Efficiently detecting outlying behavior in video-game players

YoungBin Kim, Shin Jin Kang, SangHyeok Lee, Jang Young Jung, Hyeong Ryeol Kam, Jung Lee, YoungSun Kim, Joonsoo Lee, Chang Hun Kim

In this paper, we propose a method for automatically detecting the times during which game players exhibit specific behavior, such as when players commonly show excitement, concentration, immersion, and surprise. The proposed method detects outlying behavior based on the characteristics of game players. These characteristics are captured non-invasively in a general game environment. In this paper, cameras were used to analyze observed data such as facial expressions and player movements. Moreover, multimodal data from the game players (i.e., data regarding adjustments to the volume and the use of the keyboard and mouse) was used to analyze high-dimensional game-player data. A support vector machine was used to efficiently detect outlying behavior. We verified the effectiveness of the proposed method using games from several genres. The recall rate of the outlying behavior pre-identified by industry experts was approximately 70%. The proposed method can also be used for feedback analysis of various interactive content provided in PC environments.

Efficiently detecting outlying behavior in video-game players

Abstract

In this paper, we propose a method for automatically detecting the times during which game players exhibit specific behavior, such as when players commonly show excitement, concentration, immersion, and surprise. The proposed method detects outlying behavior based on the characteristics of game players. These characteristics are captured non-invasively in a general game environment. In this paper, cameras were used to analyze observed data such as facial expressions and player movements. Moreover, multimodal data from the game players (i.e., data regarding adjustments to the volume and the use of the keyboard and mouse) was used to analyze high-dimensional game-player data. A support vector machine was used to efficiently detect outlying behavior. We verified the effectiveness of the proposed method using games from several genres. The recall rate of the outlying behavior pre-identified by industry experts was approximately 70%. The proposed method can also be used for feedback analysis of various interactive content provided in PC environments.

Young Bin Kim[1], Shin Jin Kang[2], Sang Hyeok Lee[3], Jang Young Jung[2], HyeongRyeol Kam[1], Jung Lee[3], YoungSun Kim[3], JoonSoo Lee[4], Chang Hun Kim[3]*

[1]Interdisciplinary Program in Visual Information Processing, Korea University, Seoul, Korea

[2]School of Games, Hongik University, Seoul, Korea

[3]Department of Computer and Radio Communications Engineering, Korea University, Seoul, Korea

[4]AI Laboratory, NC Soft, Korea

* Corresponding author

E-mail:

Telephone: +82 – 2 – 3290 – 3199

1. Introduction

It is considerably useful for game developers to have the ability to predict whether game players will react in a way that corresponds to the developers' intentions. Such predictions allow developers to verify that their intended design is correct, and this verification process can enhance the overall game(*Bauckhage et al. 2012; Bernhaupt 2015; Isbister & Schaffer 2008; Ravaja et al. 2004; Sánchez et al. 2012*). When games are being designed, developers benefit from checking whether game players react with excitement, concentration, immersion, and surprise, and from whether they exhibit these emotions at the intended time. However, this is especially difficult to determine(*Bernhaupt 2015*).

To check for such reactions, a large game company (viz., one that develops AAA games) organized a quality-assurance team. The team measures game players' reactions in a sealed space called the user-experience test room (UX Room). In the UX room, the team records the test on video in a separate, isolated location to detect specific player behavior using a heart-rate monitor and electromyography(*Bernhaupt 2015; Mirza-Babaei et al. 2013*). When the test participants finish playing, game developers and analysts gather to observe the feedback from the participants, after which a survey is conducted(*Bernhaupt 2015; El-Nasr et al. 2013; Vermeeren et al. 2010*).

When analyzing such data, the core content concerns the degree of interest and the concentration of players. Detecting outliers based on a time series, such as joyful or bored reactions at unexpected times, provides crucial feedback for game development. Based on this feedback, games can be redesigned during their development and even after their release(*Bernhaupt 2015; Hazlett 2008; Ravaja et al. 2004; Tammen & Loviscach 2010*). However, this approach requires many people over a long period, and small independent game companies, for instance, cannot afford to adopt this approach. Moreover, the process is not automated, making it difficult to understand the behavior of multiple players.

In this paper, we define “outlying behavior” as excitement, concentration, immersion, or surprise commonly exhibited by game players at specific times during gameplay. The proposed method is not an invasive method that requires additional equipment, such as a heart-rate monitor or electromyography, to measure this outlying behavior. Rather, we propose a non-invasive method that allows this behavior to be detected automatically without additional equipment. Several factors required to detect outlying behavior in a game environment were modeled in the proposed system. Multimodal data was collected, including keyboard strokes, mouse inputs, volume adjustments, and data from game players observed with webcams and PC cameras.

With the proposed system, 30 facial feature points are defined based on the active-appearance model (AAM)(*Cootes et al. 2001*) using data obtained from the observation of game players and their movements. Additionally, an artificial neural network was developed based on these facial feature points (viz., the eyebrow, eye, nose, and mouth)(*Karthigayan et al. 2008; Thai et al. 2011*) to detect basic facial expressions such as happiness, sadness, anger, surprise, disgust, and fear(*Li 2001; Picard & Picard 1997*). In addition, keyboard strokes, mouse inputs, and auditory changes in the immediate environment were collected and used as factors to detect specific behavior. Based on this game-player multimodal data collected in the game environment and the specific behavior expected by industry experts at particular times, we designed a classifier to analyze specific behavior at particular times using a support vector machine (SVM)(*Chang & Lin 2011*). We verified the proposal using actual games, demonstrating that specific player behavior can be classified efficiently.

The characteristics of the proposed system are as follows:

- We propose a non-invasive method that has few measurement errors and is friendly to game players. We solve an incompatibility issue related to the measuring environment using additional equipment, such that outlying behavior can be detected in a general game environment.

- We propose a new model that can detect the outlying behavior of game players during gaming. Unlike existing methods that use fragmentary analysis in a separate environment, our proposed method detects outlying behavior based on multimodal data from players during gameplay.
- With our method, it is possible to detect outlying behavior in an environment where multiple players are participating simultaneously. Compared to existing methods that analyze a considerable amount of data from only a few players, outlying behavior can be analyzed with our method even when several players participate in the game concurrently.
- We verified the significance of the proposal by applying it to actual commercial video games. Existing methods are in many cases limited to only a few particular genres, or they focus on a theoretical design for checking players' reactions. The proposed method was evaluated using popular games (e.g., Tomb Raider, BioShock Infinite, and League of Legends) from several genres.

2 Related Work

Research regarding the collection of player data within games and the use of this data for analysis is a field that is becoming more prevalent as the game market develops(*Bernhaupt 2015; El-Nasr et al. 2013; Mirza-Babaei et al. 2011; Mirza-Babaei et al. 2013*). Player data is generally classified into survey-based opinion data and play data (or log data) from players acquired during gameplay(*Bauckhage et al. 2012; Bauckhage et al. 2014; Bernhaupt 2015; Chittaro et al. 2006; Medler et al. 2011; Thawonmas & Iizuka 2008; Wallner & Kriglstein 2012*). To collect this data, a crawling tool is generally used, and log data can often be customized to collect a specific kind of data. Alternatively, players can be observed directly in order to analyze their behavior(*Canossa et al. 2011; Drachen et al. 2014; Zoeller 2010*). In this paper, outlying behavior is detected based on multimodal data—using log data to determine the players' behavior along with data collected from observation. In this section, the relevant research is discussed.

