

무선으로 연동된 동물의 뇌 신호를 온라인 게임에 적용하기 위한 사운드 윈도우 설계

이현주^{0,1}, 김용범², 한승훈², 허현¹, 랑이란¹, 김은주², 송창근², 신형철¹

¹한림대학교 의과대학

²한림대학교 컴퓨터공학과

{bci⁰,stylemove,cgsong,hcshin}@hallym.ac.kr

A Sound Window System for Online Game with Animal of Wireless Brain Computer Interface

Hyun Joo Lee^{0,1}, Young-Bum Kim², Seung Hoon Han², Hyun Heo¹, Yiran Lang¹

Eun Ju Kim², Chang Geun Song², Hyung-Cheul Shin¹

¹Dept. of Physiology, College of Medicine, Hallym Univ.

²Dept. of Computer Engineering, Hallym Univ.

요 약

본 논문은 BCI 시스템을 이용하여 동물의 뇌 신호를 분석하고 온라인 상에서 인간과 동물이 상호 게임이 가능하도록 사운드 윈도우 시스템을 설계한다. 게임에 참여하기 위하여 컴퓨터 모니터를 이용한 인간의 시각적인 참여와는 달리 동물은 시각적 방법이 아닌 다른 방법을 제안해야 한다. 동물이 게임과 상호작용을 하기 위해서 사운드 윈도우 시스템은 게임에 사용될 지형의 크기에 따른 위치 정보와 게임 상황에 따른 소리를 각각 사상하여 소리를 이용한 가상공간을 구축하고, 게임에 참여할 동물이 공간에 제약을 받지 않도록 동물의 목에 헤드셋을 착용하여 블루투스로 신호를 전달하여 소리를 출력하는 방법을 사용한다. 본 논문은 사운드 윈도우 시스템을 이용하여 인간 대 인간, 인간 대 컴퓨터 간의 기존의 게임 방식을 동물이 참여하는 새로운 콘텐츠를 제공한다.

1. 서론

최첨단으로 발전된 하드웨어를 바탕으로 고급 그래픽 기술과 음향 효과를 적용한 게임들이 급속도로 발전하고 있다. 이러한 기술의 발전은 게임 수의 증가와 모바일 게임과 온라인 게임 등 성격이 상이한 플랫폼에서도 다양해졌다. 이 중 온라인 게임은 최근 몇 년간 주요한 인터넷 콘텐츠 산업으로 부상하고 있다[1].

온라인 게임의 대전 방식은 기존의 아케이드, 개인 컴퓨터, 비디오 게임이 인간 대 컴퓨터의 대전 형태로 이루어지는 것과는 다르게, 인간 대 인간의 대전 형태로 구성 되어지는 차이점을 보인다 인간 대 컴퓨터 간의 대전 방식은 게임의 패턴이 한정되어지고 반응 패턴이 지루해지는 반면, 인간 대 인간의 관계로 이루어진 대전 방식은 미리 적용된 반응 패턴을 갖기 힘들고 예측이 불가능하기 때문에 다양성의 재미를 더해준다 따라서 인간 대 인간의 게임은 다양성의 차이에서 게임 소프트웨어의 수명에 영향을 주며 인간 대 컴퓨터의 대전 방식 보다 긴 수명을 준다[2]. 그러나 아케이드 게임과 달리 온라인 게임은 컴퓨터 처럼 항상 상대가 존재하는 것이 아니다. 좋은 스토리와 콘텐츠로 구성을 하여도 충분한 접속자가 뒷받침되지 않는다면 게임은 성공할 수 없다.

본 연구는 제한된 인간 대 인간만을 위한 게임이 아니라 게임의 사용자 범위를 확대하여 인간 대 동물

간에도 게임이 가능하도록 동물의 뇌 신호 움직임을 이용하여 게임에 적용하였다. 네트워크가 연결된 컴퓨터로 원격지에 연결된 동물과 게임을 즐길 수 있을 뿐만 아니라, 동물과 인간 사이에 게임을 적용한다는 것에 새로운 시도를 보인다.

2. 관련연구

뇌에 대한 연구는 의학, 공학 등 여러 분야에서 진행되고 있으며, 그 한 예로 동물이 자신의 뇌로부터 발생하는 신호를 이용하여 기계를 조작할 수 있도록 하는 BCI(Brain Computer Interface) 시스템이 있다[3]. 또한 TV나 전등을 켜고 끌 수 있는 기능을 컴퓨터에 등록하여 거동이 불편한 장애인들이 그들의 뇌 신호를 이용하여 컴퓨터의 마우스 커서를 움직이게 하고 TV나 전등을 작동 가능하게 한 연구 결과도 있다[4].

