Event Detection and Highlight Detection of Broadcasted Game Videos

Wei-Ta Chu
National Chung Cheng University
Min-Hsiung, Chiayi, Taiwan
wtchu@ccu.edu.tw

Yung-Chieh Chou
National Chung Cheng University
Min-Hsiung, Chiayi, Taiwan
andy0983011@gmail.com

ABSTRACT

Efficient access of game videos is urgently demanded due to the emergence of live streaming platforms and the explosive numbers of gamers and viewers. In this work we facilitate efficient access from two aspects: game event detection and highlight detection. By recognizing predefined text displayed on screen when some events occur, we associate game events with time stamps to facilitate direct access. We jointly consider visual features, events, and viewer's reaction to construct two highlight models, and enable compact game presentation. Experimental results show the effectiveness of the proposed methods. As one of the early attempts on analyzing broadcasted game videos from the perspective of multimedia content analysis, our contributions are twofold. First, we design and extract game-specific features considering visual content, event semantics, and viewer's reaction. Second, we integrate clues from these three domains based on a psychological approach and a data-driven approach to characterize game highlights.

Categories and Subject Descriptors

I.4.8 [**Artificial Intelligence**]: Scene Analysis—*Color, motion, object recognition*; I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*Video analysis*

General Terms

Experimentation, Performance, Human Factors

Keywords

Event detection, highlight detection, game video analysis

1. INTRODUCTION

Online live streaming platforms like UStream[1], Livestream[2], and Twitch[3] emerge rapidly in recent years. Various streaming videos, including edited video programs like movies and TV shows, and live broadcast of events like sports, festival, and video games, attract millions of users and cause heavy network traffic. In year 2014 there were 100 millions unique users per month watching 16 billion minutes of streaming video on Twitch, and over 11 millions videos were broadcasted per month. In this work we will take game videos, particularly *League of Legend* (LoL) broadcasted on Twitch, as an instance to conduct analysis.

LoL is a multiplayer online battle arena video game developed by Riot Games[4]. It is one of the most popular PC games recently. As of Jan. 2014, there are 27 million gamers playing per day, and over 7.5 million gamers playing at the same time during each day's peak play time. Riot Games organizes the League of Legends Championship Series in many countries and continents. The biggest championship series attracted 32 million viewers online and granted the champion one million US dollars. This is a big business attracting tremendous amounts of gamers and viewers. From the viewpoint of multimedia research, explosive number of broadcasting videos and rich viewer behaviors give rise to significant demands as well as research opportunities on efficient game video access, retrieval, and summarization.

Although there have been many works on video event detection and highlight extraction, studies focusing on game videos are relatively fewer. An LoL game usually lasts for 30–50 minutes, while important events like a battle or someone being slain only occur in very short periods. A highly interactive interface showing detailed events and timestamps is thus urgently demanded, which has been an important alternative to show game progress by text in sports games of Major League Baseball (MLB) and National Basketball Association (NBA). By associating game time and important events, viewers can click events of interest and jump to a specific point to watch the game.

In this work, we make an early attempt to automatically detect events and highlights in game videos, in order to facilitate efficient access and compact representation. Contributions of this paper are summarized as follows.

- Event detection: Predefined messages are shown on screen when special game events occur. We detect events through detecting and recognizing text displayed on screen, and then construct an index linking events and time stamps of game videos. An interface to facilitate direct access at the event level thus can be built.

- Highlight detection: Important events, prominent visual effects, as well as viewer's reaction are jointly considered to detect highlight parts of a game. Game highlight like the

[1]http://www.ustream.tv
[2]https://new.livestream.com/
[3]http://www.twitch.tv/

[4]http://www.riotgames.com

HCMC'15, October 30, 2015, Brisbane, Australia.
 ISBN 978-1-4503-3747-2/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2810397.2810398.

ones edited by professional reporters can be automatically generated to facilitate efficient browsing.

The rest of this paper is organized as follows. Section 2 provides literature survey on live streaming, game video analysis, and video event detection. Section 3 describes details of event detection. Feature extraction and highlight models are described in Section 4. Section 5 provides performance evaluation from various perspectives, and Section 6 concludes this work.

2. RELATED WORKS

In this section we review related literature from three perspectives: live streaming systems, game video analysis, and video event detection and summarization.

2.1 Live Streaming Systems

As the emergence of live streaming systems, many works have been proposed to study such systems from various perspectives. Kaytoue et al. [12] focused on electronic sports videos streamed by Twitch and advocated that much potential revenue can be made to professional gamers, casters, and streaming platforms. They also showed that number of viewers is predictable and explainable. Pires and Simon [18] presented a dataset consisting of data collected from two main user-generated live streaming systems, i.e., Twitch and live service of YouTube. With this rich dataset, they studied overall bandwidth, number of unique channels, and popularity distribution in these systems. In [17], some observations from Twitch motivated them to implement adaptive bitrate streaming in order to reduce delivery bandwidth and to increase quality of experience of viewers. Hamilton et al. [9] presented an ethnographic investigation of the live streaming of video games on Twitch. They interviewed several Twitch users and found that difficulty of interaction influenced user's feeling. They explored the design problems and the implications of streaming systems to be clues to improve not only the Twitch streaming system but also other streaming services.

