

複数のアクティブカメラを用いた指定人物のリアルタイム追跡

楊長輝 白井良明 三浦純 先山卓朗

大阪大学大学院 工学研究科

Real-Time Specified Person Tracking Using Multiple Active Cameras

ChangHui Yang Yosiaki Shirai Jun Miura Takurou Sakiyama

School of Industry Science, Graduate School of Osaka University

あらまし 複数のアクティブカメラを用い、複雑な環境で指定人物をリアルタイム追跡する手法を提案する。まず、入力画像中のオプティカルフローを抽出する。フローに基づき、移動人物を検出して追跡を始める。複数のアクティブカメラを用いることにより、各カメラ同士の人物実世界位置を交換して、ロバストに指定人物を追跡する手法を提案する。また、PC クラスタを利用することで、実時間人物追跡を実現した。

1 はじめに

視覚を用いた物体追跡は、自律移動やヒューマンインタフェース、侵入者監視、運動解析など様々な応用に必要な技術である。物体追跡を幅広く応用するために、一般的なシーンでの安定した追跡手法が求められている。

これまで移動人物の抽出および追跡に関する様々な手法が提案されてきた。代表的な手法としては、背景画像を与えておく方法[2]、フレーム間差分をとる方法[3]、相関をとる方法[4]、オプティカルフローを用いる方法[5]がある。

背景画像を用いる手法は、異なる時間で背景が変化するので適用できない。フレーム間差分は、追跡のために視線を動かすときには、背景画像も変化するので適用できない。背景画像が変化する場合にも、連続する入力画像間で相関の高い領域の対応をとれば人物追跡を行えるが、対象人物の一部が障害物に隠蔽された場合、人物追跡は困難となる。

ここでは、移動人物の動きに合わせてカメラの視線向きを動かし、移動人物が隠蔽された場合に人物領域を予測し、移動人物を追跡することを考え、複数のアクティブカメラを利用して、オプティカルフローに基づき追跡することにより、上記

の問題を解決した。また、追跡には画像処理の計算量が大きいため、PC クラスタを利用した並列処理により、画像処理を高速化し、リアルタイム指定人物追跡を実現した。

2 人物追跡の基本的な流れ

本稿では、画像中の物体は滑らかに動き、人物領域では動きは一樣であるとする。時系列画像からブロック SAD マッチングに基づいて、再帰的な手法[1]で高速に求めたオプティカルフローを、セグメンテーションによって指定人物と指定人物の近くにいる人物とに分類して、指定人物を追跡する。まず、指定人物の近くに他の移動人物がない場合について述べる。

単一人物を追跡する場合に、移動人物が最初に出現するフレームでは、0 でないフローの検出された領域を初期領域として追跡ウィンドウをセットし、その後は以下の 1. ~5. の手順を繰り返して、追跡を行う。

1. 現フレームの追跡ウィンドウから前フレームの平均フローだけ平行移動させたとこに予測ウィンドウをセットする。ただし、1 フレーム目は前フレームの平均フローがないので初期領域を予測ウィンドウとする(図 1)。

