

SmartTennisTV: Automatic indexing of tennis videos

Anurag Ghosh and C.V.Jawahar

CVIT, KCIS, IIT Hyderabad

anurag.ghosh@research.iit.ac.in, jawahar@iit.ac.in

Abstract. In this paper, we demonstrate a score based indexing approach for tennis videos. Given a broadcast tennis video (BTV), we index all the video segments with their scores to create a navigable and searchable match. Our approach temporally segments the rallies in the video and then recognizes the scores from each of the segments, before refining the scores using the knowledge of the tennis scoring system. We finally build an interface to effortlessly retrieve and view the relevant video segments by also automatically tagging the segmented rallies with human accessible tags such as 'fault' and 'deuce'. The efficiency of our approach is demonstrated on BTV's from two major tennis tournaments.

1 Introduction

Sports streaming websites are very popular with many services like TennisTV and WatchESPN offering full game replays on demand. Millions of users use these services for entertainment, education and other purposes. However, tennis matches are usually very long, often running into hours. It's very hard to infer playing styles and patterns of players without investing hundreds of hours of viewing time. Thus, it's cumbersome to find "useful" parts. Streaming websites provide the video as-is, i.e. it's only possible to access the video stream sequentially. However, in case of sports and other event-rich video streams, an useful extension is to provide random access (like accessing an array) grounded in events along with sequential access, so that extensions like skipping to next event, filtering events etc can be provided.

In this paper, we focus on constructing a point wise index of a tennis match and thus providing random access to the match. We propose a method to segment out the match into a set of rallies, then automatically extract the scorecard and the scores. Using tennis domain knowledge, we construct a novel algorithm to refine our extracted scores. We then demonstrate the utility of the automatically constructed index by building an interface to quickly and effortlessly retrieve and view the relevant point, game and set segments along with providing human accessible tags.

There are multiple challenges in this scenario. The tennis match videos are recorded from multiple camera angles and edited to have different kind of shots, to capture various emotions and drama along with the game play. With respect

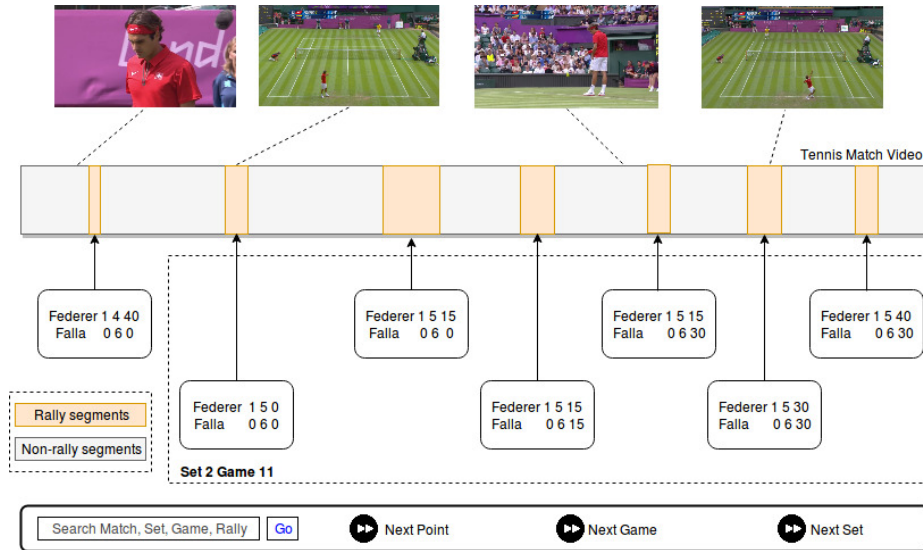


Fig. 1. We aim to provide random access to tennis match videos and construct a point wise index of a tennis match so that a user can access, jump and skip “points”, “games” and “sets”.

to extracting scores, the score board is never at a fixed position or in a specific format and the score digits are not constrained by font, size, and color.

