システムは、定点カメラでライブ映像のみを表示するシステムから、カメラ映像の録画やカメラ映像の手動切り替えの機能を持つシステムなど多様なシステムが存在する。画像や映像はその情報量が多く、人物の位置だけでなくその行動も把握できるため非常に有用なシステムである。カメラ映像の録画機能は、システムに設置されたカメラ毎の映像を記録する機能である。録画機能は、システムによっては監視カメラとセンサを共に設置し、センサに反応があった時に限りカメラ映像を記録する場合もある。監視員は、カメラのライブ映像を閲覧し確認することで、建物内等で不審人物や異常が無いかを監視したり、録画機能により記録された映像を閲覧、確認することができる。カメラ映像の手動切り替え機能は、カメラが多数存在し、監視用の画面にそれらの映像が収まり切らない場合、閲覧中のカメラの向きなどを手動で切り替えることができる機能である。しかし、人物に注目し監視をする時、映像監視システムの機能には以下のような問題点が存在する。

**記録された映像の検索には監視員の目視が必要：**

監視していた不審人物を記録された映像から検索するには、監視を行っていた監視員が記録された映像を閲覧し探し出す必要がある。

**不審者の判別には監視員の目視と判断が必要：**

不審者の判別には、各カメラのライブ映像を目視し、不審であるか判断する必要がある。人数が増えるに連れて判別の難易度も上がってくる。

**特定人物の監視時には手動でのカメラ映像の切り替えが必要：**

不審者を監視している最中に不審者が移動した場合、どのカメラの撮影範囲に移動したかを監視員が判断し、カメラを切り替える必要が生じる。

上記のような問題点により、人物の追跡は監視員への大きな負担となっている。

## 第2章 人物追跡システムについて

1人の監視員が同時に複数人の人物を監視する場合も3~4人が限界だと思われる。そのため、近年の映像監視システムでは、映像の録画だけでなく、画像処理を行い、付加価値を向上する試みが行われている。付加価値としては、顔認識<sup>18)19)</sup>や人物抽出、行動分析などと様々なものが入り入れられている。これらは実用化され、更なる性能向上の研究がされているが、以下のような問題を抱えている。

1. 画像からの人物認識は誤検出や見逃しが発生しやすい。
2. 画像処理を行うサーバに負荷が集中する。
3. サーバがダウンした場合、人物追跡が止まってしまう。
4. 一部の回線が切れた場合でも監視不可になりやすい傾向がある。

人物がGPS発信機やビーコンのような特定のデバイスを所持しておらずとも対象の人物を検出できる可能性があるが、一方でカメラの死角に人物が移動したり、カメラの設置環境、特に時間帯や照明などの明るさに影響してカメラへの人物の映り方が大きく異なるため、人物を正しく検出するためには数多くの課題が存在する。また、顔画像から人物を認識し追跡するシステム<sup>20)</sup>では、個人のプライバシーの問題<sup>21)</sup>も懸念されている。

### カメラを用いたシステムにおける人物の特徴情報の抽出と同定処理

カメラの場合は、動画情報あるいは画像情報から人物の特徴を抽出する。

人物対応付けの特徴量としては、画像に含まれる勾配や色などの低次の特徴を抽出する手法<sup>22)23)24)25)</sup>があり古くから利用されてきた。

また、特徴パラメータの抽出においては、柿内らの研究<sup>29)</sup>においてSIFT法<sup>26)27)</sup>や移動体の検出法<sup>28)</sup>を参考に簡易的な独自のパラメータ抽出手法を考案しており、事前に符号化用学習ファイルを輝度勾配による特徴量を示す特徴空間においてベクトル量子化し、代表ベクトルコードを割り当て作成する。特徴パラメータ抽出処理では、学習ファイルを用いてサンプリングデータを符号化し、さらにHSVによる色コードを加えて特徴パラメータを生成する。これにより色情報を含んだベクトルデータを特徴パラメータとして作成する。この特徴パラメータから尤度を求め、尤度がある一定の基準値を満たした場合、ターゲットと判定する。この手法での特長パラメータの抽出と判定にかかる時間は100~200ミリ秒程度であるが、どの条件下でも画像から十分な認識ができるとは言い難い。

一方で、性別や年代などの属性や人の歩き方と言ったように人間が意味を理解できる高次の特徴を抽出する手法<sup>30)31)32)48)34)</sup>が提案されてきた。特に人物属性の特徴は生体認証の研究分野からも期待されているが、人物属性の特徴は精度よく抽出することは難しいことが知られている。そのため近年では高次の特徴を中間的に表現し人物対応付けに利用する手法<sup>35)36)</sup>が注目されている。しかし、これらの既存手法は以下で述べるように、人が人を探す際に用いる手がかりと比較すると人物対応付けの特徴量として十分とは言えない。そこで、例えば警察庁が公開している情報<sup>37)38)39)40)</sup>のような高次の属性の組合せを用いてカメラ間の人物の対応付けを行う手法<sup>41)</sup>が提案されている。これによりカメラ間の人物同定処理の性能を向上することはできているが、人物を特徴を抽出するためには多量の計算が必要であり、分散構成で人物をリアルタイムに同定しながら人物追跡を行うには解決しなければならない課題がある。

これらの研究成果からも、カメラ画像から人物の同定処理を確実に行うための技術課題は高いと言える。

### 2.2.2 GPS を用いた監視システム

GPS(Global Positioning System)は、衛星を利用して現在位置を測定するシステムのことである。複数のGPS衛星からの電波を受信して現在位置を計算する。近年は測位手法が発達し、時刻同期の精度向上や他のセンサの活用により高度や携帯電話の基地局の位置情報を用いることにより、大きく精度が向上している。主として地図と組み合わせて自身の現在位置と周辺情報を把握する用途で用いられるが、人物の追跡にも用いられており、アメリカ合衆国などでは性犯罪を犯した前歴者にGPS受信機を内蔵した発信機を取り付け、監視することを行っている<sup>42)</sup>。日本の警察でも捜査対象車両にGPS発信機を取り付けた調査を行うことがある。

しかし基本的には屋外で大まかな所在地の把握に用いるシステムであり、屋内を移動する人物の追跡には向かない。また、人物の位置を取得するためにはGPS受信機を追跡対象者が保持する必要がある。



### GPS を用いたシステムにおける人物の特徴情報の抽出と同定処理

GPS を用いたシステムでは、追跡対象となる人物に持たせる端末が衛星の電波を受信し測位する。測位した地点の情報を電波で発信することによりその位置をシステムに通知する。一般的には携帯電話等の通信事業者の設備を利用して無線ネットワークで通知することが多い。例えば、携帯電話やスマートフォン自体が発信機となる。携帯電話やスマートフォンは個体の番号を識別できるようになっているため、その番号とその端末の所持者を結びつけることで人物の位置を識別する。このため、端末を誤認識することはないが、正確には人物の位置ではなく端末の位置を識別するため、端末を別の人物が所持した場合には本来追跡したい人物とは異なる人物の位置を示すことになる。

### 2.2.3 ビーコンを用いた監視システム

ビーコンは電波を発信する発信機である。昔から地上にある無線局が自身の位置を示すために電波を発信し、それを航空機や船舶が受信することにより位置の情報を取得するために用いられてきた。近年は Bluetooth Low Energy(BLE)などの低消費電力の近距離無線技術が発達してきたことにより、移動体に小型のデバイスを取り付けて受信機により検出することでビーコンの位置を把握することができるようになってきた。例えば、所持品にビーコンタグを取り付けて落としたときにその位置を調べたり<sup>43)</sup>、子どもや高齢者にビーコンを持たせ、駅や学校に設置した専用の受信機を介して位置情報を取得するなどである。

しかし、ビーコンを所持している人物でないとその位置を把握することができない。また、BLE を用いたビーコンの場合は発信電力が弱いため、その位置を取得するためには多くの受信機を設置しなければならない。

### ビーコンを用いたシステムにおける人物の特徴情報の抽出と同定処理

ビーコンを用いたシステムも、GPS の場合と同様に追跡対象となる人物にデバイスを持たせて追跡を行う。ビーコンの場合は、受信機が検出したビーコンの ID をターゲットの同定に使用する。ビーコンの ID は個体ごとにユニークな ID を付与されているため、ビーコンの ID を取得することにより容易に端末を識別できる。

GPSと同様にビーコンを意図しない人物が所持した場合には本来追跡したい人物とは異なる人物の位置を示すことになる。また、電波を使うセンサを用いる場合、複数のセンサで同時に検出することが有り得るため、センサから受信した電波の強度やRSSI等の情報を取得し、閾値から人物の検出を行う。これにはBluetoothを用いたビーコン<sup>44)45)46)47)48)</sup>の技術を利用することができる。ビーコンの場合は、電波はビーコンから発信するため、発信周期によっては人物を見逃す可能性がある。iBeaconに準拠したセンサでは、発信周期が0.1秒であるため、発信周期の問題で人物を見逃す可能性は少ない。他のBluetoothセンサでは、5秒程度の周期を持つものもあり、この場合はセンサの設置場所や検出の閾値などの調整が必要になる。ビーコンの発信周期は、そのビーコンが持つ電源やバッテリーの持続時間に関わってくるため、発信周期を単純に早くすれば良いとは限らない。ビーコンを用いた人物追跡においては、追跡エージェントはビーコンIDを用いて人物の同定処理を行い追跡処理を実装する。

ビーコンについては、追跡したい人物がビーコンを所持している必要があり、当然ビーコンを持たない人物は追跡できない。しかし近年は、スマートフォンやこどもの見守りタグの普及してきていることから、このような電波を発信するデバイスから人物を追跡する環境は整いつつある。

### 2.3 センサの配置と人物の追跡

人物追跡システムでは、人物を追跡するために複数のセンサで検出した人物の同定が必要になる。そのためにはセンサの配置と、センサ間でどのようなデータを共有するかが重要となる。

人物を追跡する手法については多くの手法が提案されている。森ら<sup>49)</sup>は同一のエリア内に撮影範囲が重なるように複数のカメラが設置された環境下で、非同期に動作する複数のカメラから得られる観測情報を統合することで人物を追跡する手法を提案している。中澤ら<sup>50)</sup>は視覚エージェントにより取得された複数の人物の特徴情報を組み合わせるための仕組みを提案している。また、浮田<sup>51)</sup>は視覚エージェントによる情報交換を効率化するためのシステムを提案している。これらの研究はカメラの撮影範囲が重なっていることを前提としており、カメラが離散的に配置されている状況に対応できない。谷澤ら<sup>52)</sup>は人物の移動とネットワー

ク上の論理的な移動を結びつけ、人物の移動に伴ってネットワーク上をモバイルエージェントが移動することで、その人物に特化したサービスを提供するシステムを提案している。しかし、移動先の決定はモバイルエージェントに任されており、その仕組みについては述べられていない。田中ら<sup>53)</sup>もモバイルエージェントによる人物追跡手法を提案しているが、人物の移動先の特定方法については検討していない。武村ら<sup>54)</sup>は人物の詳細情報を効率よく獲得するための移動エージェントの経路計画問題を提案しているが、人物の移動速度から経路を補正していくといった手法であり、人物を捕捉し続けることができるような環境が必要になる。

既存の監視システムの運用においては、コストを抑えるためにより少ないセンサで確実に人物の位置を把握できるようにセンサを配置する工夫が行われている。例えば、カメラで撮影する場合は出入口や改札のような人が確実に通る場所が映るようにカメラを設置するといった工夫である。そのため、一般の監視システムでは、センサは離散的に配置されていることが多い。人物追跡システムにおいてもセンサの数はシステムのコストに直結するため、なるべく少ないセンサで確実に人物を追跡できることが望ましい。

本稿では、センサが離散的に配置された既存の監視システムの環境を想定して、人物を追跡するためのアルゴリズムを検討する。

### 2.4 集中型のシステムと分散型のシステム

人物追跡システムはその性質上、各場所に配置されたセンサの情報を管理する必要がある。特に追跡範囲が1つのオフィス内に限られる場合は、情報を1つのサーバに集中してセンサと人物の情報を管理する方が簡単である。更に、人物の特徴情報もサーバに集積して比較する方が人物の識別の点でも有利と考えられる。しかし、追跡対象の人物や管理対象のセンサが増えるほどサーバへのアクセスが増え、サーバの負荷が大きくなる。近年はクラウドコンピューティング技術の発展により、複数の計算資源（コンピュータ）を1つのサービスとして見せることが可能になってきたため、単純に計算負荷だけを見た場合の問題は解消されつつある。

追跡したい範囲が広がり、対象の人物の数も大きくなると、計算負荷だけの問題ではなくなってくる。人物追跡システムはセンサが人物を追跡する度に、その位

## 第2章 人物追跡システムについて

置情報が更新されるため、サーバやクラウドへのアクセスが頻繁に発生する。極端な例を挙げると、日本一乗降客数の多い新宿駅<sup>55)</sup>に人物追跡システムを設置する場合は、1日平均350万人分の人物検出の通知が上がってくることになる。この人数のデータを適切に処理することはとても難しい（ただし自動改札機はこの人数分のデータを常に処理し続けている）。センサも多く設置されるため、ビーコンのようなIDだけならそれほどのデータ量でもないかもしれないが、カメラ画像の照合を考えると膨大なデータアクセスが発生することになる。計算負荷だけでなくネットワークも含めたトータルの設備強化が必要となる。特に追跡対象人物のデータを格納するストレージとリアルタイム性を維持するための各センサ間の通信ネットワークの品質は重要な要素である。

分散型のシステムにおいても、例えば新宿駅の改札口のように、場所によっては1つのセンサが同時に検出する人物の数がとても大きなものになると予想されるため、設備の増強の必要が全く不要とはならない。しかし、全ての情報を中央のデータサーバに抱えることによって、サーバやインフラであるネットワークを過大に増強する必要がなく、ある程度分散することができる。ある程度的人数であれば、スマートフォンレベルの処理能力でも人物の識別処理は可能であるため、最大数を追跡するための処理能力を必要とする設備を用意する必要なく、低コストでシステムを構築することが可能となる。計算機を多数配置し、配線も考慮しないといけないなどのコストアップの問題もあるが、センサは現場設置が前提であり、IoTの普及とともにスマートフォンレベルの処理能力を持つセンサと併せて設置できる小型計算機が普及してきていること、配線もWi-Fiなどの無線ネットワークも普及していることから、電源さえ確保できれば計算機をセンサの台数分設置することも現実的になりつつある。

本稿では、IoTを想定し小型計算機によるセンサネットワークの構築を考えて、その上に分散型の人物追跡システムを稼働させることを検討する。次節に提案する人物追跡システムを説明する。

## 2.5 提案する人物追跡システム

センサを用いた人物追跡システムにおいて、監視員の負荷を軽減するためにモバイルエージェントによる人物追跡システムを提案している。本システムでは追跡対象人物の追跡はモバイルエージェントによって行われる。モバイルエージェントによる追跡対象人物の追跡状態が監視モニタに一括表示され、監視員は複数人の人物の追跡状態を確認することができる。

本システムでは中央サーバの存在しない分散型のシステムにおいて特定のルールに基づき移動するエージェントが人物を追跡する。このため、中央サーバのスペックやネットワーク上の装置や追跡対象の人物の数を気にする必要がない。また、センサの検知範囲の変化や監視ノードの故障などの環境の変化に対して監視ノードが持つ隣接関係を局所的に更新し、更新された隣接関係を元にエージェントが人物を追跡することで環境の変化に柔軟に対応することができる。エージェントは各監視ノードが持つ局所的な情報に基づいて移動先を決定しシステム内を移動する。

この人物追跡システムは以下の利点を持つ。

### 設備の故障時における追跡継続性が高い

監視ノードや監視モニタが一時的に稼働できない状況でも、エージェントが自律的に移動することで追跡を継続することができる。

### システムの拡張や環境の変化に柔軟に対応できる

エージェントはシステム全体の経路を把握しておらず、各監視ノードが持つ局所的な情報に基づいて移動する。そのため、システムの拡張や環境の変化に柔軟に対応できる。

### 2.5.1 対象とするシステム

図 2.1 に対象とするシステムについて示す。対象システムは施設の屋内においてあらかじめ登録された人物を常に追跡し、その位置を示すものである。人物の特徴を受付で登録し、登録した人物の位置は監視モニタにて地図上で閲覧することができる。屋内での人物追跡を想定していることから、センサにはカメラまたはビーコンを用いることとした。

## 第2章 人物追跡システムについて

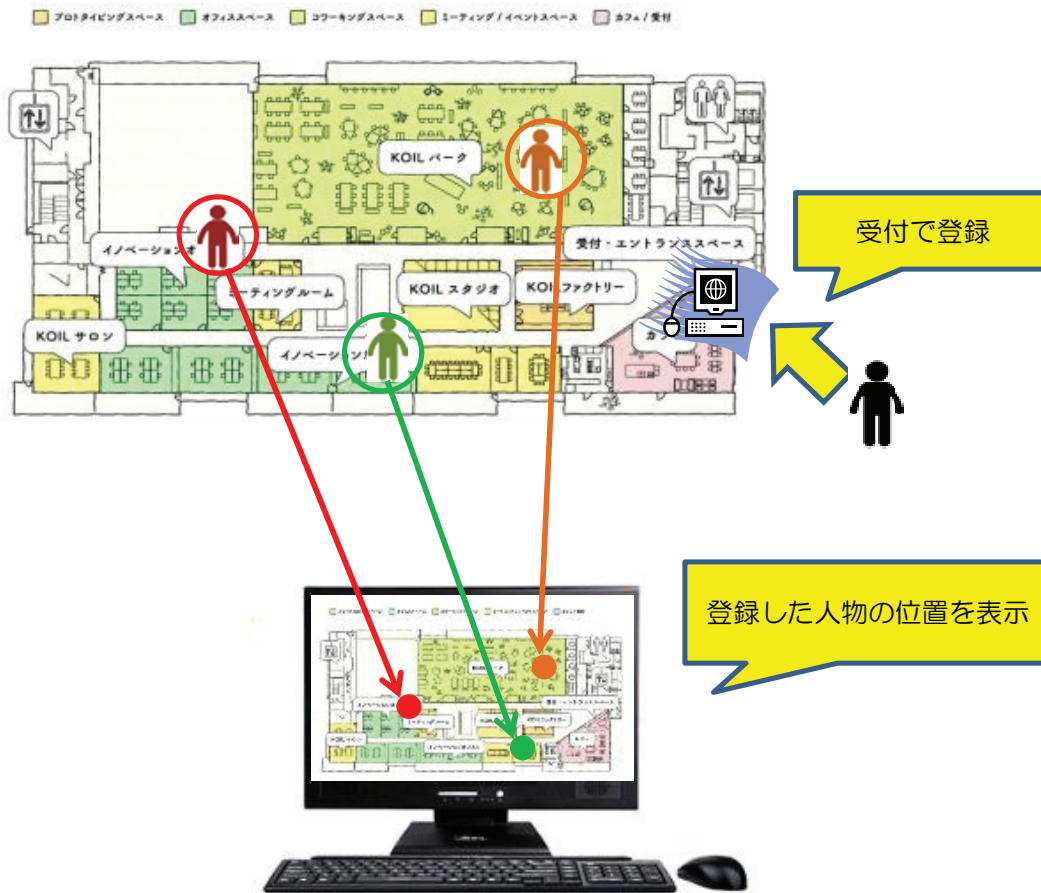


図 2.1: 提案システム

監視画面は図 2.2 のイメージとする。地図上に追跡対象の人物の位置が表示される。

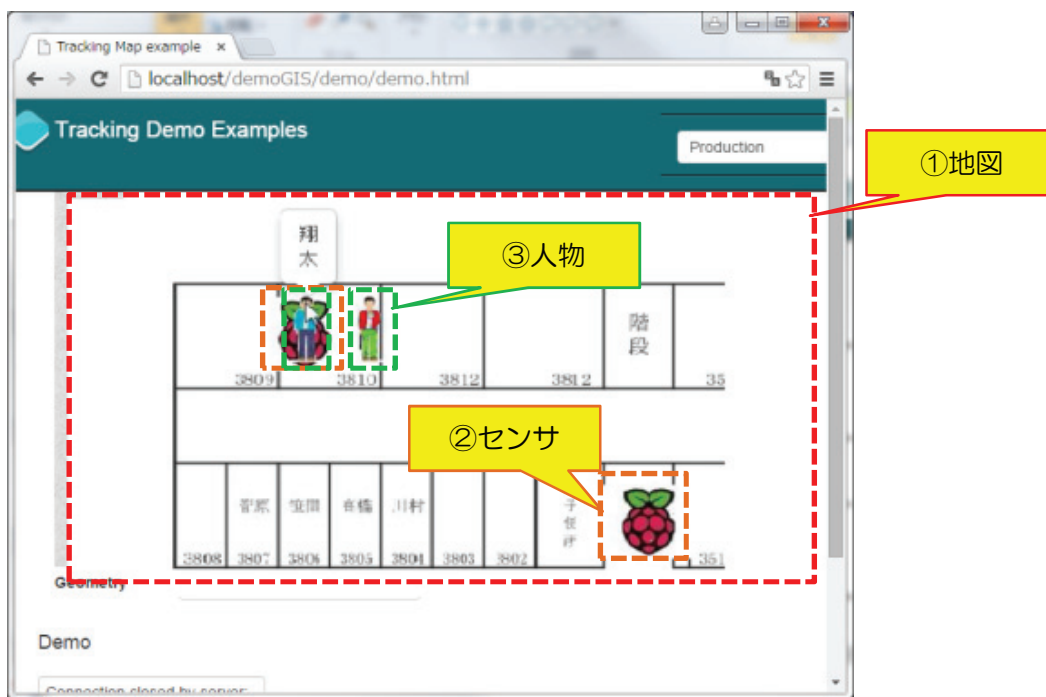


図 2.2: 監視画面イメージ

## 2.5.2 システム構成

図2.3 にシステムの概要を示す。2.4節で述べたように、本稿ではセンサの設置場所に小型計算機を配置して人物の識別処理を行う、分散型のシステムを想定する。

本システムは、センサ、監視ノード、モバイルエージェント、監視モニタから構成される。本システムではシステムを構成する装置を集中的に管理するサーバの存在を仮定せず、センサと監視ノードを任意に追加・削除できる。センサは離散的に監視エリア内に設置され、複数のセンサごとに監視ノードが準備される。監視ノードは、モバイルエージェント実行環境とセンサから受け取った情報を解析する機能、センサの隣接関係を計算する機能を持つ。モバイルエージェントは追跡対象人物ごとに生成され、その人物の特徴情報を管理する。監視モニタはエージェントの状態を表示する機能を提供し、監視員がエージェントの状態を確認することで複数の追跡対象人物の場所を確認することができる。

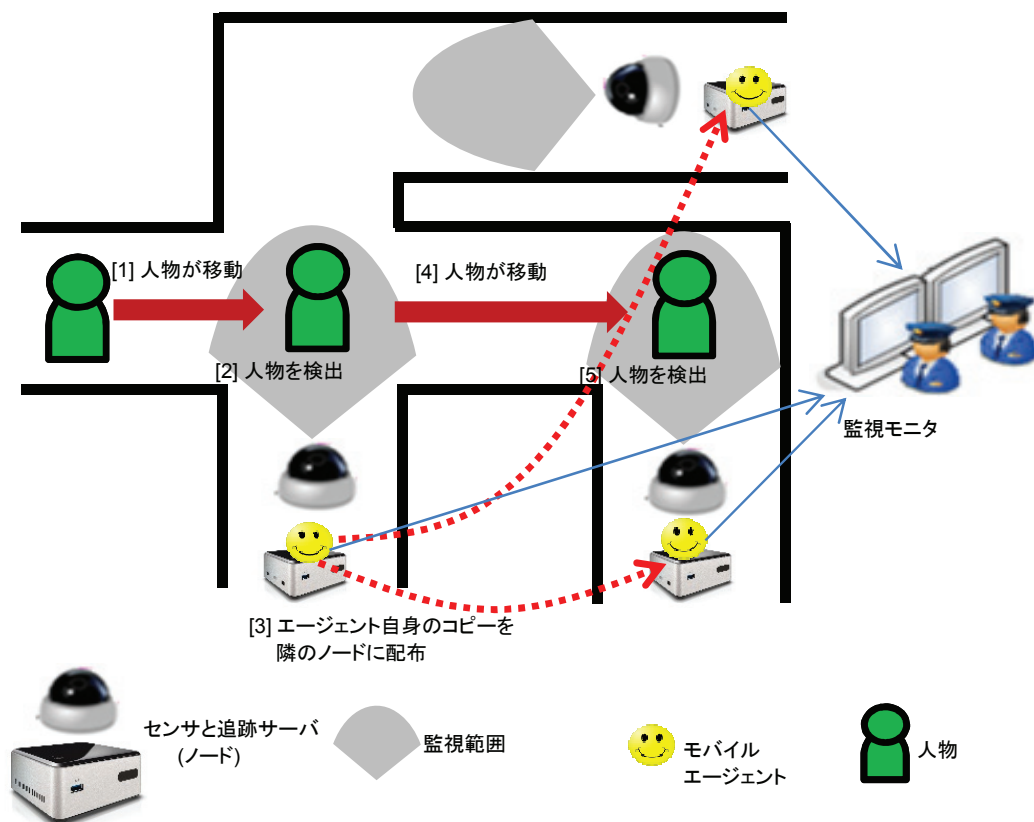


図 2.3: システム構成の概要



## 第2章 人物追跡システムについて

追跡対象の人物の位置を把握したい人物を監視員として定義する。このシステムをオフィスや病院等で運用するとして、監視員は専門の警備員に限らず、職員の誰もが監視員になりうる。そこで、それぞれの職員のPCに標準搭載されているWebブラウザで人物の位置を表示できるように、Webブラウザでの表示を実現することとした。そのため、本システムではセンサと一体化した小型計算機（ノード計算機）は常に人物の位置情報をサーバに通知し、監視員がいつでも人物の位置を知りたいときにWebブラウザでサーバにアクセスして人物の位置を表示することとした。

