

C-3

選手の3次元位置を追跡する バレーボール分析支援システム

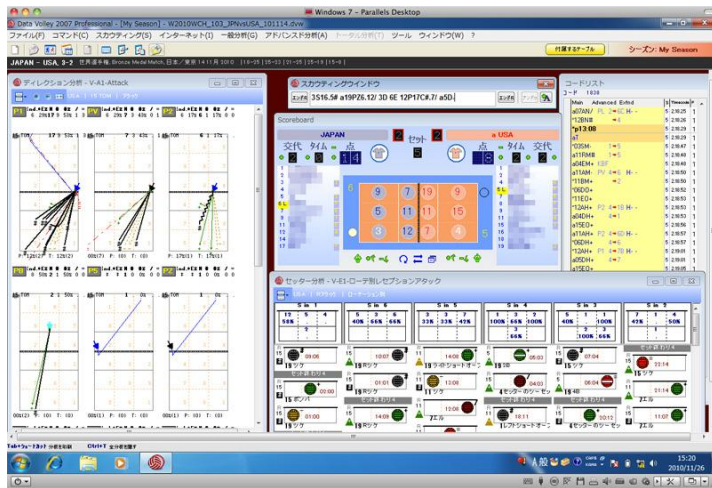
制御工学研究室 佐野裕馬

指導教員 外山茂浩

Toyama System
Design Laboratory



データバレー



課題

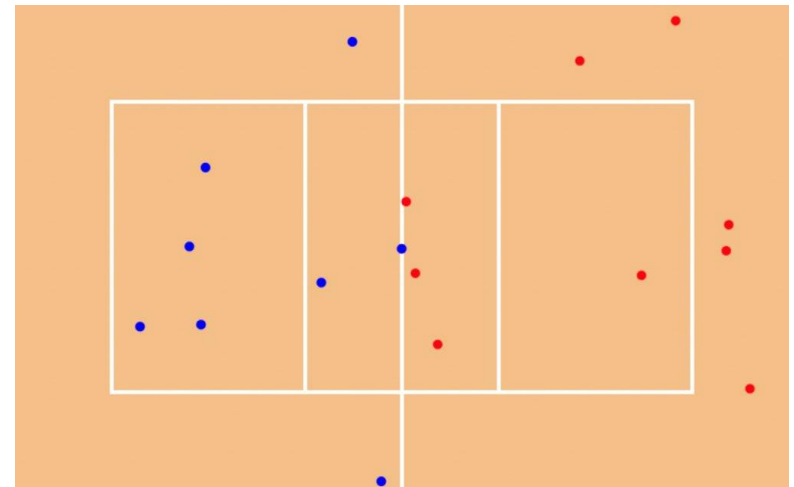
1. 複雑なコマンドを手入力

例: 15S.14#K1a26P4.17

2. 主観による選手位置の決定

先行研究

平田, “情報端末の内蔵カメラを用いた運動再現システム”, R3 長岡高専 電子制御工学科 卒業論文



選手の足が地面についてる場合、
選手の平面位置をカメラ1台で推定可能

跳躍時に位置の推定が不可能

複数台のカメラを用いて選手の3次元位置を追跡

- ①複数台のカメラでコート撮影
- ②カメラキャリブレーション
- ③AlphaPoseによる選手の姿勢推定
- ④映像間の選手の対応付け
- ⑤選手の3次元位置推定

コート撮影



カメラ1



カメラ2



iPhone SE(Apple)