2.2 Game Video Analysis

Studies designed for game videos, especially from the perspective of visual analysis, are quite few. Here we survey literature related to visual analysis for game videos. Douglass [6] utilized several image processing and computer vision techniques to show gameplay recording. For example, keyframes of game recording are shown in a grid manner, and many frames are superimposed to create average images showing recurrent visual artifacts. Lewis et al. [13] analyzed player's actions, such as actions per minute and spatial variance of action, to discover the correlation between actions and winning games. Not surprisingly, they found that gamers able to most quickly execute actions tend to win. Rioult et al. [20] extracted topological clues, such as the area of polygon where players move and the inertia of the team, to predict outcomes of multiplayer online battle arena games. Riegler et al. [19] developed a set of tools like zoom and drawing to annotate computer game videos, so that users can communicate general concepts of a specific game.

2.3 Video Event Detection and Summarization

Video event detection, highlight extraction, and summarization have been widely studied for years. In this section, we briefly describe some of the most recent works based on video genres, including sports videos, movies, TV shows, and consumer videos. Generally, no matter which video genre, the research trend starts from purely content-based analysis to adoption of external knowledge such as webcast text and social media. Most recently, crowdsourcing techniques and multiple resource integration also enable more advanced analysis.

The Bagadus system [21] seamlessly integrated data from multiple cameras mounted in a stadium and data from sensors on soccer players, and provided a real-time interaction subsystem for experts to annotate soccer events. This system enables a user to follow particular player(s), view events in the representation of panorama videos, and create video summaries. Annotating events by experts is expensive, and thus Sulser et al. [22] proposed to adopt the crowdsourcing technique to integrate annotations from crowd workers. A Bayesian network-based method was proposed to detect events for soccer videos in [23]. The proposed method captured dependencies among extracted features, based on the automatically learnt joint distribution of variables. In addition to conventional highlight events like goals and penalty kicks, Nguyen and Yoshitaka [16] proposed to include scenes of intensive competition and emotional moments in soccer video summaries. They measured interest level of a video clip based on cinematographic features and motion features. To annotate baseball videos, Chiu et al. [3] aligned high-level webcast text with video content in order to avoid unstable performance caused by purely content-based methods. For basketball videos, Hu et al. [11] proposed a robust player tracking system, and adopted player trajectories to detect highlight events and to conduct tactic analysis. Chen and Chen [2] proposed a framework consisting of scoreboard detection, text/video alignment, and replay detection for basketball videos.

For movies, Evangelopoulos et al. [8] modeled time-varying perceptual importance of movies by fusing multimodal saliency, including auditory saliency derived from frequency analysis, visual saliency derived from intensity and color, and linguistic saliency derived from part-of-speech tagging. The multimodal saliency is then used to develop a generic video summarization algorithm. Tsai et al. [24] mined relationship between role-communities, and developed a movie summarization method based on social power of role-communities. Lu et al. [15] summarized movies from the auditory perspective. Important audio events such as cheer, laugh, and gunshot were detected and concatenated to form video summaries.

Duan et al. [7] detected events in consumer videos by leveraging a large number of loosely labeled web images from multiple sources. The developed a decision function to select most relevant source domains and achieved much performance gain in event recognition. Dang and Radha [5] proposed a entropy-based measure of the heterogeneity of image patches, and utilized its temporal variation to achieve key frame extraction and video skimming for consumer videos. The idea of sparse coding reconstruction was adopted to do consumer video summarization in [4] and [25], while the former [4] used the entire video for reconstruction and needed much computation, and the latter [25] largely reduced computational cost by learning a dictionary by group sparse coding.

3. EVENT DETECTION

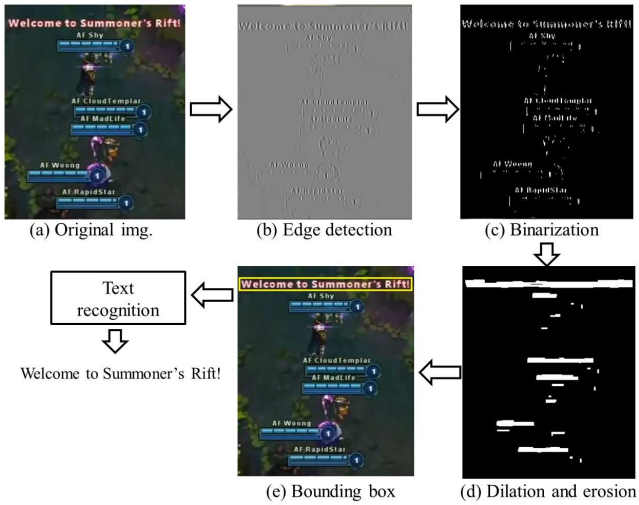
In LoL, predefined text messages are displayed on screen when important events occur. Based on detected events, we can develop a highly interactive interface. Figure 1 shows the process of event detection. For each video frame, we first apply the Sobel edge detector to extract edges (Figure 1(b)), and conduct binarization to filter out weak edges (Figure 1(c)). By morphological operations including dilation and erosion (Figure 1(d)), more noisy edge pixels are filtered out, and the bounding boxes of connected edge pixels are determined. Boxes that are too small are discarded (Figure 1(e)).

