

## 키넥트 센서 데이터를 이용한 스노보드 동작 시각화 도구의 구현

박영남\*, 서세미\*\*, 박명철\*\*\*

# The Implementation of Visualization Tool for Snowboard Using Kinect Sensor Data

Young-Nam Park\*, Se-Mi Seo\*\*, Myeong-Chul Park\*\*\*

### 요약

본 논문에서는 키넥트 센서에 의해 획득한 스켈레톤 정보를 이용하여 스노보드 동작을 시각화 할 수 있는 도구를 제안한다. 스노보드 동작에서 가장 기본이 되는 동작은 BBP(Balanced Body Position) 자세로서 안정된 턴 동작을 위한 기본 기술이다. 본 논문에서는 BBP 자세를 분석하여 가장 이상적인 동작을 실현할 수 있는 시각화 도구를 구현한다. 키넥트 센서로부터 획득한 좌표정보를 이용하여 발목, 무릎, 엉덩이 관절의 각도와 몸의 중심축을 추적하여 표준 자세와 비교분석한다. 분석결과는 OpenGL 라이브러리를 통하여 최종 결과 화면을 구성하였다. 본 연구결과는 스노보드 턴 동작을 분석하는 기초자료로 활용될 수 있을 것이다.

▶ Keywords : 스노보드, 키넥트 센서, 동작 분석, 시각화

### Abstract

This paper proposed visualization tool for motion of snowboarding using Skeleton data obtained by the Microsoft's Kinect sensor. The BBP(Balanced Body Position) posture is a most basic motion in the Snowboarding. This posture is the primary technology for stable turns. The implementation of visualization tool to analyse the BBP posture of snowboard. comparative analysis with standard postures to the ankles, knees, hips and spine angle of joints and body tracking using coordinate information obtained by the Kinect Sensor. Analysis of the final results of the screen through the OpenGL library. This research result could be used to analysis for turn postures of snowboarding.

•제1저자 : 박영남 •교신저자 : 박명철

•투고일 : 2013. 4. 20, 심사일 : 2013. 5. 10, 게재확정일 : 2013. 5. 14.

\* 송호대학교 스포츠레저과(Dept. of Sports Leisure, SongHo College)

\*\* 서울여자대학교 체육학과(Dept. of Physical Exercise, Seoul Women's University)

\*\*\* 송호대학교 보건의료전자과(Dept. of Biomedical Electronics, SongHo College)

※ 이 논문은 2013년 한국컴퓨터정보학회 제47차 동계학술대회에서 발표한 논문("스노보드 동작 시각화를 위한 설계")을 확장한 것임.

▶ Keywords : Snowboard, Kinect Sensor, Motion Analysis, Visualization

## I. 서론

동계 스포츠 종목에서의 동작에 따른 차이를 측정하고 운동학적 분석을 통한 기술적 연구는 활발하게 진행되어 왔다. 그러나 실제 동작을 그래픽컬하게 시각화하는 연구는 제한적이었다. 또한 대부분의 분석 연구는 촬영후 영상을 후처리하는 과정으로 진행되어 왔기 때문에 실시간 동작을 분석하는 연구는 일반화되지 않은 것이 현실이다[1-3].

스노보드는 슬로프 등을 라이딩하는 동계 스포츠로서 양쪽 발이 한 면에 고정된 상태에서 슬로프를 진행하며 보드를 평지에 놓았을 때 보드의 중앙부의 굴곡 부분이 올라온 캠버(camber)에서 원리가 비롯된다. 이는 체중을 보드 전체에 분산되도록 하여 보드 활주시 곡선을 그리는 사이드 커브(side curve) 현상이 나타나게 되어 보드의 에지 전체가 곡선을 그리면서 카빙하게 된다.

스노보드의 턴 동작에는 너비스 턴과 카빙 턴이 있는데 너비스 턴은 턴 동작 시 엇다운을 하여 에지를 이용하는 방법이고 카빙 턴은 어깨, 골반, 무릎 순으로 진행하여 보드 에지를 절면체 기압하고 슬라이딩을 최소화시켜서 라이딩하는 기술이다.

