

## 매칭 모듈을 이용한 이동 객체와 VR 영상의 동기화

이현섭<sup>1</sup> · 유은재<sup>2</sup> · 김진덕<sup>2\*</sup>

### Synchronization of Moving objects and VR images

Hyoun-Sup Lee<sup>1</sup> · Eun-Jae You<sup>2</sup> · Jin-deog Kim<sup>2\*</sup>

<sup>1</sup>Applied Software Engineering, Dong-eui University, Pusan 47340, Korea

<sup>2</sup>Department of Computer Engineering, Dong-eui University, Pusan 47340, Korea

#### 요 약

최근 Virtual Reality(VR)와 Augmented Reality(AR)가 ICT분야에서 관심분야로 떠오르고 있다. 가상에서 실제와 같은 현실감을 느끼고 그 매력에 많은 사람들이 찾고 있다. 하지만 VR 이미지와 이동 객체의 움직임이 정확하게 들어맞지 않아 멀미를 느끼는 문제점도 발생하고 있다. 이와 같은 문제점을 해소할 수 있는 방안으로 VR 시스템의 어트랙션 적용이 있다. 본 논문에서는 VR 영상과 이동하는 객체의 움직임을 동기화하여 지연시간을 최소화할 수 있는 모듈을 제안한다. 제안하는 모듈은 VR 기기를 통하여 사용자가 바라보는 방향과 가속도 센서를 이용하여 이동 거리를 산출한다. 이 논문에서 제안하는 모듈은 어트랙션의 움직임이 고정된 경로를 따라 움직인다는 점을 주목하여 정확한 이동거리를 계산할 수 있을 것으로 사료된다.

#### ABSTRACT

Recently, VR and AR are emerging as an area of interest in ICT. In virtual reality to feel like a real, many people are looking for them because of its charm. However, the motion of the VR image and the moving object are not synchronized exactly, and a problem of feeling nausea is also occurring. The problem should be solved by the application of Attraction in the VR system. In this paper, we propose a module that minimizes delay time by synchronizing movement of VR image and moving object. The proposed module calculates the moving distance using the direction and the acceleration sensor that the user views through the VR device. The module proposed in this paper will pay attention to the fact that the movement of the attraction moves along a fixed path, so that the accurate travel distance can be calculated.

**키워드** : 이동객체, VR, 영상, 동기화

**Key word** : Moving Object, VR, Images, Synchronization

Received 29 May 2017, Revised 02 June 2017, Accepted 13 June 2017

\* Corresponding Author Jin-deog Kim(E-mail: jdk@deu.ac.kr, Tel:+82-51-890-1745)

Department of Computer Engineering, Dong-eui University, Pusan 47340, Korea

Open Access <https://doi.org/10.6109/jkiice.2017.21.7.1435>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.  
Copyright © The Korea Institute of Information and Communication Engineering.

## I. 서 론

Virtual Reality(VR)와 Augmented Reality(AR)는 현재 ICT분야에서 많은 연구와 개발이 이루어지고 있는 영역이다. 게임, 엔터테인먼트, 의료부문 등 많은 분야에서 360도 동영상, 가상현실 공연 등 다양한 콘텐츠들이 개발되어 지고 있고, 사용자 또한 큰 폭으로 증가하고 있다[1].

2016년에는 세계적인 인기를 끌었던 애니메이션 ‘포켓몬스터’의 캐릭터들을 실생활에서 잡을 수 있는 포켓몬 고(Pokemon GO)가 개발되었다. 이는 AR로 자신이 생활하는 환경에서 게임 속 캐릭터를 볼 수 있다는 현실감을 느낄 수 있어서 인기를 끌었다[2].

VR 또한, VR 게임방이라고 하는 곳이 생겨 인기를 끌고 있는데, VR 기기가 설치된 방에 들어가서 게임을 즐길 수 있다. 이렇듯 VR과 AR은 게임 분야에서 활발하게 개발되고 있다[3].

