

# Football Video Segmentation Based on Video Production Strategy

Reede Ren and Joemon M. Jose

Department of Computing Science,  
University of Glasgow,  
17 Lilybank Gardens, G12 8QQ, UK  
{reede, jj}@dcs.gla.ac.uk

**Abstract.** We present a statistical approach for parsing football video structures. Based on video production conventions, a new generic structure called ‘attack’ is identified, which is an equivalent of scene in other video domains. We define four video segments to construct it, namely *play*, *focus*, *replay* and *break*. Two middle level visual features, *play field ratio* and *zoom size*, are also computed. The detection process includes a two-pass classifier, a combination of Gaussian Mixture Model and Hidden Markov Models. A general suffix tree is introduced to identify and organize ‘attack’. In experiments, video structure classification accuracy of about 86% is achieved on broadcasting World Cup 2002 video data.

## 1 Introduction

Many techniques have been developed in the literature for football video analysis, starting from shot classification[4] and scene reconstruction[15], to structure analysis[3][5][6], event extraction[9][14] and summarization[7][12]. These approaches primarily focus on visual cues. Ekin et al[12] categorized them into cinematic and object-based ones. Cinematic algorithms utilize features from video composition and production rules, while object-based turns to video object detection. Xu and Lei et al[6] proposed the cinematic feature ‘dominant colour ratio’ to segment video. They indicated that video reporters have to focus on play yard to convey game status. In [5], they used a set of HMMs to parse broadcasting video into *play* and *break*, where *break* presents a stop of game, while *play* contains normal game clips. Object-based features enable high-level domain analysis, but their extraction may be computationally expensive and sometimes needs manual supervision. Intille[7] and Gong et al[4] analyzed football trajectories and player interactions to detect a large set of semantic events. Both of their work rely on pre-extracted accurate object trajectories. Ekin[12] introduced a framework employing both cinematic and object-based features. It includes low-level football video processing algorithms, such as dominant colour region detection, shot boundary detection and shot classification, as well as some higher level algorithms for goal detection, referee detection and penalty-box detection.

A new trend is to combine audio and visual information under one framework [1, 2]. The idea has been examined in some recent papers, from event detection to scene boundary analysis. Baillie and Jose[9] detected game highlights by selected audio features, i.e. Mel-Frequency Cepstral Coefficients(MFCC). In [13], video segment detectors were developed for audio, colour and motion separately. Project 'Multiject'[14] fused audio sub-band data and colour histograms. The main problem behind this approach is asynchronism of audio and visual cues. It stands on two facts, (1) Audio and picture stream are independently encoded, transferred and replayed in most commercial video formats, i.e. H.263, MPGE-1/2 and AVI. There exists random delay between them when playing. (2) Audio and video are of different resolution. According to multi-sensor theory, an event in audio stream may carry on for several seconds and the resolution of audio is on coarse minute level, while video is updating at the speed of 25 frames per second with the resolution of finite second level.

Jurgen[3] discovered that there exists typical production styles and editing patterns, which make football video a loose simple-structured temporal sequence. These embedded repetitive structures are called video pattern or video structure. In this paper, we describe a new approach for video segmentation. We introduce a novel structure called '*attack*' based on video making conventions. By close observation to football video production tactics, we first define the structure '*attack*', an equivalent of scene in other video domains. Then we extend Lei's 'play' and 'break' detection framework[5] to '*play*', '*focus*', '*replay*' and '*break*' detection. These segments form the set of football semantic alphabets to construct '*attack*'. Finally, we utilize '*attack*' to setup a content-based video index and offer variable semantic summaries.

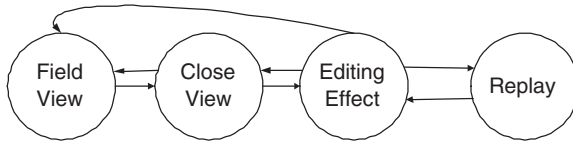
We select three salient features; field ratio, zoom size and image mean contrast. A two-pass classification system is employed to detect these video structures. Our goal is to parse continuous video stream into a sequence of '*attack*'. Subsequently, we set up a hierarchical video content index to summarize the game and allow a nonlinear navigation of video content.

