Towards Abstracting Sports Video by Highlights

Noboru Babaguchi

ISIR, Osaka University
Ibaraki, Osaka 567-0047, Japan

Abstract– Recently, video abstraction has become a demanding application in multimedia computing. It is de`ned as creating shorter video clips or video posters from the original video stream. In this paper, we present a basic approach towards abstracting sports video by highlights, dealing with American football games. Using event based indexing, we create an abstracted video clip automatically. To select the appropriate highlights of the game, an impact factor re'ecting on the importance of the event is newly introduced. It was possible to make an about 5-minute clip from the 3-hour original video.

I. INTRODUCTION

With the remarkable increase of multimedia systems, video data has been focused on as a core element [1]. It is characterized by multimodality and temporal dimension. Synchronized *multimodal information streams* (MMISs) such as visual, auditory, and text streams constitute the video data. Because it is usually too long along the time axis, a quick retrieval means is strongly required.

From the above background, *video abstraction* has become a demanding application. Its definition is creating shorter video clips or video posters from the original video stream. To complete video abstraction, a variety of semantical analysis for video contents should be integrated.

The strategy of video abstraction is divided into two classes. The first is to create a concise video clip by compressing the amount of the video data along the temporal dimension. It is sometimes referred to as video skimming. Recent researches by Smith et al. [2], Lienhart et al. [3], and He et al. [4] are classified in this class. The second is to browse image keyframes reflecting the video contents on a two-dimensional plane, specifically, on a window of a computer display. It is suitable for at-a-glance presentation. Systems developed by Yeung and Yeo [5], or Uchihashi et al. [6] are such good instances.

In this paper, we present a fundamental approach towards abstracting *sports video* by *highlights*, dealing with American football games. Our proposal belongs to the first class as above. For sports video like broadcasted TV programs, it is difficult to determine the highlights because similar scenes recursively appear. Using *event based indexing* [7], we attempt to create the abstracted video clip.

Our indexing scheme concentrates on *intermodal collaboration* [8,9], which means collaborative processing taking account of the semantical dependency between MMISs. Its aim is to improve the reliability and efficiency in the extraction of semantical information. Since the advantage of such an idea for video processing has been recognized so far, a lot of embodiments were reported [10–13].

At the abstraction stage, in order to select the highlights from the original video, a measure for an event, named an *impact factor*, is newly introduced, indicating how large the event affects the final result. As a consequence, the abstracted clip is produced as a repetitive sequence of both video captions and highlighted shots.

II. EVENT BASED VIDEO INDEXING

Event based video indexing is a kind of indexing by semantical contents. We think that an event is defined over a time interval, not on a time point. Further, a current event is closely related to preceding events or subsequent ones.

In American football games, a lot of events can be considered from diverse points of view. In what follows, we are particularly concerned with the events that change the score, simply called *score events*: 'touchdown (TD),' 'field goal (FG),' 'point after touchdown (PAT)/ extra point (EP),' etc.

Let us now consider how the events can be detected from video data. It might be impossible to automatically detect them from visual streams by means of image analysis. What is TD's image model? Defining this is a crucial but difficult problem. Hence, we introduce intermodal collaboration as an alternative strategy.

We are here interested in the collaboration between visual and closed caption (CC) streams. Since the CC stream can be regarded as textual information, it is comparatively easy to capture information suitable for semantical indexes like events. The proposed method consists of the *CC stream analysis* and the *visual stream analysis*. The latter follows the former. We proceed to describe each analysis.

A. CC stream analysis

Fig.1 shows the outline of the CC stream analysis. From the CC stream, we extract *keywords*, then form a chain of keywords that should correspond to a target event, and finally determine a *time window* in which the event may occur.

Considering the hierarchical structure of the American football game [7], we first try to detect an *offense unit*. The score events are likely to occur in the unit. To segment the

0-7803-6536-4/00/$10.00 (c) 2000 IEEE

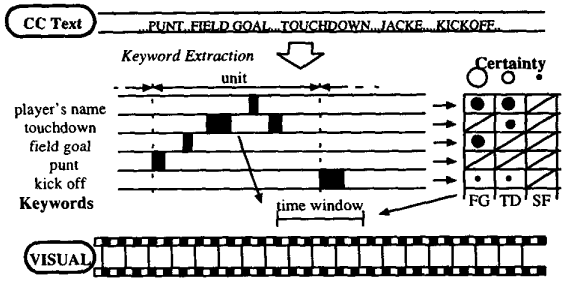

Fig. 1. CC stream analysis.

CC stream into the repetitive units, we define four classes of keywords beforehand as follows:

Class 1: events about the beginning of the offense, or related people, e.g. 'KICKOFF' and <punter>, where an actual name like 'CHRIS JACKE' is substituted into <punter>.

