TACTICAL ANALYSIS MODELING THROUGH DATA MINING
Pattern Discovery in Racket Sports

Antonio Terroba
Telefonica I+D, Madrid, Spain
ata@tid.es

Walter A. Kosters, Jonathan K. Vis
Leiden Institute of Advanced Computer Science, Universiteit Leiden, The Netherlands
kosters@liacs.nl, jvis@liacs.nl

Keywords: Data mining, pattern, sequence, tennis.

Abstract: We explore pattern discovery within the game of tennis. To this end, we formalize events in a match, and define similarities for events and event sequences. We then proceed by looking at unbalancing events and their immediate prequel (using pattern masks) and sequel (using nondeterministic finite automata). Structured in this way, the data can be effectively mined, and a similar approach might also be applied to more general areas. We show that data mining is able to find interesting patterns in real-world data from tennis matches.

1 INTRODUCTION

The analysis of tennis sequences has been studied before with an aim to either automatically annotate the score or to classify the content for later retrieval, see, e.g., (Sudhir et al., 1998; Calvo et al., 2002; Christmas et al., 2005; Zhu et al., 2006). This analysis and the methods to recognize and classify the images have been usually undertaken by the computer vision research community. However, the study of the captured data in order to find patterns and relationships between variables (Tan et al., 2005) is relatively novel. The objective of this paper is to establish a framework that allows us to obtain such knowledge.

The contributions of this paper are fourfold. Firstly, we establish a framework for multivariate data mining based on distances and thresholds. Secondly, we introduce the concept of *pattern masks* as a means to mine regular patterns. Thirdly, splitting patterns into a *prequel* and a *sequel*, we propose an efficient algorithm to mine winning patterns, anchored on so-called *unbalancing events*. For the prequel we consider a distance notion based on event similarities, whereas the sequel has to comply with a nondeterministic finite automaton. Finally, we apply the framework to real-world examples and extract novel knowledge in the sports strategy arena. In this way, where current analysis simply states winner percentages, we are able to indicate how these winners were performed and how they relate to each other.

The rest of the paper is organized as follows. Section 2 contains related work. In Section 3 we formalize a tennis match and present definitions used in the rest of the paper. In Section 4 we define the concepts of multivariate similarity, similarity thresholds and pattern masks, as well as the mining problem to consider. In Section 5, we propose an algorithm to find winning patterns. Finally, we present the results obtained in Section 6 and the conclusions in Section 7.

2 RELATED WORK

Wang et al. (Wang et al., 2005) treat the subject in a similar way, but they only consider relative player movements and no other variables. Wang and Parameswaran (Wang and Parameswaran, 2005) take into account 58 possible patterns and try to find them in the footage using Bayesian networks. Zhu et al. (Zhu et al., 2007) propose a tactic representation based on temporal-spatial interactions in soccer. Lames (Lames, 2006) focuses on relative phases of lateral displacements.

Schroeder et al. (Schroeder et al., 2005) use a framework based on short term and long term memory that allows an incremental processing of data streams. However, the tennis model used only in-

cludes one variable (the ball landing position) and only eight different locations. Chu and Tsai (Chu and Tsai, 2009) use symbolic sequences to tackle tactics analysis. They use players location (four areas), players movement direction (up, down, left, right, still) and players speed (fast, medium, still) to find frequent movement patterns.

3 FORMALIZATION

In this section we explain how we formalize a tennis match between two players, 1 and 2. For the rules of tennis, the reader is referred to (International Tennis Federation, 2010).

Although many computerized systems exist for collecting and managing observational data, our need to record the exact position of the players and the ball on the court, forced us to develop a standalone application that allowed us to calculate those positions on a *reference court model* by means of computer vision algorithms and camera calibration techniques. It is not the aim of this paper to detail the methods and algorithms used to obtain the data. The interested reader is referred to (Hartley and Zisserman, 2003; Hayet et al., 2005) for further information. Along with player and ball positions, other relevant variables were also collected as part of our sequential data.

3.1 Definitions

We will consider an *event* as a single stroke episode. This event will contain all attributes that characterize the stroke, i.e., the player that hits the stroke, the type of stroke, the position of both players at the time of hitting the ball, the position of the ball landing on the opponent's side after the stroke, the generated speed of the ball, etc. A *rally*, on the other hand, refers to the sequence or series of events that completely describe the strokes exchanged by the players during a game point. In other words, a rally will always start with a service and will end with the final stroke that leads to the conclusion of the point.

