A Framework for Processing Brain Waves Used in a Brain-computer Interface

Yunsick Sung*, Kyungeun Cho** and Kyhyun Um**

Abstract—Recently, methodologies for developing brain-computer interface (BCI) games using the BCI have been actively researched. The existing general framework for processing brain waves does not provide the functions required to develop BCI games. Thus, developing BCI games is difficult and requires a large amount of time. Effective BCI game development requires a BCI game framework. Therefore the BCI game framework should provide the functions to generate discrete values, events, and converted waves considering the difference between the brain waves of users and the BCIs of those. In this paper, BCI game frameworks for processing brain waves for BCI games are proposed. A variety of processes for converting brain waves to apply the measured brain waves to the games are also proposed. In an experiment the frameworks proposed were applied to a BCI game for visual perception training. Furthermore, it was verified that the time required for BCI game development was reduced when the framework proposed in the experiment was applied

Keywords—Brain-Computer Interface, BCI Toolkit, BCI Framework, EEG, Brain Wave

1. Introduction

The brain-computer interface (BCI) is a device that measures the brain waves from the human scalp [1]. The interface has been used to control wheelchairs for the physically challenged [2] and to control robot arms [3]. The recent launch of BCI devices [4, 5] at moderate prices has stimulated studies on approaches to developing games that use the brain waves measured by the BCI [6, 7].

The existing general frameworks for processing brain waves [8-11] do not include approaches to developing BCI games by considering the features of BCI games. As a result, the development method of BCI games using general frameworks for processing brain waves has inevitably reduced a significant amount of trials and errors. The framework for BCI games requires the following functions: firstly, brain waves must be converted into discrete values that can be applied to BCI games. In BCI games, brain waves are processed using pattern recognition or operant conditioning [12]. The former is conducted to estimate the nature of the mental task using the brain waves generated from the cerebral cortex of different users, while the latter is based on the changes in the brain waves in response to a specific event or stimulus. Such changes in brain

※ This paper is presented by summarizing part of the research, "A Development of Serious Game Technology using Brain Computer Interface" that was supported by Korea Creative Content Agency (KOCCA) in the Culture Technology(CT) Research & Developement Program 2009

Manuscript received October 6, 2011; accepted February 10, 2012.

Corresponding Author: Kyungeun Cho

* Dept. of Game Engineering, Graduate School, Dongguk University, Seoul, Korea (sung@dongguk.edu)

** Dept. of Multimedia Engineering, Dongguk University, Seoul, Korea (cke@dongguk.edu, khum@dongguk.edu)

waves are used to generate events or control virtual characters, and this necessitates the conversion of intrinsic values into values that can be easily controlled, even though the values themselves are not important.

Secondly, functions to generate the same events considering the differences among users must be provided. This is because it is difficult to design the level of BCI games since each user has a different brain wave distribution, and the reaction to an event or a stimulus varies from user to user.

Thirdly, functions to generate the signals appropriate for game control by mixing multiple brain waves is required. Since general users have no knowledge of brain waves, they have difficulty in applying the approaches to control BCI games by using the brain waves that they directly measure, like an alpha wave or a beta wave. Therefore, approaches to generate the signals that users can easily recognize by mixing multiple brain waves like those based on concentration values [4, 13] or meditation values [4] need to be developed. In the case of concentration values, users can easily understand that the values increase when they concentrate and decrease when they do not.

Finally, the game mode for the difficulty of BCI games should be controlled easily by controlling the brain waves without programming the functions for the difficulty of the game mode. Since BCI games continue through events generated by the measured brain waves, it is possible to control the difficulty of the game mode with the brain waves. Therefore the time to develop BCI games can be reduced.

This paper proposes a BCI game framework that is suitable for BCI game developments. Furthermore, it introduces the functions required to apply the measured brain waves to the games and describes the framework structure including the functions. An experiment conducted in the study reveals how the brain waves are processed and how applying the proposed approach to the BCI game for visual perception training develops the BCI games. The proposed framework will reduce the time and cost required for BCI game development.

This paper is organized as follows: Section II introduces BCI frameworks. Section III presents the framework required to develop BCI games by processing brain waves. Section IV describes the cases in which the proposed approach was applied to BCI games. Finally, Section V summarizes the proposed framework and the experiment

2. RELATED WORK

This section introduces the existing general frameworks that have been suggested to process brain waves [9-11, 14]. The problems that have arisen when developing BCI games using each framework are then analyzed.

The BCI++ framework, which is employed for laboratory purposes and practical applications, is comprised of a Hardware Interface Module (HIM) and AEnima [14]. The HIM collects and saves the measured brain waves. The AEnima includes a graphics engine and an audio engine, and it provides the user interface. The AEnima is generally installed in a PC used by a subject and the HIM is installed in the operator's PC to enable the operator to analyze the brain wave changes according to the operation of the AEnima. While BCI++ is open source and is based on C++, it comprises two modules and provides its own interface. Thus, it is not appropriate for games that are based on it to work as one independent application.

Open-ViBE is open source, and it provides the functions required to process the measured brain waves using a BCI device [9]. For example, it provides approaches to measure electroencephalography(EEG) or magnetoencephalography(MEG). In addition, it provides preprocessing to eliminate noise and strengthen signals. After a signal is preprocessed, Open-ViBE provides all the functions required in the process, from the extraction and classification of a feature vector to the conversion of a feature vector into commands. Open-ViBE provides a number of functions to generally process brain waves. However, it has none of the functions that are required to develop BCI games, including mixing or standardizing brain waves. Accordingly, it is necessary to expand these frameworks. This expansion will be difficult for developers who have no experience with BCI devices or brain wave processing.

