

## 유니티와 키넥트를 이용한 증강현실 시스템 구축 연구

\*장익준 \*\*이유진 \*박구만

\*서울과학기술대학교 전자미디어IT공학과

\*\* 일반대학원 IT미디어공학전공

e-mail : changikjun@naver.com

## A Study on the Implementation of Augmented Reality System using Unity and Kinect

\*Jang Ik-Jun, \*\*Lee Yu-Jin, \*Park, Goo-Man

\*Dept. of Electronics and Media IT Engineering

\*\*Dept. of Information Technology and Media Engineering

Seoul National University of Science and Technology

### 요약

최근 가상현실 및 증강현실 기술을 응용한 콘텐츠 개발이 활발하다. 증강현실 기술은 가상현실 기술과 다르게 현실 세계와 컴퓨터 그래픽과의 융합을 토대로 개발된다. 때문에 증강현실은 현실감, 몰입감을 극대화 하는 방향으로 개발되고 있다. 그러나 현실과 컴퓨터 그래픽은 서로 다른 차원에 존재한다. 이를 해결하기 위해 마커검출 기법과 오차를 미리 계산해 출력 전에 반영해주는 방식이 제시되었지만 이는 사용자의 자유도 향상과 공간적인 제약을 최소화하기에 부담이 크다. 본 논문에서는 캘리브레이션을 통한 증강현실 구현결과로서 사용자의 관절 정보를 증강현실 내 캐릭터의 관절과 맵핑한다. 이는 사용자의 움직임을 캐릭터와 동기화함으로써 증강현실의 자유도와 몰입도를 향상시켰다. 제안된 증강현실 시스템은 사용자의 생활에 적용될 수 있는 범위가 넓어지고 사용자의 편의를 향상시킬 수 있는 가능성이 높았다.

### 1. 서론

가상 현실과 증강 현실은 점점 더 적용 범위가 넓어짐에 따라 가상 현실과 증강 현실 모두 각각의 장점을 지니고 효과를 극대화 하는 관련 기술이 개발되고 있다. 가상 현실은 가상의 환경을 창조해 그 환경 내에서 사용자의 시각을 기반으로 콘텐츠를 개발하는 것이다. 반면 증강 현실은 현실 세계를 기반으로 가상의 콘텐츠를 융합하여 현실에서 체험한다. 때문에 가상 현실보다 높은 현실감을 느낄 수 있다. 그러나 현실기반의 증강 현실은 가상 현실과는 다르게 현실 객체를 항상 인식해서 그 움직임을 계산하고 그래픽을 합성해야 한다. 이 과정에서 현실과 스크린 상의 오차가 발생한다. 이를 해결하기 위해 현실 세계에 마커를 표시해 컴퓨터가 이를 인식하는 마커 검출 기법이 보편적으로 쓰이고 있다[1]. 그러나 이 방법은 현실 세계의 공간에 일정 부분에 마커를 표시, 위치해두어야 하며 마커가 인식 될 수 있는 환경을 조성해야 하는 단점이 있다. 이는 특히 사람처럼 동적인 객체가 대상이 될 경우 불리하다. 또한 논문[2]에서 현실 공간과 증강 현실 간의 오차를 줄이기 위해 대응한 방법으로 키넥트의 컬러카메라와 깊이 카메라 간 위치 오차를 미리 고려하여 출력했다. 이는 사

용자의 활동 범위에 제약이 있어 활동적인 증강 현실 콘텐츠 개발에 한계가 있다. 본 논문에서는 앞서 언급한 두 가지 문제점을 해결하기 위해 별도의 마커 없이 객체의 관절 정보를 얻을 수 있는 키넥트 카메라를 사용한다. 또한 키넥트의 정보를 기반으로 가상의 콘텐츠를 융합하기 위해 유니티를 이용해 증강 현실 시스템을 제안하고자 한다.



그림 1. Kinect for windows

### 2. 관련 연구

#### 2.1. Unity

유니티는 엔진 자체에 라이트 맵핑, 물리 엔진 등의 미들웨어를 탑재해 가상의 물체에 실제 물질의 특성을 적용할 수 있으며 이는 가상환경 내 실감도를 높여준다. 도형으로의 프로그래밍을 위한 언어인 유니티 볼트를 사용하

면 코딩에 익숙하지 않은 일반인도 프로그래밍이 가능하다.