We will also define a *partial rally* as a subsequence of a rally. Partial rallies are made of consecutive events, with players alternating. For instance, looking at rally $\langle A,B,C,D,E\rangle$, then $\langle B,C,D\rangle$ is a partial rally, whereas $\langle B,D\rangle$ is not.

3.2 Reference Model

All integer coordinate pairs of events will be in the set $C = \{0,1,\ldots,316\} \times \{0,1,\ldots,768\}$. The positions between $(0,0)$ and $(316,768)$ represent coordinates both inside and outside of the court, being $(50,150)$ and $(266,618)$ the coordinates of the top left corner and the bottom right corner of the doubles court respectively. This reference system gives us 2.5 m of space at each side of the doubles sidelines and 7.5 m at each side of the baselines which is sufficient to capture all the action within a match.

Because the players change sides every couple of games, a transformation in the coordinates is needed so that the data is always coherent.

3.3 Attributes Considered

We will now first focus on the *stroke level* and *rally level*. There we have the following attributes (for each attribute the possible values are mentioned):

- *pl*: player hitting the ball, $\{1,2\}$;
- *st*: stroke type, $\{FS,SS,FH,FHS,BH,BHS,VOL,SM,LOB,DSH\}$, corresponding to: first serve, second serve, forehand, forehand sliced, backhand, backhand sliced, volley, smash, lob and drop shot, respectively;
- $P_1 = (x_1,y_1)$: position of the player when the ball is hit, C;
- $P_2 = (x_2,y_2)$: position of the opponent when the ball is hit, C;
- $P_3 = (x_3,y_3)$: position of the ball when it bounces on the opponent's half of the court, C;
- *sb*: speed of the ball generated after the stroke, $\{slow,normal,fast\}$;
- *us*: unbalancing stroke that breaks the exchange equilibrium, $\{0,1,2,3\}$.

As an example, a sequence including the first events within a rally might look like this:

$\langle(2,FS,(142,618),(231,56),(163,267),fast,1),$
$(1,BHS,(191,64),(134,610),(103,566),slow,0),$
$(2,FH,(78,608),(173,55),(108,239),fast,2),\ldots\rangle$

Most attributes are self-explanatory. Attribute *us* represents the intention of one player to attack and destabilize the rally with his/her stroke. The non-zero values indicate whether it is a first, second or third attack. Very rarely a player will need more than three strokes to finish an attack, and in such a case, one could argue that the opponent did recover from the initial attack and lost the point later on due to a new and different attack.

4 PATTERN MINING

In this section we describe all necessary definitions. We start with relatively simple similarity measures, and generalize these to so-called *pattern masks*.

4.1 Similarity Measure

First, we define a similarity measure *sim* between individual events. In this case, when we have events $e = (pl, st, P_1, P_2, P_3, sb, us)$ and $e' = (pl', st', P_1', P_2', P_3', sb', us')$, we put:

$$\begin{aligned} sim(e,e') &= simplayer(P_1,P_1') + simplayer(P_2,P_2') \\ &\quad + simball(P_3,P_3') + simstroke(st,st') \\ &\quad + \delta(sb,sb') + \delta(us,us') \end{aligned} \tag{1}$$

if $pl = pl'$, where each function determines the similarity between the corresponding attributes. If $pl \neq pl'$ then $sim(e,e') = 0$ (it is in this case also possible to apply a rotation to the coordinates involved; we will return to this issue in a subsequent paper). With $dist(P,Q)$ representing the Euclidean distance between points P and Q, we define:

$$simplayer(P,Q) = f(dist(P,Q)) \in [0,1] \tag{2}$$

$$simball(P,Q) = g(dist(P,Q)) \in [0,1] \tag{3}$$

$$simstroke(st,st') = \delta(st,st') + \varepsilon(st,st') \in [0,1] \tag{4}$$

$$\delta(u,v) = \begin{cases} 1 & \text{if } u = v \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Here we have used suitable monotonically decreasing functions f and g with $f(0) = g(0) = 1$. The function ε allows for additional weight in the case of near equal stroke types. All of the six terms can get their own weight, if necessary (cf. Section 4.3). Note that $0 \leq sim(e,e') \leq sim_{\max}$ for suitable $sim_{\max} \leq 6$.

