Predicting Ball Ownership in Basketball from a Monocular View Using Only Player Trajectories

Xinyu Wei[1,2], Long Sha[1,2], Patrick Lucey[1], Peter Carr[1], S. Sridharan[2] and Iain Matthews[1]
[1]Disney Research, Pittsburgh, USA, 15213
[2]Queensland University of Technology, Brisbane, Australia, 4000
{xinyu.wei,long.sha}@connect.qut.edu.au, s.sridharan@qut.edu.au
{patrick.lucey,peter.carr,iainm}@disneyresearch.com

Abstract

Tracking objects like a basketball from a monocular view is challenging due to its small size, potential to move at high velocities as well as the high frequency of occlusion. However, humans with a deep knowledge of a game like basketball can predict with high accuracy the location of the ball even without seeing it due to the location and motion of nearby objects, as well as information of where it was last seen. Learning from tracking data is problematic however, due to the high variance in player locations. In this paper, we show that by simply "permuting" the multi-agent data we obtain a compact role-ordered feature which accurately predict the ball owner. We also show that our formulation can incorporate other information sources such as a vision-based ball detector to improve prediction accuracy.

1. Introduction

The task we focus on in this paper is tracking the ball in basketball from a monocular camera. To detect and track the ball, the intuitive thing to do would be to run an image-based ball detector on every frame and link the detections together. However in practice this approach is problematic as the ball is similar in appearance to human heads causing false alarms, and it is constantly occluded by players. An example of this is shown in Fig. 1 where we show a snapshot from a fixed monocular video camera capturing footage of a basketball match, where the players are clearly visible but the ball is not. However, given a lot of training data it is possible that we would have seen this particular situation before and would have found that it is highly probable that the ball is owned by the point-guard (circled). Instead of using multiple cameras to resolve where the ball is, our approach is to infer the most probable location of the ball given lots of previously seen tracking data.

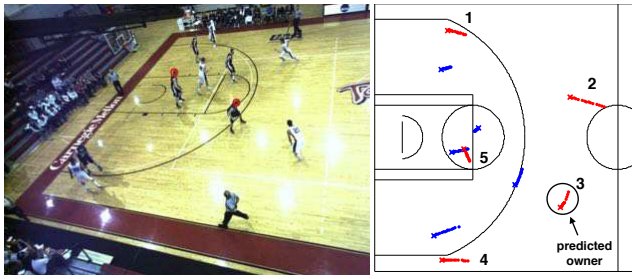
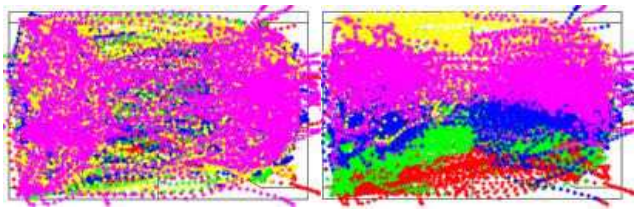


Figure 1. When the ball is not visible (left), output from a ball detector (red circle) is unreliable. In this paper, we predict the ball owner by taking into account other player motion paths.

1.1. Problem Definition

In this paper, our task is to predict which player has the ball in each frame given a monocular camera view of a basketball game. We focus on *ball ownership* - that is predict which player has the ball at each time instant (i.e., frame). We do this as the ball is an inanimate object, which means that its movement relies solely on the actions of intelligent-agents surrounding it which can be predictive of its location. The added benefit of this approach is that the variance of behaviors of an agent is significantly smaller than the object, making learning and predicting behaviors of the object as a function of an agent a more viable task. In group/team settings, the behavior of an intelligent agent is further constrained by the actions/motions of the other intelligent agents. The key problem we tackle in this paper is dealing with the high variance of player tracking data. An example highlighting this issue is shown in Fig. 2(a), where we show the player locations of each player of one team across a half of a game (i.e., 5 players in a team and each color refers to each player). As can be seen in this example, players tend to be in all parts of the court – devoid of any team structure – which we call the "misalignment" of player tracking data. By effectively "aligning" the tracking data to a team-template which enforces team structure, we mini-



(a) (b)

Figure 2. (a) Player locations of a team in basketball ($M = 5$) across a half (T=29432). Each color refers to a player in a particular starting position – players are randomly located over the half. (b) If we align the data, structure of team emerges.

mize the variance of the tracking data which improves the prediction performance (Fig. 2(b)). This alignment essentially tells us which position a player is in relative to his/her teammates in each frame – which we call *player role*. We show how we can learn a team template (i.e., set of roles) directly from data in an unsupervised fashion (Sec. 3). Using this representation we then show by predicting which *player role has the ball and not player identity* we can greatly improve ball ownership prediction performance.

