"Sweet-Spot": Using Spatiotemporal Data to Discover and Predict Shots in Tennis

Xinyu Wei[1], Patrick Lucey[2], Stuart Morgan[3] and Sridha Sridharan[1]
Queensland University of Technology[1], Disney Research Pittsburgh[2], Australian Institute of Sport[3]
Email: xinyu.wei@connect.qut.edu.au, patrick.lucey@disneyresearch.com, stuart.morgan@ausport.gov.au, s.sridharan@qut.edu.au

Abstract

In this paper, we use ball and player tracking data from "Hawk-Eye" to discover unique player styles and predict within-point events. We move beyond current analysis that only incorporates coarse match statistics (i.e. serves, winners, number of shots, volleys) and use spatial and temporal information which better characterizes the tactics and tendencies of each player. Using a probabilistic graphical model, we are able to model player behaviors which enables us to: 1) find the factors such as location and speed of the incoming shot which are most conducive to a player hitting a winner (i.e. "sweet-spot") or cause an error, and 2) do "live in-point" prediction - based on the shots being played during a rally we estimate the probability of the outcome of the next shot (e.g. winner, continuation or error). As player behavior depends on the opponent, we use model adaptation to enhance our prediction. We show the utility of our approach by analyzing the play of Djokovic, Nadal and Federer at the 2012 Australian Tennis Open.

1 Introduction

In Rafael Nadal's recent biography [1], he candidly describes his strategy against Roger Federer as, *"If I have to hit the ball twenty times to Federer's backhand, I'll hit it twenty times, not nineteen. If I have to wait for the rally to stretch to ten shots or twelve or fifteen to bide my chance to hit a winner, I'll wait. There are moments when you have a chance to go for a winning drive, but you have a 70% chance of succeeding; you wait five shots more and your odds will have improved to 85%... That's the plan. It's not a complicated plan. You can't even call it a tactic, it's so simple. I play the shot that's easiest for me and he plays the one that's harder for him - I mean, my left-handed drive against his right-handed backhand. It's just a question of sticking with it..."*

Despite the simplicity of his tactic, what is compelling is Nadal's probabilistic mindset of maximizing the chance of a winner or waiting to force an error. Motivated by this insight, in this paper, we use ball and player tracking information from Hawk-Eye [2] to model a player's behavior based on a myriad of spatiotemporal variables (e.g. shot location, speed, angle, number of shots, feet location). We use a probabilistic graphical model to determine the combination of variables which maximizes a player's chance of hitting a winner from the incoming shot. We call this analysis "sweet-spot", and we also use it find the combination of variables which frequently lead to an error - a method which can be used to highlight a player's tactics and also their strengths and weaknesses.

Our work differs from current tools used to analyze tennis matches such as IBM's Slamtracker [3], which only incorporate coarse match statistics (i.e. serves, winners, number of shots, volleys), and typically lack rich spatial and temporal information. This is despite that fact that ball and player tracking technology has become commonplace at major tennis tournaments to aid in officiating and broadcast visualization. Our approach also allows us to do live "in-point" prediction - where given the previous shots during a rally we can predict the outcome of the current shot. As tennis is an adversarial game, the behavior of a player is heavily conditioned on the behavior of the opponent. To model specific opponent tendencies, we employ a model adaptation technique which greatly improves the predictive power.

2 Methodology

Since Intille and Bobick's seminal work on football play recognition over a decade ago [4], numerous efforts have concentrated on using probabilistic methods to model spatiotemporal data to aid in the analysis and prediction of plays in sport [5-8]. Even though notable, the lack of tracking data to adequately train models has limited its widespread use. Recently however, the release of STATS SportsVU data for basketball has enabled interesting analysis of shots and rebounding [9-10]. For soccer, researchers have characterized team behaviors in the English Premier League using ball-motion information across an entire season using OPTA data [11].

