Detection of Ball Hits in a Tennis Game Using Audio and Visual Information

Qiang Huang*, Stephen Cox*, Xiangzeng Zhou†, Lei Xie†
* {h.qiang, s.j.cox}@uea.ac.uk
University of East Anglia, Norwich, UK
† lxie@nwpu.edu.cn, xzzhou@nwpu-aslp.org
Northwestern Polytechnical University, Xi'an, China

***Abstract*—In this paper we describe a framework to improve the detection of ball hit events in tennis games by combining audio and visual information. Detection of the presence and timing of these events is crucial for the understanding of the game. However, neither modality on its own gives satisfactory results: audio information is often corrupted by noise and also suffers from acoustic mismatch between the training and test data, and visual information is corrupted by complex backgrounds, camera calibration, and the presence of multiple moving objects. Our approach is to first attempt to track the ball visually and hence estimate a sequence of candidate positions for the ball, and to then locate putative ball hits by analysing the ball's position in this trajectory. To handle the severe interferences caused by false ball candidates, we smooth the trajectory by using locally weighted linear regression and removing the frames where there are no candidates. We use Gaussian mixture models to generate estimates of the times of hits using the audio information, and then integrate these two sources of information in a probabilistic framework. Testing our approach on three complete tennis games shows significant improvements in detection over a range of conditions when compared with using a single modality.**

I. Introduction

Automatic analysis of sports games is an area that is attracting considerable research attention, not only because of commercial applications, but because sports games contain rich audio and visual information within a well-organised structure and hence are an excellent testing-ground for developing systems that "understand" interactions. In analysing a sports game, detection and recognition of sequences of key events is important in many applications, such as video retrieval of events [1], [2], video summarization and object tracking [3], [4] and analysis of player tactics [5]. One of the key events in a tennis game is the ball-hit: reliable detection of this event is essential.

Our previous work on ball-hit detection focused on the use of audio information, both low-level acoustic features [7] and high-level contextual information[8]. However, the audio information extracted from the soundtrack of a tennis game is often corruption by crowd noise, players yelps, commentary etc. and there is always some degree of acoustic mismatch between the training and test data.

To reduce the effect caused by audio noise, in [6], we attempted to estimate ball trajectories. Satisfactory acquisition of the ball trajectory requires a high frame rate (at least 50 frames per second) to reduce the problems caused by camera calibration and to reduce blur. This high frame rate is only found in interlaced videos, in which each frame actually consists of two fields, and both fields can be treated as separate frames during processing. However, most sports videos, after compression, are converted into progressive format at a frame rate of 25 frames per second. This lower frame rate creates a larger timing gap between frames, which is manifest as changes of shape and size of the image of the ball, resulting in a high number of false candidates, and hence poor tracking performance. In addition, in [6], direction of velocity change was used, which meant that we were unable to distinguish ball bounces from ball hits. Occlusion of the ball, either by a player or a court line, is also a major problem.

In this work, we improve the approach reported in [6] by

1) reducing false candidates by masking the court line and the players;
2) utilising the Viterbi algorithm to estimate the most likely ball trajectory;
3) connecting trajectory fragments by smoothing using approximate fitting curves.

The remainder of this paper is organised as follows. In section 2, some related work is introduced. Sections 3–6 describe the approaches to ball hit detection using audio and video information, and the theoretical framework for their integration. We list the data used in section seven, and the results are presented in section eight. Finally, conclusions are given in section nine.

II. Related Work

Various researchers have studied the use of audio processing to discover audio events in sports videos. [9], [10], [11], [12], [13], [14]. [9] employed an unsupervised technique using a spectral clustering algorithm to discover the audio elements. [10] proposed a discriminative feature set for acoustic event detection according to approximated Bayesian accuracy. [11] utilized rule-based classification according to audio type and speaker identity. [12] built a two-stage classifier for vocal and non-vocal events classification and then for normal and "excited" events classification. [13] employed a Bayesian network to combine the context information for audio stream segment. [14] divided the audio stream into short sequences, and then classified them into three classes: speaker, crowd and referee whistle. In our own previous work, the dependencies

between audio events were used to enhance the robustness of audio event detection in [8],

The problem of detecting and tracking the ball using visual information been studied by Yu, who developed an enhanced trajectory based ball detection system with camera motion recovery to capture the motion of a soccer ball [15] and analysed the 2D distribution of ball candidates and exploit the characteristic that the ball trajectory presents in a near parabolic curve in video frames [16]. This work used homography projection to form a background template over all frames by computing a Gaussian modal of each pixel. [17] used homography transformation between multiple observed frames to locate ball and players. [18] presented a real-time computer vision system that tracks the motion of a tennis ball in 3D using multiple cameras. [19] employed the Viterbi algorithm and the Kalman filtering to detect and track the ball in a playground. [20] used a bi-directional Viterbi search method to improve the precision of ball tracking. In [21], ball candidates were obtained in each video frame and then Kalman filtering was utilised to generate candidate trajectories from which ball trajectories were selected and extended. In [22], a coarse-to-fine strategy is used to identify ball in a single frame, and then CONDENSATION algorithm was for ball tracking.

Whilst the research cited above has considerably advanced the quality of ball detection and tracking, the presence of interfering noise and acoustic mismatch (in audio tracking) and the difficulties posed by occlusion, blur, colour and shape distortion (in video tracking) means that research on this topic using a single modality will inevitably lead to diminishing returns. Hence we turn to an approach that integrates audio and visual information.