The specific tasks we focus on are predicting which player has the ball given the following information:

1. Full player location information only (Sec. 4),
2. Full player location information + partial (noisy) ball information (Sec. 5), and

In terms of data, we used two sources: a) 36 games of fully annotated player and ball tracking data, and b) an automatically tracked game via a vision system, coupled with annotations (33min) which has both noisy and its cleaned-up counterpart. We conducted tasks i)-ii) to see how predictive player motion was of ball location.

2. Related Work

In terms of tracking and predicting behaviors of multiple agents, an abundance of work has recently focussed on the topic due to the influx of real-world data sources and a myriad of useful applications, most notably in the crowd and security domains [1, 2, 3, 4, 5, 6, 7]. Recent progress in this area has been gained by utilizing contextual features which can greatly reduce the solution space, making prediction tractable [5, 3, 7]. Tracking multiple objects moving in formation has predominantly pertained to rigid formations, such as the approach proposed by Khan and Shah [8]. Recently, Liu and Liu [9] used a mixture of Markov networks to dynamically identify and track lattice and reflection patterns in video. However, the rigid assumption falls down when considering more dynamic scenarios like tracking sports players [7], where the formations tend to be non-rigid (i.e., particles freely move around locally, whilst adhering to the overall global structure). With respect to tracking objects like a ball, typical approaches [10, 11, 12, 13] detect the ball frame by frame then extract the optimum path by linking and smoothing detections. While effective if the ball is observable, these fail when the ball is occluded for a period of time. Recently, Wang et al. [14] proposed a *ball occupancy map* (BOM) to predict the ball owner when the ball is hard to track. BOM is built by accumulating multi-view evidence for the ball in a sparse ground-plane representation. However, such an approach is not applicable for monocular view apporaches. Wang et al. [15] proposed a fully connected graphical model to track interactive objects where one type of object may contain the other. However, inference in this approach is computational expensive.

In terms of minimizing the variance of the tracking data, the task is *given the position information of multiple agents across many frames, permute them to a fixed canonical template*. This is similar to the idea of ensemble image alignment, where the requirement is to align all images to a canonical template [16]. Learned-Miller [17] proposed one of the first methods to do this where he aligned a stack of images which minimized the total entropy. Cox et al., [18] formulated congealing as a least-squares problem, while the RASL algorithm [16] uses rank as a measure of similarity. Other low-rank objectives, such as transformed component analysis [19] or robust parameterized component analysis [20] have also been used. More recently, methods which can deal with multiple modes (or semantically meaningful groups), have been used to simultaneously align and cluster images [21, 22]. The key difference between the work in image alignment compared to multi-agent data is that we want to find the set of permutation matrices rather than a warp, which makes it a non-convex problem. To counter this issue, Lucey et al. [23] recently used hand-crafted templates to form a "role-representation" to align the data to clean up noisy detections. In this paper, we aim to learn the templates directly from data and apply it to object prediction. This approach is similar to one recently proposed by Bialkowski et al. [24, 25].

Our work differs from current approaches as we: i) use the permuted location data of to represent multi-agent behavior to predict the location of an object (i.e., ball), and ii) incorporate image-based object detector with our group representation to improve the prediction.

3. Aligning Multi-Agent Team Tracking Data

Given we have the continuous raw positions of M agents within a team, we can represent the set of observations, $\mathcal{O}$, across T of multi-agent behavior as the matrix of concatenated sequence of 2D points

Figure 3. Shows the drop in entropy of the probability distribution for each agent as we converge to a solution.

$$\mathbf{D}_{F\times M} = \begin{bmatrix} \mathbf{x}_1^1 & \dots & \mathbf{x}_1^M \\ \vdots & & \vdots \\ \mathbf{x}_T^1 & \dots & \mathbf{x}_T^M \end{bmatrix} \tag{1}$$