BCI2000 is a BCI system that has been proposed for general usage purposes [10]. BCI2000 is comprised of a Source Module, Signal Processing Module, User Application Module, and Operator Module. The Source Module provides functions to acquire and save data. The Signal Processing Module extracts and translates the features. The User Application Module is the application developed for users, and the Operator Module provides the system parameters and so enables each module to refer to them. BCI2000 has no preprocessing capability, unlike other frameworks. Thus, it does not provide functions to convert the measured brain waves for applications. For BCI game development, BCI2000 therefore requires additional functions to convert the measured brainwaves into signals that can control events.

The last framework is BioSig, which provides a framework to process brain waves in Matlab, which was developed by MathWorks [11]. BioSig is a software library for biomedical signal-processing tools. It provides functions to process diverse brain waves, which is similar to what Open-ViBE does. However it has no functions to convert brain waves, which are required to develop BCI games.

This paper introduces a variety of brain wave processing approaches, which are required to develop games that use brain waves measured by a brain wave-measuring device. Furthermore, this paper suggests a new BCI game framework that has been developed by applying the proposed brain wave-processing approaches.

3. A BCI GAME FRAMEWORK

This section describes how to process a series of brain waves measured using a BCI device, in consideration of the features of BCI games.

3.1 An Overview of a Brain Wave Control

The framework structure suggested in this paper adopts a layered architecture for expandability. Each layer independently processes and provides brain waves. Therefore, this framework could facilitate the addition of features, depending on the BCI game. For devices from third parties the framework includes the physical layer and the data layer, which can measure brain waves and the relevant drivers. Furthermore, the acquisition layer and the preprocessing layer are suggested for processing the brain waves. Finally, there is the application layer, which includes BCI games.

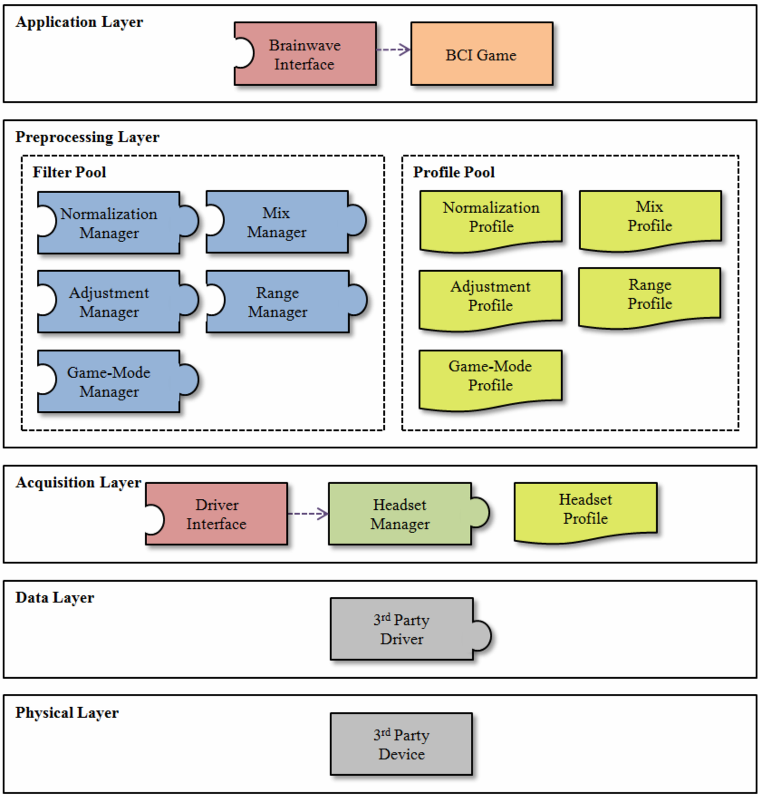


Fig. 1. Layers in the Framework

3.2 Physical Layer and Data Layer

The physical layer is comprised of a brain wave-measuring device developed by a third party and it does not include software. Each brain wave-measuring device provides an independent driver to communicate the measured brain wave values from the device to the application. The suggested framework configures the data layer using the device driver. The device driver provides the software layer with an interface to enable the access to and the use of the brain waves measured in the physical layer.

3.3 Acquisition Layer

The third-party device driver has different functions and application methods. Diverse interfaces make it difficult to develop BCI games that support a variety of brainwave measuring devices. Accordingly, it is necessary that an approach be developed to control various devices using the same interface.

Each device driver is connected to the headset manager through the driver interface. The driver interface maps the functions of a third party driver onto the functions of the headset manager. The BCI games are developed using the brain waves from the headset manager as input. This enables independent development without dependence upon any particular measuring device or

Table 1. Acquisition Layer Profile

Number	Manager	Parameter	
1	Headset Profile	$\varepsilon_{H\text{-}DeviceType}$ $\varepsilon_{H\text{-}DriverType}$ $\varepsilon_{H\text{-}Type}$	Brainwave measuring device model to use Brainwave measuring device driver to use Brainwave types to use (e.g.) Alpha wave, beta wave

device driver. The headset manager selects the device and the device driver to use from multiple brain wave measuring devices that are connected, in accordance with the settings of the headset profile, as shown in Table 1. Next, the headset manager selects the brain waves to apply to the BCI games from among the measured brain waves.

3.4 Preprocessing Layer

The preprocessing layer converts the brain waves from the headset manager to signals that are in the appropriate range for use in BCI games. Many different parameters are required to convert the brain waves. Those parameters need to be separated from the manager in converting the brain waves, in order to allow for their easy modification. The proposed preprocessing layer defines function to convert the brain waves using multiple managers. It also classifies the parameters into several groups and defines profiles of the multiple managers. Furthermore, the filter pool administers multiple managers, while the profiles are managed in the profile pool.