Previous work [23], [15] has investigated using both audio and visual information, but this work used only a section of a single game rather than several complete games, as used here. In addition, the work presented here takes into account the impact of noise interference in the audio track on ball hit detection, which is important, because the audio quality on video soundtracks is often poor. [24] focused more on a coarse scene segmentation rather than fine detection of events, and on processing changes of view, switching between the global view and the close-up view. This kind of visual information has limited application to ball hit detection, because changes of camera view are not often observed during a rally. In [25], the authors treated the sound of ball hit as an indication to locate the players' position in the court using visual information, and then to infer the ball trajectory.

Our approach is to make independent estimates of the timings of the ball hits using visual and audio information, and then combine them in a probabilistic framework to generate improved estimates. A detailed description of the approach follows in the next sections.

III. Theoretical Framework

Our approach to ball-hit detection using multimodal information begins by finding the most likely sequence of visual events E_v^* together with the most likely sequence of audio events E_a^*, given the observed low-level visual (F_v) and audio (F_a) features. E_v^* and E_a^* can be estimated using equation 1:

$$(E_v^*, E_a^*) = \text{argmax}_{\{E_v, E_a\}} Pr(E_v, E_a | F_v, F_a) \quad (1)$$

Equation 1 can be re-written as:

$$(E_v^*, E_a^*) = arg \max_{\{E_v, E_a\}} Pr(E_v|F_v) Pr(E_a|F_a) Pr(E_a|E_v) \quad (2)$$

Equation 2 factors the ball hit detection into three processes:

1) ball hit detection only using visual information ($Pr(E_v|F_v)$)
2) ball hit detection using audio information ($Pr(E_a|F_a)$)
3) refinement of the audio events given the detected visual events ($Pr(E_a|E_v)$)

The first process, using only visual information, uses ball-tracking to provide coarse detection of ball hits, based on the positions of the peak and trough points in the tracked ball trajectory in each play shot scene. This technique generates false positives because of noise and unconnected trajectory fragments (caused by occlusion). The second process, using only audio information, follows our previous work [7]: we train a set of acoustic models for seven different audio events found in the game (one of which is a ball hit) and treat detection of ball hits as a classification problem. The third process treats the visually detected ball hits as constraints that reduce the impact of noise and other types of audio event on the audio-detected ball hits. We combine audio and visual information at the *event* level rather than using a low-level audio-visual fusion approach. There are two reasons for this approach: firstly, fusion at low-level requires good synchronisation between the audio and visual information, which is a not always obtained on recordings. Secondly, making independent audio and visual estimates of the ball hits sidesteps the problem of mismatch between the training and test data, which causes considerable problems in fusing the data.

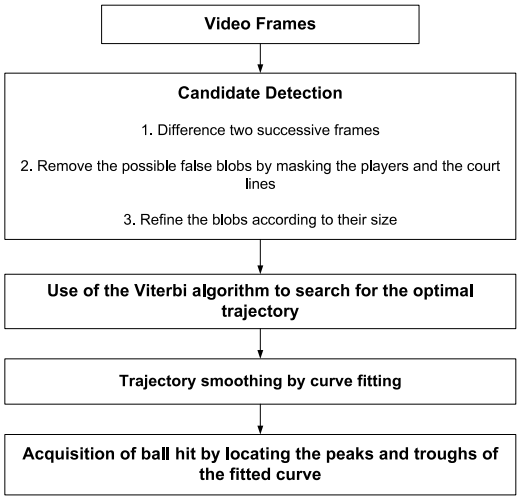
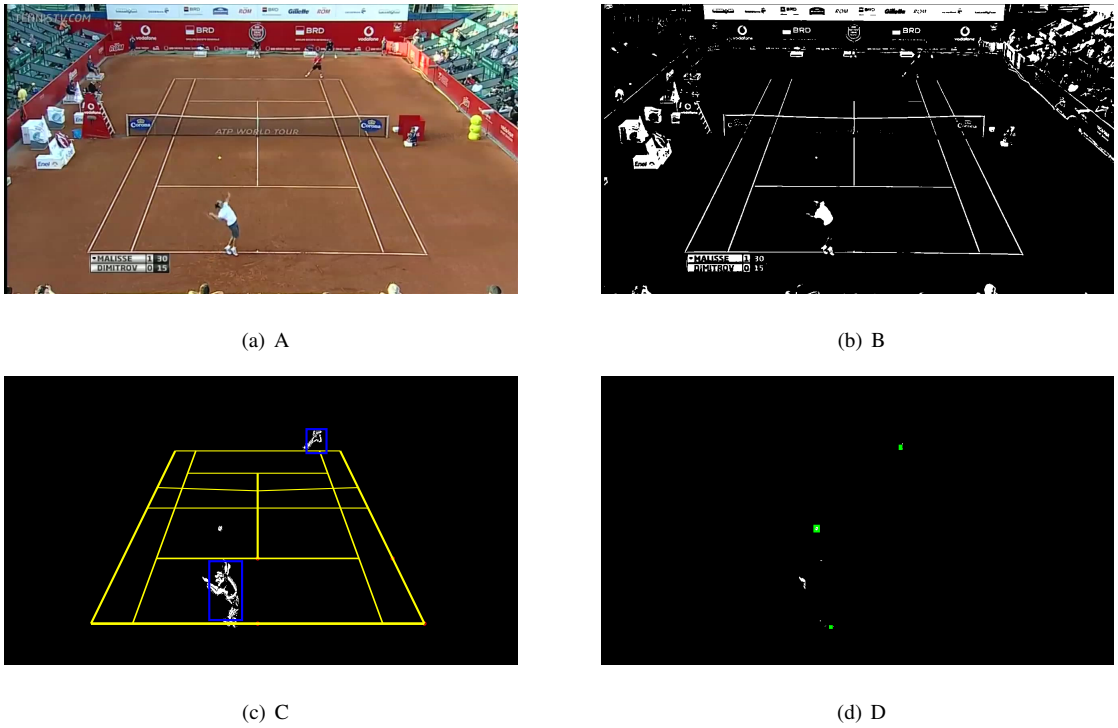