カメラ性能

- 1080 × 1920 pixel
- 30FPS

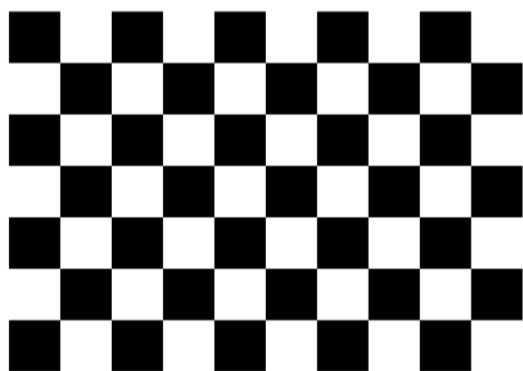
撮影条件

- 複数台のカメラで同時に撮影
- カメラを静止させた状態で撮影

カメラキャリブレーション

カメラ内部パラメータ

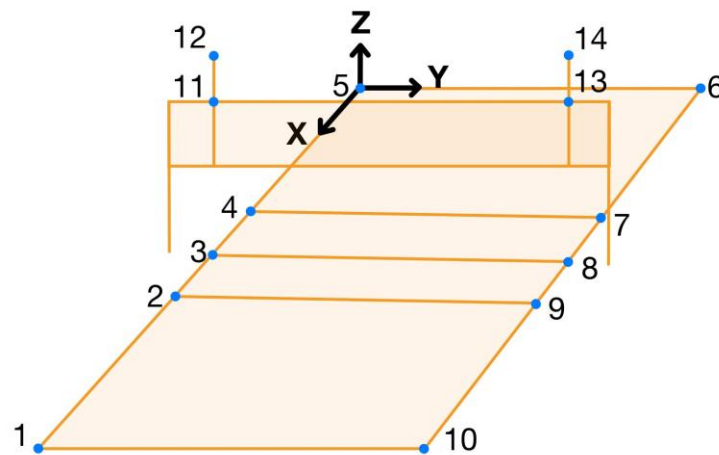
- ・焦点距離
- ・光学的中心



キャリブレーションパターン
(既知の平面パターン)

カメラ外部パラメータ

- ・カメラ位置
- ・カメラ姿勢



バレーボールコートの既知点

AlphaPoseによる選手の姿勢推定



- ・選手へのID振り分け
 - ・選手の腰の位置を取得
- (選手位置は選手の腰の位置に設定)

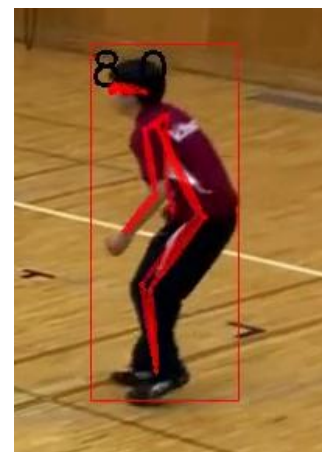
映像間の選手の対応付け(手動)