監視員はWebブラウザでフロア内の人物の位置を知ることができる。そのための仕組みを図2.4に示す。まずセンサと一体となった監視ノードに配置されるノード計算機が人物を検出すると、どのノードでどの人物を検出したかをサーバに通知する。サーバには地図と人物とノードの情報を持っており、監視端末からのアクセスによりその情報を監視端末へ渡し、監視端末に表示される地図上にノードと人物の位置を表示することができる。

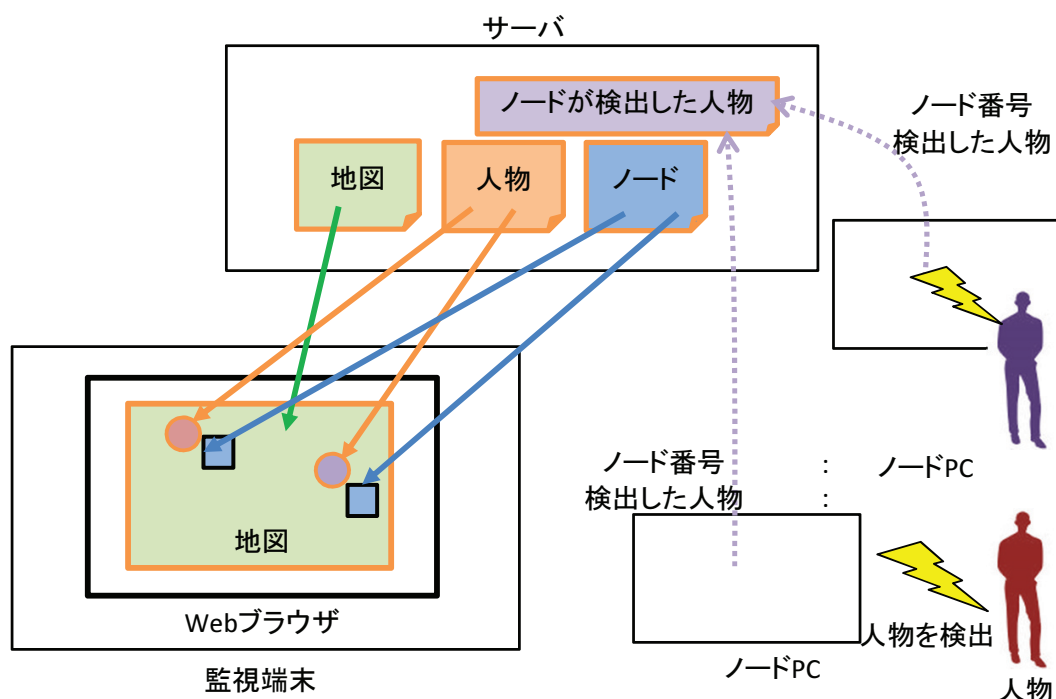


図 2.4: 人物を検出し監視画面に表示する仕組み

地図表示用として、オープンソースソフトウェアである OpenLayers<sup>56)</sup> を使用する

る。これは地図の拡大や縮小、スクロール等の GoogleMap<sup>57)</sup> のように Web 画面で地図を扱う機能を提供する。地図は地図データを扱うことが可能だが、本システムでは屋内の画像化したフロア図に対して、センサの位置などを座標として与えて表示することとした。

また、地図画面にセンサで検出した人物の位置を表示するために、各センサと一緒に設置してある小型計算機から情報をサーバに通知し、同時にサーバと監視端末間で人物の位置の変化を通知するための通信手段として WebSocket<sup>58)</sup> を用いる。

### 2.5.3 システムの特徴

本稿で提案する人物追跡システムは、複数の人物を自動で同時に追跡し、分散構成で動作するといった特徴を持つ。本システムでは、追跡エージェントがターゲットを追跡し、その追跡情報はエージェント自身が保持する。監視員は、追跡エージェントが存在する追跡サーバを知ること、ターゲットがどのセンサで検出されたかを知ることができる。また、本システムは1人のターゲットに対し1つの追跡エージェントを生成するため、ターゲットを追加したい場合は追跡エージェントを新たに生成する、つまり、ターゲットの数と同じだけ追跡エージェントを生成し、実行することで、複数のターゲットを同時に追跡可能としている。また、追跡エージェントは追跡サーバ上で動作しているため、情報を集中して管理するサーバは存在せず、各センサの設置場所の追跡サーバが動作していれば、稼働中の追跡サーバ間において人物の追跡は継続される。これらの特徴によって、従来の監視システムの問題を解決することができる。

#### 本研究におけるモバイルエージェントの人物追跡への応用

本システムでは、人物の特徴を保持し人物の移動に合わせた追跡を行うために、モバイルエージェント技術を利用している。モバイルエージェントとは、ネットワークを自由に移動し、自律的にユーザの代わりまたは補助となり、処理を実行するプログラムである。モバイルエージェントは、D. B. Lange らの研究<sup>59)</sup>においても、モバイルエージェントの特徴が紹介されている。モバイルエージェントは、プログラムの雛形により基本動作が構成され、雛形を使うことで機能製作や改造の負担を軽くする特徴がある。またモバイルエージェントを使用することにより、

## 第2章 人物追跡システムについて

複数の計算機間を跨った知的処理が可能となり、ネットワークやCPUといった資源の効率的な利用や、障害時に柔軟な処理が行えるシステムが構築できる。これらの特徴は、システム運用が開始された後において、システムを構成する計算機を追加する場合などの規模の変化に対し、システム内のプログラムを変更せずにシステム規模の拡大ができることである。さらにシステム機能を増強するため、プログラムを変更する場合においても、モバイルエージェントの修正で機能の増強に対応できる。

モバイルエージェントによる人物追跡を行うシステムでは、文字通り監視員の代わりにモバイルエージェントに人物を追跡させる。モバイルエージェントは追跡対象の1人の人物の特徴情報を持ち、同じ特徴を持つ人物が検出されていないかを各センサに対してチェックする。追跡対象の人物が複数になる場合は、対応するモバイルエージェントを増やせば良い。つまり監視員を増員しなくても、モバイルエージェントを増やすことによりソフトウェアのリソースを増加させるだけで目的の人物の追跡を実現できる。

ここで課題となるのは、モバイルエージェントが人物をどのように追跡するかである。例えば、監視員が不審人物を見つけ、その人物を見逃した場合にどのように人物を探すを考える。1つの方法としては全てのカメラに対して監視員が記憶している人物が映っていないかをチェックすることである。同じことをモバイルエージェントで実現しようとする、全てのセンサが配置されている場所にモバイルエージェントを派遣あるいは巡回させることになる。これではセンサの情報を中央の管理サーバに送信して人物を探すのと大きく変わらず、システムとしては大きな処理コストの負担となる。

常に中央サーバに問い合わせはしないとすると、監視員は人物を見逃した位置から近い位置のカメラから探すはずである。そこでモバイルエージェントも最後に人物を検出した位置に近い、次に人物が出現すると思われる箇所に派遣することが処理コストの点でも望ましい。次に人物が出現すると思われる箇所とは、人物が最後に出現した位置のセンサに通路でつながっているセンサ、すなわち隣接するセンサである。

どのようにモバイルエージェントを効率よく派遣し、各センサの位置における追跡サーバの処理コストを低減できるかが本研究の課題であり、本研究では、モバイルエージェントで人物を追いかける追従手法と人物を探索する探索手法を

考案し、これらを組み合わせ導入した追跡エージェントを実装する。

本研究では、センサは必ず監視場所に設置する必要があるため、センサと安価な小型計算機をセットで設置し、中央サーバとその間のネットワークに大きなコストをかける必要がなく、設備の増設が用意かつ低コストの人物追跡システムを実現することを想定している。また、システムの適用対象としては、複数の建物や組織を跨って追跡できることを考慮する。センサの種別にも依存しない。そのため、中央で集中して情報を管理するシステムではなく、分散構成としセンサに依存しない設計とする。分散構成のメリットは、各サーバの負担を抑えられることと、任意に拡張できることである。分散構成を前提とすることで、全ての人物を全てのエリアを対象に監視するための膨大な処理コストを現実的な範囲に抑制することができる。

これらの点を踏まえてモバイルエージェント技術の特徴を応用することで、低コストでかつ拡張性に優れた人物追跡システムの実現を検討する。

### 2.5.4 人物追跡の流れ

本システムにおける人物追跡の流れを図 2.5 に示す。監視エリア内に追跡対象人物が来ると、監視ノードはその人物を追跡するエージェントが監視ノード内に存在するか確認する。存在すればそのエージェントが人物を追跡する。存在しなければその人物を追跡するためのモバイルエージェントを生成する。生成されたモバイルエージェントはその人物のセンサ情報を解析し特徴情報を取得・保持する。追跡対象人物がセンサの検出範囲から離れた場合、モバイルエージェント（親エージェント）は隣接センサを管理する監視ノードに自身の複製（子エージェント）を派遣する。親／子エージェントはセンサ情報を定期的に解析し、追跡対象人物が検出されているか判定する。検出されていた場合には追跡対象人物が見つかったことを親エージェントに知らせる。親エージェントはすべての子エージェントに追跡対象人物が発見されたことを通知し実行を停止する。通知を受け取ると、追跡対象人物を発見した子エージェントは親エージェントに変化し、それ以外のエージェントは実行を停止する。これを繰り返すことで、追跡対象人物の移動に併せてモバイルエージェントが監視ノードを移動し追跡を継続する。これにより複数台のセンサに跨った人物の自動追跡を実現する。

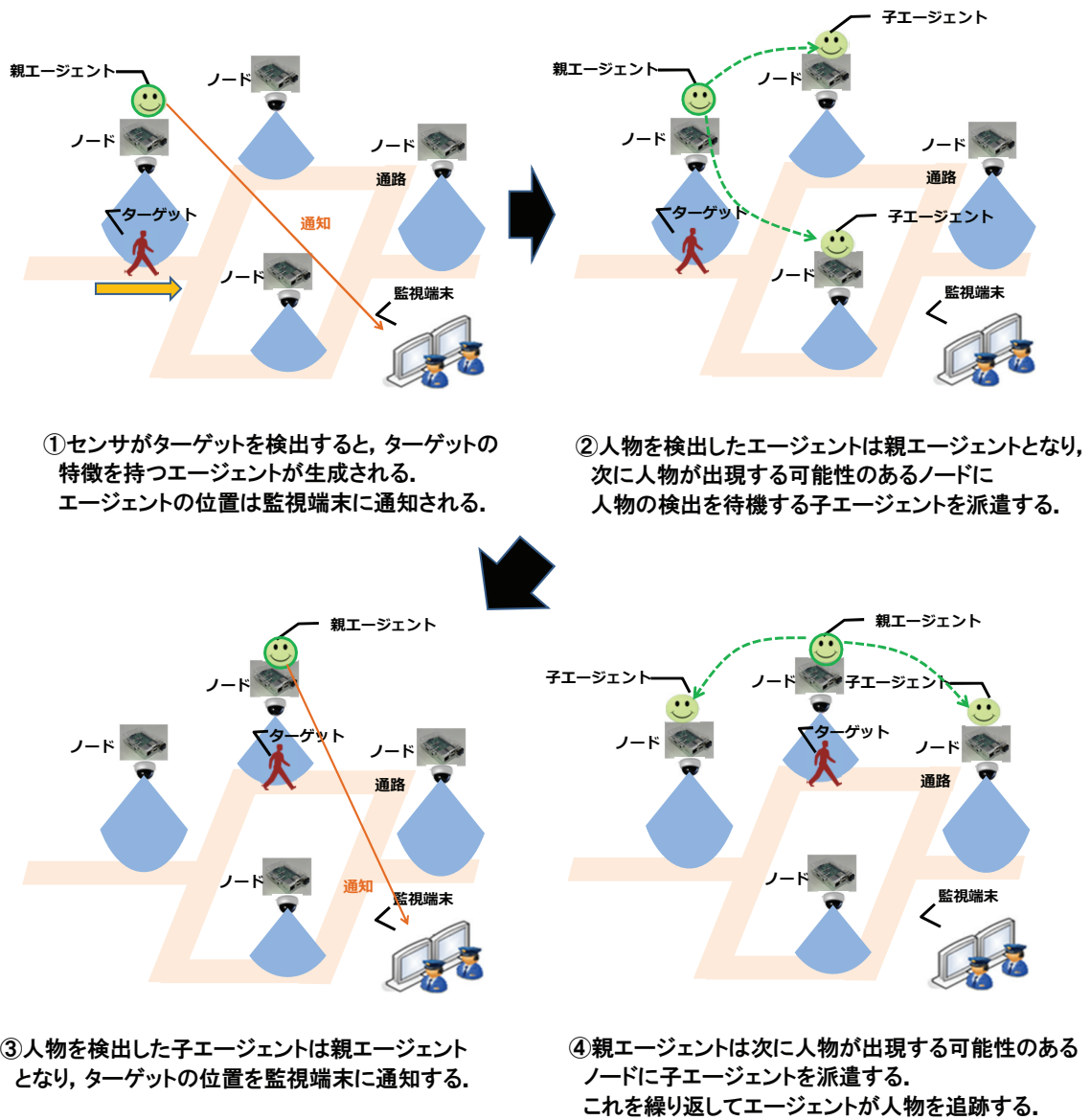


図 2.5: 人物追跡の流れ

## 2.6 人物追跡システムの実装と実験

本節では、前節に述べたモバイルエージェントを用いた人物追跡処理を実装し実験を行った。センサと一緒に配置する小型で安価な計算機として Raspberry Pi を使用し、ビーコンを所持した人物を追跡する実験を行った。実装について 2.6.1 節で説明し、実験について 2.6.2 節で説明する。

### 2.6.1 人物追跡システムの実装

図 2.6 に示す構成で人物追跡システムを実装した。システムの構成は、センサ台数と同じ数設置されるノード計算機（追跡サーバと呼ぶ）、ノード計算機からの情報を受け付けて人物の位置を把握するサーバ（管理サーバ）、職員が人物の位置を目視で確認するための監視端末の 3 種類の計算機が存在する。

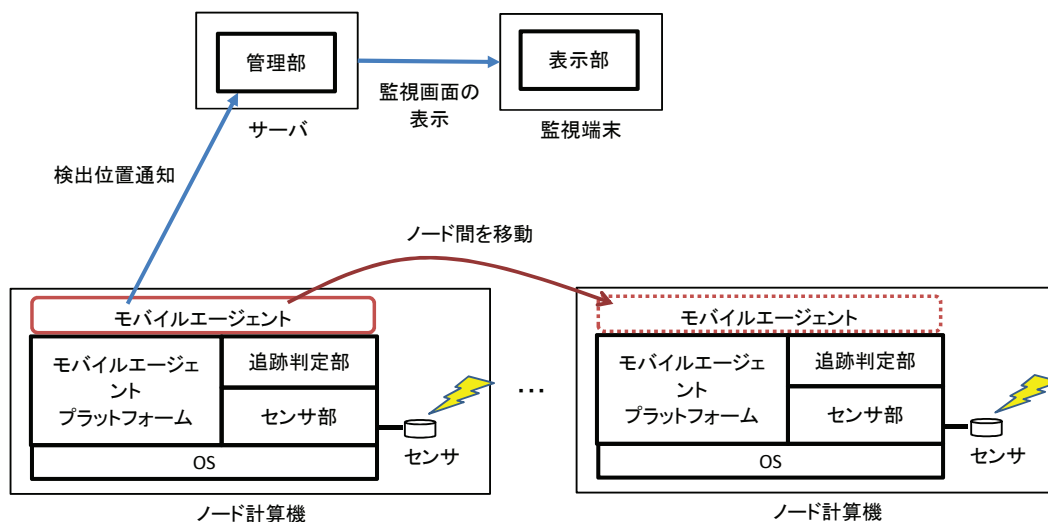


図 2.6: 人物追跡システムの実装

ここで、図 2.6 においてサーバは管理部、監視端末は表示部の機能ブロックをそれぞれ持つ。ノード計算機（追跡サーバ）におけるシステムを構成する要素を以下に示す。

- OS

ノード計算機の最下層のレイヤである。本実装では 1 ボードコンピュータ

である Raspberry Pi をノード計算機として使用するため、Linux ベースの Raspbian OS を使用している。

- モバイルエージェントプラットフォーム  
Java 上で動作する OSGi<sup>60)</sup> フレームワーク準拠のサーバを用いて、更にその上にモバイルエージェント実行環境を実装している。人物追跡用のモバイルエージェントが本プラットフォーム上で実行される。
- センサ部  
Bluetooth ビーコンまたはカメラからのデータを取得する。これらのデバイスは Java プログラムから直接アクセスできない場合があるため、通信を介して Java プログラムとのデータの受け渡しを行っている。
- 追跡判定部  
センサ部とモバイルエージェントの間をつなぐ役割を持つ。マップの情報をもち、人物の検出を判定し次にモバイルエージェントを派遣すべき移動先を計算する。
- モバイルエージェント  
人物の特徴情報を持ち、他のノード計算機に移動して同じ特徴情報を持つ人物を追跡するエージェントプログラムである。

本実験ではノード計算機として図 2.7 に示す Raspberry Pi を使用した。大きさは 9cm × 6cm のケースに収まる小型の計算機である。Raspberry Pi は複数のモデルが存在し、今回は Raspberry Pi 2 と Raspberry Pi 3 を用いた。それぞれのスペックを表 2.1, 表 2.2 に示す。

Raspberry Pi 3 には Bluetooth と Wi-Fi 機能が内蔵されているが、Raspberry Pi 2 には内蔵されていないため、Bluetooth Low Energy に対応した Bluetooth USB ドングルと Wi-Fi アダプタを装着した。

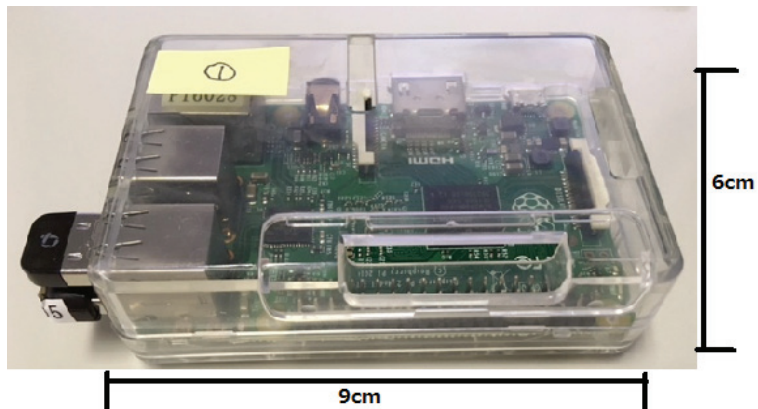


図 2.7: Bluetooth と Wi-Fi アダプタを装着した Raspberry Pi 2

表 2.1: Raspberry Pi 2 Model B のスペック

CPU	ARM Cortex-A7 900MHz
OS	Raspbian GNU/Linux 7.8
Bluetooth	バージョン 4.1 (PLANEX BT-Micro4 アダプタ装着)
Wi-Fi	IEEE 802.11 b/g (BUFFALO WLI-UC-GNM2 アダプタ装着)
Java	JRE 1.8.0

表 2.2: Raspberry Pi 3 Model B のスペック

CPU	ARM Cortex-A53 1.2GHz
OS	バージョン Raspbian GNU/Linux 8.0
Bluetooth	バージョン 4.1
Wi-Fi	IEEE 802.11 b/g/n
Java	JRE 1.8.0



本実験では、エージェントによる人物の追跡動作の確認を目的として、より人物の識別を確実にするためにカメラではなく iBeacon<sup>61)</sup> タグを追跡対象者に所持してもらい実験を行った。

iBeacon は Bluetooth<sup>62)</sup> を利用した通信技術であり、タグ情報を発信する。追跡サーバがタグ情報を受信することで追跡を行う。人物が所持するビーコンのタグ ID を用いて識別を行うため、人物（タグ）を誤認識することはない。ビーコンタグは図 2.8 に示す Aplix 社製 MB002 を使用した。スペックを表 2.3 に示す。



図 2.8: iBeacon タグ (MB002)

表 2.3: iBeacon タグ (MB002) のスペック

型番	Aplix Beacon MB002 Ac
Bluetooth	バージョン 4.1
電源	コイン型リチウム電池 (CR2032) 1 個
外形寸法 (W × D × H)	35 × 65 × 11mm
重量	約 15g (電池除く)

ビーコンタグは Bluetooth Low Energy に対応したタグであれば利用可能である。実験において、より軽量の図 2.9、表 2.4 に示すビーコンタグを使用して追跡が可能であることを確認している。ビーコンタグは機種により送信電力の強度と発信周期が異なるため、2.6.2 節の実験では使用するビーコンタグを MB002 に統一して実験を行った。

実装では、人物が今検出されているノード計算機に存在するエージェントを親エージェント、次に人物が出現する可能性のあるノード計算機に派遣されているエージェントを子エージェントと呼ぶ。動作確認を容易にするために、親/子エー



図 2.9: ビーコンタグ (MAMORIO)

表 2.4: ビーコンタグ (MAMORIO) のスペック

名称	MAMORIO
Bluetooth	バージョン 4.0
電源	内蔵リチウム電池
外形寸法 (W × D × H)	35.5 × 19 × 3.4mm
重量	約 3g

エージェントが各ノードに存在するかどうかを視覚的に判別できるようにするために、LEDを配線した基盤を作成した。作成した回路図を図 2.10 に示す。LED は 10 個存在する。人物の検知を行うセンサとして赤外線センサも取り付けられることも可能である。そのため回路図には赤外線センサの記載があるが、今回の実験では使用していない。

エージェントがどこに存在するかログを確認しなくても視覚的に確認することを可能とするために、エージェントが追跡サーバに到着した際に LED 点灯プログラムを起動し、受け取った親/子エージェント情報から親エージェントであれば点灯、子エージェントであれば点滅することとした。

図 2.11 に LED を取り付けした Raspberry Pi を示す。この LED 基板を取り付けた Raspberry Pi を構内に設置し簡易実験を行った。

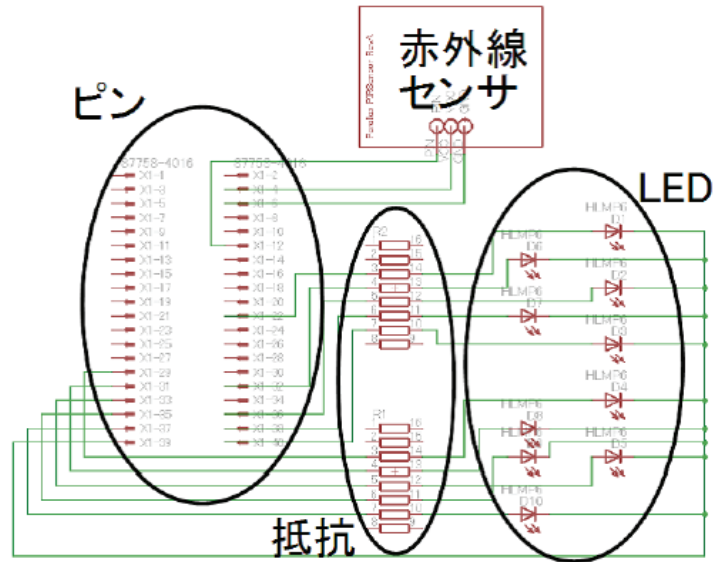


図 2.10: LED 点灯回路の回路図

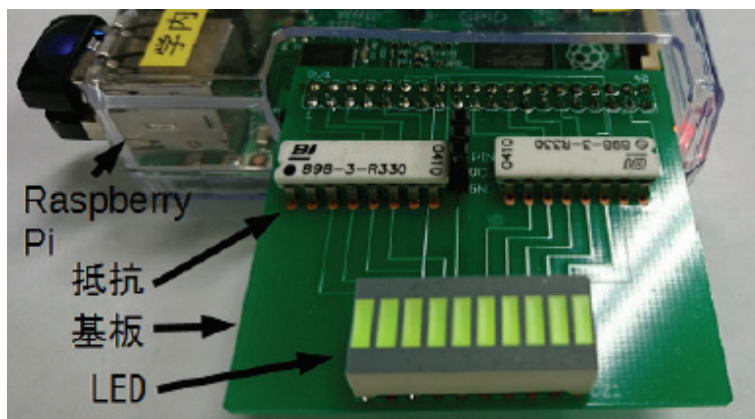


図 2.11: LED 基盤を搭載した Raspberry Pi

### 2.6.2 モバイルエージェントによる人物追跡の実験

前節で述べた実験機材を用いて簡易的な追跡実験を実施した。本実験では、使用するビーコンタグを Applix 社製 MB002 に統一して実験を行った。あるセンサが人物を検出したときに、通路の情報から次に人物を検出する可能性のあるセンサを隣接関係を持つセンサとして定義する。本実験では各ノードが持つ隣接関係に従ってエージェントがノード間を移動しながらビーコンを持つ人物を追跡できることを確認した。

#### 簡易実験

エージェントの動きを確認するために図 2.12 に示す簡易マップを使い、実験を行った。黒の点に監視ノードとしてセンサを搭載した Raspberry Pi が設置しており、ノード同士は図 2.12 の矢印で示す隣接関係を持つ。各ノードの隣接関係を表 2.5 に示す。