We employ the Tesseract OCR package[5] to recognize text in

[5]https://code.google.com/p/tesseract-ocr/

Table 1: Text showing important events in LOL. The terms (A) and (B) could be replaced with gamer's name.

S_1	Welcome to Summoner's Rift!	S_2	Thirty seconds until minions spawn.
S_3	Minions have spawned.	S_4	First Blood!
S_5	Double Kill!	S_6	Shut Down!
S_7	(A) has slain (B)!	S_8	(A) is on a killing spree!
S_9	(A) is on a Rampage!	S_{10}	(A) is Unstoppable!
S_{11}	(A) is Dominating!	S_{12}	(A) is Godlike!
S_{13}	(A) is Legendary	S_{14}	The red team has slain the Dragon!
S_{15}	The red team has slain Baron Nashor!	S_{16}	(A) has destroyed a blue turret!
S_{17}	(A) has destroyed a blue inhibitor!	S_{18}	A minion has destroyed a blue turret!



(a) Original img. (b) Edge detection (c) Binarization (d) Dilation and erosion (e) Bounding box

Figure 1: Flowchart of event detection.

each detected bounding box. Let $W = \{w_1, ..., w_M\}$ be the set of recognized words. We compare W with predefined messages $S_1, ..., S_{18}$ shown in Table 1. The bounding box with W is recognized to represent event i^* if $i^* = \arg\max_i |W \bigcap S_i|$. If recognized words in a box do not match with any of S_i, this box is viewed as noise and discarded.

The detected text regions are often with cluttered background, which impedes accurate text recognition. Figure 2 shows samples of detected regions. The two regions at the top row are with clear background, while the two at the bottom row are with cluttered background. To reduce the influence of noises, we collect the matching results and determine the occurred event by majority voting. The event text is usually displayed for 3 to 4 seconds. If a text region is recognized to contain S_i, we would check the recognition results of frames in the following 4 seconds. This 4-second clip is determined to have event j^* if $j^* = \arg\max_j \Delta_j$, where $\Delta_j = \sum_{k=1}^{K} \delta(f_k, S_j)$ and $\delta(f_k, S_j) = 1$ if the frame f_k contains text S_j.

4. HIGHLIGHT DETECTION

To detect highlight in game videos, we extract features from both video and viewer's chats, and then construct highlight models based on two approaches, i.e., arousal model and support vector machine.

4.1 Features

In this work, whether a video segment is attractive is described by three types of features: visual features like motion intensity and frame dynamics, event features indicating occurrence of events,

Figure 2: Good (top row) and bad (bottom) detected text regions.

and chat features derived from viewers' chat logs. We divide a given game video into pieces of one-second segments, and extract these features from each of them.

- Motion intensity (G_1). Generally there is more motion when important events occur, e.g., battle between two groups of gamers. We estimate motion between consecutive video frames by the optical flow method, and calculate sum of motion magnitude as motion intensity between video frames. The average intensity value over all frames in a one-second segment is finally obtained as G_1. Larger motion intensity usually indicates more interesting content.

- Frame dynamics (G_2). Special effects often appear when gamers invoke special attack skills, and appearance of frames would suddenly change. We extract color histogram difference between frames, and the average frame difference over all frames in a one-second segment is calculated as G_2. Usually larger frame dynamics indicates more interesting content.

- Event ratio (G_3). Important events certainly indicates highlighted content. Based on events that have already detected by the method mentioned in Sec. 3, we calculate the number of frames conveying an event in a one-second segment. The ratio of the number of event frames to 30 frames (30 fps in our game videos) is calculated as G_3. A larger ratio indicates that more events occur in this segment, and this segment is thus more attractive.

- Number of gamers (G_4). Numbers of gamers in a battle is a strong clue to show how important this battle is. More gamers indicate more violent or more important events. Because gamer's name is always shown right above the character (Figure 1(a)), we calculate the number of gamers shown on screen by detecting text regions showing gamers' names. The average number of gamers over all frames of a one-second segment is then calculated as a feature.