Fig. 2. Visual event detection framework



(a) A (b) B

(c) C (d) D

Fig. 1. Detection of ball candidates (a) Original image; (b) Binary image (c) Image after locating court lines and removing all non-court regions (d) Three ball position candidates (in green)

IV. Detection of Visual Event

The essential idea of ball hit detection using visual information is to locate the peaks and troughs of the smoothed ball trajectory obtained by searching for the optimal path from possible candidates over the observed frames of a play shot scene. Figure 2 presents the four main steps in this technique:

1) Detection of ball candidates
2) Search of the optimal path through the ball candidates
3) Smooth of ball trajectory
4) Location of peaks and bottoms on the fitted curve

A. Ball Candidates Detection

To a good approximation, the ball's colour is white in long view shots, so white pixels are first segmented according to equation 3

$$B(x,y) = \begin{cases} 1 & r(x,y) \geq I \wedge g(x,y) \geq I \wedge b(x,y) \geq I \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where B is a binary image and (x, y) is the pixel location, $r(x, y)$, $b(x, y)$ and $g(x, y)$ denote the RGB values of each pixel, respectively. The threshold I is set to be 150 empirically. Figure 1 (b) shows the binary image converted from the original image (Figure 1(a)).

When detecting the ball candidates, the false candidates are mainly from court lines, players' motion, spectators around the tennis court, and from artefacts due to camera calibration. To reduce the number of false candidates, we try to segment the court region from the spectators' region, and then mask the court line and players in the court.

The court segmentation is based on the acquisition of court lines. To accurately find all court lines, we utilise a homography transform, described by a 3×3 matrix H, to find the mapping between points in a "virtual" tennis court template and some points in the current frame. The pixel coordinate in the template is represented by a vector $[x\ y\ 1]^T$ which is multiplied by H yielding the vector $[u\ v\ w]^T$:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4)$$

The final target coordinate is $(x', y') = (u/w, v/w)$. The division by w warps the coordinates properly to account for perspective foreshortening. For a detailed description of this process, refer to [26].

The homography transform enables us to obtain the coordinates of four points at the corner of the court, to then filter out the region for spectators, and to mask the court line. To further reduce the possible false ball candidates, we also try to locate the players in the court by finding the regions in the top-half and bottom-half court with the largest variations in intensity by differencing two successive frames. After locating players, the region is masked by a rectangular box, whose size is adaptively changed according to the player's position in the court. Figure 1 (c) shows the re-plotted court line after the homography transform has been applied.

To further reduce the number of false ball candidates, we

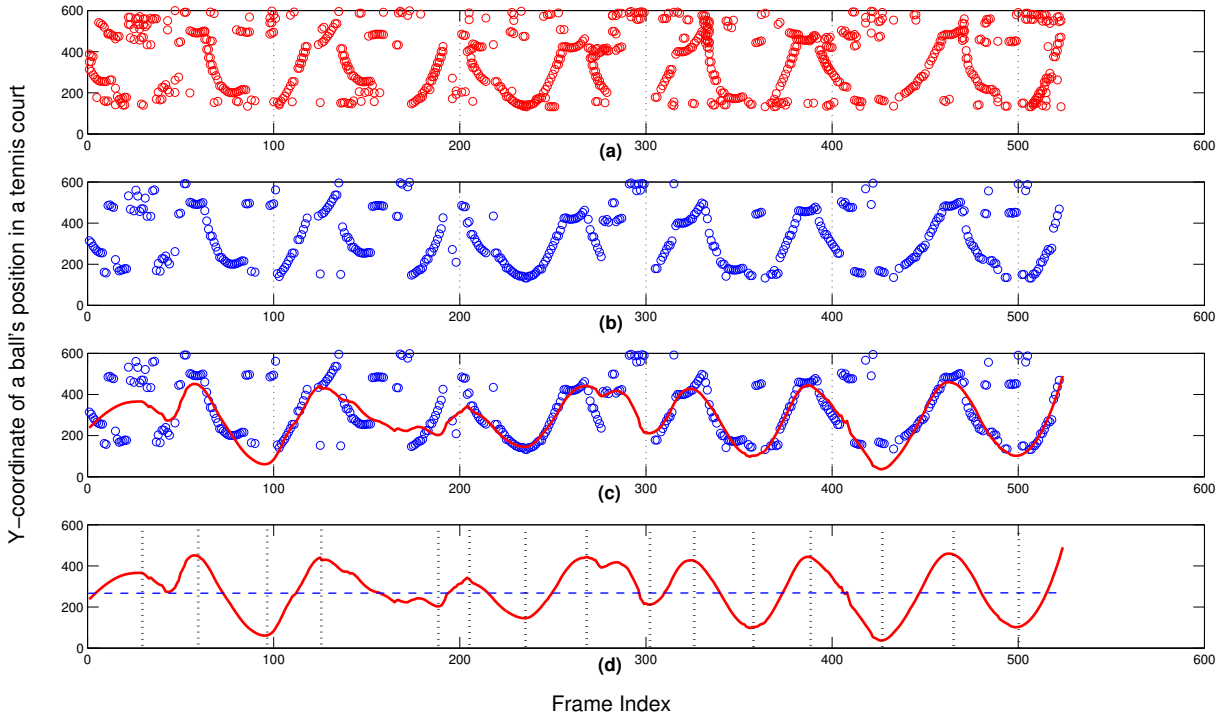


Fig. 3. An example of locating ball hits after searching for the optimal path with the Viterbi algorithm

set limitations to the size of the candidate "blob" (B) after court segmentation and application of masks to court lines and players.