where $\mathbf{x}_i^j = [x_i^j,\ y_i^j]$ denotes the 2D coordinates of the jth agent at the ith time instance and $\mathbf{X}_i$ is the representation of all M agents for the ith frame. The *first problem* we address is that of representation. In terms of fine-grain analysis, we can use the raw position data which is attractive as we do not have to quantize the input signal (which is lossy), and it provides a low-dimensional representation of the signal. For example in basketball, we can represent a team of five players by their 2D locations which results in a 10 dimensional vector. However, if we plot their locations across T frames, we can see by Fig. 2(a) that the data is the variance is quite large. But if we permute the data at each frame which minimizes the variance (or entropy), we can discover the hidden structure of the data which enables us to perform better prediction. Given that our similarity measure is entropy, $H^m(\mathbf{x}^j) = -\frac{N}{T}\log_2\frac{N}{T}$, where N is the frequency of the jth agent occupying the nth spatial bin, our goal is to find the permutation matrix at each frame $\mathbf{P}_i$ that minimizes the overall entropy of each agent's position. Given we have a reasonable initialization, we can use the EM algorithm [26] to learn a probability distribution template for each agent. The method is summarized in Algorithm 1. We first estimate the set of 2D probability distributions of the M agents, $\mathcal{R} = \{P(\mathbf{x}^1, \dots, P(\mathbf{x}^M)\}$, where $P(\mathbf{x}^m) = \sum_{n=1}^{N} P(\mathbf{x}^m|n)P(n)$ and N is the number of areas of the quantized court. As the court is 94×50 feet, we used an occupancy map of 120×60 as the players are sometimes off the court at times, and we estimated the probability distribution by a normalized count for each bin. We then iterate through each frame by calculating the permutation for each frame which has the lowest entropy. We do this by calculating the change in entropy that assigning each agent to a particular probability distribution. The assignment is then done using the Hungarian algorithm [27] on the basis of minimizing the total entropy. We then permute each frame by the current alignment $\mathbf{X}_t$ and the permutation matrix $\mathbf{P}_t$. We then recalculate the probability distribution, and calculate the change in entropy. We continue this process until the change is below a threshold or the number of maximum iterations is reached. Given training data, we use this approach to learn the probability distribution for a template for each particular role. In Fig. 3 we show how these converge to lower the overall entropy. At test time, given a frame of detections we find the cost matrix between these detections and the set of probability distributions. The Hungarian algorithm is then used to find the permutation matrix at each frame. This gives us the aligned data $\mathbf{D}^*$ which can be described as

$$\mathbf{D^*}_{F\times M} = \begin{bmatrix} \mathbf{P}_1\mathbf{X}_1 \\ \vdots \\ \mathbf{P}_T\mathbf{X}_T \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1^1 & \dots & \mathbf{r}_1^M \\ \vdots & & \vdots \\ \mathbf{r}_T^1 & \dots & \mathbf{r}_T^M \end{bmatrix} \tag{2}$$

where $\mathbf{r}_i^j$ refers to the jth role a player is performing at time i. We use the term *role* to denote the dynamic position an agent has at any time relative to their team-mates instead of an agent maintaining the same feature correspondence which has high variance. Using this method we see distinct group patterns emerge (Fig. 2(b)).

Algorithm 1 EM to Learn Templates

procedure LEARNTEMPLATES($\mathbf{D}$)
  Estimate the initial probability distributions, $\mathcal{R}$
  while $\nabla \text{Entropy}_{\mathbf{D}} <$ threshold **or** iterations $<$ max **do**
    for 1 to T **do**
      Calculate $\text{C}_t(i,j) = -\log P(\mathcal{R}(j) \mid \mathbf{X}_i)$
      Find $\mathbf{P}_t$ using Hungarian algorithm
      Permute current frame $\mathbf{X}_t$ by $\mathbf{P}_t$
    end for
    Update probability distributions, $\mathcal{R}$
    Find change in total entropy, $\nabla \text{Entropy}_{\mathbf{D}}$
  end while
  return $\mathcal{R}^* \leftarrow \mathcal{R}$ ▷ Our final set of templates
end procedure

4. Prediction using Clean Data

Given the clean data source, we assume we know the identity, location and team affiliation for every player at every frame. Additionally, in training we know the current owner of the ball. At test time, our aim is to predict the owner of the ball *solely* from the spatial location and short-term motion patterns of all the players across a window of time. This can translated to the problem of predicting the most likely state sequence $Y = \{y_1, \ldots, y_T\}$, given a set of observations $\mathcal{O} = \{\mathbf{X}_1, \ldots, \mathbf{X}_T\}$ over T frames, where y_t is the state of the ball at time t where $y_t \in \{1, ..., M+1\}$. As the ball is an inanimate object, we assign the state to be the *ball owner*, which can be one of the M players on the court. We have an additional state which corresponds to when the ball is in the air (i.e., shot or pass). Formulating ball tracking as a ball ownership problem was first introduced by Wang et al. [14], but instead of assigning the ball to a player identity, *we assign ball ownership to a particular role*. We formulate the cost of the sequence in terms of a Conditional Random Field

