ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/261163495

Predicting Shot Locations in Tennis Using Spatiotemporal Data

Conference Paper · November 2013

DOI: 10.1109/DICTA.2013.6691516

CITATIONS
25

READS
5,569

4 authors, including:

Stuart Morgan
Australian Institute of Sport (AIS)

24 PUBLICATIONS **699** CITATIONS

SEE PROFILE

All content following this page was uploaded by Stuart Morgan on 19 August 2015.

The user has requested enhancement of the downloaded file.

Predicting Shot Locations in Tennis using Spatiotemporal Data

Xinyu Wei*, Patrick Lucey†, Stuart Morgan‡, Sridha Sridharan*
*SAIVT Lab, Queensland University of Technology, Australia
Email: xinyu.wei@connect.qut.edu.au
†Disney Research Pittsburgh, USA
Email: patrick.lucey@disneyresearch.com
‡Australia Institute of Sport, Australia,
Email: stuart.morgan@ausport.gov.au

***Abstract*—Over the past decade, vision-based tracking systems have been successfully deployed in professional sports such as tennis and cricket for enhanced broadcast visualizations as well as aiding umpiring decisions. Despite the high-level of accuracy of the tracking systems and the sheer volume of spatiotemporal data they generate, the use of this high quality data for quantitative player performance and prediction has been lacking. In this paper, we present a method which predicts the location of a future shot based on the spatiotemporal parameters of the incoming shots (i.e. shot speed, location, angle and feet location) from such a vision system. Having the ability to accurately predict future short-term events has enormous implications in the area of automatic sports broadcasting in addition to coaching and commentary domains. Using Hawk-Eye data from the 2012 Australian Open Men's draw, we utilize a Dynamic Bayesian Network to model player behaviors and use an online model adaptation method to match the player's behavior to enhance shot predictability. To show the utility of our approach, we analyze the shot predictability of the top 3 players seeds in the tournament (Djokovic, Federer and Nadal) as they played the most amounts of games.**

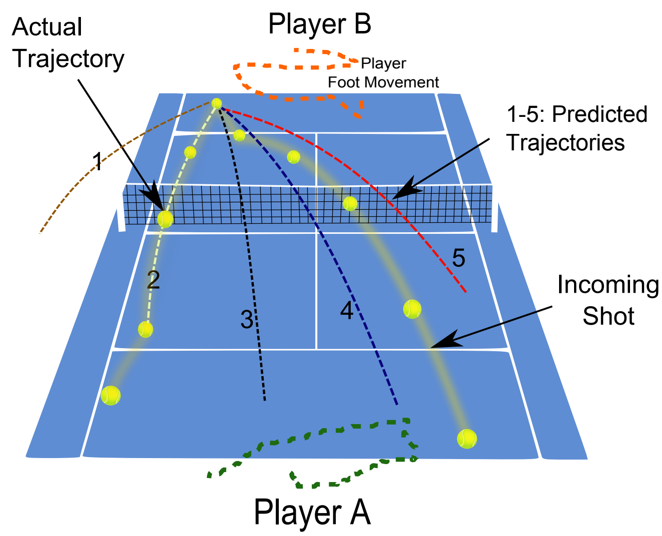

Fig. 1. The Hawk-Eye data contains both ball trajectory and player feet movement information. In this example, Player A (bottom) serves the ball to Player B (top) who then will try to return the ball back to Player A. We are interested in accurately predicting the location of this next shot based on the incoming shot (Dashed lines indicate possible trajectories of next shot).

I. Introduction

In professional sports, top players have a remarkable ability to quickly and accurately predict the behavior of their opponent to gain an advantage. For example in tennis, given the speed, location, angle of the shot with respect to the relative position and movement of the player's involved - a top player will anticipate where the next shot will come based on these previous shot factors. Similarly, a top camera operator and director will have a similar intuition of how the play will evolve to obtain the best shot to show viewers. In both cases, players and broadcasters have developed perceptual expertise and pattern recognition skills that are akin to a "biological probabilistic engine".

In this paper, we aim to emulate these perceptual skills by generating a probabilistic model which can model and predict the behavior of a player in tennis by using an entire tournament of spatiotemporal data from Hawk-Eye [7] of the 2012 Australian Open Men's Tournament. Over the past decade, vision-based systems such as Hawk-Eye have provided ball tracking for enhanced broadcast visualizations and due to their accuracy have also been used for aiding umpiring decision in both tennis and cricket. Despite the high-level of accuracy of the tracking systems and the sheer volume of spatiotemporal data they generate, the use of these systems for quantitative player performance and prediction has been lacking. For example in tennis, tactical and strategic analysis does exist (e.g. *IBM's Slamtracker* [10]), but use only superficial quantities (e.g. winners, aces, volleys, forced-errors, etc) and do generally not include spatiotemporal information.

The goal of this paper is to accurately predict the location of where the *next shot* will go, given the information of the previous shot (see Figure 1). This task is much more challenging than the previous work of Wei et.al. [28], where they predicted "what" type of shot (i.e. winner, error or continuation) but not "where" which is a potentially infinitely larger output state space. Having the ability to predict the location of future short-term events has enormous potential in area of automatic broadcasting. In sports like tennis, intelligent camera systems that can anticipate the play could augment current human operator broadcasting. Reliable predictive intelligence in tennis and other sports also has considerable potential value in high performance sport coaching.

