

# 室内における赤外線センサを用いた複数人の移動軌跡推定法の提案

笠間 勇輝<sup>†</sup> 宮崎 敏明<sup>†</sup>

会津大学大学院コンピュータ理工学研究科<sup>†</sup>

## 1. はじめに

ユビキタス環境実現のために、人の位置推定は非常に重要である。近年、赤外線センサ (IR センサ) を室内の天井に設置し、得られる IR センサの反応パタンから、人の移動軌跡を推定する手法がいくつか提案されている [1][2]。それら手法は、カメラを使用した方法に比べプライバシー保護の観点から有意義である。しかし、従来手法は、推定対象が 1 人の場合が主であった。また、我々が以前提案した手法 [3] では、複数人の移動軌跡を推定できるものの、センシングデータを全て読み込んだ後に推定処理を開始するため、実時間推定が困難であった。本稿では、室内の人数等の前情報を必要とせず、IR センサの反応パタンのみから複数人の移動軌跡を動的かつ短時間で推定する手法を提案する。

## 2. 提案手法

本稿では、室内の天井に検知半径  $r$  を持つ IR センサが複数個ランダムに配置され、それら IR センサで室内の全領域が十分検知できるものと仮定する。ここで、各 IR センサは、人を検知したとき “1” を出力し、そうでない場合は “0” を出力する。また、各 IR センサの位置座標は既知とする。上記環境下で、IR センサから送られてくるセンシングデータ (1 or 0) のみを用いて各人の移動軌跡を推定することを考える。提案手法は、位置推定と軌跡推定の 2 つの処理から成り、それらを交互に繰り返すことにより、複数人の移動軌跡を推定する。まず、提案手法の説明に使用する変数類を図 1 のように定義する。

- $t$ : 時刻。  $t > 0$
- $S$ : 設置したセンサの数
- $\mathbf{RD}(t)$ : 時刻  $t$  におけるセンシングデータ集合 (1 or 0)。  $\mathbf{RD}(t) = \{b_1, \dots, b_s, \dots, b_s\} | b_s \in \{0, 1\}$ 。
- $\mathbf{RDL}$ :  $\mathbf{RD}(t)$  のリスト。
- $WS1$ :  $\mathbf{RDL}$  の大きさ。  $WS1 > 0$
- $\mathbf{ORD}(t)$ :  $\mathbf{RDL}$  に対して論理 OR 演算を計算した集合 (1 or 0)。  $\mathbf{ORD}(t) = \{b_1, \dots, b_s, \dots, b_s\} | b_s \in \{0, 1\}$
- $\mathbf{WL}(t)$ :  $\mathbf{RDL}$  に対して反応回数をカウントした集合。  $\mathbf{WL}(t) = \{I_1, \dots, I_s, \dots, I_s\} | I_s$  は 0 以上の整数。
- $C_{\max}(t)$ : 時刻  $t$  におけるクラスタ数。
- $\mathbf{CC}(t)$ : 時刻  $t$  におけるクラスタの重み付き中心座標  $cc_i(t)$  の集合。  
 $\mathbf{CC}(t) = \{cc_i(t) | i = 1, \dots, C_{\max}(t)\}$
- $\mathbf{CCL}$ :  $\mathbf{CC}(t)$  のリスト。
- $WS2$ :  $\mathbf{CCL}$  の大きさ。  $WS2 > 0$
- $TTL_{\max}$ : 各人の最大 Time To Live (TTL)。  $TTL_{\max} > 0$
- $\mathbf{dv}(n)$ : 時刻  $t$  における直近  $WS2$  個の位置座標  $dc_i(n)$  の集合。  $\mathbf{dv}(n) = \{dc_1(n) | i = 1, \dots, WS2\}$ 。方向ベクトルは “ $dc_{ws2}(n) \rightarrow dc_1(n)$ ” と定義される。
- $\mathbf{DVL}$ :  $\mathbf{dv}(n)$  のリスト。
- $td$ : ターゲット情報の構造体。メンバとして、 $td.ID$ 、 $td.Path$ 、 $td.TTL$  を持つ。
- $td.ID$ : ターゲット ID。
- $td.Path$ : ターゲットの経路座標リスト。
- $td.TTL$ : ターゲットの TTL。
- $\mathbf{TDL}$ :  $td$  のリスト。
- $\mathbf{mv}(td.ID)$ : ターゲット  $td.ID$  の動きベクトル。  $\mathbf{mv}(td.ID) = \{mc_i(td.ID) | i = 1, \dots, WS2\}$ 。動きベクトルは “ $mc_{ws2}(td.ID) \rightarrow mc_1(td.ID)$ ” である。
- $|\cdot|$ : リストの要素数 (大きさ)。

