

# バレーボールの試合映像からのボール・選手領域抽出

生産システム工学専攻 伊藤 健

## Study on Ball and Player Area Extraction from Game Image of Volleyball

Ken Itoh

Advanced Course of Production System Engineering, Toba National College of Maritime Technology

### Abstract

In recent years, developing the home video camera have begun to be used for data analysis in sports. In particular, volleyball has allowed to analyze the game data by scouting system during the game. Therefore many national volleyball teams are using scouting system, and Japan National Team of Volleyball too. However, less in general teams. Because it's very expensive and difficult to operate.

So, I have developed automatically tracking system for the player position in game vision of volleyball. I adopted the Template Matching to tracking the player in this research. There are two steps to extract the player as follows: (1) To extract player areas by using the picture processing to improve the pursuit success rate by limiting the range of the search only to the player area. (2) To estimate player positions and to resolve the Occlusion (a player gets out of sight because to covering other players) by using Kalman Filter.

In this paper, I'll state about the experiment to pursue the player from the game vision and how many players could we pursue in detail. I used a picture of the volley ball game with the home video camera, and the picture that the image was converted into JPEG images is used as an object image. As a result of proceeding to six players before me, I succeed to track about 100% of the player position.

**Keywords:** Volleyball, Player Area Extraction, Player Position Tracking, Template Matching, Kalman Filter, Occlusion

### 1. はじめに

近年、家庭用映像機器の発展を背景に、個人でも手軽に高画質の映像を撮影できるようになりつつある。そのため、映像情報はスポーツのプレーシーン解析などの研究に利用されてきている。中でもバレーボールは他のスポーツとは異なり、試合中にデータ収集や相手チームの分析を行い監督等にアドバイスを送るアナリストの存在が認められている。以前までは手作業によるデータ分析が行われていたが、最近ではコンピュータを用いたスカウティングシステムと呼ばれるソフトウェアが利用されている。瞬時にデータの解析が行えるようになり、より高度かつ複雑な分析が可能となった。現在では女子バレーボール日本代表もスカウティングシステムを導入するなど、バレーボール競技の情報化は世界的に活発化しつつある。

一方で、スカウティングシステムの発展は様々な問題を生み出す要因にもなっている。現在市販されているスカウティングシステムの中に、(株)バレーボール・アンリミテッドによる“DataVolley”[1]がある。

DataVolley はナショナルチームをはじめ、多くのトップレベルのクラブチームが使用している。このソフトウェアはデータ入力をキーボードで行い、試合を見ながら適切な記号を入力する必要があるため、パソコンの扱いに慣れている人でも利用のためには訓練が必

要となる。また非常に高価格な製品であり、中高生などの一般的なチームが利用することは難しい。

その他にも“TouchVolley”[2]や(株)ミカサより発売されている“VSCT”[3]などがある。これらは低コストと直感的な操作を特徴としており、DataVolley に比べると初心者でも扱いやすい。しかし、結局はデータ入力を手入力で行う必要があり、入力者の主観による誤入力や操作ミスが発生する場合がある。

スポーツ分野においては、試合映像から自動的にボール・選手位置の追跡やイベント抽出を行う研究は多数存在するが、特に問題とされるのがオクルージョン（選手同士の重なり）である。複数の高性能カメラを用いる手法が広く提案される中で、単視点映像を用いてオクルージョンの克服を試みた研究は確かに存在するが、バレーボール等の3人以上のオクルージョンが発生し得るスポーツに適用できる手法は未だ例が無い。

そこで本研究では、単視点からなる試合映像において、オクルージョンに対応した選手位置追跡を可能とするシステムの開発を行った。状態推定器であるカルマンフィルタ[4]をテンプレートマッチング[5]に応用し、オクルージョンへの対応を実現する。

本稿では家庭用ビデオカメラで撮影した試合映像から選手位置を追跡する評価実験を行い、その結果から本手法の有用性と問題点について述べる。

## 2. 選手位置追跡の流れ

選手位置の追跡は、テンプレートマッチングを用いて行う。選手の特徴が表れている部分、本研究では選手の背番号をテンプレートに選んでフレームごとにテンプレートマッチングを行い、選手位置を追跡していく。また、状態推定器であるカルマンフィルタを利用した選手位置推定を実装することで、オクルージョン対策と追跡の精度向上を実現している。選手位置追跡の処理の流れを図 2.1 に示す。流れの中の各プロセスについては、次章以降で説明していく