カメラごとに選手IDが異なる

カメラ1: 12



カメラ2: 8



同じ選手

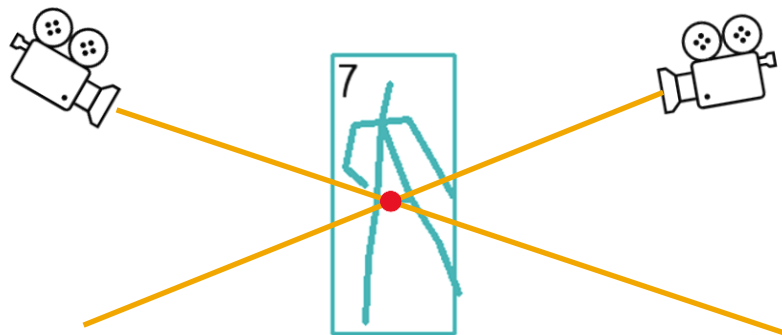


映像の初めのフレームにおいて
目視によって同じ選手を対応付ける

選手の3次元位置推定

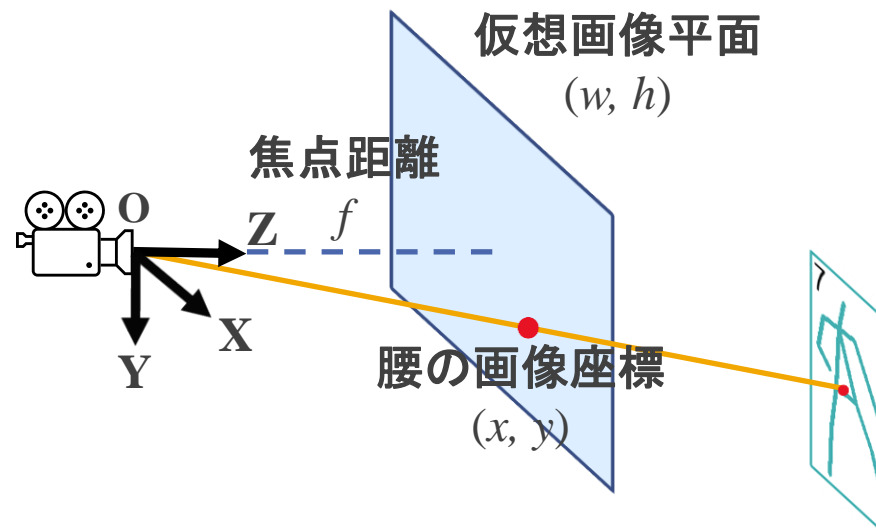
選手位置の推定

ある選手を通る直線を各映像で
選んだときの直線の最近点



カメラ座標系がカメラごとに存在する
→それぞれの直線の座標系が異なり、
最近点を計算できない

カメラ座標系

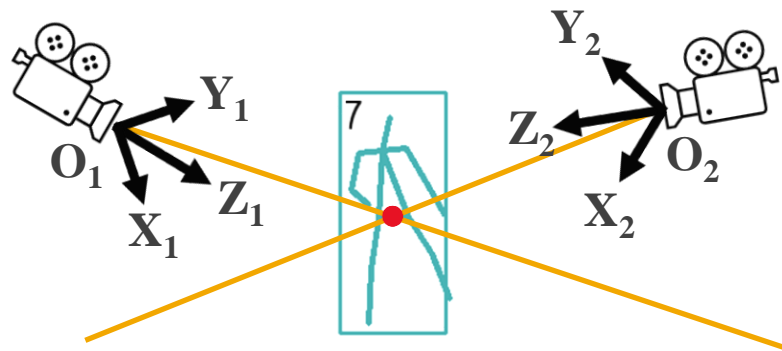


カメラから選手の腰に向かうベクトル

$$V = (x - w/2, y - h/2, f)$$

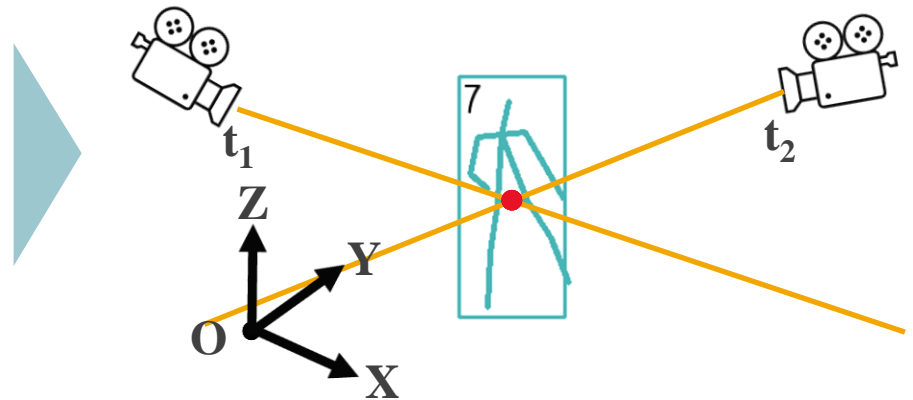
選手の3次元位置推定

カメラ座標系



カメラ座標系はカメラごとに存在
→それぞれの直線の座標系が異なり
最近点を計算できない

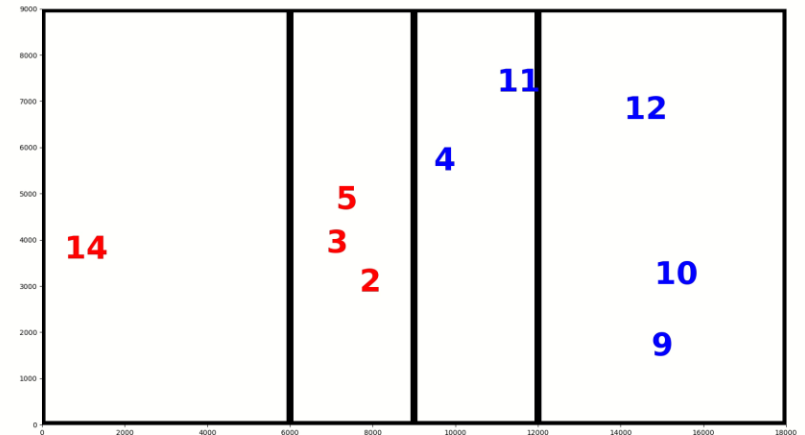
実空間座標系



それぞれの直線が同じ座標系に存在
→最近点を計算できる

選手位置の推定結果(手動)