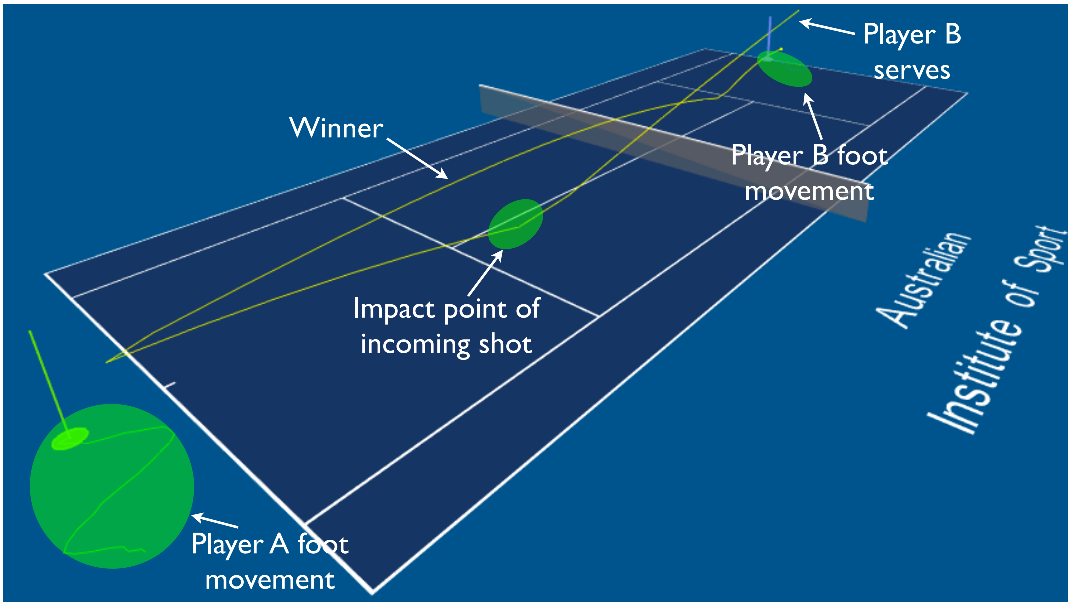

Figure 1. The Hawkeye data contains both the ball trajectory information as well as the player feet movement information. In this example, player B serves the ball to player A who then hits a winner back to player B.

In this paper, we use a Bayesian Network (BN) framework to model player behavior using ball and player tracking information. BN's are a type of probabilistic model that allow us to construct a single compact model that captures the varying properties of a rally. To facilitate this work, we used Hawk-Eye [2] data from the Men's draw of the 2012 Australian Open. The Hawk-Eye data contains ball trajectories and player foot position over time. An example of the data is illustrated in Figure 1. This rich set of data allows us to see which variables (or combination of variables) increase the likelihood a player will hit a winner or commit an error[1] - which in essence suggests a player's strengths and weaknesses. The genesis of a player's behavior can be described by a number of underlying variables, such as: shot speed, shot location, player position, opposition position, number of shots in rally and identity or rank of opposition (e.g. Federer, or top 10 rank). Except for the identity of the opponent, all of these factors or variables vary temporally which makes this problem an ideal candidate for a BN. Additional factors such as: set number/length of match, environment conditions (e.g. hot, humid, cold or windy), specific match context (e.g. game/set/match/break point), court surface (e.g. grass, hard-court, clay) would no doubt enhance the predictive power of our model, however, as we increase the number of variables the demands on the amount of training data required to train our BN exponentially increases. For this work, we specifically modeled the winners and errors associated with the top 3 seeds at the tournament (Novak Djokovic, Rafael Nadal and Roger Federer) as they had the most data and it also allowed us compare the different styles of play. The respective opponents and outcomes of the matches used in this study are shown in Table 1.

Table 1. Shows the opponent and outcome for the Djokovic, Nadal and Federer at the 2012 Australian Open.

Player	Stat	Round						
		1st	2nd	3rd	4th	Qtr	Semi	Final
Djokovic	Opp	Lorenzi	Giraldo	Mahut	Hewitt	Ferrer	Murray	Nadal
	Score	6-2,6-0,6-0	6-3,6-2,6-1	6-0,6-2,6-1	6-1,6-3,4-6,6-3	6-4,7-6,6-1	6-3,3-6,6-7,6-1,7-5	5-7,6-4,6-2,6-7,7-5
Nadal	Opp	Kuznetsov	Haas	Lacko	Lopez	Berdych	Federer	Djokovic
	Score	6-4,6-1,6-1	6-4,6-3,6-4	6-2,6-4,6-2	6-4,6-4,6-2	6-7,7-6,6-4,6-3	6-7,6-2,7-6,6-4	7-5,4-6,2-6,7-6,5-7
Federer	Opp	Kudryavtsev	Beck	Karlovic	Tomic	Del Potro	Nadal	-
	Score	7-5,6-2,6-2	walkover	7-6,7-5,6-3	6-4,6-2,6-2	6-4,6-3,6-2	7-6,2-6,6-7,4-6	-