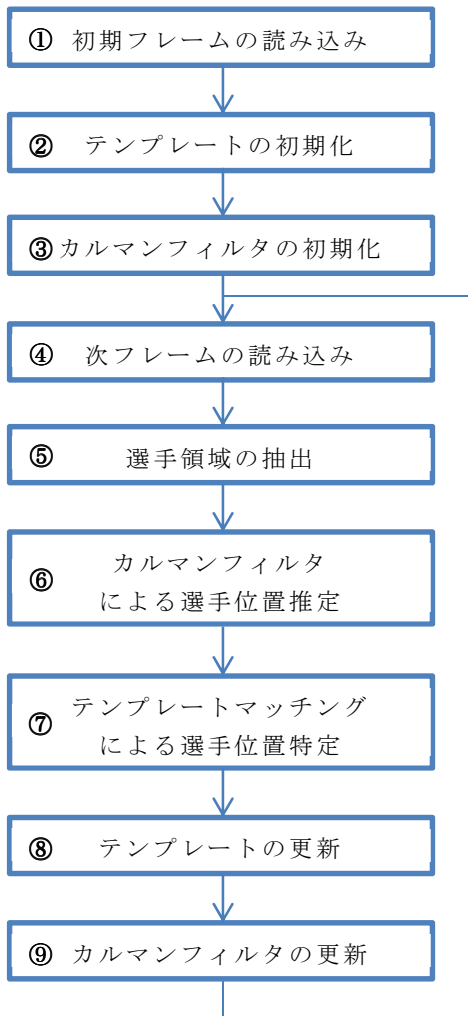


図 2.1 選手位置追跡の流れ

## 3. 選手領域抽出

### 3.1. 選手領域の抽出方法

選手位置の追跡を行う前処理として、対象選手とそれ以外の領域を区別するために図 2.1⑤に示した選手領域の抽出を行う。図 3.1 に選手領域抽出を行う際の処理の流れを示す。なお、各処理の実装にはオープンソースのコンピュータビジョンライブラリである OpenCV[7]を利用した。