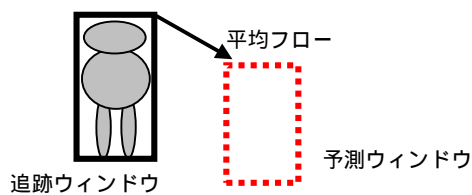


図 1：人物の検出と追跡ウィンドウの設定

2. 予測ウィンドウ内で平均フローを計算する
3. 予測ウィンドウ内とその周囲で平均フローに類似したフローを探す(図 2)。

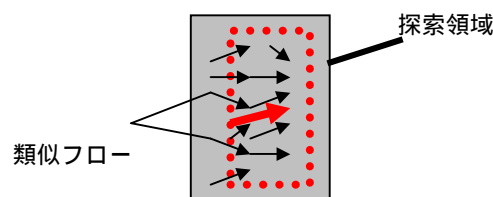


図 2：平均フローと類似したフローの探索

4. 類似したフローの領域を追跡ウィンドウとして更新する(図 3)。

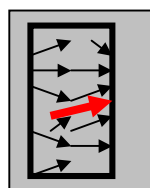


図 3：追跡ウィンドウの更新

5. ステップ 1. に戻る。

3 指定人物の追跡

本研究では,最初に画面内に現れた人物を指定人物として追跡する。

3.1 人物同士が離れているときの追跡

複数の移動人物が存在しても,カメラの回転,障害物などによる隠蔽が発生していないとき,2章で述べた追跡の手順に従って,フローのある領域を個別に追跡する(図 4)。



図 4(a)：一人の場合



図 4(b)：複数人物の場合

3.2 指定人物がカメラの境界に来るときの追跡

通常はカメラを固定しておき,指定人物がカメラ視野の境界付近に来た場合のみカメラを動かして追跡を続ける(図 5)。

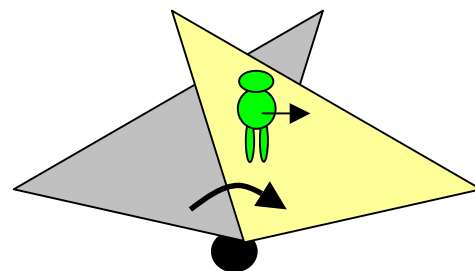


図 5：カメラ視野範囲の更新

カメラが回転し終わるまでに,約 1 秒待たなければならない.カメラ回転中はオプティカルフローの計算はしないため,単一のカメラでは,回転後の人物領域を予測できなくなり,指定人物を見失ってしまうことがある(図 6)。



フレーム 45

フレーム 46

図 6：単一カメラの場合

複数のカメラを用いて追跡を行うことで,この問題を解決する.カメラの向きを変えた後,正常に指定人物を追跡している他のカメラから指定人物の実世界位置情報をもらうことによって,画像中の人物領域を推定することができる.カメラ回転処理の流れは,図 7 のようになる。

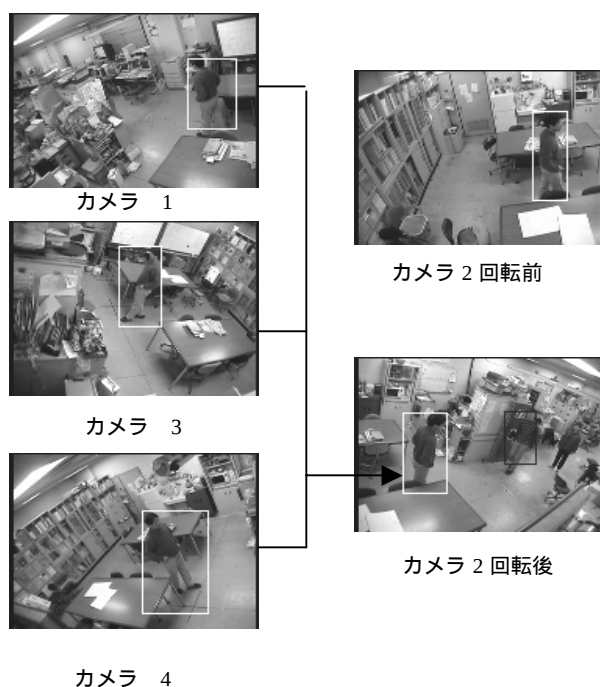


図 7： カメラ回転の処理結果

図7はカメラ2の向きが変わる時の処理である。カメラ2の回転後、他のカメラからの追跡情報を用いて指定人物の実世界位置を計算し、そこから画面上の人物領域を予測して、指定人物を引き続き追跡する。このような処理により、カメラの視野範囲を変えても、口バストに目標指定人物の追跡を続けることができる。カメラの視野範囲を変えることによって、より広範囲にわたる追跡が可能となる。

3.3 静的障害物に隠蔽されているときの対応

追跡人物が高い戸棚などの後ろを通ると、静的障害物に隠蔽されてしまい、画面上では移動人物の特徴とするオプティカルフローを抽出することができない。そのカメラ自身の情報だけでは、追跡したい人物を見失うことがある。