選手位置をおおむね推定できた
(跳躍時においても問題なく推定できた)



課題

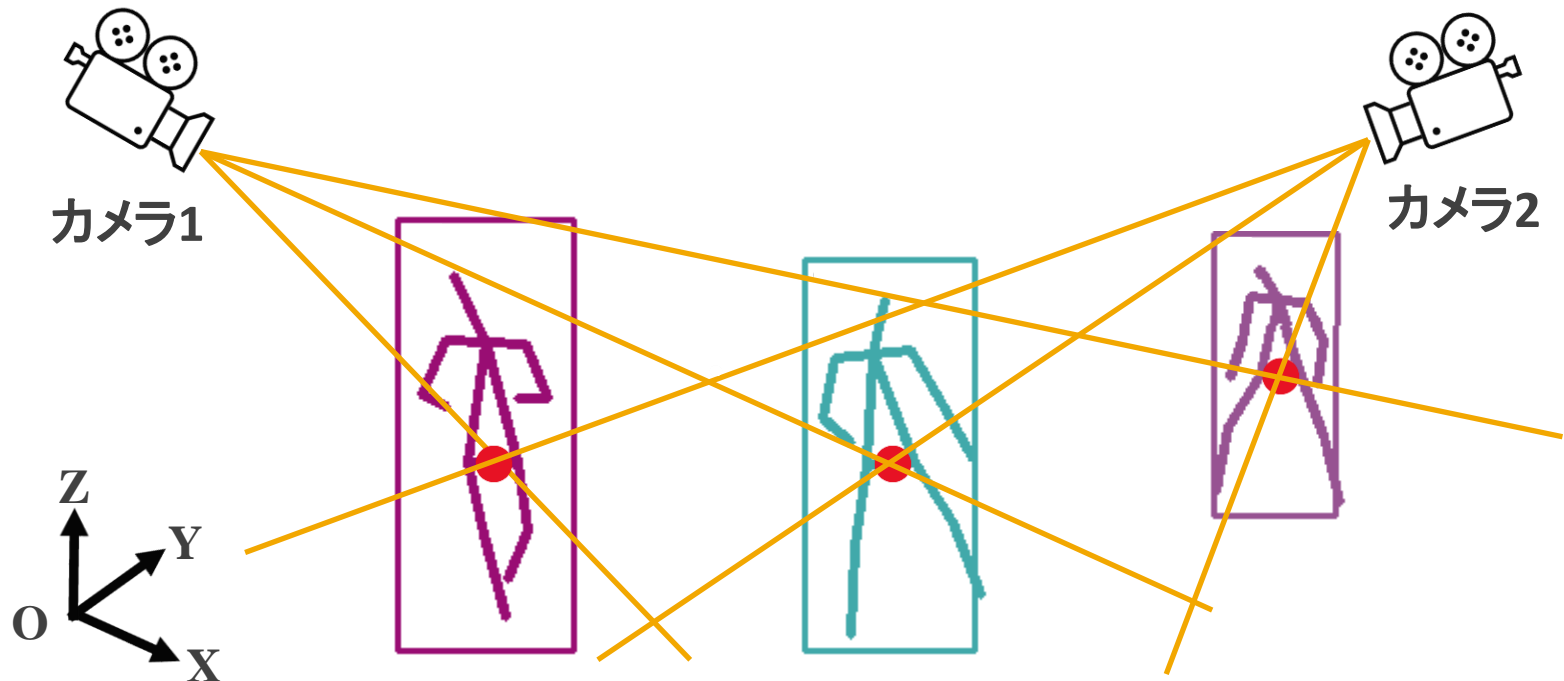
- ・ 選手の交差が起こり、選手位置を推定できなくなる

複数台のカメラを用いて選手の3次元位置を追跡

- ①複数台のカメラでコート撮影
- ②カメラキャリブレーション
- ③AlphaPoseによる選手の姿勢推定
- ④映像間の選手の対応付け(手動)→(自動)
- ⑤選手の3次元位置推定

