

Design of Virtual Reality Contents for Upper-limbs Rehabilitation Using Kinect Sensor

Myeong-Chul Park *, Hyon-Chel Jung **, Hyun-Syug Kang ***

Abstract

The purpose of this study is to establish the contents of virtual reality for a patient who suffers from various diseases and needs Upper-limbs Rehabilitation. First, the system provides the movement content to remote patient. Then system is tracking information in the joints by using Kinect Sensor. And similarity comparison of a given content to diagnose the movement of the patient. The tracked movement information is stored in the database with similarity and is delivered to the rehabilitation therapist. The result of this study will enhance the effectiveness and concentration of the rehabilitation therapy and be used as basic data evaluating the function of the Upper-limbs Rehabilitation.

▶ Keyword : Kinect Sensor, Upper-Limbs Rehanilitation, Virtual Reality

1. Introduction

뇌혈관 질환의 대부분은 반신마비, 언어장애, 치매 등의 후유증을 유발하는 경향을 가진다. 일상생활의 대부분이 상지기능을 중심으로 이루어지는 것을 고려한다면 상지기능의 손상 및 불편으로 인한 영향은 환자에게 큰 절망감을 줄 수 있다[1]. 이와 관련된 여러 치료 중 가상현실 치료는 대상자에게 다차원적이고 다감각적인 가상 환경을 제공하여 컴퓨터와 대상자의 긴밀한 상호 작용 제공으로 증강된 몰입감을 갖도록 하는 장점이 있다[2]. 또한, 상지재활을 위한 치료사의 재활 치료 활동은 노동집약적 업무 요소를 많이 갖고 있기 때문에 환자가 자율적으로 사용할 수 있는 재활 콘텐츠가 필요하며 의료시설의 공간적 부족으로 환자가 원하는 시간에 지속적으로 치료를 받는 것 또한 현실적으로 많은 어려움이 있다. 본 논문에서는 범용적인 키넥트 센서를 이용하여 상지재활을 위한 원격진료 보조 시스템을 제안한다. 관절의 위치정보를 추적하기 위해서는 다양한 센서를 부착한 추적 시스템이 필요한데 이는 도구의 사용성을 저해하는 요인이 되며 손쉽게 사용할 수 있는 도구의 개발이

절실하다[3]. 본 논문에서는 속도계, 자이로스코프, 지자기 센서의 하드웨어 구성없이 상용적으로 사용되는 저가의 키넥트 센서를 이용하여 관절의 위치를 추적하는 시스템을 사용한다. 시각적인 부분은 간략한 인체 모델을 구축하여 2D 가상현실 콘텐츠를 제시하여 실제 궤적을 피드백 하는 가상현실 시스템 개발, 재활훈련의 평가 및 난이도 조절 가능하게 설계하고자 한다. 그리고 반복적인 운동 학습을 사용자가 주도적으로 행위를 할 수 있도록 다양한 재활 콘텐츠를 개발한다. 시스템 설계의 전체적인 구조는 [Fig. 1]과 같다. 환자의 상지운동을 통한 관절정보는 키넥트 센서에 의해 추적되고 추적된 상지활동은 재활 치료사가 제공하는 프로그램에 따라 진행되게 된다.

본 논문의 구성은 다음과 같다. 2절에서는 이론적 배경에 대해 설명하고 3절에서는 구체적인 설계와 구현에 대해 기술한다. 마지막 4절에서 결론 및 향후 연구에 대해 기술한다.

• First Author: Myeong-Chul Park, Corresponding Author: Hyun-Syug Kang
*Myeong-Chul Park (africa@songho.ac.kr), Software Academy, SongHo College
**Hyon-Chel Jung (jhc@medicalsapply.co.kr), Dept. of Biomedical Engineering, Konkuk University
***Hyun-Syug Kang (hskang@gnu.ac.kr), Dept. of Computer Science, GyeongSang National University
• Received: 2015. 09. 08, Revised: 2015. 09. 30, Accepted: 2015. 10. 15.