일반적으로 많은 사용자들은 가상의 콘텐츠에서 현실감을 느낄 수 있다는 것에 높은 관심을 가지고 있으며 이를 위한 여러 가지 VR 기술들을 접하고 있다. 최초 사용자들이 단순 새로운 기술에 대한 환상으로 VR 콘텐츠에 접근하였다면 최근 사용자들은 많은 관련 지식으로 보유하고 새로운 니즈를 충족하기 위한 기술을 끊임없이 요구하고 있다.

가장 큰 사용자의 니즈는 더욱 높은 현실감에 대한 기대 부분이다. 기존의 VR 콘텐츠는 고정되어 있는 공간 속에서 정적인 영상 처리 및 콘텐츠의 사용이 대부분이었으며 이동 콘텐츠의 적용은 정적인 사용자의 환경에서 콘텐츠 자체만 움직임을 가졌기 때문에 현실감에 많이 떨어지는 문제가 발생하였다.

따라서 본 논문에서는 사용자의 현실감을 더 높일 수 있도록 VR 영상과 이동하는 객체의 움직임을 동기화하는 시스템에 대하여 제안한다. 제안하는 기술은 사용자가 탑승한 HMD기반의 이동 객체의 이동 거리에 대응하여 VR콘텐츠의 영상이동을 실행하는 것으로 탑승체의 이동과 VR 콘텐츠의 실행을 동기화 하는데 목적이 가진다.

그리고 본 논문에서 제안하는 모듈은 어트랙션의 움직임이 고정된 경로를 따라 움직인다는 것인데 이를 주목하여 정확한 이동거리를 계산할 수 있도록 한다.

## II. 관련 연구

최근 국내에서 가상현실 시장을 확보하기 위하여 국가 및 대기업들이 적극적인 노력을 하고 있다. 대부분의 콘텐츠들은 VR 콘텐츠로 전환될 것으로 예측된다. 따라서 VR 콘텐츠와 그에 맞는 소프트웨어 플랫폼 개발이 필요하다[4].

VR 기술은 사용자의 감각과 인지능력을 활용하여 실제 존재하지 않은 가상의 인공 환경을 제공하는 것을 목적으로 하며, 이는 실제 환경의 시간적, 공간적 제약을 받지 않고 가상환경에 따라 다양한 서비스를 제공할 수 있다. VR 기기의 지연시간이 유발하는 멀미는 VR 대중화에 큰 걸림돌이 되고 있다. 지연시간을 단축하기 위하여 대부분의 기기들은 유선 연결을 통하여 영상전송을 하고 있으나, 공간상의 제약으로 무선 연결을 사용한다. 그러나 무선 전송의 경우 전송 지연에 의한 떨림 현상으로 멀미를 심화시키게 된다[5].

이를 해결하기 위하여 본 논문에서는 실제 테마파크에서 운용되는 고정 어트랙션의 방향이 결정되어 있는 것을 활용하여 최소화된 연산을 이용하여 지연시간을 최소화 하여 문제를 해결하고자 한다.

데이터베이스를 이용한 이동 객체 관리의 경우 매 시간 변형된 위치정보를 저장하는 것이 불가능하다는 것이다. 그러므로 일정한 비율의 시간 주기를 지정하고 매 주기마다 위치 정보를 이산적으로 저장하여야 하지만 주기 사이의 불확실한 위치 정보에 대하여 적절히 응답할 수 없는 문제가 발생되며 이를 해결 하고자 3차 스플라인 보간법을 이용하여 불확실한 위치 정보에 대한 정량화를 제시하였다[6].

그러나 스플라인 보간법을 이용한 위치 정보의 정량화 기법은 연산 시간이 길다는 장점이 있다. 따라서 본 논문에서는 고정 어트랙션의 방향이 예측 가능한 경로라는 것을 활용하여 최소화된 연산으로 정확한 이동 거리를 판단하고자 한다.

웨어러블 입력장치를 이용한 VR 이미지와 실세계간의 동기화 시스템 모델을 제안하였다[7].

이에 착안하여 사용자가 착용하는 웨어러블이 아닌 어트랙션 장비에 다양한 센서 들을 부착하여 센서를 통해서 전송되는 데이터를 분석하고 제어하여 어트랙션 장비와 사용자의 움직임에 맞추어 실공간과 가상공간을 매칭 시켜주는 시스템을 제안한다.