$$B(i) = \begin{cases} 1 & (3 \leq W(i) \leq 40) \wedge (3 \leq L(i) \leq 40) \\ & \wedge (W(i) * L(i) \leq 300) \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where $W(i)$ and $L(i)$ denotes the width and length of the ith ball candidate, $B(i)$. In figure 1 (d), we show the final set of ball candidates, which are labelled with green squares.

After the initial very coarse location of the ball position based on white pixels, we do not take its colour into account because it can change in different matches due to the light and the colour of background: in fact, the colour can change at the different ends of a tennis court in the same match. Shape is also not considered, because it is greatly affected by motion blur and low frame rate.

B. Estimating the Ball Trajectory

The search for the optimal ball trajectory is made using the Viterbi algorithm, which requires three probabilistic parameters:

- the probability of a state ($\Pr(S)$),
- the probability distribution of the observations ($\Pr(b)$)
- the transition probability between any two states ($\Pr(T)$).

We treat each ball candidate (B) in the current frame F_t as one state of the total number of ball candidates N_t at time t. This means that the number of states in a frame varies. The number of frames depends on the duration of the rally and varies greatly from about one second (where the "rally" is a single serve) to as long as 40 seconds for a very long rally. We assume that all ball candidates in one frame have a uniform probability distribution. Unlike previous work which bases the observation probability b on variables such as the colour, size and shape of the ball, we define b only by its position (Pos)

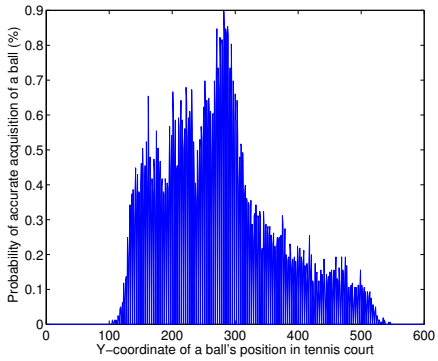


Fig. 4. Observation probability

in the court.

$$b(B_i) = \Pr(Pos(B_i)),\ 1 \leq i \leq N_t \tag{6}$$

One reason for this has already been mentioned in section II—the colour and shape can change, even within a game. The second one is that the ball position can be fairly accurately captured when the ball is in the middle of the court, but at the ends of the court, the position is much harder to observe because of players' motions and background noise. Figure 4 shows a histogram of the y-coordinate of the ball, made from ground-truthed data. To simplify the computation, $b(B_i)$ is modelled as a normal distribution based on this histogram.

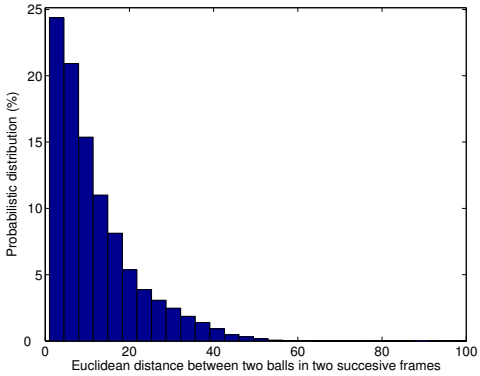


Fig. 5. Probabilistic distribution of two balls in two successive frames

The transition probability between states is estimated computing the distribution of the Euclidean distance (D) between the ball candidate (B_i) in the previous frame F_{t-1} and the candidate (B_j) in the current frame F_t.

$$T(S_{ij}) = \Pr(D_{t,t-1}(B_i, B_j)),\ 1 \leq i \leq N_{t-1},\ 1 \leq j \leq N_t \tag{7}$$

Figure 5 shows the probabilistic distribution of the distance between any two balls in two successive frames, obtained from training data.

Figure 3, the top and second pane show respectively the number of ball candidates before and after the Viterbi search. The y-axis is the value of the y-coordinate of a candidate position extracted from a frame, the x-axis is the frame (time) index. In the top pane, there are multiple candidates for the ball position in each frame. Use of the Viterbi algorithm reduces the number to a single candidate per frame, as shown in the second pane.

C. Smoothing and Event Location

Although the use of the Viterbi algorithm reduces the number of false candidates, some incorrect ball positions remain in the trajectory. This means that the path formed by connecting the detected positions is very noisy and thus location ball hits, which correspond to peaks and troughs in the trajectory is inaccurate.

Firstly, we use curve fitting to smooth the tracked candidate positions. Smoothed values are determined by neighbouring data points defined within a span of five frames. A locally weighted linear least-squares regression based on a quadratic polynomial is performed on these five points [28]. The third pane of figure 3 shows the fitted curve (solid line). However, the curve does not fit the ball trajectory very well at times when there are a high number of incorrect candidates (e.g. between frames 280 and 300) and missing data (e.g. around frame 90).

For detection, we use the value of the y-coordinate to divide the fitted curve into two parts, one part corresponding to balls in the half of the court nearest the camera, the other to balls in the distant half of the court. We can thus easily locate peaks and troughs by finding the positions of the maximal or minimal values within these two regions. The bottom pane of figure 3 shows the smoothed trajectory with the estimated ball hit positions shown as dashed vertical lines.