$$\text{loss} = \sum_{t=1}^{T} \Psi_1(y_t; \mathcal{O}, \theta_1) + \sum_{t=1}^{T-1} \Psi_2(y_t, y_{t+1}; \mathcal{O}, \theta_2), \quad (3)$$

where Ψ_1 is the unary potential which measures the compatibility between a label and observations at each frame. Ψ_2 is the pairwise potential which measures the compatibility between two labels and the observations. The set of parameters, θ_1, correspond to $\mathcal{O}$ and the state y, and θ_2 is a set of parameters that correspond to feature $\mathcal{O}$ and edges between y_t and y_{t+1}. In our formulation, both potential functions take the negative log form

$$\Psi_1(y_t; \mathcal{O}, \theta_1) = -\log p(y_t; \mathcal{O}, \theta_1) \; and \quad (4)$$

$$\Psi_2(y_t; \mathcal{O}, \theta_2) = -\log p(y_{t+1} | y_t; \mathcal{O}, \theta_2), \quad (5)$$

The assignment of ball owner can be found by minimizing the loss function with dynamic programming. By modeling group behavior via a CRF, we are able to incorporate spatial prior within the unary term by aligning the data, in addition to team tactics and game context via the pairwise terms.

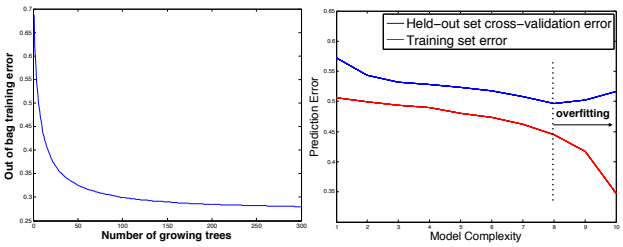
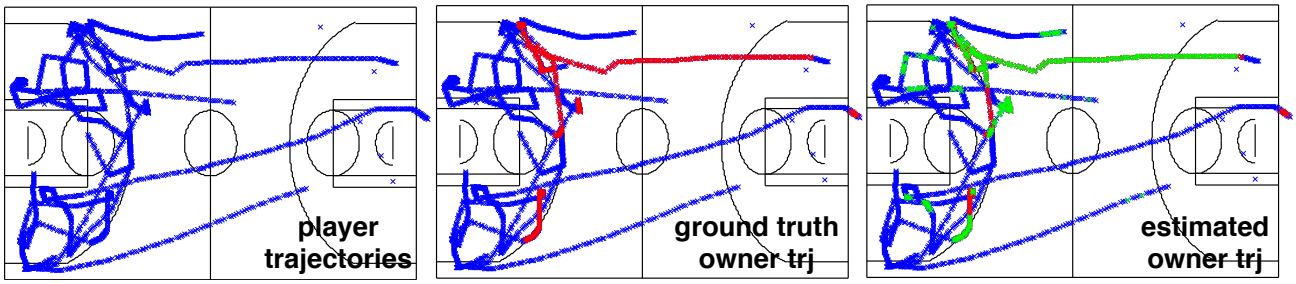


Figure 4. Training and testing error against model complexity.

4.1. Unary Term: Frame-Based Prediction

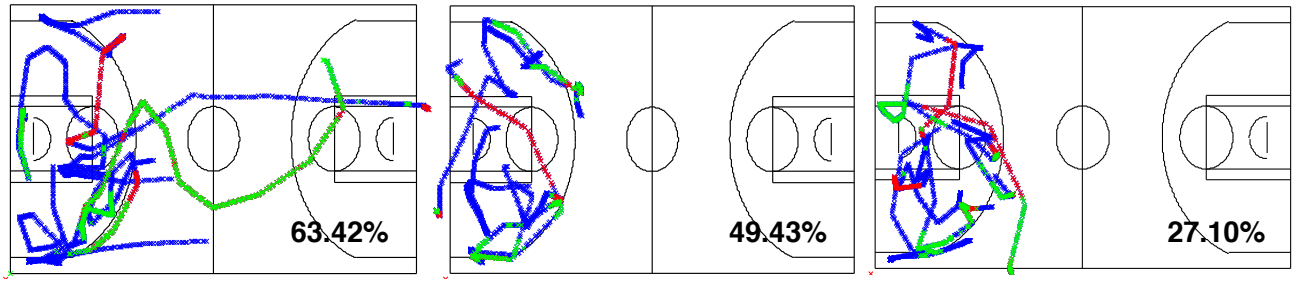
Given observations, we first want to determine how well we can predict the owner of the ball at a given frame. In terms of the CRF, this corresponds to calculating the unary potential $p(y_t; \mathcal{O}, \theta_1)$. This term refers to the probability of an agent in a particular role owning the ball given features and parameters. Due to its simplicity, the representation given in Eqn. 2 is ideal as there is no need to and store specific hand-crafted features. The trade-off though, is that the representation maybe needlessly high-dimensional which can effect the overall prediction. An alternative is to explicitly specify the more relevant features by hand-engineering as set of features (e.g., distance from basket and other players etc.). Another approach which circumvents this issue is to quantize the court into a spatial grid and count the occupancy of players in each grid. As such, we compared the following representations: i) hand-crafted features, ii) occupancy maps, and iii) raw position data (aligned and misaligned).