### III. 매칭 모듈 시스템

#### 3.1. 시스템 전체 구조

이 논문에서 제안하는 매칭 모듈 시스템의 구성도는 아래와 같다.

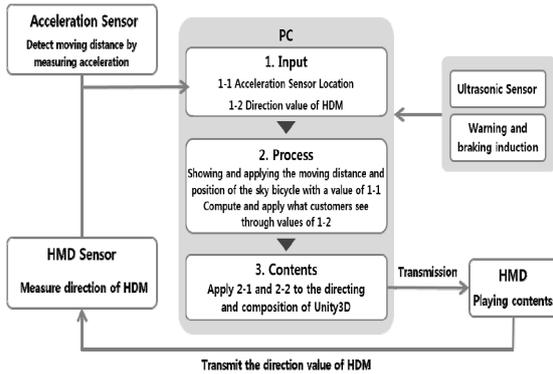


Fig. 1 Matching Module System Diagram

그림 1의 입력부에서 탑승객이 착용한 VR 기기의 HMD(Head Mounted Display)를 통하여 현재위치에서 바라보는 방향 값을 입력받으며, 가속도 센서를 이용하여 이동거리를 입력받도록 한다.

연산부에서는 입력부에서 전달되는 가속도 센서 값을 이용하여 탑승한 어트랙션의 이동거리 및 위치를 표시하고 이를 unity 모듈에서 적용할 수 있도록 하는 연산 알고리즘이 구성된다.

이와 함께 HMD로부터 수신되는 값을 통하여 탑승객이 바라보는 곳을 계산하여 적용하도록 한다. 콘텐츠부는 Unity 엔진으로 제작된 연출 및 구성에 연산부의 값이 적용되어 콘텐츠가 완성되고 HMD로 전송이 되어 플레이가 된다. 제작된 시스템의 전체에는 탑승객의 안전거리를 확보하기 위하여 탑승기계 앞에 초음파센서를 부착하고 센서에 의하여 앞차가 감지되면 경고음과 내레이션으로 제동을 유도한다.

#### 3.2. 가속도 센서를 활용한 이동거리 검출

그림 2는 매칭 모듈의 흐름도를 나타낸 것이다. 이 모듈은 3개의 모듈과 1개의 DB로 구성된다.

Sensor Data input module은 가속도 센서의 정보를 수집하는 모듈이다. Sensor Data Input Module은 가속도 센서로부터 입력받은 Sensing Data를 처리하도록

한다. 이때 Sensor Data Input Module 은 센서의 입력값에 따라 분석 모듈에 데이터를 전달하고 이와 동시에 DB에 값을 저장한다.

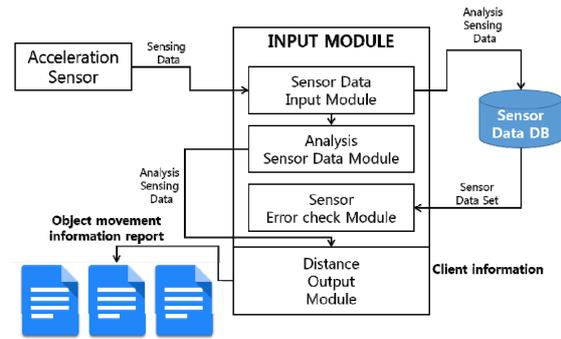


Fig. 2 Matching Module flowchart

Analysis Sensor Data module은 수집 모듈을 통해 입력되어 가공된 센서 데이터를 활용하여 (이중 적분) 현재 이동한 거리를 연산하는 모듈이다. 이때 손실이나 인식 불가와 같은 비정상 데이터는 모듈에서 제거된다.

Sensor Error Check Module은 수집한 센서 데이터와 기존 DB에 저장되어 있는 정상 데이터와 비교하여 이상 유무를 판단하며, 어트랙션 점검시점에 구동되는 모듈이다. Distance Output Module은 처리된 이동거리를 연산부로 제공해 주는 모듈로 입력된 Sensor의 데이터 분석과 에러 체크가 완료된 데이터는 각 VR모듈에 Report되도록 한다.