- Number of viewers chat (G_5). In addition to visual content, we also extract clues from chat logs given by viewers in online broadcasting. Chats directly reflect how viewers perceive the events just happened. More viewers chat or say praising words like "What a shot" or "WOW" when important events occur. We thus calculate the number of speaking viewers per second as a feature. Note that the burst of chats emerge after important events occur. Therefore, the feature extracted from the tth segment actually indicates the importance of the $(t-b)$th segment. In the evaluation section, we will show this effect with varied b's.

- Number of emotion symbols (G_6). Twitch designs several emotion symbols to let viewers quickly express their feeling about the visual content. We calculate the average number of emotion symbols over all frames of a one-second segment as the feature.

These features represent different clues to detect highlight, and will be integrated by the proposed highlight models in the following.

4.2 Highlight Models

We model game highlight based on two approaches: the psychophysiological approach based on the arousal model [10], and the data-driven approach based on support vector machine (SVM). From psychophysiological experiments, when a user watches a video, the level of arousal rises as a consequence of an increase in sound and motion intensity. In this work, we investigate integrating several clues to build the arousal model. On the other hand, we can collect features extracted from highlighted/non-highlighted segments and view highlight detection as a classification problem with the help of SVM.

Before highlight detection, we detect shot change boundaries based on color histogram difference and edge change ratio [14]. We use one-second video segment as the unit for feature extraction and arousal model construction, and use video shot as the unit for SVM model construction.

4.2.1 Arousal Model

The concept of the arousal model is that the level of arousal of a user rises as a consequence of increase of various stimuli, in the representation of various features in this work. For example, we can view the temporal evolution of G_1 constitutes a curve showing how motion intensity drives a user's arousal. Evolutions of six features, therefore, constitute six curves. To make these curves comparable and smooth, a smooth process with normalization is applied to each curve. These normalized curves are then combined to show the integrated arousal evolution, which provides important clues for us to select highlight video segments. Details of the arousal model are described in the following.

Evolution of each feature mentioned above describes the level of arousal from one perspective. We jointly consider all perspectives by integrating these evolutions. Inspired by [10], the level of arousal of the kth video segment can be described as

$$A(k) = F(\hat{G}_i(k)), i = 1, ..., 6; k = 1, ..., N, \quad (1)$$

where $\hat{G}_i(k)$ is the ith feature value G_i at the kth second after the smooth process, and the function F is for integrating excitements from various perspectives. The smooth process is defined as

$$\hat{G}_i(k) = \frac{\max_k(G_i(k))}{\max_k(\tilde{G}_i(k))} \times \tilde{G}_i(k), \quad (2)$$

where $\tilde{G}_i(k)$ is the result of the convolution of the curve $G_i(k)$ with a Kaiser window of the length and shape parameter l and β, respectively, i.e., $\tilde{G}_i(k) = G_i(k) * K(l_i, \beta_i)$. The smooth process is designed to account for the degree of memory retention, and ensures that arousal does not change abruptly for consecutive video segments. The normalization defined in eqn. (2) makes different curves comparable. The integration function F in eqn. (1) can simply be a linear combination, and is defined as $F(\hat{G}_i(k)) = \frac{1}{6}\sum_{i=1}^{6}\hat{G}_i(k)$.

Figure 3 shows examples of arousal curves obtained from the six features and the final integration curve. From the final arousal curve we can extract highlight by selecting appropriate video segments associated with high arousal values. Given a test video, we extract features from all one-second segments and construct arousal curves, and H highlight parts are detected by selecting H highest peaks of the integrated arousal curve. Suppose that the jth video segment v_j corresponds to the ith selected peak, the video segments preceding (v_p's) or following (v_q's) the jth video segment v_j are all selected as in the ith highlight H_i, if they all belong to the same video shot $\mathcal{S}_j$:

$$H_i = \{v_p, v_j, v_q | v_p \in \mathcal{S}_j, v_q \in \mathcal{S}_j, p = j-1, j-2, ...; q = j+1, j+2, ...\}. \quad (3)$$

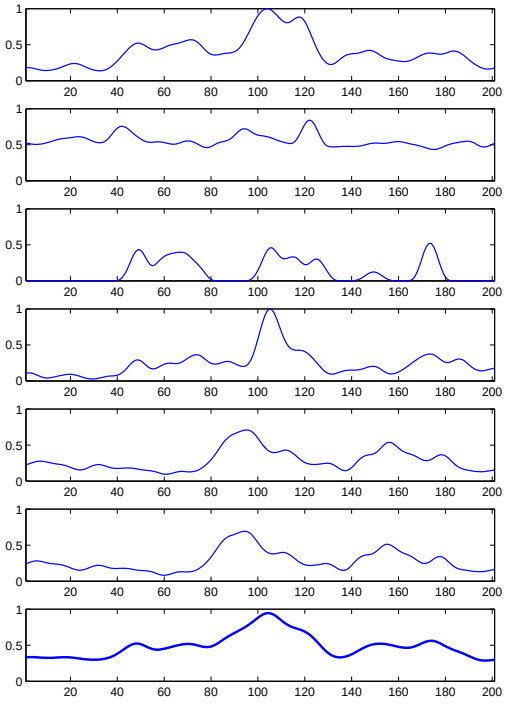


Figure 3: Arousal curves constructed from different features. 1st row to 6th row: $\hat{G}_1(k)$ $\hat{G}_6(k)$; 7th row: the integrated curve $A(k)$.