The major contributions of this paper are,

1. An effective score recognition algorithm using domain knowledge which can be adapted for different games. Here, we do our experiments on tennis videos by using the tennis domain knowledge.
2. We propose a score based indexing system, to navigate and retrieve segments from large volumes of video data with considerable ease.
3. Our method also enables many applications of indexing, we demonstrate one such application, human accessible event tagging.

Section 2 discusses advances and related work in literature. Section 3 forms the core of the paper, describing our core approach. Lastly, Section 4 provides a brief background of tennis and a high level description of our dataset(s), describes the implementation details and the experiments we performed along with obtained results.

2 Related Work

Sports Understanding: Using domain specific cues, several researchers have previously worked on improving sports understanding (specially tennis), with

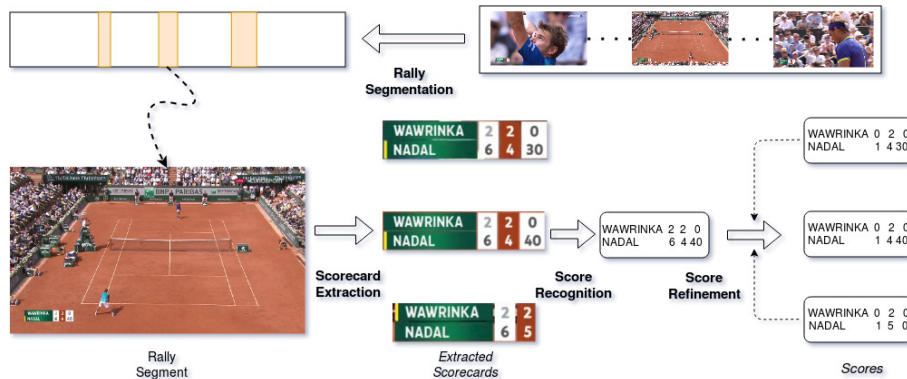


Fig. 2. Our approach is illustrated in this figure. We start by temporally segmenting out the rallies, extracting the scoreboard and then recognizing the scores where we use contextual and domain knowledge to refine the recognized scores.

strides made in video summarization and automatically generating highlights [3, ?,?], generating descriptions [10] and automatically segmenting coarse temporal scenes [5], annotating players [14, ?] and tracking the ball [13, ?].

Sports Video Indexing and Applications: Xu et al. [12] and Miyamori et al. [9] focus on semantic annotations exploiting tennis domain knowledge to build retrieval systems based on positions and actions. Sukhwani et al. [11] proposed a dictionary learning method for frame level fine grained annotations for a given video clip, but their annotations are also computed at the level of actions, useful in the context of computing player statistics. Kolekar et al. [6] use audio features to detect events in soccer scenes and generate highlights. Liu et al. [7] perform multimodal analysis to generate tennis video highlights while Connaghan et al. [8] attempt to segment out game into point, game and set, however, perform no score keeping and use multiple cameras to perform the task. However, these methods do not attempt to robustly index point level information to enable retrieval from the point of view of a viewer. Our work differs from all of these as we attempt to annotate point level information for a match.

Scorecard and Score Extraction: Liao et al. [16] focus only on detecting the scorecard while Miao et al. [17] focuses on both detection and extraction of scores, however the algorithm is specific for Basketball. Tesseract [18] is the commonly used OCR pipeline to detect text from images and documents which have a plain background. Convolutional Recurrent Neural Network (CRNN) [20] is applicable for performing end-to-end scene text recognition while Textspot [19] introduces a Fully Convolutional Regression Network (FCRN) which performs end-to-end scene text detection and for recognition, uses the intermediary stage of the pipeline based on the lexicon-encoding CNN from Jaderberg et al. [21].

3 Approach

Our goal is to automatically create an index for tennis videos. We begin by describing a method to automatically segment rallies. Then we detect and localize the scorecard in each of these rallies and recognize the text to abstract out the game score state to annotate the video with the accessible tags. An overview of our pipeline can be seen in Fig. 2.