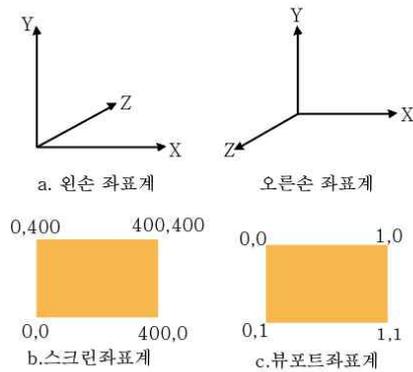


그림2. 유니티 좌표계

유니티 좌표계는 그림2와 같이 3차원 월드 좌표계, 유니티 스크린 좌표계, 2차원 뷰포트 좌표계가 있다. 그림2 a는 월드 좌표계이다. 월드 좌표계는 화면의 중심을 원점 (0,0,0)으로 하는 3차원 상대 좌표계이다. 카메라의 위치나 회전상태에 따라서는 화면의 중심이 아닌 점이 원점이 될 수도 있다. 그림2의 b는 스크린 좌표계로 화면의 왼쪽 아래를 원점으로 한다. 그림2의 c는 뷰 포트 좌표계로 스크린 좌표계와 비슷하지만 이를 정규화한 좌표계이다.

본 논문에서 가상현실 내 환경, 캐릭터를 디자인하고 키넥트와 연동하여 활용하기 위해 유니티를 사용한다.

## 2.2. 키넥트(Kinect)

키넥트는 사용자의 동작을 인식하고, 마이크 모듈로 음성을 인식한다. 실시간으로 깊이 정보뿐만 아니라 RGB영상과 관절 추적 정보를 제공한다. 키넥트는 동시에 최대 6명을 인식할 수 있으며 각 신체에서 25개의 관절 정보를 얻을 수 있다.



3.a 컬러카메라 이미지 3.b 깊이카메라 이미지



3.c IR카메라 이미지

그림3은 키넥트의 각 카메라로 얻은 이미지이다. 카메라마다 다른 좌표계를 가진다. 그림3의 a는 컬러 영상이다. 좌표는 좌 상단을 원점(0,0)으로 하며 0과1 사이의 값

을 가진다. 왼쪽으로 가면 X축의 증가, 아래쪽으로 가면 Y축의 증가이다. 해상도(캔버스)에 따라 0과 1 사이의 값에 곱해주면 해상도에 따른 좌표 값을 얻을 수 있다. b는 깊이 카메라이며 깊이 카메라의 좌표계도 0과1 사이의 값을 반환한다. c의 적외선 카메라의 원점은 키넥트 카메라의 적외선 센서 중앙이다. 원점을 중심으로 실제 공간에서의 좌표를 찾는다. 그림1을 따라 좌표가 증감하며, 캘리브레이션을 통해 각 좌표계 간 변환이 가능하다.

본 논문에서 유니티 상에서 3D 캐릭터의 관절에 사용자의 실제 관절 상에 맵핑 시키기 위하여 키넥트를 유니티와 연동한다.

## 2.3 캘리브레이션

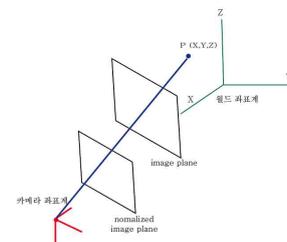


그림4.a 카메라 좌표계

실제 3차원인 세상의 좌표계를 월드 좌표계라 한다. 카메라가 이 정보를 가져오게 되면 3차원에 있는 정보들이 2차원 평면에 투사된다. 카메라는 내부, 외부에서 각 파라미터를 가진다. 내부 파라미터로는 초점거리, 주점 그리고 비대칭 계수가 존재한다. 내부 파라미터는 일정하며 대부분 카메라 공정 과정에서 발생하며 일정하다. 외부 파라미터는 카메라의 위치와 각도에 따라 달라지는 특징을 지닌다. 캘리브레이션은 3차원 정보를 2차원으로 가져오기 위해 카메라 내부 및 외부 요인의 파라미터 값을 구하는 과정이다.

## 3. 증강현실 시스템

### 3.1 증강현실 시스템 설계

그림 5는 증강현실 시스템을 구현하기 위한 설계도이다. 먼저 키넥트에서 관절정보를 받아오면 이를 유니티로 옮기기 위한 과정이 필요하다. 마이크로소프트사에서 제공하는 유니티용 플러그인을 사용한다. 다음으로 유니티에서 받은 정보를 유니티 캐릭터의 관절과 동기화 한다. 그 전에 유니티 3D 캐릭터에 대한 리깅이 요구된다. 유니티에서 하나의 3D 캐릭터를 생성할 때 다수의 오브젝트를 생성한다. 이 오브젝트들은 변형 후 캐릭터의 신체를 이룬다. 이 오브젝트들에 관절을 심어 서로 간의 움직임을 정의해 리깅(Rigging)을 한다. 유니티 플러그인 내 스크립트와 유니티 캐릭터 스크립트 간 각 관절에 대한 맵핑이 필요하다. 사용자와 캐릭터의 관절이 동기화 된 후, 사용자의 움직임과 캐릭터의 움직임이 동기화 된다. 마지막으로

3D월드 좌표계와 2D 컬러좌표계를 캘리브레이션 한다.