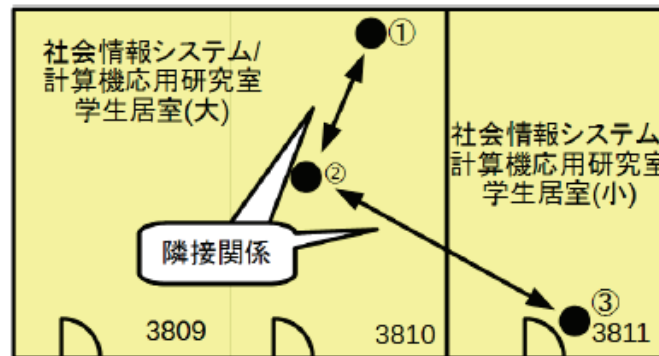


図 2.12: 簡易実験に用いたマップ

表 2.5: 簡易実験における監視ノードの隣接関係

ノード番号	隣接ノード
1	2
2	1, 3
3	2

タグを所持する人物に図 2.12 に示す番号順に移動してもらい、エージェントの

動きを確認した。1番の監視ノードにいる時には1番のノードが点灯し、2番が点滅した。人物が2番、3番に移動すると、移動に合わせて親エージェントがノードを移動していることが確認できた。

### 範囲を広げた追跡実験

簡易実験から範囲を広げ、4階建ての建物4フロア（鳥取大学の智能情報工学棟，社会開発システム工学棟）に跨って実験を行った。監視ノード（Raspberry Pi）を9台準備し，図2.13に示す監視マップの番号の位置にノードとして設置した。ノード番号1と2は室内に設置し，残りは廊下に設置した。建物内での追跡を考え，分岐が発生する階段を中心に設置した。各ノードの隣接関係は表2.6の通りである。

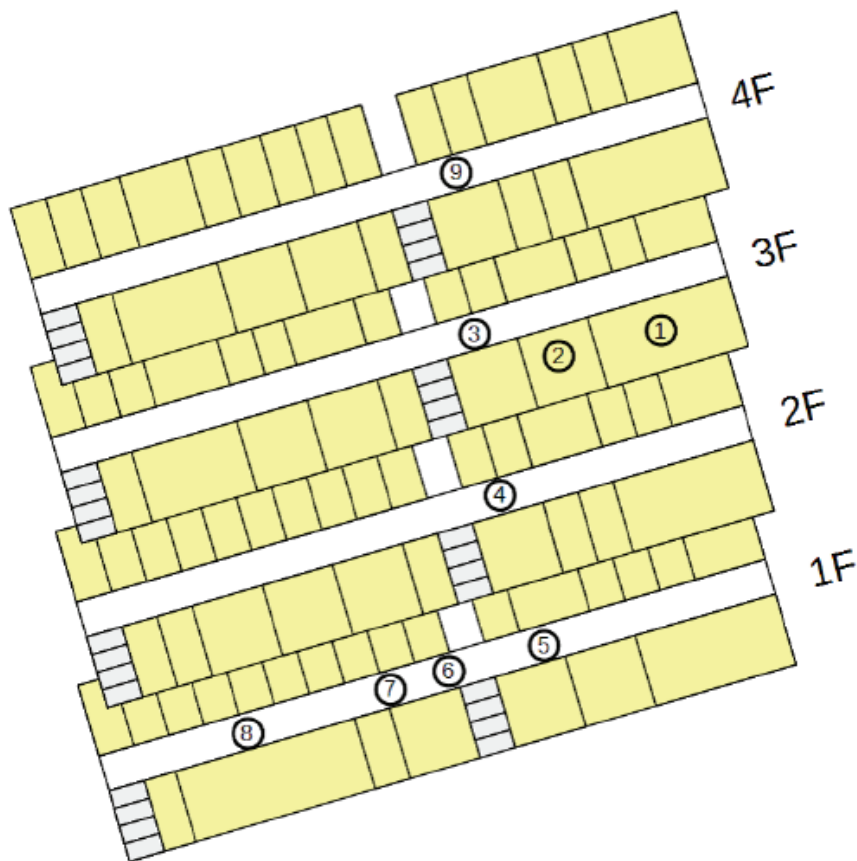


図 2.13: 実験に用いたマップ

表 2.6: 監視ノードの隣接関係

ノード番号	隣接ノード
1	2, 3
2	1, 3
3	1, 2, 4, 5, 6, 8, 9
4	3, 5, 6, 8, 9
5	3, 4, 6, 8, 9
6	3, 4, 5, 7, 8, 9
7	6, 8
8	3, 4, 5, 7, 9
9	3, 4, 5, 6, 8

追跡対象人物にはビーコンタグを所持してもらい、監視マップを任意に歩いて移動してもらった。追跡対象人物は移動時に通過したノード番号と通過時刻を記述し、それを実際に人物が移動した経路を示す正解データとした。正解データとシステムに記録されるモバイルエージェントの移動ログとを比較し、人物の移動に伴いエージェントが正確に追跡できているかの確認を行った。実験は人物にエージェントの追従を意識して移動してもらった場合（ケース1）と、追従を意識せずに移動してもらった場合（ケース2）の2つのケースで実験を行った。この2つのケースの違いは表 2.7 に示す通りである。

表 2.7: 簡易実験における追跡対象人物の移動方法

項目	エージェントの追従を意識して移動 (ケース1)	エージェントの追従を意識せずに移動 (ケース2)
移動速度	LEDの点灯を確認しながらゆっくり移動する。	LEDの点灯は意識せずに通常の方法で移動する。
タグの持ち方	電波を遮らないように持つ。	タグをポケットに入れる。

追従を意識して移動した場合の人物の移動データとエージェントの追跡結果を重ね合わせた結果を図 2.14 に示す。

縦軸はノード番号を示し、横軸は経過時間を示す。追従を意識して遅い速度で

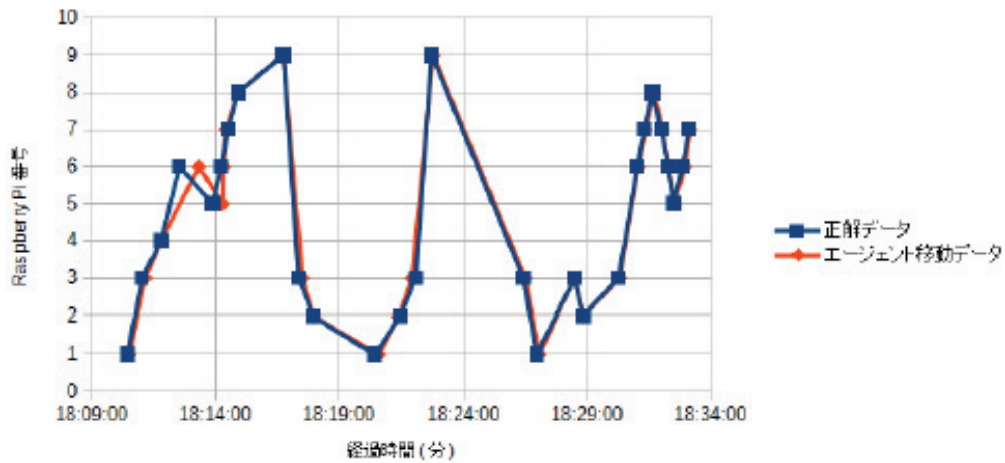


図 2.14: 人物の移動とエージェントの追跡結果 (追従を意識して移動した場合)

歩いてもらう実験では、人物にはビーコンタグを手に持ち、親エージェントを表す LED の点灯が起こるまでノードの近くに待機してから移動してもらった。人物は 1 番のノードから移動し、それから 3 番、4 番、6 番と移動した。

図 2.14 を見ると、人物の移動に追従してエージェントがノードを移動していることが分かる。次に、人物を検知したノードと隣接するノードに子エージェントを配布先を示した図を図 2.15 に示す。図 2.14 の最初の 5 分のみを示している。

図の三角で示した点が子エージェントを配布したノードである。図に示す同時間列に並ぶ子エージェントの点は、その点の一つ前のエージェントが隣接するノードに配布した子エージェントである。一つ前のエージェントが隣接するノードに配布した子エージェントが存在するノードに人物が移動しているため、正しい隣接関係に沿って追跡ができています。

エージェントによる追跡を意識せずに移動した場合 (ケース 2) の実験では、ビーコンタグを服のポケットなどに入れてもらい普段歩く早さで歩いてもらった。ケース 2 における人物の移動データとエージェントの追跡結果を重ね合わせたグラフを図 2.16 に示す。

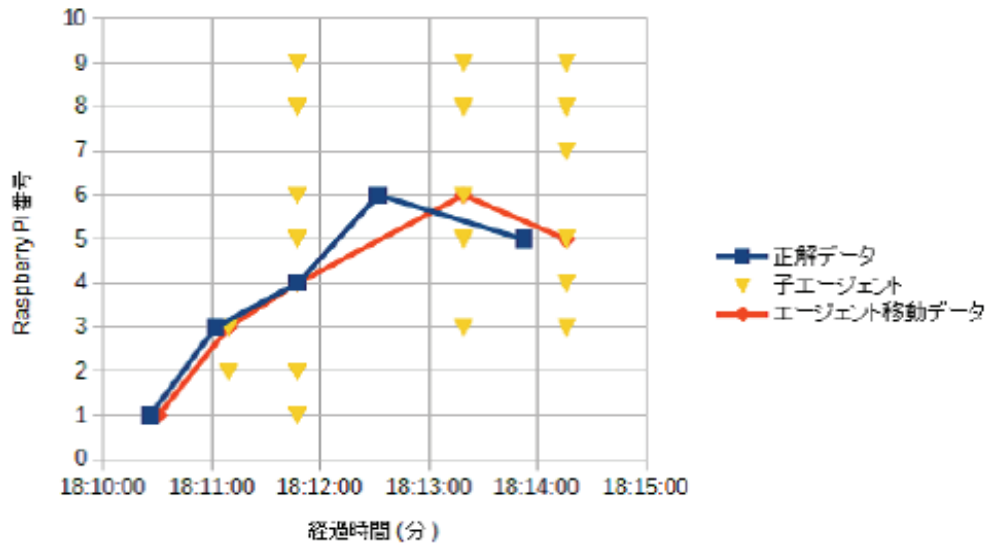


図 2.15: 子エージェントの配布先

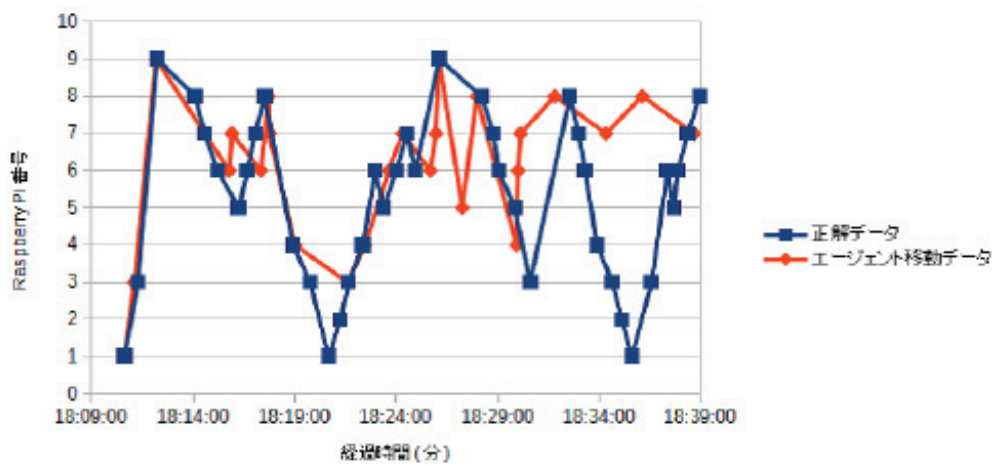


図 2.16: 人物の移動とエージェントの追跡結果（追従を意識せずに移動した場合）



実験結果では、エージェントが人物を正しく追跡できていないことが分かる。正しく追跡できなかった理由として一つはビーコンの持ち方が挙げられる。ビーコンは電波によりタグ情報を発信しているため、服のポケットに入れるなど持ち方により、電波を受信しにくくなったため、ノードで人物を検出できずエージェントが追跡できなかったと考えられる。もう一つの理由として、ノードの設置場所が考えられる。ノード間の距離が近い場合、人物に一番近いノード以外のノードでタグを受信した。このため、正しい追跡位置ではないところで人物が検出されることにより正しい追跡が行われなかったと考えられる。

### 2.7 結言

本節では、システムがある特定の人物をセンサで検出し、人物を検出した場所と時刻を記録し、追跡情報を可視化できるシステムを人物追跡システムと定義した。また、システムを分散構成とすることで拡張を容易にし、複数の人物を同時に追跡できるシステム構成を考案した。分散構成のシステムで人物の特徴情報を受け渡すためにモバイルエージェント技術を適用した。

本節で示した簡易実験でも述べたとおり、センサの位置で人物が通過した時間に正しく人物を検出して追跡できていないケースがあることが分かった。これはビーコンをポケットに入れたり人物が早く移動することでセンサがビーコンの発する電波を検出できないといったデバイスやセンサの性能や精度によることが大きい。また、本節の実験ではカメラを用いておらず、カメラのパン・チルト・ズームのようにセンサの検出範囲が変化する場合を考慮していない。

センサの検出範囲の変化や、センサの性能や精度の人物追跡への影響とその対応について、3章、4章、および5章にて述べる。

## 第3章 センサの検出範囲の変化に対応した隣接関係の導出法

### 3.1 序言

センサを利用したシステムにおいて、センサの検出範囲はかならずしも固定ではない。例えば、カメラの場合はパン、チルト、ズームの操作を行うことができ、その場合に映る映像は向きも異なれば対象の映り方も大きく変化する。その結果、あるカメラで撮影されたターゲットが、パン、チルト、ズームにより次に撮影されるであろうとされていたカメラで撮影されずに、別のカメラで撮影されることもある。このため、ターゲットを正しく追跡できない状況が発生する。そこで、この追跡の阻害要因を排除するために、グラフ理論<sup>63)</sup>を参考にしてセンサの検出方向や検出距離の変化を元にカメラの隣接関係を再計算し、新たなセンサの隣接関係を導き出すアルゴリズムを開発した。全ての指向性があり検出方向が変わるセンサが該当するが、本章では可動式カメラを対象にアルゴリズムを説明する。

ここでは、3.2節でセンサの検出範囲の変更における問題点の詳細を述べ、3.3節でその問題を解決するためのアルゴリズムの概要について説明する。3.5節にまとめを述べる。

### 3.2 センサの検出範囲と人物の追跡

一般にセンサを用いて監視を行うシステムでは、センサは離散的に配置されており、その検出範囲はマップ上を網羅していることは少ない。例えば、オフィスの防犯であっても、カメラの撮影範囲は出入口や通路などに限定されるのが一般的である。このため、センサが検出した人物はセンサに検知されない領域をどのように移動するか不明である。従って、あるセンサで検出された人物を追跡する場合に、その人物が次にどのセンサで検出されるかは分からない。そこで、一般的

### 第3章 センサの検出範囲の変化に対応した隣接関係の導出法

な監視システムでは全てのセンサについて全ての追跡対象の人物の検出をチェックする。このようにすることで、いつどのセンサで人物を検出してもその情報は全てのセンサを管理する中央サーバにより把握され、人物の追跡が可能となる。この場合、中央サーバはそのマップとセンサの位置と追跡対象の人物を全て把握していることになる。

しかし、追跡範囲を大きく広げていく場合、中央サーバで管理する方式では情報量の扱いに限界が生じる。1台の計算機が適切な速度で人物を追跡するためのリソースは限られ、また全てのマップ情報を1つの計算機に持つことは現実的ではない。近年はクラウドコンピューティングの発展により、サーバサイドで分散処理を行いスケーラビリティを確保することはできるが、トラフィックの問題は残ると考えられ、また全ての監視対象範囲の情報と追跡対象の人物の情報をクラウドという単一のプラットフォームに載せることに対する抵抗感が全くないとは言えない。

そこでシステムを分散構成とし、お互いに連携する構成を考慮する。あらかじめ分散構成を前提としているため、追跡範囲の拡張には柔軟に対応できる。また、対象の人物をどこまで追跡するか、情報をどのように渡していくかは装置間の連携に委ねられる。

ここで必要になるのは、分散している装置間の連携をどのように取って追跡を継続するかになる。そもそもセンサの検出範囲は地図上を完全に網羅していないため、センサの検出範囲に含まれない部分をどのように人物が移動するかが分からない。そこで、通路の情報からあるセンサで検出された人物が次にどのセンサを検出するかを接続情報を持ち、それらのセンサを重点的に監視するようにする。このようにすることで、全てのセンサの情報を監視しなくても人物の追跡を行うことができる。

#### 3.2.1 センサの検出範囲の変更における問題点

一般に中央サーバが全ての情報を持つシステムでは、センサの検出範囲の変更状況を全て把握できるため、センサの検出範囲が変更されても人物の追跡に大きな支障は出ない。ただし、前述したように中央サーバ、あるいはクラウドであったとしても、全てのセンサが常にサーバにアクセスし更新し続けるのかの問題は

ある。一方で、分散システムにおいては全てのセンサの情報を管理する必要はないが、反面、追跡対象の人物の情報を関係する場所、あるいは装置に伝えていく必要がある。本稿では、低コストで拡張性の高い分散構成のシステムを対象としているため、人物の情報を周囲の装置に伝えて追跡を続けるにあたって、センサの検出範囲が変更された場合の対応を考慮する必要がある。

センサの検出範囲は、例えばカメラのパン、チルト、ズームのように監視員などによって操作される可能性がある。あるいは、定期的にセンサの検出範囲を自動で切り替える場合もありうる。センサの検出範囲が変わりうるということは、次に人物を検出する可能性のあるセンサも変わるということである。従って、少なくともセンサの検出範囲の影響を受ける可能性のある装置には、検出範囲が変更されたという情報を伝達して、次に人物を検出するセンサがどれになるかを見直す必要がある。

## 3.3 隣接ノード決定アルゴリズムによる問題解決

センサの検出範囲がマップを完全に網羅せず、その範囲が任意に変更されることを考慮すると、分散システムではそれぞれの装置がそれぞれの検出範囲を把握し、かつ周辺の装置への影響を考慮する必要がある。これを実現するために、通路の関係と検出範囲の情報を簡素化し、装置間での連携を行いやすくすることが求められる。本研究ではグラフ理論をベースに、通路の関係とセンサの検出範囲をグラフで示しこれを簡素化する。具体的にはセンサ間の関係をつないだグラフを人物が移動する可能性のある経路とみなして監視と追跡を行う。ここでつながりのあるセンサを隣接センサとして定義する。隣接センサについて次項で説明する。

### 3.3.1 隣接センサ

屋内におけるセンサの設置場所について考えると、すべての場所で検出できるようにセンサを設置するよりも、部屋の入口や建物の出入り口、通路が交差する場所といった場所だけに離散的に設置することが現実的である。すなわち、センサがカバーしていない場所が存在する。エージェントが人物を追跡するためには現在人物を検出しているセンサと、次に人物を検出する可能性があるセンサを関

### 第3章 センサの検出範囲の変化に対応した隣接関係の導出法

連付ける必要がある。最も単純な方法としては、全てのセンサを全てのセンサと関連付ける方法が考えられる。しかし各センサが他の全てのセンサの存在を知っておく必要があり、システムの拡張や環境の変化に柔軟に対応することが難しい。

そこで、あるセンサで検出した人物が次に出現する可能性のあるセンサを隣接センサとして定義し、そのセンサ間の関係を隣接関係と定義する。隣接センサを隣接関係から計算することで、人物が次に出現すると予測される監視範囲を限定し、エージェントが監視すべき範囲を制限することができる。

フロアに配置されたセンサ（カメラ）とその検出範囲（撮影範囲）による隣接関係の例を図3.1に示す。矢印はセンサ間の隣接関係を示す。隣接センサの定義により、検出範囲が変化すると隣接関係は変化する。左の図では線さC1の隣接センサは、C2とC3である。一方、右の図ではセンサC1の隣接センサはC2のみであり、C3はC2の隣接センサとなる。

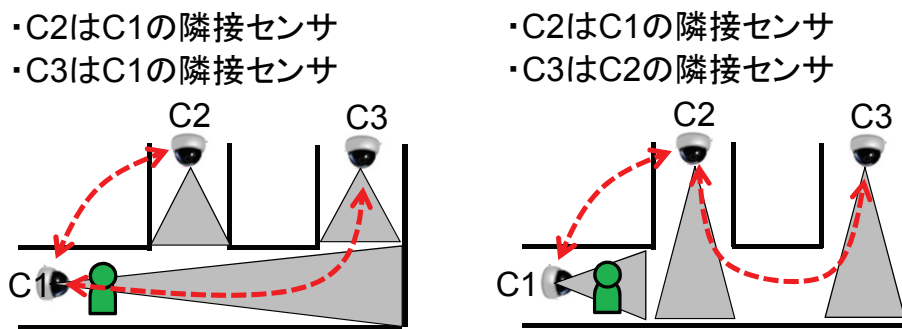


図 3.1: 隣接センサ

### 3.3.2 追跡の流れ

監視エリア内に追跡対象人物が来ると、監視ノードはその人物を追跡するエージェントが監視ノード内に存在するか確認する。存在すればそのエージェントが人物を追跡する。存在しなければその人物を追跡するためのモバイルエージェントを生成する。生成されたモバイルエージェントはその人物の特徴情報を取得し保持する。センサがカメラの場合は画像を解析し特徴情報を取得・保持し、ビーコンの場合はその人物が保持するビーコンが発する電波に含まれる識別子 (ID) を保持する。追跡対象人物がセンサの検出範囲から離れ、そのセンサで検出されなくなった場合、モバイルエージェント (親エージェント) は隣接センサを管理する監視ノードに自身の複製 (子エージェント) を派遣する。親/子エージェントはセンサの情報を定期的に解析し、追跡対象人物が検出されているか判定する。検出されていた場合には追跡対象人物が見つかったことを親エージェントに知らせる。親エージェントはすべての子エージェントに追跡対象人物が発見されたことを通知し実行を停止する。通知を受け取ると、追跡対象人物を発見した子エージェントは親エージェントに変化し、それ以外のエージェントは実行を停止する。これを繰り返すことで、追跡対象人物の移動に併せてモバイルエージェントが監視ノードを移動し追跡を継続する。これにより複数台のセンサに跨った人物の自動追跡を実現する。

追跡アルゴリズムを図 3.2 に示す。実際の追跡では例えば画像解析の失敗により人物を見逃した場合、人物の追跡に失敗するといったことが発生する。このような場合、その人物を追跡する親/子エージェントが存在しない監視ノードにてその人物が検出されると、追跡エージェントが生成され仮追跡を行うが、本稿では隣接関係の計算に焦点を当てているため詳細は省く。

第3章 センサの検出範囲の変化に対応した隣接関係の導出法

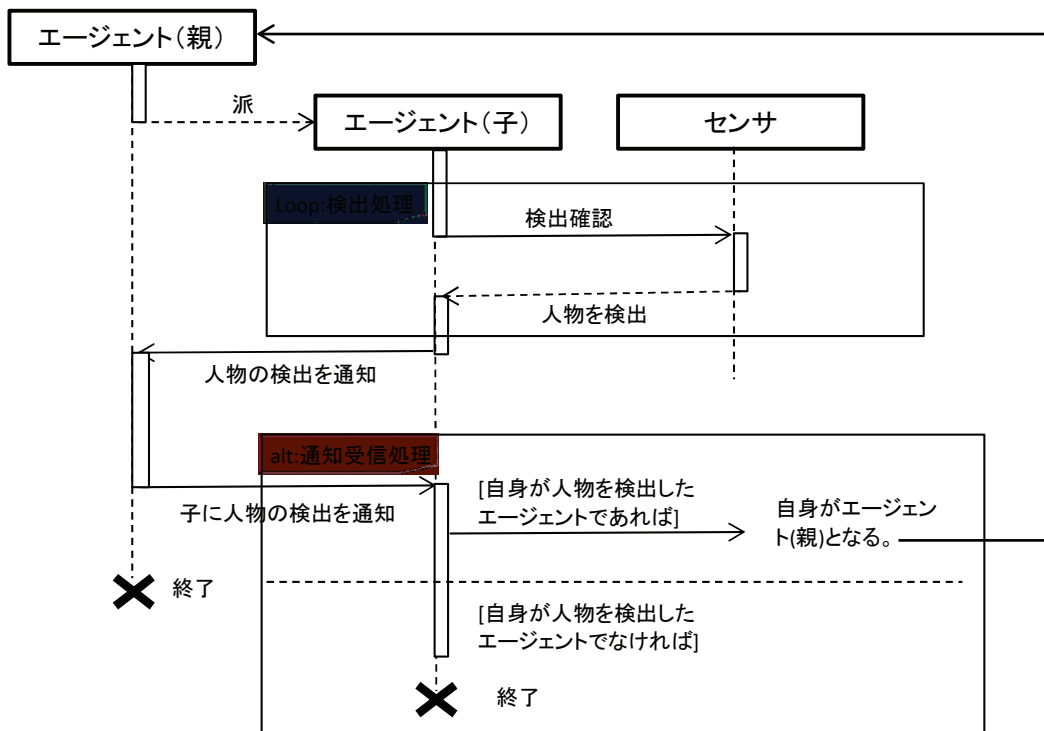


図 3.2: エージェントの追跡アルゴリズム

### 3.3.3 隣接関係の計算

センサの隣接関係を計算するために、分岐点、センサ点、検出点を定義する。分岐点は移動経路で分岐が発生する場所であり、管理エリアの地図などにより予め設定することができる。センサ点はセンサの設置場所であり、センサの設置および撤去に伴い変化する。検出点は、検出範囲を表現可能とするために、分岐点間、センサ点間、分岐点-センサ点間にそれぞれ一つずつ設ける。これらの点により、センサの検出範囲を  $|S| \times |P|$  の行列  $X$  とし式3.1のように表現する。ここで、 $S$  はセンサ点、 $P$  は分岐点、センサ点、または検出点とする。

$$X_{ij} = \begin{cases} 0 & \text{センサ } S_i \text{ の検出範囲に点 } P_j \text{ を含まない場合} \\ 1 & \text{センサ } S_i \text{ の検出範囲に点 } P_j \text{ を含む場合} \end{cases} \quad (3.1)$$

このとき、検出範囲が重なっているセンサは式3.2を計算し要素を確認することで求めることができる。

$$D = X \cdot X^T \quad (3.2)$$

また、分岐点、センサ点、検出点間の接続関係を  $|P| \times |P|$  の行列  $Y$  とし式3.3のように表現する。

$$Y_{ij} = \begin{cases} 0 & \text{点 } P_i \text{ と点 } P_j \text{ が隣り合っていない場合} \\ 1 & \text{点 } P_i \text{ と点 } P_j \text{ が隣り合っている場合} \end{cases} \quad (3.3)$$

ここで例として、センサの検出範囲と行列  $X, Y$  の内容を図3.3に、行列  $X$  と  $Y$  の内容を式3.4と式3.5に示す。

$$X = \begin{matrix} & \begin{matrix} C1 & C2 & C3 & C4 & B1 & B2 & B3 & B4 & V1 & V2 & V3 & V4 & V5 & V6 & V7 \end{matrix} \\ \begin{matrix} C1 \\ C2 \\ C3 \\ C4 \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{matrix} \quad (3.4)$$



第3章 センサの検出範囲の変化に対応した隣接関係の導出法

センサ  $C_1$  の隣接センサは  $C_2$  と  $C_4$ .  
 センサ  $C_2$  の隣接センサは  $C_1$  と  $C_3$ .  
 センサ  $C_3$  の隣接センサは  $C_2$ .