3.1 Rally Segmentation

Our method of segmenting out rallies stems from the observation that in BTV’s, the camera is only overhead when the rally is in play and nowhere else. The background is mostly static after the serve begins, and remains the same till a player wins a point. HOG features are appropriate in such a scenario, so we extract frames from the Segment Dataset, downscale them, and extract HOG features. We then learn a χ -squared kernel SVM to label a frame either as a rally frame or a non-rally frame. Then, we use this learned classifier to label each frame of the BTV as part of a rally or otherwise and smoothen this sequence using Kalman filter to remove any false positives/negatives to obtain the segmented rallies.

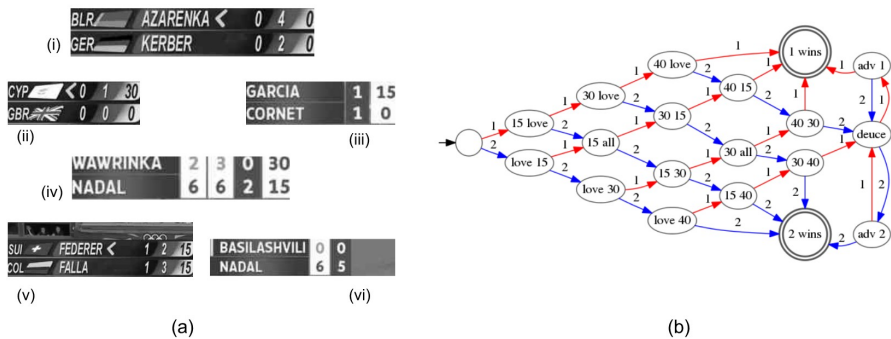


Fig. 3. (a) depicts some of the extracted scorecards from different matches from our dataset. As one can see, the scorecards detected are of different sizes and formats, and differences across tournaments is noticeable. We have also included some of our failure cases, (v) and (vi) have extra regions that have been detected. (b) depicts the tennis point automaton that can be constructed from the tennis scoring system which is used to refine our extracted scores.

3.2 Scorecard Extraction

We utilize the observation that the scorecard position is stationary in a rally, while the camera pans and moves around to cover the game. However, the scorecard may disappear and is not necessarily of the same size across the game as

opposed to the assumptions in [17]. So, to overcome these issues, we extract the scorecard independently from each rally segment instead of assuming a single scorecard template.

We adapt the method described in [16]. We start by finding the gradient for each frame (say $I_x(i, j, t)$) using the sobel filter, and then calculate the normalized temporal sum for each frame using, $I_{norm}(i, j, n) = \frac{1}{n} \sum_{t=1}^n I_x(i, j, t)$. Then, we subtract I_x and I_{norm} to obtain the temporally correlated regions I_g . Further, we binarize the image using the following equation,

$$I_r(i, j, t) = (1 - \frac{I_x(i, j, t)}{\max_{t,i,j}(I_x)}) I_{norm}(i, j, t) \quad (1)$$

Empirically, the scorecard is found in one of the corners of the frame, we identify the four regions of size $(h/5, w/2)$ in the corners as the regions to search for the scorecard. Note, w and h are the width and height of the frame respectively. We identify the coarse scorecard region by selecting the region with the maximum number of white pixels in the specified regions in $I_r(i, j, t)$ summed over time. Further, after we have identified the coarse region, we apply morphological operators to remove small aberrations present and fit a rectangle which encloses the scorecard area. Our qualitative results can be seen in Fig. 3 (a).

3.3 Score Recognition

Traditional OCR based methods like Tesseract [18] can recognize text printed on a clear background however need the image to be preprocessed if the background is textured and shaded, and the contrast in the text fragments varies widely. However, with the advent of deep learning based OCR and scene text detection methods, a more general approach can be formulated.