이 논문에서는 가속도와 자이로 센서를 활용한 현재 위치를 판단하는 알고리즘을 사용한다. 가속도와 거리 관계를 이용하여 현재 위치를 추정하고 가속도를 적분하여 속도를 계산하고 이를 다시 적분하게 되면 거리가 계산되는 것을 이용하여 가속도를 이중 적분하게 된다.

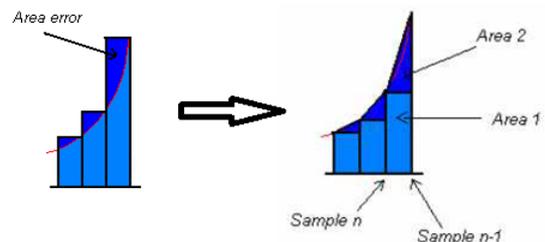


Fig. 3 Trapezoidal Method Routine

특히 이중 적분 과정에서 샘플링의 한계로 인해 발생 되는 값의 오류는 이론상으로 적분할 때 무한히 쪼개면 완벽한 값이 나오지만 양자화 과정 중 Area error가 발생하며 이를 줄이기 위한 방법으로 그림 3과 같은 Trapezoidal Method 방식을 채택하여 오차 줄인다.

sampling의 한계오차를 줄이기 위하여 Trapezoidal Method를 적용할 때 높이를 sample1, Sample2로 가정 하여 수식을 다음과 같이 구성할 수 있다.

$$Area = Area1 + Area2 = Area1 + \frac{(Sample2 - Sample1)}{2} \times T \quad (1)$$

$$Area_n = Sample_n + \frac{|Sample_n - Sample_{n-1}|}{2} \times T$$

수식 1에서 Area는 구하고자 하는 영역이며 이전 Sample과 현재 Sample을 이용하여 적분을 한다. T는 이전 Sample에서 현재 Sample까지의 시간을 나타낸다.

본 논문에서는 Trapezoidal Method를 적용하여 오차 줄인다. 이는 정지 중에 발생하는 중력 가속도를 고려 하며, 물체가 정지하면 가속도는 0으로서 적분 값 또한 0으로 나와야 하지만 등속도 운동과 정지 상태를 파악 하기 위한 알고리즘 적용한다. 또한 이전 상태 정보 들 을 분석하여 정지인지 등속도인지 판단하기 위한 알고 리즘 적용한다.

자이로 센서와 GPS(외부 탑승 기계일 경우)를 통해 현재 위치를 보정하여 VR 동영상의 흐름 및 실행 속도 에 대한 오차를 최소화한다.

가속도 센서와 자이로 센서의 정보를 통해 이동거리를 판단하기 위해서는 위에서 언급한 2중 적분 이외에 도 많은 보정 연산이 필요하다. 특히 관성 센서의 위치 에 따라서 센서의 자세를 보정하는 연산이 필수로 수행 되어야 하며 센서의 기울기에 따른 중력가속도의 적용 보정 회전 정보에 따른 회전 가속도 값의 제거 등 많은 보정이 필요하다. 그러나 본 논문에서 제시한 어트랙션 의 경우 고정형 어트랙션의 경로(레일)를 통해 이동하 기 때문에 센서 데이터의 변동 정보를 보정하기 위한 상당량의 연산이 제거된다.

즉, x 축과 z축의 변화량에 고정 경로 값을 대입하여 이동 거리를 연산하기 때문에 실시간 연산에서의 연산 성능을 보장할 수 있다.

### 3.3. unity 매칭 모듈

그림 4는 이동거리를 검출하는 input module에서 처 리된 값을 unity 영상에 매칭 하는 모듈이다.