To emulate the *game intelligence* of a player, we use a Dynamic Bayesian Network (DBN) to model and predict their short-term behavior. As competitive sports such as tennis are adversarial, the behavior of a player will be conditioned on the opponent's behavior as well as the environment (e.g. court surface (grass vs hard-court vs clay), temperature, etc). To account for such variation, we employ an adaptive technique which we use in an online fashion to improve our predictive capacity. Using this method, we predict the location of the output shot given the features of the incoming shot. Our specific contributions of this work are: i) we show that a combination of shot factors, especially the relative short-term player positions and velocities are the most predictive of future shot location; ii) to gain a specific x, y shot location, we augment the output state-space of our DBN to include local regions to enrich our probability estimate; and iii) we developed an online model adaptation method which incorporates adversarial player behavior as well as court conditions. To demonstrate the utility of our approach, we compared and predicted the behavior of three of the top players at the 2012 Australian Tennis Open (Djokovic, Nadal and Federer).

II. Related Work

A recent slue of work has looked at the use of vision-based systems for the early-prediction of events. Hoai and de la Torre [9] used a maximum-margin classifier to predict the onset of facial expressions. In [25], a dynamic bag-of-words approach to predicting human activities was used. Kitani et.al [14] proposed a method to forecast the path of a pedestrian using the semantic knowledge of the scene (such as obstacles)to forecast the most likely trajectory path using a hidden variable Markov decision process. Ziebert et.al [33] employed a similar strategy, where they forecasted pedestrian trajectories using a soft-max version of goal-based planning. In terms of larger scale tasks such as crowd analysis, Rodriguez et.al [24] track individuals in a crowd by first learning behaviors priors on a large database of crowd videos using a correlation topic model.

These works are similar in spirit to the myriad of research conducted in the area of multi-agent tracking. Recently, good performance has been gained by posing the task as a data association problem. Notably, Collins [5] developed an iterative approximate solution to the multidimensional assignment problem using high-order motion models. Similar efforts using a network flow algorithm based on K-shortest paths, and linear/dynamic programming have also be used [2], [22], [31]. Choi and Savarse [4] proposed a multi-layer framework to jointly track multiple people, recognizing individual activities and interactions between pairs of people using contextual information between all these modes of information.

In terms of sport-related research, there is a considerable amount of research which has focussed on predicting activities and events. The seminal work in this area was conducted by Intille and Bobick [11] over a decade ago, where they used a Dynamic Bayesian Network to predict a football play from manually annotated player trajectories. Since then, multiple approaches have centered on recognizing football plays [16], [17], [26], but each example only considers a relatively small number of plays (approximately 50-100). Zhu et.al [32] analyzed tactics in soccer matches by building multiple trajectories using analysis of spatiotemporal interactions. Hervieu et al. [8] also used player trajectories to recognize low-level team activities using a hierarchical parallel semi-Markov model. Kim et.al [13] used motion fields to predict where play will evolve in the short-term. Lucey et al. [18], [19] used ball-tracking data to discriminate team's playing style in soccer. In basketball, Masheswaran et al. [20] use a data-driven approach to predict the location of rebounds given the incoming shot. Perse et al. [21] used trajectories of player movement to recognize three types of team offensive patterns. More recently, Bialkowski et al. [3] recognized activities from noisy player detections. Atmosukarto et al. [1] were able to detect the line of scrimmage for plays in American Football, and the type of player formation the offensive team takes on. Wang et al. [27] addressed the problem of ball tracking in team sports by formulating the tracking in terms of deciding which player, if any, owns the ball at any given time and Wei et.al [28] predicted the type of shot (e.g. winner, error or continuation).

As environmental conditions between points and matches may change considerably for various reasons (e.g. time of day, weather, event , etc), recent work has focussed on adapting models according to the transient conditions. Yamada et.al [29] proposed an unsupervised domain adaptation method to counter issues of dataset bias in 3D pose estimation. Similarly, Jain and Miller [12] presented an online approach for rapidly adapting the output of a black-box classifier using a Gaussian process regression model. Kulis et.al [15] introduced a domain adaptation technique based on learning an asymmetric non-linear cross-domain transformation that map points from one domain to another using supervised data. Other domain adaptation approaches include SVM-based methods [6], [30].

In this paper, we extend ideas from both areas (i.e. early-event prediction and model adaptation) to enhance our shot prediction capability. Additionally, to enrich our dataset to enable specific shot location prediction at a fine granular level, our approach is also able to use local information to adapt the model to the specific query.