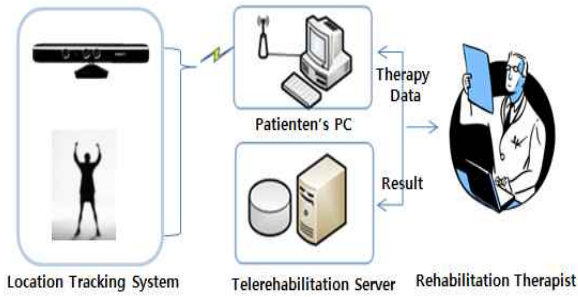


Fig. 1. The configuration of the proposed system

II. Theoretical background

1. Kinect Sensor

마이크로소프트사의 키넥트는 특정한 제어기가 없이 해당 객체의 신체 정보를 추적할 수 있는 센서이다. 키넥트 센서는 기본적으로 3개의 센서 렌즈를 가지고 있는데 영상을 인식하는 RGB 카메라와 적외선 송출 빔과 송출된 적외선 빔의 반사광을 받아들이는 적외선 감지 카메라로 구성되어 있다. 이를 통하여 실시간 깊이 정보와 RGB 영상과 관절 추적 정보를 얻을 수 있다. 이는 동작 인식을 위한 신체 검출과 자세 추정의 수고를 덜어주고 동작 인식과 관련된 다양한 분야에 쉽게 적용할 수 있는 장점을 가진다. 그리고 윈도우 환경에서 키넥트 센싱 정보를 얻기 위하여 키넥트 SDK를 사용하게 된다. 키넥트로부터 얻을 수 있는 스켈레톤 정보는 총 20개로서 최대 4개의 키넥트를 단일 시스템에 연결하여 인식할 수 있다[4, 5].

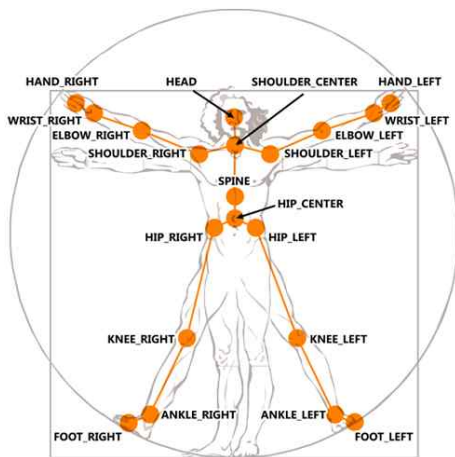


Fig. 2. Skeleton of Kinect Sensor

[Fig. 2]는 키넥트가 인지 가능한 20가지의 스켈레톤을 보이고 있다. 획득한 정보의 3차원 좌표를 사용하여 각 부위별 각도 값과 중심 값을 계산하여 시각화한다. 센서는 한번에 2명까지 스켈레톤 인식이 가능하고 사람의 인식은 최대 6명까지 가능하다. 그리고 사람 인식과 별도로 거리를 인식할 수 있기 때문에 관절의 뒤틀림 등을 쉽게 인식할 수 있다. 본 연구에서는

상지 운동을 위한 관절을 좌표 값을 획득하여 제공되는 가상현실 콘텐츠와 매핑하여 시각화 한다.

2. Skeleton Recognition using Kinect Sensor

본 연구에서 측정하는 관절 정보는 크게 세 가지이다. 먼저, 어깨의 위치 정보를 측정하기 위한 상완(upper arm) 정보와 팔꿈치와 손의 높이나 각도를 측정하기 위한 하완(lower arm) 정보, 그리고 손을 위치를 파악하기 정보를 추출한다. 팔꿈치 구부림 각도는 벡터의 내적을 이용하여 구하는데, 먼저 다음 공식 (1)과 같이 두 직선의 방향벡터를 구한다.