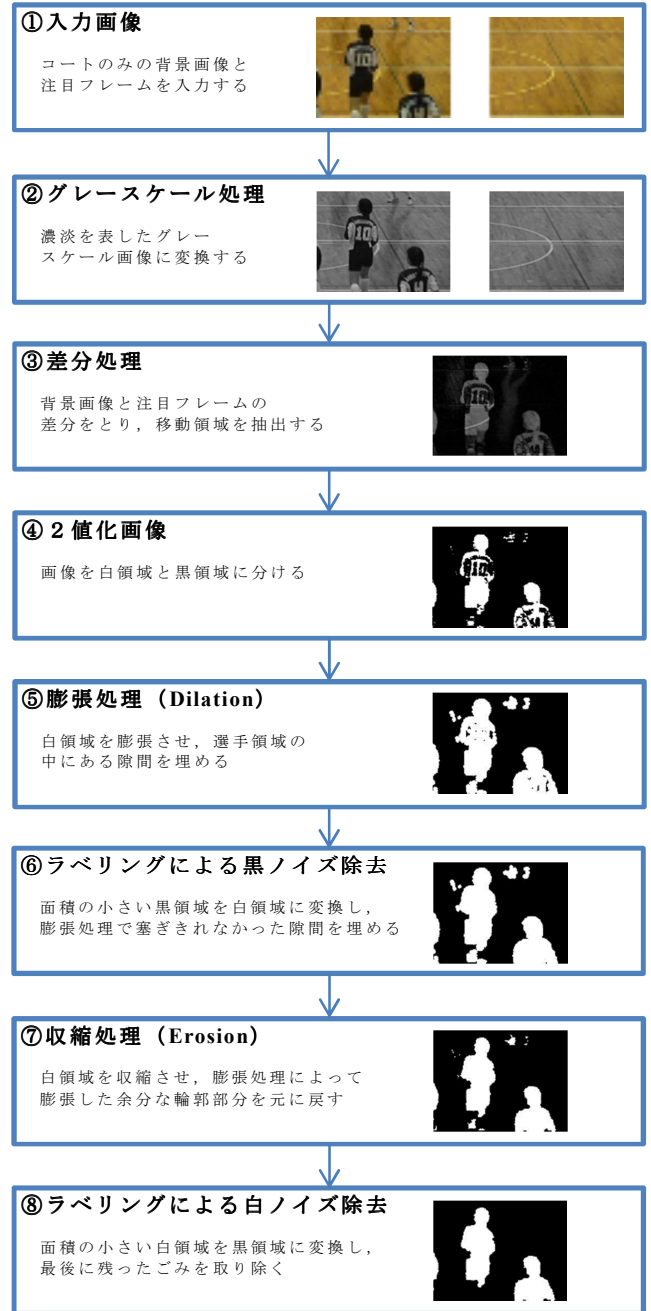


図 3.1 選手領域抽出の流れ

### 3.2. グレースケール処理

図 3.1②のグレースケール画像[5]とは各画素の濃淡（明暗）の強さの値のみによって表現される画像で、濃淡をどれだけ細かい段階で表すかによって濃淡の滑らかさが決まる。単に濃淡画像とも呼ばれる場合もある。以降の処理では、グレースケール画像に変換することにより処理の簡易化・高速化を図ることが可能になる。

実際にカラー画像からグレースケール画像に変換する処理の流れは、注目する画素の RGB 成分の平均値を、その画素の新しい濃度値にすることで実現できる。任意の点  $(x, y)$  における画素の赤成分を  $f_R(x, y)$ 、緑成分を

$f_G(x,y)$ , 青成分を  $f_B(x,y)$  とした場合, グレースケール処理を行った画素値  $f_{gray}(x,y)$  は (3.1) のように表現できる.

$$f_{gray}(x,y) = \frac{f_R(x,y) + f_G(x,y) + f_B(x,y)}{3} \quad (3.1)$$

また, (3.1) に基づいてグレースケール処理を行った結果を図 3.2 に示す.



(a)元画像 (b)グレースケール画像

図 3.2 グレースケール処理

### 3.3. 背景差分

図 3.1③で利用した差分処理[5]とは, コートのみの背景画像を用意し, 対象画像と比較して同座標の画素同士の引き算を行うことである. これにより本来コートには無い物体, つまり選手やボール等を検出できる. ここで, 画像の位置  $(x,y)$  の時刻  $t$  における画素の値を  $f_t(x,y)$ , 背景画像における注目画素の値を  $f_m(x,y)$  とし て差分処理  $f_{sub}(x,y)$  を (3.2) に示す.

$$f_{sub}(x,y) = |f_t(x,y) - f_m(x,y)| \quad (3.2)$$

これにより背景と類似する領域は画素値が 0 に近い値 (黒) となり, 物体が存在する領域は画素値が大きくなる. 図にグレースケール処理を適用した元画像と, それらで差分をとった画像を示す.



(a)元画像 (b)背景画像 (c)差分画像

図 3.3 背景差分

### 3.4. 2値化処理

④に示した 2 値化処理[6]とは, 画像をある閾値  $Th$  ( $0 \sim 255$ ) を境に白領域 (画素値 1) と黒領域 (画素値 0) に分ける処理である. 注目画素の濃度値と閾値を比較して, 濃度値が閾値以上なら注目画素を白く, 閾値未満なら注目画素を黒くする. 2 値画像  $f_{bin}$  は (3.3) で表される.

$$f_{bin} = \begin{cases} 1, f(i,j) \geq Th \\ 0, f(i,j) < Th \end{cases} \quad (3.3)$$

選手領域を欠けることなく抽出するには閾値を低く設定する必要がある. 試合映像は全体を通して明るさの変化が小さいことから, 本研究では  $Th = 48$  の固定閾値を採用した. 2 値化処理の様子を図 3.4 に示す.



(a)元画像 (b)2値画像

図 3.4 2 値化処理

### 3.5. 膨張処理

⑤の膨張処理[6]とは Dilation と呼ばれ, 処理対象画像の図形の縁にそってフィルタをかけてずらしていき図形を膨張させる処理である. 膨張処理を行うことで小さな穴等がふさがれ分離していた領域が結合することもある. 本研究においては選手領域の中にできる隙間を埋めるために用いる. 図 3.5 に元画像(a), 膨張処理を行った画像(b)を示す.