Multiple profiles are defined to specify the parameters that each manager requires when converting the brain waves. A developer can convert the brain waves by manipulating the profiles, without doing any programming. Each profile defines the parameters as illustrated in Table 2.

The series of processes used to convert the brain waves varies between BCI games. Thus, it is difficult to convert them using a standard process. To solve this problem, the filter pool breaks down the brain wave conversion functions and defines them into each manager. The filter pool

Table 2. Parameters in Each Profile

Number	Manager	Parameter	
1	Normalization Profile	$\varepsilon_{N\text{-}Time}$ $\varepsilon_{N\text{-}Min}$ $\varepsilon_{N\text{-}Max}$	Time in seconds to collect brain waves for standardization Minimum value Maximum value
2	Adjustment Profile	$\varepsilon_{A\text{-}AdditionValue}$ $\varepsilon_{A\text{-}MultiplicationValue}$	Increase/decrease of brain waves Multiply brain waves
3	Game-Mode Profile	$\varepsilon_{G\text{-}EasyValue}$ $\varepsilon_{G\text{-}NormalValue}$ $\varepsilon_{G\text{-}HardValue}$	Increase/decrease of brain waves at the easy game mode Increase/decrease of brain waves at the normal game mode Increase/decrease of brain waves at the hard game mode
4	Mix Profile	$\varepsilon_{M\text{-}Src1Type}$ $\varepsilon_{M\text{-}Src2Type}$ $\varepsilon_{M\text{-}Src3Type}$ $\varepsilon_{M\text{-}Src4Type}$ $\varepsilon_{M\text{-}Equation}$	First brain wave to mix Second brain wave to mix Third brain wave to mix Fourth brain wave to mix Mixing formula
5	Range Profile	$\varepsilon_{R\text{-}Min}$ $\varepsilon_{R\text{-}Max}$ $\varepsilon_{R\text{-}Time}$	Minimum value Maximum value Time in seconds to collect brainwaves to calculate the expected maximum and minimum value of brainwave

Table 3. Filter Pool Manager Composition

Number	Manager	Parameter
1	Normalization Manager	Collect the brain waves for $\varepsilon_{\text{N-Time}}$ seconds to transform those that exhibit a diverse distribution that is dependent on users, and reduce the user-specific features reflected in brain waves, through normalization [15].
2	Adjustment Manager	$\varepsilon_{\text{A-Value}}$ is added to the measured brain waves.
3	Game-Mode Manager	Control the brain waves depending on the game mode level of difficulty. $\varepsilon_{\text{G-EasyValue}}$, $\varepsilon_{\text{G-NormalValue}}$ or $\varepsilon_{\text{G-HardValue}}$ is added to Easy, Normal, or Hard, respectively.
4	Mix Manager	Brain waves from $\varepsilon_{\text{M-Src1Type}}$ to $\varepsilon_{\text{M-Src4Type}}$ are mixed in accordance with the Mix-Equation formula and the new type of brain wave is generated.
5	Range Manager	Measured brainwaves are converted into the brainwaves from $\varepsilon_{\text{R-Min}}$ to $\varepsilon_{\text{R-Max}}$.

is comprised of such managers. The filter pool provides five kinds of managers, as shown in Table 3. Each manager provides intrinsic functions to convert brain waves. The filter pool can include multiple managers, depending on the conversion approach, and it does not need to use any specific managers. In addition, it selects the processing order of a manager, depending on the conversion process.

The Normalization Manager eliminates the features of a user from the measured brainwave. The brainwave process is divided into the process to analyze brain waves and the process to convert brain waves [15]. The analysis process measures the brain waves of a user as shown below before using brain waves and calculates the average and standard deviations. The brain waves are collected once a second for $\varepsilon_{\text{N-Time}}$ seconds. Next, the average b_μ and the standard deviation b_σ are calculated by collected brain waves to equations (1) and (2), respectively. b_t is the brain wave measured at the time t.

$$b_\mu = \frac{\sum_{t=1}^{\varepsilon_{\text{N-Time}}} b_t}{\varepsilon_{\text{N-Time}}} \tag{1}$$

$$b_\sigma = \sqrt{\frac{\sum_{t=1}^{\varepsilon_{\text{N-Time}}} (b_t - b_\mu)^2}{\varepsilon_{\text{N-Time}}}} = \sqrt{\frac{\sum_{t=1}^{\varepsilon_{\text{N-Time}}} {b_t}^2}{\varepsilon_{\text{N-Time}}} - {b_\mu}^2} \tag{2}$$

When a user for measuring brain waves is changed, the average and standard deviation of brain waves are changed. Thus, it is required to recalculate the average and the standard deviation of brain waves using the analysis process.

The conversion process converts the measured brain waves as shown in below. First, the Z-score z is calculated using the measured brain waves, average b_μ and standard deviation b_σ. The Z-score is calculated using equation (3):

$$Z = \frac{b_t - b_\mu}{b_\sigma} \tag{3}$$

Next, the probability $P(b_t)$ is calculated using the Z-score. The probability, $P(b_t)$, is the relative value of measured brain waves, which is between 0 and 1. When it is 1, it's the high value which can be measured. When it is 0, it is the smallest value. Third, the value b_t using the equation (4) to apply the measured brain waves is calculated depending on contents.

$$b_t = b_{N\text{-}Min} + (b_{N\text{-}Max} - b_{N\text{-}Min}) \times P(b_t) \tag{4}$$

$b_{N\text{-}Min}$ is the minimum value used in contents. $b_{N\text{-}Max}$ is the largest value used in contents. The brain waves are normalized in equation (4) and are converted into the waveforms between $b_{N\text{-}Min}$ and $b_{N\text{-}Max}$. Using the series of processes described above, the features of a user as reflected on brain waves are eliminated. Therefore, the conversion is for making the distribution of brain waves similar, even for different users.