4.2.2 SVM Model

In this model, we view highlight detection as a classification problem, i.e., classify each test segment as a highlight or not. Here we use video shot as the analysis unit, and describe a shot by features extracted from one-second segments belonging to the same video shot. Mean, maximum, minimum, variance, and dynamic range (maximum minus minimum) of motion intensity (G_1) in this shot, for example, are calculated. The five features derived from G_1 to G_6, respectively, are concatenated as a 30-dimensional feature vector to represent a video shot. We collect highlight/non-highlight video shots as positive/negative examples, and utilize the libSVM package [1] to train an SVM classifier with probability estimation. Given an unknown video shot, we estimate its probability of being a highlight. H highlights can thus be detected when we pick the video shots with the H highest probabilities.

5. EVALUATION

5.1 Evaluation Dataset

We collected 24 games of 2014 League of Legends World Champion broadcasted by Twitch, which can be grouped into six game series, and each series comprises multiple games between two teams. As the highlight ground truth, we downloaded highlight of each game that was edited and released by a team[6] constituted by professional gamers. By comparing highlight with the raw game video, we know the timestamps of each highlight part in the original video.

[6]https://www.youtube.com/user/Kazawuna/about

Table 2: Statistics of the evaluation dataset.

Videos	Avg. Length (mm:ss)	Avg. Length of Highlight (sec.)
2014 NWS VS OMG GAME 1–3	51:28	550
2014 SHR VS EDG GAME 1–5	40:27	459
2014 SHR VS OMG GAME 1–5	44:08	594
2014 SHR VS SSW GAME 1–4	34:36	441
2014 SSB VS C9 GAME 1–4	42:28	628
2014 SSB VS SSW GAME 1–3	35:42	517
Average	40:35	531

(a) NWS VS OMG (b) SHR VS EDG

(c) SHR VS OMG (d) SHR VS SSW

(e) SSB VS C9 (f) SSB VS SSW

Figure 4: Snapshots of each game series.

Table 2 shows average lengths of six game series and the edited highlights[7]. There are more than 16 hours of game videos, associated with 3.54 hours of highlight segments in total. To evaluate performance of event detection, we also obtained manually detected events corresponding to each game from a computer game website[8]. Availability of these data in different web platforms shows that these games widely attract gamers around the global. These datasets will be publicly available soon later.

5.2 Event Detection

We manually examine overlap between the ground truth and the detected events. Table 3 shows performance of event detection, in terms of average recalls of different game series and the overall average recall. As can be seen in this table, over 90% important events can be accurately detected by the proposed method.

Note that the number of events detected by this system is often larger than that in the manually edited ones. The reason is that the manual edition only consists of big events due to space limitation, and sometimes multiple events are summarized as a single big event. To facilitate efficient browsing, we tend to list every event so that viewers can select anyone of interest. We thus demonstrate recall rate rather than precision rate here.

5.3 Highlight Detection

Performance measurement. Performance of highlight detection is measured in terms of precision, recall, and F-measure. Let $\mathcal{G}$ denote the set of true highlights and $\mathcal{D}$ the set of detected highlights. The precision rate is calculated as $p(C_i, C_j) = |C_i \cap C_j|/C_j$, where $C_i \in \mathcal{G}$ and $C_j \in \mathcal{D}$. The notation $|\cdot|$ denotes the length in term of seconds. The recall rate is calculated as $r(C_i, C_j) = |C_i \cap C_j|/C_i$. By jointly considering precision and recall, the F-measure F is calculated as:

$$F = \frac{1}{Z} \sum_{C_i \in \mathcal{G}} |C_i| \max_{C_j \in \mathcal{D}} \{f(C_i, C_j)\}, \quad (4)$$

$$f(C_i, C_j) = \frac{2 \times p(C_i, C_j) \times r(C_i, C_j)}{p(C_i, C_j) + r(C_i, C_j)}, \quad (5)$$

The value $Z = \sum_{C_i \in \mathcal{G}} |C_i|$ is the normalization factor. Higher F means better clustering performance.