III. Ball and Player Tracking Data

Using multiple fixed cameras, Hawk-Eye records (x, y, z) positions of the ball as a function of t. For each point in a match, we also have additional meta-data such as shot type, outcome, server, score, etc. An example of the ball trajectories in a point is given in Figure 1. Player court positions are

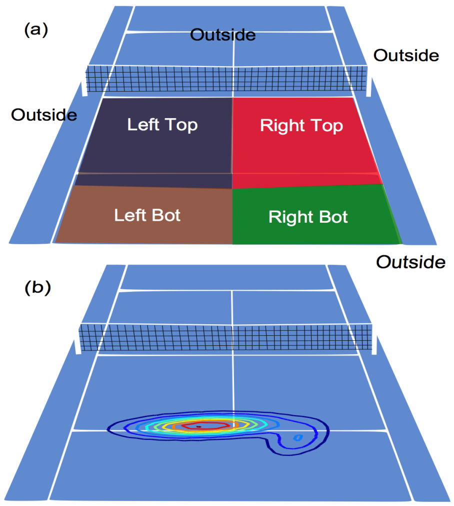

Fig. 2. In this paper, we predicted shots based on (a) zones – 5 regions: left-top, right-top, left-bottom, right-bottom and outside, and (b) continuous region – we give the specific x, y shot location. The quantized zonal region experiments were used to evaluate our method on a small output state-space.

recorded as the (x, y) positions of players on the court at 20 frames per second. By leveraging these tracking data, we are able to augment any point of interest with detailed ball trajectories and player court positions over time. For this work, we used the data from the 2012 Australian Open Men's draw which consisted of more than 10,000 points. We specifically modeled the behavior 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 (Nadal is left handed while Federer and Djokovic are right handed.)

A. *Output State-Space*

Ideally, we want to predict the location of the shot at the most precise level (e.g. millimeter). However, as this essentially represents an infinite output state-space, obtaining enough data to adequately train the model is problematic. To first show the utility of our approach we divided the field into a coarse quantization scheme (see Figure 2(a)). As can be seen from this figure, we divided the receiving player's side of the court into four areas in addition to a catch-all area which captured all shots that fell outside these four areas. The number of shots for each player in each zone is given in Table I.

Our final result uses a weighted Gaussian Mixture Model (GMM) (Figure 2(b)) to represent the probability distribution of a future shot location. We used a local clustering scheme to compensate for the lack of training data, and the discrete prediction result to adjust the weight of the GMM therefore improve the result (see Section IV-C for full details).

B. *Shot Representation*

Before generalizing a player's behavior, we need to find features/factors to represent each shot. Given a shot's start and end time, we are able to find its starting and ending location.

TABLE I

THE TOTAL NUMBER OF POINTS, TOTAL SHOTS, AND NUMBER OF SHOTS SEPARATELY IN EACH ZONE, FOR DJOKOVIC, NADAL AND FEDERER AT THE 2012 AUSTRALIAN OPEN.

Player	Djokovic	Nadal	Federer
Total No. of Points	1916	2234	1372
Total No. of Shots	3410	3488	1882
Outside	516	547	350
Left Top	750	834	473
Left Bot	776	791	355
Right Top	685	607	390
Right Bot	683	709	314

TABLE II

DESCRIPTION OF THE SHOT VARIABLES USED IN THIS PAPER.

Feature	Description
Speed	Shot average speed
Angle	Angle between shot & center line
Feet Location	Player and opponent court position when shot starts
Shot-Start Loc.	Location where shot starts
Shot-End Loc.	Location where shot impacts the court
No. of shots	Total number of shots in the point
Opponent Movement	Local speed & direction of the opponent before the player strikes the ball

Using this information, we calculate the angle, maximum height, average speed and instantaneous speed of the shot. To add player information, we calculate the court position for both players at start and end time of the shot. Table II presents a summary of the shot features used in this paper.

The speed and direction of an opponent is also an important factor when trying to predict the next shot. For example, when a player is forced to return a ball from the deep corner on the fore-hand side of the court, he leaves open the other-side of the court. Depending on his location and speed (i.e. his ability to get to the other side of the court), an opposition player may be more likely to hit to the open court, aiming to win the point. To illustrate this point, we examined some specific examples (see Figure 3). In the top left corner, we show all possible shots over two related time-stamps – the red-dots represent the initial shot location at t_0, and the yellow-dots represent the foot position of player A, also at t_0. The blue dots indicate the court impact location (bounce) of the shot at t_1. The black arrows represents the location and displacement of player B between $(t_0 - 1s)$ and t_0. When we segment the shots according to their starting and ending location, we see some interesting behavior. For instance, in the second column, we show that when a player is hitting the ball in one corner and the opponent is in the other corner of the court - many shots are hit down the line (i.e. in the region where the opponent is not). In the third column (bottom), we see that when player B is not out of position, the normal cross court shot played which has a higher probability of going in and less chance of the opponent hitting a winner.

Additional features such as the player rank, set number, length of match, environment conditions (e.g. hot, humid, cold