Class 2: preceding events related to offense sub-units, e.g. 'FIRST DOWN.'

Class 3: target events, or related people, e.g. 'TOUCHDOWN' and <kicker>.

Class 4: events about the end of the offense or the beginning of the next offense, or related people, e.g. 'KICKOFF.'

An extractor scans the CC stream from its head, performing the following procedure recursively. It tries to find possible keywords to form a chain in ascending order of their corresponding class. From each occurrence of the keywords between Class 1 and Class 4, it checks whether the offense unit exists or not. In this process, the time difference between the keywords is taken into consideration.

Once the unit is given, we determine a time window. Let $KEY(X)$ be a set of keywords that are concerned with an event 'X.' $KEY(X)$ is a subset of Class 3 or Class 4. We also define the *reliability* $R(key)$ of a keyword *key* on the basis of the frequency of each word during the time interval where the actual event occurs. Subsequently, a *certainty* factor $C(X)$ for the event 'X' is obtained as the summation of $R(key_i)$, where the keywords key_i ($key_i \in KEY(X)$) appear in the offense unit. If $C(X)$ exceeds some threshold, it is determined that there is the target event 'X' in the unit. TD and PAT are viewed as a single target because they are coordinated together. Therefore, the time window is labeled as either TD+PAT or FG. Table 1 shows the two sets: KEY(TD+PAT) and KEY(FG).

Finally, the *time window* is given by $[t^* - \alpha,\ t^* + \alpha]$, where t^* is the middle time point of the span of the keyword chain, and α is set 1 minute in this case.

B. Visual stream analysis

The time window given as above has a span of a couple of minutes. Because we aim at *shot-by-shot indexing*, we have to analyze the visual stream in the window in more detail.

Table 1. Event keywords.

Event	Keywords
TD+PAT	TOUCHDOWN, EXTRA POINT, KICKOFF, <kicker>
FG	FIELD GOAL, KICKOFF, <kicker>

We try to discover a shot that is similar to the target event by matching of feature vectors.

First, the stream is segmented into shots. The fundamental element in video processing is usually a shot, which is defined as consecutive image frames at a single camera view. The shots are given by detecting shot change operations such as cuts or dissolves.

Next, a shot to be matched is selected from all the segmented shots. For the shot, we extract $N1$ image frames from its head and $N2$ frames from its tail. Consider why we select such frames. In general, a TD scene is a transition from a scrimmage line to an end zone. The intermediate part of the scene is variable. This suggests that the meaningful part is both the beginning and end of the scene. So is a FG scene. Namely, the end of FG scenes may be a field scene taken behind a goal post.

Each frame is divided into 4 $\times$ 4 rectangular blocks. *Color distribution* in each block is given as feature parameters. A feature vector is formed as $(R_{1\times1}, G_{1\times1}, B_{1\times1}, \cdots, R_{4\times4}, G_{4\times4}, B_{4\times4})$, where $R_{k\times l}$, $G_{k\times l}$, $B_{k\times l}$ are average RGB values in the kth $\times$ lth block. As shown in Fig.2, we make this vector for each of $N(= N1 + N2)$ image frames.

Finally, we measure the distance between the vector of the shot and that of the *example image sequence*, which may be viewed as a temporal image model of the target event. If the distance is smaller than some threshold, the shot is indexed by the score event. The example sequence is provided for each score event, and is, at present, obtained from the sample stream.

We here comment on the accuracy of indexing with the proposed method. It was tested for three sample streams, each of which is about 3-hour video. The CC stream analysis enabled us to reduce the search length of the visual stream by 87 %. For 40 score events, we obtained the recall rate of 86% and the precision rate of 74% for shot indexing.

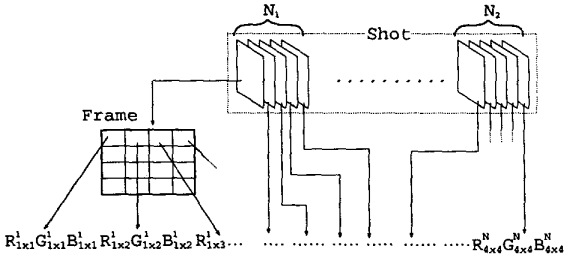

Fig. 2. Visual stream analysis.

Fig. 3. An example of an overlay called CDO.

See [7] for the details of the experimental results.

C. Extraction of textual information

For annotation of event shots, we make full use of specific overlays that appear at times in a fixed subregion of an image frame. We call them *content description objects* (CDOs) [7, 9] because they contain textual information that expresses the on-going scenes. An example of the CDO is displayed in Fig.3. From several sorts of CDOs, we capture the textual information such as 'Quarter No.,' 'Time,' 'Ball position,' 'Offense team,' and 'Score' through character recognition process.