Many studies pertaining to existing games, research, and tests for determining users' reactions rely on manually collected self-reports from the players themselves (i.e., interviews, questionnaires, focus groups, etc.) or manual video analysis of experimental subjects(*Bernhaupt 2015; El-Nasr et al. 2013; Pagulayan et al. 2002; Vermeeren et al. 2010*). These studies are reliable and easy to conduct. User reactions can be accurately deduced, and the analysis becomes easier as user data accumulates. Although research has been conducted by detecting outlying behavior in games based on this method, users must invest ample time in addition to game playing for these self-reports, and developers must spend a lot of time analyzing the reports. Our research focuses on an automated method that does not require additional time for analysis.

Research that involves observing users with biometric equipment to automatically detect outlying behavior has also been conducted(*Bernhaupt 2015; Bersak et al. 2001*). The use of biometric data is helpful for video-game analysis(*Bernhaupt 2015; Kivikangas et al. 2011; Vermeeren et al. 2010*). In particular, researchers have used biometric factors such as the heart rate, skin conductance, facial electromyography, and electroencephalography(*Guger et al. 2004; Hazlett 2008; Mirza-Babaei et al. 2013; Nacke & Lindley 2010; Nakasone et al. 2005; Nogueira et al. 2013*). Some research has proposed methods for determining the user's reaction based on expanded data adopted from the field of psychophysiology(*Herbelin et al. 2004; Kivikangas et al. 2011; Lang 1995; Ravaja & Kivikangas 2008; Ravaja et al. 2004; Ravaja et al. 2006*). Such methods can predict a user's reaction objectively, continuously, precisely, and sensitively. Moreover, these predictions can be made in real-time. However, special equipment that is not used in gaming environments is required, and it is considerably difficult to process data from multiple users. This paper aims to determine the user's reaction with a non-invasive method for detecting outlying behavior.

Additionally, methods have been proposed for detecting the common outlying behavior of users by visualizing user data within a game environment based on user logs(*Bauckhage et al. 2012;*

Bauckhage et al. 2014; Bernhaupt 2015; Chittaro et al. 2006; El-Nasr et al. 2013; Medler et al. 2011; Thawonmas & Iizuka 2008; Wallner & Kriglstein 2012). These visualization methods aim to analyze player progress (e.g., the elapsed time and the location of a player's death) to balance the gameplay(*Bernhaupt 2015; Vermeeren et al. 2010*). Such methods benefit from having access to an abundance of user data. This data is easy and straightforward to obtain. However, many such methods are limited to particular games or genres, and they cannot be easily integrated with other genres and games.

Based on this related research, we propose a method that can efficiently detect outlying behavior in a non-invasive manner, based on an automated user-observation method and log data. In Section 3, we explain our proposal for detecting outliers, and we describe a classifier using user data. The experiment we conducted to evaluate the proposed method is detailed in Section 4, and the experimental results are provided in Section 5. Finally, we discuss these results and offer our conclusion in Section 6.

3 Methods

3.1 Defining outlying behavior

The proposed method is designed to detect outlying behavior such as excitement, concentration, immersion, and surprise that is commonly exhibited at specific times by the players in a game environment. A diverse range of player data is collected in the game environment in order to detect this outlying behavior. To determine the various types of data needed, we first conducted an experiment to extract common player reactions.

We asked three AAA developers to identify the game events they considered important in two commercial games (viz., Tomb Raider and Action Adventure). To determine how players react to each of these events, we randomly selected 30 experimental subjects online. We observed the outlying behavior of these 30 subjects during gameplay. After playing the games, the dominant behavior exhibited at each of the events was judged to be the outlying behavior.

Common player behavior can be summarized by noting increased body movements, facial contortions, widely opened eyes, raised eyebrows, opened mouths, blinking patterns, noises emitted by the user, and changes in the use of the keyboard or mouse. There were cases where these particular actions occurred independently, but most occurred as a composite of these actions. The complexity of these compound actions makes it difficult to check for such cases automatically. In the following section, we illustrate our method for detecting this complex behavior.

3.2 System

To detect outlying behavior, this system detects 30 input parameters in real-time. Based on these parameters, the system utilizes a machine-learning-based outlier classifier by employing an SVM to analyze the upper level of behavior (see Fig. 1).

Fig 1. System overview. The outlier classifier is generated based on the multimodal data of users collected in a PC game environment.

This system does not require the use of biometric factors such as skin conductance, facial electromyography, and electroencephalography, because our aim is to apply the proposal to a general PC gaming environment. Therefore, the parameters are detected with slightly less accuracy, but data can be collected more easily from multiple players. The types of parameters detected are listed below.

- Facial feature points: 30 feature points are extracted based on the active appearance model (AAM), and these feature points are categorized into 24 groups. Next, the angle variations of the characteristics in these 24 groups are measured.
- Facial expressions: Facial expressions are classified into six groups: neutral, happy, sad, angry, surprised, and disgusted(*Li 2001; Picard & Picard 1997*). These are classified

using an artificial neural network based on the facial feature points.

- Keyboard: Keyboard usage is measured in strokes per unit of time.
- Mouse: Mouse input is measured in terms of the distance covered by the mouse cursor per unit of time.
- Volume: The relative volume is measured according to the volume of the initial environment.

Each of the above items is detailed in the following sub-sections.

3.2.1 Facial Feature Points

To identify the facial expressions in players, this study detected facial feature points and established methods for tracing them. Research is available on the detection of such facial feature points (*Wang et al. 2014*) and it has been used in various fields(*Gu & Ji 2004; Kang et al. 2014; Kaulard et al. 2012; Li et al. 2013; Pejsa & Pandzic 2009*). There are a number of commercial solutions related to these research results(*Pantic & Bartlett 2007*). Facial feature points are modeled using an active appearance model (AAM). AAM is one of the best-known facial feature extraction techniques, and it has been described in various works(*Cootes et al. 2001; Gao et al. 2010; Gross et al. 2005*). Research involving AAM-based facial feature point detection(*Anderson et al. 2013; Fanelli et al. 2013; Martins et al. 2013; Tzimiropoulos & Pantic 2013; Wang et al. 2014*) has been conducted using various techniques. We detected players' facial feature points through a generic version of the AAM, rather than with person-independent AAM(*Cootes et al. 2001*). For learning, this study used variations in the 30 feature points associated with the eyes, nose, mouth, eyebrow movements, and blinking movements (see Fig. 2). The applicable modules were implemented using OpenCV(*Bradski 2000*).

Fig 2. 30 facial feature points used for facial-expression and blinking recognition (left), and all 24 feature points constructed from these 30 feature points for measuring angle variations (right).

As parameters, we used the variations in movement that occur between the previous time-step and the current time-step in the 30 facial feature points during common player behavior. To analyze these variations in movement, associated feature points are grouped to measure the angle that is formed in the applicable part. The 30 feature points are condensed into 24 feature points for the calculation (all of the groups for the right eyebrow, left eyebrow, right eye, left eye, nose, and mouth are composed of an entire edge, start edge, middle edge, and end edge). The internal angle formed between three feature points connected within a set is then used. With this process, the movement of common parts that are composed of feature points and their local movements can be detected efficiently, regardless of the player's location. In the case of blinking, if the distance value between feature points at the upper and lower part of the eye is reduced to zero, this movement is regarded as a blink.