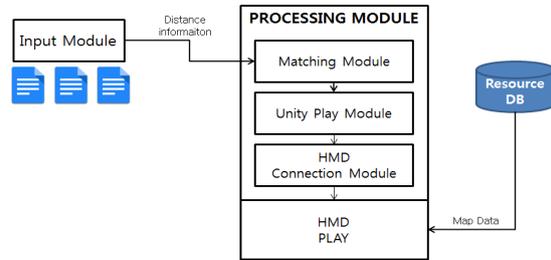


Fig. 4 Move Data Matching Routine

본 모듈에서는 input module에서 전송된 이동거리 값을 변환하여 unity play 모듈의 camera object의 transform 속성에 대응되는 값으로 변환하는 작업을 수 행한다. VR 프로젝트 또한 unity의 영상 처리 엔진을 통 해 나온 결과물이기 때문에 프로젝트 구조는 기본 프로 젝트와 동일하다. 따라서 사람의 시각을 통해 화면을 처리하므로 camera object의 정보를 처리하여 VR환경 을 제어한다.

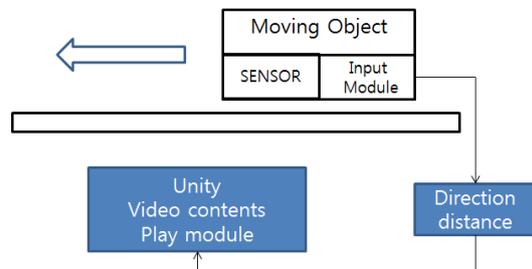


Fig. 5 Move Data Matching Process

일반 VR콘텐츠의 경우 사용자가 정적인 상태에서 camera object의 정보가 변경이 되어 현실과의 괴리감 이 나타나지만 제안된 시스템의 경우 그림 5에서 추출 된 이동 거리와 방향 정보와 camera object의 transform 및 inspector 정보가 정확하게 동기화 되어 이동 거리에 동일한 패턴으로 VR 콘텐츠가 실행된다. 따라서 제안 된 기술을 구현할 경우 현실과의 괴리감을 최소화 하여 사용자에게 높은 만족도를 제공할 수 있는 시스템으로 구성이 가능하다.

### 3.4. Unity contents 실행 모듈

그림 6은 Unity3d (ver 5.2 later)에서 특정 오브젝트의 위치에 따라 camera object의 이동을 수행하는 기본적인 구현 code 이다.

```
public float offsetX = 0f;
public float offsetY = 25f;
public float offsetZ = -35f;

Vector3 cameraPosition;

void LateUpdate ()
{
    cameraPosition.x = player.transform.position.x + offsetX;
    cameraPosition.y = player.transform.position.y + offsetY;
    cameraPosition.z = player.transform.position.z + offsetZ;

    transform.position = cameraPosition;
}
```

Fig. 6 Move Unity Camera object control code

Unity의 경우 camera object를 통해 콘텐츠의 화면을 사용자에게 보여 준다. VR의 콘텐츠 이동의 경우에도 화면상 보이는 것은 camera object의 값의 변경을 통해서이고 실제로 VR 전용 API가 존재하는 것이 아닌 camera object를 통해 보이는 왼쪽 눈이 보는 영상과 오른쪽 눈이 보는 영상을 서로 결합하여 하나의 VR 영상으로 컴파일 하게 된다. 이는 Unity 5.2 버전 이후의 엔진에서 지원하며 본 논문에서 제안한 3.3 절에서의 매칭 모듈은 본 실행 모듈과의 매칭을 수행하게 된다.

LateUpdate() 함수의 경우 Unity의 프레임별 수행되는 모든 Update() 함수의 동작 이후 마지막에 실행되는 함수로 이 함수 내에 있는 플레이어(사용자 탑승체)의 이동 정보를 앞서 언급한 여러 모듈을 통해 연산하고 그 결과를 플레이어 위치 정보 값으로 처리하여 camera object의 방향, 이동정보 등을 갱신한다. 이 과정을 통해 사용자는 자신이 움직이고 있는 어트랙션의 이동정보에 따라 Unity의 콘텐츠가 실행되는 것을 HMD를 통해 경험할 수 있게 된다.

### 3.5. Unity sensor 통신 모듈

그림 7은 Unity와 외부 Arduino 보드의 Sensing Data 정보를 시리얼 통신을 통해 주고받는 C# 구현부이다.