図 1. 提案手法で用いる変数類の定義

### 2.1 位置推定

位置推定では、IR センサの反応から時刻  $t$  における各人の位置を推定する。詳細を図 2 に示す。初めに  $WS1$  個のセンシングデータを保持する  $\mathbf{RDL}$  に対して論理 OR 演算を行い、 $\mathbf{ORD}(t)$  と  $\mathbf{WL}(t)$  を得る。ここで、 $\mathbf{ORD}(t)$  とは

論理 OR 演算の結果 (1 or 0) であり、 $\mathbf{WL}(t)$  は各センサの  $WS1$  個のセンシングデータの内にいくつ 1、すなわち何回反応したかを表す。次に  $\mathbf{ORD}(t)$  の値が 1 であるセンサに対して、Ward 法 [4] に基づいたクラスタリングを行う。この際、IR センサの検知範囲を閾値としてクラスタリングを終了する。求めた各クラスタにおいて、 $\mathbf{WL}(t)$  を重みとして各センサの重心座標を求め、それらの集合である  $\mathbf{CC}(t)$  を得る。本稿では、求めた  $\mathbf{CC}(t)$  の各位置が時刻  $t$  における各人の存在位置であるとする。この  $\mathbf{CC}(t)$  を  $\mathbf{CCL}$  に追加し、その個数が  $WS2$  であれば軌跡推定に移る。

### 2.2 軌跡推定

図 3 と図 4 に軌跡推定の手順を示す。初めに、各人の動きを得るため、 $\text{CalcDV}()$  を呼び出し、 $\mathbf{CCL}$  に対して方向ベクトルを求める。その方法は、下記の通りである。まず、 $\mathbf{CCL}$  の最後の要素である  $\mathbf{CC}(t)$  の各要素 (位置座標) を最初の要素とする  $C_{\max}(t)$  個の集合  $\mathbf{dv}(n) | n=1, \dots, C_{\max}(t)$  を作成する。次に  $\mathbf{CCL}$  内の要素  $\mathbf{CC}(t-1)$  から  $\mathbf{CC}(t-WS2+1)$  に対して、距離および角度が最小となる候補位置を順に 1 つずつ選択し、それを各  $\mathbf{dv}(n)$  に append していく。ここで、距離とは、 $\mathbf{dv}(n)$  の最後の要素から候補位置への距離である。角度とは、 $\mathbf{dv}(n)$  の最初の要素から最後の要素へのベクトルと、 $\mathbf{dv}(n)$  の最後の要素から候補位置へのベクトルが成す角である。また、距離の上限は IR センサの検知範囲の直径  $2r$  以内、角度の上限は  $90^\circ$  以内として候補位置選択に制限を付ける。これら制限は、「人は目的地に対して最短経路で、直線的に移動する」という仮定から来ている。最終的に得られる方向ベクトルは、時刻  $t-WS2+1$  から  $t$  へのベクトル “ $dc_{ws2}(n) \rightarrow dc_1(n)$ ” であり、 $\mathbf{DVL}$  に追加・保持される。

次に、得られた  $\mathbf{DVL}$  から各人の次位置を推定する。現在追跡している人 (ターゲット) の情報  $td$  を要素とするリスト  $\mathbf{TDL}$  が空の場合、 $\text{NewTarget}()$  を呼び出し、 $\mathbf{DVL}$  内の各  $\mathbf{dv}(n)$  の最後の要素 “ $dc_{ws2}(n)$ ” を開始位置とした新たな  $td$  を作成する。これは、新たに人が入出したことに対応する。 $\mathbf{TDL}$  が空でない場合、 $\text{SelectDV}()$  より各人の移動軌跡を推定する。 $\mathbf{TDL}$  と  $\mathbf{DVL}$  の中から、相互の距離と角度を最小するターゲット  $td$  と方向ベクトル  $\mathbf{dv}(n)$  の

```

procedure Main() {
  RDL = {}; CCL = {}; TDL = {}; t = 0;
  h = 0; // ターゲットID

  repeat {
    t++; // 時刻更新

    // 区間 [t, t-WS2+1] におけるデータの準備
    RD(t) = 時刻 t におけるデータの取得;
    RDL ← RD(t) を追加;
    if (|RDL| < WS1) continue;
    if (|RDL| > WS1) RDL の最初の要素を削除;

    // 時刻 t における各人の位置推定
    ORD(t) = RDL に対して OR 演算;
    WL(t) = RDL に対して反応回数のカウント;
    ORD(t) に対してクラスタリングを行い、
    WL(t) を用いて重み付き重心座標の集合 CC(t) を計算;

    // 軌跡推定への準備
    CCL ← CC(t) を追加;
    if (|CCL| < WS2) continue;
    if (|CCL| > WS2) CCL の最初の要素を削除;

    // 軌跡推定
    PathEstimation();
  }
}