[1] Due to the ambiguity in differentiating between forced and unforced errors from the data directly, we label both these errors into a single category. We found due to the different characteristics of these errors - two discriminant modes for each error type resulted.

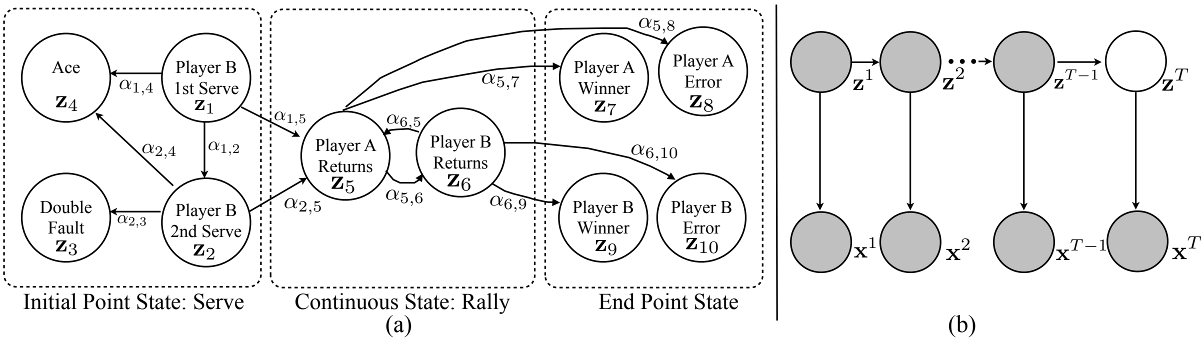

Figure 2. (a) The model of a point in tennis - given player B serves the ball and it is not an ace or double fault, player A returns the ball back to player B and the rally ensues until player A or B hits a winner or an error. (b) The probability of the i^{th} point-state at time T, $\mathbf{z}^T_i$, given the current observation $\mathbf{x}^T$ and previous state $\mathbf{z}^{T\text{-}1}$, can be calculated using the sum and product rule (gray nodes are observed and clear nodes are hidden). The state with the highest probability is our in-point prediction.

3 Modeling Player Behavior

To model player behavior we first have to construct an accurate model of how a point is played. At a coarse level, a point consists of three game states: 1) the initial state (i.e. the serve), 2) the middle state (i.e. the rally which can consist of many shots), and 3) the end state. Given that there was no ace or double fault, the end-point state consists of either player A or player B hitting a winner or an error. A depiction of our point model is given in Figure 2(a), which shows that we have six possible point-states after the serve $\{z_5,\ldots,z_{10}\}$ and the transition probabilities between different states is given by $a_{i,j}$. Given that our observation or feature vector $\mathbf{x}^T$ contains information about the incoming shot (i.e. location, speed, angle, player's feet position, number of shots in rally etc.), and we know the previous state $\mathbf{z}^{T\text{-}1}$ (i.e. player A or player B returned the ball), using our model topology shown in Figure 2(b), we can infer the probability of next state $\mathbf{z}^T$ being a returning shot, winner or error by using Bayes' law as . This expression describes a dynamic Bayesian Network (DBN), where the next state is conditioned on the previous state, but seeing that there is only one possible state for the returning shot and we observe this (otherwise the point is over), we can simplify this into a Bayesian Network (BN), which yields:

$$P(\mathbf{z}^T|\mathbf{x}^T) = \frac{P(\mathbf{x}^T|\mathbf{z}^T)P(\mathbf{z}^T)}{P(\mathbf{x}^T)} \tag{1}$$