To recognize scores, we experiment with three different methods, Tesseract, CRNN and Textspot. Textspot combines FCRN [19] which is an end to end text detection network, which constructs a field of predictors where each predictor is responsible for detecting a word if the word centre falls within the corresponding cell, akin to the YOLO network architecture. The recognition is performed by the intermediary stage of the pipeline based on the lexicon-encoding CNN from Jaderberg et al [21]. CRNN [20] is a scene text recognition network which treats the image as a sequence of strips. It proceeds by treating a CNN as a feature extractor to extract feature maps and construct a sequence of feature vectors. The sequence is fed into a bi-directional LSTM to obtain label sequence probabilities and CTC loss is employed to obtain labels. We adapt and perform a comparison of the various score recognition baselines in Section 4.

3.4 Score Refinement

To further refine our recognized scores, we use the knowledge of the tennis scoring system. As any structured game, score keeping in tennis is governed by a set of rules and thus, can be modeled as a finite automaton. Tennis in specific can be

modeled as 3 automatons, one each for tracking the point, game and set score (See Fig. 3 (b)). Also, the vocabularies for point, game and set are restricted, so, we find errors by checking if the value belongs to the vocabulary or not. For instance, the the vocabulary for a point score is restricted to $\{0, 15, 30, 40, AD\}$.

Let $J = (game_1, set_1, point_1, game_2, set_2, point_2)$ be the score state where game, set and point have the same meanings as in tennis. Firstly, we exploit the fact that the game and set scores are usually remain constant in a window, and thus replace errors with the mode of the value in the temporal window (with exceptions for score change within the window).

Consider the tennis scoring automaton T which is composed of score states and the transition function is constructed using the tennis scoring rules. Then we define a function $nextStates(s)$ which returns all possible states for the next game state. Likewise, $previousStates(s)$ provides the set of originating states for the current state s . For instance, from Fig. 3 (b), if we assume that we are at state $s = (0, 0, 30, 0, 0, 30)$ (referred to as 30 all in the figure), the function $previousStates(s)$ will return $\{(0, 0, 30, 0, 0, 15), (0, 0, 15, 0, 0, 30)\}$ and $nextStates(s)$ would return $\{(0, 0, 40, 0, 0, 30), (0, 0, 30, 0, 0, 40)\}$.

Assuming that the set of scores is $S = \{s_1, s_2 \dots s_n\}$, and that s_i is erroneous (using vocabulary constraints), we compute the set $P = nextStates(s_{i-1}) \cap previousStates(s_{i+1})$, then we find the corrected score using,

$$s'_i = \arg \max_{p \in P} \frac{1}{|J|} \sum_{j \in J} \delta(s_i(j), p_i(j)) \quad (2)$$

where J is the set of game score states and δ is the Kronecker delta function. This equation is only needed if there are more than one possible score. It is to be noted that this method is extensible to any game which follows a structured scoring system like tennis.

4 Experiments and Results

4.1 Dataset

A tennis match is divided into sets, each set is divided into games and each game has certain number of points or rallies. We restrict ourselves to “singles” matches and work with broadcast tennis video (BTV) recordings at 720p for 10 matches. 5 matches are taken from the French Open 2017 and remaining matches are from London Olympics 2012 for all our experiments. For performing rally segmentation, we created a ”Rally Dataset” by manually annotating 2 matches into rally and non rally segments. The training and test set images are derived by dividing all the images in a 50-50 split. For evaluating score extraction, we further annotated 4 matches with score of each segment using the automated segmented rallies from our described algorithm. All together, we have annotated 1011 rallies to create the ”Match Scores Dataset”.



Fig. 4. The developed interface supports the indexing and retrieval of a match as a point, game and set.

4.2 Rally Segmentation

For learning the rally segmentation classifier, we extracted every 10th frame from Rally Dataset and cross validated using a 0.8 split to find the optimal values of the hyper-parameters C and the period of the χ -squared kernel. The optimal value of C is 0.05 and the period of the χ -squared kernel SVM is found to be 3.

The mean $F1$ score on the test set for the task was found to be 97.46%, the precision for the non-rally segments was 98.94% and the rally segments was 95.41%.