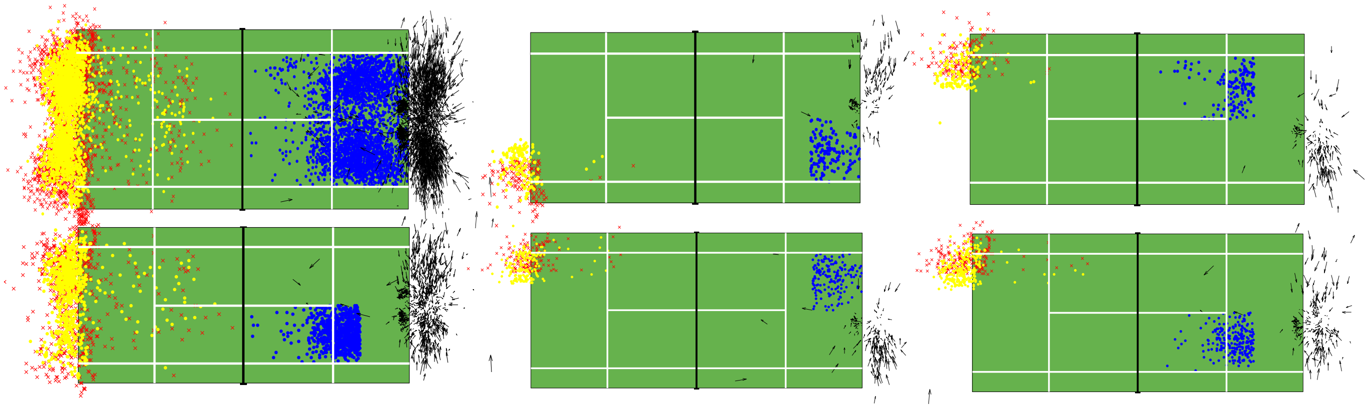

Fig. 3. Plots show the spatiotemporal relationship of events at two time points – red-dots indicate the initial shot location struck by Player A at t_0, and the yellow-dots indicate the foot position of player A at t_0. The blue dots indicate the ball impact location of the shot at t_1, and the black arrows represents the location and displacement of player B over $(t_0 - 1s)$ and t_0.

or windy), specific match context (e.g. game/set/match/break point), court surface (e.g. grass, hard-court, clay) could also be added to enrich the player model. However, as we increase the number of variables the demands on the amount of training data required exponentially increases. Here, we consider a single tournament only, and have therefore omitted those additional factors.

IV. Modeling Player Behavior

A. Game-State Representation

Since most parameters of a shot vary temporally, the relationship between consecutive shots is important. To model and predict shots over time, we used a Dynamic Bayesian Network (DBN). The DBN framework has two levels: 1) a Bayesian Network (BN) that captures varying factors of a rally, and 2) the temporal aspect using the previous state information which can be obtained via the 2-timeslice Bayesian Network (2TBN).

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 (winner/error). A depiction of our point model given a discrete output state-space is given in Figure 4(a), which shows that we have eight possible point-states after the serve $\{\mathbf{z}_3, \ldots, \mathbf{z}_{10}\}$, where $\mathbf{z_i}$ can take a value of {zone1, zone2, zone3, zone4}, when $i = 3, 4, 5, 7, 9$ and $\mathbf{z_i} = \{\text{Out}\}$, when $i = 6, 8, 10$.

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) at time t, and we know the previous state $\mathbf{z}^{t-1}$ (i.e. player A or player B returned the ball), using our model topology shown in Figure 4(b), we can infer the probability of next shot landing at each zone by using Baye's law:

$$P(\mathbf{z}^t|\mathbf{z}^{t-1}, \mathbf{x}^t) = \frac{P(\mathbf{x}^t|\mathbf{z}^t)P(\mathbf{z}^t|\mathbf{z}^{t-1})}{P(\mathbf{x}^t|\mathbf{z}^{t-1})} \quad (1)$$

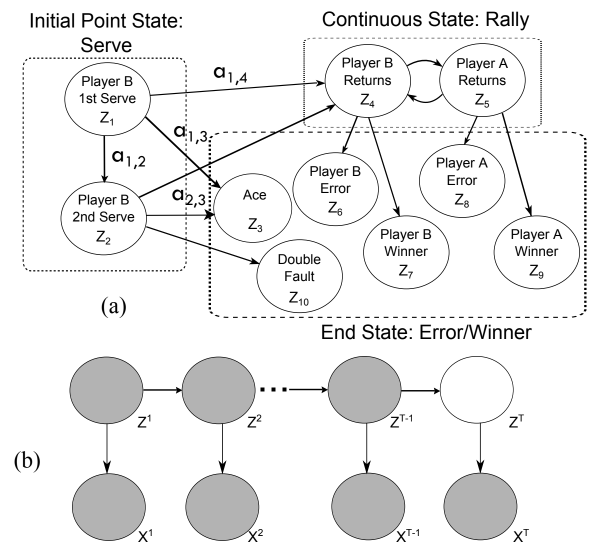

Fig. 4. (a) The model of a point in tennis - given player B serves the ball and it is not an ace or double fault, player A returns the ball back to one of the four zones of player B and the rally ensues until player A or B hits a winner or an error. (b) We use a DBN to predict the next state $\mathbf{z}^t$ given the current observation $\mathbf{x}^t$ and previous state $\mathbf{z}^{t-1}$(gray nodes are observed and clear nodes are hidden).

where the next state is conditioned on the previous state. Depending on the player returning the shot, we infer the probability of $\{\mathbf{z}_5^t, \mathbf{z}_8^t, \mathbf{z}_9^t\}$ for player A and $\{\mathbf{z}_3^t, \mathbf{z}_4^t, \mathbf{z}_6^t, \mathbf{z}_7^t, \mathbf{z}_{10}^t\}$ for player B - and our prediction is the state, $\mathbf{z}_i^t$, with the highest probability.