The rest of paper is organized as follows. Section 2 introduces the semantic sensitive video structure framework and provides a HMM football video model. In Section 3, we describe the video structure discrimination system and related feature extraction algorithms. Section 4 covers '*attack*' scene construction. Experimental results are shown in Section 5. In 6, a brief of our nonlinear video browser and summarization system based on '*attack*' segmentation is previewed. The final Section 7 comes with discussion and conclusion.

## 2 Football Video Structure

### 2.1 Video Production Strategy in Football Broadcasting

A football game is made up by a series of team movements called '*attack*' in sports jargon[10]. They are mostly independent and sorted by time throughout the game. In some sense, '*attack*' decides broadcasting strategy. During broadcasting, video reporters focus on two issues, (1) how to record the game or



**Fig. 1.** The Video Making Sequence During Attack

*'attack's*; (2) how to avoid missing interesting issues in an *'attack'*. They employ field view to describe team tactics and middle view or close-up view to catch players' detailed movement. When an important event or highlight takes place, such as goal, it will be replayed. The strategy (Fig.1) can be stated as following,

1. When an attack begins, a global view will be used until the ball passes the centre circle.
2. When the ball comes into front field, a middle view is going to be employed to show how groups of players attack and defend.
3. When the ball come into or close to the penalty area, a close-up view is here to catch possible highlights and players' action in detail.
4. When there is a highlight, such as shoot and foul, a close-up slow motion replay will come to state the event.

With these observations, we conjecture,

1. Video making methods in football game dictate the structure of video and compose semantics.
2. As a time sequence, a football game can be modelled by Hidden Markov Model with *'attack'* video structure.
3. *'Attack'* is an independent semantic video unit, which can be treated as a scene in football video domain. It is useful in video segmentation, indexing and summarization.

## 2.2 Four-Class Video Structure

*'Attack'* takes the role of **scene** in our framework. To detect it, we define a new video structure layer between shot and *'attack'*(Fig.2). It includes four mutually exclusive video structures in broadcasting video data(*play*, *focus*, *replay* and *break*). During **play**, video makers convey global status of the game and employ long and medium shots or field view in the general video terminology[10]. **Focus** is a short stop of game, in which the video maker traces a player to show his or her detailed actions. In video production terminology, it is called player close-up. **Replay** is for slow-replays. **Break** includes non-game video clips, such as interview and advertisement. These structures are useful in event detection[9] and helpful in shot boundary allocation. Moreover, they bring following advantages, (1) We can identify video segment with clear game content. It helps in video summarization and indexing, and promises a compact meaningful highlight set. (2) These video segments will not overlap in both time and semantics. It eases

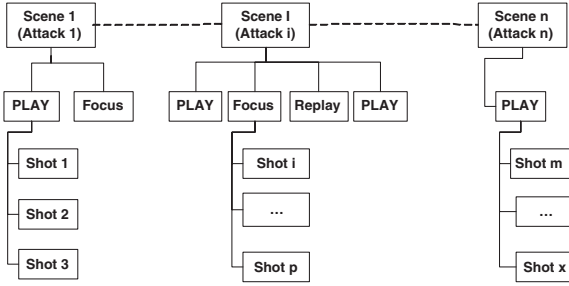


Fig. 2. Hierarchical Video Structure for Football

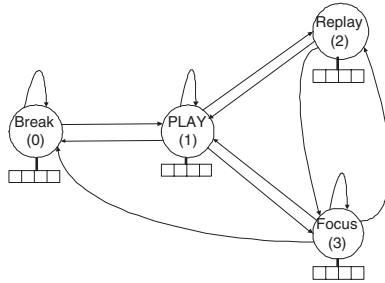


Fig. 3. Football Game Model

video indexing, which has developed complex index structure[16], such as X-tree and R+ tree, to keep overlapped video segments for retrieval. (3) These video structures maps actual film production skills, such as focus and replay, which can be detected automatically.