The Adjustment Manager increases or decreases measured brain waves. In addition it increases or decreases at a certain ratio. To this end, the brain waves are converted using equation (5).

$$b_t = (b_t + \varepsilon_{A\text{-}AdditionValue}) \times \varepsilon_{A\text{-}MultiplicationValue} \tag{5}$$

The Game-Mode Manager provides the function to easily control the game level depending on the game mode of difficulty. The BCI game is controlled by brain waves. Then, adjusting the brain waves can control the game level. As shown in equation (6), the brain waves are increased or decreased depending on the game mode level of difficulty.

```
IF $\varepsilon_{Mode}$ = EASY THEN                                        (6)
        $b_t = b_t + \varepsilon_{G\text{-}EasyValue}$
IF $\varepsilon_{Mode}$ = NORMAL THEN
        $b_t = b_t + \varepsilon_{G\text{-}NormalValue}$
IF $\varepsilon_{Mode}$ = HARD THEN
        $b_t = b_t + \varepsilon_{G\text{-}HardValue}$
END IF
```

When ε_{Mode} is EASY, $\varepsilon_{G\text{-}EasyValue}$ is added to brain wave b_t. If ε_{Mode} is NORMAL, $\varepsilon_{G\text{-}NormalValue}$ is added to the brain wave $b_{t.}$ Finally if ε_{Mode} is HARD, $\varepsilon_{G\text{-}HardValue}$ is added to the brain wave b_t.

The Mix Manager provides the approach to receive and mix a variety type of brain waves. For example, the measured brain waves can be converted to a meditation value or concentration value, which a user intuitively and easily understands. The expression to mix the brain waves is defined in $\varepsilon_{M\text{-}Equation}$. Next, the brain waves that are to be mixed are described by $\varepsilon_{M\text{-}Src1Type}$ to $\varepsilon_{M\text{-}Src4Type}$.

The Range Manager converts the measured brain waves to waveforms in a certain range. Finally, its process is classified into the analysis process and conversion process as the same as the process of Normalization Manager. The analysis process measures the brain waves of a user for $\varepsilon_{R\text{-}Time}$ seconds to estimate the maximum and minimum values of the brain waves that need to be measured. The maximum value in measured brain waves is defined as $\varepsilon_{R\text{-}ExpectedMax}$. The minimum value is defined as $\varepsilon_{R\text{-}ExpectedMin}$. The conversion process converts the brain waves measured in real time and applies them to the BCI game. The conversion applies the estimated maximum and minimum values, as shown in equation (7).

$$b_t = \varepsilon_{R\text{-}Min} + (\varepsilon_{R\text{-}ExpectedMax} - \varepsilon_{R\text{-}ExpectedMin}) \times (b_t - \varepsilon_{R\text{-}Min}) \qquad (7)$$

Various managers can be configured into the Filter Pool and Profile Pool, as shown in Fig. 2, to apply brain waves to a game.

Dividing the gauges used in a BCI game into 10 steps controls the brain waves. The Mix Manager converts brain waves into a concentration value for games. The concentration value is calculated by mixing the SMR wave, β wave, and ϴ wave from among the measured brain waves as shown in equation (8) [12].

$$T = \frac{SMR + \beta}{\theta} \qquad (8)$$

The Headset Manager measures the SMR wave, β wave, and ϴ wave to apply equation (8). The Mix Manager sets up the SMR wave, β wave, and ϴ from $\varepsilon_{M\text{-}Src1Type}$ to $\varepsilon_{M\text{-}Src3Type}$ in that order. Then $\varepsilon_{M\text{-}Equation}$ is set up as shown in equation (9):

$$b_t = \frac{b_{M-Src1Type} + b_{M-Src2Type}}{b_{M-Src3Type}} \qquad (9)$$

For controlling a BCI game with brain waves, the Range Manager generates 10 step values with brain waves from 1 to 10 which can be applied to the BCI game. 1 and 10 shall be set up to $\varepsilon_{R\text{-}Min}$ and $\varepsilon_{R\text{-}Max}$, each. However, for increasing the occurrence of upper signal 10, the brain waves are converted into 13 steps from -3 to 10 and 3 are added by the Adjustment Manager. Then, the signals from -3 to 6 are added to by 3 and the signals from 7 to 10 are converted to 10

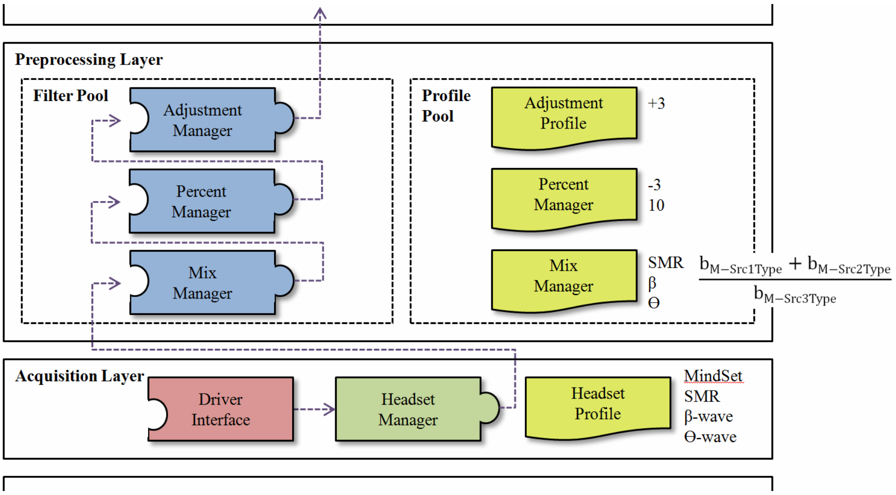


Fig. 2. Configuration Example of the Filter Pool and Profile Pool

with the limit of maximum value. Signal 10 shows the signal being 4 times as high as other values.