하지만 BCI 기술에 대한 연구가 게임에 직접적으로 개발된 사례는 많지 않다. 위에서 설명한 두 가지의 연구는 오프라인에서 진행되어 있지만 본 연구는 온라인으로 진행된 인간과 동물간의 게임 대전으로 동물의 뇌 신호를 이용하였다. 이와 같이는 동물이 어떠한 생각 및 행동을 할 때 발생하는 뇌 신경세포의 활동성을 이용하여 본 BCI연구에 이용하고자 한다 이는 최근연구결과로 보아 새로운 기능에 대한 연구기능을 뇌의 가소성의 기능을 이용할 수 있다는 생각에서 시작되었으며 이를 계

임분야에 이용하려고 한다

3. 본론

3.1 전체 시스템의 개요

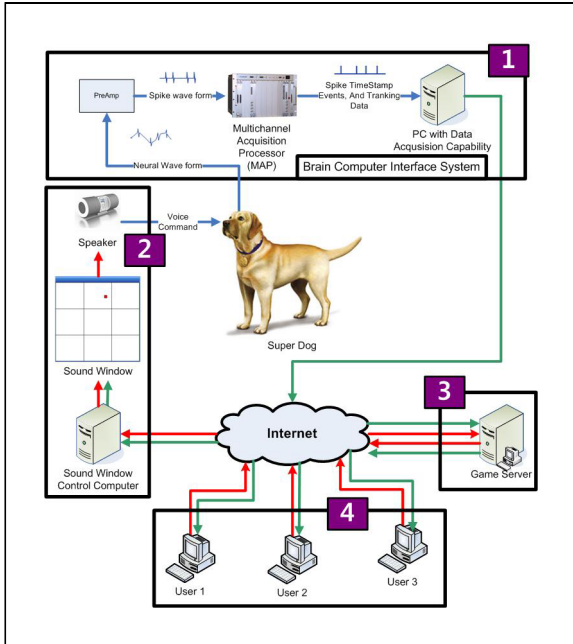


그림 1. Schematic diagram of the system

그림 1은 BCI(Brain Computer Interface) 시스템[3]을 동물에 연결하여 온라인 게임이 이루어지도록 4단계로 설계된 다이어그램이다. 본 연구는 그림 1의 다이어그램에서 ②영역에 해당하는 부분을 개발하였다.

게임에 참여한 동물에게 소리를 이용하여 온라인 게임에 참여하도록 하였고, 실시간으로 게임 진행 상황 및 동물의 뇌 입력 신호에 따른 Feedback을 주도록 하였다.

①에서 BCI 시스템은 실험 동물의 뇌 신호를 분석하고, 동물과 사상(Mapping)된 캐릭터의 행동을 결정할 수 있게 동물의 뇌 신호를 패킷으로 변경한 후 인터넷을 통하여 ③의 게임 서버에 보낸다. ③의 게임 서버에서는 BCI 시스템에서 보내온 패킷을 파싱(Parsing)하여 게임에 적용시키기 위해 ②번 사운드 제어 시스템과 ④번 사용자 시스템에서 사용할 수 있도록 패킷을 재구성하여 인터넷을 통해 두 영역으로 나누어 보내진다. ④번 영역은 게임 사용자용 컴퓨터들로 구성되어 인터넷을 통해 시간과 장소에 제한 없이 접속이 가능하다. 또한 ②번 영역에 도착한 신호는 SoundWindow를 통해 동물의 뇌 신호에 따른 결과 및 게임 상황을 동물에게 알려준다. 데이터 전달은 공간의 제약을 받지 않도록 블루투스를 이용하여 무선으로 연결된 헤드셋을 동물의 목에 착용하여 소리를 출력한다. 청각신호는 실험동물인 개의 발달된 능력을 사용함으로써 장소에 얽매이지 않는 시스템의 개발에 이용하였다.

3.2 제안한 사운드 윈도우 시스템

본 논문에서 제안하는 SoundWindow 시스템은 그림 2와 같이 총 4단계로 구성된다.

(1) 사운드 입력 단계

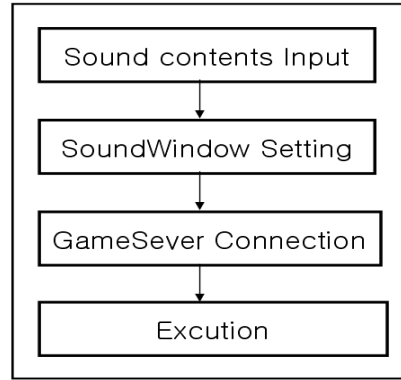


그림 2. Block diagram of presented system

사운드 입력을 위해서는 두 개의 텍스트 파일이 필요하다. 본 연구에서는 sound1.txt 파일과 sound2.txt 파일을 적용하였다. sound1.txt 파일은 동물이 제어하려는 게임상에서의 물체의 위치에 대한 소리 파일 목록이며, sound2.txt 파일은 동물에게 알려줄 게임 진행에 따른 소리(메시지) 파일 목록이다.