Given the repeat nature of ‘attack’, a football game can be modelled by the hidden Markov Model(Fig.3). The model has four states: (0) Break, (1) Play, (2)Replay and (3) Focus, starting from *break* and ending in *break*.

### 3 Video Structure Detection System

In all four types, *replay* is ad hoc. It is a replenisher of prior frames, while *play*, *focus*, and *break* are characterized by view content. A two-pass classification(Fig.4) is proposed to deal with the difference. It identifies video structures and labels video frames with their video structure type, ‘*play*’, ‘*focus*’, ‘*replay*’ and ‘*break*’. The first pass discriminates *play*, *focus* and *break* by a GMM classifier and its output label sequence is smoothed by dynamic programming process(Fig.5a). The second pass detects ‘*replay*’. From domain knowledge, *replay* is a slow motion video clip sandwiched by editing effects. So the process consists of a slow motion identification[8] and an editing effect detector. The HMM(Fig.5b) identifies slow motion clips among *play* and *focus*, while the editing effect detector looks for editing effect sequences before and after slow motion clips. Both of

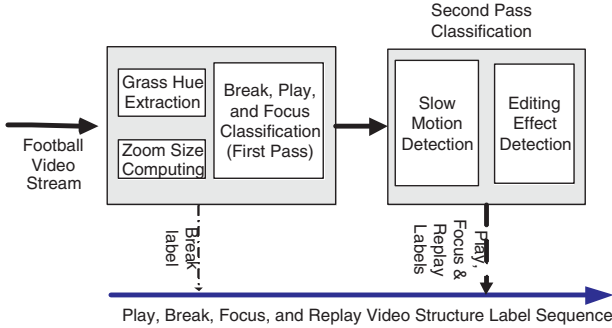


Fig. 4. Video Structure Identification System

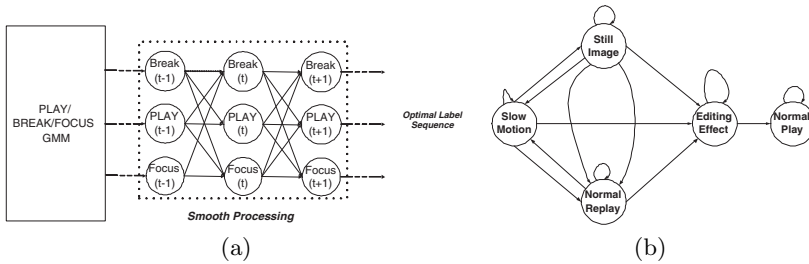


Fig. 5. (a) Mixed Classifier for Break, Play and Focus Discrimination (b) HMM for Replay Scene Detection([8])

them allocate ‘replay’. After the classification, we get the video structure label sequence.

Three middle-level salient features are computed in current system for the first pass GMM classification, namely field ratio  $R_{field}(t)$ , zoom size  $P(t)$  and image mean contrast  $Con(t)$ (Eq.3). In following subsections, we will introduce our algorithms for feature calculation and edit effect sequence detection.

### 3.1 Field Ratio

Xu[6] proposed *dominant colour ratio* to classify sports video. It is defined as play field area ratio over image,  $R_{field}(t) = \frac{\|H_{field\_colour}(t)\|}{\|H(t)\|}$ , where H is colour histogram of image blocks. Ekin[12] gathered grass pixels manually and calculated a prior grass colour model, in which grass occupies  $65^\circ - 85^\circ$  hue interval in HSV colour space. However, we argue that play field hue varies greatly with light, weather and location, it is difficult for a unified model to abide these variations while keeping high precision. On the other hand, play field is not always the dominant area throughout a video. In test data, more than 37% sample frames are with a field ratio lower than 20%, i.e. those belonging to ‘focus’ and ‘break’. In this work, we introduce an automatic pre-processing to detect play field colour distribution by two observations, (1) **play field is a homogeneous area**; (2)