스노보드 턴 동작에서 가장 기본이 되는 동작은 BBP(Balanced Body position) 동작으로 발목과 무릎, 엉덩이 관절을 구부려서 엉덩이가 조금 낮아지도록 하고, 몸의 축은 보드와 수직이 되고 어깨선이 평행선이 되도록 하는 자세이다.

이러한 턴 동작 기술에 대한 분석연구가 꾸준히 지속되어 왔는데 Doki[1]는 턴 동작의 운동역학적 분석의 결과를 보였고 현무성 등[2]은 관절의 각도와 각속도 등을 분석하는 등 대부분의 연구가 운동학적 분석연구가 주류였다.

이에 본 연구에서는 스노보드 턴 동작의 효율성과 적합성을 가시화하기 위하여 시각화 도구를 제안한다. 분석에 사용되는 턴의 종류는 너비스 턴과 카빙 턴을 대상으로 하고 해당 턴 시에 BBP 자세를 분석하여 그래픽 라이브러리를 통하여 시각화 한다. 또한 보더의 동작 인식을 위하여 동영상 촬영이 아닌 키넥트 센서를 이용하여 실시간으로 동작을 인식하고 상태 정보를 정형화된 수치데이터와 비교하여 자세를 교정할 수 있는 정보를 제공한다. 이를 통하여 스노보더 라이딩 동작을 이해하고 교정할 수 있는 기초 자료로 활용될 것이다.

본 논문의 구성은 다음과 같다. 2장에서는 연구 대상과 키

넥트 센서 및 연구 절차와 환경을 중심으로 연구방법을 소개한다. 3장에서는 제안하는 구현시스템의 구조와 결과를 보이고 마지막으로 4장에서 결론 및 향후 연구 방향에 대해 기술한다.

## II. 연구방법

### 1. 연구 대상 선정

본 논문의 시스템 구현을 위하여 동작 정보를 제공하는 보더들은 [표 1]과 같이 S대학교 스포츠레저과(강원도 횡성군 소재)의 스노보드 대표선수 5명을 대상으로 하였다. 분석치의 차이를 구분하기 위하여 초급, 중급, 상급자를 각각 구분하여 선정하였다.

표 1. 실험 대상자의 특성  
Table 1. Characteristic of Subjects

대상	나이 (세)	신장 (Cm)	체중 (Kg)	수준
L1	21	180	75	초급
M1	22	172	62	중급
M2	22	176	65	중급
H1	23	181	78	고급
H2	24	170	60	고급

### 2. 키넥트 센서를 이용한 동작 인식

마이크로소프트사의 키넥트는 특정한 제어가 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].

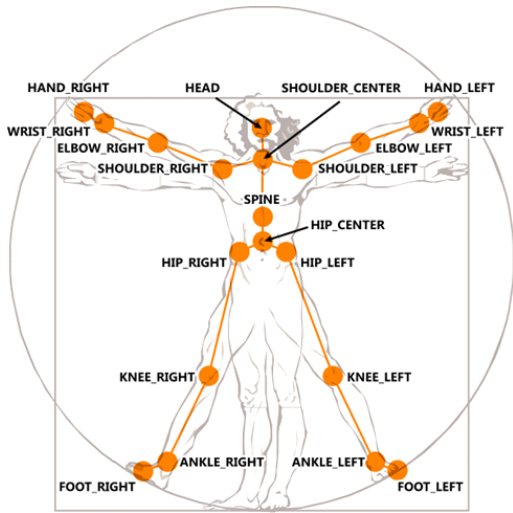


그림 1. 키넥트 센서의 스켈레톤  
Fig. 1 Skeleton of Kinect Sensor

기존의 연구에서는 관절정보를 인식하기 위하여 스노보더가 진신 타이즈를 입고 관절에 테이프를 부착하여 실제 슬로프를 내려오면서 비디오카메라를 이용하여 구간별 촬영을 하게 된다. 이때 셔터의 스피드나 촬영 각도 등의 차이에 의해 사용될 정보를 수집하기가 매우 어려웠다. 그러나 키넥트 센서를 사용하므로 실시간 동작을 인식할 수 있게 되어 손쉽게 동작 분석용 정보를 얻을 수 있다[5].

[그림 1]은 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 일정한 한 방향의 측정값이 아닌 전후방과 좌우의 네방향에서 각각 좌표 값을 획득하여 보다 사실적이고 정확한 시각화 정보를 적용한다.