Because a player's face loses its feature points in pitch, yaw, and roll situations, parameters are not sufficiently guaranteed in such cases. Indeed, research has been conducted on the AAM to resolve this problem(*Anderson et al. 2013*). We used a simpler approach, because our study focuses on real-time analysis. To measure the facial rotation of players based on the feature points, we created a triangle that connects the tip of the player's nose to the sides of both eyes (Fig. 3). The aim is to identify the direction in which the player's face is rotating, by analyzing changes in the triangle's internal angles. When feature points are rotated, they are weighted to the same extent as the movement variations of the adjacent feature points for calculation.

Fig 3. To identify the direction of a player's rotation and to detect lost feature points, a triangle is created to connect feature points on both sides of the eye and the tip of the nose. The direction of rotation is defined according to the triangle's internal angles, and weights are applied if feature points are lost in the applicable state.

3.2.2 Facial Expression

In cases involving facial expressions, we used facial feature points from previous research efforts that employed artificial neural networks (ANNs) to conduct facial analysis(*Karthigayan et al. 2008; Thai et al. 2011*). Thai et al.(*Thai et al. 2011*) performed edge detection using the Canny algorithm to recognize facial expressions with an ANN. Thai et al.(*Thai et al. 2011*) did not extract feature points from all faces, but instead analyzed local features that might have a direct impact on facial expressions involving the eyes, mouth, brow, and other facial areas. Similarly, we divided 30 facial feature points into five groups to recognize facial expressions. An applicable group consists of feature points that may have the largest impact on changes in facial expressions in general(*Kanade et al. 2000*)(see Fig. 4). For the ANN model, we used the sum of the movement variations of the feature points that belong to the applicable group of the five facial feature points defined above in the five input neurons. The model also contained a hidden layer of 20 neurons and 7 output neurons. The output comprises mutually exclusive binary bits representing an emotion (neutral, happy, sad, angry, surprised, disgusted, and fearful). To perform ANN learning, datasets were created using training data collected from a group of ten participants. Regarding the seven facial expressions, data was collected twice from each facial expression of the participants, and this data was then used for learning. Using the datasets, tests were conducted with ten-fold cross validation to determine whether an accuracy rate of 70% or greater could be achieved. Based on these results, the facial recognizer is prepared beforehand.

Our model does not require highly accurate facial feature points or facial expressions. Rather, it is designed merely to detect outlying behavior based on these features. Indeed, a state-of-the-art method could also be applied to achieve more accurate results.

Fig 4. All 30 feature points divided into 5 groups. The sum of the variations in movement for the applicable group is used as an ANN learning input node for facial-expression recognition.

3.2.3 Log data and environmental sound

Log data based on the keyboard and mouse usage is significantly relevant for identifying the status of players. Using this data, we can establish the degree of the player's immersion in a game and measure the level of difficulty felt by players(*Kang et al. 2014; Kim et al. 2013*). This can be used to better understanding how the UI is used. With our proposal, we measured keyboard and mouse usage per second during gameplay. The keyboard is used more often when players are concentrating and less often when they are frustrated. Thus, the keyboard plays an important role in detecting outlying behavior.

To record mouse usage, we measured the distance covered by the mouse cursor each second. With this information, it is possible to identify the player's concentration level with additional indirect measurements of the player's main eye points. We identified whether outlying behavior was likely to occur based on the movement of the mouse.

Finally, we measure the environmental sound. This sound can be obtained easily using the microphone on the webcam. We used the NAudio sound processing package to measure the volume with the microphone, and we recorded the change in volume from its initial value. The impact of sounds was relatively small, but it was nevertheless considered because it represents a significant component of the users' reactions.

3.2.4 Classification

Learning was performed with an SVM based on player parameters that belong to marked events, along with player parameters extracted in other frames for these parameters. Our aim is not to analyze the intricacies of the players' behavior, but rather to establish the times during gameplay when certain outlying behavior is exhibited. Therefore, it is appropriate to use a SVM. The data we used for learning consisted of a few instances of outlier data, and a larger volume of general behavior data. To balance the data, we used the Synthetic Minority Over-sampling Technique (SMOTE)(*Chawla et al. 2002*) to perform oversampling on the outlier data during the learning process. We used LibSVM(*Chang & Lin 2011*) and RBF Kernel ($K(x,y) = e^{-\gamma\|x-y\|^2}$)-based learning. Using LibSVM, the learning data was cross-validated in order to search for optimum parameters for the RBF Kernel. This method is exceptionally efficient, based on a relatively small amount of learning data. Although this study uses an SVM, we found that the use of other two-class learning techniques or classification algorithms produce similar results. The outlier classification process is designed to be customizable.

4 Experimental Material

To conduct the experiment, a total of 18 participants were recruited. The participants were sought online with anonymous participation requests, and they were allowed to inspect the program for online system verification. The program was automatically implemented to collect multimodal data from the participants (keyboard and mouse usage, game screen, facial expression, volume, and player movement) during the game used in the experiment. The data was collected after the participants finished playing the game. The timeline of actual games was detected using the game screen, and only data corresponding to particular events was used in the experiments. The 18 participants were instructed to provide learning data for the individualized outlier classifiers in advance of the experiment. Tomb Raider—an action-adventure first-person shooter (FPS) genre—was played for 25-30 minutes in order for the outlier classifier to learn individually. This reduced deviations between individuals that occurred mainly during the emotive tests administered in advance.

To acquire learning data, we utilized data obtained during previous gameplay. And facial signals and player log data that appear during those times were used as learning data. These events where outliers are likely to occur were selected in advance by experts in the field (viz., AAA developers). Identical gameplay videos were played five times by three developers, and the expected user reactions were checked during the events. Events that were marked twice or more were judged as events where outliers were likely to occur. However, for the multiplayer online battle arena

(MOBA) genre, where the same game results cannot be guaranteed among all players, the time at which the player dies was checked. Further data regarding the correspondence of the timelines can be found in the supporting materials.

To constitute a learning group, approximately 2.5 s of data was marked as an outlier event, beginning at the point where the outlier occurs. Other non-outlier events were marked as general behavior. During non-outlier events, abnormal behavior can occur, and such behavior can reduce the learning accuracy. However, the impact of such behavior is relatively minor, judging from our experiments. Based on the applicable video, facial signals were detected within a 1 s time window, using the methods explained above. Prior to detection, the player's initial states were stored in 5 s increments. Empirically, the time unit was set to 5 s because there were an unacceptable number of errors in the data itself when the initial state was set to less than 5 s. The relative angle variations per second among the 24 facial feature points were used for learning, beginning from the initial state. Additionally, facial expressions, the frequency of blinking, player log data, and the sound volume were used for learning.

The 18 participants were divided into three groups prior to the experiment. To collect the data required for prediction, the groups were allotted 15 min., 25 min., and 45 min., respectively. Each group played Bio Shock Infinite from the action-adventure FPS genre, and League of Legends from the MOBA genre. By assigning the participants to play games from different genres, the experiments resulted in responses that were more diverse.

5 Experimental Results

The system was created using the methods explained above. Then, four different experiments were conducted. The first experiment was devised to verify the performance of the individualized classifiers. The second and third experiments were designed to determine the efficiency of the outlier detection using other players' classifiers, rather than individualized classifier customized for the player. The fourth experiment was designed to determine whether the individualized outlier classifier provides detection efficiently with different game genres. The respective aims of the experiments are summarized as follows (see Fig. 5):

Experiment #1: Verification of the individualized outlier classifier

Experiment #2: Verification of outlier classification using classifiers from other players

Experiment #3: Verification of an outlier classifier integrated for all players

Experiment #4: Verification of outlier classification in games from other genres

Fig 5. Experimental design overview.