Depending on the player returning the shot, we infer the probability of $\{\mathbf{z}^T_5, \mathbf{z}^T_7, \mathbf{z}^T_8\}$for player A and $\{\mathbf{z}^T_6, \mathbf{z}^T_9, \mathbf{z}^T_{10}\}$ for player B - and our prediction is the state, $\mathbf{z}^T_i$, with the highest probability. To learn a model for each player, we first created a shot database where each shot was labeled as either an ace, fault, continuing shot, ground stroke winner or ground stroke error. Given these labeled shots, we then learnt probability distribution functions (pdf's) for each variable (e.g. shot location, speed, angle, position, feet position, number of shots), which yielded a multi-dimensional pdf for each shot type for each player. To obtain a continuous distribution for each variable x, we represented the probabilities as a Gaussian Mixture Model (GMM), where given that the GMM has the form:

$$G(x;\mu_k\Sigma_k) = \frac{1}{2\pi^{\frac{d}{2}}|\Sigma_k|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \tag{2}$$

where μ is the mean, $\sum$ is the covariance and $\boldsymbol{\theta} = \{(w_1;\, \mu_1,\, \sum_1),\, \ldots\, ,\, (w_M;\, \mu_M,\, \sum_M)\}$are the parameters of the GMM for M mixtures. The parameters of $\boldsymbol{\theta}$ are learnt using the Expectation Maximization (EM) algorithm. It is important to note these parameters as they allow us to adapt to specific opponent behavior, which we describe in Section 5.

In this paper, we are only concerned with shots within the rally (i.e. all non-serve behavior). The pdf's for the incoming shot preceding winners are shown in Figure 3. Before, we analyze these plots it is wise to revisit Figure 1 to get a better understanding of what the different variables mean. Given the opponent has hit the ball to the player of interest - which we call the incoming shot - the shot impact location refers to the (x, y) location of where the ball lands prior to the player hitting the ball. The speed and angle of the incoming shot are also quantified. The feet

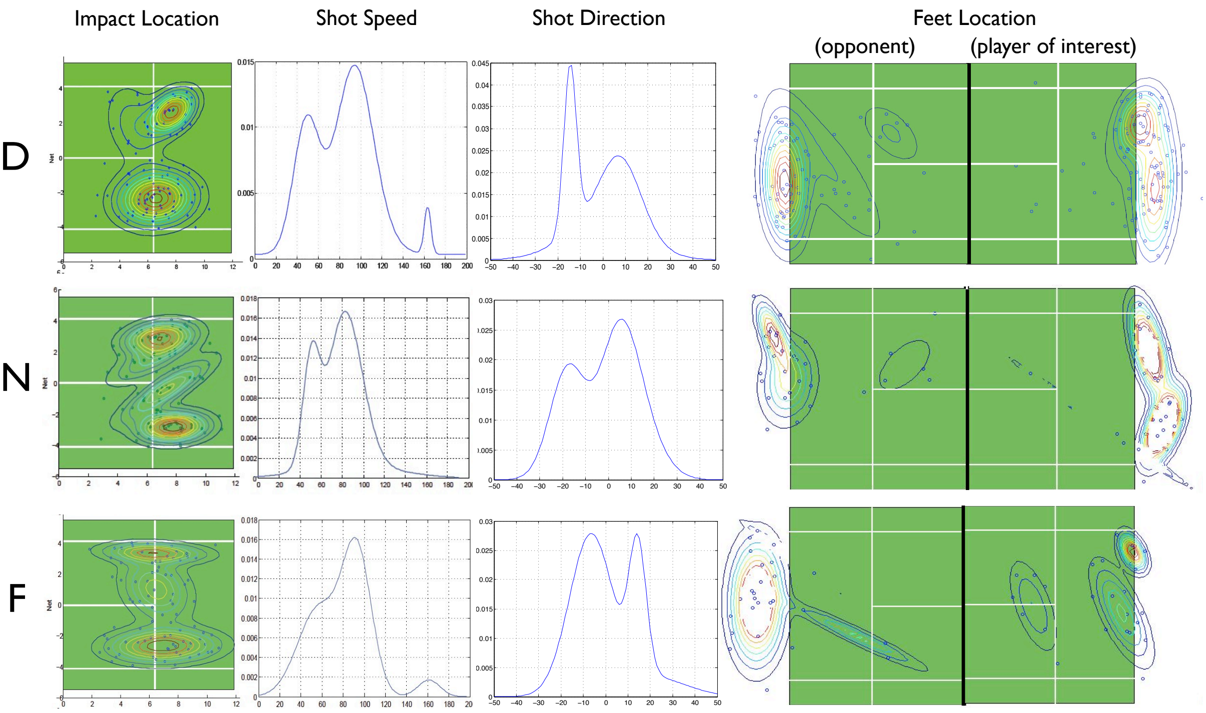

Figure 3. The probability distribution functions (pdf's) of winners for Djokovic (D), Nadal (N) and Federer (F) in the 2012 Australian Open with respect to various variables.