**A video frame with dominant play field is more homogeneous than others.** We designed a two-layer booster to filter original video data to gather most possible play field blocks. The first layer rejects non-homogeneous frames and the second one excludes non-homogeneous area in homogenous frames. The *s\_rgb* colour space is selected to reduce lighting effect.

$$RGB \Rightarrow s\_rgb : \begin{cases} r = \frac{R}{R+G+B} \\ g = \frac{G}{R+G+B} \\ b = \frac{B}{R+G+B} \end{cases}$$

Given MPEG block encoding, we define block mean hue(Eq. 1) and block covariance(Eq.2) of  $n \times n$  ( $n = 8$ ) image blocks to reduce noise,

$$mean(i, j) = \frac{1}{n^2} \sum_{x=1}^n \sum_{y=1}^n C_{(i*n+x, j*n+y)} \quad (1)$$

$$cov(i, j) = \frac{1}{n^2} \sum_{x=1}^n \sum_{y=1}^n |C_{(i*n+x, j*n+y)} - mean(i, j)| \quad (2)$$

where  $C_{(i,j)}$  is the colour of Pixel( $i, j$ ). So the mean covariance of a frame will be,

$$MeanCov_{frames} = \frac{1}{IJ} \sum_{i=0}^{I-1} \sum_{j=0}^{J-1} cov(i, j) \quad (3)$$

where a frame contains  $I \times J$  blocks. The threshold is calculated by maximum entropy,

$$threshold = arg \max_N \sum_{n=0}^N (-P_n \log(P_n)) \quad (4)$$

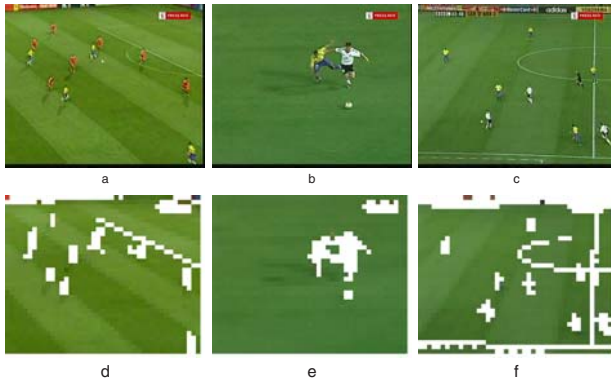
where  $P_n$  is the probability of frames whose mean block covariance is  $n$ .

All frames with a higher frame covariance than threshold will be rejected as the first layer of booster classifier. Another similar threshold is computed for every frame left, by which blocks with high covariance are removed as the second layer of booster. Fig.6 shows the effectiveness of this rejection stratagem, which keeps most of grass blocks while removing non-grass blocks. Then a GMM model is trained to simulate the grass colour distribution throughout the game by K-mean algorithm.

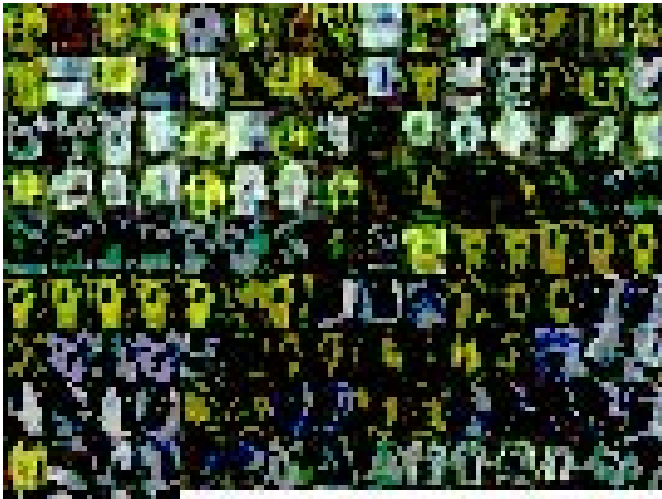
### 3.2 Zoom Size

Football uniform is an obvious domain feature. Compared with human face, it has following merits,

1. It is with bright colour and special pattern and can be easily discriminated.
2. it associates with the appearance of player only.
3. It is rotation robust.