(a)元画像 (b)膨張処理後の画像

図 3.5 膨張処理

### 3.6. 収縮処理

⑦の収縮処理[6]とは Erosion と呼ばれ, 処理対象画像の図形の縁にそってフィルタをかけてずらしていき図形を収縮させる処理である. 収縮処理を行うことで細かい領域は削られて無くなり, 内部の穴が拡大するなどの特徴があるが, 本研究においては膨張処理で太くなった輪郭を元に戻すことを目的としている. そのため, 収縮処理は膨張処理後, 面積の小さい黒領域を除去して選手領域の隙間を完全に埋めたあとに実行する. 隙間を埋める処理の具体的な内容は次節で説明する. 図 3.6 に元画像(a), 収縮処理を行った画像(b)を示す.



(a)元画像 (b)収縮処理後の画像

図 3.6 収縮処理

### 3.7. ラベリング処理

⑥,⑧で利用したラベリング処理[5]とは, 2 値化された画像において連結している領域に同じ番号を割り当てることで, 対象画像の解析を行う手法の一つである. ラベリングの概念を図 3.7 に示す.



(a)元画像 (b)黒領域の処理 (c)白領域の処理  
図 3.7 ラベリング処理

これにより、画像中の対象物の数や各領域の面積(画素数)を計測することが可能となる。本研究においては、面積が小さい領域を取り除くために利用している。

黒領域に対するラベリング(図 3.7(b))を行い、面積が小さい黒領域を取り除くことで、膨張処理によって塞ぎきれなかった選手領域の隙間を埋めることができる(図 3.8)。



(a)元画像 (b)黒ノイズ除去後  
図 3.8 黒ノイズ除去

また、白領域に対するラベリング(図 3.7(c))を行い、面積が小さい白領域を取り除くことで、最後まで処理しきれなかった余分な領域を消すことができる(図 3.9)。



(a)元画像 (b)白ノイズ除去後  
図 3.9 白ノイズ除去

白ノイズ除去は選手領域抽出の中で最後に行われる処理である。3.7節で述べた通り、黒ノイズ除去後に収縮処理を行ったあとで実行される。

## 4. カルマンフィルタ

### 4.1. カルマンフィルタとは

カルマンフィルタとは、(離散的な)誤差のある観測から、時間的に変化する状態を推定するために用いられる方法である。カルマンフィルタは機能工学分野で広く用いられており、例えばカーナビゲーションでは人工衛星からの誤差のある情報にカルマンフィルタを適用して、最も確からしい現在位置の推定に応用されている。

### 4.2. カルマンフィルタの動作

カルマンフィルタは、時間ステップを一つ進めるた

めに“予測”と“更新”の二つの手続きを行う。予測の手続きでは、前の時刻の推定状態から、今の時刻の推定状態を計算する。更新の手続きでは、より正確な推定を行うために、今の時刻の観測値を用いて推定値を補正する。標準的なカルマンフィルタにおける状態値と観測値の定義式を以下に示す。

$$x_k = A \cdot x_{k-1} + B \cdot u_k + p_k \quad (4.1)$$

$$z_k = H \cdot x_k + q_k \quad (4.2)$$

ここで、 $x_k$ は時刻 $k$ におけるシステムの真の状態、 $z_k$ は時刻 $k$ における観測値、 $u_k$ は時刻 $k$ における制御入力を表している。 $p_k$ と $q_k$ はそれぞれ正規分布に従うプロセスノイズ、および観測ノイズである。また、 $A$ はシステムの状態遷移行列、 $B$ は制御行列、 $H$ は観測行列である。

やや複雑な式に見えるが、要するに「システムにノイズ $w_k$ 、観測値にノイズ $v_k$ が加わっているとき、システムがどのような挙動をするか尤もらしい推定を行う(状態変数 $x$ を求める)」ということである。