### 3. 연구 절차와 방법

연구의 실험은 키넥트 센서의 특성상 실내에서 이루어진다. 슬로프 영상을 보여 주면서 해당 시점에서 자유롭게 동작할 수 있게 하고 4대의 키넥트 센서를 이용하여 전후좌우의 동작을 인식한다. 본 실험에서는 별도의 보드 구동체를 사용하지 않아 동작을 역동성이 부족하다는 제한점을 가지고 있지만 향후 하단 보드의 유동적인 구동체를 이용하여 실험에 대한 정확성과 효율성을 배가시킬 예정이다. [그림 2]는 전체적

인 실험 환경이다. 자료의 추적간격은 0.5초 단위로 수집하여 자료 정합 모듈의 입력으로 이용한다. 전체적인 시스템의 구조는 3장에서 언급한다.

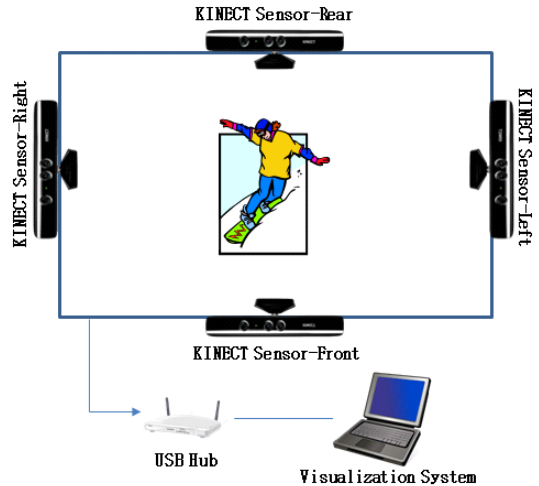


그림 2. 실험 환경  
Fig. 2 Experimental conditions

본 연구에서 측정하는 운동학적 변인은 BBP 자세를 분석하기 위한 용도로 크게 다섯 가지이다. 먼저, 발목의 구부림 정도(BBP1)를 측정하기 위하여 좌우 발목관절(Ankle.R, Ankle.L)을 좌표정보와 좌우 무릎(Knee.R, Knee.L)의 좌표정보 그리고 좌우 발(Foot.R, Foot.L) 좌표정보를 추적하여 사이각을 산출한다. 이때 각도계산의 기준 점은 좌우 발목관절(Ankle.R, Ankle.L)을 점점으로 본다. 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 발목관절 위치좌표( $x_1, y_1, z_1$ )와 발 위치좌표( $x_2, y_2, z_2$ )의 방향벡터이고  $\vec{b}$ 는 발목관절 위치좌표( $x_1, y_1, z_1$ )와 무릎관절 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. 위와 동일한 방법으로 무릎의 구부림 정도(BBP2)를 측정하기 위하여 좌우 무릎관절(Knee.R, Knee.L)의 좌표정보, 좌우 엉덩이(Hip.R, Hip.L)의 좌표정보, 좌우 발목관절(Ankle.R, Ankle.L)을 좌표정보를 추적하여 사이각을 산출한다. 이때 각도계산의 기준 점은 좌우 무릎(Knee.R, Knee.L)을 접점으로 본다.

엉덩이의 구부림 정도(BBP3)를 측정하기 위하여 척추 중심 좌표정보와 좌우 엉덩이(Hip.R, Hip.L)의 좌표정보, 좌우 무릎관절(Knee.R, Knee.L)의 좌표정보를 추적하여 사이각을 산출한다. 이때 각도계산의 기준 점은 좌우 엉덩이(Hip.R, Hip.L)를 접점으로 본다.

몸의 축과 보드의 수직성 여부(BBP4)를 분석하기 위하여 엉덩이의 중앙(Hip.C)과 척추(Spine) 좌표정보, 그리고 보드의 방향 벡터를 위하여 두 발의 좌표 정보(Foot.R, Foot.L)를 이용한다. 여기서,  $\vec{S}$ 는 엉덩이와 척추의 방향벡터이고  $\vec{F}$ 는 양 발의 방향벡터이다. 식 (3)과 같이 두 방향벡터가 수직일 조건은  $\vec{S} \cdot \vec{F} = 0$  이다.