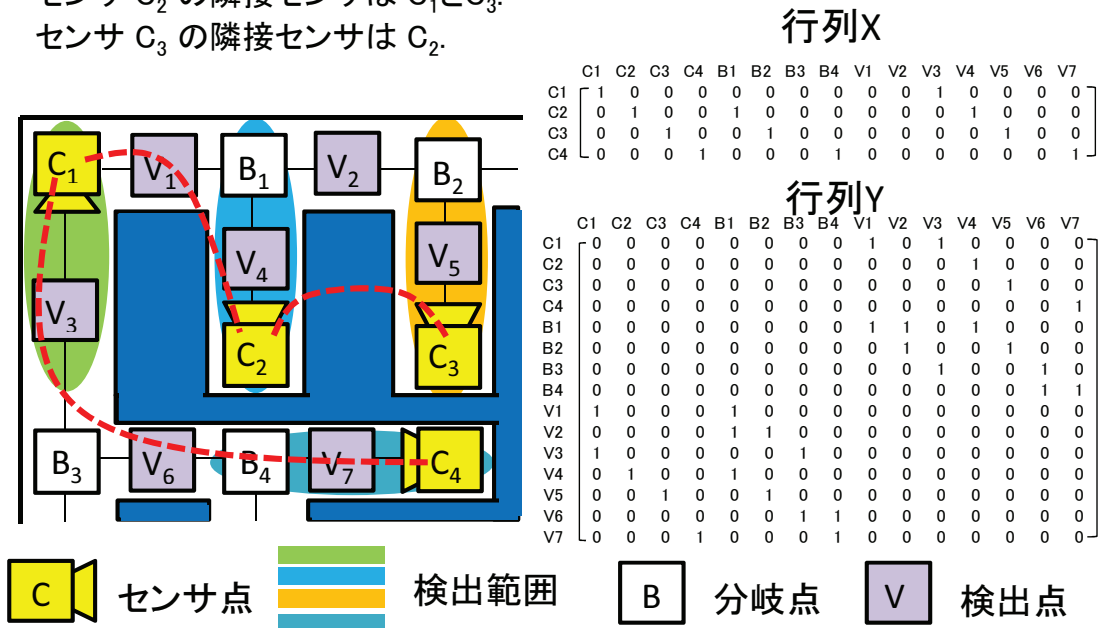


図 3.3: センサの検出範囲と行列 X, Y

	C1	C2	C3	C4	B1	B2	B3	B4	V1	V2	V3	V4	V5	V6	V7
C1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
C2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
C3	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
C4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
B1	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0
B2	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0
B3	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
B4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
V1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
V2	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
V3	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
V4	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
V5	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
V6	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
V7	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0

$$(3.5)$$

このとき、式3.6を計算し  $E_{ij} \geq 1$  となる要素を確認することで、センサ  $S_i$  の検出範囲から  $(n - 1)$  個以内の点を隔てた点にセンサ  $S_j$  の検出範囲が存在することが確認できる。

$$E = X \cdot Y^n \cdot X^T \quad (3.6)$$

さらに行列  $X$ ,  $Y$  からどのセンサの検出範囲にも含まれていない点を削除した行列  $X'$ ,  $Y'$  を作成する。例えば図3.3では、センサ C1 とセンサ C2 の検出範囲の間にどのセンサの検出範囲にも属していない V1 が存在するため、 $E = X \cdot Y^2 \cdot X^T$  により C1 と C2 が隣接することを求めることができる。また、 $E = X \cdot Y^3 \cdot X^T$  により、C1 と C4 が隣接することを求めることができる。しかし、検出範囲はセンサの制御により変化する。このため、何個の点を隔てた場所に隣接センサが存在するか、すなわち  $n$  をどのように設定すればよいかかわからない。そこで、どのセンサの検出範囲にも含まれていない点を削除した行列  $X'$ ,  $Y'$  を作成する。

行列  $X'$  は、式3.7を満たす、すべての  $j$  について、行列  $Y$  から行  $j$  と列  $j$  を削除する。このとき、行  $j$  と列  $j$  の削除により接続関係が断たれることを防ぐために、 $Y_{ij} = 1$ ,  $Y_{jk} = 1$  を満たす  $Y_{ik}$  を 1 とする。これにより行列  $Y'$  を作成する。

$$\sum_{k=1}^m X_{kj} = 0 \quad (3.7)$$

このように、どのセンサの検出範囲にも含まれていない点を削除したイメージを図3.4に、行列  $X'$  と  $Y'$  の内容を式3.8と式3.9に示す。図3.4で赤字で示された箇所は、削除された点を介して接続されている2点を接続するために1としたものである。

$$X' = \begin{matrix} & \begin{matrix} C1 & C2 & C3 & C4 & B1 & B2 & B4 & V3 & V4 & V5 & V7 \end{matrix} \\ \begin{matrix} C1 \\ C2 \\ C3 \\ C4 \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \end{matrix} \quad (3.8)$$

センサ  $C_1$  の隣接センサは  $C_2$  と  $C_4$ .  
 センサ  $C_2$  の隣接センサは  $C_1$  と  $C_3$ .  
 センサ  $C_3$  の隣接センサは  $C_2$ .

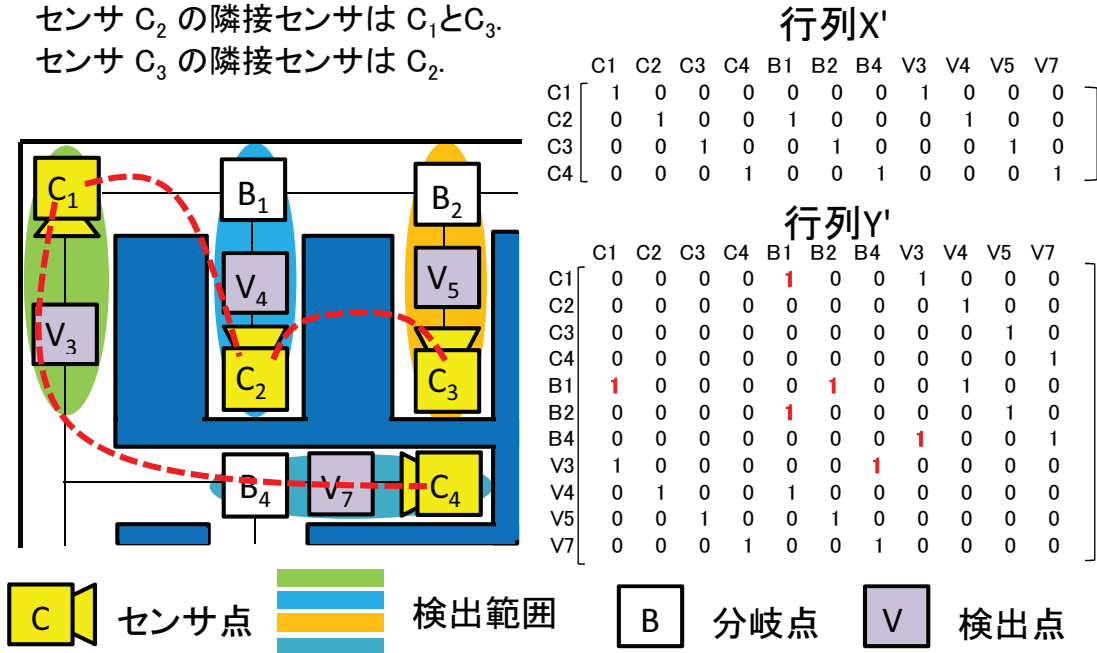


図 3.4: センサの検出範囲と行列  $X'$ ,  $Y'$

$$Y' = \begin{matrix} & \begin{matrix} C1 & C2 & C3 & C4 & B1 & B2 & B4 & V3 & V4 & V5 & V7 \end{matrix} \\ \begin{matrix} C1 \\ C2 \\ C3 \\ C4 \\ B1 \\ B2 \\ B4 \\ V3 \\ V4 \\ V5 \\ V7 \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

(3.9)

これらの行列  $X'$ ,  $Y'$  から, 式 3.10 を計算することで隣接センサを求めることができる.

$$E' = X' \cdot Y' \cdot X'^T \quad (3.10)$$

### 3.4 評価

提案手法の有効性を検証するために、OSGi フレームワークの仕様に準拠したバンドルとして実現されているモバイルエージェント実行環境に本提案手法を実装した。実験は21m × 13.5m の監視エリアに11台のセンサを設置することを想定して行った。センサはパン・チルト・ズーム可能なカメラとし、その撮影範囲を不定期に変更する。また追跡対象人物は監視エリア内を1.5m毎秒の速度でランダムに移動する。カメラの撮影範囲には死角がなく、追跡対象人物が出現した場合は必ず検出し、追跡対象人物の特定も必ず成功する条件で、追跡対象人物を2人ずつ出現させ消失させるシミュレーションを10回実施した。図3.5に1人分のシミュレーション結果を示す。

縦軸が移動先の監視ノード、横軸が経過時間であり、追跡対象人物の移動を点線、モバイルエージェント(親エージェント)による追跡結果を実線で示す。また、パン・チルト・ズーム値の変更タイミングを矢印で示す。追跡結果から、11秒目に追跡対象人物が監視エリアに侵入したことを監視ノード9で検出し、監視ノード11、監視ノード6、監視ノード5と移動する。43秒目に監視ノード10まで移動した時点で、監視ノード10のパン・チルト・ズーム値が変更されるが、その後も130秒目に追跡対象人物が監視エリア外へ出るまで継続して追跡できていることが確認できる。

このシミュレーションを24時間分動作させた結果、すべての追跡対象人物をモバイルエージェントが追跡できており、追跡中にパン・チルト・ズームの値を変更しても追跡対象人物を逃すことなく追跡が可能であることが検証できた。

第3章 センサの検出範囲の変化に対応した隣接関係の導出法

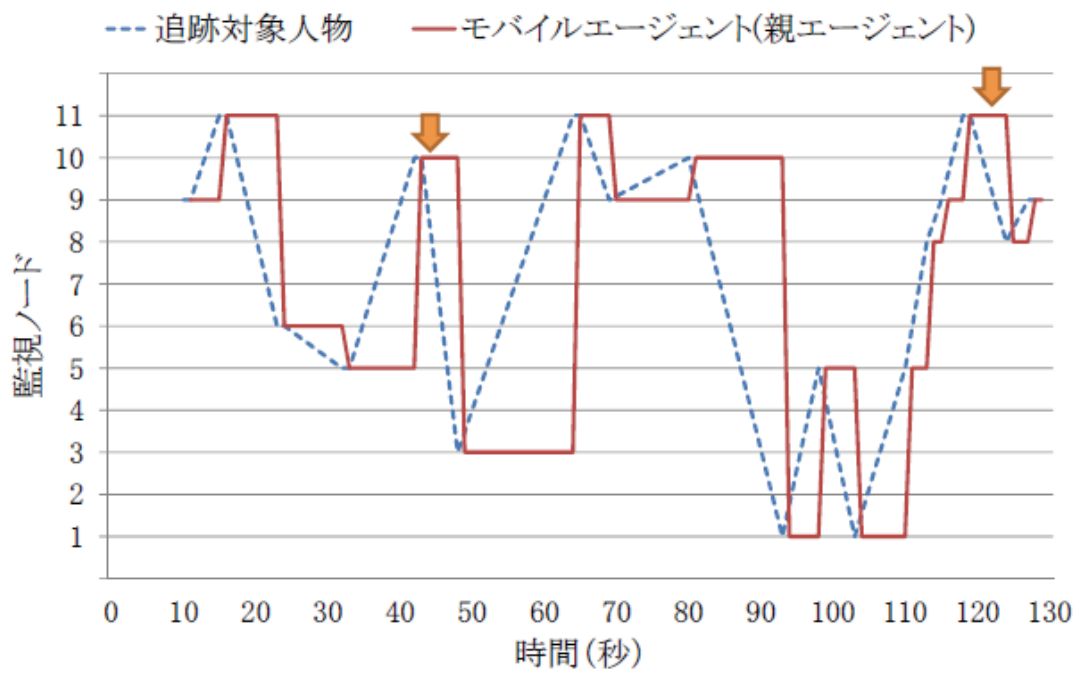


図 3.5: 追跡実験の結果

### 3.5 結言

本節ではセンサの検出範囲を考慮してモバイルエージェントが移動する監視ノードを動的に決定するための手法を提案した。センサが離散的に配置されている環境では、マップ上のどこに人物が存在するかを常に把握することはできない。そこで、通路の情報からある線さで検出された人物が次にどのセンサで検出されるかの接続情報、すなわち隣接関係を計算することによって、全てのセンサを監視しなくても人物の追跡が行えるようにした。しかし、パン・チルト・ズームの機能を持ったカメラのような指向性を持つセンサの検出範囲が変更される場合は、検出範囲の変更により次に人物を検出する可能性のあるセンサが変化するため、センサの検出範囲の情報を取得し、検出範囲の情報を持つ行列を更新することにより最新の検出範囲に対応した隣接関係を計算することで検出範囲の変化に対応した。課題としては、1台のセンサの検出範囲が変更になった場合、マップ上の全ての行列が更新および再計算の対象となることである。1台のセンサの検出範囲の変更を、システム上の全てのセンサへ展開する必要がある。この問題については、次章にて解決策を提案する。

## 第4章 隣接関係の局所化手法

### 4.1 序言

分散構成のシステムは、処理を分散することで1台の計算機における処理コストを低減することを目的とする場合と、ある計算機が故障したときに他の計算機で処理を継続することにより可用性を高めることを目的とする場合がある。これらの目的の場合は、それぞれの計算機がそのシステムが扱う全てのセンサなどの管理情報を持っていたとしても不都合は生じない。しかし、システムの情報が状況に応じて随時更新される場合は、それぞれの計算機が持つ情報を更新する必要がある。そこで、組織のシステムでは、処理する計算機は分散していても、情報としてのストレージはDBやファイルサーバといった形で一本化していることが多い。そのためDBやファイルサーバには特にI/Oやネットワークに高いスペックが求められる。更にシステムや組織の規模が大きくなってくると、情報自体も分散して配置した方が都合がよくなることがある。単純にI/Oやネットワークのスペックに限界がある場合もあるが、組織を跨る場合に全ての情報を単一のサーバで保持せず、その情報を外部に出さないようにしたいといったニーズが出てくるからである。

本稿で提案している人物追跡システムも同様である。2章ではカメラ間の人物の同定処理を分散しコストを下げるためにモバイルエージェントを用いた人物追跡システムを提案した。3章ではセンサの検出範囲に変化に対応するアルゴリズムを提案したが、ある1つのノードで発生した検出範囲の変化はシステムが持つ情報への影響があるため、全てのノードに展開して検出範囲の情報を更新する必要があった。システム内の全てのノードの情報を各ノードが同じように持つことは情報を重複して持つことになるため望ましいことではない。しかし、そのために共通的に情報を管理する計算機を設置することは、システムの規模に合わせてその集中して情報を管理する計算機のI/Oやネットワーク性能を考慮する必要が

あることを意味する。そもそも遠く離れた、例えば別の建物にあるセンサの検出範囲の情報まで管理しなければならないのか、といった問題が出てくる。

そこで、本章では人物追跡システムにおいて、各ノードが扱う情報をノードごとに局所化する手法を提案する。各ノードが扱う情報はそれぞれのノードでの検出状況に変化が発生したときに通信を介して他のノードに展開できるように考慮する。各ノードが持つ情報を1箇所で共有するのではなく、各ノードがそれぞれが持つ情報を用いて他のノードと連携することによって、システムを構成するノードがいかなる状況であっても連携して追跡ができる追跡手法を提案する。

### 4.2 各センサが持つ情報の局所化

一般的に、オフィス内を監視する人物追跡システムでは、人物やセンサの情報を集中して管理する場合が多い。2章と3章では、各ノードの処理コストの増加を低減する分散型のシステム構成を提案しているが、人物の認識処理は分散しているものの、センサの隣接関係の情報は全てのセンサで同じ情報を持ち、その情報に変化が生じた場合にはその情報を全てのノードに展開する必要があった。そこで、以下の観点でセンサの隣接関係の情報を局所化し分散する。

**センサの検出範囲は有限であるため、検出範囲の変更の影響は限定的である：**

センサの検出範囲は有限である。システムの規模が大きくセンサ（ノード）の数が増えるほど互いに近い位置にあるノードは検出範囲の変更の影響は受けるが、全く影響を受けないノードの方が多数となると考えられるため、各ノードが持つ情報を局所化することは大処理や通信コストの大きな削減に大きな効果が見込める。

**ノード同士は通信が可能であり、情報の更新は随時可能である：**

各ノードに設置されている機材はモバイルエージェントを動作させることが可能な小型計算機である。ノード自身が通信が可能であるため、状況の変化を通信を介して他のノードに随時伝播することが可能である。

**各ノードが必要最小限の情報を持つ方がシステム全体の拡張性が高くなる：**

各ノードが自身のノードの周辺の情報だけを持ち、通信で随時その情報を更新できるようになると、故障等で無効になったノードや新たにセンサの設置個所が増え追加されたノードを考慮した情報の展開ができるようになり、シ



システムの拡張性が高くなる。

### 4.3 隣接関係の局所化による計算手法

3章で述べた隣接関係の計算アルゴリズムは、各点の接続関係を示す行列  $Y$  と、センサの検出範囲を示す行列  $X$  を元に計算を実行する。これらの行列は、マップ上の全ての点を含んでいる。そのため、マップ内のどれか1つのセンサの検出範囲が変更になった場合には、行列  $X$  を更新することになり、さらに全てのノードに行列  $X$  を展開する必要が出てくる。マップが小さい場合は大きな負担にはならないが、マップが大きい場合には、その受け渡す情報の量と再計算のコストは無視できない大きさとなる。

そこで、各センサがそれぞれの隣接関係を求めることで隣接関係の計算を局所化する。それぞれのセンサがそれぞれの隣接関係を求めるためには、対象となるセンサの検出範囲内の点から他のセンサの検出範囲内の点までに存在する分岐点、センサ点、検出点が必要となる。そこで、それぞれのセンサは、これらの点だけからなる行列  $Y_s$  を生成する。

$$Y_{s_{ij}} = \begin{cases} 0 & \text{点 } P_{s_i} \text{ と点 } P_{s_j} \text{ が隣り合っていない場合} \\ 1 & \text{点 } P_{s_i} \text{ と点 } P_{s_j} \text{ が隣り合っている場合} \end{cases} \quad (4.1)$$

$P_{s_i}$ ,  $P_{s_j}$  は、対象のセンサの検出範囲内から隣接センサの検出範囲内までの経路に含まれる点である。同様に行列  $X$  もこれらの点だけからなる行列  $X_s$  を生成する。

$$X_{s_{ij}} = \begin{cases} 0 & \text{センサ } S_i \text{ の検出範囲に点 } P_{s_j} \text{ を含まない場合} \\ 1 & \text{センサ } S_i \text{ の検出範囲に点 } P_{s_j} \text{ を含む場合} \end{cases} \quad (4.2)$$

これらの行列  $X_s$ ,  $Y_s$  から、3.4節と同様の方法でどのセンサの検出範囲にも含まれていない点を削除することで行列  $X_{s'}$ ,  $Y_{s'}$  を作成し、4.3式により隣接センサを求めることができる。

$$Es' = Xs' \cdot Ys' \cdot Xs'^T \quad (4.3)$$

これにより，すべてのセンサの検出範囲を管理せず，局所的な範囲の隣接センサの情報だけを利用し，それぞれのセンサの隣接関係を計算する。

各センサが局所的な情報のみを扱い，更にセンサ間がそれぞれ局所化された行列  $Xs$ ,  $Ys$  の受け渡しを行うことで，センサ台数や追跡範囲が広がっても把握する対象の情報が無制限に増え続けることが無く，環境が変化した場合の影響を局所的な範囲に収めることができる。

#### 4.4 削減例

局所化では，1つのノードが扱うべき点が削減されるため，計算コストが低減する．局所化の例を図4.1のマップを用いて説明する．

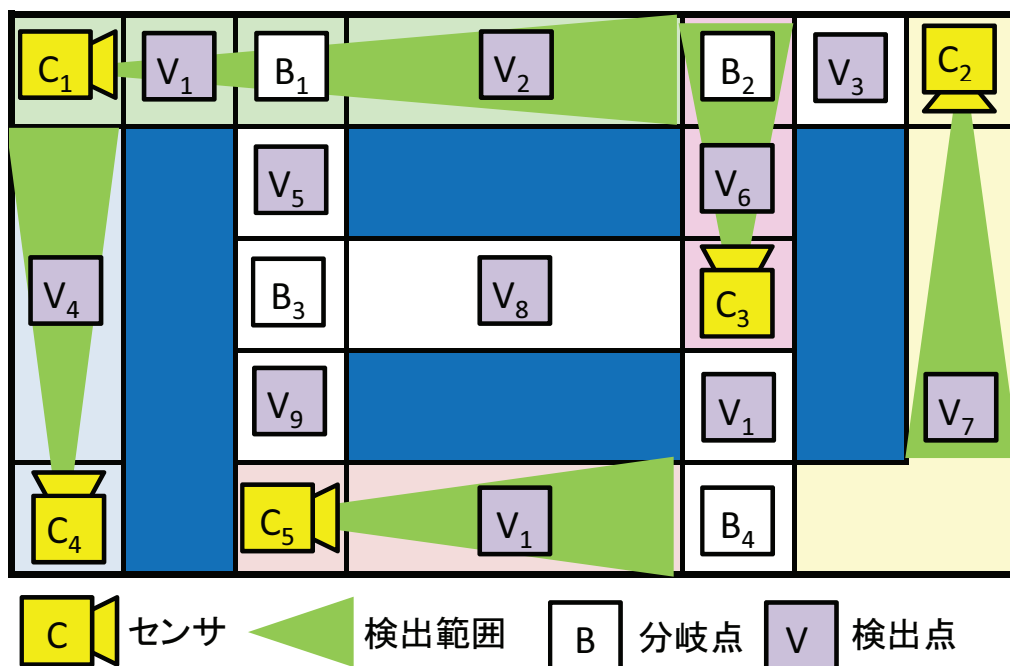


図 4.1: マップにおける局所化の例

図 4.1 では， $C_i$  はセンサ点， $B_i$  は分岐点， $V_i$  は検出点を示している．このマップには5つのセンサ点，4つの分岐点，10の検出点が存在する．合計の点の数は

19点となる。行列  $X$  のサイズは  $5 \times 19$  (センサ点の数  $\times$  全ての点の数), 行列  $Y$  のサイズは  $19 \times 19$  (全ての点の数  $\times$  全ての点の数) となる。検出範囲は三角の範囲で示している。

ここで, センサ  $C1$  に注目する。  $Y_{S_1}$  はセンサ  $1$  の検出範囲に含まれる点とその隣接センサに到達するまでに含まれる点で構成される。ここで,  $Y_{S_1}$  は  $C1, C3, C5, B1, B2, B3, V1, V2, V4, V5, V8$  と  $V9$  の点で構成される。行列  $Y_{S_1}$  のサイズは  $12 \times 12$  となる。

$$Y_{S_1} = \begin{matrix} & C1 & C3 & C5 & B1 & B2 & B3 & V1 & V2 & V4 & V5 & V8 & V9 \\ \begin{matrix} C1 \\ C3 \\ C5 \\ B1 \\ B2 \\ B3 \\ V1 \\ V2 \\ V4 \\ V5 \\ V8 \\ V9 \end{matrix} & \left( \begin{array}{cccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right) & \end{matrix} \quad (4.4)$$

同様に,  $X_{S_1}$  のサイズは 4.5 式に示すように  $4 \times 12$  となる。

$$X_{S_1} = \begin{matrix} & C1 & C3 & C5 & B1 & B2 & B3 & V1 & V2 & V4 & V5 & V8 & V9 \\ \begin{matrix} C1 \\ C3 \\ C4 \\ C5 \end{matrix} & \left( \begin{array}{cccccccccccc} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right) & \end{matrix} \quad (4.5)$$