本研究においては、各選手の位置座標と速度を状態として扱い、テンプレートマッチングによって特定した選手位置を観測として扱う。すなわち、状態2次元、観測1次元のカルマンフィルタを構成することになり、座標を $i$ 、速度を $v$ とすると、

$$x = \begin{bmatrix} i \\ v \end{bmatrix}, z = [i]$$

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, H = [1 \ 0]$$

となる。本研究では制御入力を用いないため、結局(4.1)(4.2)の関係は

$$x_k = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} i_{k-1} \\ v_{k-1} \end{bmatrix} + p_k = \begin{bmatrix} i_{k-1} + v_{k-1} \\ v_{k-1} \end{bmatrix} + p_k \quad (4.3)$$

$$z_k = [1 \ 0] \begin{bmatrix} i_k \\ v_k \end{bmatrix} + q_k = i_k + q_k \quad (4.4)$$

と表現される。つまり、等速度移動に誤差情報が加算されるだけの単純なものとも言える。なお、カルマンフィルタにおけるプロセスノイズと観測ノイズは平均値0の正規分布型のホワイトノイズである。

追跡処理において観測される座標は、テンプレートマッチングによって特定された選手位置となる。本稿の実験でのカルマンフィルタの動作は、以下に示す①~③の繰り返しである。

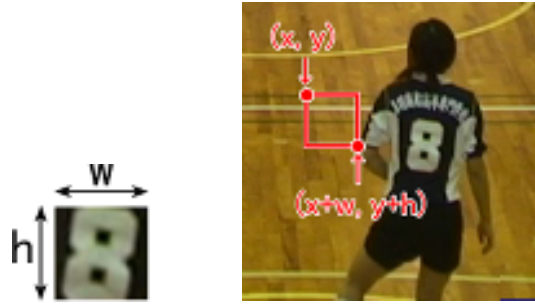
- ①カルマンフィルタで現在の選手位置を推定する。
- ②推定した選手位置の周辺でテンプレートマッチングを行い、実際の選手位置を特定する。
- ③特定した選手位置をもとにカルマンフィルタを更新し、次の推定をより正確にする。



## 5. テンプレートマッチングによる選手位置特定

### 5.1. テンプレートマッチングとは

テンプレートマッチングとは、対象画像の中からあるパターンを見つけ出す手法の一つである。今回は選手の特特定が目的であるため、パターンを表すテンプレートとして図 5.1(a)に示すように選手の背番号を採用した。このテンプレートを用いて図 5.1(b)のように対象画像を走査し、テンプレートと類似する領域を特定する。



(a)テンプレート (b)対象画像  
図 5.1 テンプレートマッチング

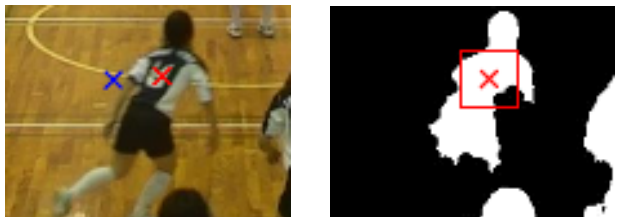
注目領域とテンプレートの類似度は (5.1) の相関計算によって求める。相関値  $r$  は  $-1 \sim 1$  の値をとり、相関値が  $0$  の場合は相関が無く、 $1$  に近づくほど相関が強くなる。また、相関値が負の値をとる場合は負の相関があると言える。

$$r = \frac{\sum_{i=1}^w \sum_{j=1}^h (a_{i,j} - \bar{a}) \sum_{i=x}^{x+w} \sum_{j=y}^{y+h} (b_{i,j} - \bar{b})}{\sqrt{\sum_{i=1}^w \sum_{j=1}^h (a_{i,j} - \bar{a})^2} \sqrt{\sum_{i=x}^{x+w} \sum_{j=y}^{y+h} (b_{i,j} - \bar{b})^2}} \quad \dots (5.1)$$

$a$ : テンプレートの濃度値  
 $b$ : 対象画像の濃度値  
 $\bar{a}$ : テンプレートの平均濃度値  
 $\bar{b}$ : 対象画像の平均濃度値  
 $w, h$ : テンプレートの大きさ  
 $x, y$ : 探索開始座標

### 5.2. 探索範囲の決定

選手が 1 フレームの間に動ける距離には限界があるため、テンプレートマッチングを行う際に画面全体を操作することは処理時間の無駄であり、探索失敗の可能性も高くなってしまう。そこで、3 章で抽出した選手領域と 4 章のカルマンフィルタを利用して、探索範囲を限定したテンプレートマッチングを行う。その様子を図 5.2 に示す。