$$\vec{a} = (x_2 - x_1, y_2 - y_1, z_2 - z_1) \quad (1)$$

$$\vec{b} = (x_3 - x_1, y_3 - y_1, z_3 - z_1)$$

여기서, \vec{a} 는 팔꿈치 관절 위치좌표(x_1, y_1, z_1)와 하완 관절 위치좌표(x_2, y_2, z_2)의 방향벡터이고 \vec{b} 는 팔꿈치 관절 위치좌표(x_1, y_1, z_1)와 상완 관절 위치좌표(x_3, y_3, z_3)의 방향벡터이다. 이를 풀이하여 다음 공식 (2)에 대입하여 최종적인 사이각을 산출한다.

$$\theta = \cos^{-1} \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \cdot |\vec{b}|} \quad (2)$$

그리고 실제 시각화 영역에서 사용하는 사이각은 라디안 값을 디그리 값으로 변환($\theta \cdot 180/\pi$)하여 사용한다. 본 논문에서는 두 벡터의 수직성 여부를 각도 값으로 표기한다. 마지막으로 어깨선의 평행성 여부를 분석하기 위하여 양 어깨 관절과 어깨의 중심과 척추와 엉덩이를 잇는 방향벡터를 이용하여 평행성 여부를 분석한다. 두 θ 값을 분석하여 90도를 이루면 완전한 평행선을 이루었다고 판단한다.

3. SDK for Kinect Sensor

본 논문에서 제안한 재활 시각화 도구를 개발하기 위하여 마이크로소프트사에서 개발한 비영리 목적의 키넥트 윈도우용 소프트웨어 개발 키트(Kinect for Windows Software Development Kit)을 사용한다. 이 개발 도구는 윈도우 7 환경에서 키넥트 기술을 이용하여 깊이 감지, 동작 트래킹 등의 정보를 인식하여 응용 프로그램을 개발 할 수 있는 도구이다. 그리고 드라이버와 미가공 센서 스트림(Raw Sensor Streams), 내추럴 유저 인터페이스(Natural User interfaces), 설치 문서, 리소스 자료 등 풍부한 API를 포함한다. 키넥트 윈도우용 소프트웨어 개발 키트는 개발자들이 C++, C#나 Microsoft Visual Studio 2010을 이용한 비주얼 베이직(Visual Basic) 으로 응용 사례를 개발할 수 있도록 도와준다. 특히, 본 논문에서 사용하고자 하는 골격 트래킹에서는 가시 범위 안에서 움직이는 사람의 골격 이미지

를 트래킹 할 수 있다. 논문에서는 주로 사용되는 API는 골격 정보와 관련된 메소드로서 깊이 정보를 가져와서 해당 조인트 정보를 매핑하기 위한 좌표변환을 실시하여 실제 시각화 뷰에 표시한다. 주요 API 함수는 [Table 1]과 같다.

Table 1. Major Method of Kinect SDK

API	Funtion
KinectSensors[n]	Kinect information acquisition
SkeletonStream.Enable()	Skeleton processing enabled
OpenSkeletonFrame()	Skeleton Information Method
OpenDepthImageFrame()	DepthImage Information Method

아래의 코드는 척추(Spine)에 해당하는 조인트정보를 가져와 해당 정보를 이용하여 실제 깊이 영상에 매핑 시키는 코드의 일부분이다. 이 좌표 값은 상대 좌표이기 때문에 컬러스트림에 매핑 시키기 위하여 별도의 좌표 변환이 필요하다. 하지만 상지 관절 분석을 위한 영역에서는 별도의 좌표 변환 없이 조인트정보를 그대로 이용한다.

```

SkeletonFrame SkFrame =e.OpenSkeletonFrame();
Skeleton[] SkData = new
    Skeleton[SkFrame.SkeletonArrayLength];
SkFrame.CopySkeletonDataTo(SkData);
Spj = SkData.Joints[JointType.Spine];
depPoint =
    depthImageFram.MapFromSkeletonPoint(Spj.Position);
    
```