Feature Settings. In the following, we first evaluate the influence of different feature settings on highlight detection based on the arousal model. As mentioned in Sec. 4.1, chats usually largely emerge after some events happen for a while. We evaluate F-measure of detection results based on different backtrack settings, i.e., the parameter b mentioned in Sec. 4.1, of the features G_5 and G_6, and show performance variations in Figure 5. From this figure, the best detection performance can be achieved if we backtrack 20 seconds for G_5 and 11 seconds for G_6, respectively. That means, G_5 and G_6 of the tth video segment (at the tth second) is actually extracted from the text at the $(t+20)th$ second and at the $(t+11)th$ second, respectively. The difference between settings for G_5 and G_6 is not surprising, because viewers usually type predefined emotion symbols just after some events happen, and then type text comment. Based on this result, we use the backtrack settings in the following experiments.

Generally three types of features are used in highlight detection, i.e., visual features (G_1 to G_3), event feature (G_4), and viewer's chat features (G_5 and G_6). We separately evaluate each type of features as well as jointly consider all of them, and show highlight detection performance in Table 4. Comparing the first three rows shows that visual features are the most robust, which is not surprising because viewer's reaction (chat) is unlimited and is often noisy. By jointly considering all features, the best highlight detection performance can be obtained.

Smooth Settings. The influence of Kaiser window size, which simulates human's short-term memory and reduces noise, on highlight detection performance is shown in Fig. 6. Overall, detection performance is sharply raised as the window size increases from 5

Table 3: Performance of event detection.

Videos	Avg. Recall
2014 NWS VS OMG GAME 1–3	0.89
2014 SHR VS EDG GAME 1–5	0.93
2014 SHR VS OMG GAME 1–5	0.92
2014 SHR VS SSW GAME 1–4	0.89
2014 SSB VS C9 GAME 1–4	0.96
2014 SSB VS SSW GAME 1–3	0.95
Average	0.92

Table 4: Performance of highlight detection based on the arousal model with different feature settings.

Features	Precision	Recall	F-measure
G_1, G_2, G_3 (visual)	0.501	0.584	0.540
G_4 (event)	0.495	0.540	0.517
G_5 and G_6 (chat)	0.437	0.538	0.482
G_1 to G_6	**0.545**	**0.626**	**0.583**

[7]NWS, OMG, SHR, EDG, SSW, SSB, and C9 are all team names.
[8]http://www.tgbus.com/

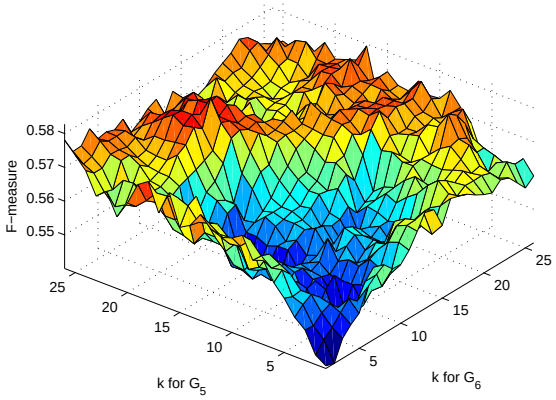


Figure 5: Backtracking parameters vs. precision of highlight detection.

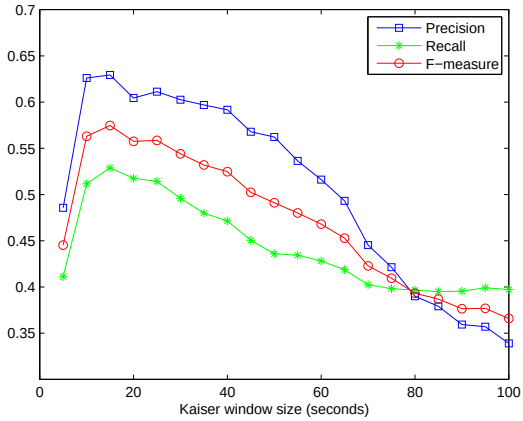


Figure 6: Kaiser window size vs. performance of highlight detection.

seconds to 15 seconds, and gradually decreases as the window size enlarges more. From this figure we set the size of Kaiser window as 15 seconds in the following experiments.

Peak Selection and Segment Boundaries. After the integrated arousal curve is constructed, appropriate number of peaks are selected and the corresponding video segments are extracted as game highlights. Therefore, two problem arises: how many peaks should we pick, and how to determine the video segment corresponding to each peak? We design two schemes to deal with these problems.

- Scheme 1: From statistics of the evaluation dataset, there are averagely 16 highlight segments in a game, and each segment averagely lasts 34 seconds. Therefore, we develop a baseline scheme by selecting the 16 highest peaks from a game. For a peak located at the tth second, the video clip ranging from $t-17$ seconds to $t+17$ seconds is extracted as the highlight. Sixteen extracted video clips corresponding to the 16 selected peaks are concatenated as the final highlight for a game.