```

図 2. 提案手法の主要手順

An Approach to Movement Loci Estimation for Multiple Humans in a Room Using Binary Infrared Sensors  
<sup>†</sup> Yuki Kasama, <sup>†</sup> Toshiaki Miyazaki  
<sup>†</sup> Graduate School of Computer Science and Engineering, The University of Aizu

組み合わせを選択する。ここで距離とは、 $mv(td.ID)$ の最初の要素 $mc_1(td.ID)$ と $dv(n)$ の最後の要素 $dc_{ws2}(n)$ との距離である。角度とは、 $td$ の動きベクトル $mv(td.ID)$ と方向ベクトル $dv(n)$ とのなす角である。この $mv(td.ID)$ は $td.Path$ における過去 $WS2$ 個の位置座標より求められる。最適な組み合わせが見つかった場合、そのターゲットの経路座標リスト $td.Path$ に $dv(n)$ の最後の要素 $dc_{ws2}(n)$ を追加し、 $td.TTL$ に1を加える。選択された $td$ と $dv(n)$ は**TDL**と**DVL**からそれぞれ削除される。上述した処理を、最適な組み合わせがなくなるまで繰り返す。上記処理で、最適な方向ベクトルが見出せなかった $td$ があった場合、

$td.TTL$ を1減らし、 $td.TTL < 0$ となった場合、当該 $td$ は**TDL**から削除される。これは、対応する人が退出したことに相当する。さらに、選択されなかった方向ベクトルが**DVL**に存在する場合、新たに人が入室したと仮定して、**NewTarget()**により新たに $td$ を作成する。上述したように、提案手法では、動的に人の入退室にも対応できる。また、ある時刻の人の位置推定は、過去 $WS1$ 個のセンシングデータを使用するが、 $WS1$ は小さくて良いため、ほぼ実時間で各人の移動軌跡を推定できる。

### 3. 評価

本手法を評価するために実験データを人工的に作成し、それを用いて評価を行った。実験データの内容は以下の通りである。10 m×10 mの部屋の天井に、半径 $r=2.0$  mの検知範囲を持つIRセンサを5×5の等間隔に設置する。ここで、各IRセンサは、毎秒6個のセンシングデータを出力する。上記環境において下記3シナリオに従って、実験データをそれぞれ3パターン、計9パターンを作成し、それを用いて提案手法を評価した。

1. 1人が入室、その後退出
  2. 2人が同時に入室、その後各々が退出
  3. 2人が時間差に入室、その後時間差で各々が退出
- また、各パラメータは、予備実験を行い、結果が良好であった値を用いた。具体的には、 $WS1=3$ 、 $WS2=3$ 、 $TTL_{MAX}=6$ を使用した。実験結果を表1に示す。各値は、同一シナリオにおいて実験を実施した3パターンの平均値である。ここで、平均誤差とは真の移動経路と推定経路との誤差である。また、標準偏差とは平均誤差の標準偏差である。人数推定成功率とは、各時刻において室内における真の人数を正しく推定できた割合である。表1から、すべてのシナリオにおいて平均誤差0.6 m以下の精度で移動軌跡が推定できている。しかも、室内人数の推定率も、86%以上である。このことから、提案手法は人数が増減する複雑な場面でも、適応可能であるという知見を得た。

表 1. 実験結果

シナリオ	平均誤差 (m)	標準偏差 (m)	人数推定成功率 (%)
1	0.48	0.29	98.5
2	0.57	0.43	92.8
3	0.56	0.44	86.1