3.5 Application Layer

The brain waves converted by the preprocessing layer are finally applied in the BCI application. BCI games are developed using the brain wave interface. Since brain wave values are acquired through the brain wave interface, the BCI games can be developed to suit each brain wave measuring device.

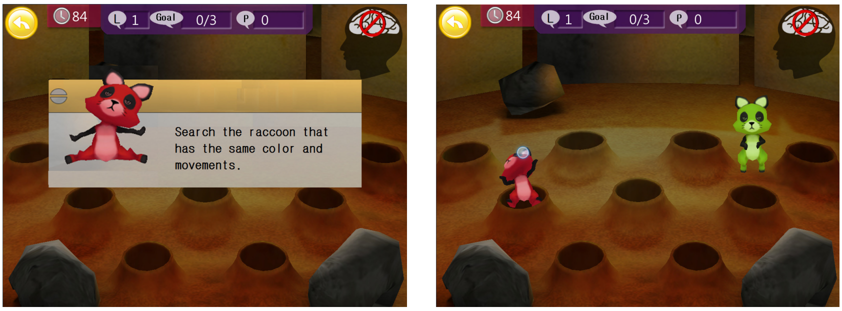
4. Experiment

The experiment applied the suggested approach to a BCI game that improves visual perception, as shown in Fig. 3. This was done to verify the suggested approach. When the game starts, a raccoon shape is depicted. The user is required to click on a raccoon that has the same color and movements as that which is initially displayed. The user watches the raccoons popping out of several holes. The user then finds and clicks on the matching raccoon. The repetition of this process strengthens the user's visual perception.

Mindset of NeuroSky [4] was adopted as the brain wave measuring device for this experiment. When the user's measured concentration value exceeds a certain value during the game, the raccoons' popping up speed slows down to make the game easy. If the user's concentration value is below a certain value, the raccoons pop out faster and the game becomes more difficult. The concentration value is determined by the brain wave obtained from Mindset. The more a user concentrates, the higher the concentration value is. This approach enhances the learning effect by inducing the users to concentrate on the games.

Fig. 4 shows the filter pool configuration and profile setting for the experiment. Only the concentration values from the brain waves measured by Mindset are delivered to the filter pool through the headset manager.

In the filter pool, the normalization manager was applied to reduce user-specific features reflected in the brain waves. The normalization manager requires the collection of a certain amount of brain waves for normalization. Therefore, the concentration values were collected for about 50 sec. before the first stage was begun. The concentration values after 50 sec. were normalized and transferred to the Range Manager. The game was set to be slow when brain wave



Search the raccoon that has the same color and movements.

Fig. 3. BCI Game for Visual Perception Training

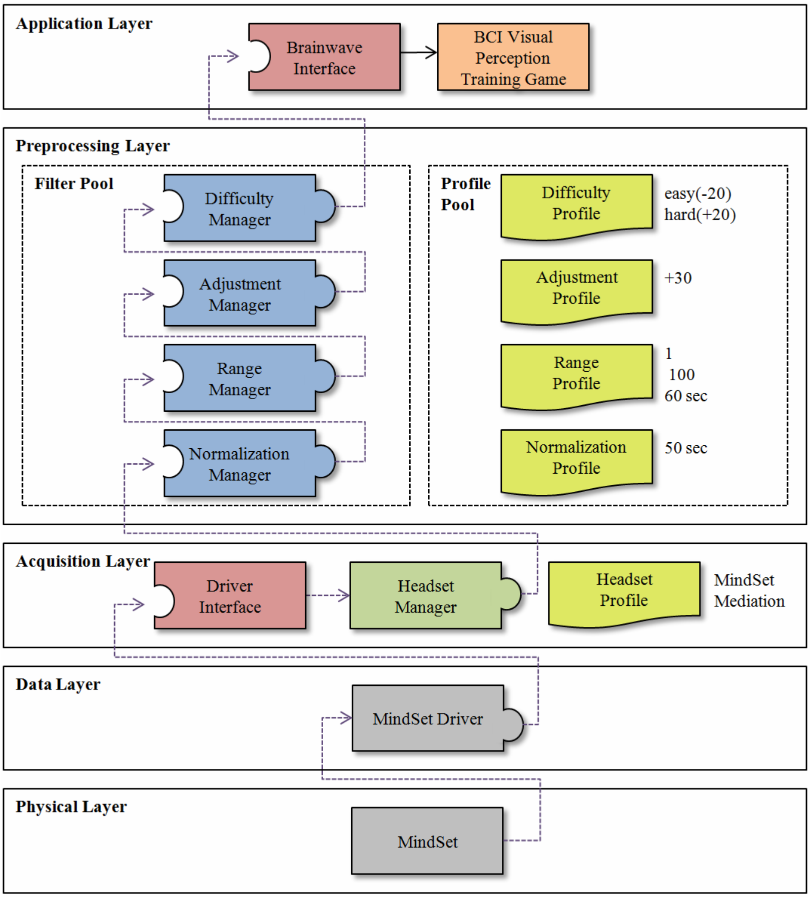
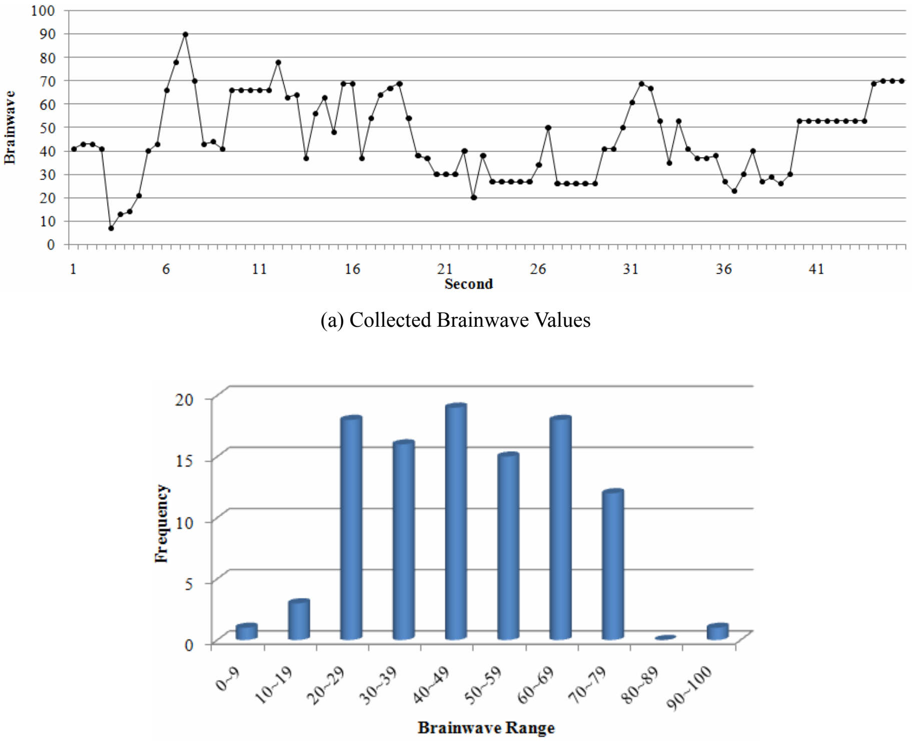