The first experiment was intended to determine whether the outlier detection was conducted properly, using the individualized outlier classifiers trained in advance by the participants. Using these classifiers, an experiment was conducted to find outliers among players who played a game from same genre (viz., BioShock Infinite). In the game timeline that was provided by experts (as explained above), the expected outlier events were pre-defined to determine whether these events matched the events that the proposed system predicted. If matches occurred one or more times within the pre-defined 2.5 s outlier event, these were regarded as successfully predicted. When outlier events were predicted based on individualized outlier classifiers, there were slight differences between each group. The group that collected learning data for 15 min. achieved a recall rate of 73.31% (SD = 10.72). The group that collected learning data for 25 min. achieved a recall rate of 75.29% (SD = 5.4). The group that collected learning data for 45 min. achieved a recall rate of 77% (SD = 4.64). These results are listed in Table 2. Thus, better results were obtained as the learning period increased. However, adjustments were unnecessary because the differences between these groups were insignificant. Participants who generated low recall rates tended to be “poker face”-type players with unusual reactions during gameplay. Such players constituted an extremely small fraction of the participants, and their behavior can nevertheless be detected to some extent.

The second experiment applied individualized outlier classifiers to other players. The purpose of this experiment was to determine the efficiency of the proposed technique, even when individualized outlier classifiers were not used. In each group, one participant's outlier was detected using individualized outlier classifiers for the other five individuals within the group. Five tests were conducted for each of the 18 participants. The group that collected learning data for 15 min. achieved a recall rate of 64.17% (SD = 6.73). The group that collected learning data for 25 min. achieved a recall rate of 63.94% (SD = 5.43). The group that collected learning data for 45 min. achieved a recall rate of 68.26% (SD = 5.98). These results are listed in Table 2. We thus concluded that the results are less accurate when individualized outlier classifiers are not used.

In the third experiment, the parameters from the classifiers of the 18 subjects were integrated into a single classifier to predict outlier events. As a result of these experiments, it was found that the outlier recall rate for the entire group of participants was 66.54% (SD = 3.69). Altogether, false-positive results increased and the accuracy of the results decreased. Moreover, outlier classifiers were created for each group corresponding to the learning process. As a result, when the outliers for individual players were analyzed, the group that was allotted 45 min. of learning time produced the most accurate results. As indicated here, outlier events can be predicted to some extent, even if individual learning machines for each individual do not exist, and the accuracy of such predictions improves as the learning data accumulates.

The fourth experiment determined whether outlier detection could be performed properly for a game from a different genre (whether an FPS or MOBA game), using individualized outlier classifiers. When performing tests in these genres, log data such as mouse inputs and keyboard strokes were not used. In League of Legends, events do not occur according to a timeline. Therefore, predictions were made by marking the time of a character's death as the expected time for an outlier occurrence. When individualized detectors were used, the results revealed that, for the entire group of 18 participants, the recall rate was 69.70% (SD = 6.24). Similar results were obtained even when each participant's integrated outlier classifier was used. This suggests that the proposed technique can be utilized with various genres. Log data was not used for learning, because of mouse and keyboard patterns specific to the genre. However, this data could be used to obtain improved results if the proposal were trained in a single genre exclusively. In particular, in MOBA games such as League of Legends, the player's log data changed significantly whenever characters died. Therefore, such data utilization has the potential to produce better results.

Table 1. Results for Experiments 1–4

6 Discussion and Conclusions

In this paper, we proposed a player-friendly, non-invasive method for detecting the outlying behavior of players during gameplay. The proposed method detects outlying behavior in environments where several game players participate simultaneously, and its significance was verified through experiments using actual commercial games. Our small-scale experiment demonstrated that the proposed system efficiently detects outlying behavior exhibited by individual players. The experiment showed the proposal's feasibility for reducing the manual work of game developers, although not by a considerable amount. Our system can be utilized by the gaming industry for evaluating games, adjusting the difficulty of levels, and balancing game designs. If the system is used for evaluating games, it can simplify the development process significantly.

With existing methods, subjects must be recruited to participate in an extensive experiment. Based on external observations or participant surveys, data coding and analysis must be performed by specialized analysts, and this typically requires weeks to complete. Evaluations based on our proposed system can be completed using an independent detection program that runs concurrently alongside the game in order to collect data for automatic analysis and visualization. This simplification thus reduces the time and human resources required. Furthermore, our method has fewer restrictions in terms of time and location, because additional equipment is unnecessary.

We introduced our proposal to ten professional game developers (some of whom were independent, whereas other worked for AAA companies). In general, their feedback was positive:

one developer noted that the proposal “showed good effects with minimum equipment” and that he was “willing to use [the proposal] at the development stage”; another reports that“it has a lot of potential for improving game design.” Interestingly, one of the developers told us that our method would be beneficial when using data selectively and for adjusting weights.

Our system can run in a web-based environment, because relatively little data is required for processing. Near-real-time processing is possible when using a classifier that is generated beforehand. Therefore, we are able to collect general player data and game-developer data. Such base data can be used to generate large-scale data that extends beyond the small-scale participant-based testing that is currently performed. Based on this information, it is expected that new discoveries will be made in areas such as outlier detection, facial expressions, and concentration. Thus, the research results will improve in turn.

The proposed system is also expected to expand the research results, because it can be used to analyze long-term content in games, videos, and other programs. Currently, the system is designed exclusively for PC environments. However, the performance of smartphones is gradually approaching that of PCs, and the former typically contain built-in cameras. Therefore, we shall pursue additional research to apply our system to smartphones. Moreover, the use of the system can be expanded to applications for smart TVs and mobile environments. Other methods can enhance the system performance by combining other bio-signal detection methods(*Nummenmaa et al. 2014*) and by increasing the efficiency of emotive extraction. Likewise, the results can be improved using deep learning.

This technique targeted the detection of outlying behavior. However, it can also be employed in other fields, such as bio-signal-based child and youth indulgence- prevention systems. We believe that the proposed technique can be used in such fields, and that it can be expanded for applications in other fields.

Regarding future research directions, we intend to devise a method for processing data that is continuously accumulating, and to design system improvements based on this data. To utilize such accumulating data, we shall improve the current algorithm-reliant supervised learning system such that it can use a semi-supervised learning algorithm. By exploiting existing log-data visualization studies, we shall investigate a means for predicting the status of players (i.e., character deaths, elapsed playing time, etc.) with the ability to predict outlier events and without the need for experts to index the game's timeline. Further, although the player's keyboard and mouse input data focuses exclusively on usage, improvements shall be designed to identify the player's game situation by analyzing player chats using repetitive pattern analysis.

References

Anderson R, Stenger B, Wan V, and Cipolla R. **2013**. Expressive visual text-to-speech using active appearance models. Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on: IEEE. p 3382-3389. DOI: 10.1109/CVPR.2013.434.

Bauckhage C, Kersting K, Sifa R, Thurau C, Drachen A, and Canossa A. **2012**. How players lose interest in playing a game: An empirical study based on distributions of total playing times. Computational Intelligence and Games (CIG), 2012 IEEE Conference on: IEEE. p 139-146. DOI: 10.1109/CIG.2012.6374148.

Bauckhage C, Sifa R, Drachen A, Thurau C, and Hadiji F. **2014**. Beyond heatmaps: Spatio-temporal clustering using behavior-based partitioning of game levels. Computational Intelligence and Games (CIG), 2014 IEEE Conference on: IEEE. p 1-8. DOI: 10.1109/CIG.2014.6932865.

Bernhaupt R. **2015**. *Game User Experience Evaluation*: Springer.

