Inferring the Structure of a Tennis Game using Audio Information

Qiang Huang, Member, IEEE, and Stephen Cox, Senior Member, IEEE

Abstract—We describe a novel framework for inferring the low-level structure of a sports game (tennis) using only the information available on the audio track of a video recording of the game. Our goal is to segment the games into a sequence of points, the natural unit for describing a tennis match. The framework is hierarchical, consisting of, at the lowest level, identification of audio events, followed by “match” (i.e., semantic) events, and at the highest level, game points. Different techniques that are appropriate to the characteristics of each of these events are used to detect them, and these techniques are coupled in a probabilistic framework. The techniques consist of Gaussian mixture models and a hierarchical language model to detect sequences of audio events, a maximum entropy Markov model to infer “match” events from these audio events, and multigrams to infer the segmentation of a sequence of match events into sequences of points in a tennis game. Our results are promising, giving an F-score for the final detection of points of > 0.7.

Index Terms—Audio characterization, classification, and categorization, hierarchical language model, maximum entropy Markov model, multigram model.

I. INTRODUCTION

Our ambitious long-term goal is to understand multimodal interaction between humans, and we use a sports game, tennis, as a starting-point. In tennis, the goals of the interaction are clearly defined and the interaction is subject to clear rules. As such, the game can be effectively analysed in terms of sequences of “events”. The work described here focuses on the retrieval of these sequences from audio information, and on inferring the basic low-level structure of the game from them. At a later stage, our intention is to integrate the techniques described in this paper with similar techniques developed for video signals. We expect that the two media streams will provide complementary and synergistic information, but for present purposes, we focus on use of the audio sound track alone.

There have recently been several studies in the field of the content analysis of sports video. Multimodal analysis systems that are able to identify automatically events occurring within sporting games, describe their contents, explore their dependencies, and summarize logical relations between them have been developed [1], [2], [3], [4], [5], [6]. These systems are now being applied in the technologies of computer games, multimedia information retrieval and human-computer interaction.

These approaches utilize both video and audio signals to identify significant events. The visual signal is clearly a major source of information about events and interactions in a game. However, the accompanying audio track, even if it is sparse or noisy, can be a rich source of information about the progress of a game. In [7], it was shown that using purely visual features does not yield very high performance in event recognition, and our work will demonstrate the advantages of audio information in efficiently and effectively detecting events in the domain of sports video, especially when the task moves from detection to understanding.

In this work, we have moved beyond low-level information classification or clustering of features to inferring the low-level structure of the game, a task which we believe could also be accomplished by an intelligent human who had no previous exposure to the game of tennis. The process of segmenting the stream of events present in the game is somewhat akin to a child learning how to segment a stream of speech into a sequence of words: the child notices that some phonetic sequences tend to re-occur, and that there are patterns of co-occurrence across different sequences. In this spirit, we use a variable-length multigram model (VLMM) to search for regular occurring patterns of match events that constitute a “point” in a tennis match, and we also learn correlations between these events (we use a “point” as the natural unit of segmentation in a tennis game). Our results indicate that this is a promising approach to learning the kinds of sequential structures that occur in sports games.

We use the soundtracks of DVDs of tennis games in this work, for several reasons.

1) Tennis is a highly-structured game in which the basic short-term unit (a “point”) is characterized by sequences of events that have some clear audio correlates (e.g. ball hits, crowd applause, umpire announcements, line judges’ shouts, etc.) and by contrast with more “continuous” sports, such as soccer, points are punctuated by periods of non-activity, making them easier to segment in an audio or video signal.

2) The long-term structure of tennis is straightforward, being based on a player exceeding a threshold in the difference of the number of points won to win a game, exceeding a threshold in the difference of the number of games won to win a set, and exceeding a threshold in the difference of the number of sets won to win the match (there is some slight variation in these thresholds depending on the exact state of the game, but it need not concern us overly here). This hierarchical structure, in which the result is based on one player accruing
sufficient points over a long period to win, makes analysis of a complete game more robust than a game such as soccer, in which a single missed event can lead to an error in the final result.  

3) Crucially, tennis game recordings are rich in both video and audio information that can be used to analyze and then understand the progress of the game.

The paper is structured as follows. In Section II, we review previous work in this field, and related fields. Section III introduces the data used in the experiments documented in this paper. In Section IV, we describe our overall theoretical framework and develop its underlying mathematics. Sections V, VI, and VII deal respectively with the application of this framework to audio event detection, prediction of “match events” and inference of low-level game structure. Section VIII describes the experimental set-up and presents the main results, and Section IX analyses and summarises main results in the paper and discusses future work in this area.