Fig. 4. Framework Applied to BCI Game for Visual Perception Training

values that were above 80% of the maximum amplitude were measured in the game. The concentration value was converted into a value between 1 and 100 for the observed brain waves. In designing the levels for the normal game mode, the Adjustment Manager was applied. A developer identified the game level by directly playing the game. Next, the parameter $\varepsilon_{\text{A-Value}}$ was determined for the Adjustment Manager. For the Game-Mode Manager, a developer determined $\varepsilon_{\text{G-EasyValue}}$ and $\varepsilon_{\text{G-HardValue}}$ to add or deduct when implementing the easy game mode and hard game mode, respectively. These values were again determined by directly playing the game. The converted values were applied to the BCI game for visual perception training. The framework suggested in this paper does not require any compilers for brain wave adjustment because it modifies the difficulties only by adjusting the profile.

Fig. 5 illustrates the distribution of brain waves that were collected for 50 seconds for normalization. Fig. 5(1) shows the diverse concentration values from 7 to 90. Fig. 5(b) presents the measured brain waves by section.

On the graph, about 95% of brain waves are distributed between 20 and 79. The concentration value at both ends is about 5%, which means that almost no concentration value was measured. Accordingly, it is required to make a normal distribution by normalization in order to make the



(a) Collected Brainwave Values

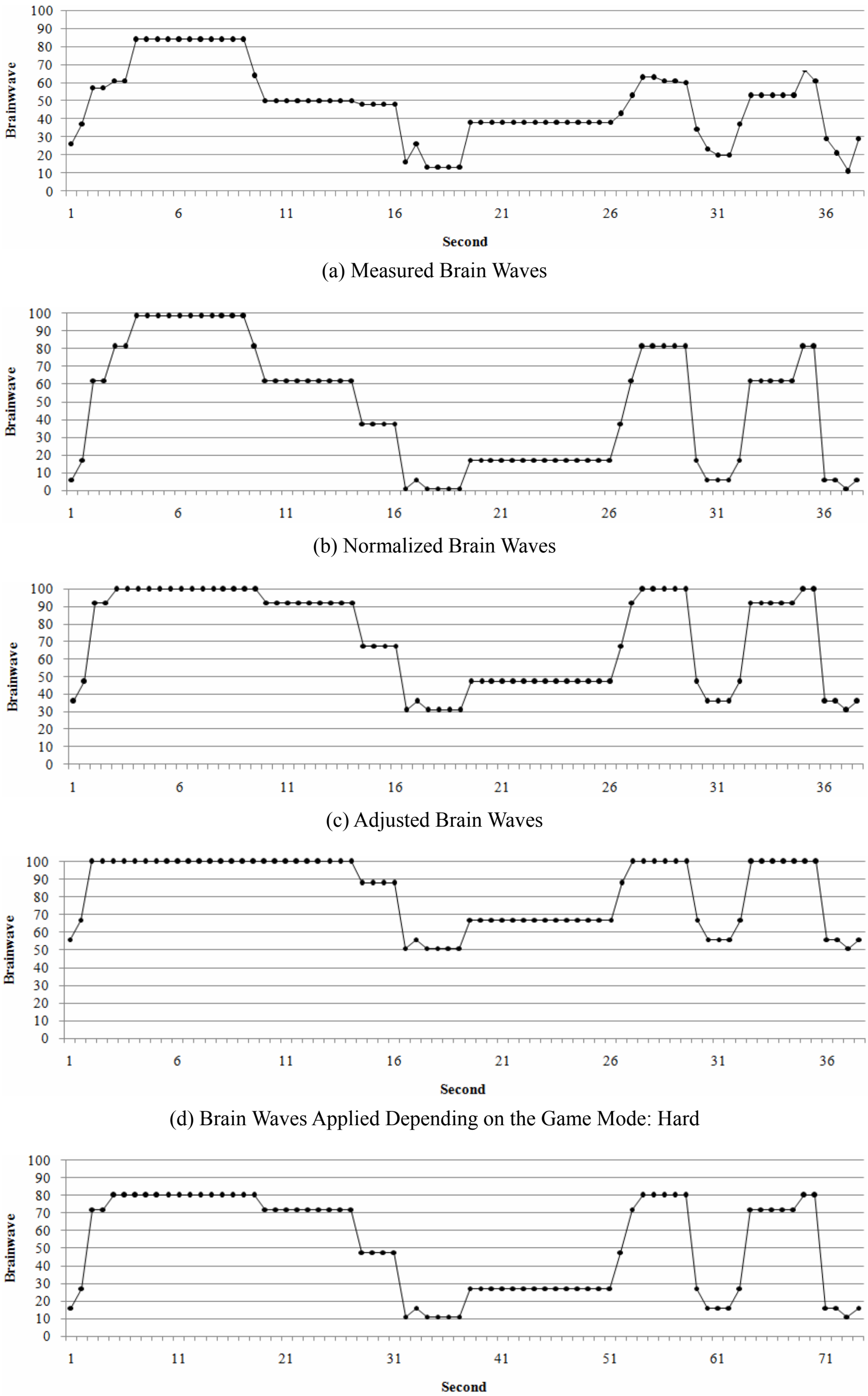
(b) Distribution of Collected Brain Waves