また, 行列  $X_{S_1}$  から, 列  $B3, V5, V8, V9$  を削除した行列  $X_{S'_1}$  を作成する。こうして,  $X_{S'_1}$  は式 4.6 に示すような  $4 \times 8$  のサイズの行列となる。

$$Xs'_1 = \begin{matrix} & C1 & C3 & C5 & B1 & B2 & V1 & V2 & V4 \\ \begin{matrix} C1 \\ C3 \\ C4 \\ C5 \end{matrix} & \left( \begin{array}{cccccccc} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right) \end{matrix} \quad (4.6)$$

行列  $Ys'_1$  を生成するために、まず最初に B3, V5, V8, V9 の行と列を行列  $Ys_1$  から削除する。その後、 $Ys_{ik}=1$  と  $Ys_{kj}=1$  を満たす B3, V5, V8, V9 の行列  $Ys'_1$  上の点は1にセットされる。例えば、V5の削除により、 $[i, j] = [B1, B3]$  と  $[B3, B1]$  の点が1に設定される。こうすることにより、B1とB3との間の経路がV5が削除されても切断されることなくつながることになる。同様に、B3の削除により  $[i, j] = [B1, V8], [V8, V9], [B1, V9], [V9, B1]$  の点が1に設定される。V8によって  $[B1, C3], [C3, B1], [C3, V9], [V9, C3]$  の点が1に設定され、V5によって  $[B1, C5], [C5, B1], [C3, C5], [C5, C3]$  の点が1に設定される。このようにして、4.7に示すように行列  $Ys'_1$  が生成される。

$$Ys'_1 = \begin{matrix} & C1 & C3 & C5 & B1 & B2 & V1 & V2 & V4 \\ \begin{matrix} C1 \\ C3 \\ C5 \\ B1 \\ B2 \\ V1 \\ V2 \\ V4 \end{matrix} & \left( \begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right) \end{matrix} \quad (4.7)$$

## 4.5 検出範囲の変化への対応

3章と同様にセンサには指向性があり、例えばカメラの場合はパン・チルト・ズームの変更によりセンサの検出範囲が変化する。検出範囲が変化すると、検出範囲内に含まれる点が変わり、隣接センサの検出範囲内の点を重なるか跨ったときに隣接センサが変化する。このとき、隣接関係を再計算する必要がある。3章で説明したアルゴリズムでは、各センサは全ての点の情報を持っており、検出範囲の変化により一部が変更された全ての点の情報を全てのセンサへ展開する。局所化を行うことにより各センサが持つ点の情報は削減されるため、あるセンサの検出範囲が変更されても自身が持つ点の情報に影響がないセンサが存在する。この場合、点の情報を展開する必要がなく、システムの処理コストの低減につながる。本節では、検出範囲の変化やセンサの追加、削除が行われた場合に変化に影響を受けるセンサの情報のみを更新することでシステムの処理コストを低減する方法について説明する。

検出範囲が変化したセンサは、検出範囲の変更を自身の隣接センサに通知する。このとき、新たに検出範囲の重なりが生じる隣接センサを計算することができる。また、新たに検出範囲になった点が行列  $X_s, Y_s$  に存在しないとき、隣接センサの検出範囲を跨いだと判定できる。例えば図4.1ではセンサ C2 の検出範囲が点 V15 を含むような範囲に変化した場合、センサ C2 が持つ行列  $X_s, Y_s$  に点 V15 が含まれていないためセンサの C4 の検出範囲を跨いだと判定する。新たに重なりが生じるか跨いだとき、隣接センサの計算に必要なすべての点の情報を持たず、隣接関係を計算することができない。そこで、重なるか跨いだセンサに対して、そのセンサが持つ行列  $X_s, Y_s$  を要求する。検出範囲が変化したセンサは、受け取った行列  $X_s, Y_s$  と自身の行列  $X_s, Y_s$  を統合する。統合された  $X_s, Y_s$  から隣接関係を計算することで新たな隣接センサを見つけることができる。

検出範囲の更新により新たな隣接センサが追加されると、追加されたセンサは、そのセンサ上に存在するエージェントに隣接センサを通知する。親エージェントは追加を受けた隣接センサに対して子エージェントを派遣する。このようにして、センサの検出範囲の変化に動的に対応しつつ人物の追跡を実現する。

### 4.5.1 センサの追加への対応

新たにセンサが追加された場合、追加されたセンサを含んだ隣接関係に更新する必要がある。しかし、追加されたセンサは行列  $X_s, Y_s$  を持たず、隣接関係を計算することができない。そこで、新たに追加するセンサ  $C_{new}$  には、そのセンサの隣接センサ  $C_1$  へのアクセス情報（例えば、IP アドレスや認証するための情報）を初期設定として与える。センサ  $C_{new}$  は、センサ  $C_1$  にアクセスし、センサ  $C_1$  が持つ行列  $X_{s_1}, Y_{s_1}$  を取得する。 $X_{s_1}, Y_{s_1}$  はセンサ  $C_1$  の隣接センサまでの点を含んだエリアの情報を持つため、センサ  $C_{new}$  が追加されるエリアを含む点の情報を持つ。

そこで、 $C_{new}$  は行列  $X_{s_1}, Y_{s_1}$  に  $C_{new}$  のセンサ点を追加する。追加の例を図5に示す。分岐点を表す範囲にセンサが追加された場合には分岐点をセンサ  $C_{new}$  のセンサ点とし、検出点を表す範囲にセンサが追加された場合には検出点を二つに分割し、その間にセンサ点を設置する。こうして更新された点を  $X_{s_1}, Y_{s_1}$  に追加すると共に、行列  $X_{s_1}$  をセンサ  $C_{new}$  の検出範囲により更新する。ここで追加・更新された  $X_{s_1}, Y_{s_1}$  をセンサ  $C_{new}$  が保持する行列  $X_{s_{new}}, Y_{s_{new}}$  とする。 $C_{new}$  は  $X_{s_{new}}, Y_{s_{new}}$  により4.1節と同様の計算を行うことで隣接関係を求めることができる。

更にセンサ  $C_{new}$  はセンサ点、検出点が追加されたこととセンサ  $C_{new}$  の検出範囲内の点を隣接センサに通知する。通知を受けた隣接センサはそれぞれの行列  $X_s, Y_s$  を更新することでセンサ  $C_{new}$  の追加による隣接関係の変化に対応する。

### 4.5.2 センサの削除への対応

センサの追加時と同様に、センサが削除された場合も隣接関係を再計算する必要がある。センサの削除はセンサが計画的に撤去された場合と障害により不意に削除された場合の2通りが考えられる。不意の削除では、削除されたセンサが持つ行列  $X_s, Y_s$  が失われ、計画的な撤去による削除への対処よりも困難であるため、ここでは不意の削除への対処方法を示す。

不意の削除への対処として、それぞれのセンサは、予めそれぞれの隣接センサと行列  $X_s, Y_s$  を交換する。それぞれのセンサは、隣接センサにアクセス可能かどうかを常に相互監視し、アクセスできなくなった場合に削除されたと判断する。

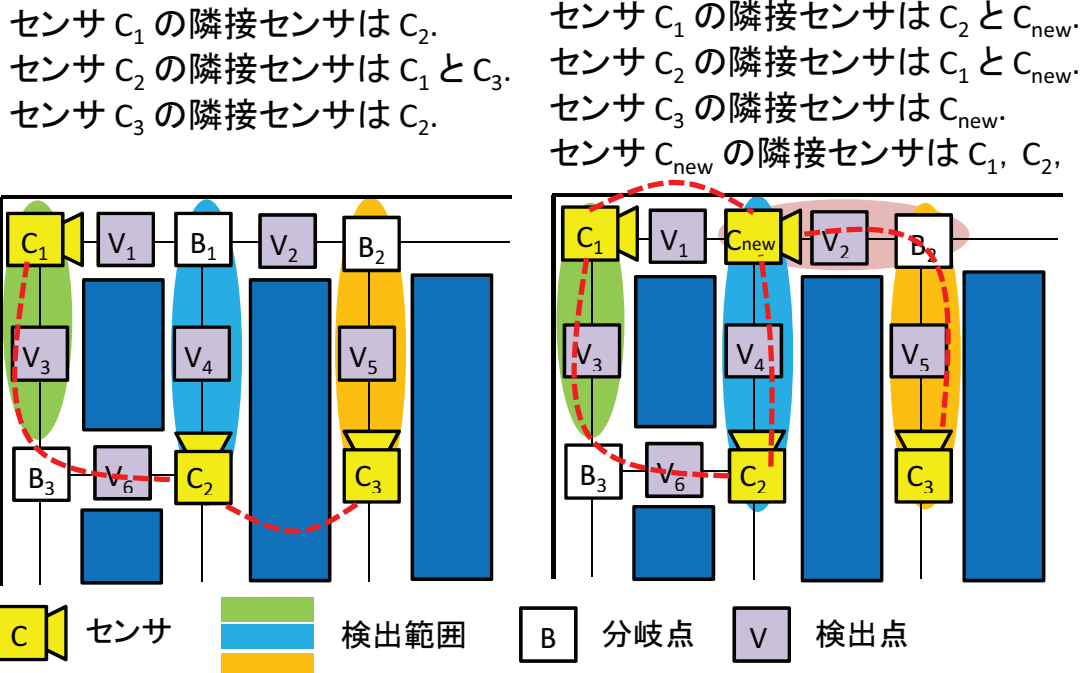


図 4.2: センサの追加による点の変化

例えば、センサ  $C_{del}$  の削除を隣接センサ  $C_1$  が検出したとする。このとき、センサ  $C_1$  は予め交換したセンサ  $C_{del}$  の行列  $X_{S_{del}}, Y_{S_{del}}$  と自身の行列  $X_{S_1}, Y_{S_1}$  を統合する。統合した行列のセンサ  $C_{del}$  のセンサ点に対応する行を全て 0 とする。これによりセンサ  $C_{del}$  の検出範囲は 0 となる。統合された行列  $X_s, Y_s$  を用いて 4.1 節と同様の計算を行うことで、センサ  $C_{del}$  が削除されたときの隣接関係を計算できる。このことで、センサが不意に削除されたとしても追跡を続けることができる。

## 4.6 評価

提案手法の有効性を検証するため、(1) 隣接関係の計算に必要な点の局所化による削減効果、(2) 隣接センサを正しく計算し追跡を継続できること、の 2 点を確かめるためのシミュレーション実験を行った。センサにはカメラを用いた。

実験では監視エリアのマップを作成し、そこに複数台のセンサ（カメラ）が設置されていることを想定して行った。カメラは不定期にパン・チルト・ズームにより検出範囲（撮影範囲）を変化させる。隣接関係の局所化に焦点を当てている

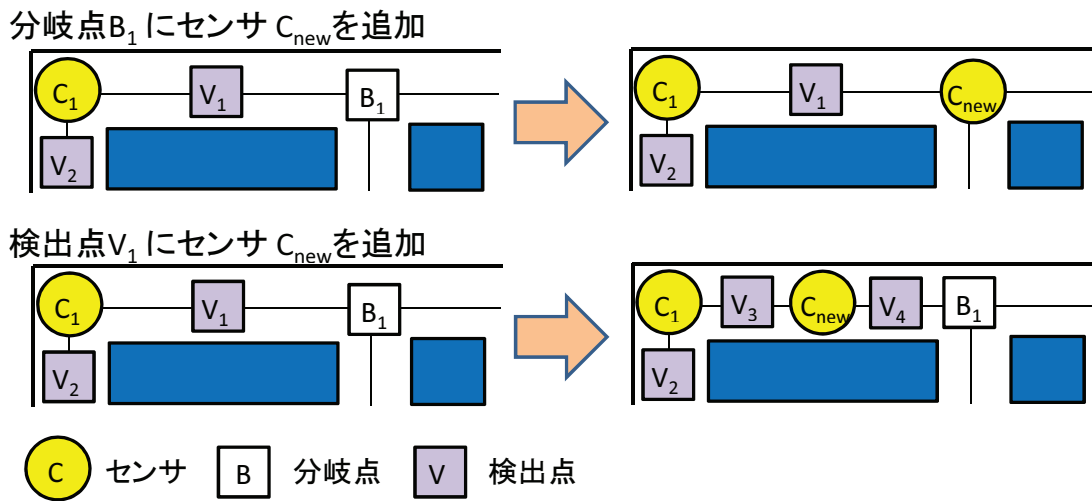


図 4.3: センサの追加による点の変化

ため、センサによる追跡対象の検出を必ず成功するものとした。

#### 4.6.1 局所化による点の削減効果

局所化による対象の点の削減効果を検証する実験を行った。実験は  $30\text{m} \times 30\text{m}$  から  $210\text{m} \times 210\text{m}$  までの大きさの異なる7種類のマップを準備した。実験対象のマップの点の数を表 4.1 に示す。実験ではこの点の数が局所化によってどのくらい削減されるかを評価した。実験の結果を表 4.2 に示す。

局所化ありの場合の  $Y_s$  を構成する点の数の平均値を見ると、局所化無しの場合に比べて約3~60%の数になっていることが確認できる。また、監視エリアが大きいほど点の数が削減できていることが確認できる。このため、各カメラにおける隣接関係の計算量を削減でき、各カメラが全てのノードを意識することなく通信などのオーバヘッドを削減できる。

センサの検出範囲が変更された場合、毎回隣接ノードを再計算する必要がある。監視領域が広い、すなわち点の数が多くなるほど隣接関係の計算にかかる時間は非常に長くなる。対象的に、局所化した場合は、点の範囲が抑えられるため計算にかかる時間も局所化しない場合と比べて短くなる。隣接関係にかかる時間を測定した結果を表 4.4 に示す。また、隣接関係の計算に要する時間を測定した。測定に用いた計算機のスペックを表 4.3 に示す。



表 4.1: マップ全体の点の数

マップの広さ [m <sup>2</sup> ]	点の数			
	センサ点数	分岐点数	検出点数	合計
30 x 30	8	18	44	70
60 x 60	32	80	184	296
90 x 90	72	189	418	679
120 x 120	128	328	740	1196
150 x 150	200	518	1160	1878
180 x 180	288	751	1674	2713
210 x 210	392	1030	2286	3708

表 4.3 の例では、局所化をしない場合は 210m × 210m のマップの場合には隣接関係の計算に 7496ms を費やしている。この実験ではカメラのパン／チルト／ズームによるセンサの検出範囲の変更が 30 秒に 1 回の割合でランダムに発生するため、検出範囲の変化による隣接関係の更新が人物の移動に追い付かず、見逃しが発生する。一方で、同じサイズのマップで局所化したときには、隣接関係の計算には平均で 0.192ms で計算することができている。つまり、監視領域が広がっても人物の追跡を継続するのに支障のない時間で隣接関係を計算することができる。

#### 4.6.2 追跡継続性の確認

124.5m × 51m の監視エリア（図 4.4）に初期状態として 18 台のカメラを設置することを想定し、継続的に人物が追跡できているかを確認した。

追跡対象は入口から監視エリア内に入り、1.5m/秒から 3.0m/秒の速度で監視エリア内のランダムな 1 地点に向かった後、出口から監視エリア外に移動する。移動速度は宮原らの調査結果<sup>64)</sup>による人物の移動速度が、1.7m/秒から 2.7m/秒であることを参考に設定した。監視エリア内には 3 人の追跡対象が出現し、追跡対象が監視エリア外に移動した場合には新たな追跡対象が入口から入ってくるものとした。撮影範囲の変更は 30 秒に 1 回、カメラの追加・削除は 8 時間に 1 回の頻度で発生するものとした。追跡中に撮影範囲の変化が発生した時とカメラの追加と

表 4.2: 局所化したときの点の数

マップの広さ [m <sup>2</sup> ]	Xsに含まれる センサ点の数			Xs/Ysに含まれる 全ての点の数		
	平均	最大	最小	平均	最大	最小
30 x 30	6	8	5	34	35	52
60 x 60	12	20	5	75	138	25
90 x 90	12	26	4	77	162	25
120 x 120	16	33	3	99	213	19
150 x 150	16	40	3	100	262	17
180 x 180	16	41	4	95	256	21
210 x 210	16	53	4	95	330	25

表 4.3: 測定に用いた計算機

CPU	Intel Core i7-4770
メモリ	8GB

削除が発生したときの結果を図 4.5 に抜粋する。

縦軸をカメラ番号、横軸を経過時間とし、追跡対象人物（ターゲット）の移動を点線、エージェントによる追跡結果を実線で示す。ターゲットがカメラの撮影範囲内に存在する間は該当するカメラ番号上の線として表現し、カメラの撮影範囲を離れ次のカメラの撮影範囲に移動するまでの間を斜線として表現している。エージェントの線はエージェントが存在するカメラの場所を表すため、最後にターゲットが検出されたカメラ番号上に存在する。また、パン・チルト・ズームの発生時を◇印、カメラの追加時を○印、削除時を×印で示す。

初期状態では、カメラ C6 を通過する追跡対象は、カメラ C1、カメラ C10、カメラ C2 の順で検出されていた。このことはターゲット A がカメラ C6 で検出後、カメラ C1、カメラ C10 の順で検出されていることから確認できる。その後、経過時間が 225 秒の時にカメラ C6 の撮影範囲がカメラ C1 とカメラ C10 の撮影範囲を跨ぐように変化した。その結果、カメラ C6 で検出されたターゲット B が次にカメラ C2 で検出されていることが確認できる。このことから、撮影範囲の変化に対応し、隣接関係が正しく計算できていることが分かる。

表 4.4: 隣接関係の計算にかかる時間

マップの広さ [m <sup>2</sup> ]	局所化有り 平均計算時間 [ms]	局所化無し 平均計算時間 [ms]
30 x 30	0.054	0.192
60 x 60	0.064	0.735
90 x 90	0.091	9.268
120 x 120	0.128	54.815
150 x 150	0.117	216.471
180 x 180	0.130	1299.056
210 x 210	0.192	7945.471

また、ターゲット C はカメラ C4 で検出され次にカメラ C5 で検出され、その後カメラ C6 で検出されている。その後、経過時間が 3337 秒の時にカメラ C5 が削除されている。カメラ C5 の削除後、同じ経路を通ったターゲット D はカメラ C4 で検出された後、カメラ C6 で検出された。この結果、カメラ C5 が削除された後も隣接関係が正しく計算されていることが確認できる。

また、経過時間 3380 秒でカメラ C19 が追加されている。カメラ C19 の追加前までは追跡対象がカメラ C4 で検出後、カメラ C6 で検出されていた。しかし、カメラ C19 の追加後、ターゲット E はカメラ C4 で検出後、カメラ C19 で検出されている。カメラ C19 が追加された場所を確認すると、カメラ C4 とカメラ C6 の間であった。これにより、C4 の隣接カメラが C19 に更新され、カメラの追加に対応できていたことが確認できた。

本実験を 72 時間分シミュレーションした結果、ターゲットが 8892 人通過し、撮影範囲の変化が合計 155505 回、カメラの追加と削除がそれぞれ 9 回発生したが、ターゲットが通過した撮影範囲に該当するカメラを結ぶ経路と該当するターゲットを追跡するエージェントが移動したカメラを結ぶ経路が、全てのターゲットにおいて一致することが確認できた。

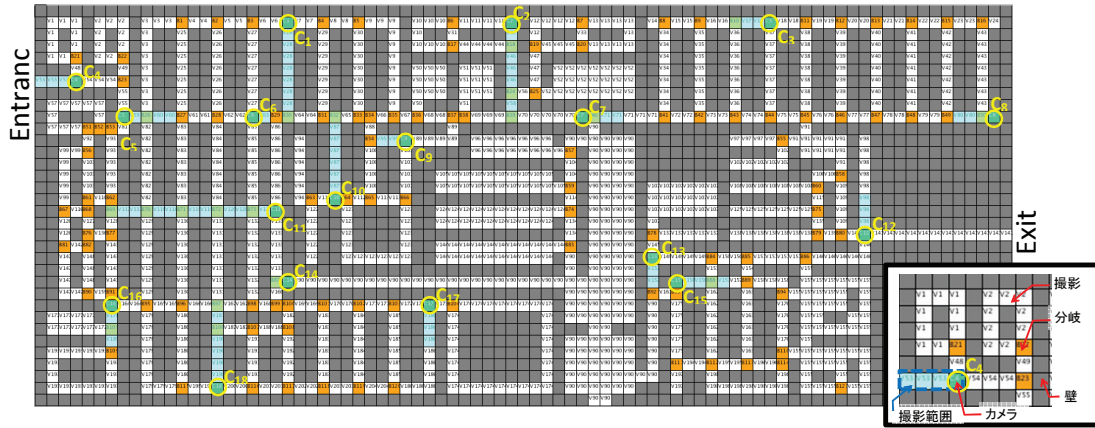


図 4.4: シミュレーション環境のマップ

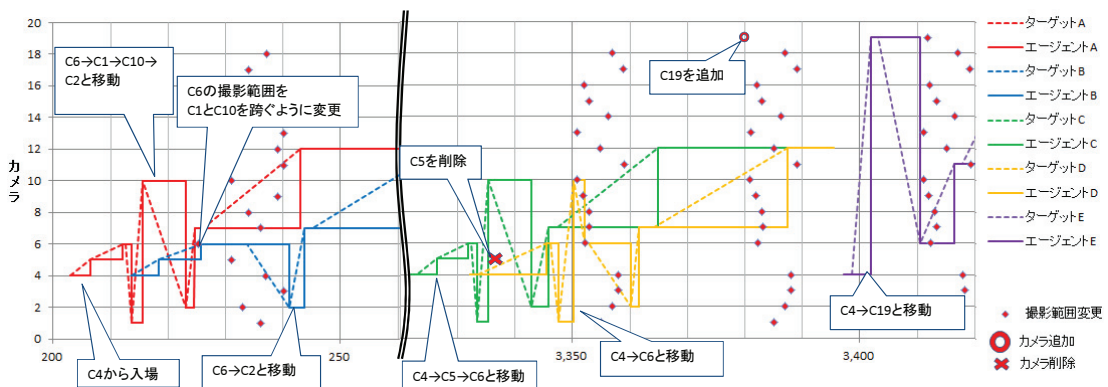


図 4.5: 追跡実験の結果

## 4.7 結言

3.1節で述べた手法では、センサの検出範囲を元に隣接センサを計算することで検出した人物の追跡範囲を限定するため、検出範囲が変更される場合やセンサの追加や削除が行われた場合にシステム内の全てのセンサの検出範囲を取得し隣接関係を再計算しなければならないという問題があった。

そこで、本節では隣接関係の計算を局所化する手法を提案した。本提案手法では、各センサがシステム内の全てのセンサの情報を持たず、隣接する可能性のある局所化されたセンサの情報のみを持つことにより、検出範囲の変更やセンサの追加と削除の影響を局所化し人物の追跡を継続することができる。本提案手法を実装しシミュレーションによる実験を行った結果、センサの検出範囲や隣接関係の変化が発生しても隣接関係が更新される範囲が局所化された限られた範囲に収まることが確認できた。

## 第5章 センサの不確実性に対応した 人物追跡手法

### 5.1 序言

センサの検出範囲を人物が通過したとしても、必ずしもそのセンサが人物を確実に検出できるとは限らない。例えば図5.1のイメージでは、カメラの撮影方向によって、真正面からは撮影対象を人物として検出できるが、横や上から撮影した場合に人物と認識しない場合がある。また、カメラによる人物が何者であるかの識別は、まずカメラの撮影領域から人物の映っている範囲を切り出してから人物の同定を行う場合が一般的であり、人物が映っていると認識しなければ人物の識別も行われない。

また、ビーコン等のセンサにおいても、棚などの遮蔽物により電波が遮られることによりビーコンを検出しないことがあり、また対象の人物がビーコンをどのように所持しているかによって、センサに正対している場合はよく検出するが、反対を向いている場合は検出しにくいといったことが考えられる。このようにセンサが100%確実に人物を検出できるわけではないことを、センサの不確実性と定義する。

4章までに述べた手法においても、センサ自体の不確実性に由来する見逃しが発生することにより人物の追跡が失敗するという問題があった。そこで本稿では、センサの検出結果から見逃した追跡経路を見つけ出すことでセンサの不確実性に対処する手法を提案する。

センサの不確実性により人物を検出できなかった例を示す。図5.2は遠藤らの研究<sup>65)</sup>においてビーコンを用いた追跡システムでセンサが未検出した例である。このシステムでは、ビーコンを使って人物の追跡を行う。ビーコンから受信する電波が一定の閾値を超えると、センサはビーコンを所持した人物がセンサの付近にいると判断する。人物は点線の矢印の経路で移動する。この図の例では、センサ

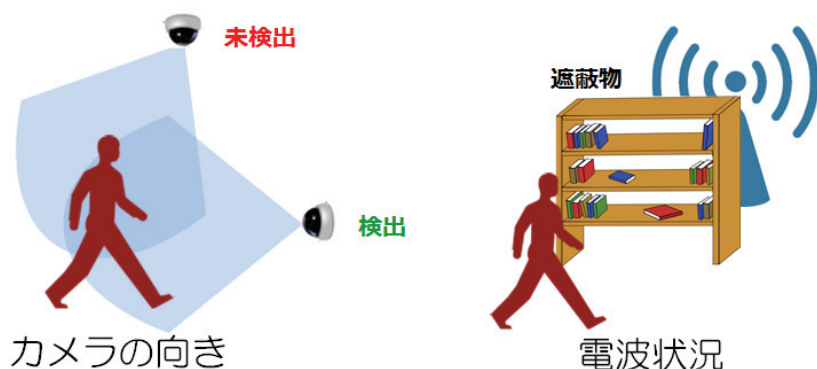


図 5.1: センサの未検出のイメージ

2とセンサ13で未検出が発生していることが分かる。このときにシステムが判断する移動経路は本来であれば、センサ1からセンサ2を経由しセンサ3と移動したところをセンサ1からセンサ3へ移動したと認識し、センサ6からセンサ13を経由しセンサ7と移動したところをセンサ6からセンサ7と移動したと認識する。