Bersak D, McDarby G, Augenblick N, McDarby P, McDonnell D, McDonald B, and Karkun R. **2001**. Intelligent biofeedback using an immersive competitive environment. Paper at the Designing Ubiquitous Computing Games Workshop at UbiComp.

Bradski G. **2000**. The opencv library. *Doctor Dobbs Journal* 25:120-126.

Canossa A, Drachen A, and Sørensen JRM. **2011**. Arrrgghh!!!: blending quantitative and qualitative methods to detect player frustration. Proceedings of the 6th International Conference on Foundations of Digital Games: ACM. p 61-68. DOI: 10.1145/2159365.2159374.

Chang C-C, and Lin C-J. **2011**. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2:27.

Chawla NV, Bowyer KW, Hall LO, and Kegelmeyer WP. **2002**. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16:321-357.

Chittaro L, Ranon R, and Ieronutti L. **2006**. Vu-flow: A visualization tool for analyzing navigation in virtual environments. *IEEE Transactions on Visualization and Computer Graphics* 12:1475-1485.

Cootes TF, Edwards GJ, and Taylor CJ. **2001**. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence* 23:681-685.

Drachen A, Thurau C, Sifa R, and Bauckhage C. **2014**. A comparison of methods for player clustering via behavioral telemetry. *arXiv preprint arXiv:14073950*.

El-Nasr MS, Drachen A, and Canossa A. **2013**. *Game analytics: Maximizing the value of player data*: Springer Science & Business Media.

Fanelli G, Dantone M, and Van Gool L. **2013**. Real time 3D face alignment with random forests-based active appearance models. Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on: IEEE. p 1-8. DOI: 10.1109/FG.2013.6553713.

Gao X, Su Y, Li X, and Tao D. **2010**. A review of active appearance models. *IEEE transactions on systems, man and cybernetics Part C, Applications and reviews* 40:145-158.

Gross R, Matthews I, and Baker S. **2005**. Generic vs. person specific active appearance models. *Image and Vision Computing* 23:1080-1093.

Gu H, and Ji Q. **2004**. An automated face reader for fatigue detection. Automatic Face and Gesture Recognition, 2004 Proceedings Sixth IEEE International Conference on: IEEE. p 111-116. DOI: 10.1109/AFGR.2004.1301517.

Guger C, Edlinger G, Leeb R, and Pfurtscheller G. **2004**. Heart-rate variability and event-related ecg in virtual environments. *2004: The 7th Annual International Workshop on Presence*.

Hazlett R. **2008**. Using biometric measurement to help develop emotionally compelling games. *Game usability: Advancing the player experience*:187-205.

Herbelin B, Benzaki P, Riquier F, Renault O, and Thalmann D. **2004**. Using physiological measures for emotional assessment: a computer-aided tool for cognitive and behavioural therapy. The Fifth International Conference on Disability, Virtual Reality and Associated Technologies 2004. p 307-314.

Isbister K, and Schaffer N. **2008**. *Game usability: Advancing the player experience*: CRC Press.

Kanade T, Cohn JF, and Tian Y. **2000**. Comprehensive database for facial expression analysis. Automatic Face and Gesture Recognition, 2000 Proceedings Fourth IEEE International Conference on: IEEE. p 46-53. DOI: 10.1109/AFGR.2000.840611.

Kang SJ, Kim YB, and Kim SK. **2014**. Analyzing repetitive action in game based on sequence pattern matching. *Journal of Real-Time Image Processing* 9:523-530.

Karthigayan M, Rizon M, Nagarajan R, and Yaacob S. **2008**. *Genetic algorithm and neural network for face emotion recognition*.

Kaulard K, Cunningham DW, Bülthoff HH, and Wallraven C. **2012**. The MPI Facial Expression Database—A validated database of emotional and conversational facial expressions. *PloS one* 7:e32321.

Kim YB, Kang SJ, and Kim CH. **2013**. System for evaluating usability and user experience by analyzing repeated patterns. *Design, User Experience, and Usability Design Philosophy, Methods, and Tools*: Springer, 322-329.

Kivikangas JM, Chanel G, Cowley B, Ekman I, Salminen M, Järvelä S, and Ravaja N. **2011**. A review of the use of psychophysiological methods in game research. *Journal of Gaming & Virtual Worlds* 3:181-199.

Lang PJ. **1995**. The emotion probe: studies of motivation and attention. *American psychologist*

50:372.

Li H. **2001**. Computer recognition of human emotions. Intelligent Multimedia, Video and Speech Processing, 2001 Proceedings of 2001 International Symposium on: IEEE. p 490-493.

Li H, Yu J, Ye Y, and Bregler C. **2013**. Realtime facial animation with on-the-fly correctives. *ACM Transactions on Graphics* 32:42.

Martins P, Caseiro R, and Batista J. **2013**. Generative face alignment through 2.5 d active appearance models. *Computer Vision and Image Understanding* 117:250-268.

Medler B, John M, and Lane J. **2011**. Data cracker: developing a visual game analytic tool for analyzing online gameplay. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: ACM. p 2365-2374. DOI: 10.1145/1978942.1979288.

Mirza-Babaei P, Long S, Foley E, and McAllister G. **2011**. Understanding the contribution of biometrics to games user research. DiGRA '11 - Proceedings of the 2011 DiGRA International Conference: Think Design Play: Citeseer.

Mirza-Babaei P, Nacke LE, Gregory J, Collins N, and Fitzpatrick G. **2013**. How does it play better?: exploring user testing and biometric storyboards in games user research. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: ACM. p 1499-1508. DOI: 10.1145/2470654.2466200.

Nacke LE, and Lindley CA. **2010**. Affective ludology, flow and immersion in a first-person shooter: Measurement of player experience. *arXiv preprint arXiv:10040248*.

Nakasone A, Prendinger H, and Ishizuka M. **2005**. Emotion recognition from electromyography and skin conductance. Proc of the 5th International Workshop on Biosignal Interpretation: Citeseer. p 219-222.

Nogueira PA, Torres V, and Rodrigues R. **2013**. Automatic Emotional Reactions Identification: A Software Tool for Offline User Experience Research. *Entertainment Computing–ICEC 2013*: Springer, 164-167.

Nummenmaa L, Glerean E, Hari R, and Hietanen JK. **2014**. Bodily maps of emotions. *Proceedings of the National Academy of Sciences* 111:646-651.

Pagulayan RJ, Keeker K, Wixon D, Romero RL, and Fuller T. **2002**. User-centered design in games. *The human-computer interaction handbook*, 883-906.

Pantic M, and Bartlett MS. **2007**. *Machine analysis of facial expressions*: I-Tech Education and Publishing.

Pejsa T, and Pandzic I. **2009**. Architecture of an animation system for human characters. Telecommunications, 2009 ConTEL 2009 10th International Conference on: IEEE. p 171-176.

Picard RW, and Picard R. **1997**. *Affective computing*: MIT press Cambridge.

Ravaja N, and Kivikangas JM. **2008**. Psychophysiology of digital game playing: The relationship of self-reported emotions with phasic physiological responses. Proceedings of Measuring Behavior. p 26-29.

Ravaja N, Laarni J, Saari T, Kallinen K, Salminen M, Holopainen J, and Järvinen A. **2004**. Spatial presence and emotional responses to success in a video game: A psychophysiological study. *Proceedings of the PRESENCE*:112-116.

Ravaja N, Saari T, Salminen M, Laarni J, and Kallinen K. **2006**. Phasic emotional reactions to video game events: A psychophysiological investigation. *Media Psychology* 8:343-367.