III. Design

1. Content Design Scenario

수평 운동은 왼쪽 시작 시점과 오른쪽 종료 지점과의 거리를 30cm로 한다. 각 경로는 점선으로 표시하고 도달지점에서 5cm 크기의 원을 제시한다. 수직 운동은 상의의 단추 채우기 등으로 활용할 수 있는데 위쪽 시작 지점과 아래쪽 종료 시점까지의 거리를 50cm로 하고 도착지점을 무작위로 변경하거나 선택적으로 지점을 위치하도록 한다. 원 운동은 반지름이 25cm 원을 제시하고 시계방향이나 반시계방향으로 선회를 있게 한다. 회전운동은 한 손으로 핸들을 잡고 제시한 각도만큼 빠르게 회전하게 한다. 난이도에 따라 각도와 회전 반경을 조절할 수 있게 한다.

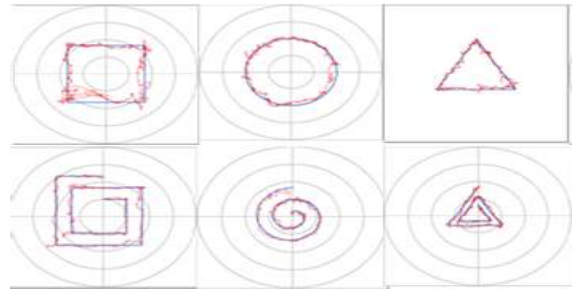


Fig. 3. Example of 2D Tracking Training

2. Virtual Reality Content

실시간으로 제공되는 콘텐츠에 따라 환자는 제시된 문제를 풀어가는 형식으로 구성된다. 2D 콘텐츠에서는 관절가동범위와 속도를 특정하여 가상현실 공간상에 피드백을 주어 문동의 정확성과 협응성을 가시화한다.

먼저, 2차원 관절 추적 훈련의 예시는 [Fig. 3]과 같이 구성한다. 도형을 중심으로 크기를 변화시키면서 상지 운동성을 측정하게 된다. 화면상에 도형이 등장하면 환자는 도형을 선분을 따라 상지 관절을 움직이고 실시간으로 추적된 정보를 가시화시킨다. 훈련 콘텐츠에는 크게 수직/수평운동, 원 운동, 회전 운동, 추적 운동으로 구성되어 있다.

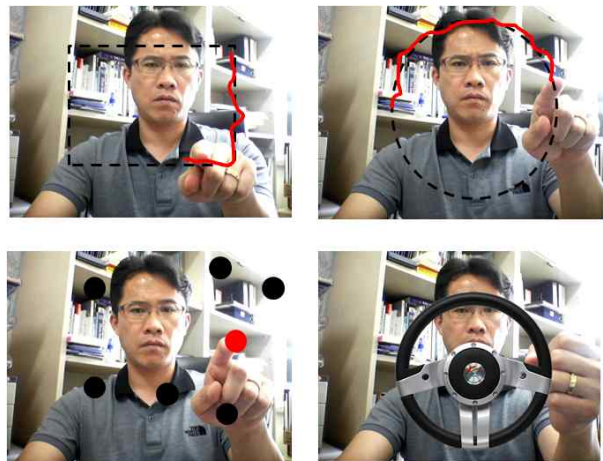


Fig. 4. Virtual reality content

[Fig. 4]의 상단 왼쪽은 수평/수직운동을 위한 가상 콘텐츠를 보이고 있다. 제시된 검정 가이드라인을 따라 가면 추적 정보를 실시간으로 빨간색으로 표시해 준다. 상단 오른쪽은 원 운동의 예시이고 하단 오른쪽은 회전 운동을 보이고 있다. 그리고 하단 왼쪽은 무작위로 등장하는 원 중에서 빨간색 원을 선택하는 콘텐츠로서 난이도에 따라 원의 크기와 개수를 조절할 수 있다. 그 외 재활 치료사의 설계에 따른 물결모양, 깃발 들기, 유리잔 옮기기 등의 다양한 콘텐츠를 구성할 수도 있다.