複数のカメラを利用する場合、あるカメラで追跡指定人物が障害物に隠蔽されたと判断すれば、指定人物を安定に追跡している他のカメラから指定人物の実世界位置情報をもって、それを用いて、指定人物の画面上の領域を予測する。この処理を隠蔽された指定人物が再び現すまで繰り返し続ける。

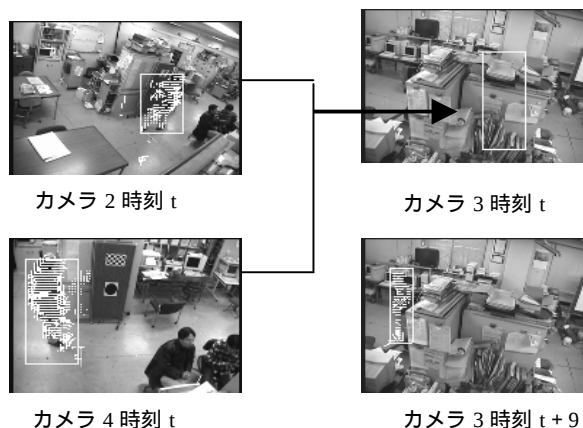


図 8： 隠蔽時の追跡結果

図8では、カメラ3で指定人物が隠蔽されているが、カメラ間の情報交換により正しい人物領域を予測することができている。

3.4 人物同士が重なっているときの追跡

複数人物の追跡では単一の場合と違い、二人の人物が画面上で重なり、片方の人物が他の人物を隠してしまう場合がある。二人の人物が重なる場合、後ろの人物のオプティカルフローが見えなくなるため、フローの数によって重なっている人物の前後関係を判定できる。後ろの人物領域は他のカメラからの情報によって追跡を続ける。追跡の流れは図9のようになる。

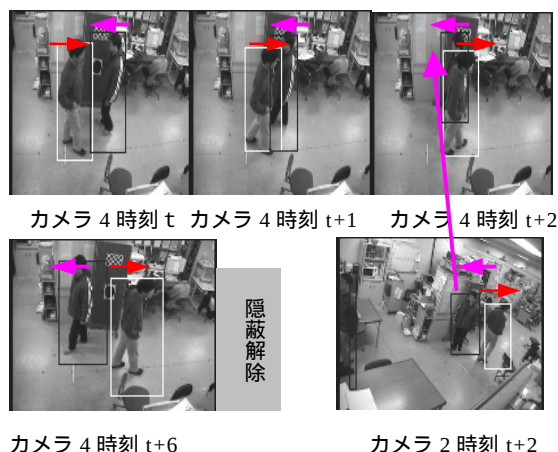


図 9： 人物同士が重なっているとき

他のカメラでも重なりが発生している場合、各カメラ自身の情報をしか使えない。人物同士が離れるまで、後ろにいる人物の平均フローと追跡ウィンドウの大きさを固定して追跡を行う。

以上は、人物の移動速度が異なる場合の隠蔽追跡である。もし、二人の人物がほぼ同じ向き、速

さで移動すると、全てのカメラから見ても、人物領域を区別できなくなり、上記の手法は適用できなくなる。

4 多視点カメラシステムの応用

4.1 多視点カメラシステム

本研究では、室内の天井付近に配置された4台のアクティブカメラを利用する。これらはあらかじめキャリブレーションがとれているものとする。

4.2 人物の実世界位置推定

本研究では、カメラ間で交換している情報は指定人物の実世界位置である。ここでは指定人物の実世界位置の計算方法について説明する。

本稿では、人物は床に対して垂直に立っていると仮定する。また、人物のモデルとして円柱を考える。ここでは、まず2つ以上のカメラが指定人物を追跡しているときの実世界位置の計算について説明する。