Sánchez JLG, Vela FLG, Simarro FM, and Padilla-Zea N. **2012**. Playability: analysing user experience in video games. *Behaviour & Information Technology* 31:1033-1054.

Tammen H, and Loviscach J. **2010**. Emotion in Video Games: Quantitative Studies? *Emotion in*

HCI–Designing for People:25.

Thai LH, Nguyen NDT, and Hai TS. **2011**. A facial expression classification system integrating canny, principal component analysis and artificial neural network. *arXiv preprint arXiv:11114052*.

Thawonmas R, and Iizuka K. **2008**. Visualization of online-game players based on their action behaviors. *International Journal of Computer Games Technology* 2008:5.

Tzimiropoulos G, and Pantic M. **2013**. Optimization problems for fast aam fitting in-the-wild. Computer Vision (ICCV), 2013 IEEE International Conference on: IEEE. p 593-600. DOI: 10.1109/ICCV.2013.79.

Vermeeren AP, Law EL-C, Roto V, Obrist M, Hoonhout J, and Väänänen-Vainio-Mattila K. **2010**. User experience evaluation methods: current state and development needs. Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries: ACM. p 521-530.

Wallner G, and Kriglstein S. **2012**. A spatiotemporal visualization approach for the analysis of gameplay data. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: ACM. p 1115-1124. DOI: 10.1145/2207676.2208558.

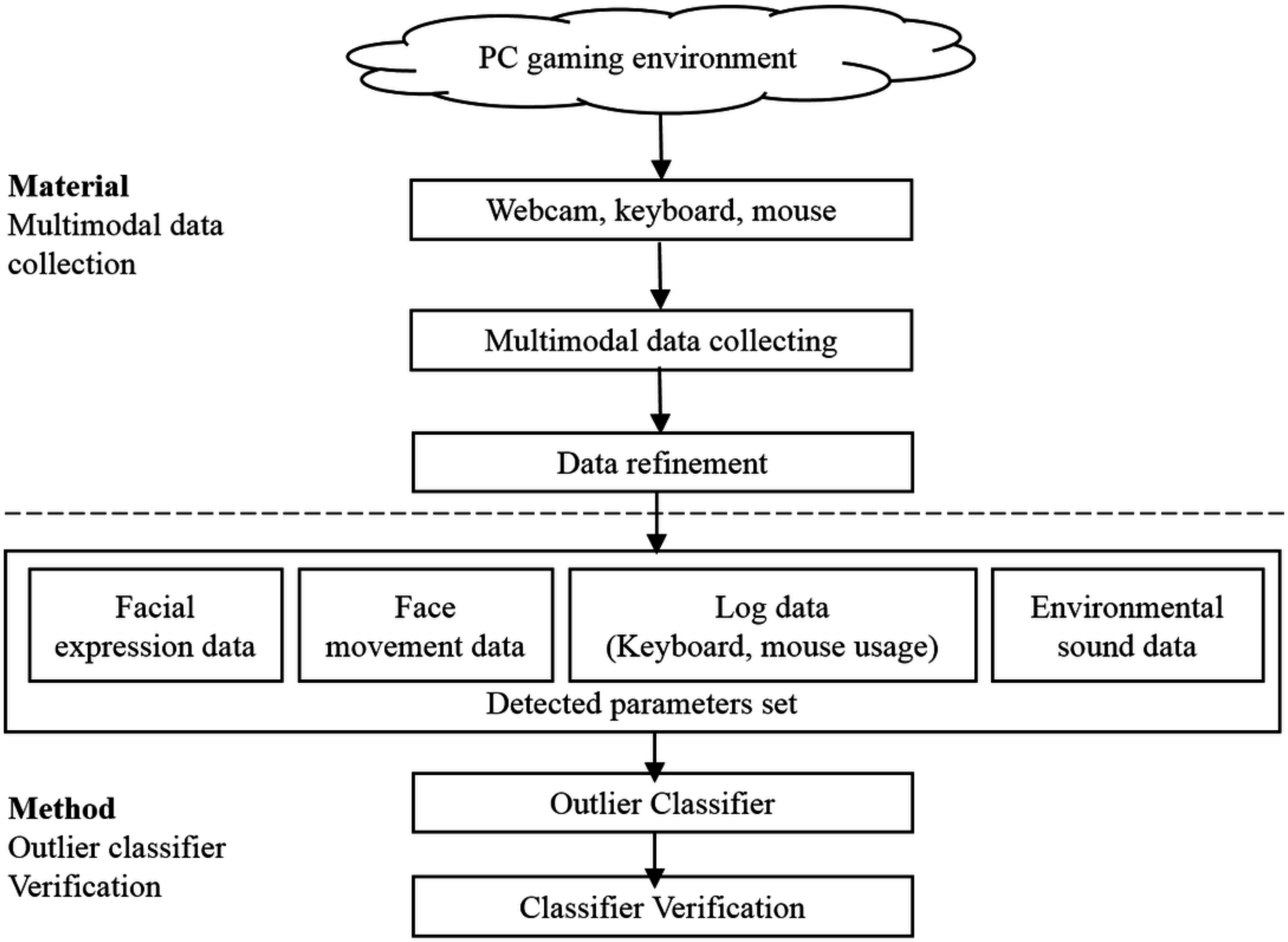
Wang N, Gao X, Tao D, and Li X. **2014**. Facial Feature Point Detection: A Comprehensive Survey. *arXiv preprint arXiv:14101037*.

Zoeller G. **2010**. Game development telemetry. Proceedings of the Game Developers Conference.

1

System overview.

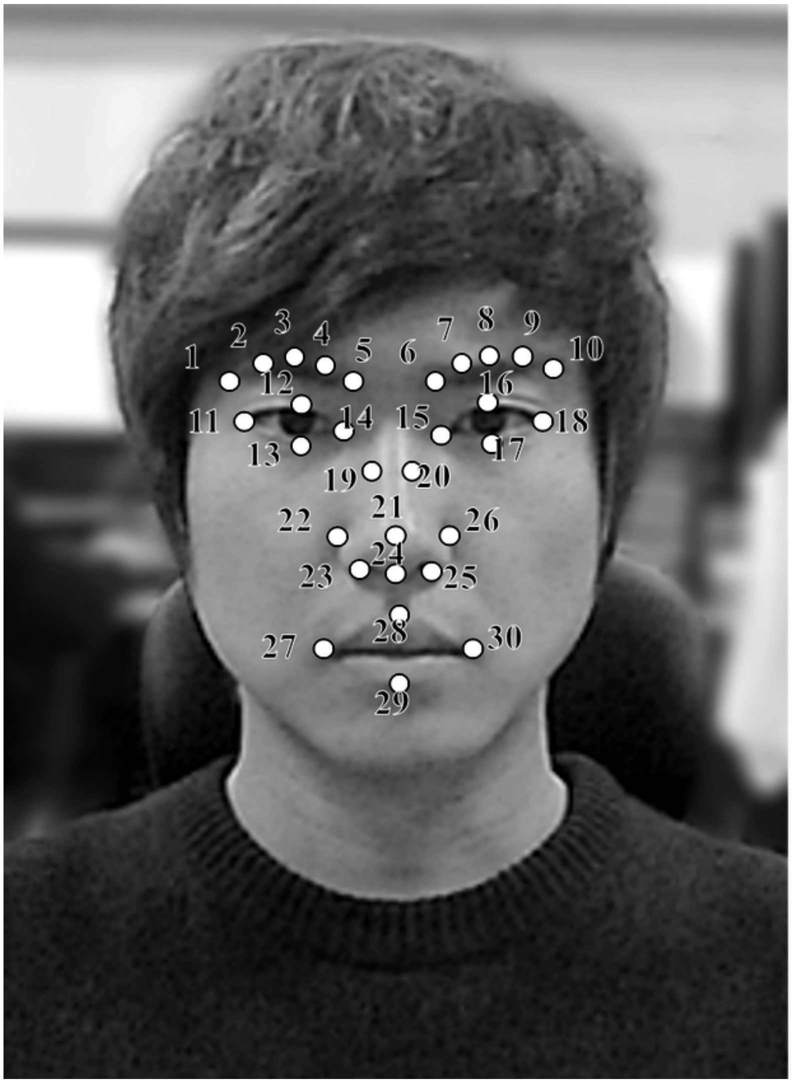
The outlier classifier is generated based on the multimodal data of users collected in a PC game environment.



