Event Detection and Highlight Detection of Broadcasted Game Videos

Wei-Ta Chu
National Chung Cheng University
Min-Hsiung, Chiayi, Taiwan
wttchu@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 UStream1, Livestream2, and Twitch3 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 Games4. 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

---

1http://www.ustream.tv
2https://new.livestream.com/
3http://www.twitch.tv/
4http://www.riotgames.com
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. Fries 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. They developed a decision function to select most relevant source domains and achieved much performance gain in event recognition. Dang and Radha [5] proposed an 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 gamers’ names.

<table>
<thead>
<tr>
<th>$S_i$</th>
<th>Welcome to Summoner’s Rift!</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{1}$</td>
<td>Minions have spawned.</td>
</tr>
<tr>
<td>$S_{2}$</td>
<td>Welcome to Summoner’s Rift!</td>
</tr>
<tr>
<td>$S_{3}$</td>
<td>Double Kill!</td>
</tr>
<tr>
<td>$S_{4}$</td>
<td>A minion has destroyed a blue turret!</td>
</tr>
<tr>
<td>$S_{5}$</td>
<td>Shut Down!</td>
</tr>
<tr>
<td>$S_{6}$</td>
<td>First Blood!</td>
</tr>
<tr>
<td>$S_{7}$</td>
<td>(A) has destroyed a blue turret!</td>
</tr>
<tr>
<td>$S_{8}$</td>
<td>(B) is on a Rampage!</td>
</tr>
<tr>
<td>$S_{9}$</td>
<td>(A) has destroyed a blue turret!</td>
</tr>
<tr>
<td>$S_{10}$</td>
<td>(B) is Unstoppable!</td>
</tr>
<tr>
<td>$S_{11}$</td>
<td>(A) is Dominating!</td>
</tr>
<tr>
<td>$S_{12}$</td>
<td>(A) is Godlike!</td>
</tr>
<tr>
<td>$S_{13}$</td>
<td>The red team has slain Nashor!</td>
</tr>
<tr>
<td>$S_{14}$</td>
<td>The red team has slain the Dragon!</td>
</tr>
<tr>
<td>$S_{15}$</td>
<td>A minion has destroyed a blue turret!</td>
</tr>
<tr>
<td>$S_{16}$</td>
<td>The red team has slain Baron Nashor!</td>
</tr>
<tr>
<td>$S_{17}$</td>
<td>(A) has destroyed a blue turret!</td>
</tr>
<tr>
<td>$S_{18}$</td>
<td>A minion has destroyed a blue turret!</td>
</tr>
</tbody>
</table>

Figure 1: Flowchart of event detection.

![Event Detection Flowchart](image)

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 $t$th 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 arises 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 $\tilde{G}_i$ 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 $k$th video segment can be described as

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

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

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

where $\hat{G}_i(k)$ is the result of the convolution of the curve $G_i$ with a Kaiser window of the length and shape parameter $\alpha$ and $\beta$, respectively, i.e., $G_i(k) = G_i(k) * K(\alpha, \beta)$. 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 $\hat{F}$ in eqn. (1) can simply be a linear combination, and is defined as

$$\hat{F}(\tilde{G}_i(k)) = \frac{1}{6} \sum_{i=1}^{6} \tilde{G}_i(k).$$

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 highlight parts are detected by selecting $H$ highest peaks of the integrated arousal curve. Suppose that the $j$th video segment $v_j$ corresponds to the $i$th selected peak, the video segments preceding ($v_p$’s) or following ($v_q$’s) the $j$th video segment $v_j$ are all selected as in the $i$th highlight $H_i$, if they all belong to the same video shot $S_j$:

$$H_i = \{v_p, v_j, v_q | v_p \in S_j, v_q \in S_j, p = j - 1, j - 2, \ldots; q = j + 1, j + 2, \ldots\}.$$ 

Figure 3: Arousal curves constructed from different features. 1st row to 6th row: $G_1(k)$ to $G_6(k)$; 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 ($G_i$) 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.