4.3 Score Recognition

Table 1. Averaged Edit Distance for score recognition (Lower is better)

Match	Textspot	CRNN	Tesseract-P
Match 1 (186 rallies)	0.2070	0.4272	0.2612
Match 2 (218 rallies)	0.2178	0.4476	0.3780

For employing Tesseract, we carefully preprocess the scorecard image and threshold the image manually. For each tournament such a preprocessing step needs to be manually defined. To train the CRNN, which is constrained to recognize words as sequences, we divided the scorecard to two parts horizontally. For employing Textspot, we don't train the network and use the model trained on "SynthText in the Wild" dataset as [19] note state-of-the-art performance

on standard benchmarks. However, we post-process the text detection boxes and sort them to extract the scores. We used edit distance instead of the usual text recognition metrics because the “spaces” between scores (in the recognized string) are relevant in our case. For instance, CRNN removes repetitions of numbers, which causes the decrease in accuracy. Table 1 here presents our experimental results on a subset of the matches and as we can see, Textspot performed the best and thus, for the next set of experiments we use that as our baseline.

4.4 Score Refinement

It is important to reiterate that our aim is not to recognize the text in the scorecard, but rather capture the game score state. To evaluate our results, we formulate a new metric, which inputs computed game state C_i and the actual game state G_i , and computes the following (for a set of rallies say R),

$$AC(R) = \sum_{i \in R} \frac{1}{|J|} \sum_{j \in J} \delta(C_i(j), G_i(j)) \quad (3)$$

where J and δ as defined earlier.

Table 2. Averaged Score Accuracy $AC(R)$ for our method and the defined baseline, FCRN (Higher is better)

Match	Textspot	Ours
Match 1 (186 rallies)	79.30%	91.66%
Match 2 (218 rallies)	77.90%	80.58%
Match 3 (201 rallies)	92.45%	95.19%
Match 4 (194 rallies)	85.22%	92.18%

As can be seen from Table 2, our refinement algorithm shows a consistent improvement in the averaged score accuracy across matches over the best performing baseline method, Textspot [19]. However, as it is apparent, the performance of our method is dependent on the performance of the baseline score recognition and that is possibly the reason in the relatively meager improvements in score accuracy in the second match.

4.5 Event Tagging

Further, we automatically tagged common tennis events of importance to viewers such as “fault”, “deuce” and “advantage” using simple rules which define these tennis terms and our extracted scores. We compare our accuracy with and without score refinement and can observe that there is an improvement corresponding to improvement in the score accuracy. Accuracy for each tag per match (the matches are same as Table 2) can be seen in Table 3.

Table 3. Averaged Accuracy score of automatic event tagging (in percentage)

	Match 1		Match 2		Match 3		Match 4	
	Textspot	Ours	Textspot	Ours	Textspot	Ours	Textspot	Ours
Fault	66.66	70.83	52.24	56.71	87.87	90.90	84.44	84.44
Deuce	100.0	100.0	73.68	78.94	100.0	100.0	94.73	94.73
Advantage	100.0	100.0	77.77	77.77	100.0	100.0	95.65	95.65
Overall	75.00	79.41	60.58	64.43	92.45	94.33	89.65	89.65

5 Conclusion

We have presented an approach to create a tennis match index based on recognizing rallies and scores, supporting random access of “points” (Fig. 4) tagged with common tennis events. Further extensions to this work are numerous, such as providing point based semantic search and performing tennis player analytics using videos instead of expensive sensor-based technologies.