2

30 facial feature points and all 24 feature points constructed from these 30 feature points

30 facial feature points used for facial-expression and blinking recognition (left), and all 24 feature points constructed from these 30 feature points for measuring angle variations (right).

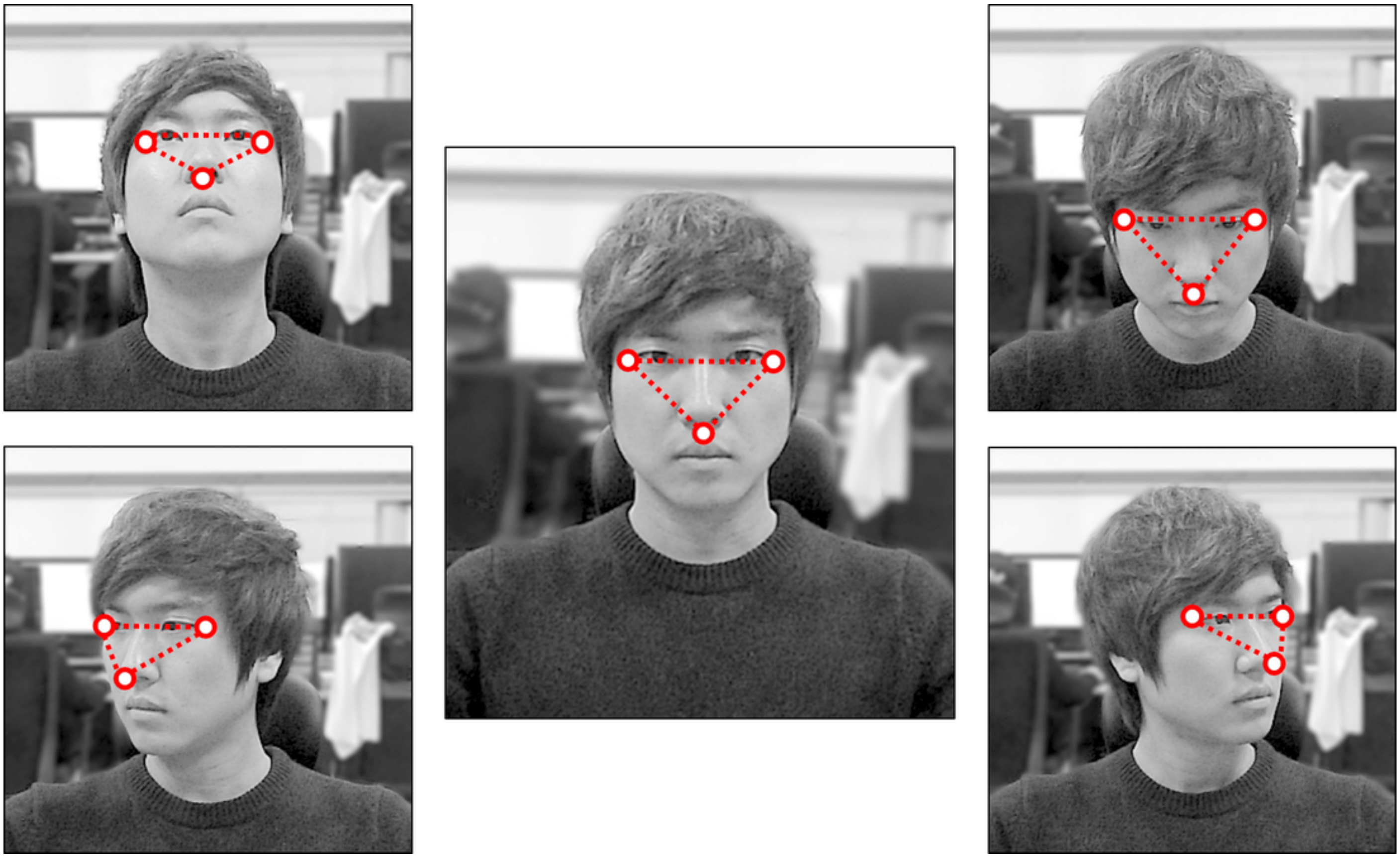


1 – 3 – 5:	Right Eyebrow (entire edge)
3 – 4 – 5:	Right Eyebrow (start edge)
2 – 3 – 4:	Right Eyebrow (middle edge)
1 – 2 – 3:	Right Eyebrow (end edge)
6 – 8 – 10:	Left Eyebrow (entire edge)
6 – 7 – 8:	Left Eyebrow (start edge)
7 – 8 – 9:	Left Eyebrow (middle edge)
8 – 9 – 10:	Left Eyebrow (end edge)
11 – 12 – 14:	Right Eye (up edge)
11 – 13 – 14:	Right Eye (down edge)
12 – 14 – 13:	Right Eye (start edge)
12 – 11 – 13:	Right Eye (end edge)
15 – 16 – 18:	Left Eye (up edge)
15 – 17 – 18:	Left Eye (down edge)
16 – 15 – 17:	Left Eye (start edge)
16 – 18 – 17:	Left Eye (end edge)
19 – 21 – 20:	Nose (top edge)
21 – 22 – 23:	Nose (right edge)
21 – 26 – 25:	Nose (left edge)
23 – 24 – 25:	Nose (under edge)
27 – 28 – 30:	Mouth (top edge)
28 – 27 – 29:	Mouth (right edge)
28 – 30 – 29:	Mouth (left edge)
27 – 29 – 30:	Mouth (under edge)

3

To identify the direction of a player’s rotation and to detect lost feature points