II. Previous Related Work

Previous work on sports video analysis has involved different types of sports games, such as soccer[8], [9], [10], tennis [11], [12], [13], [6], [14], baseball[15], [3], [16], [17], and cricket [18], etc. From our point of view, these studies can be classified into two classes: low-level feature based event classification and high-level semantic analysis.

A. Low-level Feature Based Event Detection

For low-level feature based event detection, the goal is to use the audio and visual features, separately or jointly, to detect related events or scenes in the sports video.

Some previous work [14], [8], [19], [20], [6], [13], [21] has employed only visual information. [6], [14] focus on tennis game analysis by tracking players, locating the tennis court, and classifying the match events. [13] builds event HMMs with binary classification according to the players’ positions in the court to detect different types of racket hit. [19] attempts to rank highlights in racket games by analysing human behavior. [8] built a classification tree with visual features to detect match events. [21] utilized a segmental HMM for view-based soccer analysis.

Some work [22], [23], [24], [25], [26], [27] has put more focus on audio information. [22] employed an unsupervised technique using a spectral clustering algorithm to discover the audio elements. [23] proposed a discriminative feature set for acoustic event detection according to approximated Bayesian accuracy. [24] utilized rule-based classification according to audio type and speaker identity. [25] built a two-stage classifier for vocal and non-vocal events classification and then for normal and “excited” events classification. [26] employed a Bayesian network to combine the context information for audio stream segment. [27] divided the audio stream into short sequences, and then classified them into three classes: speaker, crowd and referee whistle.

Some work has also combined visual and audio information [28], [11], [3], [29], an approach that can reduce the impact of overlapping interference on the event detection performance.

B. High-level Semantic Analysis

There has also been considerable activity in moving from detection of low-level features to the extraction of semantic information, as a first step to understanding games.

Using the assumption that auditory scenes with similar semantics usually contain similar sets of typical key audio elements, [32] attempted to group auditory scenes according to the affinity between the two adjacent key audio elements to detect their semantic content coherence. Rule based approaches for discovery of semantics have also been employed in [33], [34], [8], [35], [33] used low-level audio/visual features for video summarization, while [34] attempted to build a unified framework using a rule-based method. In a similar way, [15], [36] describe a “storyline” model dependent on AND-OR graphs, which also builds a tree-like hierarchical structure. [12] treated a class of scene as a single unit and built an HMM for each of them.

Much of this work has focused on signal-processing and pattern classification techniques that enable a particular object, activity, scene or event within a game to be classified. In general, the techniques depend heavily on the fixed structure of graphical models and require prior knowledge of the scene structure, which is mostly acquired in a supervised way. In addition, the fixed structure of such models limits their re-application to a task in a different domain. In order to understand a game, we must move beyond low-level recognition of events and activities themselves to understanding how the events relate to each other and how they form the “grammar” of a sports game: we can term this the game syntax. If we could realise this goal, we would be able to understand different examples of the same game and also understand, with minimal human intervention, different games that were comprised of similar events but with a different syntax (for instance, in our case, badminton or squash). We should also be able to infer the presence of new events in a domain. Achieving this goal would pave the way to more ambitious tasks of understanding interactions.

III. Data and Experimental Baseline

The audio data we used were extracted from DVDs of Wimbledon tennis games played in 2008, two male singles games, Murray vs. Gasquet (M_G) and Federal vs. Nadal (F_N). Table II summarises salient information about the two
tennis games, each of which consists of a number of tracks, a track lasting anywhere between 17 and 60 minutes.

We used three audio tracks (about 70 minutes) from “M_G” for training, and a test set consisting of two parts: seven audio tracks from “M_G” (about 140 minutes) and three audio tracks from “F_N” (about 150 minutes). The amount of training-data available is limited by the time taken for manual annotation.

We use seven types of audio event (AE) in our work, and these are defined in Table I. These audio events include the sound of a ball hit and crowd noise, which are standard in this work, but also some more detailed events, such as umpire’s speech, the electronic “beep”, and line judge shouts. These audio events can be used in different ways to infer the state and progress of the game: for example, the applause, gasps, cheers, roars, etc. of the crowd correlate with the end of rally, and the voice of the chair umpire furnishes us with information about the scores and the long-term progress of the match, which is strongly relevant to the end of a game point. Figure 1 shows a manually annotated segment of the audio waveform from the soundtrack of a tennis game. Two consecutive game points within the stream are marked as GP1 and GP2, and the positions of occurrences of six audio events are marked below the clip (Silence is omitted for clarity).