**Fig. 6.** Effect of Grass Area Booster(a,b,c is origin images and d,e,f is respective result after boosting)



**Fig. 7.** Training Samples for Polo-shirt Detection

In broadcasting, uniform size varies significantly from  $9 \times 13$  pixel to more than  $180 \times 150$  pixels in  $352 \times 288$  video frame. We range it 13 scales, from 0 to 12, to measure zoom depth.

A FST(Foley-Sammon Transform) football uniform detector[11] is employed on multiple resolution from coarse to detail and decides its size. An 11-layer pyramid is built, in which every layer is 1.25 larger of the prior. The bottom one is of  $352 \times 288$  pixels. The detector scans every layer from left to right and from top to bottom. If it finds a polo shirt on a certain layer, for example, the second layer, the frame will be labelled with zoom size 2. If a polo shirt is not discriminated in any layer, zoom size will be zero. The training set includes about 300  $9 \times 11$  pixel samples(Fig.7) from different view.



Fig. 8. Log Transition Frames in World Cup 2002

### 3.3 Edit Effect Detection

‘*Replay*’ is sandwiched between logos of broadcaster. An automatic post-process of edit effects or logo transition detection will increase the precision of *replay* detection. Different from the algorithm in [8], we relies on colour histogram distance instead of pixel distance. It is robust on the presence of banners, which significantly changes the position of logo area and incapacitates the logo template detection algorithm[8].

A logo transition, usually 1-2 sec long or more than 30 frames in MPEG-1, is a set of consecutive frames that contain special logo. Fig.8 shows a logo transition sample for the football competition, World Cup 2002. It took place just before and after replay frames. A pre-log colour histogram array template is computed to detect edit effects before replay, while a post-log array template for these after replay. By slow-motion detection(Fig.5b), we build a ‘*replay*’ candidate set. Pre-log frame array and post-log frame array are computed for every candidate. They include  $n(n = 25)$  frames just before the start frame of the candidate or after the last frame, respectively. Then we align them. For two arrays  $u$  and  $v$ , the histogram array match measurement is defined as,

$$HC(u, v) = \min_{i \in [0, n)} \sum_{j=0}^n \|H_{i+j, u} - H_{i, v}\| + 1 \quad (5)$$

where  $H_{i, v}$  is the colour histogram of frame  $i$  in pre-log or post-log array of candidate  $v$ , and  $j$  is the match parameter. When the sum of  $i$  and  $j$  is greater than  $n$ , a large value will be assigned as a punishment. The algorithm seeking for the pre-log histogram array template can be described as,

1. Find two matching pre-log arrays with the smallest histogram array measurement in all;
2. Align them according to their match parameter  $j$  and compute histogram bin difference frame by fame;
3. The top 10 non-zero bins with smallest bin difference in every frame are characterized as eigen bins;
4. All eigen bins are sorted according to frame sequence to set up the  $n-j$  length histogram array template.

The histogram array template is employed to calculate the histogram match for all candidates. Mismatched candidate will be removed.



## 4 Attack Scene Construction

After video structure classification, we get the video structure label sequence "...BPFPPRP...", where **B** is the abridgement for 'Break', **P** for 'Play', **F** for 'Focus' and **R** for 'Replay'. The string records the process of video making and keeps the information of 'attack'. So the job of 'attack' construction is to divide it into a serials of substrings, which contain only one 'attack' sequence each. But the string is too long for the attack model (Fig.1) to detect 'attack' scenes directly.

From domain knowledge, 'replay' stands for game events and interrupts the game as 'break'. The occurrences of 'replay' and 'break' divide the whole sequence into a set of strings. But they are still too long. Given following facts, (1)Such a string may contain more than one 'attack' sequence; (2)'attack' is the largest video structure in our framework(Fig.2); (3)All 'attack' are similar in the video making sequence; video pattern of 'attack' can be treated as the longest common repetitive substring in these video making strings. This assumption also brings robustness to discrimination error and rouge artefact, such as a producer not using a slow motion as usual. Moreover, the attack model(Fig.1) will find the boundary between 'attacks'.