B. Model Learning

The transition probabilities $a_{i,j}$ in a point model are computed from the conditional probability distribution (CPD) of two player models. To learn the CPDs for each player, we first created a shot database where each shot was labeled with its type (i.e. zone 1, zone 2, winner, error, serve). Given these labeled shots, we then built probability distribution functions

(PDF) for shot variables which yielded a multi-dimensional PDF for each shot type for each player. To obtain a continuous distribution of these PDFs, we employed a Gausian Mixutre Model, where

$$P(x|\boldsymbol{\theta}) = \sum_{k=1}^{M} \omega_k G(x; \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}) \tag{2}$$

given that the GMM has the form:

$$G(x; \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}) = \frac{1}{2\pi^{\frac{d}{2}} |\boldsymbol{\Sigma_k}|^{\frac{1}{2}}} \exp(-\frac{1}{2}(x-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(x-\boldsymbol{\mu})) \tag{3}$$

where $\boldsymbol{\mu}$ is the mean, $\boldsymbol{\Sigma}$ is the covariance and $\boldsymbol{\theta} = (\omega_1; \boldsymbol{\mu_1}, \boldsymbol{\Sigma_1}), ..., (\omega_M; \boldsymbol{\mu_M}, \boldsymbol{\Sigma_M}))$ are the parameters of the GMM for M mixtures. The parameters of $\boldsymbol{\theta}$ are learned 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 V.

C. Continuous Output

In the previous section, the predicted location is represented as a discrete variable. To get a more precise prediction, a continuous output location is required (i.e. (x, y) position). This is challenging however, as the amount of data required to train a model is prohibitively large, because the combination of shot factors and locations is essentially infinite. To combat this we developed a local clustering method. The process is as follows:

1) Given a shot s_1 and its previous state z_0, our goal is to predict where the next shot s_2 will land for a player. We denote the speed, location and angle of s_1 as $\mathbf{v_1}$.
2) After that, in feature space, we define a region which is centered at $\mathbf{v_1}$ and has a $L2$ distance less than l to $\mathbf{v_1}$. This can be interpreted as a hypersphere centered at $\mathbf{v_1}$ in a high-dimensional space.
3) Next, we retrieve shots from an historical player shot database where the preceding shot feature falls within this defined region.
4) We build a 2D GMM on the court impact locations of the retrieved shots. So each possible impact location (x, y) will be assigned a probability. We denote the output of GMM as $\mathbf{y_c}$
5) Finally, we compute the complete features vector for s_1. Using this feature vector and z_0, we infer a discrete output $\mathbf{y_d}$ using our Dynamic Bayesian Model. In this case, each *region* will be assign a probability. The discrete prediction result $\mathbf{y_d}$ is then used to compensate $\mathbf{y_c}$ by adjusting the weight of the GMM.

This emphasizes the probability distribution for $\mathbf{y}_c$ when it's within the discrete prediction region while leaving it the same when it's outside the discrete predicted result. The idea is, we first employ a GMM to roughly estimate the next shot location use only a subset of features. Then we use our well trained DBN model to get a discrete prediction of next shot location. The discrete result is then used to compensate the continuous estimation.

V. Player Adaptation

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 a "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 difficult since players may only play each other several times a year, and this is often on different surfaces.

A method to resolve this issue is to employ adaptive model techniques which are commonly used in speech and speaker verification tasks [23]. 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. Since UBM is well-trained using all available data, this approach often improves the result when lacking of data.

Given the initial model parameters or UBM parameters, $\boldsymbol{\theta}_{UBM} = (\omega_1; \boldsymbol{\mu_1}, \boldsymbol{\Sigma_1}), ..., (\omega_M; \boldsymbol{\mu_M}, \boldsymbol{\Sigma_M}))$ as well as the parameters of the held-out matches $\boldsymbol{\theta}_{Adapt} = (\omega_1; \boldsymbol{\mu_1}, \boldsymbol{\Sigma_1}), ..., (\omega_M; \boldsymbol{\mu_M}, \boldsymbol{\Sigma_M}))$. We then update the parameters of the UBM by using the following equations:

$$\omega_k^* = \frac{\alpha^{\omega_k}}{n} + (1 - \alpha^{\omega_k})\omega_k \tag{4}$$

$$\boldsymbol{\mu}_k^* = \alpha_k^m E_k(x) + (1 - \alpha_k^m)\boldsymbol{\mu}_k \tag{5}$$

$$\boldsymbol{c}_k^* = \alpha_k^v E_k(x^2) + (1 - \alpha_k^v)(c_k + \boldsymbol{\mu}_k^2) - \boldsymbol{\mu}_k^{*2} \tag{6}$$

where $(\omega_k^*; \boldsymbol{\mu}_k^*, \boldsymbol{\Sigma}_k^*)$ are the new parameters of the adapted model, and $\alpha_k^p, p \in (\omega, \mu, v)$ are used to control the balance between old and new estimates for weights, means and covariances. In this work, we investigate two model adaptive methods: 1) *pre-game* adaptation, and 2) *online* adaptation.

In Pre-Game Adaptation, a UBM is first trained to model a player's behavior. The train sets consist of all shots from this player's matches regardless of opponents. Next, held-out data are collected from two matches from previous tournaments. These matches were arguably 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. Finally, we adapt our UBM to these matches using above algorithms.