Audio analysis was standard: the audio sequence was windowed into 30ms-length frames with 20ms overlapping, from which 39-D vectors were generated. These vectors consist of 12 MFCCs, a log energy, and their velocity and acceleration coefficients. Cepstral mean normalization was applied at the track level. Our baseline is frame-based classification, which is accomplished by using a Gaussian mixture model to model each event [37].

### Table I

<table>
<thead>
<tr>
<th>Number</th>
<th>Event Name</th>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Silence</td>
<td>SIL</td>
<td>either silence, or a sound not in one of the next six categories</td>
</tr>
<tr>
<td>2</td>
<td>Umpire</td>
<td>UMP</td>
<td>speech from the chair umpire usually current score or match progress</td>
</tr>
<tr>
<td>3</td>
<td>Commentary</td>
<td>COM</td>
<td>commentators’ speech</td>
</tr>
<tr>
<td>4</td>
<td>Line Judge</td>
<td>LJ</td>
<td>a line judge’s call for a fault or a ball that is “out”</td>
</tr>
<tr>
<td>5</td>
<td>Ball Sound</td>
<td>BS</td>
<td>the sound generated by the ball striking a racquet, the ground or the net</td>
</tr>
<tr>
<td>6</td>
<td>Crowd Noise</td>
<td>CN</td>
<td>the crowd’s applause, gasps, roars, etc.</td>
</tr>
<tr>
<td>7</td>
<td>Beep</td>
<td>BP</td>
<td>the electronic sound generated when the ball is out or touches the net during a serve</td>
</tr>
</tbody>
</table>

### Table II

<table>
<thead>
<tr>
<th>Description</th>
<th>Male singles game 1 (“M_G”)</th>
<th>Male singles game 2 (“F_N”)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total duration (mins)</td>
<td>340</td>
<td>320</td>
</tr>
<tr>
<td># tracks</td>
<td>17</td>
<td>9</td>
</tr>
<tr>
<td>Training set</td>
<td>Sound track 1 – 3 (70 mins)</td>
<td>Sound track 2 – 4 (150 mins)</td>
</tr>
<tr>
<td>Test set</td>
<td>Sound track 4 – 10 (140 mins)</td>
<td></td>
</tr>
</tbody>
</table>

### IV. Theoretical Framework

As we have indicated in Section I, we can liken the process of learning the low-level syntax of a tennis game from a continuous audio signal to that of learning words from a continuous speech signal. Just as the word is a natural semantic segment in a speech signal, a “game point” (which we shall also refer to as just a “point”) is a natural segment within the stream of audio associated with a tennis game. In the same way that words are composed of a sequence of different phonemes, points in a tennis game are composed of a number of events. We describe the nature of these events later in this section.

The machine-learning approach to recognition of a sequence of words from a stream of continuous speech is to find

\[ W^* = \arg \max_W \Pr(W|O), \]

where \( O \) indicates the sequence of acoustic observations and \( W \) is the set of possible word candidate sequences.

Recognition is performed by expanding a word network into sub-word unit sequences by dictionary lookup, where the sub-word units \( S \) can be fixed-length or variable-length phoneme
sequences, so that

\[ Pr(W|O) = \sum_S Pr(W|S) Pr(S|O). \]  \quad (2)

In equation 2, the second term is estimated by a sub-word unit classifier, while the first term can be estimated from a pronunciation model using Bayes theorem and a word segmentation algorithm.

In the tennis-game paradigm, our “words” are sequences of game-points (GPs) and our “sub-word sequences” are sequences of audio events (AEs) within a point, so we could re-write equation 2 as:

\[ P(GP|O) = \sum_{AE} Pr(GP|AE) Pr(AE|O) \]  \quad (3)

However, the correspondence between sequences of audio events and points is not clear-cut. In terms of our analogy, to say that points are made from sequences of audio events would be like saying that words are formed from sequences of acoustic units, which are physical and measurable, rather than from sequences of abstract units, i.e. phonemes. We hence introduce another level of symbols between the audio events and the points, which bridge the divide between the acoustic and the abstract: “match events” (MEs). These correspond closely to interpretations of events that occur in the game, and in machine-learning terms, could be thought of as a set of “latent” variables. The set of nine match events that were defined and their corresponding audio events, is shown in Table III.