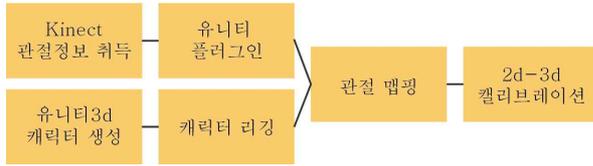
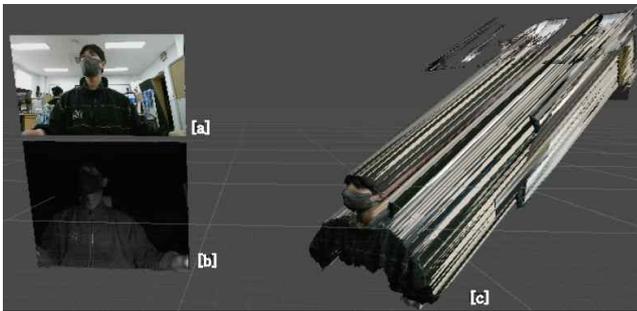


그림5 증강현실 시스템 설계도

### 3.2 유니티 플러그인



7. Unity 내 Kinect V2 연동

마이크로소프트사에서 키넥트 SDK를 기반으로 하는 유니티용 플러그인을 제공한다. 플러그인 내 스크립트는 키넥트의 함수를 유니티 내에서 쓸 수 있도록 Wrapper 역할을 수행한다. 그림은 유니티에 키넥트용 플러그인을 적용한 결과이다. 유니티 월드 내에 스크린 모양의 구조물을 생성하고 그 위에 데이터를 표시했다. 그림 7의 a는 RGB 컬러 이미지, b는 적외선 이미지, c는 깊이정보를 나타낸다. 깊이 데이터는 2D로 표현할 수 없어 원점으로부터의 거리를 등고선으로 표시해 3D로 표현된다.

### 3.3 사용자와 캐릭터 관절 동기화

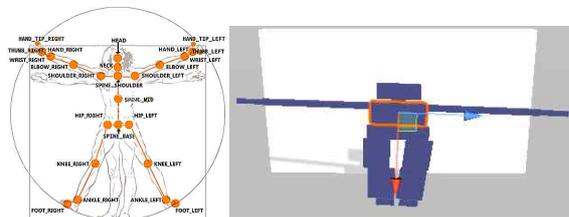


그림8.a Kinect V2 Joint[6]

그림8.b 유니티 캐릭터의 Joint

그림8-1은 키넥트가 얻을 수 있는 인체의 관절이다. 그림 8-2는 유니티 내에서 생성한 3D 캐릭터이다. 3D 캐릭터가 완성되면 신체의 관절정보 대응하는 관절을 설정해 준다. 유니티 내 물리엔진이 속성부여에 편의를 제공한다. 유니티에서 캐릭터에 대한 준비를 완료하면, 키넥트 카메라

라에서 유니티 플러그인을 통해 전달한 사용자의 관절정보를 인식한다. Wrapping 스크립트 내에서 키넥트가 관절정보를 전달하기 위해 설정한 변수와 유니티 캐릭터가 가진 관절에 대한 변수를 관절 위치에 맞게 맵핑 시켜준다. 이 때 관절의 위치 뿐만 아니라 관절의 방향도 고려한다.

### 3.4 컬러카메라와 깊이카메라의 캘리브레이션

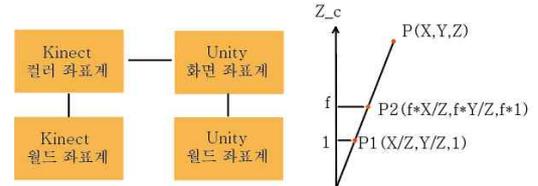


그림9.a 3D-2D변환 다이어그램

그림9.b 삼각비 변환

9는 키넥트에서 유니티로 3D 관절 정보를 전달하기 위한 변환 과정이다. 2.3에서 언급한 캘리브레이션 과정을 따른다. 먼저 키넥트에서 사용자의 관절정보를 얻는다. 이 때 얻는 정보는 키넥트의 월드 좌표계에서 얻어진다. 그림4의 a에서 카메라 좌표계를 카메라의 광학축 방향을 Z, 오른쪽을 X, 아래를 Y라고 설정한다. 이때 점 P는 현실세계의 좌표이다. P1은 정규이미지 위의 한 점이다. 정규이미지는 카메라로부터 거리가 1인 평면이다. P2는 정규이미지에 카메라의 실제 초점거리를 적용한 결과이다. 3D월드 좌표계를 2D좌표계로 변환하는 과정은 다음과 같다.