Fig. 5. Brain Waves Collected for Normalization by the Normalization Manager

collected brain waves have the standard distribution. Such processes are implemented in the Normalization Manager.

Fig. 6 illustrates the concentration values converted by the manager in the filter pool. Fig. 6(a) shows the concentration values collected during the BCI game for visual perception training. The concentration values obtained up until 16 sec. are the brain waves measured in the first stage. The brain waves between 19 sec. and 37 sec. are measured in the second stage. Fig. 6(b) depicts the brain waves acquired by normalizing the measured brain waves. The measured brain waves are distributed on both sides with the collected waves at the center. The brain waves were increased by a value of 30 in the adjustment manager after normalization. If the value exceeds 100, a value of 100 is indicated as shown in Fig. 6(c). Finally, the values are converted and applied to the game, as shown in Figs. 6(d) and (e), depending on the game mode.

The experiment described in this paper entailed the configuration of the filter pool and the profile pool applied to the BCI game for visual perception training. The paper also presented a series of processes to develop BCI games by setting a variety of parameters that are required to convert the brain waves. The converted brain waves are sensitive to the fun of the game. Thus, the developer needed to change the parameters required for the conversion by directly playing the game. However, a BCI game can easily be developed by changing the parameters of the profiles without the use of any separate compiler.



(a) Measured Brain Waves

(b) Normalized Brain Waves

(c) Adjusted Brain Waves

(d) Brain Waves Applied Depending on the Game Mode: Hard

(e) Brain Waves Applied Depending on the Game Mode: Easy

Fig. 6. Control of Measured Brain Waves

Table 4. Change of the Number of Brain Waves Exceeding the Event Limit Depending on the Brain Wave Conversion using Managers

Step	Manager		Number of cases when brain waves exceed the event limits	Percentage of cases when brain waves exceed the event limit
1	Headset Manager		11	About 15%
2	Normalization Manager		21	About 28%
3	Range Manager		21	About 28%
4	Adjustment Manager		38	About 51%
5	Difficulty Manager	Easy	43	About 58%
6		Hard	21	About 28%

Without BCI game frameworks, the developers need to directly implement the brain wave conversion processes required to develop BCI games by investing a large quantity of time. Furthermore, developers would need to change a variety of parameter values through repetitive compiling. Accordingly, development may require the investment of much time and cost. This paper verified that the BCI game framework, suggested through the experiment, solved the problems that arise when developing BCI games.

In BCI visual perception training game, if the brain wave exceeds the event limit, the pop up speed of a raccoon becomes slower. Table 4 shows the change of numbers in which the brain waves exceed the event limit when using each manager. The brain wave firstly measured by the headset manager exceeded the limit 11 times (15%). So the number of slowdown was less than with other managers. However, the normalization using the Normalization Manager increased the high or low brain wave distribution. Thus, the measured number of brain waves exceeding event limits increased to about 28%. Mindset measures the concentration value using the value from 0 to 100. Accordingly, no change was observed in the Range Manager. However, the values were adjusted to the value between 0 and 100 when using other brain wave measuring devices. Since the brain waves exceeding the event limits are 28%, the slowdown moment comprises a very small percentage in the game. Thus, the brain waves exceeding the limit were increased to 50% by using the Adjustment Manager. The measured number of brain waves exceeding the limit was adjusted to 43 times and 21 times at the easy game mode and hard game mode, respectively, when using the Game-Mode Manager. Accordingly, the game became difficult by reducing the number of slowdowns at a high difficulty level.

5. CONCLUSION

This paper suggested a framework to convert measured brain waves into input signals for games. The framework was comprised of five layers and the functions of each layer are described. The physical layer and the data layer included the device and the device driver developed by a third party. The acquisition layer measured the brain waves using only the Headset Manager. Thus, a developer could successfully develop a BCI game without considering specific BCI measurement devices. The preprocessing layer was comprised of the filter pool and the profile pool for brain wave conversion. The brain waves could be converted in a variety of ways by defining functions to perform the conversion, using managers. The brain wave values were adjusted without any other compilers by defining the parameters required for conversion in the profile. Finally, the application layer included the developed game. The BCI game was developed using the Brain Wave Interface. Thus, the direct involvement of complicated brain wave processes was reduced.