$$\vec{S} \cdot \vec{F} = |\vec{S}| \cdot |\vec{F}| \cos\theta = 0 \quad (3)$$

$$\cos\theta = 0$$

$$\therefore \theta = \frac{\pi}{2}$$

본 논문에서는 두 벡터의 수직성 여부를 각도 값으로 표기한다. 마지막으로 어깨선의 평행성 여부(BBP5)를 분석하기 위하여 양 어깨 관절(Shoulder.R, Shoulder.L)과 어깨의 중심(Shoulder.C)과 척추(Spine)와 엉덩이(Hip.C)를 잇는 방향벡터를 이용하여 평행성 여부를 분석한다. 아래 그림과 같이 두  $\theta$  값을 분석하여 90도를 이르면 완전한 평행선을 이루었다고 판단한다.

#### 4. Kinect for Windows SDK

본 논문에서 구현하고자하는 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 이미지를 트래킹 할 수 있다. 논문에서는 주로 사용되는 API는 골격정보와 관련된 메소드로서 깊이 정보를 가져와서 해당 조인트 정보를 매핑하기 위한 좌표변환을 실시하여 실제 시각화 뷰에 표시한다. 주요 API 함수는 [표 2]와 같다.

표 2. 키넥트 SDK의 주요 메소드  
Table 2. Methods of Kinect for SDK

API	기능
KinectSensor.KinectSensors(n)	n번 키넥트 정보 취득
SkeletonStream.Enable()	스켈리톤 처리 활성화
OpenSkeletonFrame()	골격정보 메소드
OpenDepthImageFrame()	깊이영상정보 메소드

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 BBP 자세 분석을 위한 영역에서는 별도의 좌표 변환 없이 조인트정보를 그대로 이용한다.