1. 점P를 정규좌표계로 옮기면  $P1(X/Z, Y/Z, 1)$ 이 된다.
2. 정규좌표계를 초점거리만큼 떨어진 실제 이미지 평면으로 변환하면  $P2(X/Z * f, Y/Z * f, 1 * f)$ 가 된다.
3. 픽셀 좌표계의 중심에 대해 평행이동 한다.

유니티 월드좌표계 내 3D 캐릭터도 월드좌표계에서 스크린좌표계로의 변환을 시행한다. 과정은 키넥트와 같다. 각 캘리브레이션이 완료되면 관절정보를 유니티 스크린 좌표계 위에 표시한다.

### 4 구현 결과



그림 7. 캘리브레이션 구현 결과

그림5는 관절 인식 후 캘리브레이션 결과이다. 키넥트가 신체를 인식하기 위해 평균적으로 2X2 공간은 확보해 주어야 했다. 사용자의 움직임에 따라 유니티 캐릭터 또한 따라서 움직이는 것을 볼 수 있다. 별도의 마커 없이 사용자를 인식 할 수 있었다. 또한 캘리브레이션을 통해 컬러

좌표계와 월드 좌표계의 오차를 계산할 필요 없었으며 공간이 확보 된다면 보다 자유로운 활동이 가능한 몰입도 높은 증강현실 시스템을 설계했다.

## 5. 결론

본 논문에서는 RGB컬러이미지에서 취득한 이미지에 적외선 카메라가 얻은 3D 관절 정보를 맵핑 하기 위해 각 카메라가 얻은 영상 간 캘리브레이션을 진행한 후 서로 다른 좌표계에서의 관절 좌표를 맵핑 시켰다. 스크린 상에서 적외선 센서가 아닌 컬러 이미지를 이용한 바디 트래킹을 함으로써 사용자는 더 직관적인 상호작용이 가능해졌다. 각 관절 간의 관계를 이용하여 심화된 제스처 인식을 통해 다양한 콘텐츠를 산출 할 수 있다. 몸의 움직임이 반영되는 콘텐츠는 재활치료, 운동, 감시 등 여러 목적을 위해 쓰일 수 있다. 또한 이를 인공지능과 결합하면 보안 및 사물인터넷 까지 연관 시킬 수 있다.

구현을 위한 간단한 실험 결과 키넥트 카메라와 적정 거리를 유지해야 한다는 점과 여러 명이 인식되었을 때 겹치는 사람이 생길 경우 인식에 문제가 발생하기도 한다는 점은 앞으로 개선해야할 문제점이다.

## 6. 참고문헌

- [1] 이혜미(Hye-Mi Lee), 류남훈(Nam-Hoon Ryu) , 김응곤(Eung-Kon Kim). (2011). 다수 마커의 제작을 위한 증강현실 기반의 IR LED 마커 검출 기법. 한국전자통신학회 논문지, 6(3): 457-463
- [2] 김율곤, 오수경, 오정아. (2020). Kinect를 이용한 AR 가상 피팅룸 설계 및 구현. 한국정보처리학회 학술대회논문집, 27(2), 296-299.
- [3] 김예나, 김수지, 이지수, 조은선. (2013). Kinect와 Unity3D를 이용한 체감형 3D 가상현실 재활치료 시스템. 대한전자공학회 학술대회, 935-938.
- [4] KINECT for Windows SDKプログラミングKinect for Windows v2センサー対応版 (Japanese) Tankobon Hardcover - May 21, 2015
- [5] Wikipedia, unity, "https://ko.wikipedia.org/wiki/유니티\_(게임\_엔진)"
- [6] Wiedemann, L.G.; Planinc, R.; Nemeč, I.; Kappel, M. Performance Evaluation of Joint Angles obtained by the Kinect. In Proceedings of the IET Conference Proceedings, The Institution of Engineering & Technology, England, 5 November 2015.