Such information will be exploited as video captions at the next abstraction stage. You should consult [9] for the method of extracting and analyzing the CDOs.

III. VIDEO ABSTRACTION

A. Impact factors of events

An essential problem for video abstraction is to determine important parts of the whole video according to their contents. In general, it is a rather difficult problem. For example, Yeung and Yeo [5] proposed a measure called *dominance* based on repetitive times of similar scenes and length of the scene, but it does not necessarily reflect on semantical contents.

For team sports games of our concern, we have to select some events of importance from all events as highlights of the game. We are greatly interested in how large an event affects the final result. Because the result is obviously either to win or to lose, an event deriving the final result will have a large impact.

To evaluate the importance of events, we introduce an *impact factor* (IMF). Let E_i and $F(E_i)$, $(i = 1, \cdots, M)$ represent an event and its IMF, where the subscript i is arranged in temporal order. To define the IMFs, we consider a state transition diagram as illustrated in Fig.4. Assuming that we deal with a game between team A and team B, we can consider three states about the situation of the game. Let S_T, S_A and S_B denote 'the two teams tie,' 'team A leads,' and 'team B leads,' respectively. φ is a transition function mapping from the state set and the event set onto the state set. Now we define a *state change event* E_j^* such that

$$\varphi(S_n, E_j^*) \neq S_n, \ \ (S_n \in \{S_T, S_A, S_B\}).$$

In Fig.4, the bold arrows indicate the state transition events.

We can easily design a procedure of selecting events by providing a stack L. For $i = 1, \cdots, M$, if E_i^* is a state change event, then push the E_i^* into L. It is satisfied that if $i < j, (i \neq j)$, then $F(E_i^*) < F(E_j^*)$; the IMF's magnitude increases in order of occurrence of the events. The event located at the top of the stack has the largest IMF. This implies that the state change event occurring at the latter or final stage affects the result of the game.

B. Generation of video clip

Recently, He et al. [4] proposed the desirable attributes for video abstraction, characterized by 4Cs: conciseness, coverage, context, and coherence. Following this, we attempt to make an abstracted video clip from the original video stream.

We think of the state change events stored in the stack as the highlights of the game. Recall that the later occurring events have a large IMF. In the abstracted video clip, the highlight events are displayed in reverse order of the stack. For coherence of abstraction, this process assures us that the occurrence of the highlights should be in original temporal order. In addition to these, we make the final event one of the highlights. It is noted that the last highlight but one is the game-winning event.

The generating rules for the abstracted video clip, denoted by **CLIP**, are as follows:

CLIP := **INTRO MAIN ENDING**
INTRO := **INTRO-VC INTRO-SN** fade
ENDING := **ENDING-VC ENDING-SN** fade
MAIN := [**PRE-VC HIGHLIGHT-SN POST-VC** fade]$^+$
INTRO-SN := stadium | team | coach | audience | player | ...
ENDING-SN := player | coach | audience | ...
INTRO-VC := date | location | team-name | ...
ENDING-VC := result | game-stats | ...
PRE-VC := status | quarter | time | ball-position | offense-team |...
POST-VC := score | team | player |...
HIGHLIGHT-SN := [**LIVE-SHOT** | **REPLAY-SHOT**]$^+$

The notation [ ]$^+$ means one and more repetitions. In the above rules, **SN** and **VC** stand for a scene and a video caption, respectively. **SN**'s element is a scene of an object; e.g., stadium's scene. **VC**'s element is a video caption describing the data about an object or a situation; e.g., the final score of the game.

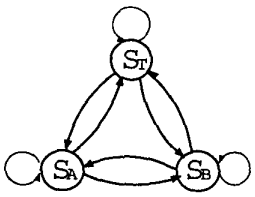

Fig. 4. State transition diagram.

Table 2. Events in the sample stream.

Qtr.	Time	Team A	Team B	Score	Length
1st	3:32	TD,PAT(1)		7-0	41.1 s
	6:18	FG		10-0	
	8:20		TD,PAT(1)	10-7	
	12:27		TD,PAT(1)	10-14	33.3 s
2nd	0:56	TD,PAT(1)		17-14	51.1 s
	6:45	FG		20-14	
	13:49	TD,PAT(1)		27-14	
3rd	12:27		TD,PAT(1)	27-21	
4th	11:50	TD,PAT(2)		35-21	71.0 s

The produced **CLIP** consists of not only actual highlighted scenes but also video captions explaining the scenes. Fade-out operations are inserted between the highlights. In addition, synchronized audio is associated with the shot. The audio is a segment of the original auditory stream corresponding to the span of the shot of concern.