(a)予測した選手位置 (b)マッチングを行う範囲  
図 5.2 探索範囲の決定

図 5.2 は選手が左から右に動いている時の画像である。図 5.2(a)の左側の×印は前フレームで推定した選手位置、右側の×印は今のフレームで推定した選手位置である。選手位置を推定し、推定した位置の周辺（本研究では推定位置から  $\pm 5$  pixel）のみでテンプレートマッチングを行うようにすれば、偶発的な誤認識を防止できる。さらに、背景領域を追跡してしまう可能性を無くすために、抽出した選手領域のみで 5.1 節の相関計算を行うことにしている。つまり、図 5.2(b)の矩形範囲の中で、白領域のみに対してマッチングを行うことになる。

### 5.3. テンプレートの更新

テンプレートマッチングを行う際、初期フレームから選んだテンプレートだけを使っていると、選手の回転による見た目の変化に対応できない。その具体例を図 5.3 に示す。初期フレーム  $f_0$  から選んだテンプレート (a) では、次フレームにあたる  $f_1$  の選手 (b) を特定することはできるが、100 フレーム後にあたる  $f_{100}$  の選手 (c) を特定することは困難である。



(a)初期テンプレート (b)  $f_1$  の選手 (c)  $f_{100}$  の選手  
図 5.3 テンプレートの更新

そこで、テンプレートマッチングを行うたびに特定した類似領域を新しいテンプレートに選ぶことで、次々にテンプレートを更新していくことにした。これにより、常に前フレームから選んだテンプレートを用いてテンプレートマッチングを行うことができるようになり、選手の見た目の変化への対応が可能となった。

### 5.4. オクルージョンへの対策

選手同士が重なって見えなくなったフレームにおいては、テンプレートマッチングによる特定が不可能になる。オクルージョンが発生している試合映像を例に挙げ、発生フレームを  $f_t$  としてオクルージョンの様子を図 5.4 に示す。



(a)  $f_{t-5}$  (b)  $f_t$  (c)  $f_{t+5}$   
図 5.4 オクルージョンの発生例

選手の体に重なっている矩形領域が、テンプレートマッチングによって特定したテンプレートの位置であ

る。  $f_{t-5}$  から  $f_t$  までは選手を追跡できているが、  $f_{t+5}$  の段階で手前側の選手を追跡してしまい、奥側の選手の追跡に失敗している。

ここで、本稿の実験で扱う試合映像の中からオクルージョンが発生した2人の選手を選び、それぞれオクルージョンが発生したフレーム  $\pm 10$  フレームの類似度を抽出した。グラフに表すと図 5.5 のようになる。

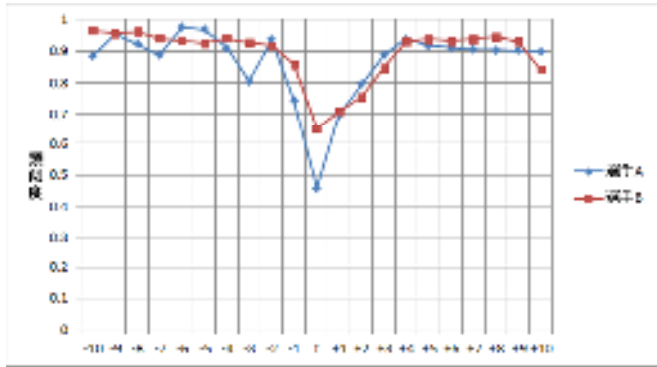


図 5.5 類似度の推移

この結果から、オクルージョンが発生したフレームで類似度が大幅に減少していることが分かる。その直後に類似度が上昇したのは、誤抽出領域を新たなテンプレートとして更新したためである。場面 A のフレーム  $t$  での類似度を  $RA_t$ 、場面 B のフレーム  $t$  での類似度を  $RB_t$  とすれば、オクルージョンが発生した瞬間の下げ幅を (5.2) (5.3) に示す。

$$RA_{t-1} - RA_t \cong 0.286 \quad (5.2)$$

$$RB_{t-1} - RB_t \cong 0.206 \quad (5.3)$$

以上から、「類似度が 0.2 以上減少した」場合にオクルージョン発生フレームとして検出することにした。さらに、選手の見た目が通過前と通過後でほとんど変化しないことに注目する (図 5.6)。