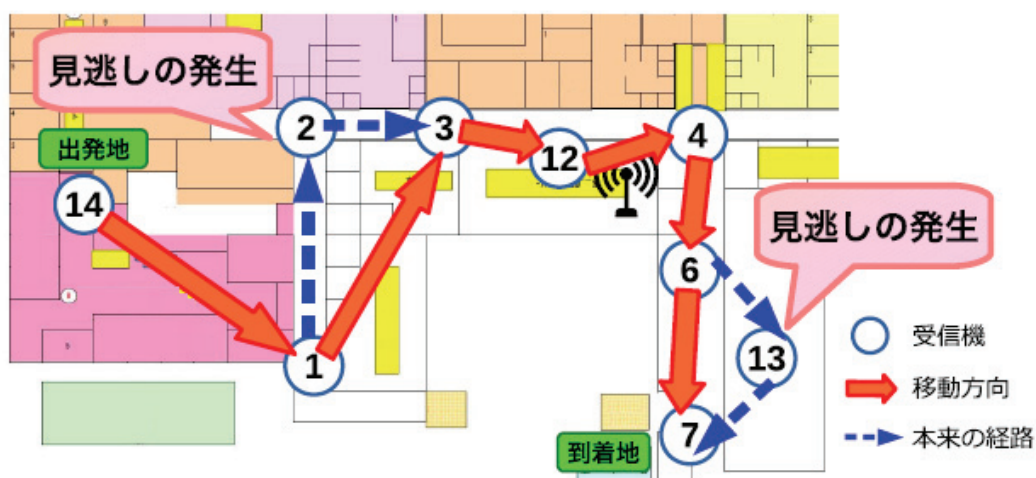


図 5.2: 見逃しの発生

この実験では検出した情報は中央サーバに集積されるが、分散環境で3章および4章で述べたアルゴリズムを用いる場合は、モバイルエージェントを派遣したノードで人物が検出されないため、見逃しにつながる。

## 5.2 隠れ隣接関係の導出による問題解決

### 5.2.1 センサの未検出への対応

センサが人物を確実に検出できていれば、本システムで人物を追跡し続けることができる。しかし、センサの不確実性により、あるセンサで検出した人物が隣接関係にあるセンサで検出されず、遠く離れたセンサで検出されることが起こり得る。例えば図5.2の例では、センサ1とセンサ3が隣接関係を持たない場合、センサ1からセンサ2へモバイルエージェントが派遣されるが、人物はセンサ2で検出されずにモバイルエージェントの派遣されていないセンサ3で検出されるため、追跡の継続に失敗する。

センサがどの程度確実に人物を認識できるかについては様々な研究成果がある。例えば、監視カメラの映像からの個人認証の一位認識率が単一視点の顔認証の場合で90.39%、歩容認証の場合で91.14%であるとの研究成果がある<sup>66)</sup>。更にこの研究では顔特徴、歩容特徴、身長特徴を組み合わせたマルチモーダル特徴による個人認証で99.17%の一位認識率を得ている。また、近年はビーコンを用いる屋内測位手法に関する多くの研究が進んでいるが、ビーコンの保持方法や移動方向による測位誤差が見られることが指摘されている<sup>67)</sup>。このように、カメラやビーコンを用いて人物を追跡するシステムにおいて、人物を常に100%の確率で検出できるとは限らない。

そこでセンサの不確実性に対応し、センサの検出結果に応じた新たな隣接関係を見つけ出す手法を提案する。表5.1にセンサの検出結果と現実の事象の分析結果を示す。ここで、“正常”は追跡中の人物を正しく検出できた場合、“未検出”は追跡中の人物を検出できなかった場合、“誤検出”は追跡中の人物と異なる人物を検出した場合を示している。未検出と誤検出が含まれている場合に隣接関係に基づいた人物の追跡に失敗する可能性がある。ただしビーコンやRFIDのセンサを対象としたシステムでは誤検出は発生しない。本稿ではビーコンやRFIDの利用を想定し、未検出時の対応のみに焦点を当てる。



表 5.1: センサの検出結果と現実の事象の分析結果

現実の事象 センサ	人物は存在せず	人物Aが存在	人物Bが存在	人物AとBが存在
検出せず	正常	未検出	未検出	未検出
人物Aを検出	誤検出	正常	誤検出	正常+未検出(※1)
人物Bを検出	誤検出	誤検出	正常	正常+未検出(※1)
人物AとBを検出	誤検出	正常+誤検出(※2)	正常+誤検出(※2)	正常

※1 1人の人物が未検出であり、未検出の人物の追跡に失敗する。  
 ※2 追跡中の人物を正常に検出しており、追跡は継続できる。

### 5.2.2 未検出への対応

未検出の場合、人物の見逃しが発生し追跡に失敗する。このことを解決するためのもっとも単純な方法は、全てのノードで追跡対象人物を監視することである。しかし、センサと追跡対象の人物が増える程、追跡エージェントの数が増加し各ノードの負担が重くなる。エージェントを用いた人物追跡における研究<sup>29)</sup>では追跡エージェントが人物を見逃した場合に、隣接したノードを辿りエージェントを派遣していくことで検出範囲を広げる方法が提案されている。この方法では追跡対象の人物を検出した時点で探索を終えるため、全ノードを常に監視する方法と比較するとシステムへの負担は限られる。しかし、見逃したことをどのように検出するのか、見逃した場合に探索範囲をどこまで広げるのかといった課題が存在する。そこで、エージェントに追跡されていない人物が検出された時に仮追跡する方法を提案する。図 5.3 に仮追跡手法のイメージを示す。

各追跡サーバにてエージェントが追跡していない人物を検出した場合、その人物を追跡するためのエージェントを生成する。ここで、元々人物を追跡していたエージェントとここで生成されたエージェントを区別するために、元々人物を追跡していたエージェントを本追跡エージェント、ここで生成されたエージェントを仮追跡エージェントと呼ぶこととする。<sup>1</sup>

<sup>1</sup> エージェント自身に本追跡／仮追跡の区別はなく、追跡を開始した時刻が古い方のエージェントを本追跡エージェント、新しい方のエージェントを仮追跡エージェントと相対的な関係で決定する。

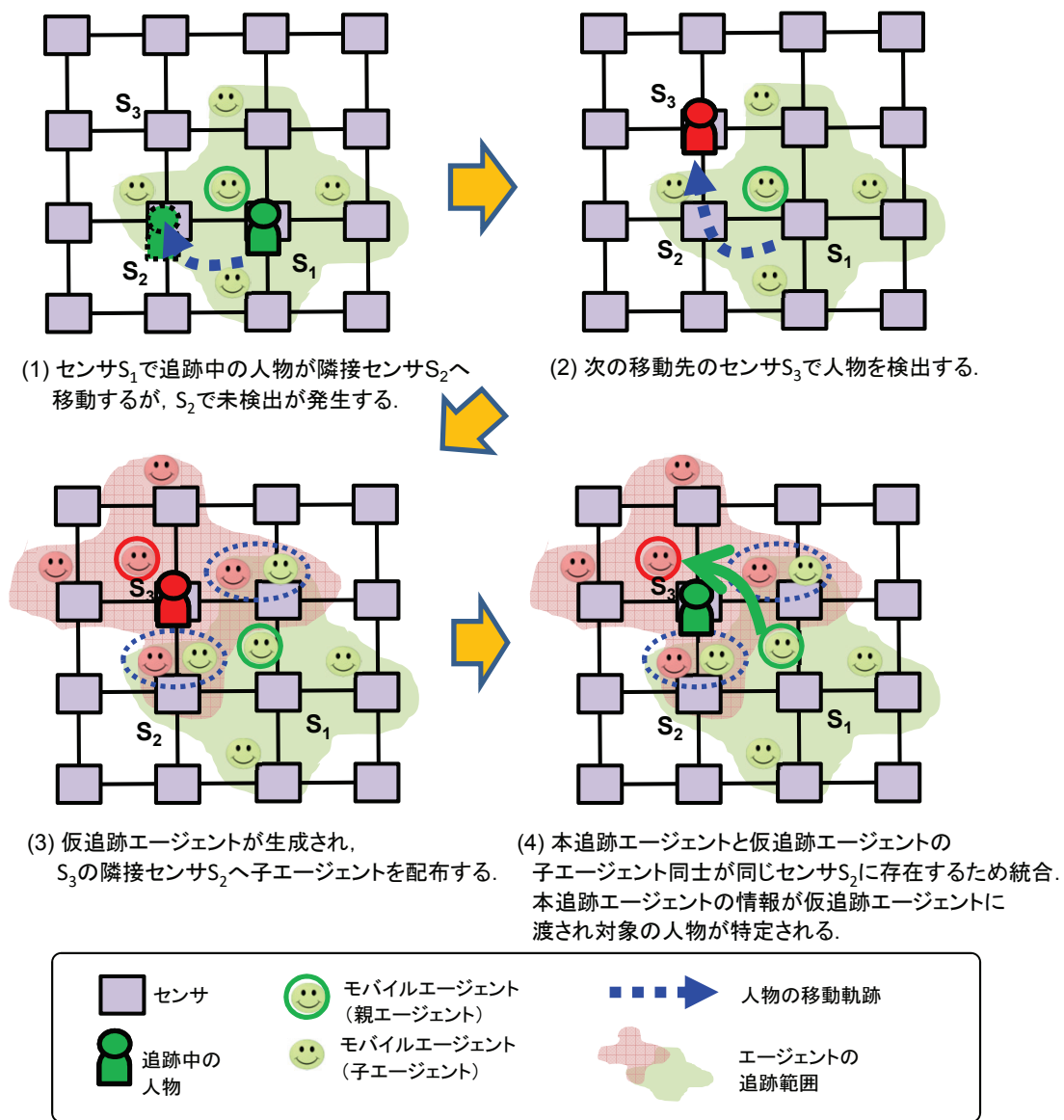


図 5.3: 未検出への対応

## 第5章 センサの不確実性に対応した人物追跡手法

未検出への対応は3節の図3.2で示した子エージェントを派遣する処理の前に、図5.4の処理を追加することで行う。エージェントは自身が派遣されたノードにおいて動作しているエージェントに対して、自身が追跡する人物の特徴情報と追跡開始時刻を送付する。それらを受け取ったエージェントは、自身の追跡開始時刻が受け取った追跡開始時刻より古く、かつ、同一の人物を追跡している場合、本追跡エージェントであることを返信する。これにより、仮追跡エージェントと本追跡エージェントが確定する。これ以降の処理は、子エージェントから親エージェントにメッセージが転送され、親エージェント同士で行われる、本追跡エージェントの存在を確認した仮親エージェントは、本親エージェントに統合を要求する。本親エージェントは、自身が持つ人物の追跡履歴を仮親エージェントに渡す。仮親エージェントは、受け取った追跡履歴を自身が持つ追跡履歴と統合し、統合が完了したことを本親エージェントに通知する。通知を受けた本親エージェントは、本子エージェントの動作を停止すると共に自身も動作を停止する。これによりセンサの未検出に対応する。

仮追跡を行うことにより、センサの未検出が多い環境では仮追跡エージェントが頻繁に生成され、統合されずに残り続ける可能性がある。見逃しや故障等により監視ノードにエージェントが残り続けることを回避するために、一定時間人物を検出できなかった場合はエージェントを消去する。

第5章 センサの不確実性に対応した人物追跡手法

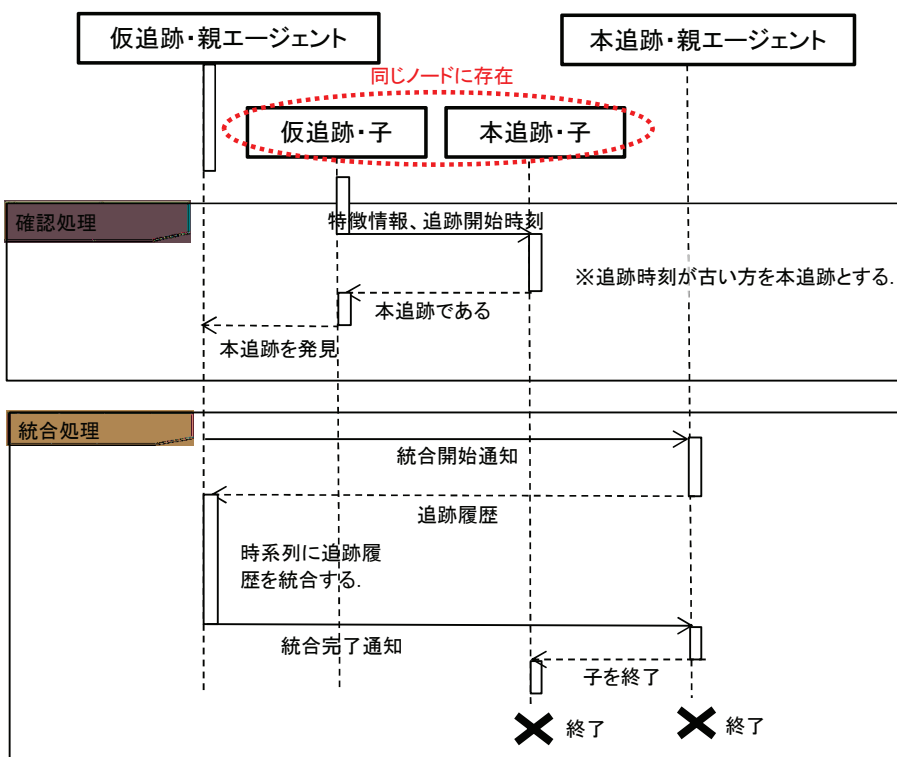


図 5.4: 統合のために追加する処理

### 5.2.3 隠れ隣接関係の導出

仮追跡エージェントが仮追跡している人物が、本追跡エージェントが追跡している人物と一致した場合、その人物は本追跡エージェントが最後に人物を検出したセンサの隣接センサにて未検出が発生し、その後仮追跡エージェントが追跡を始めたセンサで検出されたことを示す。図 5.2 のビーコンを用いた予備実験でもこのような未検出が特定のセンサで頻発することを確認している。これは、センサの設置位置とセンサ周辺の遮蔽物が電波の感度に影響を与えており、そのために発生しているものと思われる。

図 5.3 の例では一つのセンサ ( $S_2$ ) で未検出が発生している。この結果、仮追跡子エージェントと本追跡子エージェントが同じセンサ ( $S_2$ ) に派遣され、これらのエージェントを統合することで追跡を継続できている。このため、仮追跡エージェントの生成と、仮追跡エージェントと本追跡エージェントを統合するための負荷が発生する。また、2つ以上のセンサで未検出が発生した場合、同じ人物を追跡する仮追跡エージェントと本追跡エージェントが、偶然出会った場合にのみ追跡が継続できる。

そこで、これらの負荷や偶然性を軽減するために、隠れ隣接関係を導入する。仮追跡エージェントと本追跡エージェントの統合が発生したときに、本追跡エージェントが最後に人物を検出したセンサと、仮追跡エージェントが最初に人物を検出したセンサの間に隠れ隣接関係が存在するものとして、隣接関係を更新する。図 5.5 に隠れ隣接関係が構築される例を示す。

図 5.5 の例ではセンサ A まで追跡できていた人物を見失い、センサ B から人物の仮追跡が始まり、その後本追跡エージェントと仮追跡エージェントの統合が発生している。これは、センサ A とセンサ B の間に不確実性の高い1つ以上のセンサが存在していることを表す。そこで、それらのセンサ（センサ A とセンサ B 間）に隠れ隣接関係があるものとみなす。これにより未検出を発生しやすい不確実性の高いセンサをバイパスして追跡できるようになり、以降それらのセンサが人物を検出しなくても追跡を継続することができる。

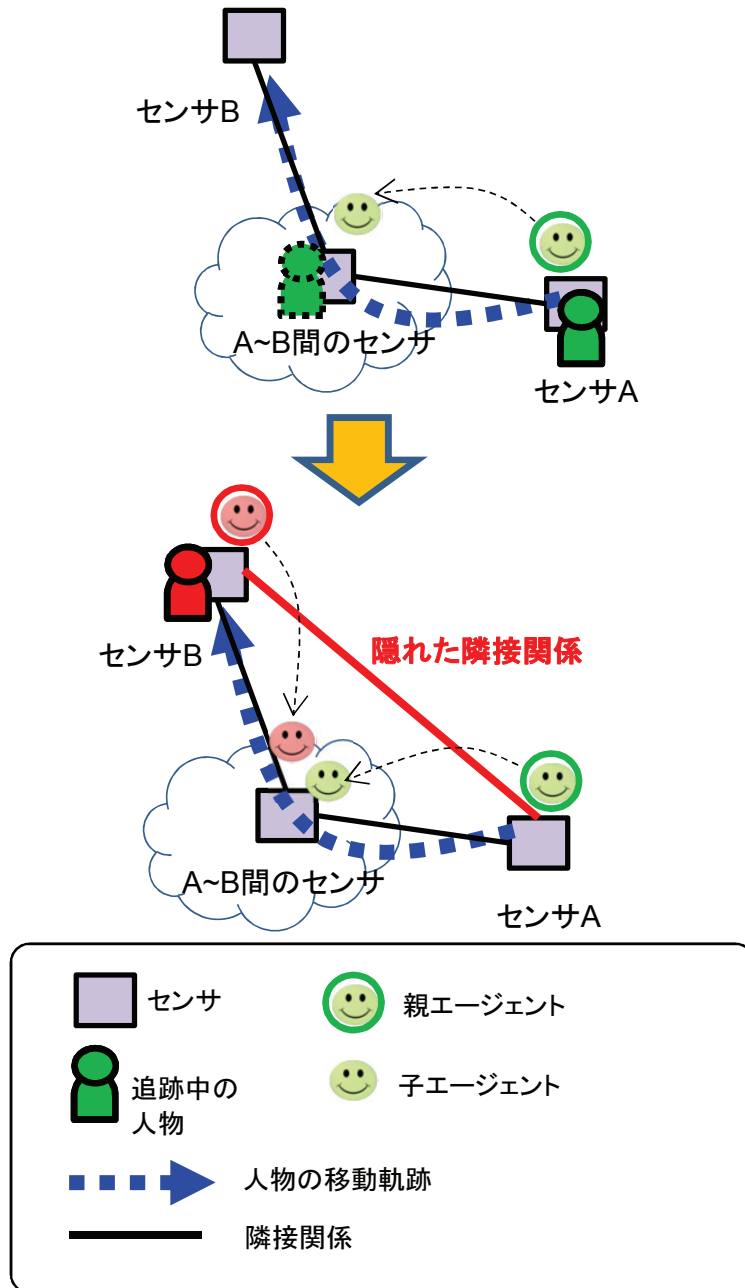


図 5.5: 隠れ隣接関係の導出

### 5.2.4 隠れ隣接関係の反映

3節の隣接関係の計算を隠れ隣接関係を求められるように修正する必要がある。隣接関係の計算では行列  $X_s, Y_s$  を利用する。行列  $Y_s$  はセンサ間の接続関係を表しており、隠れ隣接関係を計算できるようにするためには、未検出が発生する不確実性の高いセンサをバイパスできるような接続関係に更新する必要がある。

しかし、各センサは局所化された範囲の点しか管理せず、隠れ隣接関係を行列  $Y_s$  に追加することはできない。そこで、隠れ隣接関係を構築するセンサ間で、それぞれのセンサが持つ行列  $X_s, Y_s$  を通知し合い、自身が持つ行列  $X_s, Y_s$  と統合する。図 5.5 の例であれば、センサ A はセンサ A が持つ行列  $X_{sa}, Y_{sa}$  をセンサ B に通知し、センサ B は自身が持つ行列  $X_{sb}, Y_{sb}$  と行列  $X_{sa}, Y_{sa}$  を統合する。統合した行列を  $X_{sab}, Y_{sab}$  とする。センサ B からセンサ A へも同様な操作を行い行列  $X_{sab}, Y_{sab}$  を生成する。統合した行列  $Y_{sab}$  に対して、

$$Y_{sab_{ij}} = 1, \text{ただし, } i=\text{センサ A, } j=\text{センサ B}$$

に変更することで、センサ A とセンサ B が隠れ隣接関係を持つものとして更新する。これにより隠れ隣接関係を計算できるようになる。隣接関係を持つ監視ノード間では、お互いにアクセス可能かを相互監視する。予め隣接する監視ノードが持つ行列情報を保持しておくことにより、隣接する監視ノードの消滅を感知した場合に、これらの行列情報を用いて隣接関係を再構築する。

### 5.2.5 隣接関係の広がりの抑制

本手法では隠れ隣接関係を反映することによって、1つのノードが持つ隣接センサ数が増加する。このため、未検出を起こすセンサが増加するほど隣接関係が増加する。例えば、複数のセンサが連続して未検出を起こし、それが繰り返される場合、複数の仮追跡が行われ、それらの時系列順に隠れ隣接関係が構築されるとは限らない。このような稀に発生するケースにより生成された隠れ隣接関係や、ほとんど未検出を起こさないセンサをバイパスする隠れ隣接関係が残り続けることにより、隠れ隣接関係が増加する。

そこで、隠れ隣接関係を利用した追跡が行われなかった場合に、閾値を超えた

## 第5章 センサの不確実性に対応した人物追跡手法

隠れ隣接関係を削除することにより隠れ隣接関係の増加を抑える。閾値として各ノードにおける隠れ隣接関係の上限値を設定し、上限値を超えた隠れ隣接関係はLRU (Least Recently Used), もしくはLFU (Least Frequently Used) により削除する。

未検出を起こす確率が高いセンサは隠れ隣接関係を利用することで追跡が継続される一方、ほとんど未検出を起こさないセンサをバイパスする隠れ隣接関係は削除される。利用されない隠れ隣接関係を削除することで、隠れ隣接関係の増加を抑制しつつ、追跡成功率を高めることができる。



### 5.3 実験

提案手法の有効性を検証するためのシミュレーション実験を行った。実験では監視エリアのマップを作成し、そこに複数台のセンサが設置されていることを想定して行った。シミュレーションでは図 5.6 に示した 16.5m × 13.5m のマップに 10 台のセンサを配置した。追跡対象は入口から監視エリア内に入り、監視エリア内のランダムな 1 地点に向かった後、出口から監視エリア外に移動する。

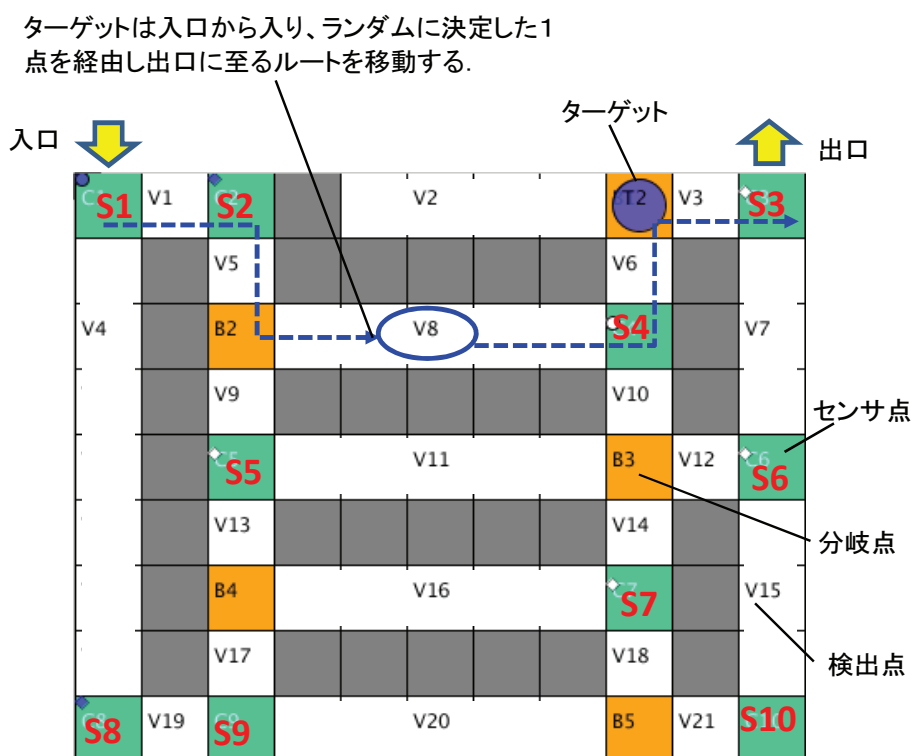


図 5.6: シミュレーションマップ

#### 5.3.1 仮追跡の動作確認

マップ中の 1 台のセンサ S2 が 60% の割合で人物を検出し、他のセンサは 100% の割合で人物を検出する条件で、ターゲットの移動に対応してエージェントが人物を追跡できているかを確認した。エージェントが人物を検出した結果の例を図 5.7 に示す。縦軸をセンサ番号、横軸を経過時間とし、追跡対象人物の移動を点線、エー