For the raw position data as well as occupancy maps, not only do we include their spatial positions, but also their deltas to incorporate their short-term motion. Our classifier takes the form of a Random Decision Forest, which is robust against the overfitting that might occur via bootstrapping. It also has good local-feature space adaptivity via randomly splitting the feature space at multiple levels of each tree. We use 70% for the data for training and 30% for testing. To determine the hyper-parameters of the classifier, we further split the training set into k folds for cross validation. Each time $k-1$ folds are used for training and the remaining one is for validation. Fig. 4(left) plots the out of bag error against different number of trees. Fig. 4(right) shows the training error and the validation error with respect to the model complexity (minimum number of observations in the leaf). We set number of trees to 150 and minimum leafs as 30 to avoid overfitting. As our aim is to learn behaviors from a lot of data, we first compared performance using 30 games for training. The quantitive results for the different representations are shown in Table 1. We first compared performance by just using the information about the offensive team. We then incorporated the defensive team into the representation, which further boosted performance. As it



(a) (b) (c)

Figure 5. An example of our frame-based prediction: (a) the player trajectories of the offensive team, (b) ground-truth the ball owner (red curve), (c) Our predicted ball owner (green curve).



(a) (b) (c)

Figure 6. More examples of our frame-based prediction only using player positions. Issues such as prediction flicker and passing and shooting caused the most error.

can be seen, the permuted raw detections yield the best performance with a prediction rate of over 63%. While this rate may not appear to be high, visualizing the prediction shows its impressive performance, which we explain via Fig. 5. In (a), we first show the trajectories for the offensive team in blue, in (b) we have superimposed a red-line on top to depict the ball owner on the relevant trajectory, and in (c) we show our frame-based prediction in green. In Fig. 6, we show three more examples with varying degrees of success. An issue with doing it at the frame-level is that there is constant flicker between the predictions, and the prediction also fails when the ball is in the air. The first issue can be dealt with by incorporating the pairwise term, while the second can be overcome by using an image-based ball detector. As expected, the performance drops when reduce the number of games used for training ($> 50\%$). However, this is very useful, as we can incorporate the vision-based ball detections to boost performance.

4.2. Pairwise Term: Tactics and Context

The pairwise potential, $p(y_{t+1}|y_t; \mathcal{O}, \theta_2)$, measures the transition probability between potential owners at two consecutive frames given observation $\mathcal{O}$ and parameter θ_2. Similar to [14], we factorize this term into p_{tactics} and p_{context}

$$p(y_{t+1}|y_t; \mathcal{O}, \theta_2) = p_{\text{tactics}}(y_{t+1}|y_t; \theta_2) \times p_{\text{context}}(y_{t+1}|y_t; \mathcal{O}, \theta_2) \quad (6)$$

Team Used	Representation	Prediction Rate (%)	
		30 games	1 game
Offense	Hand-crafted	52.3 ± 1.0	44.0 ± 1.6
	Occ map 4×2	53.5 ± 1.6	40.1 ± 2.3
	Occ map 10×6	47.7 ± 2.5	31.5 ± 2.0
Offense	Original Data	32.1 ± 1.2	35.9 ± 3.7
	Permuted Data	58.6 ± 1.5	45.9 ± 2.0
Off & Def	Permuted Data	$\mathbf{63.1 \pm 2.3}$	$\mathbf{50.2 \pm 3.0}$

Table 1. Ball ownership prediction performance with different features and different number of games for training.

where $p_{tactics}$ describes the passing preference between two roles regardless of location (Fig. 7(left)). The other term, $p_{context}$, is the transition probability conditioning on current observation. In our work, $p_{context}$ is conditioned on the distance between roles at two consecutive frames (Fig. 7(right)). We use this term to add penalty into the system if owners between two frames are not close to each other which forces the continuity of owner's trajectory. The term $p_{tactics}$ can be learnt directly from the data, while $p_{context}$ is computed by putting the distance between two roles into a laplacian distribution. We then learn the parameter b of the laplacian distribution from the held-out set. In our experiment, b is set to -5. The contribution of each pairwise term is listed in Table 2, and we can see adding these pairwise terms boosts performance.