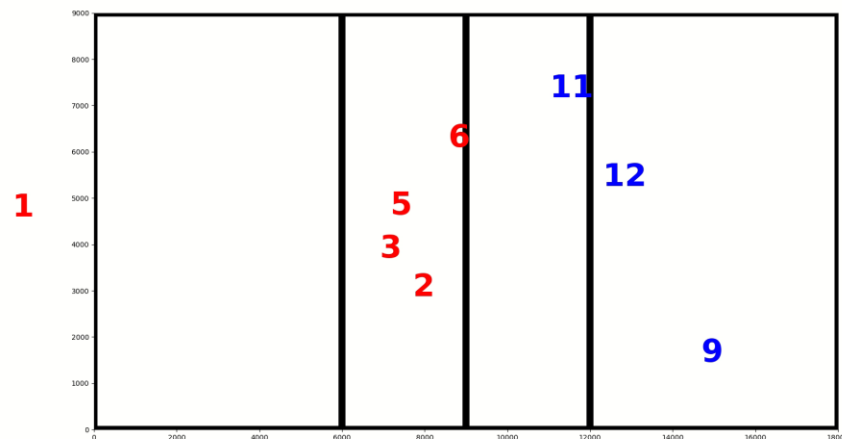
映像間の選手の対応付け(自動)

映像ごとに選手を通る直線を選ぶ全ての組み合わせを考える
(映像の全フレーム)



直線間距離の近い組から順に対応付ける

選手位置の推定結果(自動)



課題：自動の選手対応付けが上手く機能しない



原因：カメラキャリブレーションの誤差

カメラ1：約29cmの誤差

カメラ2：約13cmの誤差

- 複数台のカメラで選手の3次元位置を追跡
- 3次元位置を求めることで
跳躍時においても位置を推定可能
→ 先行研究の課題を解決
- 今後の課題
 - ① 選手の交差への対応
 - ② カメラキャリブレーションの精度向上

目次

- p2 研究背景
- p3 研究目的・内容
- p4 コートの撮影
- p5 カメラキャリブレーション
- p6 AlphaPoseによる姿勢推定
- p7 映像間の選手対応付け(手動)
- p8 選手の3次元位置推定
- p10 推定結果(手動)
- p12 映像間の選手対応付け(自動)
- p13 推定結果(自動)
- p14 まとめ

本研究で撮影・使用した映像

カメラ1



カメラ2

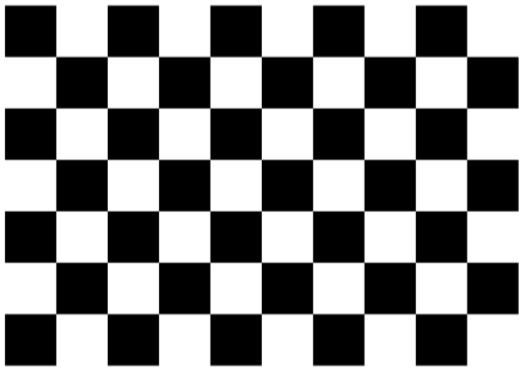


カメラキャリブレーション(カメラ内部パラメータ)

カメラ内部パラメータ

- ・焦点距離
- ・光学的中心
- ・(レンズ歪み)

OpenCVを用いて推定



キャリブレーションパターン
(既知の平面パターン)

カメラレンズの歪み補正



歪み補正前



歪み補正後

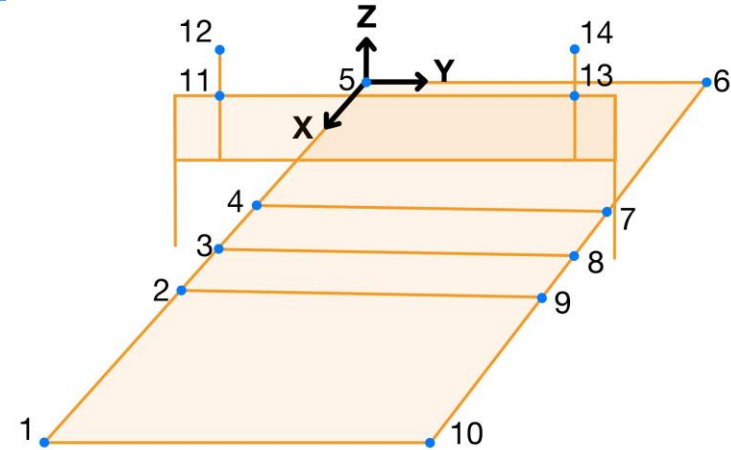
カメラキャリブレーション(カメラ外部パラメータ)