V. Audio Event Detection

In previous work [7], we defined seven types of audio events for the description of tennis matches, one of which was the sound of the racquet hitting the ball. Table I gives descriptions of each audio class and their related functions in a tennis game. For audio event detection, there are two issues to be addressed:

TABLE I
AUDIO CLASSES USED IN THIS WORK

Audio Event	Name	Function
Chair umpire's speech	UMP	Report Score
Line judge's shout	LJ	Report serve out, fault etc.
Sound of ball hit	BH	Serve, Rally
Crowd noise	CN	Applause
Beep	BP	Let
Commentators' speech	COM	
silence	SIL	-

1) distinguishing between the seven types of audio events;
2) reducing the impact of acoustic mismatches between the training and test data.

The first problem is solved in a standard maximum-likelihood framework by finding the most likely audio event given the "observed" low-level audio information, F^a , as shown in equation 8:

$$E^{a^*} = \arg\max_{E^a} \Pr(E^a|F^a) \tag{8}$$

$$\propto \arg\max_{E^a} \Pr(F^a|E^a)\Pr(E^a) \tag{9}$$

$\Pr(F^a|E^a)$ indicates a posterior probability computed using a Gaussian mixture model (GMM), and $\Pr(E^a)$ can be regarded as a prior distribution of each audio type (set equal in this paper). Equation 9 is the tranformation of equation 8 after using Bayes theorem.

To reduce the impacts of acoustic mismatch, we employ a confidence meaure (CM). The likelihood of each audio event class for a frame is estimated using the Gaussian mixture models of audio events built from the training-data, and the difference between highest log likelihood and the next highest is used as a CM for that frame. This use of a difference between likelihoods provides some immunity from mismatches

between the training- and test-set channel conditions: if the mismatch is high, then all the likelihoods will be low, but the overall mis-match will be cancelled out by the differencing operation, and the differences will be relatively stable within a range. A suitable threshold for the CM corresponding to a positive detection of an audio event ball hit can be determined from the training data.

VI. Combination of Audio and Visual Information

The combination of audio and visual information for ball hit detection is based the assumption that they provide complementary information. We employ a probabilistic framework to combine the audio ball-hit probabilities (E^a_{BH}) with the visual ball-hit probabilities (E^v_{BH}) at the "event" level.

Equation 9 can hence be changed to:

$$E^{a^*}_{BH} = \arg\max_{E^a_{BH}} \prod_t \Pr(F^a_t|E^a_{BH}) \Pr(E^a_{BH}) LH^\alpha(F^v_t|E^v_{BH}) \tag{10}$$

The term $\Pr(F^a_t|E^a_{BH})$ gives the audio probability of frame F^a_t given a ball hit at time t. The term $LH(F^v_t|E^v_{BH})$ gives the visual probability of frame F^v_t given a ball hit at time t. $LH(F^v_t|E^v_{BH})$ is defined to be high when a ball hit has been visually detected and to be 1.0 at other times. Specifically, at a time when a ball hit has been detected (t_0), it is modelled as the absolute difference between the y-coordinate of the ball and the y-coordinate of the horizontal middle line of the court, C_0:

$$LH(F^v_t|E^v_{BH}) = \begin{cases} abs(Pos_y(B(t)) - C_0) & t = t_0 \pm 2 \\ 1 & t \neq t_0 \pm 2 \end{cases} \tag{11}$$

This equation shows that, in practice, we extend the period at which a visual event is detected by ± 2 visual frames to deal with the synchronisation problems referred to in section III.

We include a parameter α to control the influence of the visually detected events on $E^{a^*}_{BH}$: the higher the value of α, the more emphasis is given to the decoded visual events at the expense of the decoded audio events. α is empirically set to be 4.0 in our experiments (we experiment with varying α in the last set of experiments).

VII. Data

Soundtrack data from four tennis matches was used, one match for training and the other three for test. Table II gives essential information about these matches. The training data is extracted from a men's single match of the Wimbledon Open (2008), while the test matches are from the Australian Open (Test 1), the US Open (Test 2), and the French Open (Test 3). The soundtracks of the four matches are segmented into 30 ms frames using a sliding window with a 20-ms overlap. This means the audio frame rate is 100 frames per second, higher than the visual frame rate (25 frames per second). Each audio frame is converted into a vector of 39-D MFCCs (13 static components, plus velocity and acceleration). Gaussian mixture models (GMMs) are built from frames labelled as belonging to each of the seven audio classes shown in Table I.

TABLE II
Data for training and test

	Game	Type	Dur. (mins.)	# ball hit
Train	Wim-08	singles	180	1528
Test (1)	AUS-10	singles	106	736
Test (2)	US-11	singles	81	719
Test (3)	French-12	singles	83	572

The training audio data (Wim-08) was fully annotated with the seven classes. For the audio test data, we marked only the positions of the ball hits. Care was taken to ensure that the audio and video were synchronised.

VIII. Experimental Set-Up

Audio detection performance is measured under two conditions:

- AC1: not using the confidence measure
- AC2: using the confidence measure

Visual detection performance is measured under four conditions:

- VC1: curve fitting over all frames
- VC2: curve fitting over the frames after removing blank frames
- VC3: search of the optimal path using a sliding window
- VC4: search of the optimal path not using a sliding window

After applying smoothing, there are many frames not containing any ball candidates, which we term "blank" frames. These frames are usually generated when the ball is occluded by a player or a court line during a rally, or when the scene switches to one in which there is no play, and are obviously detrimental to ball hit position estimation.