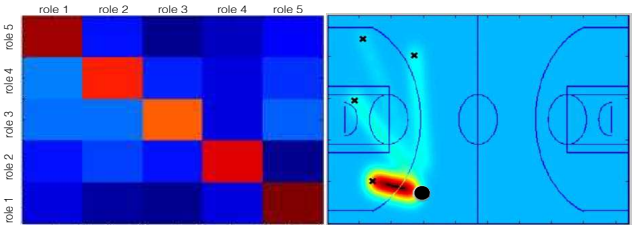


Figure 7. (Left) P_{tatics}: transition probability between roles. (Right) $P_{context}$: transition probability conditioned on the location of the agents.

5. Incorporating Image-Based Ball Detector

In the previous section, while we obtained reasonable performance, poor prediction was experienced when the ball was either passed or shot. This is fortuitous however, as these are the situations where image-based detectors work very well. Incorporating this into our model as an auxiliary information source should boost performance as it reduces the number of predictions we need to make, and thus limits the number of possibilities (see Fig. 8). Given F frames from a monocular video, our system will segment it into two states F_1 and F_2, where F_1 are frames in which the ball is clearly visible (i.e., long passes) and F_2 are frames in which the ball is hard to detect. After detecting the frames which we can reliably detect the ball, we assign labels in all those frames in the CRF as observed and set to *in the air* before decoding the sequence. Since CRFs are undirected model, these revealed labels will also help predicting the owner before and after.

5.1. Estimating Ball Candidates

To estimate possible locations of the ball, we employ a standard ball detection framework which consists of: i) background subtraction using eigen-background segmentation, ii) color filtering, iii) region selection and iv) Hough transform. A visualization of this pipeline is shown in Fig. 9. To test the performance of the ball detector, we randomly extracted 3796 frames of images where the ball is visible. Even though it was visible, there were examples where the frames where partially occluded or had a similar color to the background. We tested two color space which are RGB and HSV, with the RGB working best. The performance is reported in Table 3. Parameters are set loosely since we want to keep the precision high (false alarms can be filtered at a later stage).

Method	Percentage Accuracy	
	30 games	**1 game**
unary only	63.1	50.2
unary + $P_{tactics}$	63.8	51.3
unary + $P_{context} + P_{tactics}$	**66.4**	**56.0**

Table 2. Ownership prediction for unary and pairwise potentials.

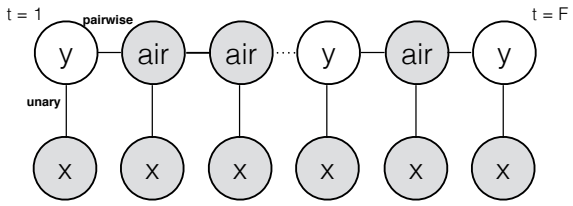


Figure 8. By knowing the frames in which we can accurately locate the ball position via an image-based ball detector, we can limit the number of predictions we need to make.

Detector	Hit Rate	Avg False Alarm
HSV	1802/3796 (49.56%)	4.93/frame
RGB	2435/3796 (64.15%)	2.83/frame

Table 3. Performance of the various color spaces for ball detection.

Method	Percentage Accuracy
Without Ball Evidence	55.98%
With Ball Evidence	71.33%

Table 4. Ball prediction rates with and without ball evidence.

5.2. Segmentation

To segment long trajectories, we fit a 3D projectile model [10] into ball detections across n frames. Depending on how many detections can be fit into the model, the system will decide if a pass or shot is detected. This threshold is set to 10 in our experiments. To test its performance, we annotated 206 long passes in our data set. Each pass has at least 10 frames in the air. Our algorithm is able to detect 157 of them (76.21% hit rate). The performance of the system after adding ball evidences are reported in Table 4. Examples from the fixed cameras are shown in Fig. 10, while Fig. 11, shows an example of the result of our tracking system based on each component.