#### Code:

```

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

### III. 3차원 가시화 시스템

#### 1. 전체 시스템 구조

본 논문에서 구현하는 전체적인 시각화 시스템은 [그림 3]에서 보이고 있다. 먼저, 동작정보를 수집하여 스켈레톤 DB에 저장한다. 저장된 데이터는 관절정보 정합 과정을 거치고 각 관절에 대한 추적정보를 생성한다. 최종적으로 사용자 UI를 이용하여 원하는 관절에 대한 자세정보를 선택한다. 선택된 자세 정보는 동작시각화 엔진을 통하여 사용자에게 보이게 된다. 이때 스켈레톤 DB와 시각화 엔진의 데이터 동기화를 위하여 공유 메모리를 사용하여 전달된다. 제안하는 시각화 시스템을 구현하기 위한 환경은 윈도우즈 기반에서 C# 2010 언어로 구현되었고 OpenGL을 이용하여 그래픽을 표현하였다. OpenGL은 산업 표준화된 그래픽 라이브러리로서 향후 내장형 시스템 등에 개발할 때 사용가능성이 유리하다는 장점을 가진다.

MMF(Memory Mapped File)[6]를 이용하여 프로그램 간에 데이터를 주고받을 수 있는데 파일과 메모리 객체를 연결시킨 후 그 메모리 객체에 이름을 지어준다. 그 이름을 이용하여 메모리에 데이터를 읽고 쓰는 작업을 하게 되면 그 내용이 메모리와 연결되어 있는 파일에 똑같은 효과가 적용되는 구조이다. 본 시스템에서는 파일을 사용하지 않고 특정 메모리 객체를 생성하여 통신한다.

동작 모션의 상태정보는 BBP와 관련된 다섯 개의 단위 정보와 상태의 변화를 위한 교정 정보로 이루어지며 [표 3]과 같다. 그리고 정합 과정을 통하여 동작 그래프를 드로잉하기에 불분명한 데이터는 삭제하여 테이블에 적재한다.

표 3. 시각화를 위한 주요 정보  
Table 3. Main Information for Visualization

정보	내용
BBP1	발목의 구부림 정도
BBP2	무릎의 구부림 정도
BBP3	엉덩이의 구부림 정도
BBP4	몸의 축과 보드의 수직성 여부
BBP5	아개선의 평행성 여부
SX1	BBP1의 정합 정보
SX2	BBP2의 정합 정보
SX3	BBP3의 정합 정보
SX4	BBP4의 정합 정보
SX5	BBP5의 정합 정보

#### 2. 시각화 정보처리 모듈

센서로 부터 전송된 정보는 시각화를 위하여 정보처리 모듈에 입력되게 된다. 정보처리 모듈은 크게 네 영역으로 나뉘는데 첫 번째 세부 모듈은 개별 관찰정보 수집 모듈로서 네 개의 센서에서 전달된 데이터를 센서별로 구분하는 역할을 한다. 그리고 두 번째 세부 모듈인 관절정보 정합 과정에서는 불분명하거나 이전 데이터보다 비상식적으로 급변하는 데이터는 오류정보로 간주하여 삭제하게 된다. 세 번째 세부 모듈인 관절 추정정보 분석에서는 이전 데이터의 좌표 정보와 비교하여 방향성과 각도를 측정하여 분석 그래프에 보일 데이터를 가공하는 역할을 한다. 마지막으로 시각화 정보처리 모듈에서는 최종적으로 시각화 시스템에 표시될 데이터를 선택된 메뉴의 BBP정보에 따라 OpenGL 좌표 데이터로 변환하여

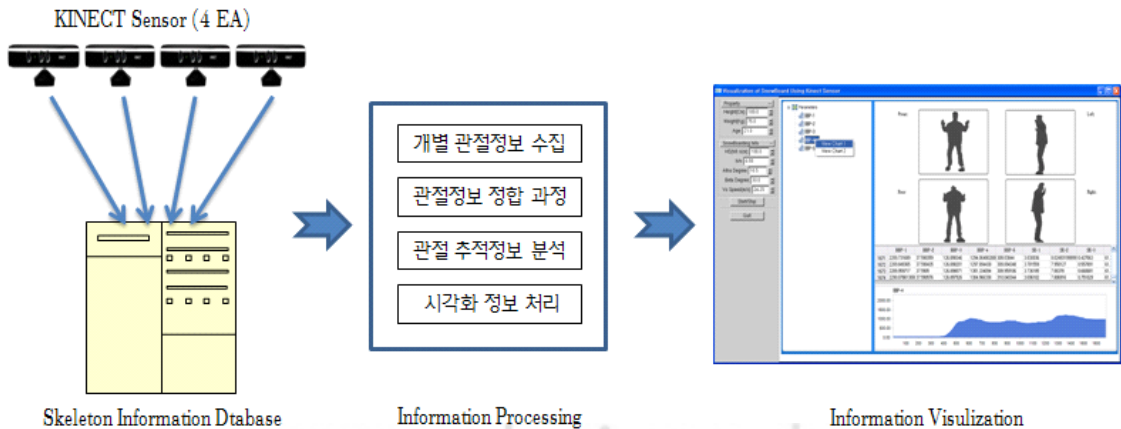


그림 3 시각화 시스템의 구조  
Fig. 3 An architecture of Visualization System

시각화 시스템에게 넘겨주는 역할을 한다.

### IV. 구현 환경과 결과

#### 1. 구현 환경

본 논문에서 제안하는 시각화 시스템의 구현 환경은 [표 4]와 같다.

표 4. 시스템의 구현 환경  
Table 4. Implementation Environment of System

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