Some frames contain many candidates, and these cause a problem in the Viterbi search. When estimating the most likely trajectory using a global search, the presence of these frames allows too much freedom, especially if the number of frames is large, and consequently, the resulting trajectory can be very inaccurate. To combat this problem we experimented with using a set of local searches, in which dynamic programming is performed over a window of D frames, and the window is then shifted by s frames. The resulting set of optimal paths is then joined to find the overall best trajectory.

To measure performance, an *F-score* is used, defined as:

$$P = \frac{\text{\# correctly detected ball hits}}{\text{\# detected ball hits}} \tag{12}$$

$$R = \frac{\text{\# correctly detected ball hits}}{\text{\# ball hits in ground truth}}$$

$$F - score = \frac{2PR}{P+R}. \tag{13}$$

A ball hit is considered to be correctly detected when it is located within the manually annotated range of an event labelled as a ball hit. Detected ball hits that lie outside such a region are regarded as false positives, and undetected ball hits are false negatives.

IX. Results and Analysis

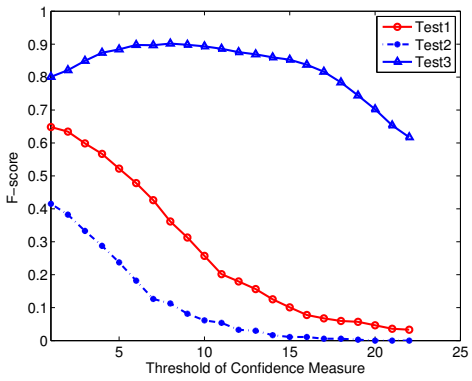


Fig. 6. Performances of ball hit detection using only audio information on three test matches

Figure 6 shows the F-score performance for ball hit detection using only audio information on the three matches in the test-set. The x-axis on this figure is the value used for the confidence measure (CM). When the CM is zero, no confidence measure has been used. We see that the best detection performances are obtained on Test 1 and 2 without using any CM, whilst we obtain the best performance on Test 3 when $CM = 7$. This may be due to the acoustic mismatch between the training and the test data. The acoustic characteristics of the training match (a Wimbledon singles match) are similar to that of Test 3, but are quite different from Test 1 and Test 2. This leads to overall higher performance on Test 3, which is further enhanced by the use of a CM. Table III shows the F-score performance using only visual information for the cases VC1 and VC2 defined in section VIII. Performance on all three matches is fairly similar, and for Test 1 and Test 3, lower than performance obtained using audio information. Significant improvements are obtained on all three matches after removing the blank frames.

TABLE III
Performances of ball-hit detection in the conditions of VC1 and VC2 on three test matches

	Test(1)	**Test(2)**	**Test(3)**
VC1	0.4391	0.3201	0.3882
VC2	0.5427	0.4728	0.5250
Impr.(%)	+23.59	+47.70	+35.24

Table IX shows performance using a local search with various window sizes (VC3) and a global search (VC4). Table IX shows the performances when using a local and global search, respectively. For some values of D, the local search gives slightly better performance, but the results are not conclusive.

Table V shows the performances on three test matches when changing the number of shift frames. As with a fixed window size ($D = 10$), some shift sizes can give slightly better performance but overall, results are inconclusive. .

TABLE IV
Detection performance on the three test matches using visual information only. VC3 uses a local search with a variable window size, D. VC4 uses a global search.

Condition		**Test(1)**	**Test(2)**	**Test(3)**
Local Search (VC3)	D=5	0.5418	0.4725	0.5215
	D=10	0.5427	0.4728	0.5250
	D=15	0.5390	0.4731	0.5267
	D=20	0.5340	0.4728	0.5267
	D=25	0.5359	0.4728	0.5267
	D=30	0.5312	0.4738	0.5262
Global Search (VC4)		0.5364	0.4698	0.5129

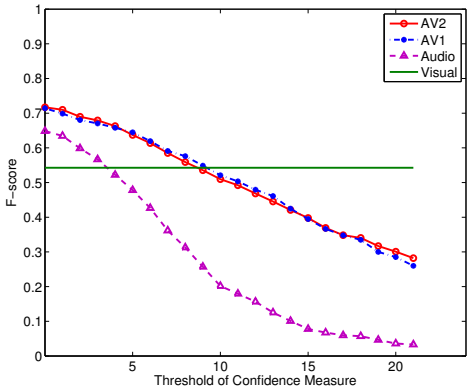
TABLE V
Detection performance on the three test matches using visual information only. VC3 uses a local search with a fixed window size of 10 and a variable shift size, s.

Condition	**Test(1)**	**Test(2)**	**Test(3)**
S=1	0.5427	0.4728	0.5250
S=2	0.5424	0.4750	**0.5254**
S=3	0.5360	0.4661	0.5254
S=4	0.5381	**0.4836**	0.5197
S=5	0.5269	0.4663	0.5160
S=6	0.5370	0.4678	0.5146
S=7	**0.5437**	0.4657	0.5152
S=8	0.5255	0.4810	0.5174
S=9	0.5335	0.4619	0.5121
S=10	0.5325	0.4638	0.4942

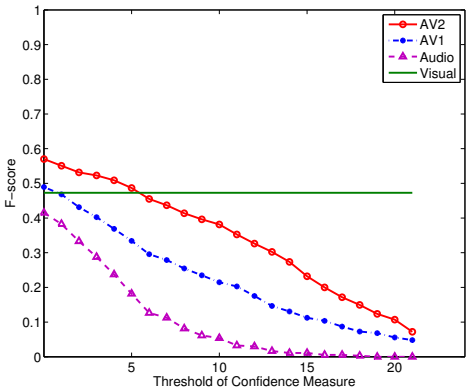
Figure 7 compares the performances on each of the three test matches when audio and visual information is combined with performance using audio information alone and visual information alone. Performance is plotted as a function of the CM value used in the audio detector. $AV1$ and $AV2$ show the performance when blank frames are not removed (AV1) and when they are removed (AV2). $Audio$ is the performance using only audio information, and $Visual$ performance using only visual information.