エージェントによる追跡結果を実線で示す。

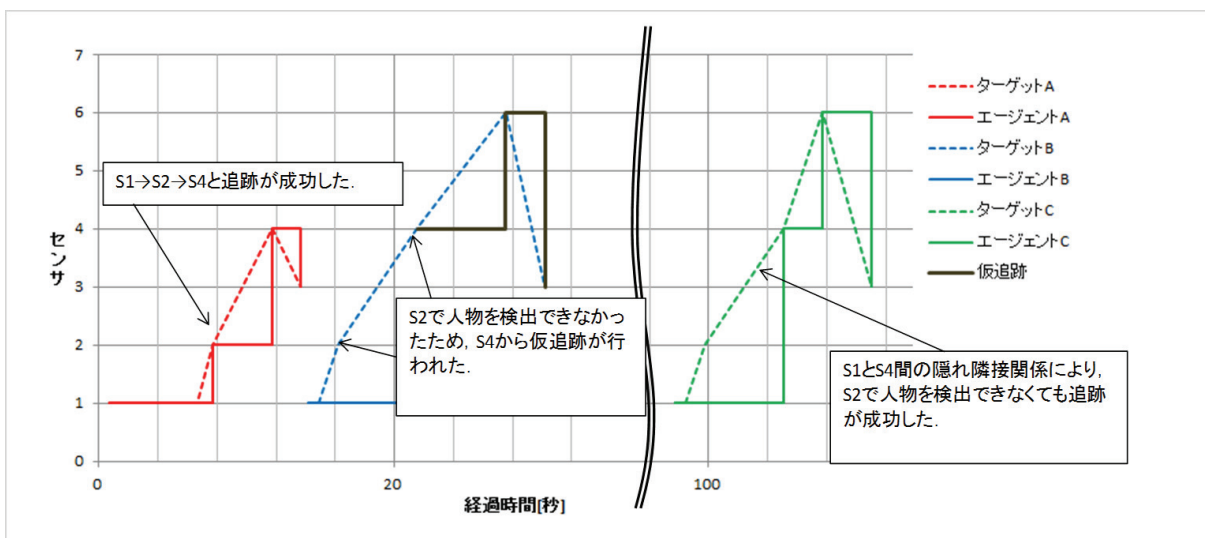


図 5.7: シミュレーション結果

初期状態では、S1とS2、S2とS4の隣接関係が存在する。実験開始直後に移動を開始したターゲットAはS1で検出されエージェントによる追跡が始まる。経過時間8秒のときにS2で検出され、経過時間12秒のときにS4で検出された。この間エージェントによる追跡は成功した。一方、経過時間13秒から移動を開始したターゲットBは経過時間16秒でS2を通過するがS2のセンサは人物を検出せず、エージェントによる追跡に失敗した。ターゲットBはその後、経過時間22秒のときにS4にて検出され、S4から仮追跡が開始された。このとき、S1に存在する本追跡エージェントの子エージェントがS2に、S4に存在する仮追跡エージェントの子エージェントがS2に存在するため、本追跡エージェントと仮追跡エージェントが統合され、S1とS4の間に隠れ隣接関係が発見された。

その後、経過時間97秒から移動を開始したターゲットCは、ターゲットBと同様に経過時間100秒でS2にて未検出を起こしている。しかし、ここではS1とS4の間の隠れ隣接関係によりS4はS1の隣接センサとなり、この結果、ターゲットCの追跡に成功した。

### 5.3.2 隠れ隣接関係による追跡成功率の変化

隠れ隣接関係の構築による各ノードが持つ隣接センサの数の広がり进行评估した。実験では  $31.5\text{m} \times 31.5\text{m}$  の広さの格子上的通路が存在するフロアに  $4.5\text{m}$  置きに 61 個のセンサが均等に配置されている条件で実施した。入口と出口のセンサは 100% で人物を検出することとし、他のセンサはあらかじめランダムに決めた 50 ~ 100% の確率で人物の検出に成功する条件で人物追跡の成功率を評価した。入口のセンサで追跡を始めた人物が出口のセンサに到達した時に、人物が入口のセンサで検出した人物と同一人物であることをエージェントにより特定できているかで追跡可否を判断することとした。隠れ隣接関係を反映する場合と、隠れ隣接関係を反映しない場合で比較する実験を行った結果を図 5.8 に示す。

図 5.8 は、横軸に経過時間を示し、縦軸に最新 30 分間の平均の追跡成功率を示す。隠れ隣接関係を更新しない場合ではセンサが人物を検出できない場合に追跡に失敗するため、出口到達時点における追跡成功率が 10% に満たない結果となった。センサの検出確率は 50 ~ 100% であるため、経路上のいずれかのセンサで未検出を起こし、最後まで追跡できていないことを示している。一方で隠れ隣接関係を更新する場合では人物追跡の成功率が 95% 前後に上昇した。

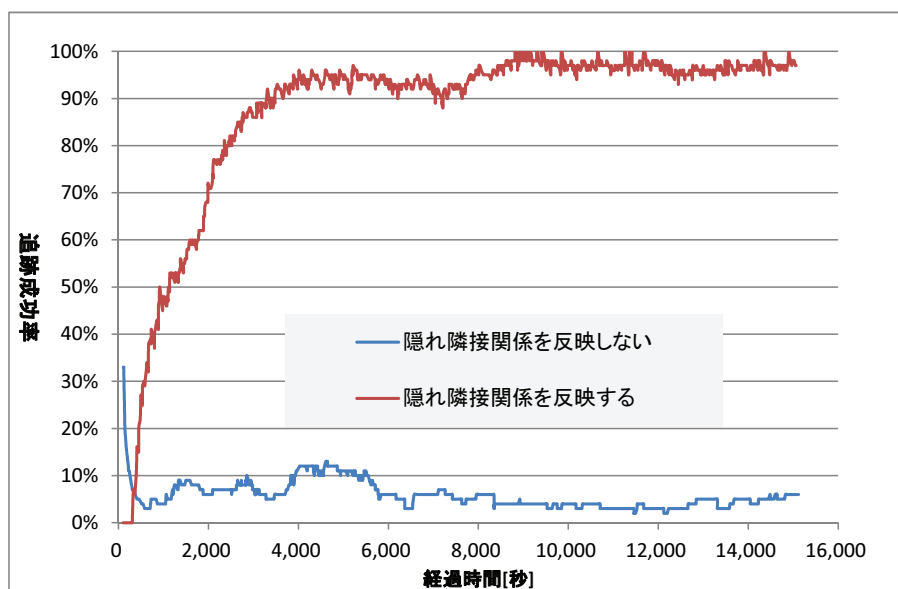


図 5.8: 追跡成功率

また、このときの各ノードが持つ隣接センサ数の平均値を図 5.9 に示す。隠れ

隣接関係を更新しない場合は一定の隣接ノード数となるが、隠れ隣接関係を更新する場合は隣接センサの数の平均が増加し続けた。

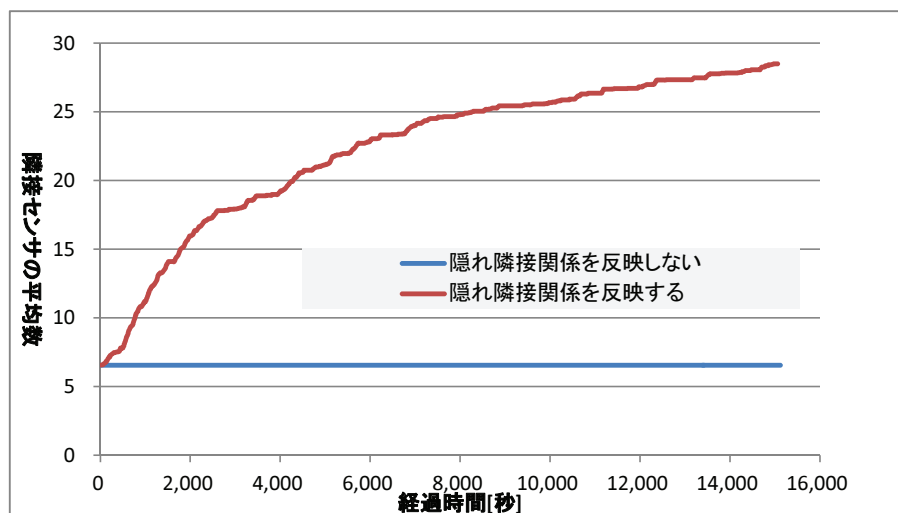


図 5.9: 隣接センサ数の平均

数回実験した結果を図 5.10 に示す。同じ傾向であることが確認できた。

センサの検出率を 20~50%としたときの結果を図 5.11, 図 5.12 に、全てのセンサの検出率を 90%としたときの結果を図 5.13, 図 5.14 に示す。

### 5.3.3 隠れ隣接関係の削除方式の比較

LRU と LFU により隠れ隣接関係を削除した場合の追跡成功率と隣接センサ数の平均数を評価した。図 5.15 は追跡成功率を、図 5.16 は隣接センサ数の平均数を示している。閾値（各ノードにおける隠れ隣接関係の上限値）は 8 とした。図 5.15 を見ると LFU の方が LRU より追跡成功率が高い結果となっている。また、図 5.16 を見ると LFU の方が LRU より平均隣接センサ数が少ない結果となっている。

これより、LRU より LFU で隠れ隣接関係を削除した方が効率がよい結果が得られることがわかる。

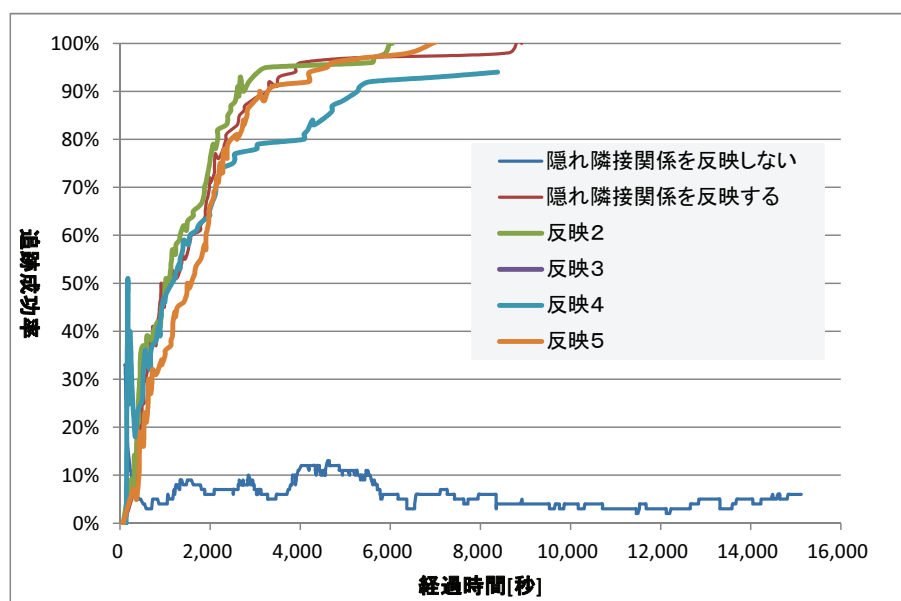


図 5.10: 複数回の実験結果 (追跡成功率)

### 5.3.4 閾値による追跡成功率の比較

LFUにより隠れ隣接関係を削除する閾値を変更したときの追跡成功率を評価した。閾値を0, 1, 2, 4, 8, 16, 閾値なし, としたときの実験結果を図5.17に示す。ここで閾値が0の場合は隠れ隣接関係が構築されないことを表す。図5.17を見ると、閾値が高くなると共に追跡成功率が上昇し、閾値が16の場合で閾値なしの場合と同程度の追跡成功率となった。

実験結果から、隠れ隣接関係を無制限に増やすことなく、高い追跡成功率が得られることがわかった。

第5章 センサの不確実性に対応した人物追跡手法

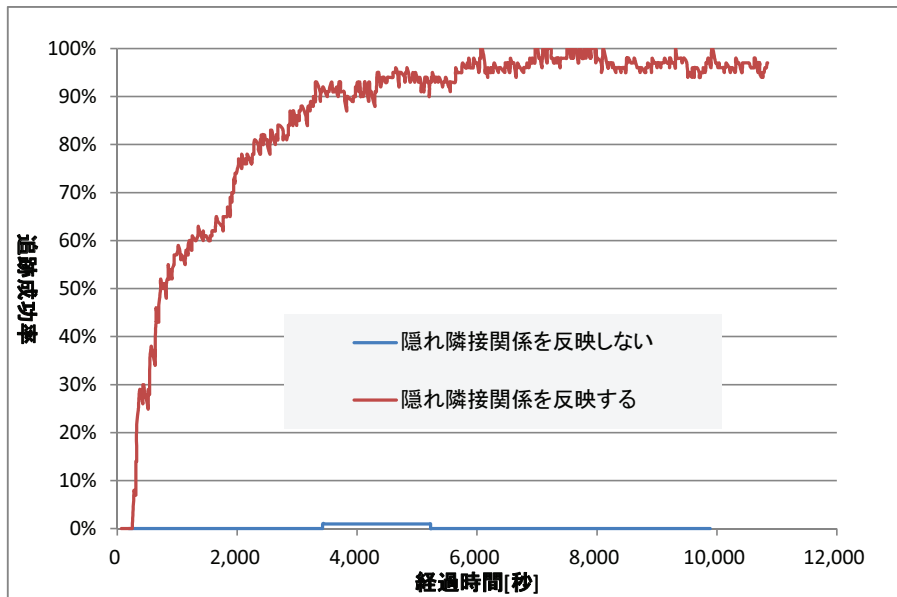


図 5.11: 追跡成功率 (センサ検出率 20~50%)

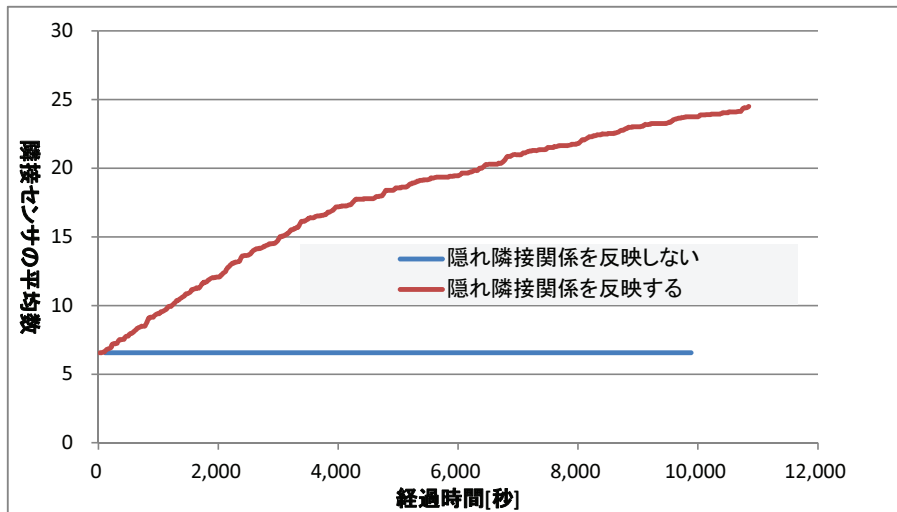


図 5.12: 隣接センサ数の平均 (センサ検出率 20~50%)

第5章 センサの不確実性に対応した人物追跡手法

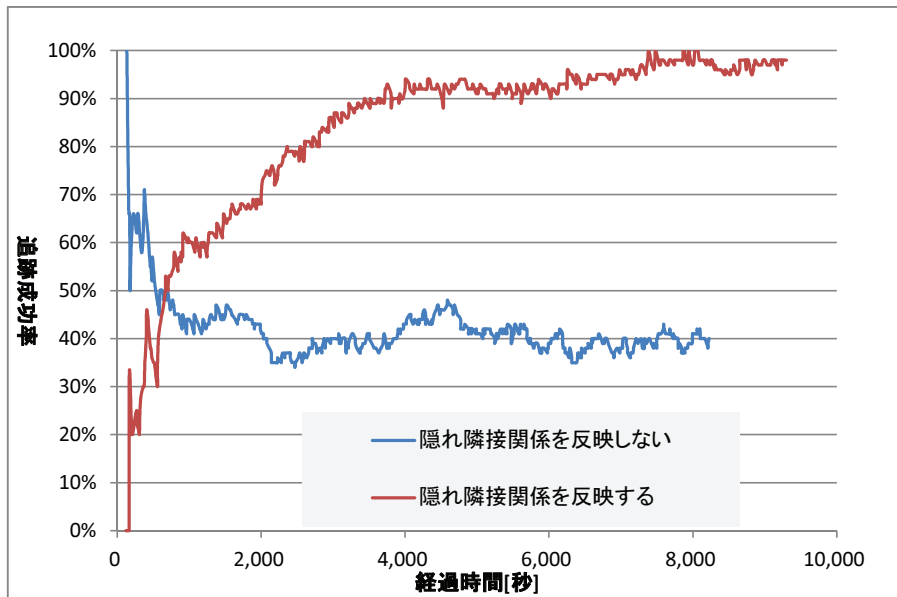


図 5.13: 追跡成功率 (センサ検出率 90%)

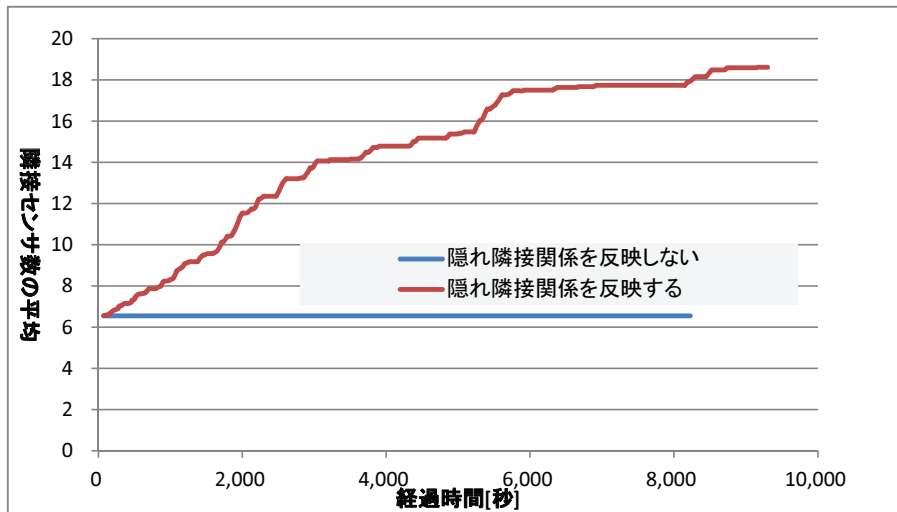


図 5.14: 隣接センサ数の平均 (センサ検出率 90%)

## 第5章 センサの不確実性に対応した人物追跡手法

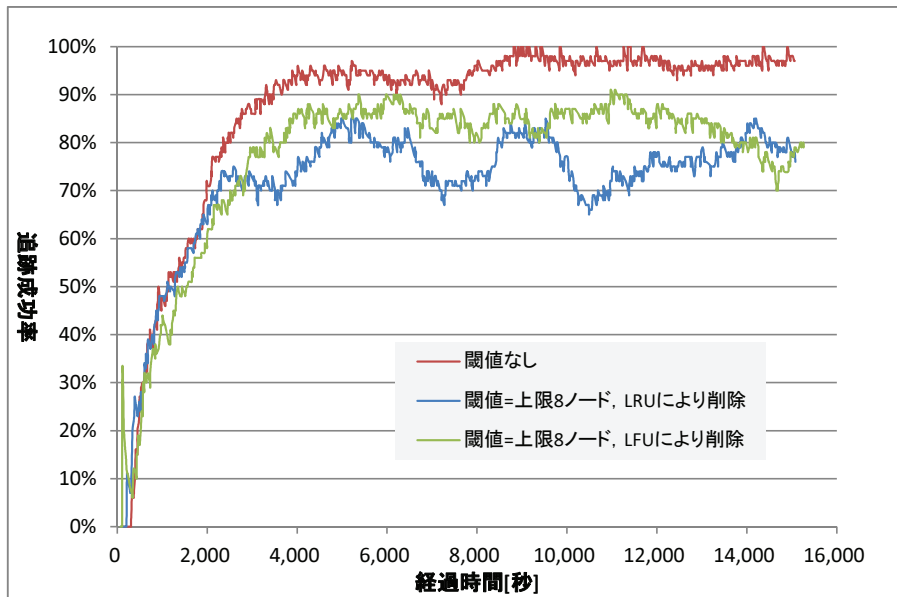


図 5.15: 追跡成功率 (削除方式による比較)

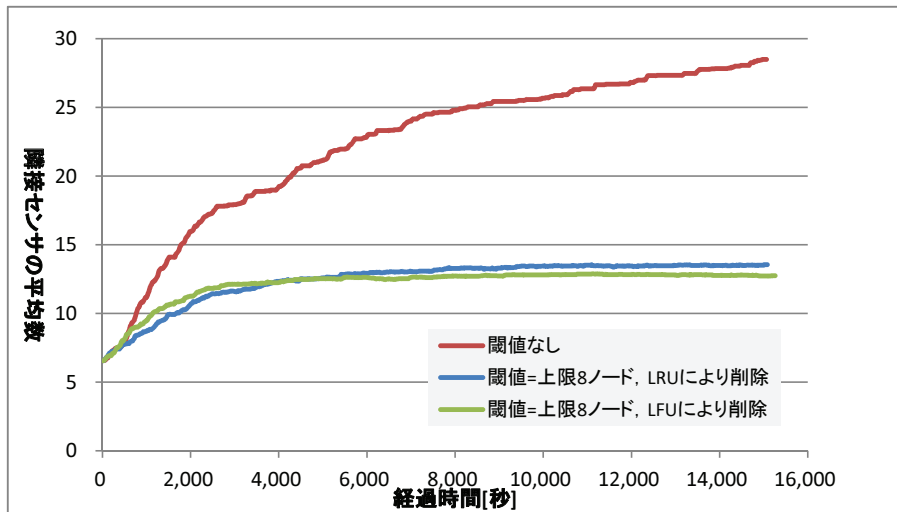


図 5.16: 隣接センサ数の平均 (削除方式による比較)



第5章 センサの不確実性に対応した人物追跡手法

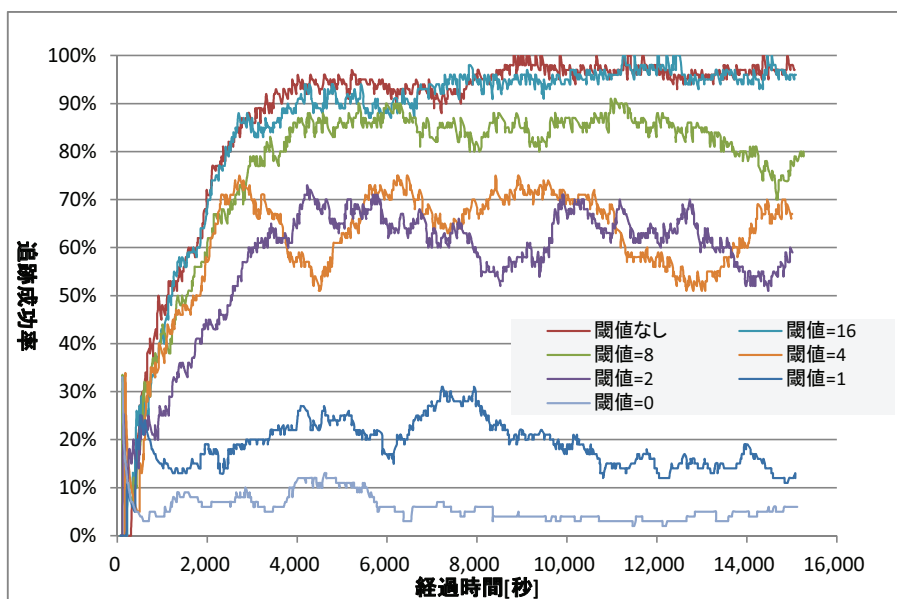


図 5.17: 追跡成功率（閾値による比較）

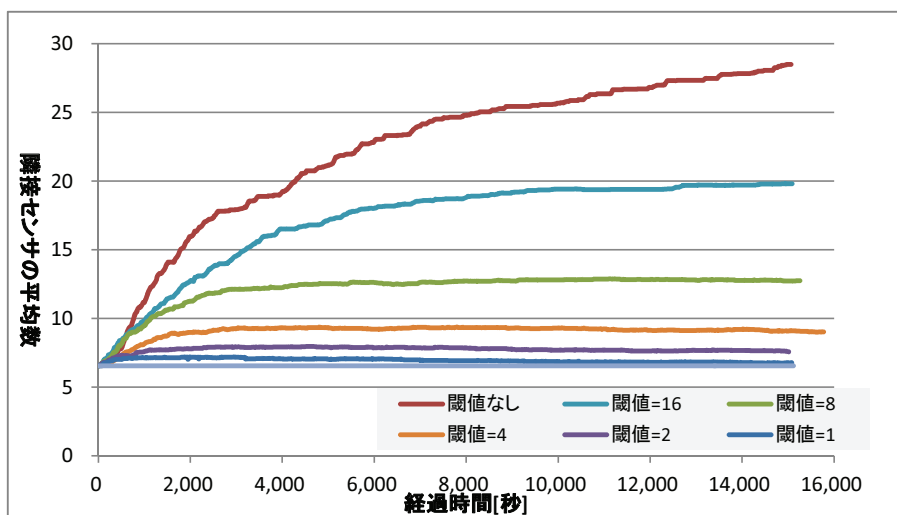


図 5.18: 隣接センサ数の平均（閾値による比較）

## 5.4 センサの検出率が低い場合における考察

各センサの検出率が低い場合、本手法により新たな隠れ隣接関係が構築されていく。そのため、連続した未検出により誤った隠れ隣接関係が生じないかという問題と、隠れ隣接関係が爆発的に増加しないかという問題がある。この点について考察する。

図5.19に、連続した2つ以上のセンサでの未検出が、複数回発生することで仮追跡エージェントが複数生成される例を示す。このとき、本追跡と仮追跡2の間での統合が発生し隠れ隣接関係が構築される場合がある。しかし、このようなことが発生する確率は低いと考えられる。例えば、50%の検出率のセンサが2回連続して人物を見逃す確率は、 $50\% \times 50\% = 25\%$ となる。3回連続して人物を見逃す確率は更に低くなる。また、5.2.5節で述べた方法により、利用されない隠れ隣接関係を削除するため、一時的な見逃しによって生成された隠れ隣接関係は削除され、頻繁に未検出を起こす不確実性の高いセンサがバイパスされるため問題にならないと考える。