Hence we can further update equation 3 by inserting ME:

\[ Pr(GP|O) = \sum_{AE, ME} Pr(GP, ME, AE|O) \]  \quad (4)

Here, \{AE\} and \{ME\} denote the set of all possible candidates of audio events and match events. Maximization of the right hand side of equation 4 leads to

\[ (GP^*, ME^*, AE^*) = \arg \max_{(GP, ME, AE)} Pr(GP, ME, AE|O) \]  \quad (5)

where \(GP^*\), \(ME^*\) and \(AE^*\) represent the most likely sequences of points, match events and audio events, respectively. Using Bayes theorem, the right side of equation 5 can be re-written as:

\[ Pr(GP, ME, AE|O) \approx Pr(O|AE, ME, GP) \cdot Pr(GP|AE, ME) \cdot Pr(ME|AE) \cdot Pr(AE) \]  \quad (6)

After taking various kinds of dependencies between these terms into account, we can further simplify equation 6 to

\[ Pr(GP, ME, AE|O) \approx Pr(O|AE) \cdot Pr(AE) \cdot Pr(ME|AE) \cdot Pr(GP|ME) \]  \quad (7)

After this simplification, we can view the right hand side of equation 7 as follows:

- The probability of a sequence of audio events is given by the first two terms
V. Audio Event Detection

In this section, we develop a hierarchical language model (HLM) of the audio events to improve detection performance, and show how we can further improve performance by combining duration and pitch models with the HLM.

A. Construction of a Hierarchical Language Model

The most likely sequence of audio events, \( A E^* \), given a sequence of acoustic features, \( O \), is obtained as

\[
AE^* = \arg \max_{AE} \Pr( AE | O ).
\]

Using Bayes’ theorem:

\[
AE^* = \arg \max_{AE} \Pr( O | AE ) \Pr( AE ).
\]

We now introduce an extra “latent” variable \( F \), so that we can re-write equation 9 as

\[
AE^* = \arg \max_{AE} \sum_F \Pr( O | F ) \Pr( F | AE ) \Pr( AE ) (10)
\]

\[
AE^* = \arg \max_{AE} \sum_F \Pr( O | F ) \Pr( AE | F ) \Pr( F ) (11)
\]

In equation 11, \( F \) represents a sequence of audio event labels, labelling the frames that comprise a track, and \( \sum_F \) is read as “sum over all possible label sequences”. A label for a frame has the value \( \{ 1, 2, \ldots N_{AE} \} \), where \( N_{AE} \) is the number of distinct audio event classes: the label is the most likely audio event associated with the frame, and is estimated from a Gaussian mixture model (GMM) of each audio event.

The three terms in equation 11 can be computed as follows:

1. The term \( \Pr( O | F ) \) is computed from acoustic models of the audio events: we used GMMs, which are trained using manually labelled data. We assume independence of frames: this obviously false assumption is corrected during the later stages of processing. Hence

\[
\Pr( O | F ) = \prod_t \Pr( O_t | F_t ) . (12)
\]

2. The term \( \Pr( AE | F ) \) can be modelled as depending on the history of the audio events (approximated here by a bigram) and the probability of an event given a certain frame labelling:

\[
\Pr( AE | F ) \approx \prod_t \Pr( AE_t | AE_{t-1} ) \Pr( AE_t | F_t )
\]

where \( \Pr( AE_t | F_t ) \approx \Pr( AE_t | f_t, f_{t-1} f_{t-2} ) \). (13)

Here, \( AE_t \) denotes the audio event \( AE \) that occurs at time \( t \). \( \Pr( AE_t | AE_{t-1} ) \) corresponds to a bigram “language model” of audio events, which is estimated from the labelled training data. Estimation of the trigram term \( \Pr( AE_t | F ) = \Pr( AE_t | f_t, f_{t-1} f_{t-2} ) \) was performed using standard linear interpolation techniques, and the estimates were then smoothed.
We use Maximum Entropy techniques.

3. The probability of the sequence of labels \( F \) can be estimated as if it were a sequence of words or phones using a trigram model:

\[
Pr(F) = \prod_{t} Pr(f_t | f_{t-1} f_{t-2}). \tag{15}
\]

Practically, it is not possible to use a model of frame events that is derived from the manual labelling of the frames. In such a model, \( Pr(AE_t = AE_{t-1} | AE_{t-2} = AE_t) \approx 1 \), because an event lasts for many frames and all the frames within an event have the same label. We therefore learn a model that is based on the labelling of the training-set frames by the acoustic models. Although this model is errorful, it is a valuable source of information.