カメラ外部パラメータ

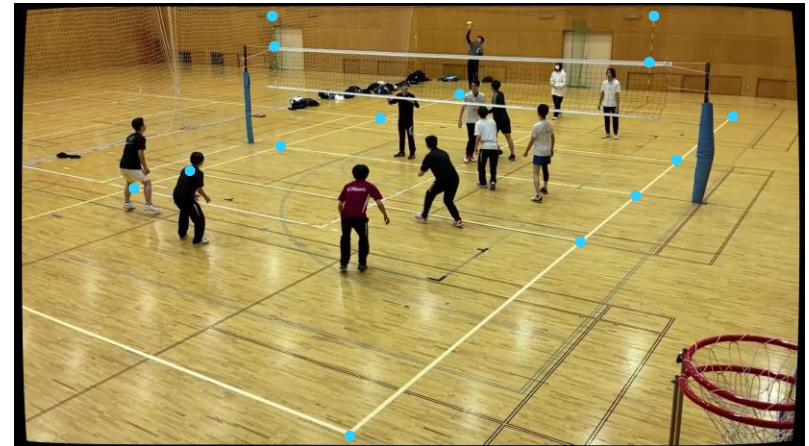
- ・カメラ位置
- ・カメラ姿勢

OpenCVを用いて推定

全ての既知点が
映像に映る必要はない



バレーボールコートの既知点



直線の最近点の求め方

直線L1 : $sv_1 + t_1$

直線L2 : $uv_2 + t_2$

s, u : 媒介変数

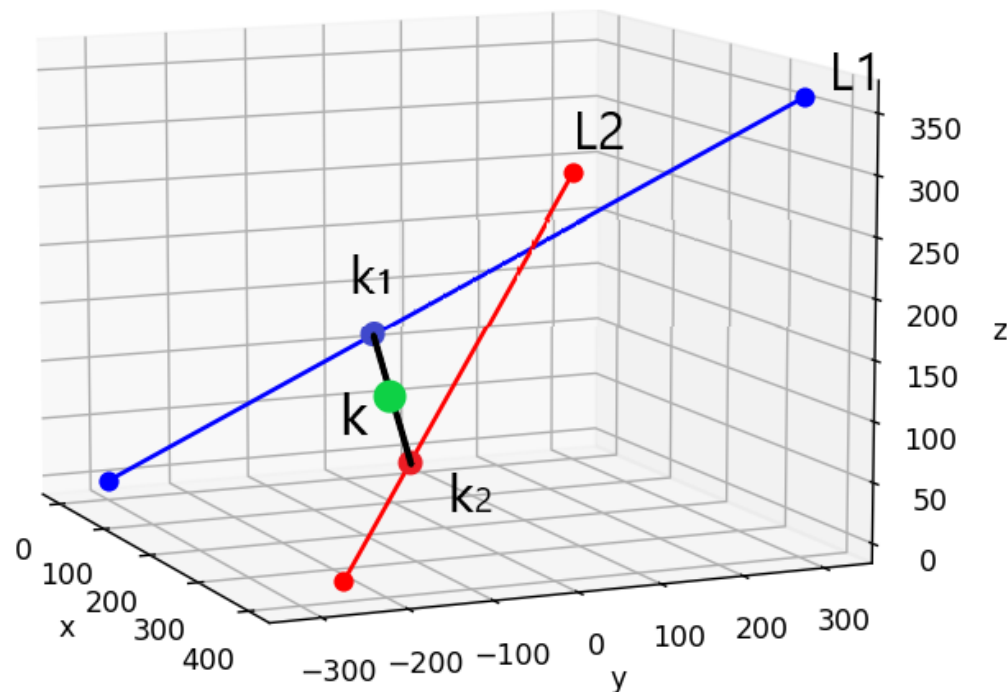
v_1, v_2 : 単位方向ベクトル

t_1, t_2 : 位置ベクトル

→直線L2に対する直線L1上の最近点 k_1

$$k_1 = t_1 + \frac{\{(v_1 - (v_1 \cdot v_2)v_2) \cdot (t_2 - t_1)\}}{1 - (v_1 \cdot v_2)^2}$$