In contrast, online adaptive method adapts the UBM continuously to the data just observed (i.e. current match). The parameters of the GMM are updated after every shot and tested

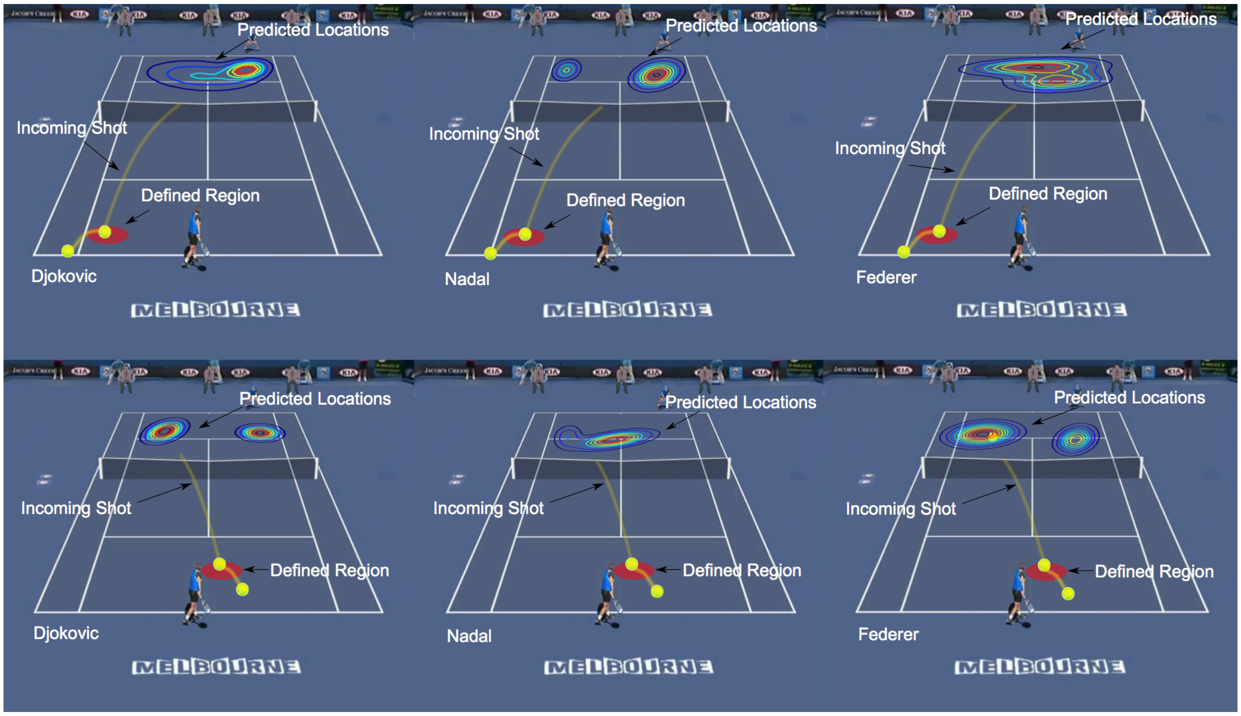

Fig. 5. Given the same incoming shot, we can predict the next shot in continuous space for Djokovic, Federer and Nadal. For each player, we shown examples of two different incoming shots. In the figure, red circle refers to our defined region. Distributions on the top represent the probability of predicted shot locations.

on next set. This idea is inspired by [12] where they re-adapt a pre-existing face classifier to each new image it encounters to improve face detection performance. In our case, we wish to rapidly adapt our UBM to a new test data set (e.g. shots from a new game, shots from a new set) from the current match to improve predictive performance. While similar in spirit, the algorithms are different as they are dealing with a binary classification problem where they defined a small margin near the boundary of a classifier. Data outside this small margin (confident positive or negative examples) are then used to learn a Gaussian Process Regression model which is used to reclassify the data points with prediction values lying inside the margin. Our approach on the other hand deals with an infinite output state problem where we are interested in adapting the distribution of our training sets to a new test data set.

TABLE III
PERFORMANCE OF OUR PLAYER MODELS FOR PREDICTING THE NEXT SHOT LOCATION USING AN "ONE-VERSUS-EVERYONE ELSE" OR UNIVERSAL BACKGROUND MODEL.

Shot Variable	Out	L-T	L-B	R-T	R- B
Speed	60,1	57.7	63.7	52.8	58.1
Angle	51.2	47.9	50.3	54.9	49.9
Feet Location	59.4	54.2	59.3	52.0	54.3
Player Movement	65.8	59.7	62.8	60.9	61.3
Start Location	61.4	59.8	59.3	57.7	60.5
No. Shots in Rally	55.1	51.0	50.6	47.6	54.2
Speed+Angle	59.2	55.5	57.5	51.6	57.4
Speed+Start Loc+ Player Mov	69.8	65.7	73.4	70.8	71.5
Speed+Start Loc+ Player Mov+Pre info	76.2	70.6	77.7	74.1	71.8

VI. EXPERIMENTS

A. Experimental Setup

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. 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). For each model, the training set was separated from the testing set. For example, the Federer and Djokovic models were trained on matches, against all opponents except Nadal. For discrete prediction, 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).

B. Discrete-Output Results

1) Universal Background Model Result: In Table III, we show the aggregate shot prediction performance for each zone using different factors and combinations. As can be seen from the results, player movement was the best single predictor of

TABLE IV
PRE-GAME ADAPTATION MODEL RESULTS.