훈련 콘텐츠의 주요인수는 원주의 중심과 반경, 시작 위치, 방향, 위치 개수, 커서 크기 등을 고려해볼 수 있다. 이들 모든 인수를 환자의 상황을 고려하여 난이도도 설계될 수 있다. 예를

들어, 어깨의 외전이 힘든 환자의 경우 치료사가 이를 고려하여 몸의 중심 밖으로 훈련 궤적의 중심을 설계하고 어깨의 외전 운동을 유도할 수 있는 것이다. 훈련이 거듭되어 환자의 운동기능이 개선됨에 따라 커서의 크기를 줄여 난이도를 높여갈 수 있다.

3. The Storage System of Trace Information

훈련결과를 저장하고 재활 프로그램을 연동하기 위한 목적으로 원격재활 서버를 구성하여 정량화된 평가 결과가 원격재활 서버에 저장되고, 원격지에 위치한 치료사가 이 데이터를 참고하여 환자의 재활치료를 안내, 혹은 실시간으로 환자의 치료 과정을 모니터링 할 수 있도록 한다. 웹 서버와 데이터베이스 서버를 기반으로 재활훈련에 필요한 서버 측 스크립트와 DB 테이블을 구현한다. 주요 관리 데이터는 환자의 운동 상태에 따라 맞춤형 훈련제공. 운동결과와 과정의 정확성, 운동 중 발생한 인체 역학적 인수(관절가동범위, 속도, 방향성, 운동의 질)에 대한 가상현실 피드백 정보이다. 먼저 환자가 로그인을 하면 저장된 인적 정보에 의해 해당 훈련 프로그램을 보이고 키넥트 센서의 동작 시험 운동을 하게 된다. 센서가 각 관절을 찾아내면 해당 훈련 프로그램을 시작하는 음성 메시지가 나온다. 음성 메시지에 따라 방향등을 제시하면 환자는 메시지에 따라 훈련을 하게 된다. 환자의 관절 이동 정보는 해당 훈련 프로그램의 위치정보와 비교하여 실시간으로 표시하게 되고 해당 좌표정보는 재활 데이터베이스에 전송된다. 재활 치료사는 로그인 후 해당 좌표정보의 정합과정을 거친 실제 데이터를 보고 환자의 재활 정도를 평가하고 향후 프로그램을 설정한다.

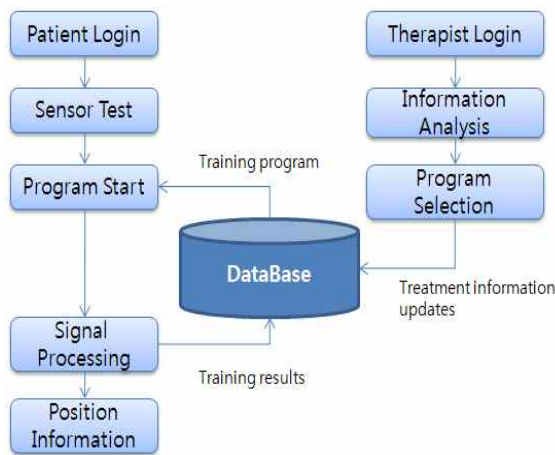


Fig. 5. The Storage System of Trace Information

4. Implementation Environment

본 논문에서 제안하는 가상현실 콘텐츠의 구현 환경은 [Table 2]와 같다.