References

1. Huang, Yo-Ping, Ching-Lin Chiou, and Frode Eika Sandnes. “An intelligent strategy for the automatic detection of highlights in tennis video recordings,” *Expert Systems with Applications*, Vol. 36, No. 6, pages 9907-9918, 2009
2. B. Ghanem, M. Kreidieh, M. Farra, and T. Zhang. “Context-aware learning for automatic sports highlight recognition,” In *21st International Conference on Pattern Recognition (ICPR)*, pages 1977-1980. IEEE, 2012.
3. A. Hanjalic. “Generic approach to highlights extraction from a sport video,” In *International Conference on Image Processing (ICIP)*, Vol. 1, pages 11. IEEE, 2003.
4. M. Mentzelopoulos, A. Psarrou, A. Angelopoulos, and J. Garcia-Rodriguez. “Active foreground region extraction and tracking for sports video annotation,” In *Neural processing letters*, Vol. 37, No. 1, pages 33-46, 2013.
5. Y. Zhang, X. Zhang, C. Xu, and H. Lu. “Personalized retrieval of sports video.” In *Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 313-322, 2007.
6. M. H. Kolekar, S. Sengupta, “Bayesian Network-Based Customized Highlight Generation for Broadcast Soccer Videos”, *IEEE Transactions on Broadcasting*, Vol. 61, No. 2, pages 195-209, 2015.
7. Liu, Chunxi, Qingming Huang, Shuqiang Jiang, Liyuan Xing, Qixiang Ye, and Wen Gao. “A framework for flexible summarization of racquet sports video using multiple modalities.” *Computer Vision and Image Understanding*, Vol. 113, No. 3, pages 415-424, 2009
8. Connaghan, Damien, Philip Kelly, and Noel E. O’Connor. “Game, shot and match: Event-based indexing of tennis.” In *Content-Based Multimedia Indexing (CBMI)*, 2011 9th International Workshop on, pages. 97-102. IEEE, 2011.
9. H. Miyamori and S.-I. Iisaku. “Video annotation for content based retrieval using human behavior analysis and domain knowledge,” In *Fourth IEEE International Conference on Automatic Face and Gesture Recognition Proceedings*, pages 320-325, IEEE, 2000.

10. M. Sukhwani and C. Jawahar. "Tennisvid2text: Fine-grained descriptions for domain specific videos," In Proceedings of the British Machine Vision Conference (BMVC), pages 117.1-117.12, BMVA Press, September 2015
11. M. Sukhwani and C. Jawahar. "Frame level annotations for tennis videos," In International Conference on Pattern Recognition, 2016
12. C. Xu, J. Wang, H. Lu, and Y. Zhang. "A novel framework for semantic annotation and personalized retrieval of sports video," IEEE transactions on multimedia, Vol. 10, No. 3, pages 421-436, 2008.
13. F. Yan, W. Christmas, and J. Kittler. "All pairs shortest path formulation for multiple object tracking with application to tennis video analysis," In British Machine Vision Conference, 2007.
14. F. Yan, J. Kittler, D. Windridge, W. Christmas, K. Mikolajczyk, S. Cox, and Q. Huang. "Automatic annotation of tennis games: An integration of audio, vision, and learning," Image and Vision Computing, Vol. 32, No. 11, pages 896-903, 2014.
15. X. Zhou, L. Xie, Q. Huang, S. J. Cox, and Y. Zhang. "Tennis ball tracking using a two-layered data association approach," IEEE Transactions on Multimedia, Vol. 17, No. 2, pages 145-156, 2015.
16. S. Liao, Y. Wang, and Y. Xin. "Research on scoreboard detection and localization in basketball video," In International Journal of Multimedia and Ubiquitous Engineering, Vol. 10, No. 11, pages 57-68, 2015.
17. G. Miao, G. Zhu, S. Jiang, Q. Huang, C. Xu, and W. Gao. "A real-time score detection and recognition approach for broadcast basketball video," In IEEE International Conference on Multimedia and Expo, pages 1691-1694, 2007.
18. R. Smith. "An overview of the tesseract ocr engine," In Ninth International Conference on Document Analysis and Recognition, Vol. 2, pages 629-633, 2007.
19. Gupta, Ankush, Andrea Vedaldi, and Andrew Zisserman. "Synthetic data for text localisation in natural images," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2315-2324. 2016.
20. Shi, Baoguang, Xiang Bai, and Cong Yao. "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," IEEE transactions on pattern analysis and machine intelligence. 2016.
21. Max Jaderberg and Karen Simonyan and Andrea Vedaldi and Andrew Zisserman. "Synthetic data and artificial neural networks for natural scene text recognition," arXiv preprint arXiv:1406.2227. 2014.