(2) 사운드 윈도우 설정

동물이 게임에 접속하여 게임을 진행을 하기 위해서는 가상공간이 필요하다. 인간은 컴퓨터의 모니터를 이용하여 시각적으로 게임을 이용할 수 있지만 동물은 불가능하기 때문에 본 연구는 소리를 이용한 가상공간을 설계하였다.

표 1. Sound window layout

1	2	3
4	5	6
7	8	9

사운드 입력 단계에서 입력한 sound1.txt 파일의 소리 목록을 이용하여 표 1에서 보인 각 셀의 인덱스마다 각각 다른 소리(메시지)로 사상(Mapping)하여 구성한다. 이 때 각 셀은 가상공간에서 발생하는 소리의 위치를 동물에게 알려준다.

빗줄 모양의 셀{1,3,7,9}과 수직 모양의 셀{8}에는 사운드 입력 단계에서 생성한 sound2.txt 파일의 소리 목록이 사상되어 게임의 진행에 따른 소리메시지를 출력

할 수 있게 구성되며 이로써 동물의 가상공간이 만들어진다.

게임의 종류와 범위에 따라 맵은 다양하게 나타나며, 동물마다 성향이 다르기 때문에 최적화된 공간을 구성하기 위하여 전체 테이블의 크기 설정 행과 열의 개수, 가상 사물의 시작 위치 사물의 이동 폭, 가상 사물의 크기 등을 조절할 수 있게 하였다 설정된 목록들은 파일에 저장할 수 있어 실행할 때마다 원하는 설정에 의해서 게임 환경을 사용할 수 있다

동물이 참여한 게임에 동물의 올바른 참여를 위해서는 동물의 뇌 신호를 모니터링 할 수 있는 훈련자가 필요하다. 동물의 뇌 신호 움직임을 표 1를 통해 확인할 수 있다.

```
int CClient::Send(const void* lpBuf, int nBufLen, int nFlags) {
    stTest sendPacket;
    sprintf(sendPacket, "%s\r\n", lpBuf);
    CRaviDuelSoundWindowDlg *pMainDlg =
(CRaviDuelSoundWindowDlg *)AfxGetMainWnd();
    pMainDlg->OnMakeWindow();
    return CSocket::Send(&joinPacket, nBufLen+3, nFlags);
}

void CClient::OnReceive(int nErrorCode) {
    CRaviDuelSoundWindowDlg *pMainDlg =
(CRaviDuelSoundWindowDlg *)AfxGetMainWnd();
    HWND hwnd = pMainDlg->GetSafeHwnd();
    stTest receivePacket;
    memset(receivePacket.content, '\0', sizeof(char)*512);
    int count = Receive(&receivePacket, sizeof(receivePacket));
    pMainDlg->m_strMessageWindow = receivePacket.content +
(pMainDlg->_strMessageWindow);
    pMainDlg->m_strSendMessage = "";
    pMainDlg->SetDlgItemText(IDC_EDIT1,
pMainDlg->m_strMessageWindow);
    if(strnicmp(receivePacket.content, "<START></START>", strlen("<START></START>"))==0) {
        pMainDlg->SetDefault();
        pMainDlg->OnButtonSoundOn();
        run_state = true;
    }
    if(strnicmp(receivePacket.content, "<END></END>", strlen("<END></END>"))
==0) {
        run_state = false;
        pMainDlg->OnButtonSoundOff();
    }
    if(run_state == true) {
        if(strnicmp(receivePacket.content, "<UP></UP>", strlen("<UP></UP>"))==
0) {
            SendMessage(hwnd, WM_UP, 10, 20);
        }else
        if(strnicmp(receivePacket.content, "<DOWN></DOWN>", strlen("<DOWN></DOWN>")
==0) {
            SendMessage(hwnd, WM_DOWN, 10, 20);
        }else
        if(strnicmp(receivePacket.content, "<LEFT></LEFT>", strlen("<LEFT></LEFT
>"))==0) {
            SendMessage(hwnd, WM_LEFT, 10, 20);
        }else
        if(strnicmp(receivePacket.content, "<RIGHT></RIGHT>", strlen("<RIGHT></RI
GHT>"))==0) {
            SendMessage(hwnd, WM_RIGHT, 10, 20);
        }else
        if(strnicmp(receivePacket.content, "<START></START>", strlen("<START></
START>"))==0) {
            pMainDlg->OnBnClickedButton2();
        }
    }
    CSocket::OnReceive(nErrorCode);
}
```