図 5.6 オクルージョン発生前後の選手

つまり、通過前のテンプレートを利用すれば通過後の選手を十分に特定できると考えられる。このことを利用して、オクルージョンが発生する直前の選手の速度とテンプレートを保存し、見えていなくとも等速度運動をしていると仮定して、類似度が大幅に増加する (再び見えるようになった選手を特定する) フレームを検出することとした。このときの類似度の上げ幅を (5.4) (5.5) に示す。

$$RA_t - RA_{t-1} \cong 0.543 \quad (5.4)$$

$$RB_t - RB_{t-1} \cong 0.422 \quad (5.5)$$

以上から、類似度が 0.2 以上減少したらオクルージョン発生、類似度が 0.4 以上増加したらオクルージョン終了と判定させることに決めた。

本実験においては、オクルージョンが発生している間はテンプレートの更新を行わず、カルマンフィルタによって選手位置を推定していく。そして、オクルージョンが終了した時点からテンプレートの更新を再開する、という動作を実装した。

## 6. 評価実験

### 6.1. 実験概要

手前側のチームの選手 6 人に対してテンプレートの大きさを変更しながら追跡処理を行い、各選手を何フレーム目まで追跡できたかを検証した。

実験映像には、家庭用ビデオカメラを用いて撮影した 1 つのラリーの動画から変換した JPEG 形式の静止画を使用した。また、ビデオカメラの性能による違いを検証するために、2 つのビデオカメラを用いて撮影を行った。

- 動画①「SONY HDR-SR8」  
(1440×1080, 30fps, 全 317 フレーム)
- 動画②「SANYO DMX-FH11」  
(1920×1080, 60fps, 全 560 フレーム)

また、各動画の初期フレームを図 6.1, 図 6.2 に示す



図 6.1 動画①の初期フレーム

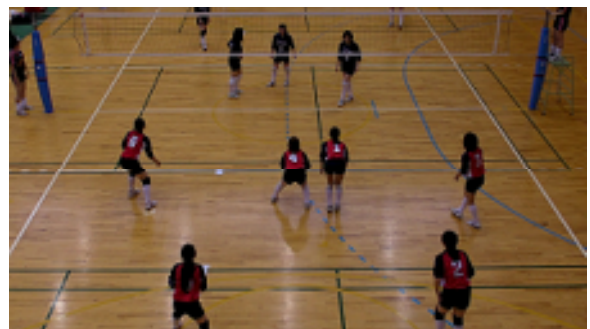


図 6.2 動画②の初期フレーム

## 6.2. 実験方法

動画①と動画②で、各選手が画面内に入ってから出て行くまでのフレームに対して追跡を行う。対象選手の体以外を追跡した時点で追跡失敗とし、評価式として(6.1)の追跡率を定義する。

$$\text{追跡率} = \frac{\text{追跡成功フレーム数}}{\text{選手が存在したフレーム数}} \quad (6.1)$$

## 6.3. 実験結果

今回の実験結果を示す前に、前年度までのカルマンフィルタを適用していなかった手法の結果を表6.1に紹介する。このときの実験では、本実験の動画①を使用していた。

表 6.1 過去の実験結果

テンプレート の大きさ	各選手(背番号)の追跡率(%)					
	4	5	6	8	9	10
21×21	50.7	100	100	100	22.7	100
25×25	100	52.1	100	100	23.0	100
29×29	51.0	22.1	100	100	22.7	100

当時の手法はオクルージョンに対応していないため、オクルージョンが発生している9番の選手は、発生フレームから先を追跡できていなかった。また、選手位置推定が行われなため処理範囲が無駄に広く、テンプレートマッチングの精度が低下していた。

次に、本実験での動画①の実験結果を表6.2に示す。

表 6.2 動画①の実験結果

テンプレート の大きさ	各選手(背番号)の追跡率(%)					
	4	5	6	8	9	10
21×21	100	100	100	100	95.7	100
25×25	100	100	100	100	95.3	100
29×29	100	100	100	98.7	95.3	100