L1に対するL2上の最近点 k_2 も
同様にして求め、 k_1, k_2 の中点を
2直線の最近点 k として求める



選手配置を知るメリット

①相手の攻撃エリア・ゾーンが分かる

→攻撃パターンが分かる

データバレーでも行っていることだが、
正確な位置情報が映像解析によって出せる

②ブロックの参加率を評価できる

→良いブロッカーの評価指標

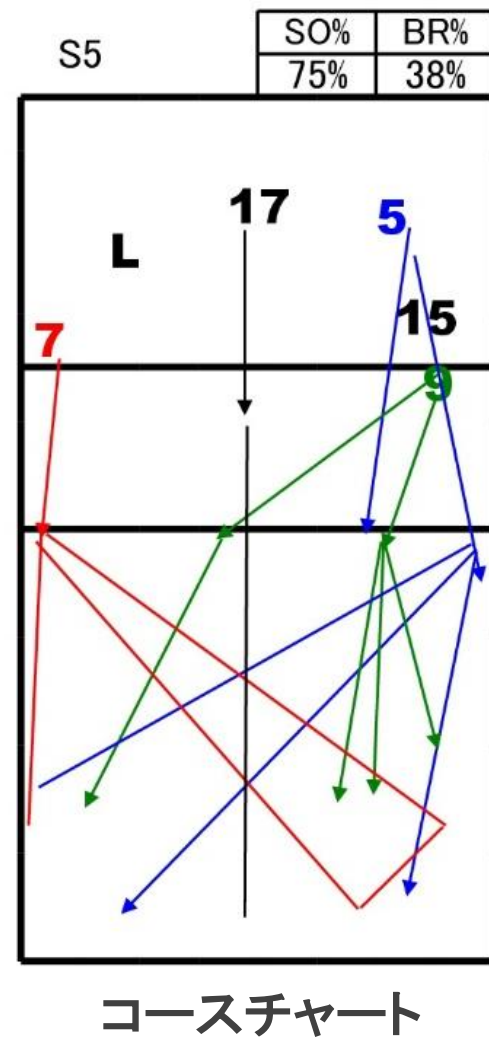
セッターにつられずに2枚揃えられるかどうか

③攻撃者の体の向きが分かる

(3次元の姿勢まで求められた場合)