It can be seen that combining audio and video with no confidence measure always gives better performance than using video alone, and at least as good performance as using audio alone. For match Test 3, if the confidence measure is correctly chosen, the combination performance is better than the already high performance of audio alone.

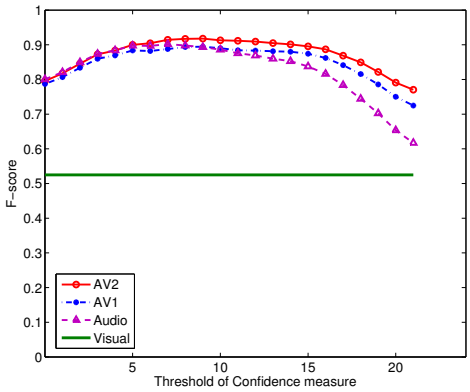
When combining the detected visual events with the audio information, we set a parameter α in equation 10 to adapt the effect of visual event detection. Figure 8 shows the performance on the three test matches with α = 1, 4 and 100. When α is high, (e.g. 100) the value of the audio confidence measure has less effect on the performance than when α is low, which is what we would expect, as more weight has been given to the visual hypotheses. However, the figures indicate that for all three test sets, if the value of $\alpha > 4.0$, there is little or no effect on the overall best performance at the optimum value of the CM, which is 0 for Test 1 and Test 2 and about 7 for Test 3.



(a) Test (1)



(b) Test (2)



(c) Test (3)

Fig. 7. Performance comparison of ball-hit detection

X. Conclusion and Future Work

In this paper, we have demonstrated that ball-hit detection can be successfully performed on a recording that may have a low visual frame rate and a poor quality soundtrack by fusing audio and visual information at the "event" level. We have presented separate approaches to detection using audio and video information, and a probabilistic technique for integrating the two modalities. Our approach copes well with frequently encountered problems in this area such as low-level audio interference, training and test-set mis-match and audio/video synchronisation problems. We believe that the approach of constraining detection in one modality by using information from the other modality has general application in many audio-visual scenarios, including audio-visual speech recognition, segmentation, and understanding.

In our future work, we firstly will consider how to further improve the effectiveness and efficiency of tracking a tennis ball in more complex conditions, such as a background with more noise and severe camera calibration. We will aim to reduce the acoustic mismatch with taking the visual information into account and we will use a similar approach in detecting the voices of the line judges, which are also key to understanding the game. This will require improved robustness to different interferences, which we aim to achieve by integrating more context information. We also intend to extend the technique to sports games in different domains.

References

[1] Miyamori, H., "Automatic annotation of tennis action for content-based retrieval by integrated audio and visual information", *in IEEE Int. Conf. on Image and Video Retrieval*, pp.331–341, 2003.

[2] Tien, M. and Wang, Y. and Chou, C. "Event Detection in Tennis Matches Based on Video Data Mining", *in IEEE Int. Conf. on Multimedia and Expo*, pp.1477–1480, 2008.

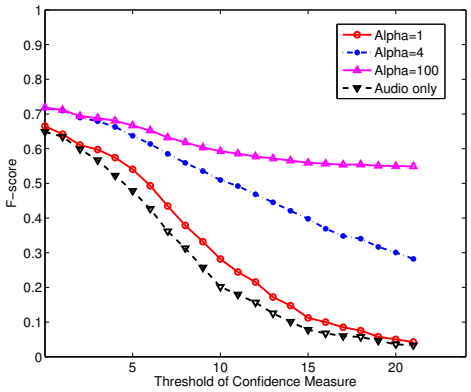
[3] Yan, F. and Christmas, W. and Kittler, J., "Layered Data Association Using Graph-Theoretic Formulation with Application to Tennis Ball Tracking in Monocular Sequences", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp.1814–1830, 2008.

[4] Figueroa, P. J. and Leite, N. J. and Barros, R. M. L., "Tracking soccer players aiming their kinematical motion analysis", *Computer Vision and Image Understanding*, vol. 101, pp.122–135, 2006.

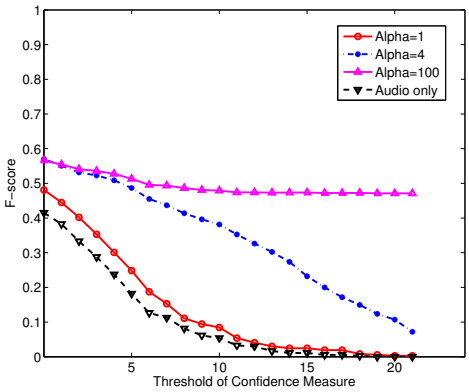
[5] Zhu, S. and Mumford, D., "A Stochastic grammar of images", *Foundations and Trends in Computer Graphics and Vision*, vol. 2, no. 4, pp. 259-362, 2006.