We assume that equation 11 can be approximated by the most likely sequence over all \( F \) (as is standard in ASR), in which case we can re-write it as:

\[
AE^* = \arg \max_{AE} \{Pr(AE_t | AE_{t-1}) \} \tag{16}
\]

Although the probability of the current state \( s' \) and the current observation \( o \) gives

\[
\max_{F} \{Pr(O | F) Pr(AE_t | F) Pr(F)\} \]

The top row shows that the duration distributions of the three audio events are quite different: the duration of the umpire’s voice ranges from 280ms to 750ms, while most of the commentator’s segments last for more than 700ms. The impulsive sound of a racquet striking a ball has a mean duration of only about 90ms. Pitch information is a good way of distinguishing between speech and non-speech events. If a pitch estimation algorithm is run on the audio events, the umpire’s voice and commentators’ voices show that voicing is often detected, and the distributions are similar, whereas the “ball hit” histogram shows very little voicing is detected, although there are a small number of voiced frames caused by the players grunting!

To integrate this information, we first set empirically derived minimum and maximum thresholds of duration and pitch for each audio event. During traceback in Viterbi decoding, the duration and the distribution of each detected audio event is noted. If the label of the decoded audio event is outside the permitted limits set by the thresholds mentioned above, it is changed to the next best event match in decoding, and this process is continued until an event that does not fall outside the bounds of its threshold is found.

**VI. Prediction of Match Event**

The second part of our framework is to infer the most likely sequence of match states given the detected audio events and the current match state, which is done by utilizing a maximum entropy Markov model (MEMM, [39]).

The MEMM can be viewed as a discriminative model in which the current state not only depends on the current observations, but also the previous state. So, in this model, the transition and observation parameters of the HMM are replaced by a single function \( Pr(s | s', o) = Pr_{s'}(s | o) \) that gives the probability of the current state \( s \) of the model given the previous state \( s' \) and the current observation \( o \).

Figure 4 depicts a tennis point as a (pseudo) finite-state automaton. Circles represent match events, and arcs show audio events associated with the transition between these match events (two match events, “Normal” and “Challenge” events).
are not shown here for reasons of clarity). The current match event depends on both the current observed audio event and the previous match event, and hence the MEMM is a suitable model for the prediction of match events. For instance, given the observed audio event sequence $o = AE_1 = \text{“ball sound”}$, $AE_2 = \text{“line judge”}$, $AE_3 = \text{“ball sound”}$, $AE_4 = \text{“crowd noise”}$, $AE_5 = \text{“umpire”}$, we can infer the match event sequence $ME_1 = \text{“Serve 1st”}$, $ME_2 = \text{“Out”}$, $ME_3 = \text{“Serve 2nd”}$, $ME_4 = \text{“Applause”}$, $ME_5 = \text{“Score or Announcement”}$.

An MEMM can be thought of as a finite state automaton with stochastic state transitions and associated observations. The observations are derived from the detected sequence of audio events in order to infer the match state sequence. Hence $s'$ and $s$ in $Pr(s'|s,o)$ respectively indicate the previous and current match event, while $o$ represents the current observation. In our case, the current observation consists of several features (details as shown below), and each feature $f_i(o,s)$ gives a function of two arguments, a current observation $o$ and a possible current state $s$.

- the identity of the current audio event
- the identity of the last audio event
- the identity of the last two audio events
- the identity of the last three audio events
- whether the duration of the current audio event is greater than a threshold (0.5s)
- whether the time gap between any two adjacent audio events is greater than a threshold (3s)

Using the known state of the game at any time and the values of the above features, we can train the MEMM. We compute the maximum entropy distribution using

$$Pr_{s'}(s|o) = \frac{1}{Z(o,s')} \exp(\sum \lambda_i f_i(o,s))$$

where $s,s'$ are different match states, $o$ is the current observation consisting of those features, and the $\lambda_i$ are parameters to be learned. $Z(o,s')$ is a normalizing factor that makes the distribution sum to one across all linked states. In the training step, generalized iterative scaling is employed to optimize the model parameters. In the test step, the Viterbi algorithm is used to search the optimal match event sequence. The reader is referred to [39] for full details of the MEMM.

VII. ACQUISITION OF GAME WORDS

In this section, we make use of the multigram model proposed by Deligne et al. [40], [41] to segment the sequence of match-states found in the previous section into a sequence of game points. The multigram model “provides a statistical tool to retrieve sequential variable-length regularities within streams of data” [41], and has been shown to be effective in modelling both variable length lexical units in language [40] and variable length acoustic units in speech [41]. We extend the technique by adding some contextual information (inter-event timing) into the model within a probabilistic framework.