Let alphabet  $\Sigma = \{B, P, F, R\}$  and T be a string over  $\Sigma$ , the problem of longest common repetitive substring extraction can be stated as,

**Definition 1 (Normal Repeat and Super Maximal Repeat).** *A string p is called a normal repeat of T if  $p = T[i..i + |p| - 1]$  and  $p = T[i'..i' + |p| - 1]$  for  $i \neq i'$ . A super maximal repeat is a maximal repeat that never occurs as a substring of any other repeat.*

**Definition 2.** *Given a set of strings  $U = \{T_1, T_2, \dots, T_l\}$ , the (k,l) longest common repeat problem is to find the longest normal repeat which is common to k strings in U for  $1 \leq k \leq l$ .*

A generalized suffix tree(GST)[17] stores all suffixes of a set of strings as a suffix tree(ST) does for a string. Fig.9 is an example of the generalized suffix tree for  $T_1 = BBPFP$  and  $T_2 = BPFPPFP$ . Each leaf node has an ID representing the original string where the suffix came. The outline of our algorithm for the longest common repeat problem is as follows,

1. Build  $ST(T_i)$  for each  $1 \leq i \leq l$ .
2. Build  $GST(T_1T_2...T_l)$ .
3. Find super maximal repeats  $T_i$  for each  $i$  in  $GST(T_1T_2...T_l)$ .
4. Remove super maximal repeat branch from  $GST(T_1T_2...T_l)$  and build the GST of super maximal repeats.
5. Go to 3 unless the length of super maximal repeat is 1.
6. Find the longest common repeat among the super maximal repeat GST built in 4.



**Table 1.** Colour GMM With Different Classes Number

Class Number	Precision of Classification			Average Precision
	Play	Focus	Break	
2	74.5%	69.3%	74.6%	72.8%
3	80.2%	70.1%	76.0%	75.4%
4	84.0%	76.4%	75.2%	78.5%
5	81.7%	70.5%	74.3%	75.5%

**Table 2.** Play,Focus,Break Scene Classification Precision

Test Set	GMM			Training Set											
	Play	Break	Focus	Final I			Final II			Jap-Tur I			Jap-Tur II		
				Play	Break	Focus	Play	Break	Focus	Play	Break	Focus	Play	Break	Focus
Final I	0.894	0.840	0.823	<i>0.963</i>	<i>0.944</i>	<i>0.905</i>	0.913	0.852	0.830	0.948	0.920	0.877	0.933	0.917	0.891
Final II	0.786	0.664	0.708	0.824	0.730	0.721	<i>0.863</i>	<i>0.817</i>	<i>0.824</i>	0.835	0.713	0.773	0.824	0.711	0.765
Jap-Tur I	0.862	0.877	0.853	0.887	0.930	0.910	0.872	0.880	0.863	<i>0.890</i>	<i>0.952</i>	<i>0.917</i>	0.887	0.925	0.912
Jap-Tur II	0.880	0.861	0.836	0.905	0.892	0.870	0.897	0.870	0.845	0.930	0.905	0.887	<i>0.971</i>	<i>0.915</i>	<i>0.903</i>
avg-gen	0.856	0.811	0.805	0.894	0.874	0.852	0.886	0.855	0.840	0.898	0.873	0.864	0.904	0.867	0.868

**Table 3.** Mean Precision and Recall of Video Structure Classification

Test Set	Average Precision				Average Recall			
	Play	Break	Focus	Over All	Play	Break	Focus	Over All
Final I	0.931	0.896	0.866	0.898	0.941	0.926	0.883	0.917
Final II	0.827	0.718	0.753	0.766	0.894	0.879	0.864	0.879
Jap-Tur I	0.882	0.912	0.895	0.896	0.902	0.907	0.872	0.893
Jap-Tur II	0.930	0.889	0.867	0.895	0.955	0.896	0.875	0.909
Mean	0.893	0.854	0.845	0.864	0.923	0.902	0.873	0.899