Now that we have defined the similarity between events, we can easily determine the similarity $sim(seq_1, seq_2)$ between same-length sequences (or partial rallies) seq_1 and seq_2 of single events as follows. If the length of both sequences equals n and $seq_1 = \langle e_1, \ldots, e_n \rangle$ and $seq_2 = \langle e_1', \ldots, e_n' \rangle$, then:

$$sim(seq_1, seq_2) = \sum_{i=1}^{n} sim(e_i, e_i') \tag{6}$$

If the sequences are of unequal length, we define their similarity to be 0.

4.2 Similarity Thresholds

Once we know the similarity value between events $sim(e,e')$ and sequences $sim(seq_1, seq_2)$, we need to establish the criteria by which we will consider two events or two sequences as similar.We will use the thresholds $event_{\text{thr}}$ and $series_{\text{thr}}$ for this matter. Note that we are defining two different thresholds to allow greater flexibility. This way, two events e and e' will be considered similar if and only if $sim(e,e') \geq event_{\text{thr}}$ and likewise, two sequences seq_1 and seq_2 of length n will be considered similar if and only if $sim(seq_1, seq_2) \geq n \times series_{\text{thr}}$.

4.3 Pattern Masks

It will be shown later that we might want to compare two sequences that do not correlate exactly. A typical example will be the response to an attack that may produce different answers. For instance, a fast first serve to the same corner may result in 1) an ace, 2) a forced error or 3) a short ball that will trigger a winner. All these cases have one thing in common: the initial attacking service. However, the short ball in case 3 might bounce in many areas and therefore the similarity measure defined above cannot be used.

Thus, in this case, the sequence similarity will be more relaxed at certain points than others, and only some events will enforce a high similarity condition. In other words, we are trying to identify sequential patterns with constraints.

Before we define the generalized pattern similarity measure, we introduce the concept of a *pattern mask* $pmask = \langle sim_1, sim_2, \ldots, sim_n \rangle$, where each sim_i represents a particular similarity measure (a simple example being $sim_i = sim$ from Section 4.1, $i = 1, 2, \ldots, n$). This definition implies that a variety of different similarity measures for each event within the sequence could be used, e.g., concentrating on the stroke types. Some similarity functions will indeed favor certain attributes over others in order to fully characterize a pattern.

In this case, a sequence $seq_1 = \langle e_1, \ldots, e_n \rangle$ will be considered similar to a sequence $seq_2 = \langle e_1', \ldots, e_n' \rangle$ (with respect to *pmask* and corresponding thresholds $event_{\text{thr},i}$ $(i = 1, 2, \ldots, n)$), if and only if:

$$sim_i(e_i, e_i') \geq event_{\text{thr},i} \text{ for } i = 1, 2 \ldots, n \tag{7}$$

Therefore, for a particular event, the similarity threshold could be very low or even 0, meaning that event wildcards could effectively be allowed. Note also that this similarity implies the sequence similarity concept defined in Section 4.2, when the pattern mask is made of equal similarity functions, all sharing the same threshold $series_{\text{thr}}$. Instead of adjusting the thresholds, it is also possible to rescale the similarity functions; however, the current approach seems to have a better underlying intuition.

4.4 Mining Problem

We are now able to define our mining problem. Given a match between two players, we want to determine the partial rallies that lead to winners or forced errors. In this case, we are not so much interested in finding very close similar partial rallies, but rather similar attacking patterns that may bring about different defensive responses that do not have to be exactly similar. These patterns should also occur often enough, i.e., be *frequent*. More precisely, we define:

Mining Problem — Winning Rallies
Given a pattern mask *pmask* of a certain length n with corresponding thresholds, and a minimum support threshold *min_support*, determine those partial rallies seq_1 in the match that end with an unbalancing event, and for which there are at least *min_support* partial rallies seq_2 that satisfy Equation 7. Such a rally seq_1 is called a *winning (partial) rally*.

---

5 APPROACH

The key to finding similar winning patterns is to identify similar attacking events. These events will act therefore as fingerprints in the process.