実世界における人物の重心の画面への投影を、平均フローに類似したフローを持つ領域の重心で近似する。画面上の人物領域の重心座標を $\bar{x} = [x, y, 1]^t$ とし、世界座標系での人物の重心座標を $\bar{X} = [X, Y, Z, 1]^t$ とすると、

$$h\bar{x} = C\bar{X} \quad (1)$$

と書ける。ここで h はスケール係数であり、 C はカメラパラメータから作った 3×4 の行列である。追跡人物領域が完全に求まるなら、その重心を $\bar{x}$ とすると、(1)式はカメラの光軸中心と目標人物の重心 $\bar{X}$ を通る投影直線となり、2つ以上のカメラによる投影直線の交点が目標人物の位置となる。人物は床に垂直に立っていると仮定しているので、(1)式の直線を含み、床面に垂直な平面を考えると、人物の重心は(2)式で表される面内に存在する。ターゲット平面 と呼ぶ(図10)。

$$\bar{a} \begin{bmatrix} X \\ Y \end{bmatrix} = b \quad (2)$$

ここでは、 $\bar{a}, b$ は式(1)中のカメラパラメータからなる平面パラメータである。

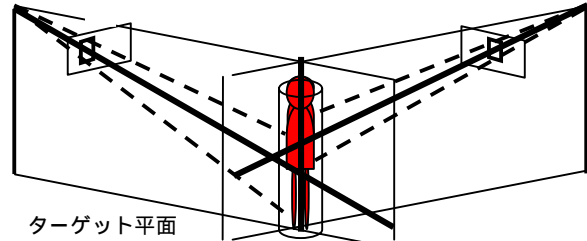


図10：3次元位置の推定

指定人物を追跡しているカメラが複数存在する場合、カメラ i から求まるターゲット平面を

$$\bar{a}_i \begin{bmatrix} X \\ Y \end{bmatrix} = b_i \quad (3)$$

とすると、式(4)により最小2乗法により人物のモデルの中心軸がターゲット平面の交線として求まる。

$$\begin{bmatrix} X \\ Y \end{bmatrix} = (A^t A)^{-1} A^t B \quad (4)$$

ここでは、 $A = [\bar{a}_1 \dots \bar{a}_4]^t, B = [b_1 \dots b_4]^t$ 。

しかし、指定人物を追跡しているカメラが1つしかないとき、上記の方法が適用できない。この場合に、人間の高さを適当に仮定し(例えば1700mm)、空間変換式を用いて実世界位置を計算する方法がある。

4.3 指定人物の検証

指定人物があるカメラの境界に来たときに、反対側から他の人物が入ってくると、指定人物以外の人を指定人物として追跡してしまうことがある(図11)。

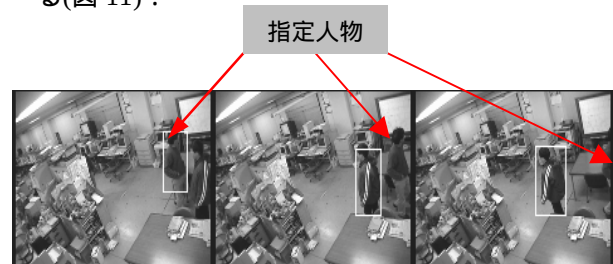


図11：指定人物以外の人物を追跡する

そのため、より安定した追跡を行うため、現在追跡している人物が指定人物であるかどうか検証する必要がある。本研究では、四つのカメラを用いているので、最大四つのターゲット平面が得られる。全部のカメラが指定人物を追跡しているとき、四つのターゲット平面はほぼ1点で交差する(図12)。

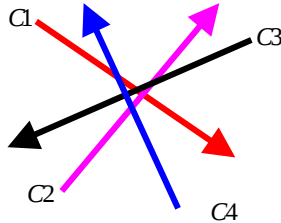


図12：追跡の異常がないときのターゲット平面の様子

しかし、あるカメラが指定人物以外の人物を追跡している場合、ターゲット平面は複数箇所で見切れる(図13)。

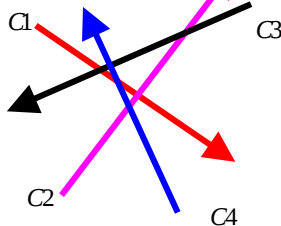


図13：追跡の異常があるときのターゲット平面の様子

このような場合、どのカメラが指定人物以外の人物を追跡しているかを判断するため、四つのカメラの中から三つの追跡情報を用いて実世界位置を計算する。 $C_3 = 4$ 回の最小2乗法を行った後、誤差の一番小さいカメラ組みが正しい追跡をしていると考え、残りのカメラは指定人物以外の人物を追跡していると判断する。そして、誤差最小のカメラ組の指定人物実世界位置情報を利用して、誤った人物を追跡しているカメラにおける指定人物の存在領域を推定する。