제안하는 시각화 시스템을 구현하기 위한 환경은 윈도우즈 7 기반에서 C# 언어로 구현되었고 OpenGL[7]를 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[8]. 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 기반한 사용자 인터페이스 라이브러리로 버튼, 체크박스, 라디오 버튼등의 고급 사용자 인터페이스를 제공한다[9]. 메인 화면의 입출력 요소에 대한 인터페이스 설계에 이용되었다.

#### 2. 구현 결과

구현된 전체 시스템의 화면 구조는 [그림 4]와 같이 선수의 정보를 보이는 화면 좌측 GLUI 화면과 분석하고자 하는 BBP 정보를 선택할 수 있는 중앙의 트리구조, 그리고 우측 화면의 상단과 같이 네 개의 센서에서 넘어오는 선수의 깊이 영상정보와 하단의 실제 데이터 및 변화의 추이를 보이는 그래프 출력부로 구성되어 있다. 분석자는 선수의 깊이 영상정보를 통하여 1차적인 동작의 상태를 점검하고 하단의 실제 데이터를 통해 어떤 시점에서 해당 동작의 신체 각도의 변화 추이

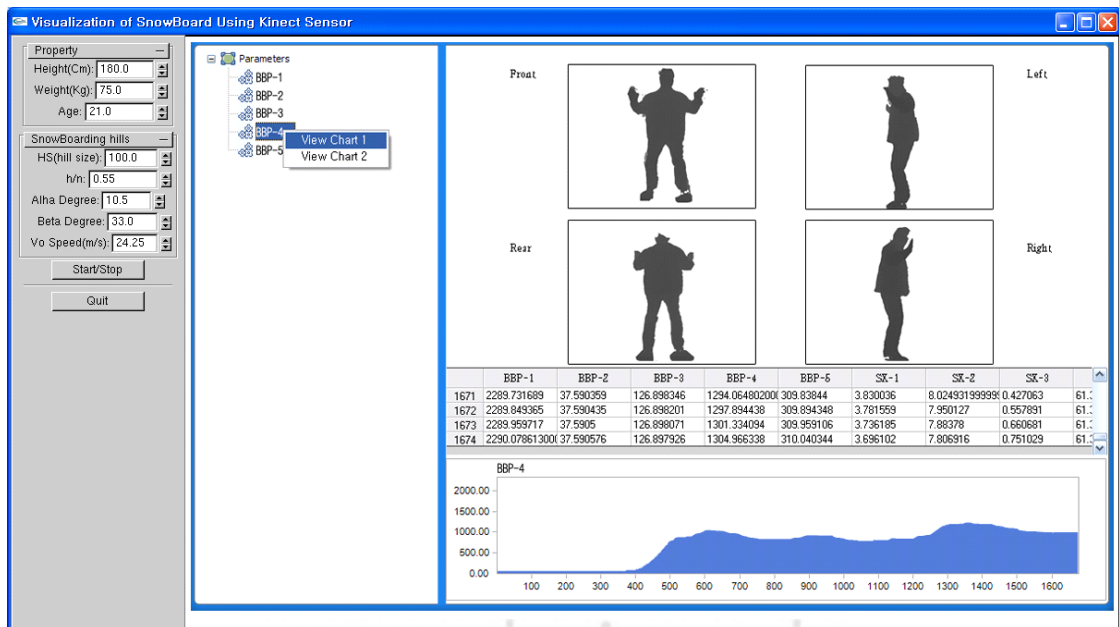


그림 4 시각화 시스템의 통합  
Fig. 4 Integration of Visualization System

를 확인 할 수 있다. 이 정보를 통하여 올바른 자세를 취하는 선수의 변화 추이와 그렇지 못한 선수의 변화 추이를 비교분석하여 선수의 동작을 교정하고 특정 시점에서의 BBP 자세의 각도를 추천할 수 있게 된다. 또한 실시간으로 BBP 자세의 다섯 가지 정보를 선택적으로 변화시키면서 특정 시점에서의 전체적인 자세의 변화 추이를 분석하여 최적의 자세를 유도할 수 있는 선수 지도 정보를 얻게 된다. 이는 기존의 동영상 상을 통한 분석 도구보다 효율적인 정보를 제공해 줄 수 있다는 장점을 가진다.

[그림 4]의 L1 선수의 경우에는 평면시점에서 경사지점으로 변화되는 시점에서 무릎의 구부림 정도(BBP2)와 엉덩이의 구부림 정도(BBP3)의 각도가 너무 커서 안정감과 가속도를 내는데 큰 문제점을 보였다. 이를 통하여 경사면의 정점에 위치하기 전에 BBP2, BBP3 각도를 줄여 향상된 자세와 경기력을 보이게 할 수 있었다.

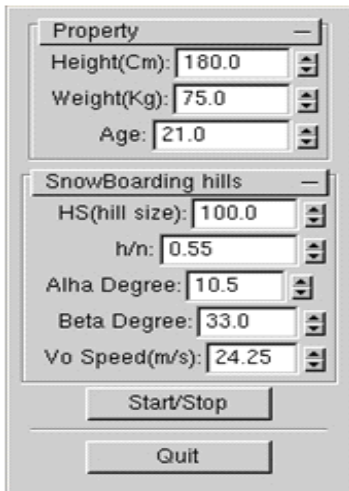


그림 5. 사용자 인터페이스 메뉴  
Fig. 5 User Interface Menu

[그림 5]는 사용자 인터페이스 화면에서는 선수정보와 슬로프의 상태정보를 조정할 수 있다. 키넥트 센서의 특성상 개방공간인 스키장에서 실험하기에는 제한적인 제약으로 인하여 가상의 슬로프 화면을 보고 동작을 하기 때문에 가상 화면에 대한 상태정보를 변화 시킬 필요가 있다. [그림 4]의 상태 정보는 반복성을 통하여 자세 변화를 확인하여 동작을 평가할 수 있고 향후 동일 영역에 대해 별도의 차트를 구성하여 비교 분석 작업을 거쳐 동작 개선에도 활용할 수 있다.

## V. 결론

본 논문은 키넥트 센서에 의해 획득한 스켈레톤 정보를 이용하여 스노보드 동작을 시각화 할 수 있는 도구를 제안하였다. 보드 동작의 기본이 되는 BBP 자세 정보를 이용하여 상태 변화를 시각화하여 사용자에게 제공한다. 기존 도구와 같이 별도의 오프라인 동영상 상을 대상으로 하지 않고 센서의 정보를 이용하여 실시간으로 시각화를 제공함으로써 경제적이고 활용성 높은 가시화 도구를 구현하였다. 또한 OpenGL모델을 이용하여 사용자에게 식별성 높은 화면을 제공한다. 본 연구결과는 스노보드 동작 인식을 위한 분석과 선수 지도에 이용될 수 있을 것이다. 그러나 본 도구는 가상의 화면을 통한 동작 변화를 보여야 하기 때문에 사실성이 떨어지고 환경 변화에 따른 영향정보를 적용치 못한다는 문제점이 있다. 향후에는 이 문제점을 극복하기 위하여 실제 환경과 유사한 모의 보드를 개발하고 불규칙적으로 변화할 수 있는 슬로프 환경도 도구에 추가하여 실용적 가치를 높이는 연구 활동을 지속할 예정이다.