location refers to the respective location of each player at the moment the incoming shot impacts the court. In this work we did not differentiate between ground-strokes, volleys and smashes as the multiple modes within the GMM encodes this information as these strokes have different speed and impact location profiles (this will be obvious after the next couple of sentences). In terms of unique player characteristics, these plots make for some interesting analysis. First of all, it can be seen that Federer tends to hit his winners from balls that land closer to boundary widths of the court compared to Djokovic and Nadal, which may allow him to generate more angle on his winning strokes. It is also evident that Federer tends to stroke more winners from volleys than Djokovic and Nadal, which can be inferred from the his foot location on striking the winners. Furthermore, in comparison to Djokovic and Nadal, Federer frequently hits winners from further inside the baseline while his opponents are pressed significantly deeper behind their own baseline. Many of Nadal's winners are characterized by his opponents playing from their left side (the backhand side for right-handed players), which may indicate his preference for backhand rallies. In contrast, Djokovic tends to stroke winners while his opponent is positioned more to the right-handed forehand (albeit more subtly). The shot speed profiles also indicate that Federer and Djokovic achieve a number of winners via over-head smashes, which may be inferred by the small peaks at the upper end of the speed spectrum. The combination of variables in a rally that are most likely to lead to a winner for any specific player, the "sweet-spot", can be visualized such as the examples seen in Figure 4. In terms of general trends, when we compare the winners (left) to the errors (right), we can see that the impact location of the incoming shot is much deeper as well as significantly more quicker.

Figure 4. "Sweet-Spot" - these visualizations show the incoming shot that gives the highest probability of (left) hitting a winner, and (right) causing an error - these shots are hit deeper and with more pace.

4 In-Point Prediction

In order to validate our player models we conducted a series of experiments to measure the accuracy of next-stroke predictions at any point in a rally - which we call "in-point prediction". As we are only interested in the non-service behavior of a player (i.e. no ace or double fault), "in-point prediction" refers to predicting whether the next shot is either: i) a winner, ii) an error, or iii) a return (i.e. continuation of the point). For our experiments, we generated player models for Djokovic, Nadal and Federer and tested these models on two matches, Nadal vs Federer (semi-final) and Djokovic vs Nadal (final). To make sure there was no overlap between training and testing data, for the Federer and Djokovic models, we trained the models on all their matches except those against Nadal. Similarly for Nadal, we created two models: the first we tested against Djokovic and trained on all the other matches he played in, the second we tested against Federer and again trained it all other matches. As there are many more continuing shots than winners or errors, the overall agreement between correctly classified shots can skew the results. To counter this we used the receiver-operator characteristic (ROC) curve, which plots the hit-rate against the false positives. From these curves, we used the area underneath the ROC curve (AUC) to assess performance. The AUC ranges from 0.5 (pure chance) to 1.0 (ideal classification). In Table 2(a), we show the aggregate shot prediction performance for winners and errors.

As can be seen from the results, the impact location of the incoming shot was the best single predictor of a winner, while the feet location was the second best predictor. When combining factors together, speed + impact location and feet location gave the best, which makes sense after our analysis in the previous section. Even though the performance improves, the overall predictive power of winners is still quite poor but the lack of opponent modeling can explain this (we look at this in the next section). It is also interesting to note that the number of shots in the rally diminishes the performance. As the variance of this variable is very high, coupled with the fact that we have relatively fewer examples that errors, it is probable that we have severely under trained this variable which can explain the noisy results. For errors, a similar trend is observed with the best predictor gaining a AUC of 76.09% which is much higher than the winner rate of 68.52%.