그림 3. Source code of soundwindow system

(3) 서버 연결

사운드 입력과 사운드 윈도우 설정 단계를 통해서 결정된 환경에 따라 동물이 게임을 할 수 있게 게임 서버를 연결하는 부분이다

게임 서버에 연결할 때에는 서버에 아이디를 보내어 참여 권한을 받음으로써 동기화가 진행되고 서버와 교환되는 데이터 패킷은 그림 3과 같이 XML 형식의 스트링으로 패킷을 구성하여 전송된다

그림 3에서 Send함수는 게임 서버에 메시지를 보내는 함수로 아이디를 부여 받은 후 그 아이디를 이용하여 서버에 신호를 보낸다 Receive함수는 게임 서버에서 받은 신호들{START, END, UP, DOWN, LEFT, RIGHT..}에 따라 그에 해당하는 이벤트를 발생하여 사운드 윈도우창에서 소리를 변경하게 한다 패킷을 파싱한 결과 <UP></UP> 스트링이면 WM_UP 이벤트가 발생하여 게임과 연결된 가상 오브젝트를 표 1의 공간에서 위쪽으로 이동하게 한다.

(4) 실행

동물이 인터넷으로 게임에 접속한 후 동물의 뇌 신호와 게임 서버의 데이터 패킷을 이용하여 소리로서 게임을 진행하는 단계이다

```
void StyleSound::InitStyleSound(HWND hwnd) {
    HRESULT hr; g_sm = new CSoundManager;
    hr = g_sm->Initialize(hwnd, DSSCL_PRIORITY);
    if(hr != DS_OK) {
        delete g_sm; g_sm = NULL;
        MessageBox(NULL, "Initialize", "error", MB_OK);
    }
}

void StyleSound::ReleaseStyleSound(void) {
    for(int i = 0; i < NUMBER_OF_SOUNDS; i++) {
        if(g_sound[i])
            delete g_sound[i]; g_sound[i] = NULL;
    }
    if(g_sm) {
        delete g_sm; g_sm = NULL;
    }
}

void StyleSound::LoadWavFile(int file_nbs, vector<CString> &sound_vec) {
    char name_buffer[1024]; HRESULT hr;
    for(int i = 0; i < file_nbs; i++) {
        string temp_sound = sound_vec[i];
        sprintf(name_buffer, temp_sound.c_str());
        hr = g_sm->Create(&g_sound[i],
name_buffer, DSBCAPS_GLOBALFOCUS);
        if(hr != DS_OK)
            MessageBox(NULL, name_buffer, "error", MB_OK);
    }
}

int StyleSound::KeyCodeToIndex(int keycode) {
    const char keyset[] = "0123456";
    return strchr(keyset, keycode) - &keyset[0];
}

void StyleSound::PlaySound(int keycode) {
    int n = KeyCodeToIndex(keycode);
    if(n < 0 || n >= NUMBER_OF_SOUNDS) return;
    g_sound[n]->Play(0, DSPLAY_LOOPING);
}
```

그림 4. Source of sound module

게임이 진행되는 동안은 소리의 출력이 중요한 요소가 된다. 동물은 소리에 민감하기 때문에 소리에 대해 정확성을 보장해야 한다. 따라서 소리의 정확성을 위해서 MicroSoft사에서 제공하는 DirectSound Interface[5]을 이용하여 모듈을 구성하였다.

그림 4는 소리의 왜곡을 최소화 하고 소리의 재생시 속도의 향상과 다른 어플리케이션 연동시 출력의 지속성을 보장하기 위해서 사용한 방법이다. 그림 4에서 InitStyleSound()는 Sound를 재생할 수 있는 객체를 만드는 부분으로 DSSCL_PRIORITY로 초기화 함으로써 재생 우선 순위를 높여준다. LoadWavFile()은 소리 파일을 로딩하고, 사운드 버퍼를 DSBCAPS_GLOBALFOCUS로 설정함으로써 윈도우창이 비활성화일 때도 소리 출력이 가능하게 한다.