図13では、カメラ組(C_1, C_2, C_4)の誤差が最小であるので、カメラ3が指定人物以外の人物を追跡していると判断できる。

しかし、2つのカメラが誤った人物を追跡して、残りの2つのカメラは正しい人物を追跡している場合や、あるカメラで指定人物が障害物に隠蔽されている場合は利用できる追跡情報が三つ以下になり、上記の方法で問題を解決することはで

きない。このときは、指定人物から一番近いカメラの情報を信頼して、指定人物の予測領域を切り出し、追跡を行う。

5 PC クラスタシステムの応用

本稿では、リアルタイムに追跡を行うため、PC クラスタを利用して、並列に画像データを処理する。以下の図14を用いて、カメラシステムとPC クラスタの仕組みを説明する。

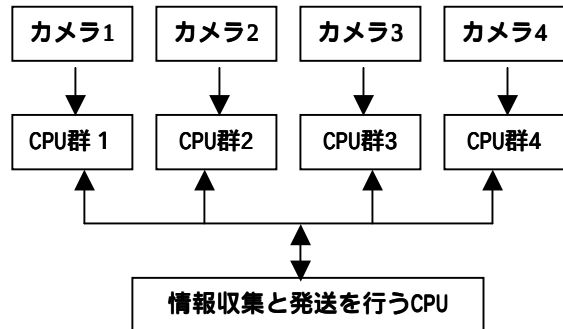


図14：PC クラスタとカメラとの仕組み

図14で示しているように、1つのカメラは1つのCPU群と繋がっている。情報収集と発送を行うCPUは四つカメラの間の情報交換を担当している。図14の1つのCPU群は9個のCPUを含んでいる。この9個のCPUにより、カメラから取り込んだ画像データを並列に処理する。図15は並列処理手法を示している。

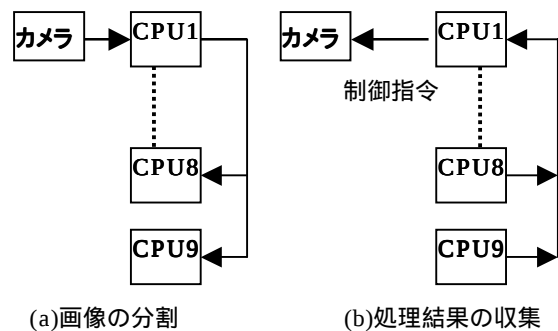


図15：並列処理

図15(a)の画像分割では、CPU1で、画像情報を9部分に分けて、それぞれのCPUに送って、フローを計算する。図15(b)の処理結果の収集では、各CPUで計算した結果を全部CPU1に集め、その結果によって、指定人物を追跡する。また、指定人物の画像情報を情報収集と発送を行う

CPU に送り，指定人物の実世界位置を計算する．計算した実世界位置情報を各 CPU1 に送り返す．各 CPU1 は，その結果によって，カメラへ制御指令を出し，カメラのパン，チルトを変える．

6 実験結果と考察

四台のアクティブカメラと PC クラスタシステムを用いて，本稿で提案した手法でリアルタイム追跡を行った．リアルタイムに処理した時系列結果は図 19 に示す．画像のないところは該当カメラが回転中であることを意味する．指定人物は白いウィンドウで囲む人物である．