5.1 Completion of Attack Patterns

We first establish the following equivalences. If we call 1 a first attacking event and *FE* a possible forced error as a consequence of 1, and then, depending on whether the first attack results in a winner (meaning a stroke that will not get a response from the opponent) or in a forced error, we state that $1, EOR \equiv 1, FE$, where *EOR* denotes the end of a rally. Note that *FE* automatically includes this last event.

The implication of the previous equation is that two sequences of different lengths can be similar and will represent nonetheless the same winning pattern. Similarly, if 2 represents a second attacking event performed by the player that produced event 1, then we have $1, \bowtie, 2, EOR \equiv 1, \bowtie, 2, FE$, where $\bowtie$ indicates an event (not being *FE*) that does not carry strategic information, as it is a forced defensive response, and therefore no similarity constraint should be enforced. It will usually be a soft ball that can be attacked. And analogously: $1, \bowtie, 2, \bowtie, 3, EOR \equiv 1, \bowtie, 2, \bowtie, 3, FE$.

The three equivalences above represent the basic patterns to finish an attack depending on whether the attacking player needed one, two or three strokes to finalize the point.

5.2 Pattern Prequel and Sequel

For each winning pattern, we define its *prequel* as the sequence of events that appear in the pattern up to the first attacking event. Similarly, we define its *sequel* to be the remaining events in the pattern. We consider the first unbalancing event as being part of both prequel and sequel.

For the remainder of the section, and in order to describe a winning pattern, we will use the following convention. We will continue to use 1, 2 and 3 to indicate the first, second and third unbalancing event, *FE* to indicate a forced error event and $\bowtie$ to indicate *any* event (again not being *FE*). We will also use X, Y, Z to indicate *a* particular event on which we may enforce a similarity function.

Take, for example, the following pattern. The two players are exchanging crosscourt strokes keeping the ball deep until one player gets a short ball that triggers an attack changing the direction and driving the ball down the line. This represents pattern 19 from (United States Tennis Association (USTA), 1996). If X represents the crosscourt stroke and assuming that we do not want to impose any similarity check on the response to the attack, then the pattern of the prequel could be represented as: $p = \langle X, \bowtie, 1 \rangle$. In this case, the possible sequels would be $\langle 1 \rangle$, $\langle 1, FE \rangle$, $\langle 1, \bowtie, 2 \rangle$, $\langle 1, \bowtie, 2, FE \rangle$, $\langle 1, \bowtie, 2, \bowtie, 3 \rangle$ or $\langle 1, \bowtie, 2, \bowtie, 3, FE \rangle$.

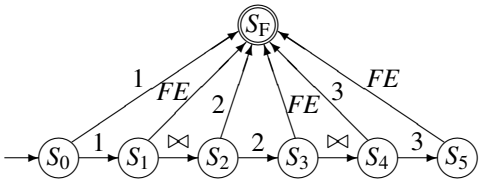

Figure 1: NFA for the winning pattern sequel.

The sequel can be represented by a nondeterministic finite state machine or nondeterministic finite automaton (NFA) which can be dealt with in the pattern mining computation. See Figure 1 where S_0 is the initial state, and S_F represents the final state.

5.3 Algorithm

In order to clarify the algorithm, and to explain the different choices made so far, we begin with an example. Note that we will use the Mining Problem from Section 4.4 for the prequel and the NFA from Section 5.2 for the sequel.

Figure 2 below shows a variation on the pattern just mentioned. Here, we are interested in studying three events prior to the attacking one. In this case,

we use the pattern $\langle X, Y, Z, \bowtie, 1 \rangle$ to try to find a similar sequence of three events $\langle X, Y, Z \rangle$ that will allow the first player to attack the ball and unbalance the opponent. The use of the pattern mask allows to select which events in the pattern should have a high similarity. This figure also takes into account both the prequel and sequel of the winning pattern.

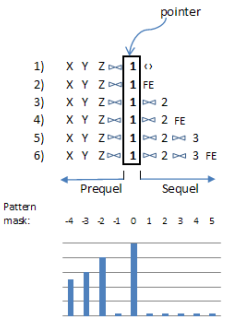

Figure 2: Winning pattern similarity.