Other three clips are employed for test. The process repeats for each video clips as the training set. We measure classification accuracy as the number of correctly classified samples over total number of samples. Training and testing accuracies are shown in Table.2. Average classification performance of each clip as test set (Table.3) is computed as the mean of the non-diagonal elements in Table.2. Average generalization performance (avg-gen) is computed for the clip as training set. Final II is noted for its lowest precision because the long celebration clip seriously garbles our classifier. A large group of people wearing uniform moved around in the play field. Those frames are compliant with ‘play’ and ‘focus’ in feature space, though we label them ‘break’. Besides Final II, our skim-how average precision is 89.6% (91.4% in ‘Play’, 89.9% in ‘Break’, and 87.6% in ‘Focus’).

The replay detecting HMM is trained by five pre-marked slow motion clips. It runs through all focus segments to find possible candidates, from which the editing effect histogram template is drawn. All of replays are found in experi-

**Table 4.** Candidate Set Size and Template Length

	Candidate Set Size	Actual Replay Segment Number	Histogram Array Template Length
Final I	31	19	18
Final II	47	14	7
Jap-Tur I	22	18	21
Jap-Tur II	29	16	22

**Table 5.** Attack Detection Precision

	Attack Number	Precision	Recall
Final I	32	0.732	0.890
Final II	40	0.541	0.794
Jap-Tur I	31	0.762	0.846
Jap-Tur II	44	0.710	0.803

ment. The result is shown in Table. 4, where the candidate set size is the number of video clips found by the slow motion HMM.

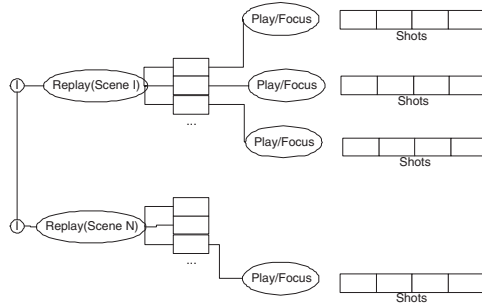
We employ TRECVID2003[18] video segmentation precision and recall to measure ‘attack’ result. The precision is the ratio of total time of correctly identified segments over total time of videos and the recall is total time of correctly identified segments over total time of reference segment.

## 6 Application: Browser Index

We propose a new indexing scheme for football video, called *browser index*. It is built from ‘attack’ and is organised along ‘play’, ‘focus’ and ‘replay’ structures, thus generating a hierarchical video index(Fig.2). ‘Replay’ covers highlights during a game and their congregation can be treated as a brief summary[8]. It represents ‘attack’. If it does not contain highlight, the ‘attack’ may be discarded as a plain one. We assign all ‘play’s and ‘focus’s in the same ‘attack’ to ‘replay’, and set up the middle layer of index, for they contain game information around ‘replay’. All of them will be decomposed into shots, which is the bottom. Fig.11 shows the index structure.

A non-linear video browser and an interactive video summarization system are developed based on this browser index. They not only supplies brief highlight summary, but can be improved to fill variant requirements through interaction. Two major interfaces are included, *related video browser*(Fig.12a) and *summary browser*(Fig.12b).

*Related video browser* retrieves ‘replay’ and its ‘play’ and ‘focus’ segments. It includes two regions in the panel, ‘replay’ segment list and related segments panel. The related segments panel displays top  $n(n=3)$  closest ‘play’ and ‘focus’ to the selected ‘replay’. User chooses ‘replay’ from the ‘replay’ segment list and



**Fig. 11.** Video Browser Index



**Fig. 12.** (a) Related Video Browser (b) Summary Browser

decides whether to contain it and its related video segments in summary or not. A double-click on icons will play the video clip by a stand alone window.

**Summary browser** shows all video segments in the proposed summary, and grants user the ability to insert and remove shots. The upper right region (Fig.12b) browses video segments, which are chosen in **related video browser**. All *'replay's* will be included as default. If user selects a video segment in the list, shots belongs to the segment will be shown in the bottom right region so that user can decide whether to include the shot or not.