[6] Huang, Q. and Cox, S. and Yan, F. and Campos, T. and Windridge, D. and Kittler, J. and Christams, W. "Improved Detection of Ball Hit Events in a Tennis Game Using Multimodal Information", *Int. Conf. on Auditory-Visual Speech Processing*, pp.123–126, 2012.

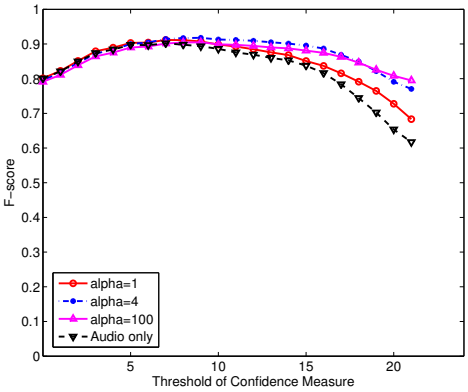
[7] Huang, Q. and Cox, S., "Hierarchical Language Modeling for Audio Events Detection in a Sports Game", *In Proceedings of ICASSP'10*, pp.2286–2289, 2010.



(a) Match Test 1



(b) Match Test 2



(c) Match Test 3

Fig. 8. Performance of ball bit detection using audio and visual information with different $\alpha = 1, 4, 100$

[8] Huang, Q. and Cox, S., Using High-level Information to Detect Key Audio Events in a Tennis Game, *In Proceedings of InterSpeech*, pp.1409-1412, 2010.

[9] Cai, R. and Lu, L., Zhang, H.-J., and Cai, L.-H., "Highlight sound effects detection in audio stream", *In Proceedings of ICME*, pp.37–40, 2003.

[10] Zhuang, X. and Zhou, X. and Huang, T and Hasegawa-Johnson, M., "Feature analysis and selection for acoustic event detection", *in Proceedings of ICASSP*, pp.17–20, 2008.

[11] Lu, L.,"Content analysis for audio classification and segmentation", *IEEE Trans. Speech and Audio Processing*, vol 10:504–516, 2002.

[12] Atrey, P. and Maddage, N. and Kankanhalli, M., "Audio Based Event Detection for Multimedia Surveillance", *in Proceedings of ICASSP*, pp.813–816, 2006.

[13] Lu, L. and Cai, R. and Hanjalic, A., "Towards a Unified Framewrok for Content-based Audio Analysis", *in Proceedings of ICASSP*, pp.1069–1072, 2005.

[14] Lefevre, S. and Maillard, B. and Vincent, N., "3 Classes Segmentation for Analysis of Football Audio Sequences", *in IEEE Int. Conf. on Digital Signal Processing*, pp.975–978, 2002.

[15] Yu, X. and Sim, C. and Wang, J. and Cheong, L., "A Trajectory-based Ball Detection and Tracking Algorithm in Broadcast Tennis Video", *in IEEE Int. Conf. on Image Processing*, pp.1049–1052, 2004.

[16] Yu, X. and Tu, X., and Ang, E. L., "Trajectory-Based Ball Detection and Tracking in Broadcast Soccer Video with the Aid of Camera Motion Recovery", *In Proceedings of ICME*, pp.1543–1546, 2003.

[17] Choi, K. and Park, B. and Lee, S. and Seo, Y., "Tracking the Ball and Players from Multiple Football Videos", *Information Acquisition*, vol.3(2), pp.121–129, 2006.

[18] Pingali, G., "Ball Tracking and Virtual Replays for Innovative Tennis Broadcasts", *In the 15th International Conference on Pattern Recognition*, vol.4, pp.152–156, 2000.

[19] Liang, D. and Liu, Y. and Huang, Q. and Gao, W., "A Scheme for Ball Detection and Tracking in Broadcast Soccer Video" *PCM 2005, Part I*, LNCS 3767, pp.864–875, 2005.

[20] Yan, F. and Christmas, W. and Kittler, J., "A Maximum A Posteriori Probability Viterbi Data Association Algorithm for Ball Tracking in Sports Video", *18th International Conference on Pattern Recognition*, vol.1, pp.279–282, 2006

[21] Yu X., Xu C., Leong H. W., Tian Q., Tang Q., and Wan K. W., "Trajectory-Based Ball Detection and Tracking with Applications to Semantic Analysis of Broadcast Soccer Video", in Proceedings of ACM Conference on Multimedia, 2003, pp.11-20.

[22] Tong X., Lu H., and Liu Q., "An Effective and Fast Soccer Ball Detection and Tracking Method", in Proceedings of International Conference on Pattern Recognition, vol. 4, 2004, pp.795-798.

[23] Dahyot, R. and Kokaram, A. and Rea, N. and Denman, H., "Joint Audio Visual Retrieval for Tennis Broadcast", *in Proceedings of ICASSP*, pp.561–564, 2003.

[24] Kijak, E. and Gravier, G. and Oisel, L. and Gros, P., "Audiovisual integration for tennis broadcast structuring", *In International Workshop on (CBMI03)*, pp. 289–312, 2003.

[25] Lao, W. and Han, J. and With, P. de, "Ball-Path Inference Based on A Combination of Audio and Video Clues in Tennis Video Sequences", *In DSP Valley Signal Processing Symposium*, 2006.

[26] Hartley, R. I. and Zisserman, A., *Multiple View Geometry in Computer Vision*, Cambridge University Press, ISBN: 0521623049.

[27] Rabiner, L., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *in Proceedings of the IEEE*, pp. 257-286, 1989.

[28] Cleveland, W. S., “LOWESS: A program for smoothing scatterplots by robust locally weighted regression”, *The American Statistician*, pp.35–54, 1981.