6. Summary

In this paper, we presented a method to predict the owner of the ball by learning the spatial and motion patterns of multiple agents. Due to the amount of data available, we focussed on basketball to show the utility of this approach. We first show that there is high variance in the tracking data, and that by permuting the data by finding the set of permutation matrices to minimize the total variance/entropy of the data, we can use this as a representation to predict the ball owner at a high rate. Incorporating the prediction problem into a CRF, we show we can include contextual and tactic features which can boost performance. Additionally, as there are instances where image-based detectors work quite well, we incorporate this information source into our model to boost performance.

(a) (b) (c) (d)

Figure 9. Images depicted each stage of our ball detector: (a) input image, (b) output after eigen-background segmentation, (c) output after color filtering, (d) output after region constraints and Hough transform.

(a) (b) (c)

Figure 10. Examples where the ball is visible and occluded (both fully and partial): (a) In the far corner the resolution is low, and the background is of a similar color to the ball, (b) The ball is occluded by the player, (c) A pass is clearly visible.

References

[1] S. Intille and A. Bobick, "A framework for recognizing multi-agent action from visual evidence," in *AAAI*, 1999. 2

[2] R. Li and R. Chellappa, "Group motion segmentation using a spatio-temporal driving force model," in *CVPR*, 2010. 2

[3] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking.," in *CVPR*, 2009. 2

[4] M. Rodriguez, J. Sivic, I. Laptev, and J. Audibert, "Data-Driven Crowd Analysis in Video," in *ICCV*, 2011. 2

[5] K. Kitani, B. Ziebart, A. Bagnell, and M. Herbert, "Activity Forecasting," in *ECCV*, 2012. 2

[6] K. Zhang, L. Zhang, and M. Yang, "Real-time compressive tracking," in *ECCV*, 2012. 2

[7] J. Liu, P. Carr, Y. Liu, and R. Collins, "Tracking sports players with context-conditioned motion models," in *CVPR*, 2013. 2

[8] S. Khan and M. Shah, "Detecting Group Activities using Rigidity of Formation," in *ACM Multimedia*, 2005. 2

[9] J. Liu and Y. Liu, "Multi-target tracking of time-varying spatial patterns," in *CVPR*, 2010. 2

[10] Y. Ohno, J. Miura, and Y. Shirai, "Tracking players and estimation of the 3d position of a ball in soccer games," in *ICPR*, 2000. 2, 6

[11] T. D'Orazio, N. Ancona, G. Cicirelli, and M. Nitti, "A ball detection algorithm for real soccer image sequences," in *ICPR*, 2002. 2

[12] X. Yu, Q. Tian, and K.-W. Wan, "A novel ball detection framework for real soccer video," in *ICME*, 2003. 2

[13] M. Leo, N. Mosca, P. Spagnolo, P. Mazzeo, T. D'Orazio, and A. Distante, "Real-time multi- view analysis of soccer matches for understanding interactions between ball and players," in *CVIU*, 2008. 2

[14] X. Wang, V. Ablavsky, H. B. Shitrit, and P. Fua, "Take your eyes off the ball: Improving ball-tracking by focusing on team play," in *CVIU*, 2013. 2, 4, 5

[15] X. Wang, E. Tureken, F. Fleuret, and P. Fua, "Tracking interacting objects optimally using integer programming," *ECCV*, 2014. 2

[16] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust Alignment by Sparse and Low-