Table 2. Implementation environment of System

Item	Contents
CPU	Intel(R) Core(TM)2 Duo CPU 3.16GHz
MEMORY	2GB
Graphic Card	NVIDIA GeForce 9300 GS
OS	MS Windows 7
Language	MS Visual Studio 2010 (Visual C#)
Graphic Library	OpenGL 1.4, GLUT, GLUI
etc.	Kinect for Windows SDK

제안하는 시스템을 구현하기 위한 환경은 윈도우즈 7 기반에서 C# 언어로 구현되었고 OpenGL[6]을 이용하여 그래픽을 표현하였다. OpenGL은 산업 표준화된 그래픽 라이브러리로서 향후 내장형 시스템 등에 개발할 때 사용가능성이 유리하다는 장점을 가진다. 그리고 윈도우 기능 및 입출력 제어를 위하여 GLUT(OpenGL Utility Toolkit) 라이브러리를 이용하였다[7]. GLUT는 OpenGL 과 드라이브 사이에서 인터페이스 역할을 담당하는데 필요한 콜백함수를 등록하고 해당 콜백함수에 원하는 내용을 채워 넣기만 하면 이에 대한 호출은 GLUT가 알아서 처리하게 된다. 그리고 GLUI(OpenGL User Interface Library)는 GLUT에 기반을 둔 사용자 인터페이스 라이브러리로 버튼, 체크박스, 라디오 버튼 등의 고급 사용자 인터페이스를 제공한다[8]. 메인 화면의 입출력 요소에 대한 인터페이스 설계에 이용되었다.

5. Implementation Algorithm

재활 프로그램은 키넥트 센서로부터 전달되는 상지의 움직임 데이터를 이용하여 재활 훈련 및 진단 평가를 위한 프로그램을 구현하였다. 2D 재활 프로그램은 진단과 훈련 프로그램 두가지로 구현된다. 진단 프로그램을 통해 훈련 실시 이전의 상지 움직임을 평가하고, 향후 훈련 난이도 조절이 가능하다. 일정 기간의 재활 훈련 후 진단 프로그램을 통해 훈련 전후의 상지 움직임을 평가하여 재활 프로그램의 유용성을 확인하고 더 나아가 환자의 재활 수준에 적절한 훈련 프로그램을 제공하고 자 한다. 뇌졸중으로 인한 장애 수준을 정량적으로 평가하기 위한 많은 검사 방법들이 있는데, 본 연구에서는 많은 검사 방법에서 운동 기능, 그 중에서 상지 장애의 기능적 회복을 평가하기 위해 상지 기능 평가 방법중 대표적인 방법으로 “Upper Extremity Motion Score”[9]와 “Fugl-Meyer Assessment Scale”[10]을 참고하여 공통점 도출하고, 그 결과 대부분의 상지 기능 평가에서 어깨의 Flexion, Abduction, Adduction, Elevation, Rotation과 팔꿈치의 Flexion, Extention, 전완의 Pronation, Supination 움직임 정도를 평가한다. 타겟이 어깨 중심의 수평면에서 시작 하여 각각 45도 각도로 증가하는 8개의 타겟을 향해 직선으로 이동하는 변화를 측정한다. 이때 타겟을 따라 상지를 움직여 커서를 함께 이동시켜 목표지점에 도달할 때까지의 Tin, Tout, Dc을 측정하고 각각의 목표지점에 따른 측정값의 평균을 구하여 결과로 나타낸다. [Fig. 6]의 이동

컨텐츠 1은 직선운동으로 어깨의 Abduction, External Rotation, Elevation과 팔꿈치의 Flexion, Extention 그리고 팔뚝의 Pronation, Supination을 평가한다.

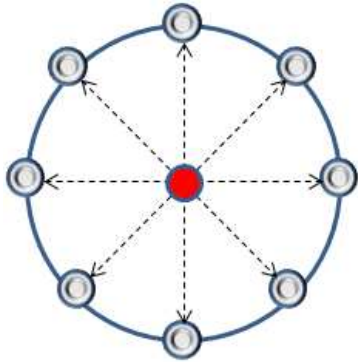


Fig. 6. Moving Contents 1

[Fig. 7]의 이동 콘텐츠 2는 원운동으로 어깨의 Adduction, External/Internal Rotation 과 팔꿈치의 Extention 그리고 팔뚝의 Pronation, Supination을 평가한다.