## 참고문헌

- [1] Doki, H., Yamada, T., Nagai, C., & Hokari M. "Studies on the Dynamic Analysis of Snowboarding Turn," The Japan Society of Mechanical Engineers, B15, pp. 4-26, 2004.
- [2] Hyun, M. S., Lee, S. C., "Kinematic Analysis of Novice Turn and Carving Turn in Snowboard," Korea Society for Wellness, Vol 7, No. 1, pp. 185-194, 2012.
- [3] Park, M. C., Park, S. G., "The Implementation of Visualization for Ski Jump Using OpenGL," Journal of the Korea Society of Computer and Information, Vol. 16, No. 11, pp. 137-143, November 2011.
- [4] Microsoft Kinect, <http://www.microsoft.com/en-us/kinectforwindows/>
- [5] S.Y. Cho, H.R. Byun, H.K. Lee, and J.H. Cha, "Hand Gesture Recognition from Kinect Sensor Data," Journal of Broadcast Engineering, vol.17, no.3, pp.447-458, May 2012.
- [6] Park, M. C., Hur, H. R., "Implementation of the Flight Information Visualization System using

Google Earth," Journal of the Korea Society of Computer and Information, Vol. 15, No. 10, pp. 79-86, October 2010.

- [7] Shreiner, Dave, "OpenGL Programming Guide," Addison-Wesley, 2009.
- [8] GLUT - The OpenGL Utility Toolkit, <http://www.opengl.org/resources/libraries/glut/>
- [9] GLUI User Interface Library, <http://glui.sourceforge.net/>

### 저 자 소 개



**박 영 남**  
2012 : 한양대학교  
생활스포츠학과 체육학박사  
현 재 : 송호대학교 조교수  
관심분야 : 동작구현 및 시각화,  
동작분석, 프로그래밍  
Email : pyn@songho.ac.kr



**서 세 미**  
2007 : 한양대학교  
생활스포츠학과 체육학박사  
현 재 : 서울여자대학교 초빙교수  
관심분야 : 스포츠 동작 분석 및 시각화  
Email : semi1013@hanmail.net



**박 명 철**  
2007 : 경상대학교  
컴퓨터과학과 공학박사  
현 재 : 송호대학교 조교수  
관심분야 : 시각화, 임베디드 소프트웨어,  
병렬프로그래밍 및 디버깅  
Email : africa@songho.ac.kr