- Scheme 2: Let $\{P_1, ..., P_N\}$ denote the set of N peaks that have already been selected. The average peak value is calculated as $\bar{P} = \frac{1}{N}\sum_{i=1}^{N} P_i$. The $(N+1)$th highest peak P_{N+1} will be selected as a part of game highlight if $P_{N+1} > 0.8 \times \bar{P}$. We sequentially select highest peaks until the next highest peak is not higher than the adaptive threshold. For a peak located at the tth second, we select the video shots containing the tth second as the highlight. This selection design makes boundaries of highlight segments coincide with shot boundaries.

Table 5: Performance of highlight detection based on the arousal model with different peak selection schemes.

Features	Precision	Recall	F-1 measure
Scheme 1	0.486	0.653	0.557
Scheme 2	0.545	0.626	0.583

Table 6: Performance of highlight detection based on the arousal model and the SVM model.

Features	Precision	Recall	F-1 measure
Baseline	0.199	0.291	0.236
Arousal model	0.545	0.626	0.583
SVM model	0.520	0.821	0.637

Table 5 shows highlight detection performance based on two different peak selection schemes. As can be seen, although Scheme 1 gives highlights that best match with the average statistics from the training data, the adaptive designs in Scheme 2 yields much better precision and thus yields better detection performance in terms of F-measure.

Arousal model vs. SVM model. After the investigation mentioned above, we adopt the best feature settings to extract features, and accordingly construct the SVM model for highlight detection. In the following experiment, we adopt the five-fold cross validation scheme to evaluate the SVM approach. Table 6 shows highlight detection performance comparison between the baseline method, the arousal model, and the SVM model (train and test with the five-fold cross validation scheme). In the baseline model, we randomly select a video shot as a highlight shot, and repeat this intuitive selection procedure until the total length of all selected videos reaches the average highlight length obtained based on the evaluation dataset. We see that both the arousal model and SVM model significantly outperforms the baseline. Comparing the arousal model with the SVM model, although both models yield similar precision rates, we see that the SVM model achieves much higher recall rate. In the arousal model, we smooth feature values to construct arousal curves. The smooth operation reduces noises but also causes information loss, and might be the reason of lower recall rate. Based on this experiment, we conclude that the SVM model generally outperforms the arousal model.

Tolerance on detection performance. In eqn. (4) and eqn. (5), precision, recall, and F-measure can be unity only when the detected highlight is 100% overlapped with the ground truth. However, highlight segments edited by professional reports usually do not start or end at video shot boundaries. They usually cut the video shots at the time instant right before/after important events occur. Viewers are usually satisfied with detected highlights with more than 50% overlapping with the "real highlights". Motivated by the evaluation setting used in the PASCAL VOC challenge, where a correct detection is claimed if the detected object is overlapped with the ground truth by over 50%, we evaluate detection performance variations with various tolerance settings. The tolerance factor T is 0 in eqn. (4) and eqn. (5), and $T = 0.3$ if we allow a detected highlight with more than 70% overlapping with the ground truth a correct detection.

Figure 7 shows performance variations based on different tolerance factors. As we expect, performance increases if we al-

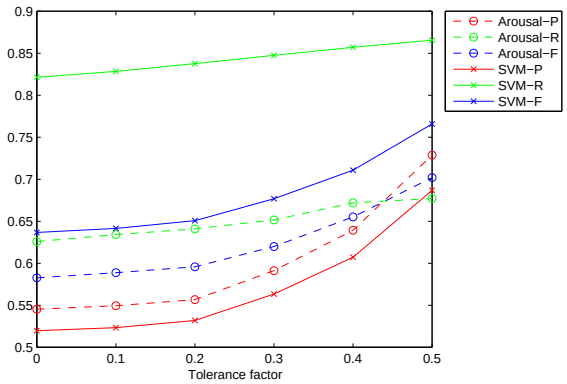


Figure 7: Detection performance vs. different tolerance settings.

low more tolerance. For the arousal model and the SVM model, the F-measures respectively increase to 0.70 and 0.77 if we allow more than 50% overlapping as correct detection. These results are promising and show the effectiveness of both highlight models.

6. CONCLUSION

We have presented event detection and highlight detection to facilitate efficient access of broadcasted game videos. Through recognizing predefined text displayed on screen, we detect events to ease direct access. For highlight detection, we describe visual appearance, events, and viewer's reaction, and then construct two highlight models. Evaluation on famous game videos shows that the proposed methods yield accurate event detection and promising highlight extraction performance. In the future, results of event detection or highlight extraction can cooperate with streaming platforms for smart streaming.

Acknowledgement. The work was partially supported by the Ministry of Science and Technology in Taiwan under the grants MOST103-2221-E-194-027-MY3 and MOST104-2221-E-194-014.

7. REFERENCES

[1] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(27):1–27, 2011.

[2] C.-M. Chen and L.-H. Chen. Novel framework for sports video analysis: A basketball case study. In *Proceedings of IEEE International Conference on Image Processing*, pages 961–965, 2014.