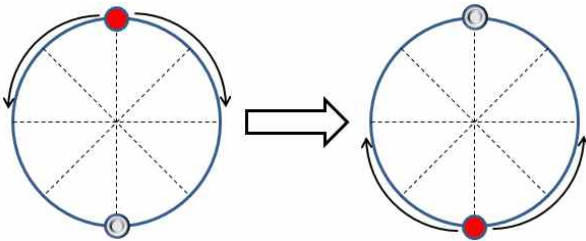


Fig. 7. Moving Contents 2

하단 Code 1에서는 실제 구현된 알고리즘 중 화면 드로잉을 위한 주요함수인 DrawScene 함수의 일부분을 보이고 있다. 프로그램이 시작되면 먼저, 좌표계를 저장하고 사용자의 위치나 화면의 크기에 따라 원근법으로 거리를 조정한다. 그리고 훈련 정보표시와 상지표시를 하고 상지에 대한 위치표시를 한다.

Code 1. DrawScene() Function

```
static BOOL bBusy = FALSE;
if(bBusy) return
bBusy = TRUE;

glClearColor(0.0f, 0.0f, 0.0f, 1.0f);
glClear(GL_COLOR_BUFFER_BIT |
        GL_DEPTH_BUFFER_BIT);

glPushMatrix();
glTranslatef(0.0f, 0.0f, -m_fRadius);
if( m_nTrain == TRAIN_PUSHING){
    DrawTrain();
```

```
}
else
{
    DrawTrain();
    DrawUpperExtreme();
}
DrawPosition();
glPopMatrix();
glFinish();
SwapBuffers(wglGetCurrentDC());
bBusy = FALSE;
```

Code 2의 DDrawTain 함수는 훈련 데이터의 가이드라인을 출력하는 함수로서 상지를 표시하기 위한 주요 정보가 되는 함수의 일부분이다. 오브젝트를 초기화하고 가이드라인을 표시한다. 그리고 커서를 표시하고 사용자의 상지 운동 정보를 출력하기 위한 함수를 호출한다.

Code 2. DDrawScene() Function

```
If( m_nMode != MODE_TRAIN) return
GLUQuadricObj *pObj;
pObj = gluNewQuadric();
gluQuadricNormals( pObj, GLU_SMOOTH);
int n = 20;
GLdouble cx, cy, cz;
cx = m_faTrainCenter(0);
cy = m_faTrainCenter(1);
cz = m_faTrainCenter(2);
GLdouble x, y, z;
z = cz;
glPushMatrix();
glColor3f(1.0f, 1.0f, 1.0f);
glBegin(GL_LINE_STRIP);
for(i=0; i<n; i++){
    double angle = (360.0 / n) * i * GL_PI / 180;
    x = m_fTrainRadius * cos(angle);
    y = m_fTrainRadius * sin(angle);
    glVertex3f( x, y, z);
}
glVertex3f( m_fTrainRadius, 0, z);
glEnd();
GLdouble gcx, gcy, gcz;
gcx = m_faPosCursor(0);
gcy = m_faPosCursor(1);
gcz = m_faPosCursor(2);
glTranslatef( gcx, gcy, gcz);
glColor3f(1.0f, 1.0f, 1.0f);
gluSphere( pObj, 0.07f, 26, 13);
glPopMatrix();
ProcTrain();
```

IV. Conclusions

본 논문을 통하여 설계한 가상현실 기반의 상지재활 콘텐츠는 뇌졸중이나 뇌경색으로 인한 뇌혈관 질환으로 인한 반신마비, 언어장애, 치매 등의 후유증을 기존의 병원이 아닌 가정에서 인터넷으로 연결된 원격재활 서비스를 통해 이루어짐으로 기존의 오프라인으로 병원에서의 치료 활동을 통한 서비스에서 원격 재활 치료 서비스의 새로운 기술 지표로 사용될 수 있다.