### 4. 結論

本稿では、室内に設置した複数のIRセンサの反応パターンから、複数人の移動軌跡を推定する手法を提案した。計算機実験より、平均誤差0.6 m以下の良好な結果を得た。今後は実環境を用いた提案手法の検証を行っていく。

**謝辞** 本研究は、日本学術振興会学術研究助成基金助成金（課題番号：23500095）および総務省戦略的情報通信研究開発推進制度（SCOPE No.121802001）の支援を受けて実施した。

### 文献

- [1] S. Honda, K. Fukui, K. Moriyama, S. Kurihara, and M. Numao, "Extracting Human Behaviors with Infrared Sensor Network," Proc. Fourth International Conference on Networked Sensing Systems (INSS '07), pp.122-125, June 2007.
- [2] Z. Wang, E. Bulut and E.K. Szymanski, "Distributed Target Tracking with Directional Binary Sensor Networks," Proc. IEEE Global Telecommunications Conference 2009 (GLOBECOM 2009), pp.1-6, Nov. 30 2009-Dec. 4 2009.
- [3] T. Miyazaki, and Y. Kasama, "Estimation of the Number of Humans and their Movement Paths in a Room using Binary Infrared Sensors," Proc. ACM International Conference on Ubiquitous Information Management and Communication (ICUIMC2012), Article No. 91, 2012.
- [4] J. H. Ward, Jr, "Hierarchical Grouping to Optimize an Objective Function," Journal of the American Statistical Association, Vol. 58, pp. 236-244, 1963.

```

procedure PathEstimation()
  DVL = CalcDV(); // 時刻tにおける方向ベクトルの計算

  if (TDL == {})
    NewTarget(); // ターゲットの初期設定
  else
    // 各人に対して最適な方向ベクトルを選択し、次位置を推定
    SelectDV();

    if (DVL ≠ {})
      NewTarget(); // 新たに人が入室したと仮定
  }

procedure NewTarget()
  for (n = 1; n <= |DVL|; n++) {
    td = 新規作成;
    td.ID = h++; // 新ターゲットID
    // 方向ベクトルの開始位置がターゲットの開始位置となる
    td.Path = {};
    td.Path ← dv(n)の最後の要素(位置座標)を追加;
    td.TTL = 1;
    TDL ← tdを追加;
  }
}
    
```

図 3. 軌跡推定の主手順

```

procedure SelectDV()
  TDL_bk = {}; // TDLの一時バックアップ用
  repeat
    distance = r×2; // IRセンサの検知範囲の直径
    angle = 90;
    ttl = 0;
    i_best = 0; n_best = 0;

    for (i = 1; i <= |TDL|; i++) {
      td = TDL[i];
      for (n = 1; n <= |DVL|; n++) {
        distance_temp = 座標 "mc1(td.ID)" と "dcws2(n)" 間の距離;
        angle_temp = ベクトル "mcws2(td.ID) → mc1(td.ID)" と
          "dcws2(n) → dc1(n)" のなす角;

        if (distance < distance_temp) continue;
        if (angle < angle_temp) continue;
        if (ttl > td.TTL) continue;

        // ターゲットtdと方向ベクトルdvの最適な組み合わせの一時保存
        i_best = i;
        n_best = n;
        distance = distance_temp;
        angle = angle_temp;
        ttl = td.TTL;
      }
    }

    if (i_best > 0 && n_best > 0) { // 最適な組み合わせがある場合
      td = TDL[i_best]のターゲットデータ;
      dv(n_best) = DVL[n_best]の方向ベクトル;
      td.TTL++;
      if (td.TTL > TTL_MAX) td.TTL = TTL_MAX;
      td.Path ← dv(n_best)の最後の要素(位置座標)を追加;
      TDL_bk ← tdを追加;
      TDL から tdを削除;
      DVL から dv(n_best)を削除;
    }
    else break; // 最適な組み合わせがない場合は終了
  } // repeatの最後

  if (TDL ≠ {}) { // 方向ベクトルを選べなかったターゲットがある場合
    foreach td in TDL {
      td.TTL--;
      if (td.TTL < 0) continue; // ターゲットの削除
      TDL_bk ← tdを追加;
    }
  }
  TDL = TDL_bk;
}
    
```

図 4. 各人の次の移動位置を求める手順