The suggested approach provided the functions required to develop BCI games as follows: first, the Headset Manager enabled diverse brain wave measuring devices to measure the brain waves using the same interface. Second, the filter pool provided the structure to convert the brain waves in diverse ways. This comprises the managers to convert the brain waves in various ways. Those functions help to reduce the time and costs required to develop the BCI games and enable BCI game coders to easily develop the games without any basic knowledge about BCI devices.

The experiment applied the suggested approach to the BCI game for visual perception training. The filter pool was configured in the following order for brain wave conversion: the Normalization Manager, Range Manager, Adjustment Manager, and Game-Mode Manager. The experiment introduced a series of processes to convert the brain waves using the suggested approach.

The proposed framework reduces development time and the cost required to implement a BCI game. Therefore, it improves the efficiency of BCI game development. Developers can proceed by concentrating on the game itself without considering particular BCI devices or brain wave processing methods. The BCI game market is expected to grow with the provision of a framework for BCI games.

REFERENCES

[1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtsheller, T. M.Vaughan, "*Brain computer interfaces for communication and control*," *Clinical Neurophysiology*, Vol.113, 2002, pp.767-791.

[2] I. Iturrate, J. Antelis, A. Kübler, and J. Minguez, "*A Non-Invasive Brain-Actuated Wheelchair based on a P300 Neurophysiological Protocol and Automated Navigation*," *IEEE Transactions on Robotics*, Vol.25, 2009, pp.614-627.

[3] E. E. Fetz, "*Operant Conditioning of Cortical Unit Activity*," *Science*, Vol.163, 1969.

[4] Neurosky, http://www.neurosky.com

[5] Emotive, http://www.emotiv.com

[6] B. Reuderink, "*Games and Brain-Computer Interfaces: The State of the Art*," WP2 BrainGain Deliverable Human Media Interaction (HMI), University of Twente, September, 2008, pp.1-11.

[7] A. Nijholt, "BCI for Games: A '*State of the Art' Survey*," *Entertainment Computing - ICEC 2008*, *Lecture Notes in Computer Science*, Vol.5309, 2009, pp.225-228.

[8] A. Kachenoura, L. Albera, L. Senhadji, and P. Comon, "*ICA: a potential tool for BCI systems*," *Signal Processing Magazine*, IEEE, Vol.25, 2008, pp.57-68.

[9] A. Lécuyer, Y. Renard, *OpenViBE: Open-Source Software for Brain-Computer Interfaces*, ERCIM News, 2009.

[10] G. Schalk, D. J. Mcfarl, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw, "*BCI2000: A general-purpose brain-computer interface (BCI) system,*" *IEEE Transactions on Biomedical Engineering, 51*, 2004.

[11] A. Schlogl, C. Brunner, "*BioSig: A Free and Open Source Software Library for BCI Research*," *Computer*, Vol.41, 2008, pp.44-50.

[12] J. Lehtonen, *EEG-based brain computer interfaces*, Master's thesis, Helsinki University of Technology: Department of Electrical and Communication Engineering, 2002.

[13] C. Lee, "*Development of the Game for Increasing Intensive Power using EEG Signal*," *Journal of Game Society*, Vol.9, 2009.

[14] P. Perego, L. Maggi, S. Parini, and G. Andreoni, "*BCI++: A New Framework for Brain Computer Interface Application*," *18th International Conference on Software Engineering and Data Engineering* (SEDE-2009), 2009, pp.37-41.

[15] Y. Sung, K. Cho, and K. Um, "A *Normalization Method to Utilize Brain waves as Brain-computer Interface Game Control*," *Journal of Korea Game Society*, 2010, Vol.10, pp.115-124.

Yunsick Sung

He received his B.S. degree in Computer Science and Engineering from Pusan National University in Busan, Korea, in 2004; his M.S. in Computer Engineering from Dongguk University in Seoul, Korea, in 2006; and his Ph.D. in Game Engineering from Dongguk University in Seoul, Korea, in 2012. He was employed as a member of the researchers at Samsung Electronics between 2006 and 2009. He was a adjunct professor at Shinheung College in 2009 and at Dongguk University in 2010. His main research interests are many topics related to brain-computer interface, programming by demonstration, ubiquitous computing, and reinforcement learning.

Kyungeun Cho

She received her B. Eng. Degree in Computer Science in 1993, her M. Eng. and her Dr. Eng. Degrees in Computer Engineering in 1995 and 2001, respectively, all from Dongguk University in Seoul, Korea. Since September 2003, she has been a Professor in the Department of Multimedia Engineering at Dongguk University in Seoul, Korea. During 1997-1998 she was a research assistant at the Institute for Social Medicine at Regensburg University, Germany, and a visiting researcher at the FORWISS Institute at TU-Muenchen University, Germany. Her current research interests are focused on the areas of the intelligence of robot and virtual characters and real-time computer graphics technologies. She has led a number of projects on robotics and game engines and also has published many technical papers in these areas.

Kyhyun Um

He received his B.S. degree in Applied Mathematics from Seoul National University in 1975, his M.S. in Computer Science from the Korea Advanced Institute of Science and Technology in 1977, and his Ph.D. in Computer Engineering in 1994 from Seoul National University in Korea. In 1978, he joined the Department of Computer Engineering at Dongguk University in Seoul, Korea. Since Sept. 2007, he has been a full Professor in the Department of Multimedia Engineering at the same university. He was a dean at the College of Information and Computer Science from March 2001 to Feb. 2003, and at the College of Digital Image and Contents from Sep. 2009 to Feb. 2011. He was also a Chair of SIGDB in Korea Information Science Society (currently: KIISE) from Sep. 1998 to Aug. 2000, and the president of the Korea Multimedia Society in 2007. Prof. Um has published over 100 papers in domestic and international journals and conferences in the areas of multimedia DB, game systems, and game artificial intelligence.