In this example, if we assume for simplicity that all similarity functions in the pattern mask are the same, $event_{\text{thr},e}$ represents the event similarity threshold for the event e, and $e_i.us$ represents the unbalancing stroke attribute of event i, then the two rallies:

$$r_1 = \langle e_{11}, e_{12}, e_{13}, e_{14}, e_{15} \rangle$$
$$r_2 = \langle e_{21}, e_{22}, e_{23}, e_{24}, e_{25}, e_{26}, e_{27} \rangle$$

where e_{15} and e_{27} are both last events, will be similar and belong to the same winning pattern $\langle X, Y, Z, \bowtie, 1 \rangle$ if all the following conditions are true:

$$sim(e_{11}, e_{21}) \geq event_{\text{thr},X},\ sim(e_{12}, e_{22}) \geq event_{\text{thr},Y},$$
$$sim(e_{13}, e_{23}) \geq event_{\text{thr},Z},\ sim(e_{15}, e_{25}) \geq event_{\text{thr},1},$$
$$e_{15}.us = 1,\ e_{25}.us = 1,\ e_{27}.us = 2$$

The algorithm implemented to identify the winning patterns is described in the pseudocode from Figure 3. Firstly, we locate events that verify the condition of being first attacking events. Then for each pattern, we expand the projected database (cf. (Pei et al., 2001)) in depth-first fashion checking from the pointer to the left using the similarity mask. For each sequence found, we expand likewise the sequel to the right checking the NFA as well. Several optimizations are possible, like search space pruning, but the current implementation does not focus on this issue, the datasets being of relatively small size.

```
input  R, a series of rallies;
       pmask, a pattern mask (with thresholds);
       NFA, an automaton;
       min_support, a threshold
output W, a set of winning patterns with support
begin
    Put all events e from R with e.us = 1 into set S
    foreach e ∈ S
        support ← 0
        foreach e' ∈ S with e ≠ e'
            if prequels similar according to pmask
            and sequels satisfy NFA
                support ++
        if support ≥ min_support
            Add prequel and support to W
    return W
end
```

Figure 3: Algorithm — Winning patterns identification.

6 RESULTS

Over 3,000 events from more than 7 hours of recordings where captured and analyzed, covering men's and women's matches in both hard and clay courts.

As a first experiment, we tried to analyze the successful service winning patterns displayed by the players. Depending on the court surface, these points can account for more than half the total points won (i.e., excluding unforced errors by the opponent).The winning pattern here is simply $\langle 1 \rangle$, equal to its prequel (no prior events: the unbalancing stroke belongs to a service) and the usual sequel of $\langle 1 \rangle$, $\langle 1, FE \rangle$, etc.

The three left panels from Figure 4 show a few examples of successful service winning patterns found for the 2010 Australian Open semifinal between Na Li and Serena Williams. Black circles represent player positions, yellow (light) circles refer to ball landing positions. Each panel represents the same winning pattern (service to the T on the Deuce court), being completed by one, two or three strokes, respectively.

A second experiment was set up to try to find groundstroke attacking patterns. The winning pattern here was set to be $\langle X, Y, 1 \rangle$. The pattern mask is set in such a way that the event threshold at the first unbalancing event (or pointer) and the event Y is fairly high, but it is lower at the event X. Note that by lowering these thresholds or even eliminating the event X from the winning pattern, we would get more results.

The outcome of this new search for the 2009 French Open match between Rafael Nadal and Robin Söderling produced the following results: 11 groundstroke attacking winning sequences by Nadal all have the same pattern. The three right panels from Figure 4 show a few examples of successful groundstroke attacking winning patterns by Nadal. We have not shown the completion of the attack (i.e., the sequel) in order to make the figures clearer.