5 Opponent Modeling

As the models used in the previous section do not model specific opponent behavior, this represents an obvious area of improvement as the behavior or tactics of a player are heavily dependent on the opponent and the court surface (e.g. Nadal's behavior in a match against another "base-liner" such as Djokovic on a clay-court is likely to be a poor predictor of his behavior against "serve-and-volleyer" Federer on a grass court). Obviously, the best model of future performance is going to be one that is trained on data which has the same conditions (i.e. same opponent, court-surface etc.). However, this is problematic as obtaining enough data to adequately train a model is extremely

Table 2. (a) Performance of our player models for predicting winners and errors using a "one-versus-everyone-else" or UBM model. (b) Adapting the player models to specific opponents yields better performance (N.B. there was no feet location data in our adaptation data - hence the absence of those results.)

(a) Universal Background Model	Prediction Accuracy (AUC)	
Shot Variable	**Winner**	**Errors**
Speed	52.49	61.63
Angle	54.13	54.76
# Shots in Rally	55.51	59.61
Feet Location	61.27	60.94
Impact Location	65.29	59.76
Speed +Angle	55.51	58.75
Speed + Impact Location	63.62	61.12
Speed + Impact Loc + # Shots	60.04	60.39
Speed + Impact Loc + Feet Loc	**68.52**	**76.09**
Speed + Impact Loc + Feet Loc + # Shots	57.83	68.60

(b) Adapted Model	Prediction Accuracy (AUC)	
Shot Variable	**Winner**	**Errors**
Speed	56.45	62.65
Angle	60.36	54.47
# Shots in Rally	60.30	68.21
Impact Location	70.30	63.29
Speed +Angle	58.67	57.74
Speed + Impact Loc	**77.28**	**71.85**
Speed + Impact Loc + # Shots	65.41	70.94

difficult as players may only play each other a couple of times a year and this is often on different surfaces. A method to resolve this issue is to employ adaptive model techniques, such as are commonly used in speech and speaker verification tasks [13]. Unlike the standard approach of maximum-likelihood training of a model, adaptive models "adapt" the parameters of an initial model or Universal Background Model (UBM) to held-out data which is indicative of the test data. To do this, our held-out data consisted of two matches from previous tournaments which were the nearest examples to the Australian Open conditions in our library: 1) Djokovic vs Nadal at a previous hard-court tournament, and 2) Federer vs Nadal at a previous grass court tournament. Using this adaptation technique (see Appendix A for full description of technique), we were able to greatly improve the prediction performance for winners from 68.52% to 77.28% - as can be seen in Table 2(b). As the held-out data did not contain feet location of the players, we were constrained to only using the ball trajectory information which can explain the diminished performance for predicting errors. Again, the number of shots in the rally had a similar impact, which suggests that this variable is undertrained. When adapting the player models, we found that the court surface had a significant impact on performance. For example, when we adapted our Nadal vs Djokovic model to a match these played on a clay surface the performance greatly dropped which suggest that the court surface as a large impact on how a match is played (which is a commonly held belief).

From these experiments it can be seen that reasonable "in-point prediction" can be obtained from using spatiotemporal data. An obvious application of this work can be seen within a broadcast environment, where this type of approach can be used for real-time (or close to real-time) analysis. The idea is, given we have the trajectory information about the incoming shot - we can not only give a prediction of the most likely outcome of the shot but we can also give a prediction of the position where we think the shot will go. In terms of in-depth analysis, we can determine whether the shot played was within the expected distribution of the player or fell outside. If it fell outside our expectation this "anomalous" behavior could then trigger some further analysis of the shot.

6 Summary and Future Work

Rich sources of player and ball tracking data are emerging in sports such as tennis, which are well suited to novel analyses using probabilistic graphic models. In this paper, we employed a Bayesian framework to build a stroke-by-stroke model that is predictive of the outcome of individual points in top level tennis matches. This analysis creates novel insights to the playing styles of individual players, and in particular, we identify the "sweet-spot" for three of the top male tennis players in the world. Our modeling approach demonstrates superior performance using adaptive techniques, which allow greater sensitivity by tuning the model to specific match parameters such as court surface. These results are insightful for coaches hoping to discover critical points of strength and weakness in opponents. Furthermore, the dynamic and intuitive nature of the analysis has excellent potential to enhance the in-game viewer experience for spectators.