Shot Variable	Out	L-T	L-B	R-T	R- B
Speed+Loc+Feet Loc +No. of shots	79.7	78.0	80.2	74.1	75.7

most zones achieving an AUC around 63%, while the shot start location was the second best predictor of Outside, Left Top, Right Top, Right Bottom. Speed outperforms all other single predictors when predicting Left Bottom zone. When combining factors together, speed + start location + player movement gave the best result around 74% AUC. Even though the performance improves, the overall predictive power is still quite poor. We believe the absence of opponent modeling could be one reason to explain this.

2) Pre-Game Adaptation Result: Using this Pre-Game adaptive method, we were able to improve the prediction performance for most zones. These results are presented in Table IV , and the biggest improvement is evident for the Left-Top zone, which improved by 7.4%. Most other zones has been improved by 3%. Right-Top remains the same performance as UBM.

3) Online Adaptation Result: As the results show in table V, this online method generally performs more poorly in the beginning, improves significantly throughout the match. By set 3, Online-Adaptation already outperforms the Pre-Game method. The results at set 4 are similar to set 5 with most zones achieving greater than 80% AUC. Sport is a highly uncertain domain, and these results represent important predictions using spatiotemporal data (Figure 7 presents a further performance comparison of the methods).

C. Continuous-Output Results

We also used distance between actual shot location and estimated location to measure the continuous prediction performance. In Figure 8, the y axis represents the hit-rate while x axis represents the distance of error in meters. The models for Djokovic and Nadal demonstrated superior performance to that for Federer. This may due to the fact that the Federer model was trained on fewer points and shots that the other

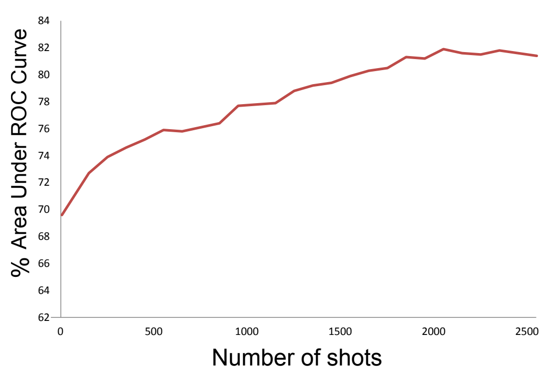

Fig. 6. Learning rate for online adaptation.

TABLE V
PERFORMANCE OF ONLINE-ADAPTIVE MODEL OVER TIME USING SPEED + LOC + FEET LOC

Shot Variable	Out	L-T	L-B	R-T	R- B
Set 1	69.9	72.3	66.6	70.3	63.5
Set 2	74.7	79.7	72.4	76.3	69.9
Set 3	82.8	79.4	77.0	76.8	73.7
Set 4	81.4	84.0	83.5	81.9	82.4
Set 5	80.8	83.1	84.2	82.1	79.8

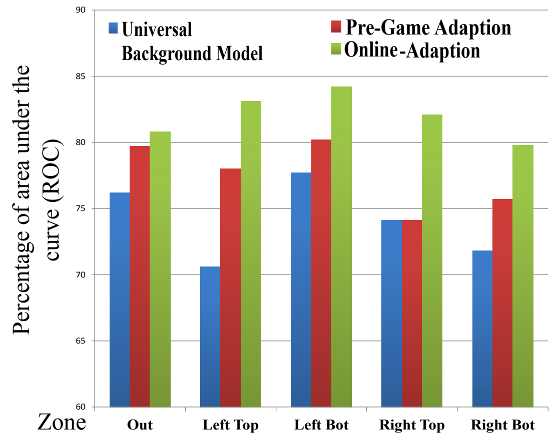

Fig. 7. Comparison of performance for three different models

models (see Table I). Average error distance for Djokovic and Nadal is around 1.7 meters while Federer has an average error distance of 2.3 meters.

D. Processing Time and Learning Rate

Processing time is an important aspect for this research. Prediction will be meaningless if the result takes too long to compute. In order to test that, we run our program for 500 times. It takes 0.136 seconds in average to get the prediction result using Matlab on a normal PC (Intel Core 2, 3GHz Processor with 4GB Ram). So it is quick and possible for real-time prediction given we have the input data. Less computation time can be achieved using more powerful machines.

We also conducted experiment to investigate learning rate of our online adaptation method (See Figure 6). Since the parameters of the GMM are updated every shot, we tested its performance after each iteration. As the figure shown, its performance starts at around 70% AUC (area under the ROC curve), rising sharply to 74% after 200 shots and reaches its peak at 82%.

VII. SUMMARY

Accurate vision-based tracking systems are emerging in sports such as tennis, which are well suited to novel analyses using probabilistic graphic models. In this paper, we employed a Dynamic Bayesian framework to build a stroke-by-stroke model that is predictive of the discrete location of any shot in a rally. Moreover, we used a local clustering scheme to enhance our discrete prediction result and estimate the actual x, y

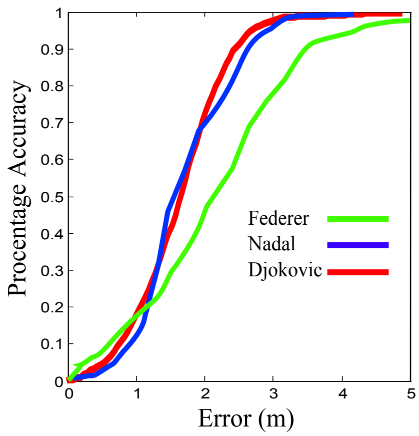

Fig. 8. Plot shows the percentage accuracy against error distance in meter to measure the performance of continuous prediction. It can be seen that the predictability of each player is different. Djokovic and Nadal is more predictable than Federer.

location of a future shot. This analysis creates novel insights to the playing styles of individual players, and in particular, we identify the style of three top male tennis players in the world. Our modeling approach demonstrates superior performance using two different adaptive techniques: 1) pre-game adaption; 2) online adaption, which allow greater sensitivity by tuning the model to specific match parameters such as opponent or 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.