カルマンフィルタによる選手位置推定とオクルージョン検出が成功し、ほぼ100%の追跡率を得ることができた。ただ、8番、9番の選手は画面外に出る直前にオクルージョンが発生しており、オクルージョン終了を検出できなかった。

最後に、動画②の実験結果を表6.3に示す。

表 6.3 動画②の実験結果

テンプレート の大きさ	各選手(背番号)の追跡率(%)					
	1	2	3	4	5	6
21×21	100	100	59.5	100	100	100
25×25	100	100	59.5	82.2	100	100
29×29	100	100	59.5	100	100	64.3

テンプレートの大きさに関わらず3番の選手のオクルージョン検出に失敗しており、オクルージョンが発生したフレームから先を追跡できなかった。この原因は、60fpsの条件下では30fpsに比べて選手の1フレームあたりの移動量が少なく、類似度の変化が緩やかだったため、5.4節で規定した「類似度の0.2以上の減少」が発生しなかったことである。動画②において、手前側の選手の頭部が、少しずつ背番号に重なっていく様子を図6.3に示す。

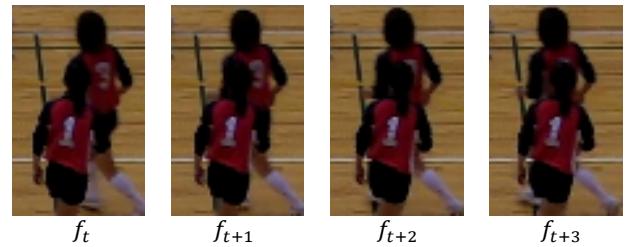


図 6.3 60fpsでのオクルージョン

この問題の解決策として、フレームレートによって条件を変えるか、フレームレートに影響されにくい検出方法を考案することが挙げられる。

また、6番の選手は相手チームの選手と色が同化してしまう箇所があった(図6.4)。テンプレートが大きい場合、相手チームの選手の体も処理範囲に含まれ、そちらを追跡してしまうケースがあった。

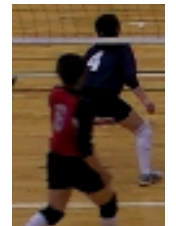


図 6.4 色の同化

色の同化に対応するためには、濃度値以外の要素でのマッチング手法を利用するなど、類似度を求める相関計算の式に工夫が必要である。

上記以外の追跡失敗には、選手領域をうまく抽出できていない部分があり、適切なマッチング処理を行えなかったケースがある。組み合わせる画像処理手法を変更するなど、選手領域抽出の精度向上が求められる。

## 7. おわりに

今回の研究では、複数の画像処理手法を組み合わせ、試合映像における選手位置の追跡を行った。また、カルマンフィルタを実装することで精度向上とオクルージョン対策を実現している。実験の結果、単純なオクルージョンが発生している試合映像ではほぼ100%の追跡率を得ることができた。

今後の課題として、初期テンプレートの自動選択や、選手のフレームイン、フレームアウトへの対策が挙げられる。また、鳥羽商船高専の中井英介による「バレーボールの試合映像からのボール領域抽出に関する研究」[8]で紹介されている手法と組み合わせ、バレーボールのイベント検出を行うシステムを開発する。

## 参考文献

- [1] (有)バレーボール・アンリミテッド：“DataVolley”  
<http://unlimited.volleyball.ne.jp/datav/>  
(2011.02.03)
- [2] 梶原修平，江崎修央，重永貴博，宮地力：“バレーボールのスカウティングシステム TouchVolley の開発”，バレーボール研究，Vol.9，pp.25-30 (2005)
- [3] (株)ミカサ：“デジタルカタログ 2010”  
<http://www.mikasasports.co.jp/catalog/fbook.html>  
(2011.02.03)
- [4] 片山徹：“新版 応用カルマンフィルタ” 朝倉邦造 (2008.07.25)
- [5] 酒井幸市：“画像処理とパターン認識入門” 森北出版 (2006.12)
- [6] 田村秀行：“コンピュータ画像処理” オーム社 (2004.12)
- [7] Opencv.jp：“リファレンスマニュアル”  
[http://opencv.jp/reference\\_manual](http://opencv.jp/reference_manual)  
(2011.02.03)
- [8] 中井英介：“バレーボールの試合映像からのボール領域抽出に関する研究” (2010)