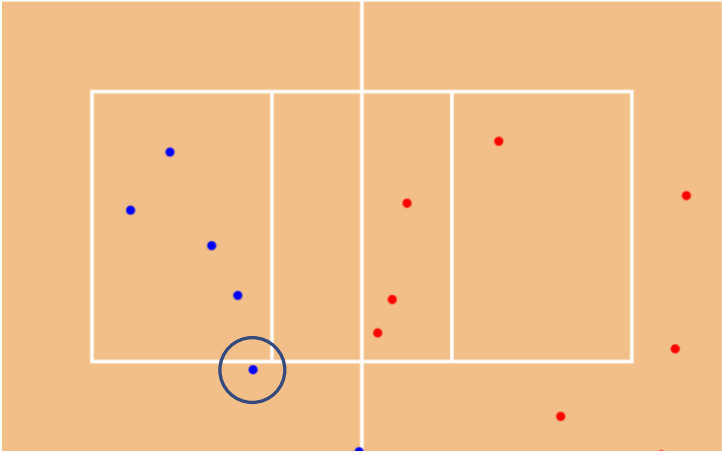
→スパイクのコースチャートが分析で用いられる

体の姿勢も合わせて、打ち方のクセまで分かる可能性

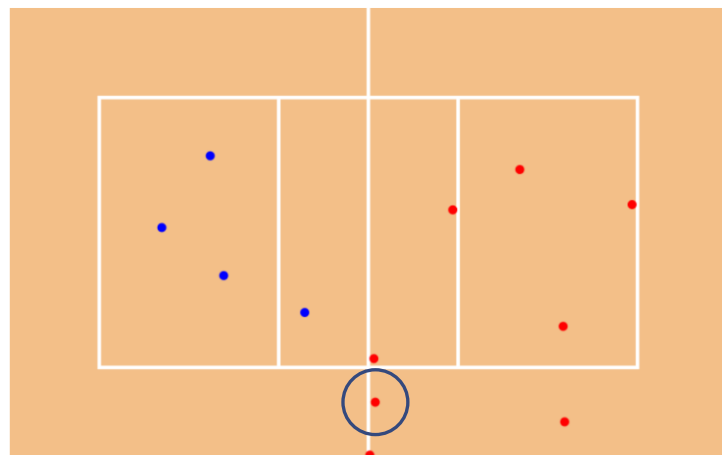


先行研究との推定結果の比較

先行研究

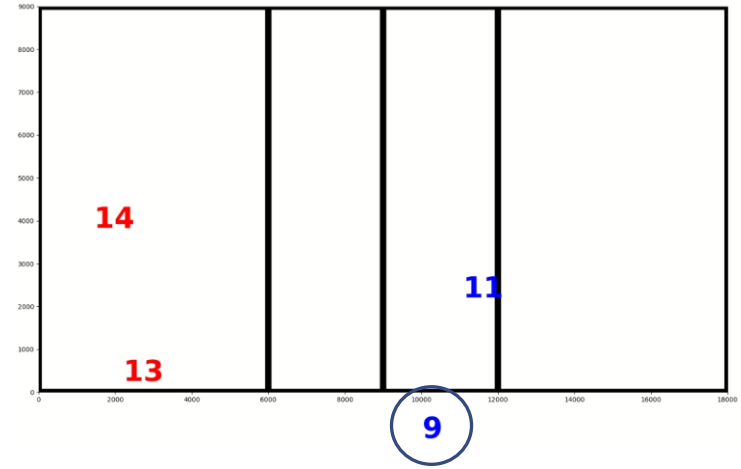


跳躍直前

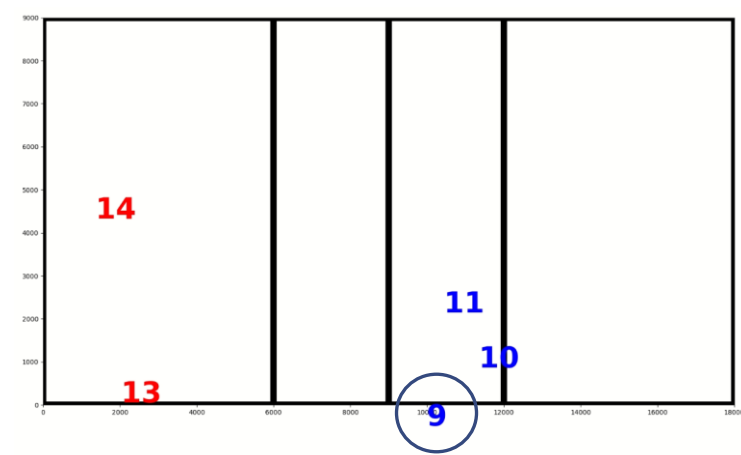


跳躍最高点到達時

本研究



跳躍直前



跳躍最高点到達時

システム処理時間

動作環境

- CPU AMD Ryzen 9 3900X
- GPU NVIDIA GeForce RTX 3090 Ti
- メモリ 32 [GB]

処理時間

①AlphaPose

10秒の動画で30秒の処理時間 → 10FPS
30FPS以上で動作する時間もあり、ばらつき有

②直線間距離の計算 124組の計算で0.0126秒

AlphaPoseの特徴

上海交通大学の研究チームが公開する
人物追跡機能を兼ねた高精度姿勢推定ライブラリ

特徴

①人物を検知した後、単一人物姿勢推定を行う

推定精度 : 82.1 AP (MPII)

比較 : OpenPose

映像内の身体部位とその繋がりを推定

推定精度 : 75.6 AP (MPII)

AP : Average Precision
平均適合率

②人物にIDを振り分けて追跡



選手の交差への対応



選手の交差が起こる例

①選手の消えた位置と再度検知できた位置から判別

→ 消えた選手と新たに検知できた選手の対応が正しく取れなかった

②選手の交差が起きていない映像を使って補完

- ・ カメラの台数を増やす
- ・ カメラの設置位置を工夫する