A. Construction of Multi-gram Model

Let $E = e(1) \cdots e(t) \cdots e(T)$ denote a stream of $T$ events, and $D$ denote a possible segmentation of $E$ into $q$ sequences of events: $s(1) \cdots s(q)$. The likelihood of the stream of events $E$ associated with segmentation $D$ is computed as:

$$L(E, D) = \prod_{t=1}^T Pr(s(t))$$

Here, our aim is to find the most likely segmentation of $E$

$$L^*(E) = \max_{D \in \{D\}} L(E, D)$$

where $\{D\}$ is the set of all possible segmentations of $E$ into a sequence of events. The multi-gram model is fully defined by a set of parameters $\Theta$ consisting of the probability of each event sequence $s_i \in V$, where $V = \{s_1, \cdots, s_m\}$, a dictionary containing all the sequences of events.

To compute the set of parameters $\Theta$ from a training corpus $E$, an iterative Maximum Likelihood (ML) is used through an Expectation Maximization (EM) algorithm:

$$L(D|E, \Theta(k)) = \frac{\sum_{s \in D(E)} c(s_i, D) \times L(D|E, \Theta(k))}{\sum_{D \in \{D\}} c(D) \times L(D|E, \Theta(k))}$$

where $c(s_i, D)$ is the number of occurrences of sequence $s_i$ in segmentation $D$; $c(D)$ is the total number of sequences in $D$; and $L(D|W, \Theta(k))$ is the conditional likelihood of the segmentation $D$ given $E$ at iteration $k$.

The estimation of the model parameters can be computed through an iterative forward-backward procedure [41]. As in the estimation of HMM parameters, it relies on the estimation of a forward variable $\alpha$ and a backward variable $\beta$, which are defined as the likelihood of the partial observed stream of events $E(1)$ and $E(T)$, respectively.

$$\alpha(t) = L(E(1))$$

$$\beta(t) = L(E(T))$$

The variables $\alpha(t)$ and $\beta(t)$ can be recursively calculated. Figure 5 shows the computation of $\alpha(t)$ and $\beta(t)$ (Training) followed by segmentation (Decoding) using the Viterbi algorithm [42].

B. Incorporating Inter-event Timing Information

It seems likely that the time period between two match events would be a good indicator of a segmentation point for game units, since there is either a short or a medium length break between games. Figure 6 shows the distribution of times between two events that occur during a game (left curve), and for the situation where the first event is the final event of a game unit and the second event is the first event in the next game unit i.e. across a game (right curve). Note that the x-axis scale of log(t + 1) has been chosen for the convenience of plotting small values of the time interval, $t$. The figure indicates that although there is some overlap of the tails of “within” and “across” units in the centre of the figure, the
Recall that there were two test-sets, the first from the same
empirically, and is set to 15 in this paper. However, we found
in the range 5–20 gave little variation (less than
1%) in segmentation performance.

Table I shows the data that was used for training and testing.

![Graph showing probability distributions of the time interval between two match events](image-url)

F-score, defined as:
\[ F\text{-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \] (24)

For event detection, a Correctly Detected Event is one that occurs in an approximately correct region. To determine these regions, we compute the mean position in time (MT) of each manually-labelled event using its start time (ST) and end time (ET):
\[ MT_{\text{event}} = \frac{ST_{\text{event}} + ET_{\text{event}}}{2} \] (27)

If \( MT_{\text{event}} \) is located within the time range of a detected event with the same labelling, then the detected event is viewed as a correct detection.

For match events prediction, the precision and recall are defined as:
\[ \text{Precision} = \frac{\# \text{Correctly Predicted Events}}{\# \text{Detected Events}} \] (28)
\[ \text{Recall} = \frac{\# \text{Correctly Predicted Events}}{\# \text{True Events}} \] (29)

where a Correctly Predicted Event is one whose predicted ME is the same as the manual annotation.

To assess the performance of the segmentation of a sequence of match-events into a sequence of points, we use the identity of the first and the last match event in a point as a reference. These are most commonly (but not always) "serve" and "score". If the first and last events in the manual segmentation are the same as the first and last events in the detected segmentation, this is defined as a correctly segmented point.
Hence we use an F-score in which

\[
\text{Precision} = \frac{\text{# Correctly segmented points}}{\text{# Points segmented}} \quad (30)
\]

\[
\text{Recall} = \frac{\text{# Correctly segmented points}}{\text{# Points in Data}} \quad (31)
\]

To clarify our evaluation scheme, an example segmentation is given in figure 7. Figure 7 shows the waveform of an example audio clip of a point (top pane), its manual labelling and segmentation (centre pane), and the automatic labelling and segmentation (bottom pane). The manual annotations consist of the manual segmentation point and the label of each audio event, and the related label of a match event corresponding to the audio event. For example, “BS\_1\_S\_Serve\_1st” indicates that the first audio event is the sound of a ball strike (BS) and the corresponding match event is Serve (the first serve), whilst “BS\_1\_E” is its end label of this segment. We can see an insertion error “BS\_7\_S\_Rally” has been generated by the automatic system, and it is clear that there are some differences in the placements of segments obtained using the automatic and manual annotations.