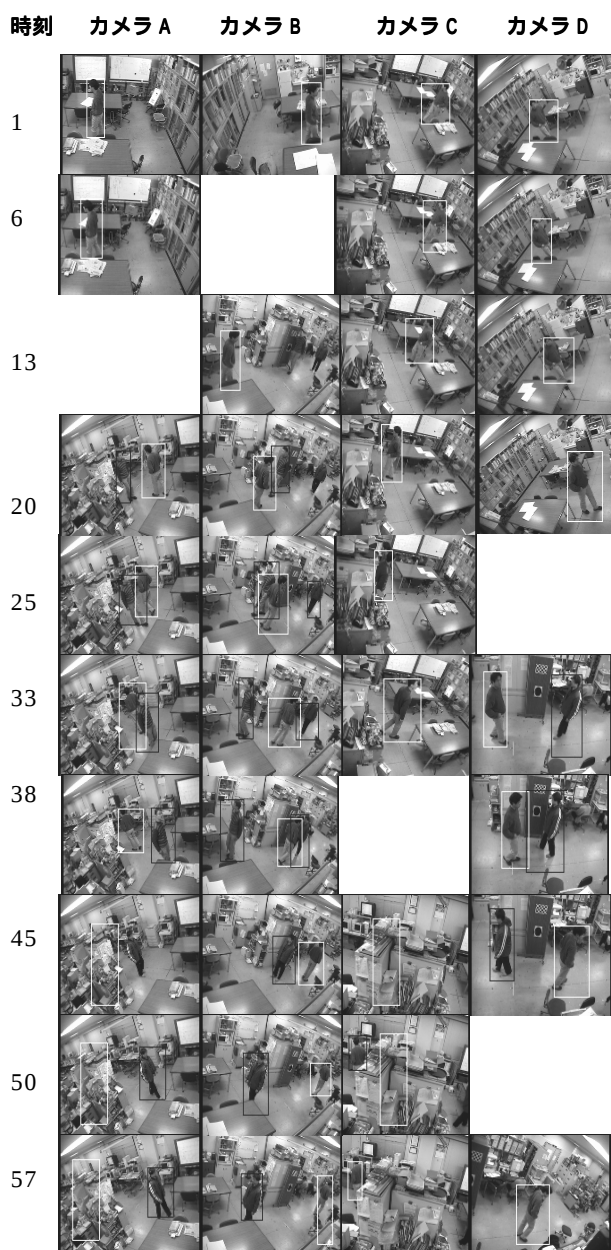


図 19：リアルタイム追跡結果

図左の数字は時刻を現す．時刻 1 の画像列のところ，指定人物はカメラ B の視野の境界に来ているので，カメラ B の向きを変える．時刻 20～45 はカメラ A，B，D で人物同士が重なっているときの追跡である．時刻 25～45 ではカメラ C で検証より指定人物が訂正されている．時刻 45～57 はカメラ A で障害物に隠蔽されているときの追跡である．

7 おわりに

本稿ではオプティカルフローに基づき，複数のアクティブカメラを用いて指定人物を追跡する手法を提案した．複数のカメラの情報交換により，あるカメラで追跡できなくなったとしても，他の追跡できているカメラからの情報を用いることにより，間違いなく追跡を続けることができる．また，PC クラスタを使うことにより，高速にオプティカルフローを計算し，リアルタイム指定人物追跡を実現した．

問題点として，人物同士がほぼ同じ速度で重なっている場合，追跡を失敗することが挙げられる．今後，この状況を解決できるような追跡を行う予定である．

参考文献

- [1] 岡田，加賀，稲葉，井上：“PC による高速対応点探索に基づくロボット搭載可能な実時間視差画像・フロー生成法と実現”，日本ロボット学会誌，Vol. 18, No. 6, pp 896-901, 2000.
- [2] A.Shio and J.Sklansky. “Segmentation of people in motion”. In IEEE Workshop on Visual Motion, pp. 325-332, 1991.
- [3] 嶺，八木，谷内田．“時系列差分画像を用いた複数移動物体の運動検出”．情報処理学会研究報告 CV81 - 7，1992．
- [4] D.H.Ballard, C.M.Brown，福村晃夫他(訳)．“コンピュータ ビジョン”，第 3-2-1 章，日本コンピュータ協会，1987．
- [5] 山根，島田，三浦，久野，白井．オプティカルフローと明度一様領域を統合した人間の長時間追跡．白井修士論文．1998．