C. Evaluation

To investigate the quality of an abstracted video clip, we provided a sample stream and assumed that all events were correctly indexed. Table 2 shows when the score events took place in the stream. The winner was Team A. The state change events and the final event are underlined.

Although only the **MAIN** part is now generated automatically, it was possible to make an about 3.5-minute video clip from the original video stream of three hours. In this case, the **HIGHLIGHT-SN** for the TD and PAT events was set as all the shots between both events. The scene was composed of live and replay shots. We are developing a method of discriminating between them so that we can select more appropriate scenes as highlights.

Four highlights were extracted from 16 score events in the sample stream as indicated in Table 2. Each length of the highlight scene is also shown in this Table. The total clip length was 206 seconds consisting of 196 seconds for the **HIGHLIGHT-SN** part as well as 10 seconds for the **PRE-VC** and **POST-VC** parts. After a couple of sports enthusiasts watched the generated clip, we received their criticism. A major point is that it included no other events than score events. That is, turnovers such as intercept and fumble recover, or fouls could be sometimes as important as TDs. We should take them into consideration.

In this method, the clip length depends on the number of highlights. Further, the highlight depends on how the game is going on. We have less highlights if the game is one-sided. If the length is asked to be constant and is shorter than that of the **MAIN** part, we have to introduce an alternative way to select the highlights. For this purpose, precise tracking of the game story should be achieved.

IV. CONCLUSION

This paper addressed a method of abstracting sports video by highlights through event based video indexing. The impact factor related to the importance of the event enables us to select the appropriate highlights of the game. As a result, an hour-length video can be compressed into a minute-length clip. At the current stage, the generated video clip is far from actual clips used in TV sports news like FOX or ESPN. Nevertheless, we believe that this method marks a step towards automatically abstracting the sports video.

The remaining problems to be explored are 1) to handle other kinds of events, 2) to improve the reliability of the indexing procedure, 3) to abstract the auditory stream, and 4) to consider an evaluation framework for video abstraction.

Acknowledgments– The author thanks Prof. Kitahashi and the members of the lab for their support. This work is partly supported by a Grant-in-Aid for scientific research from the Japan Society for the Promotion of Science.

References

[1] P. Aigrain, H. J. Zhang and D. Petkovic, Content-based Representation and Retrieval of Visual Media: a State-of-the-Art Review, *Multimedia Tools and Applications*, Vol.3, pp.179-202, 1996.

[2] M. A. Smith and T. Kanade, Video Skimming and Characterization through the Combination of Image and Language Understanding Techniques, *Proc. CVPR97*, pp.775-781, 1997.

[3] R. Lienhart, S. Pfeiffer and W. Effelsberg, Video Abstracting, *Commun. ACM*, Vol. 40, No. 12, pp.55-62, 1997.

[4] L. He, E. Sanocki, A. Gupta and J. Grudin, Auto-Summarization of Audio-Video Presentations, *Proc. ACM Multimedia'99*, pp.489-498, Nov. 1999.

[5] M. M. Yeung and B-L. Yeo, Video Visualization for Compact Presentation and Fast Browsing of Pictorial Content, *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 7, No. 5, pp.771-785, Oct. 1997.

[6] S. Uchihashi, J. Foote, A. Girgensohn and J. Boreczky, Video Manga: Generating Semantically Meaningful Video Summaries, *Proc. ACM Multimedia'99*, pp.383-392, Nov. 1999.

[7] N. Babaguchi, Y. Kawai and T. Kitahashi, Event Based Video Indexing by Intermodal Collaboration, *Proc. Intl' Workshop MISRM'99*, 1999.

[8] N. Babaguchi and R. Jain, Event Detection from Continuous Media, *Proc. 14th ICPR*, Vol.II, pp.1209-1212, 1998.

[9] N. Babaguchi, S. Sasamori, T. Kitahashi and R. Jain, Detecting Events from Continuous Media by Intermodal Collaboration and Knowledge Use, *Proc. IEEE ICMCS'99*, Vol.1, pp.782-786, 1999.

[10] A. Merlino, D. Morey and M. Maybury, Broadcast News Navigation using Story Segments, *Proc. ACM Multimedia'97*, pp. 381-391, 1997.

[11] S.Satoh, Y.Nakamura and T.Kanade, Name-It: Naming and Detecting Faces in News Videos, *IEEE Multimedia*, pp.22-35, 1999.

[12] B. Shahraray and D. C. Gibbon, Automated Authoring of Hypermedia Documents of Video Programs, *Proc. ACM Multimedia'95*, pp. 401–409, 1995.

[13] Y.Chang, W.Zeng, I.Kamel and R.Alonso, Integrated Image and Speech Analysis for Content-based Video Indexing, *Proc. IEEE ICMCS'96*, 1996.