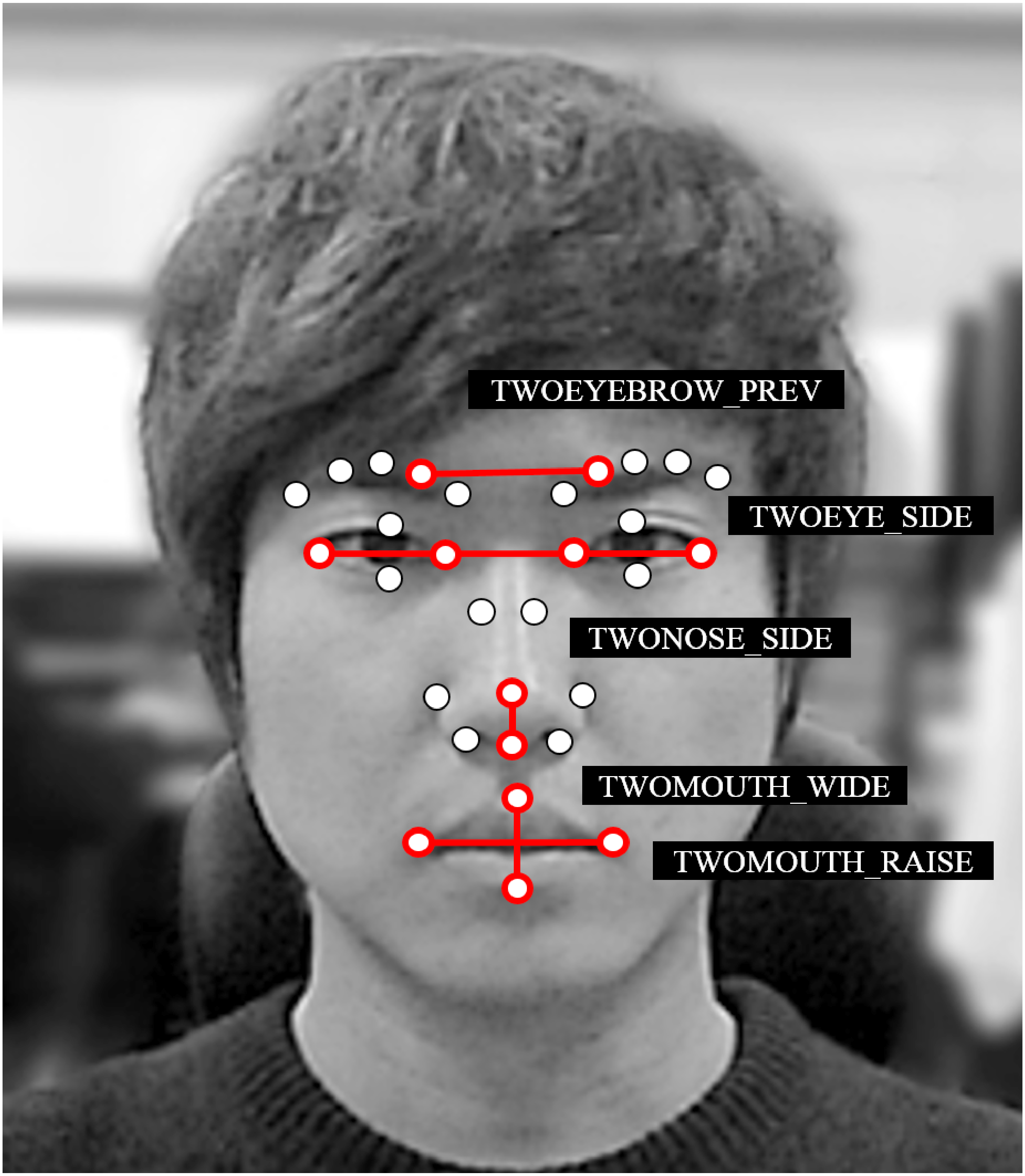
A triangle is created to connect feature points on both sides of the eye and the tip of the nose. The direction of rotation is defined according to the triangle’s internal angles, and weights are applied if feature points are lost in the applicable state.



4

All 30 feature points divided into 5 groups.

The sum of the variations in movement for the applicable group is used as an ANN learning input node for facial-expression recognition.


TWOEYEBROW_PREV
TWOEYE_SIDE
TWONOSE_SIDE
TWOMOUTH_WIDE
TWOMOUTH_RAISE

5

Experimental design overview.

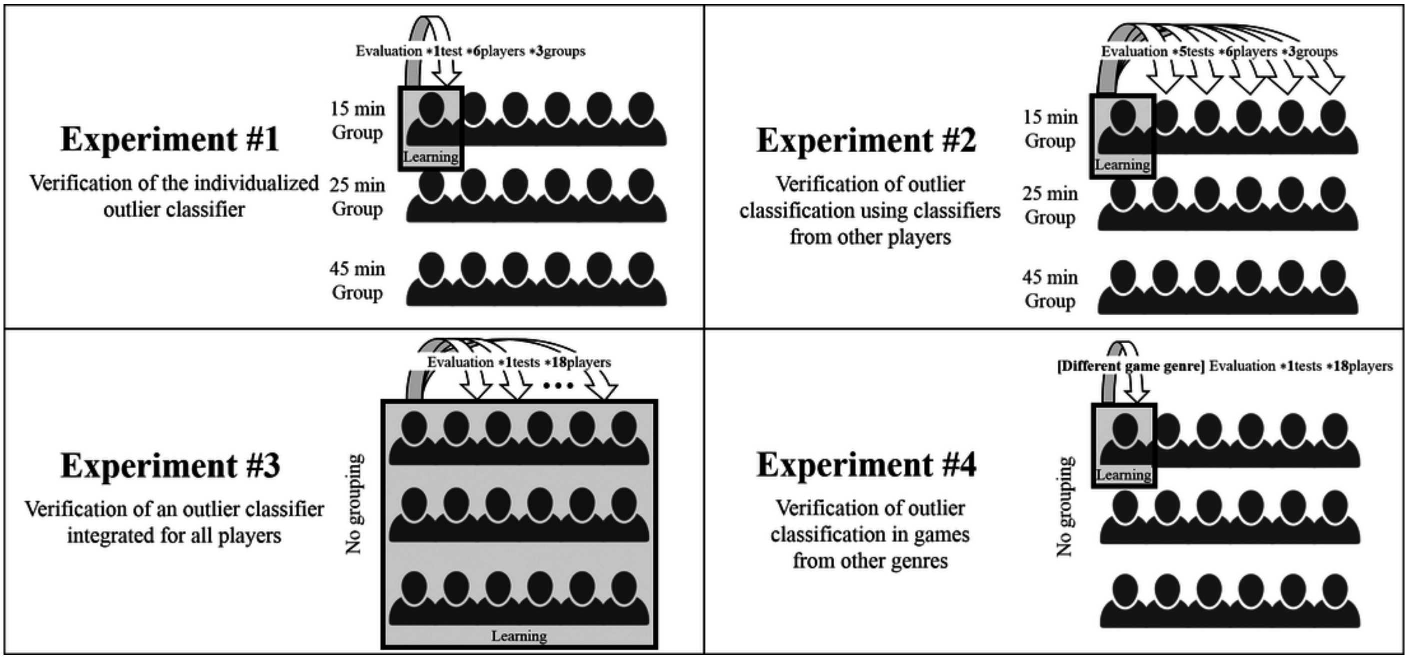

Experiment #1
Verification of the individualized outlier classifier
Evaluation *1test *6players *3groups
15 min Group
Learning
25 min Group
45 min Group
Experiment #2
Verification of outlier classification using classifiers from other players
Evaluation *5tests *6players *3groups
15 min Group
Learning
25 min Group
45 min Group
Experiment #3
Verification of an outlier classifier integrated for all players
Evaluation *1tests *18players
No grouping
Learning
Experiment #4
Verification of outlier classification in games from other genres
[Different game genre] Evaluation *1tests *18players
No grouping
Learning

Table 1(on next page)

Results for Experiments 1–4

Table 1. 4 Summarization data table for Experiment #1 through #4 Results

Experiment	**Manipulation**			**Result : recall value**			
	Game genre	**Learning**	**Evaluation**	**Grouping**			**No grouping**
				15min	**25min**	**45min**	
#1	Same	Individual	Individual	73.31%	75.29%	77%	—
#2	Same	Individual	Other players	64.17%	63.94%	68.26%	—
#3	Same	Integration	Each Players	—	—	—	66.54%
#4	Different	Individual	Individual	—	—	—	69.70%