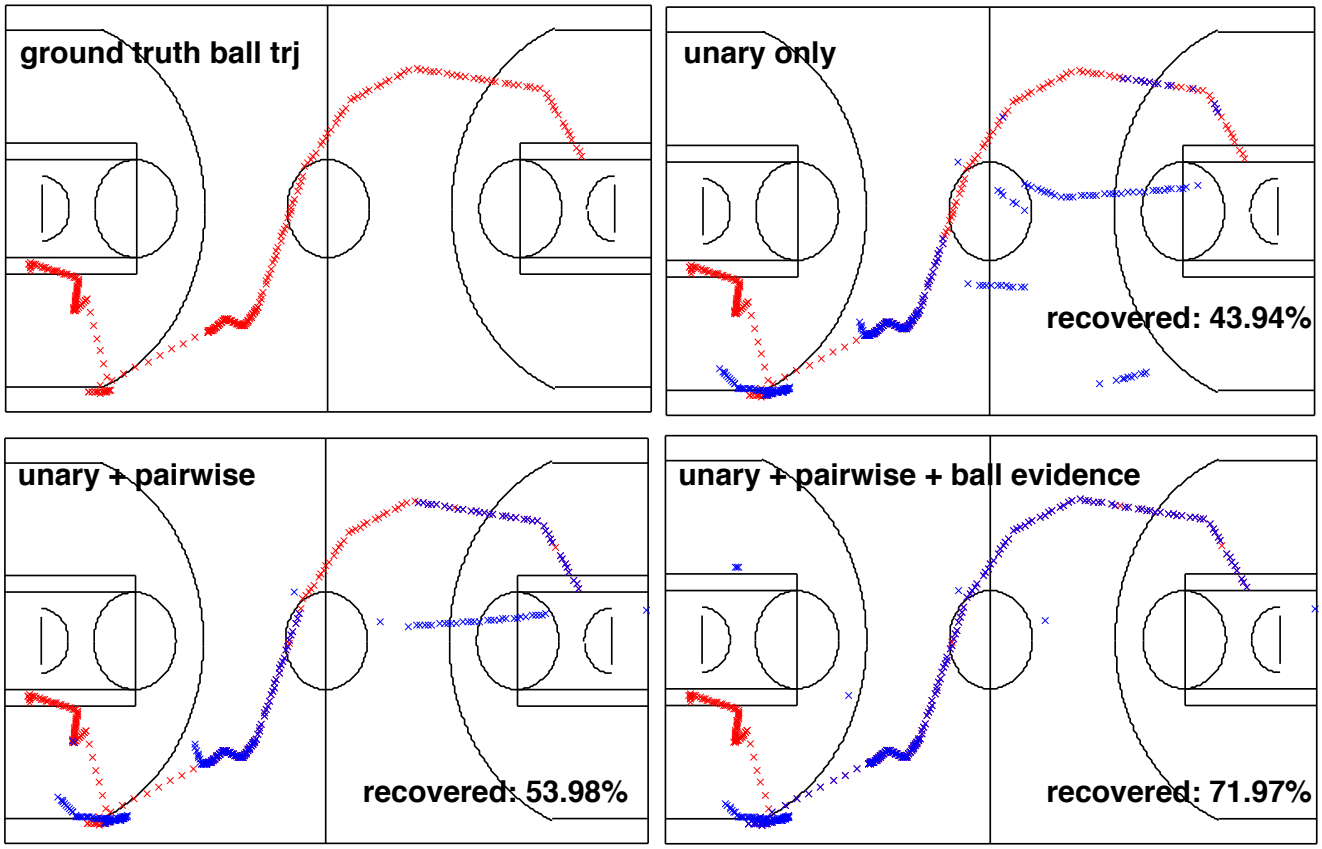


Figure 11. An example of detected ball trajectory at each stage of our system.

Rank Decomposition for Linearly Correlated Images," *PAMI*, vol. 34, no. 11, 2012. 2

[17] E. Learned-Miller, "Data Driven Image Models through Continuous Joint Alignment," *PAMI*, vol. 28, no. 2, 2006. 2

[18] M. Cox, S. Lucey, S. Sridharan, and J. Cohn, "Least Squares Congealing for Unsupervised Alignment of Images," in *CVPR*, 2008. 2

[19] B. Frey and B. Jojic, "Transformation-Invariant Clustering Using the EM Algorithm," *PAMI*, 2003. 2

[20] F. de La Torre and M. Black, "Robust Parameterized Component Analysis," *CVIU*, 2003. 2

[21] X. Liu, Y. Tong, and F. Wheeler, "Simultaneous Alignment and Clustering for an Image Ensemble," in *ICCV*, 2009. 2

[22] M. Mattar, A. Hanson, and E. Learned-Miller, "Unsupervised joint alignment and clustering using bayesian nonparametrics," in *NIPS*, 2012. 2

[23] P. Lucey, A. Bialkowski, P. Carr, S. Morgan, S. Morgan, I. Matthews, and Y. Sheikh, "Representing and discovering adversarial team behaviors using player roles," in *CVPR*, 2013. 2

[24] A. Bialkowski, P. Lucey, P. Carr, Y. Yue, S. Sridharan, and I. Matthews, "Large-scale analysis of soccer matches using spatiotemporal tracking data," in *ICDM*, 2014. 2

[25] A. Bialkowski, P. Lucey, P. Carr, Y. Yue, S. Sridharan, and I. Matthews, "Identifying team style in soccer using formations from spatiotemporal tracking data," in *SSTDM at ICDM*, 2014. 2

[26] A. Dempster, N. Laird, and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society*, 1977. 3

[27] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2(1-2), pp. 83–97, 1955. 3