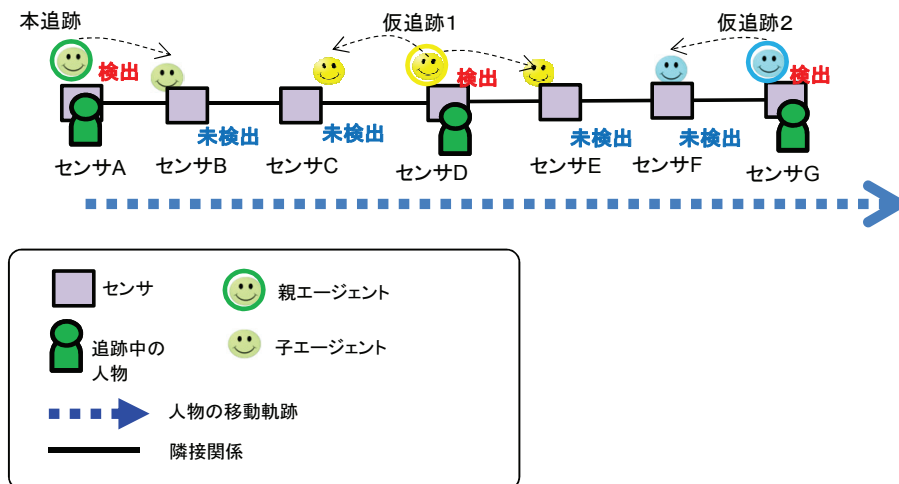


図 5.19: 仮追跡エージェントが複数生成される例

また、計算量の問題に関しては、1回未検出が発生しその次のセンサで仮追跡が開始された場合であれば、1つのノードを介してすぐに統合が発生するため計算量が爆発的に増加することはない。図5.19で示したような連続した未検出が複数回発生する場合にのみ、仮追跡エージェントが繰り返し生成され、指数的に計

算量が増加する。これによる計算量の増加について考えると、未検出率を  $n$ 、本追跡または仮追跡に伴う計算量を  $X$  とすると、2回連続で未検出が発生する確率が  $n^2$  となるため、この確率で計算量が  $X$  だけ増加することとなる。その後、更に2回連続で未検出が発生すると、すなわち、 $n^4$  の確率で更に計算量が  $X$  だけ増加することとなる。このことから1人の人物に対する追跡にかかる計算量は、

$$\sum_{m=0}^{\infty} n^m \cdot X \quad (5.1)$$

となる。これは、 $n < 1$  であるため収束することを意味する。また、実際の運用では一定時間人物を検出しないエージェントを消去することとしているため、この消去により計算量の増加が抑えられている。

## 5.5 誤検出への対応

前節までは、未検出の場合について隠れ隣接関係を抽出することで追跡成功率を改善することについて述べた。センサの種類によっては誤検出について考慮しておく必要がある。誤検出はビーコンのようなIDを正しく認識するデバイスでは発生しないが、カメラで撮影した画像から人物を認識する場合は発生する可能性が十分にある。

誤検出は検出すべきでない人物を検出し、正しく検出すべき人物を検出しなかったとみなすことができる。したがって、正しく検出すべき人物の未検出が起きたとして、隠れ隣接関係の抽出手法を適用することができる。ただし、検出すべきでない人物を検出することによる仮追跡エージェントの生成が行われるため、システム内でエージェントが増加することになる。未検出と同様に誤検出が多発する場合は隠れ隣接関係が増加する。これは、未検出にしても誤検出にしても、追跡エージェントの派遣範囲を広げることで不確実性に対応しているためであり、センサの未検出や誤検出に対応した範囲でシステム全体を検索しなくてもよいように追跡の範囲を確保していると考えることができる。

## 5.6 結言

本節では，センサの不確実性により人物が検出できない場合に追跡を継続できないという問題に対して，センサの検出結果に基づき不確実なセンサをバイパスして隠れ隣接関係を構築することにより，センサの不確実性に対応する手法を提案した．実験の結果，隠れ隣接関係の増加を抑えつつ，人物追跡の成功率が上がることを確認できた．また，本稿ではビーコンやRFIDをセンサとして利用したときに発生する未検出への対応方法を提案したが，カメラ等の機種によって性能や機能の差が大きいセンサを使用して今後実証していくことが課題として挙げられる．

センサには様々な種類のものが存在し，未検出や誤検出に対しての対応も様々である．本節では，人物を追跡するモバイルエージェントの派遣先を決定するアルゴリズムを提案することで，どのようなセンサに対しても不確実性を補う追跡を実現することができる．

## 第6章 結論

### 6.1 まとめ

本稿では、IoT時代に向けた、センサを各場所に設置し自動で人物を追跡する人物追跡システムを提案した。センサを用いた人物追跡システムでは、人物を検出するためのセンサと追跡対象の人物が増えるほど、計算機の処理や通信量への影響が大きくなる。そこで、Raspberry Piのような小型で安価な計算機をセンサとセットにして設置し、分散構成としてモバイルエージェント技術を用いて人物を追跡することで低コストで柔軟な拡張性を持つ人物追跡システムの構築手法を提案した。

本稿で提案している人物を追跡するためのアルゴリズムの特徴は、人物を検出したセンサに対して、次にどのセンサが人物を検出するかをグラフ理論をベースにセンサの位置関係から作成した隣接行列から計算し、次のセンサに追跡対象の人物の特徴データを持たせたモバイルエージェントを派遣し同じ人物の検出を監視することである。このため、人物の特徴データを一か所に集約して処理せずとも、各センサとセットで配置されるノード計算機上で局所的に監視処理を行うことで、管理サーバへの負荷を集中させずに人物の同定処理を可能とする。対象の人物を追跡するモバイルエージェントは、センサの種別に依存せずにその人物の特徴となるデータを保持し隣接するセンサへ移動する設計とすることで、各センサに合わせた人物の同定処理をシンプルな実装で実現し、柔軟に拡張することが可能となる。

しかし、実際の人物追跡システムの運用に際しては、センサの設置環境の変化やセンサ自身の精度により人物を検出する状況が変化し、次の人物の検出位置の予測に影響が出る場合があった。

この問題を解決するために、本稿ではセンサの検出範囲の変更に対応するアルゴリズムを提案した。例えばカメラを用いたシステムではパン・チルト・ズーム

によるセンサの検出範囲の変化やセンサの故障や復帰などにより、次に人物を予測するはずのセンサが変わってしまうという問題があった。そこで、センサの位置関係だけでなく検出範囲と組み合わせ隣接関係を再計算することで、センサの検出範囲の変化に対応できるアルゴリズムを提案した。

次に、システムの柔軟性と可用性を考慮し、各ノードが必要最小限の位置関係と検出範囲の情報を持つように各ノードが持つ情報を局所化することで、検出範囲が変更されたりセンサの追加や削除が発生しても、周辺の影響のあるノードのみで隣接関係の再計算を行うことで各ノードの計算量を削減し、環境の変化に柔軟に対応できる局所化手法を提案した。本提案手法を適用することで、各センサがシステム内の全てのセンサの情報を持たず、隣接する可能性のある局所化されたセンサの情報のみを持つことにより、センサの検出範囲の変更やセンサの追加と削除の影響を局所化し人物の追跡を継続することができる。本提案手法を実装しシミュレーションによる実験を行った結果、センサの検出範囲や故障・復帰などの変化が発生しても、隣接関係が更新される範囲が局所化され、センサ設置環境の変化の影響を限られた範囲に収めつつ追跡が継続できることを確認した。

最後に、センサの精度に起因する、センサの人物検出における不確実性について対応する手法を提案した。センサはその向きや精度により人物を検知しない場合がある。カメラの場合は人物の撮影する向きや画質に大きく影響を受け、100%確実に人物を検出することは困難である。ビーコン等の誤認識のないデバイスを用いても、電波の環境、例えば多くの発信機が存在し混信が発生するような場合や、ビーコンをポケットに入れたり遮蔽物が間に入ることによる電波の減衰等により、センサが人物を検出しないことが頻繁に発生すると考えられる。そこで、これまでの隣接関係の計算手法を発展させ、不確実性の高いセンサをバイパスした隠れ隣接関係を構築することにより、システムの追跡能力を向上する手法を提案した。更に、一時的に検出に失敗したセンサが数多く発生することで未検出したセンサのバイパスが多発し隠れ隣接関係が多く構築されてしまう課題について、隠れ隣接関係の構築数に閾値を設けて抑制する手法を提案した。シミュレーションによる実験の結果、隠れ隣接関係の増加を抑えつつ、人物追跡の成功率を向上できたことを確認した。

このように、センサが分散して配置されている人物追跡システムにおいて、安価で小型な計算機をセンサとセットで設置し、計算機間の通信と人物の追跡処理

と特徴データの伝播をモバイルエージェント技術を用いることで実現し、更に人物の移動先を隣接関係の計算を用いて計算することで、センサの検出環境の変化を考慮し柔軟性と可用性、処理コストの低減を実現する手法を提案した。センサとセットで設置する計算機は、Raspberry Piのようなスマートフォンレベルの処理能力の計算機を想定し、Wi-Fiなどの無線ネットワーク上に接続する。このため、高スペックで特殊なH/Wを必要とせずに簡易に構築し増設できることを考慮している。実際にRaspberry Piを用いて実施した実験では、BluetoothとWi-Fiデバイスを搭載したRaspberry Piを壁または柱のコンセントにUSB電源と併せて設置するだけでよく、配線の引き回しに苦労することなく簡易に設置できた。またRaspberry PiにUSBカメラを接続することで、カメラによる監視も可能になる。

本稿に述べた手法を適用することで、安価で小型な計算機をセンサとセットで運用し、かつセンサの設置数や追跡対象の人物の数による計算機やネットワークのスペックを大きく意識せずに低コストで柔軟な拡張性を持つ人物追跡システムを構築することが可能になる。

## 謝辞

本報告を終えるにあたり，鳥取大学大学院工学研究科 菅原 一孔 教授，川村 尚生 教授，高橋 健一 准教授には終始ご指導頂きました。下名の至らぬところが多々あり，ご迷惑をおかけしましたが，辛抱強くご指導いただきまして心から感謝申し上げます。

人物追跡を行う上で課題となる，複数のカメラ間で撮影された人物の対応付けにつきましましては 西山 正志 准教授にご助言をいただき，心から感謝申し上げます。笹間 俊彦 助教におきましては，実験方法につきましましてご指導いただき，心から感謝申し上げます。

本研究を進めるにあたって，鳥取大学大学院工学部研究科 情報エレクトロニクス専攻 社会情報システム計算機応用研究室では，隣接ノード決定アルゴリズムの開発と実装に協力して下さった平成29年修了の辻 美希様を初め，人物追跡システムの相互運用に関する仕組みに協力いただいた平成27年修了の野村 温様，異なる種類のセンサを用いたシステム間での特徴データの関連付けに協力いただいた平成27年度卒業の椿野 凌様，複数のカメラ間で撮影された人物の対応付けに協力いただいた平成28年度修了の中野 翔太様，人物追跡システムを応用した福祉施設における見守りシステムの開発に協力いただいた平成28年度卒業の尾上 悠介様，人物追跡システムの動作状況の可視化に協力いただいた平成29年卒業の佐々木 優翔様，そして人物追跡システムの実装と福祉施設での実証実験に協力いただいた遠藤 佑貴様には本システムの骨格を作り上げるために多くの時間を頂きました。誠にありがとうございました。

メルコ・パワー・システムズ株式会社におきましては，博士課程への進学を後押ししてくださり様々な面でアドバイスをくださった柿内 博人様を初め，博士課程に進むにあたって様々な面で支えてくださった社長の坂下 龍司様，技術本部長の西尾 弘道様，技術副本部長の糟谷 武則様，技術部長の淵上 政人様，技術企画



## 謝 辞

室長の星野 光勇様，私の所属するチームの皆様我心から感謝申し上げます。さらに，共同研究において様々なアドバイスをいただいた谷川 浩三様，塩塚 大様，お忙しい状況であっても快くご協力頂きました。謹んで心から御礼申し上げます。

そして，本報告を行うにあたって数々の協力を頂きました鳥取大学大学院工学研究科 情報エレクトロニクス専攻 社会情報システム計算機応用研究室の修士および工学部知能情報システム工学科計算機工学講座社会情報システム計算機応用研究室の学部生の皆様心感謝申し上げます。

本研究は多くの方に支えられ，まとめることができました。関係した方々にはいくら感謝を申し上げても足りるものではございません。今後の皆様の御健勝と御発展を心よりお祈り申し上げ，最後に感謝の言葉とさせていただきます。

平成 29 年 7 月 四元 辰平

## 参考文献

- [1] Qrio.inc. Smart Lock, <https://qrio.me/smartlock/>
- [2] ミマモルメ 阪神電車の安心サービス, <http://www.hanshin-anshin.jp/>
- [3] 伊藤昌毅, 川村尚生, and 菅原一孔. ”スマートフォンを利用したバスロケーションシステムの開発.” 電子情報通信学会論文誌 D 96.10 (2013): 2327-2339.
- [4] バスネット, <http://www.ikisaki.jp/home/about>
- [5] 大澤達哉, 若林佳織, ”映像モニタリングのための人物追跡技術,” NTT 技術ジャーナル, 2007, 17.
- [6] NTT docomo, イマドコサーチ, <https://www.nttdocomo.co.jp/service/imadoco/>
- [7] au, 安心ナビ, <https://www.au.com/mobile/service/smartphone/safety/anshin-navi/>
- [8] SECOM, ココセコム, <http://www.855756.com/index.html>
- [9] 今野将, et al., ”行動履歴を用いた児童見守り支援システムのための目的地推定手法の試作,” マルチメディア, 分散協調とモバイルシンポジウム 2011 論文集, 2011, 2011, 697-705.
- [10] 大野雄基, et al., ”弱者を遠隔地から見守るシステム TLIFES の提案と実装,” 研究報告グループウェアとネットワークサービス (GN), 2012, 2012.2, 1-8.
- [11] Panasonic, 在席・所在管理システム, <http://www.panasonic.com/jp/business/its/location.html>
- [12] 井上豊, 池田剛, 山本潔, 幸島明男, 山下倫央, 麻生英樹, 車谷浩一, ”ComPass ビーコンを用いたスマートフォン上での屋内自律型測位システム,” 情報処理学会研究報告ユビキタスコンピューティングシステム (UBI), 2008(18 (2008-UBI-017)), 195-202.

- [13] 太田義和, 菅野正嗣, 村田正幸, ” センサネットワークにおける位置測定のためのデータ収集方式の提案, ” 信学技報, IN2004-104, 2004, 437, 31-36.
- [14] Teach, Learn, and Make with Raspberry Pi. <http://www.raspberrypi.org/>.
- [15] 岩井俊弥. Java モバイル・エージェント. ソフトリサーチセンター, January 1999.
- [16] 本位田真一, 飯島正, 大須賀昭彦. オブジェクト指向トラックエージェント技術. ソフトウェアテクノロジーシリーズ3. 共立出版, 1999.
- [17] 本位田真一, 大須賀昭彦. オブジェクト指向からエージェント指向へ. ソフトバンククリエイティブ, 2001.
- [18] 浜田康志, 坂本静生, ” 顔認証製品と社会ソリューションでの活用 (社会の安全・安心を支えるパブリックソリューション特集)-(効率・公平な暮らし), ” NEC 技報= NEC technical journal 67.1 (2014), 52-55.
- [19] 浜田康志, 坂本静生, ” 異業種での映像情報メディア利用 (第 16 回) 顔認証ソフトウェア「NeoFace」とその活用事例, ” 映像情報メディア学会誌 The journal of the Institute of Image Information and Television Engineers, 2017, 71.1, 85-88.
- [20] NICT, プレスリリース, <https://www.nict.go.jp/press/2013/11/25-1.html>
- [21] NICT, 映像センサー使用大規模実証実験検討委員会調査報告書, <https://www.nict.go.jp/nrh/iinkai/report.pdf>
- [22] C. Nakajima, M. Pontil, B. Heisele, and T. Poggio, “ Full-body person recognition system, ” Proc. Pattern Recognition, vol.36, no.9, pp.1997-2006, 2003.
- [23] X. Wang, G. Doretto, T. Sebastian, J. Rittscher, and P. Tu, “ Shape and appearance context modeling, ” Proc. International Conference on Computer Vision, pp.1-8, 2007.
- [24] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani, “ Person re-identification by symmetrydriven accumulation of local features, ” Proc. Computer Vision and Pattern Recognition, pp.2360-2367, 2010.
- [25] D. Gray and H. Tao, “ Viewpoint invariant pedestrian recognition with an ensemble of localized features, ” Proc. European Conference on Computer Vision, pp.262-275, 2008.

- [26] D.G.Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, Vol.60, No.2, pp.91-110, 2004.
- [27] Y.Cui, N.Hasler, T.Thormaehlen, and H.P. Seidel, "Scale invariant feature transform with irregular orientation histogram binning," *Proceedings of the International Conference on Image Analysis and Recognition*. Springer, 2009.
- [28] F.Porikli, "Human body tracking by adaptive background models and mean-shift analysis," *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2003.
- [29] Hiroto Kakiuchi, Kozo Tanigawa, Takao Kawamura, Kazunori Sugahara, A Construction Method for Automatic Human Tracking System with Mobile Agent Technology, *Recent Developments in Video Surveillance*, pp. 21-38 (2012).
- [30] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person-re-identification," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp.152-159,2014.
- [31] J. Roth and X. Liu, "On the exploration of joint attribute learning for person re-identification," *Proc. 12th Asian Conference on Computer Vision*, pp.673-688, 2014.
- [32] L. Bourdev, S. Maji, and J. Malik, "Describing people: A poselet-based approach to attribute classification," *Proc. International Conference on Computer Vision*, pp.1543-1550, 2011.
- [33] R. Layne, T. Hospedales, and S. Gong, "Person reidentification by attributes," *Proc. British Machine Vision Conference*, pp.93-117, 2012.
- [34] R. Layne, T. Hospedales, and S. Gong, "Person re-identification," chapter *Attributes-Based Reidentification*, pp.93-117, 2014.
- [35] K. Sameh, K. Cheng-Hao, K.S. Vivek, Vinay D.S., and Larry S.D., "Joint learning for attributeconsistent person re-identification," *Proc. ECCV Workshop on Visual Surveillance and Re-Identification*, pp.134-146, 2014.
- [36] Z. Shi, T.M. Hospedales, and T. Xiang, "Transferring a semantic representation for person re-identification and search," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp.4184-4193, 2015.

- [37] 警察庁, 生活安全の確保, <https://www.npa.go.jp/safetylife/seianki/fumei/index.html>
- [38] 警視庁, 公開捜査ファイル, <http://www.keishicho.metro.tokyo.jp/jiken/jiken.htm>
- [39] Police Scotland, Missing persons, <http://www.scotland.police.uk/whats-happening/missingpersons/james-barr>
- [40] Police Scotland, Most wanted, <http://www.scotland.police.uk/whats-happening/most-wanted/>
- [41] 中野 翔太, 四元 辰平, 吉村 宏紀, 西山 正志, 岩井 儀雄, 菅原 一孔. ”身体と外見の共起属性を用いた人物対応付け,” 電子情報通信学会論文誌, Vol.J100-D, No.1, pp.104-114, Jan. 2017.
- [42] 千代原亮一, ”性犯罪者情報公開法の合憲性: ミーガン法からジェシカ法へ,” 研究紀要 5.1 (2007), 109-127.
- [43] MAMORIO, <https://mamorio.jp/>
- [44] Beutel, J. and Krasnyanskiy, M.: Linux BlueZ Howto: Bluetooth Protocol Stack for Linux, <http://jeremythompson.uklinux.net/RH8-0/bluezhowto.pdf> (2001)
- [45] Fette, I. and Melnikov, A.: IETF RFC 6455: The Web-Socket Protocol, IETF, Network Working Group (2011)
- [46] Kim, C. and Lee, S.: A research on Beacon code architecture extension using category and code Beacon structure, in Information and Communication Technology Convergence (ICTC), 2014 International Conference on, pp. 187-188, IEEE (2014)
- [47] Kohne, M. and Sieck, J.: Location-based services with iBeacon technology, in Artificial Intelligence, Modelling and Simulation (AIMS), 2014 2nd International Conference on, pp. 315-321, IEEE (2014)
- [48] Layne, R., Hospedales, T. M., Gong, S., and Mary, Q.: Person Re-identification by Attributes., in BMVC, Vol. 2, p. 8 (2012)
- [49] 森大樹, 内海章, 大谷淳, 谷内田正彦, ”非同期多視点画像による人物追跡システムの構築,” 電子情報通信学会論文誌, Vol.J84-D-II, pp.102-110, Jan. 2001.

- [50] 中澤篤志, 日浦慎作, 加藤博一, 井口征士, ”分散視覚エージェントを用いた複数人物追跡システム,” 情報処理学会論文誌, Vol.42, No.11, pp. 2669-2710, Nov. 2001.
- [51] 浮田宗伯, ”能動視覚エージェント群の密な情報交換による多数対象の実時間協調追跡,” 電子情報通信学会論文誌, Vol.J88-D-I, No.9, pp.1438-1447, Sep. 2005.
- [52] 谷澤佳道, 佐藤一郎, 安西祐一郎, ”ユーザ追従型モバイルエージェントフレームワーク“FollowingSpace”の提案と応用,” 情報処理学会論文誌, Vol.43, No.12, pp.3775-3784, Dec. 2002.
- [53] 田中孝浩, 小川剛史, 沼田哲史, 板生知子, 塚本昌彦, 西尾章治郎, ”カメラセンサネットワークにおけるモバイルエージェントを用いた適応的手法選択型人物追跡システムの設計と実装,” 情報処理学会グループウェアとネットワークサービスワークショップ2004 論文集, pp.15-20, Nov. 2004.
- [54] 武村紀子, 中村泰, 松本吉央, 石黒浩, ”可変長予測に基づく人物追跡”平成21年度情報処理学会関西支部支部大会講演論文集, 2009, (2009-09-29).
- [55] JR 東日本, 各駅の乗車人員, <http://www.jreast.co.jp/passenger/>
- [56] OpenLayers, <https://openlayers.org/>
- [57] Google Inc., Google Maps, <https://www.google.co.jp/maps/>
- [58] IETF, RFC6455 The WebSocket Protocol, <https://tools.ietf.org/html/rfc6455>
- [59] D. B. Lange and M. Oshima. ”Seven good reasons for mobile agents,” Communications of the ACM, Vol. 42, No. 3, pp. 88.89, 1999.
- [60] OSGi Alliance. OSGi Alliance Specifications OSGi Service Platform Release1. <http://www.osgi.org/Specifications/HomePage>, 2008.
- [61] iBeacon for Developers - Apple Developer.
- [62] SIG Bluetooth. Core specification v4.0, 2010.
- [63] 田澤新成, 白倉暉弘, 田村三郎. やさしいグラフ論. 現代数学社, 2003.
- [64] 宮原洋八, 竹下寿郎, 西三津代. 地域住民(17歳~92歳)を対象とした運動能力. 理学療法科学, Vol. 19, No. 4, pp. 285290, 2004.

- [65] 遠藤 佑貴, 四元 辰平, 高橋 健一, 川村 尚生, 菅原 一孔, ” 人物追跡システムの福祉施設への応用, ” 合同エージェントワークショップ&シンポジウム 2016 (JAWS-2016) , pp. 170-174 (2016)
- [66] 村松大吾, 岩間晴之, 木村卓弘, 榎原靖, 八木康史. (2014). 一歩行映像から取得される複数特徴を用いた個人認証. 電子情報通信学会論文誌 A, 97(12), 735-748.
- [67] 古館達也, 堀川三好, 工藤大希, 岡本東. (2015). M-009 Bluetooth Low Energy ビーコンを用いた屋内測位手法に関する研究 (M 分野: ユビキタス・モバイルコンピューティング, 一般論文). 情報科学技術フォーラム講演論文集, 14(4), 311-312.

## 公表論文

### 学術誌論文

1. 四元 辰平, 谷川 浩三, 辻 美希, 高橋 健一, 川村 尚生, 菅原 一孔, 人物追跡システムにおける局所化された隣接関係の計算, 電子情報通信学会論文誌, Vol. J99-D, No. 5, pp. 468-477 (2016).
2. Tappei Yotsumoto, Kozo Tanigawa, Miki Tsuji, Kenichi Takahashi, Takao Kawamura, Kazunori Sugahara, Automatic Human Tracking System using Localized Neighbor Node Calculation, *Sensors & Transducers*, Vol. 194, No. 11, pp. 54-61 (2015)
3. 谷川 浩三, 四元 辰平, 柿内 博人, 野村 温, 高橋 健一, 川村 尚生, 菅原 一孔, 人物追跡システムにおけるカメラの撮影範囲を考慮した隣接関係の計算, 電子情報通信学会論文誌, Vol. J97-B, No. 10, pp. 914-918 (2014).



## 国際会議論文

1. Tappei Yotsumoto, Kozo Tanigawa, Miki Tsuji, Kenichi Takahashi, Takao Kawamura, Kazunori Sugahara, Automatic Human Tracking System using Localization of Neighbor Node Calculation, *The Ninth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2015)*, pp. 139-145 (2015). Venice/Mestre, Italy
2. Tappei Yotsumoto, Masaru Shiozuka, Kenichi Takahashi, Takao Kawamura, Kazunori Sugahara, Hidden neighbor relations to tackle the uncertainty of sensors for an automatic human tracking, *2017 Second IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT 2017)*, pp. 000-000 (2017). Coimbatore, India