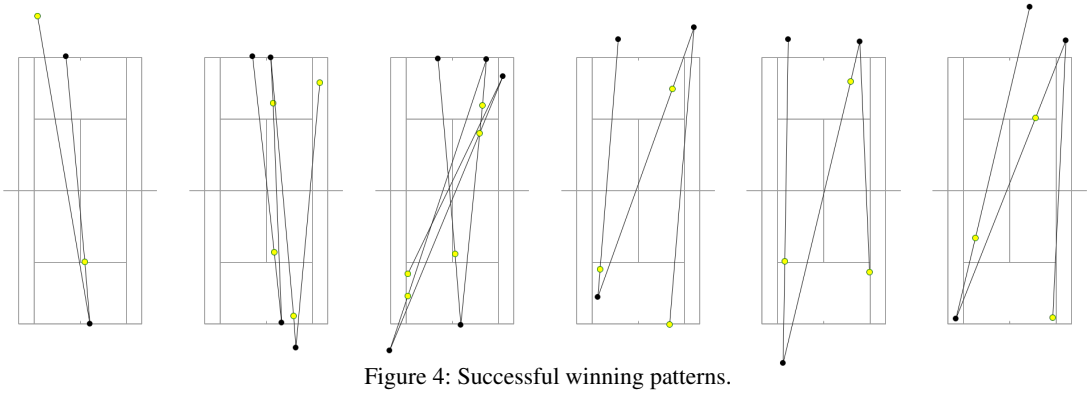

Figure 4: Successful winning patterns.

7 CONCLUSIONS

The use of multivariate sequential data mining along with a comprehensive set of spatiotemporal attributes has proved to be an effective approach in order to discover successful tennis strategies within a tennis match. To this purpose, we have introduced the concepts of event thresholds, rally similarities and pattern masks so that any winning pattern can be defined and mined. These patterns consist of a prequel and a sequel, that are characterized by a pattern mask and an automaton (that accepts unbalancing events), respectively. Results demonstrate that this framework can be of help for the analysis of tennis matches.

However, other interesting problems remain unsolved: the identification of frequent rallies, the possible characterization of a tennis player based on his/her rallies, the discovery of unforced-error and losing patterns, and the effect of the score in the game.These will be analyzed in subsequent papers.

REFERENCES

Calvo, C., Micarelli, A., and Sangineto, E. (2002). Automatic annotation of tennis video sequences. In *DAGM-Symposium*, pages 540–547.

Christmas, W., Kostin, A., Yan, F., Kolonias, I., and Kittler, J. (2005). A system for the automatic annotation of tennis matches. In *4th Int. Workshop on Content based Multimedia Indexing (CBMI)*.

Chu, W.-T. and Tsai, W.-H. (2009). Modeling spatiotemporal relationships between moving objects for event tactics analysis in tennis videos. *Multimedia Tools and Applications*.

Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge University Press, 2nd edition.

Hayet, J., Piater, J., and Verly, J. (2005). Fast 2D model-to-image registration using vanishing points for sports video analysis. In *IEEE Int. Conf. on Image Processing 2005 (ICIP'05)*, pages 417–420.

International Tennis Federation (2010). Rules of tennis. retrieved May 28, 2010, `http://www.itftennis.com/technical/rules/`.

Lames, M. (2006). Modelling the interaction in game sports — relative phase and moving correlations. *Journal of Sports Science and Medicine*, 5:556–560.

Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., and Hsu, M. (2001). Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *17th Int. Conf. on Data Engineering (ICDE'01)*, pages 215–224.

Schroeder, B., Hansen, F., and Schommer, C. (2005). A methodology for pattern discovery in tennis rallys using the adaptative framework ANIMA. In *Second International Workshop on Knowledge Discovery from Data Streams (IWKDDS)*.

Sudhir, G., Lee, J., and Jain, A. (1998). Automatic classification of tennis video for high-level content-based retrieval. In *IEEE Int. Workshop on Content Based Access of Image and Video Databases*, pages 81–90.

Tan, P.-N., Steinbach, M., and Kumar, V. (2005). *Introduction to Data Mining*. Addison-Wesley.

United States Tennis Association (USTA) (1996). *Tennis Tactics — Winning Patterns of Play*. Human Kinetics.

Wang, J. and Parameswaran, N. (2005). Analyzing tennis tactics from broadcast tennis video clips. In *11th Int. Multimedia Modelling Conf.*, pages 102–106.

Wang, P., Cai, R., and Yang, S.-Q. (2005). A tennis video indexing approach through pattern discovery in interactive process. In *Advances in Multimedia Information Processing (PCM)*, pages 49–56. LNCS 3331.

Zhu, G., Huang, Q., Xu, C., Yui, Y., Jiang, S., Gao, W., and Yao, H. (2007). Trajectory based event tactics analysis in broadcast sports video. In *15th Int. Conf. on Multimedia*, pages 58–67.

Zhu, G., Xu, C., Huang, Q., Gao, W., and Xing, L. (2006). Player action recognition in broadcast tennis video with applications to semantic analysis of sports game. In *14th Annual ACM Int. Conf. on Multimedia*, pages 431–440.