To view clearly these differences and to appreciate how our previous definition of a correct region works for audio event detection, figure 8 magnifies the section of the audio clip surrounded with a frame in figure 7. The upper line shows the manual segmentation points, the bottom line the automatic segmentation points. The two dashed lines indicate the mean time of the two manual segments shown.

\[
\text{IX. RESULTS AND ANALYSIS}
\]

\[
\text{A. Performances of Audio Event Detection}
\]

Figure 9 shows the performances of detecting the audio events on the training set and the two test sets (N.B. Although the results are shown as points joined by lines, because each point corresponds to a different audio track, the line does not represent a “trend”; it is there for the convenience of comparing different techniques). It shows that a GMM on its own is not adequate to characterize the events and the use of the hierarchical language model (HLM) gives considerable improvement. Although the trained parameters are obtained from the game “M\_G”, when applying our technique to another game, “F\_N”, we still obtain reasonable performance even though the speakers (umpire and line judges) and the environmental acoustic properties are different. Moreover, after taking the duration and pitch information into account, further improvements are obtained.

The most likely reason for the poor performance of GMMs is the fact that there is considerable overlap and hence interference between audio events, which corrupts the models, and leads especially to insertions of extra events in the results. This
is considerably corrected by the use of contextual information in the HLM, which uses a low-level frame-based language model connected to a higher-level model, putting strong constraints on the detection procedure. Duration information is mostly useful for correcting spurious “spike” events which are mis-recognised as the sound of ball hitting, while the use of pitch information can prevent some interfering noise from being classified as human voices.

B. Performances of Match Event Prediction

Figures 10(a)–10(b) show the prediction accuracy for match events on the M_G and F_N games respectively as the MEMM is iterated. The blue curve shows the prediction performance using the manual transcription of audio events, the red curve using detected audio events. The blue curve may be viewed as the upper-bound for our results. In our experiments, the prediction performances on the game M_G shown in figure 10(a) reaches 0.94 using manually annotated audio events and 0.77 using automatically detected events. For F_N, the corresponding figures are 0.89 and 0.75. These differences may be explained by the fact that some data from M_G is used in the training-set and that the characteristics of the F_N game are different from M_G: for instance, M_G contains more applause and crowd roars. In addition, the acoustic characteristics of the chair umpire are different: the umpires are different people and the recording system is different. Although these factors affect the audio event detection, we still obtain good performance on detecting the match events sequence.

C. Performance of Segmentation into Points

In the multigram experiments, we used a variable-length model for segmentation, and we show here results for maximum length (ML) sequences of four, five, six and seven match events.

Figure 11 shows the (fully automatic) segmentation performance as the MEMM is iterated, measured as described in section VIII, on the predicted sequence of match events using the manually annotated and the detected audio event
sequence of two games for different values of the maximum length (ML).

The figures suggest that a maximum length of six is about optimal in all cases. A point in tennis usually consists of four basic elements: Serve, Rally, Score, and Applause, but some other elements, such as Let or Line judge’s shout, also occur frequently. This increases the average length of the basic game unit. In the games of Murray vs. Gasquet (M_G) and Federer vs. Nadal (F_N), the average length of a match event sequence in a game is 5.6 and 5.2, respectively, which corresponds with a maximum match event length of six.

We can also see that performance on the game F_N is better than that on M_G, whether using manual (77% vs. 64%) or detected (71% vs. 54%) annotations, although the prediction correction rate on the game of M_G shown in figure 10 is higher. This may be for two reasons: firstly, the structure of the game of M_G is more complex than that of F_N, because the former contains many crowd roars, which affect the audio segmentation. Secondly, the inter-event time information plays a slightly more important role in the F_N game, giving more useful hints of the location of the segmentation points than in M_G.

The effect of using the inter-event duration is shown in figure 12. It shows that the use of inter-event duration gives a marginal improvement after several iterations.

Finally, figure 13 shows the probability distributions over the match event symbols of the first and final match events. In the figure, the results are obtained using three different types of data: manual annotation of match events, automatically predicted match events using manual annotation of audio events, and detected audio events. Figure 13(a) shows the match event “Serve” is dominant at the start position of the game segments, whilst figure 13(b) shows the match event “Score” is the most likely candidate for the final match event. These results indicate that the multigram technique does a good job at segmenting the match events into points.