Our research represents some of the first work to exploit new and rich data from the Hawk-Eye system. We have constrained our analysis to a selection of three elite players in a single tournament, but future work will demonstrate the scalability of the methods with many more players across multiple tournaments. Future work will also investigate other predictive models including conditional random fields, and structured support vector machines

REFERENCES

[1] I. Atmosukarto, B. Ghanem, S. Ahuja, K. Muthuswamy, and N. Ahuja. Automatic recognition of offensive team formation in american football plays. In *IEEE International Workshop on Computer Vision in Sports*, 2013.

[2] J. Berclaz, F. Fleuret, E. Tureken, and P. Fua. Multiple object and tracking using and k-shortest paths and optimization. *T. PAMI*, 2011.

[3] A. Bialkowski, P. Lucey, P. Carr, S. Denman, S. Sridharan, and I. Matthews. Recognizing team activities from noisy data. In *IEEE International Workshop on Computer Vision in Sports*, 2013.

[4] W. Choi and S. Savarese. A unified framework for multi-target tracking and collective activity recognition. In *ECCV*, 2012.

[5] R. Collins. Multitarget Data Association with Higher-Order Motion Models. In *CVPR*, 2012.

[6] L. Duan, I. W. Tsang, and D. Xu. Domain Transfer SVM for Video Concept Detection. In *CVPR*, 2009.

[7] Hawk-Eye. http://www.hawkeyeinnovations.co.uk.

[8] A. Hervieu and P. Bouthemy. Understanding sports video using pplayer trajectories. In *Intelligent Video Event Analysis and Understanding. Springer Berlin/ Heidelberg*, 2010.

[9] M. Hoai and F. de la Torre. Max-margin early event detectors. In *CVPR*, 2012.

[10] IBM SlamTracker. www.australianopen.com/en_AU/ibmrealtime/index.html, 2012.

[11] S. Intille and A. Bobick. A framework for recognizing multi-agent action from visual evidence. In *AAAI*, 1999.

[12] V. Jain and E. Miller. Online domain adaptation of a pre-trained cascade of classifiers. In *CVPR*, 2011.

[13] K. Kim, M. Grundmann, A. Shamir, I. Matthews, and J. Hodgins. Motion fields to predict play evolution in dynamic sports scenes. In *CVPR*, 2010.

[14] K. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert. Activity forecasting. In *ECCV*, 2012.

[15] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, 2011.

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

[17] R. Li and S. Zhou. Learning multi-modal densities on discriminative temporal interaction manifold for group activity recognition. In *CVPR*, 2009.

[18] 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.

[19] P. Lucey, D. Oliver, P. Carr, J. Roth, and I. Matthews. Assessing team strategy using spatiotemporal data. In *ACM SIGKDD Conference on Knowledge, Discovery and Data Mining (KDD)*, 2013.

[20] R. Maheswaran, Y. Chang, A. Henehan, and S. Danesis. Deconstruction the rebound with optical tracking data. In *MIT Sloan Sports Analytics Conference*, 2012.

[21] M. Perse, M. Kristan, S. Kovacic, and J. Pers. A trajectory-based analysis of coordinated team activity in basketball games. In *CVIU*, 2008.

[22] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 2011.

[23] D. Reynolds, T. Quatieri, and R. Dunn. Speaker verification using adapted gaussian mixure models. *Digital Signal Processing*, 2000.

[24] M. Rodriguez, J. Sivic, I. Laptev, and J.-Y. Audibert. Data-driven crowd analysis in videos. In *ICCV*, 2011.

[25] M. S. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *ICCV*, 2011.

[26] 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*, 32:2, 201.

[27] 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.

[28] X. Wei, P. Lucey, S Morgan, and S. Sridharan. Sweet-spot: Using spatiotemporal data to discover and predict shots in tennis. In *MIT Sloan Sports Analytics Conference*, 2013.

[29] M. Yamada, L. Sigal, and M. Raptis. No bias left behind: Covariate shift adaptation for discriminative 3d pose estimation. In *ECCV*, 2012.

[30] J. Yang, R. Yan, and A. Hauptmann. Cross-domain video concept detection using adaptive svms. In *ACM Multimedia*, 2007.

[31] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008.

[32] G. Zhu, Q Huang, C. Xu, Y. Rui, S. Jiang, W. Gao, and H. Yao. Trajectory based event tactics analysis in broadcast sports video. In *ACM Multimedia*, 2007.

[33] B. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, and K. Peterson. Planning-based prediction for pedestrians. In *IROS*, 2009.

View publication stats