[3] C.-Y. Chiu, P.-C. Lin, S.-Y. Li, T.-H. Tsai, and Y.-L. Tsai. Tagging webcast text in baseball videos by video segmentation and text alignment. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(7):999–1013, 2012.

[4] Y. Cong, J. Yuan, and J. Luo. Towards scalable summarization of consumer videos via sparse dictionary selection. *IEEE Transactions on Multimedia*, 14(1):66–75, 2012.

[5] C. T. Dang and H. Radha. Heterogeneity image patch index and its application to consumer video summarization. *IEEE Transactions on Image Processing*, 23(6):2704–2718, 2014.

[6] J. Douglass. Computer visions of computer games: Analysis and visualization of play recordings. In *Proceedings of Workshop on Media Arts, Science, and Technology: The Future of Interactive Media*, 2009.

[7] L. Duan, D. Xu, and S.-F. Chang. Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1338–1345, 2012.

[8] G. Evangelopoulos, A. Zlatintsi, A. Potamianos, P. Maragos, K. Rapantzikos, G. Skoumas, and Y. Avrithis. Multimodal saliency and fusion for movie summarization based on aural, visual, and textual attention. *IEEE Transactions on Multimedia*, 15(7):1553–1568, 2013.

[9] W. A. Hamilton, O. Garretson, and A. Kerne. Streaming on twitch: Fostering participatory communities of play within live mixed media. In *Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 1315–1324, 2014.

[10] A. Hanjalic and L.-Q. Xu. Affective video content representation and modeling. *IEEE Transactions on Multimedia*, 7(1):143–154, 2005.

[11] M.-C. Hu, M.-H. Chang, J.-L. Wu, and L. Chi. Robust camera calibration and player tracking in broadcast basketball video. *IEEE Transactions on Multimedia*, 13(2):266–279, 2011.

[12] M. Kaytoue, A. Silva, L. Cerf, W. Meira Jr., and C. Raissi. Watch me playing, i am a professional: a first study on video game live streaming. In *Proceedings of International Conference Companion on World Wide Web*, pages 1181–1188, 2012.

[13] J. M. Lewis, P. Trinh, and D. Kirsh. A corpus analysis of strategy video game play in starcraft: Broodwar. In *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, pages 687–692, 2011.

[14] R. Lienhart. Comparison of automatic shot boundary detection algorithms. In *Proceedings of SPIE Storage and Retrieval for Still Image and Video Databases VII*, volume 3656, pages 290–301, 1999.

[15] T. Lu, Y. Weng, and G. Wang. Audiotory movie summarization by detecting scene changes and sound events. In *Proceedings of International Conference on Pattern Recognition*, pages 756–760, 2014.

[16] N. Nguyen and A. Yoshitaka. Soccer video summarization based on cinematography and motion analysis. In *Proceedings of IEEE International Workshop on Multimedia Signal Processing*, 2014.

[17] K. Pires and G. Simon. Dash in twitch: Adaptive bitrate streaming in live game streaming platforms. In *Proceedings of Workshop on Design, Quality and Deployment of Adaptive Video Streaming*, pages 13–18, 2014.

[18] K. Pires and G. Simon. Youtube live and twitch: A tour of user-generated live streaming systems. In *Proceedings of ACM Multimedia Systems Conference*, pages 225–230, 2015.

[19] M. Riegler, M. Lux, V. Charvillat, A. Carlier, R. Vliegendhart, and M. Larson. Videojot: A multifunctional video annotation tool. In *Proceedings of ACM International Conference on Multimedia Retrieval*, pages 534–537, 2014.

[20] F. Rioult, J.-P. Metivier, B. Helleu, N. Scelles, and C. Durand. Mining tracks of competitive video games. *AASRI Procedia*, 8:82–87, 2014.

[21] H. K. Stensland, V. R. Gaddam, M. Tennoe, E. Helgedagsrud, M. Naess, H. K. Alstad, A. Mortensen, R. Langseth, S. Ljodal, O. Landsverk, C. Griwodz, P. Halvorsen, M. Stenhaug, and D. Johansen. Bagadus: An integrated real-time system for soccer analytics. *ACM*

Transactions on Multimedia Computing, Communications, and Applications, 10(1s):Article No. 14, 2014.

[22] F. Sulser, I. Giangreco, and H. Schuldt. Crowd-based semantic event detection and video annotation for sports videos. In *Proceedings of ACM Workshop on Crowdsourcing for Multimedia*, pages 63–68, 2014.

[23] M. Tavassolipour, M. Karimian, and S. Kasaei. Event detection and summarization in soccer videos using bayesian network and copula. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(2):291–304, 2014.

[24] C.-M. Tsai, L.-W. Kang, C.-W. Lin, and W. Lin. Scene-based movie summarization via role-community networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(11):1927–1940, 2013.

[25] B. Zhao and E. P. Xing. Quasi real-time summarization for consumer videos. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 2513–2520, 2014.