## 7 Discussion and Conclusion

In this paper, we identified a new semantic video structure called *'attack'* for football videos. It is based on video production conventions and helps in video summarization and indexing. In some sense, *'attack'* is a semantic unit of football game and is an equivalent of scene in other video domains. The result shows those high-level video structures can be computed with high accuracy using middle-level features. We focus on video structure identification and how to merge these structures into *'attack'* scene. In the future work, we will measure accuracy of *'attack'* boundary. The algorithm leaves much space for improvements: (1) Audio

event detectors, such as goal and whistle detection, can be integrated; (2) Improve GST algorithm to search more embedded video structure; (3) It will be worthwhile to investigate unsupervised learning scenarios without extensive training.

## References

1. Ricardo Lenardi, Pierangelo Migliorati and Maria Prandini, *Semantic Indexing of Soccer Audio-Visual Sequence: A multimodal approach based on controlled Markov chains*, IEEE Trans on Circuits&System for Video Technology, pp.634-643, Vol.14, No.5, 2004.
2. Ying Li, Shrikanth Narayanan and C.C.Jay Kuo, *Content-Based Movie Analysis and Indexing based on AudioVisual cues*, IEEE Trans on Circuits&System for Video Technology, pp.1073-1085, Vol.14, No.8, 2004.
3. Jurgen Assfalg et al, *Semantic Annotation of Sports Videos*, IEEE MultiMedia Vol.9, No.2,2002.
4. Y. Gong et al. *Automatic parsing of soccer programs*, Proc. IEEE Intl. Conf. on Mult. Comput. and Sys, pp.167-174, 1995.
5. Lexing Xie et al. *Structure analysis of soccer video with Hidden Markov Models*, ICASSP2002.
6. Peng Xu et al. *Algorithms and Systems for Segmentation and Structure Analysis in Soccer Video*, IEEE International Conference on Multimedia and Expo, Tokyo, Japan, 2001.
7. S.Intille and A.Bobick, *Recognizing planned, multi-person action*, Computer Vision and Image Understanding, Vol.81, No.3, 2001.
8. H.Pan et al. *Detection of slowmotion replay segments in sports video for highlights generation*, ICASSP2001.
9. Baillie,M. and Jose J.M., *Audio-based Event Detection for Sports Video*, CIVR2003, pp.300-310, July 2003
10. Bob Burke and Frederik Shook, *Sports photography and reporting*, Chapter 12 in 'Television field production and reporting', 2nd Ed, Longman Publisher, 1996
11. Yue-Fei Guo and Lide Wu, *A generalized Foley-Sammon transform based on generalized fisher discriminant criterion and its application*, Pattern Recognition Letters 24(2003) 147-158.
12. A. Ekin, A. M. Tekalp, and R. Mehrotra, *Automatic soccer video analysis and summarization*, IEEE Trans. on Image Processing, vol. 12, no. 8, pp. 796-807, July 2003.
13. J.Huang, Z.Liu and Y.Wang, *Integration of audio and visual information for content-based video segmentation*, Proceedings of IEEE Conference on Image Processing, Oct. 1998.
14. M.R.Naphade et al. *Probabilistic multimedia objects(MULTIJECTS): a novel approach to video indexing and retrieval in multimedia systems*, Proceedings of IEEE Conference on Image Processing, Oct. 1998.
15. D.You, M.Yeung and G.Liu, *Analysis and presentation of soccer highlights from digital video*, Proc. ACCV, Dec.1995.
16. Jianping Fan et al. *Class View: Hierarchical Video Shot Classification, Indexing and Accessing*, IEEE Trans. on Multimedia, Vol.6 No.1, 2004.
17. L.C.K Hui *Color set size problem with applications to string matching*, Proceedings of the Third Annual Symposium on Combinatorial Pattern Matching, pp.230-243. Springer-Verlag, 1992.
18. TRECVID 2003, <http://www-nlpir.nist.gov/projects/tv2003/tv2003.html>