X. DISCUSSION AND FUTURE WORK

Our motivation in this work was to develop techniques for learning high-level game information from low-level signals. In doing so, we have attempted to view the problem as a bottom-up learning procedure that is akin to the way that a child learns words from continuous speech signals.

In Section IV, we described a theoretical framework that integrated three processes: the detection of audio events, the prediction of match events, and the acquisition game units. In Section V, we developed a hierarchical language model by combining the low-level features with the high-level constraints. The construction of the frame-based language model, the event-based language model, and the connection between them, generate significant improvements. In addition, the use of extra information, namely the duration of audio events and pitch, brings more benefit in audio event detection. Section VI showed how we predict the match events given the detected audio events generated using the approach described in Section V. We employed the maximum entropy Markov model to find the mappings between the “observed information” (audio events) and the “semantic information” (match events). Section VII introduced an approach that was motivated by the idea of finding the boundary of word from the continuous letter/phoneme streams. A variable-length multi-gram model was employed in this section to look for the most likely location of the boundary in the predicted sequence of match events. Although there are some deletions, substitutions, and insertions, it still produced quite good performance by effectively locating the boundary. Combining this technique with the inter-event duration information, we obtain further improvements on the acquisition of high-level game information.

One part of the process that is not yet automatic is the ontology: the definitions of audio events and match events were done with knowledge of the game, and this limits the usefulness of the approach. We are now looking at ways of inferring these automatically. We are also looking at ways of using bi-directional information flow i.e. from top to bottom as
Fig. 11. Segmentation performances (as the MEMM is iterated) on the sequence of predicted match events, using the manual/detected audio events sequence for the games \( M_G \) and \( F_N \). (a) Segmentation on \( M_G \) (manually annotated match events). (b) Segmentation on \( M_G \) (automatically detected match events). (c) Segmentation on \( F_N \) (manually annotated match events). (d) Segmentation on \( F_N \) (automatically detected match events).

well as bottom to top, so that, for example, a putative match-event sequence can influence audio event detection, which in turn can re-define the match-event sequence. Another valuable source of information for a neophyte learning how the game of tennis is structured is the content of the umpire’s speech, and we intend to investigate the possibility of using ASR on this signal. Finally, we are going to integrate information derived from the visual signal with our audio information: the addition of this complementary information should greatly improve the power of our techniques and enable us to produce a more detailed and accurate description of the game.

ACKNOWLEDGMENT

This work was funded by the UK Engineering and Physical Sciences Research Council as grant number EP/F069626/1.

REFERENCES


Fig. 12. Segmentation performances (as the MEMM is iterated) on the sequence of predicted match events, using the manual/detected audio events sequence for the games M_G and F_N, showing the effect of inter-event timing. (a) Segmentation on M_G. (b) Segmentation on F_N.

Fig. 13. Probability distribution over match event symbols of the first and final match events on the two games (M_G and F_N). (a) Segmentation on M_G. (b) Segmentation on F_N.


Stephen Cox (M’00–SM’08) trained firstly as a physicist and then as an electronic engineer, and began his career at the UK Government Communications Centre developing signal-processing algorithms. He then joined British Telecom’s research laboratories to work on speech recognition, and spent two years at the speech research unit of the Royal Signals and Radar Establishment (now Qinetiq) at Malvern, where he researched into adaptation of speech recognition algorithms to new speakers. He returned to BT to lead a team of researchers developing speech recognition algorithms for use on the UK telephone network. He joined the School of Computing Sciences at UEA as a lecturer in 1991 and was appointed professor in 2003. His research interests include speech recognition, music processing, audio identification and automatic lip-reading and he is the author and co-author of over 100 publications in these fields.

He was an invited consultant at ATT Bell Labs, New Jersey in 1994, a visiting scientist at Nuance Communications Inc., CA, in 2000, and an invited researcher at Apple Inc., CA, in 2010. He has acted as a consultant and reviewer for several national governments as well as the European Commission, and also consults for industry. He is a senior member of the Institute of Electrical and Electronic Engineers and an ex committee member of the IEEE Speech and Language Technical Committee.

Qiang Huang (M’05) received the B.S. and M.S. degrees in information engineering from XiDian University, Xi’an China, in 1996 and 1999, respectively. His early research was on the development of multimedia information transmission over wireless local area network. In 2005, Qiang obtained Ph.D. in computer science from the University of East Anglia (UEA), Norwich, U.K.

From 2006 to 2008, he worked at Knowledge Media Institute, Milton Keynes, U.K., where his research was on information retrieval. In 2009, he returned to UEA and is currently working as a senior research associate at School of Computing Sciences, UEA. His research interests are in multimodal signal processing, natural language processing, speech recognition, and information retrieval.