<table>
<thead>
<tr>
<th>Videos</th>
<th>Avg. Length (mm:ss)</th>
<th>Avg. Length of Highlight (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014 NWS VS OMG GAME 1–3</td>
<td>51.28</td>
<td>550</td>
</tr>
<tr>
<td>2014 SHR VS EDG GAME 1–5</td>
<td>40.72</td>
<td>459</td>
</tr>
<tr>
<td>2014 SHR VS OMG GAME 1–3</td>
<td>34.06</td>
<td>354</td>
</tr>
<tr>
<td>2014 SHR VS SSW GAME 1–4</td>
<td>34.16</td>
<td>441</td>
</tr>
<tr>
<td>2014 SSB VS C9 GAME 1–4</td>
<td>32.28</td>
<td>628</td>
</tr>
<tr>
<td>2014 SSB VS SSW GAME 1–3</td>
<td>35.42</td>
<td>517</td>
</tr>
<tr>
<td>Average</td>
<td>40.35</td>
<td>531</td>
</tr>
</tbody>
</table>

Figure 4: Snapshots of each game series.

Table 3: Performance of event detection.

<table>
<thead>
<tr>
<th>Videos</th>
<th>Avg. Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014 NWS VS OMG GAME 1–3</td>
<td>0.89</td>
</tr>
<tr>
<td>2014 SHR VS EDG GAME 1–5</td>
<td>0.93</td>
</tr>
<tr>
<td>2014 SHR VS OMG GAME 1–3</td>
<td>0.92</td>
</tr>
<tr>
<td>2014 SHR VS SSW GAME 1–4</td>
<td>0.89</td>
</tr>
<tr>
<td>2014 SSB VS C9 GAME 1–4</td>
<td>0.96</td>
</tr>
<tr>
<td>2014 SSB VS SSW GAME 1–3</td>
<td>0.95</td>
</tr>
<tr>
<td>Average</td>
<td>0.92</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Features</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_1, G_2, G_3$ (visual)</td>
<td>0.501</td>
<td>0.584</td>
<td>0.540</td>
</tr>
<tr>
<td>$G_4$ (event)</td>
<td>0.495</td>
<td>0.540</td>
<td>0.517</td>
</tr>
<tr>
<td>$G_5$ and $G_6$ (chat)</td>
<td>0.437</td>
<td>0.358</td>
<td>0.482</td>
</tr>
<tr>
<td>$G_1$ to $G_6$</td>
<td>0.545</td>
<td>0.626</td>
<td>0.583</td>
</tr>
</tbody>
</table>

is measured in terms of precision, recall, and F-measure. Let $G$ denote the set of true highlights and $D$ the set of detected highlights. The precision rate is calculated as $p(C_i, C_j) = |C_i \cap C_j|/|C_j|$, where $C_i \in G$ and $C_j \in D$. The notation $|C_i|$ 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:

$$F = \frac{1}{2} \sum_{C_i \in G} \max_{C_j \in D} \frac{f(C_i, C_j)}{\max f(C_i, C_j)},$$

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

The value $Z = \sum_{C_i \in G} |C_i|$ 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 back track 20 seconds for $G_5$ and 11 seconds for $G_6$, respectively. That means, $G_5$ and $G_6$ of the 5th video segment (at the 5th 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

NWS, OMG, SHR, EDG, SSW, SSB, and C9 are all team names.

http://www.tgbus.com/
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 highlights. 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 \( t \)th 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 \( \bar{P} = \frac{1}{N} \sum_{t=1}^{N} P_t \). 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 \( t \)th second, we select the video shots containing the \( t \)th second as the highlight. This selection design makes boundaries of highlight segments coincide with shot boundaries.

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-

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

<table>
<thead>
<tr>
<th>Features</th>
<th>Precision</th>
<th>Recall</th>
<th>F-1 measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheme 1</td>
<td>0.485</td>
<td>0.633</td>
<td>0.557</td>
</tr>
<tr>
<td>Scheme 2</td>
<td>0.545</td>
<td>0.626</td>
<td>0.583</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Features</th>
<th>Precision</th>
<th>Recall</th>
<th>F-1 measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.199</td>
<td>0.291</td>
<td>0.236</td>
</tr>
<tr>
<td>Arousal model</td>
<td>0.545</td>
<td>0.626</td>
<td>0.583</td>
</tr>
<tr>
<td>SVM model</td>
<td>0.520</td>
<td>0.821</td>
<td>0.637</td>
</tr>
</tbody>
</table>
low more tolerance. For the arousal model and the SVM model, the F-mea-sures 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