기존의 상지 재활운동기구의 경우 수동형으로 단순 반복 운동을 통한 재활기구이며 또한 시뮬레이션을 통한 기계 기구의 경우 고가여서 병원에서도 구매를 꺼리고 있는 것이 사실이다. 따라서 본 시스템의 경우 저가의 키넥트 센서와 간단한 프로그램 설치로 사용자가 가정에서 훈련을 수행하고 병원의 관리자가 이를 원격으로 관리함으로써 가격대비 최대의 효율과 환자의 빠른 회복을 기대할 수 있으며 이와 유사한 다른 재활 기구 개발에도 많이 응용될 수 있다.

국내 노인 인구의 증가와 각종 재활 복지 분야에서의 수요가 늘어남에 따라 기존의 인력을 바탕으로 한 재활 복지 분야의 기술적인 접목이 최근 화두로 되고 있다. 본 시스템은 기존의 각종 재활 복지 분야의 연구개발내용과는 다른 기구적 물리적 시스템이 아니라 기구적 물리적 부분에 IT 기술 분야를 융합하는 제품으로 국내의 재활 복지 분야의 연구에 많은 시너지를 발휘할 것으로 예상된다. 향후 연구에서는 보다 용이한 접근성을 위하여 HTML5 기반의 웹 환경에서 구동될 수 있는 재활 시스템을 구축하고자 한다.

REFERENCE

- [1] Patel, S. et al. "A Novel Approach to Monitor Rehabilitation Outcomes in Stroke Survivors Using Wearable Technology." Proceedings of the IEEE 98.3 pp. 450-461, 2010
- [2] Yao-Jen Chang, Wen-Ying Han, Yu-Chi Tsai, "A Kinect-based upper limb rehabilitation system to assist people with cerebral palsy", Research in Developmental Disabilities, vol 34, pp. 3654-3659, 2013.
- [3] B. Mijovic, M.B. Popovic and D.B. Popovic, "Synergistic control of forearm based on accelerometer data and artificial neural networks", Brazilian Journal of Medical and Biological Research Vol. 41, pp 389-397, 2008.
- [4] Microsoft Kinect, <http://www.microsoft.com/en-us/kinectforwindows/>
- [5] Y-N Park, S-M Seo, M-C Park, "The Implementation of Visualization Tool for Snowboard Using Kinect Sensor Data," Journal of the Korea Society of Computer and Information, Vol. 18, No. 5, pp. 53-60, May 2013.
- [6] Shreiner, Dave, "OpenGL Programming Guide," Addison-Wesley, 2009.
- [7] GLUT - The OpenGL Utility Toolkit, <http://www.opengl.org/resources/libraries/glut/>
- [8] GLUI User Interface Library, <http://glui.sourceforge.net/>
- [9] Bard, G., Hirschberg, G. G.: Recovery of Voluntary Motion in Upper Extremity Following Hemiplegia. Arch. Phys. Med. Rehabil. Vol. 46, pp. 567-572, 1965.
- [10] Roden-Jullig, A., Britton, M., Gustafsson, C., Fugl-Meyer, A.: Validation of Four Scales for the Acute Stage of Stroke. J. Intern. Med. Vol. 236, pp. 125-136, 1994.

Authors



Myeong-Chul Park received a B.S. degree in Computer Science from Korea National Open University in 1999, a M.S. and Ph.D. degrees in Computer Science from GyeongSang National University in 2002, 2007. He is currently a Professor in the Department of Software Academy, SongHo College. He is interested in visualization, simulation, education of software, virtual reality, and parallel programming.



Hyon-Chel Jung received B.S degree in Biomedical Engineering Konkuk University, Korea, in 2014. He is currently a M.S Courses in the Department of the Biomedical Engineering. He is interested in digital convergence business model, Health Cloud, mobile Health Care Service.



Hyun-Syug Kang received the M.S. and Ph.D. degrees in Computer Science from Seoul National Univ. in 1983 and 1989, respectively. During 1981-1985, he stayed in ETRI. And also during 1985-1993, he stayed in Chonbuk National Univ.

He has been a professor at Gyeongsang National Univ. since 1994. His research interests are in multimedia, embedded database, and intelligence system.