4. 구현

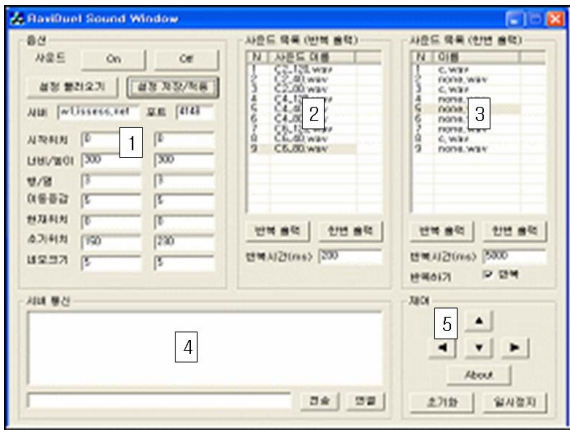


그림 5. User interface of system

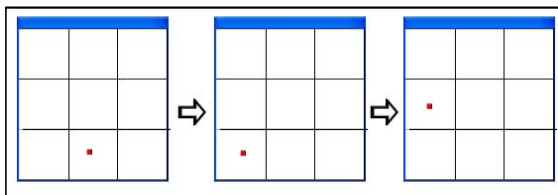


그림 6. Motion by sound

사운드 윈도우의 환경은 그림 5를 통해서 설정할 수 있다. 그림 5는 사운드 윈도우의 설정 및 제어 부분으로 크게 6가지로 구성된다.

- ① 사운드 윈도우 설정 부분
- ② 게임상의 가상 물체의 위치 소리 목록
- ③ 게임상의 이벤트에 대한 소리 목록
- ④ 게임 서버 통신 부분
- ⑤ 수동 제어 부분

그림 6은 동물의 뇌신호로 제어되는 가상 물체의 위치 (사각점)정보를 보여준다. 사용자가 접속하여 게임을 요청 하면 “게임에 접속합니다”라는 소리가 나오면서 표 1의 8번 셀에 등록된 위치를 알려주는 소리(메시지)

가 나온다. 동물은 훈련된 방법으로 표 1의 1번 셀로 이동하려고 하고, 사용자는 다른 위치로 움직이고자 한다. 동물과 사용자 각각 자신의 영역으로 이동하려고 경쟁함으로써 게임이 진행된다. 표 1의 각 모서리마다 현재 처해진 상황을 소리로 등록하여 동물의 현재 상황을 판단할 수 있게 도와준다.

5. 결론

본 연구는 동물과 게임을 하기 위해서 BCI 시스템을 이용하여 동물의 뇌 신호를 입력받고 사운드 윈도우를 이용하여 가상공간을 구축하였다. 또한 가상공간에 적용하도록 훈련을 시켜 인간과 동물이 한 네트워크 상에서 게임을 할 수 있도록 하였다.

동물에게 시각적, 물리적으로 가상환경을 구축하는 것은 공간의 제약이 따르고, 특정 공간에 동물을 가두어야 한다. 따라서 본 연구에서는 동물의 목에 헤드셋을 블루투스로 연결하여 소리(메시지)를 이용함으로써 게임에 참여가 가능하도록 가상공간을 적용하였다. 소리를 이용하여 동물은 가상공간에서 자유로이 활동을 하면서 사용자가 게임 요청시 이에 참여할 수 있게 한다. 하지만 규모가 큰 게임에 적용할 때에는 거대한 지형과 다양한 이벤트에 따른 사운드 데이터 자원이 요구된다. 많은 사운드 데이터를 대치할 수 있는 동물의 넓은 가청 주파수를 이용하여 주파수와 출력의 세기에 따른 사운드 자동 모듈을 추가해야 한다.

ACKNOWLEDGEMENT

This work was supported by the ERC program of MOST/KOSEF (grant R11-2000-075-01003-0(2007)) and by a grant(M103KV010019-06K2201-01910) from Brain Research Center of the 21st Century Frontier Research Program funded by the Ministry of Science and Technology, the Republic of Korea.

[참고문헌]

- [1] 명원식,한준탁,이양선, “인터넷 온라인 게임의 품질 평가에 관한 연구,”한국멀티미디어학회지 제 9권 제 2호 2005년 6월
- [2] 한국게임산업개발원 게임연구소 편 (2002 대한민국) 게임백서, 서울 : 한국게임산업개발원, 2002
- [3] Pfurtscheller G, Flotzinger D and Kalcher J., “Brain Computer Interface - a new communication device for handicapped persons,”J-Microcomputer Appl 1993, vol 16, pp. 203-299
- [4] Wolpaw JR, MCFarland DJ, Neat GW and Forneris CA, "An EEG-based brain-computer interface for cursor control," Electroencephalogr Clin Neurophysiol 1991, vol 78, pp. 252-259
- [5] Microsoft사, <http://msdn2.microsoft.com/>