As noted by Nadal, tennis may be thought of as a very fast moving chess match, where players systematically maneuver their opponent into a position of weakness before attempting to win the point. Therefore, it is important in future work to extend the scope of model parameters to include deeper analysis of the sequence of strokes that ultimately lead to a winning shot. Analysis of that type requires exponentially larger data sets, however, the popularity of technologies such as Hawk-Eye means that "big data" modes of analysis will become more feasible.

7 References

[1] R. Nadal and J. Carlin, "Rafa", Hyperion, 2011.
[2] Hawkeye Innovations. http://www.hawkeyeinnovations.co.uk.
[3] IBM SlamTracker, 2012. www.australianopen.com/en_AU/ibmrealtime/index.html.
[4] S. Intille and A. Bobick, "A Framework for Recognizing Multi-Agent Action from Visual Evidence", in AAAI, 1999.
[5] R. Li, R. Chellappa and S. Zhou, "Learning Multi-Modal Densities on Discriminative Temporal Interaction Manifold for Group Activity", in CVPR, 2009.
[6] V. Morariu and L. Davis, "Multi-Agent Event Recognition in Structured Scenarios", in CVPR, 2011.
[7] D. Stracuzzi, A. Fern, K. Ali, R. Hess, J. Pinto, N. Li, T. Konik and D. Shapiro, "An Application of Transfer to American Football: From Observation of Raw Video to Control in a Simulated Environment", AI Magazine, vol. 32, no. 2, 2011.
[8] M. Perse, M. Kristan, S. Kovacic and J. Pers, "A Trajectory-Based Analysis of Coordinated Team Activity in Basketball Games", Computer Vision and Image Understanding, 2008.

[9] R. Maheswaran, Y. Chang, A. Henehan and S. Danesis, "Deconstruction the Rebound with Optical Tracking Data", in MIT Sloan Sports Analytics Conference, 2012.
[10]. K. Goldsberry, "CourtVision: New Visual and Spatial Analytics for the NBA", in MIT Sloan Sports Analytics Conference, 2012.
[11]. P. Lucey, A. Bialkowski, P. Carr, E. Foote and I. Matthews, "Characterizing Multi-Agent Team Behavior from Partial Team Tracings: Evidence from the English Premier League", in AAAI, 2012.
[12] K. Murphy, "The Bayes Net Toolbox for MATLAB", Computing Science and Statistics, vol. 33, 2001.
[13] D. Reynolds, T. Quatieri and R. Dunn, "Speaker Verification using Adapted Gaussian mixture models", Digital Signal Processing, 2000.

A Appendix A - Model Adaptation

Given that we want to train a specific model between two players, but do not have enough to adequately train the model, we can use a method which is widely used in speech and speaker recognition communities called "model adaptation" [13]. The core idea behind model adaptation is to "adapt" the parameters of an initial model or Universal Background Model (UBM) to a held-out or "adaptation" dataset which should be indicative of want we can expect to see in the test data. So given the initial model parameters or UBM parameters, $\boldsymbol{\theta}_{\text{UBM}} = \{(w_1; \boldsymbol{\mu}_1, \sum_1), \ldots, (w_M; \boldsymbol{\mu}_M, \sum_M)\}$, we can find the parameters of the held-out matches $\boldsymbol{\theta}_{\text{Adapt}} = \{(w_1; \boldsymbol{\mu}_1, \sum_1), \ldots, (w_M; \boldsymbol{\mu}_M, \sum_M)\}$. We then update the parameters of the UBM by using the following equations:

$$w_k^* = \frac{a_k^w n_k}{n} + (1 - a_k^w) w_k, \quad \mu_k^* = a_k^m E_k(x) + (1 - a_k^m)\mu_k, \quad c_k^* = a_k^v E_k(x^2) + (1 - a_k^v)(c_k + \mu_k^2) - \mu_k^{*2} \tag{3}$$

where $\{w_k^*, \mu_k^*, \Sigma_k^*\}$ are the new parameters of the adapted model, and $a_k^p, p \in \{w, m, v\}$ are used to control the balance between old and new estimates for weights, means and covariances. This is a form of regularization, which guarantees improved generalization.