실험을 위해 본 논문에서는 아두이노 우노(Arduino Uno) R3 ATmega328P의 AVR 8 bit 마이크로 컨트롤러를 사용하여 시리얼 통신을 사용하였다. 통신을 하기 위해서는 Unity와 Arduino의 코드에서 통신 포트를 같게 맞추어야 한다.

```
using UnityEngine;
using System.Collections;
using System.IO.Ports;

public class asdasdasd : MonoBehaviour {
    SerialPort sp = new SerialPort("COM3", 9600); //set Serial port
    int bikeSpeed = 0;

    void Start () {
        sp.Open(); //Serial port open
        sp.ReadTimeout = 1; //set Serial timeout
    }
    // Update is called once per frame
    void Update () {
        if (sp.IsOpen)
        {
            try
            {
                sp.Write("s"); //send start data
                bikeSpeed = sp.ReadByte();

                Debug.Log(bikeSpeed);

                transform.Translate(Vector3.forward * Time.deltaTime * bikeSpeed);
            }
            catch (System.Exception) {}
        }
    }
}
```

Fig. 7 Unity-Arduino Serial communication code

Update() 함수에서 시리얼포트가 열려있는 지 확인하고 Arduino 보드에서 현재 속도를 읽어 Object를 앞으로 이동시키도록 하였다. Update() 함수는 프레임 별로 계속 호출되며 시리얼 포트가 1초 이상 응답이 없는 경우 시리얼 포트는 자동적으로 닫히게 된다.

## IV. 결론

가상현실은 현재 많은 연구와 개발로 우리에게 점점 다가오고 있다. 하지만 VR 기기가 보여주는 이미지의 지연시간으로 인해 멀미를 유발하고, 환경의 제약으로 발생하는 떨림 현상은 이를 심화시킨다. 이 논문에서는 이동 객체의 움직임과 가상현실 영상을 동기화 하는 매칭 모듈을 제안하였다. 가속도를 적분하여 속도를 산출하고 이를 다시 2차 적분하여 거리를 구한다. 계산된 거리에는 오차가 존재하므로 Trapezoidal Method 방식으로 이를 줄인다. 이 방법은 본 논문에서 제시하는 어트랙션의 움직임이 고정된 경로를 따라 움직이므로 그렇지 않은 이동 객체에 비해 최소화된 연산으로 더욱 정확한 이동거리를 산출해 낼 수 있다.

향후 연구 과제로는 3.5절에서 제시한 아두이노와 여러 가지 관성 센서를 활용하여 실제 이동거리에 대한

최적화 센서와 알고리즘을 적용하는 것이다. 고정 트랙을 이동하는 어트랙션이라고 하지만 정밀도가 낮은 MEMS 센서의 경우 회전 및 속도 보정에 많은 연산이 소모되며 9축 이상의 높은 정밀도의 센서는 수집되는 많은 데이터의 양을 처리하기 위한 알고리즘의 구현이 필요하다. 따라서 구현 결과를 통한 센서 데이터 처리와 Unity 사이의 동기화 모듈 구현이 매우 중요한 향후 과제이다.

본 논문에서 제시한 설계 기법 및 데이터 동기화 방안을 적용하면 사용자들에게 현실감이 떨어지는 기존의 VR 콘텐츠들에 비해 높은 만족감을 줄 수 있을 것으로 판단되며 VR 이미지와 이동 객체의 움직임간의 동기화 매칭 기술을 통해 VR의 가장 큰 문제점인 현실과의 괴리감 문제를 해결할 수 있을 것으로 사료된다.

### ACKNOWLEDGMENTS

This work was supported by Dong-eui University Grant(201702950001).

### REFERENCES

- [1] M. S. Kim, J. H. Kang, M. S. Jeon, "Market and Technical Trends of VR Technologies," *Korea Contents Association Review*, vol. 14, no. 4, pp.14-16, Dec. 2016.
- [2] H. S. Kim, T. J. Park, "Technology Trends of Virtual Reality (VR) and Augmented Reality (AR) and Implementation Examples in Game Engine," *Information And Communications Magazine*, vol. 33, no. 12, pp.56-62, Dec. 2016.
- [3] K. J. Kim, E. J. Song, "Joint of Game Creation and Virtual Augmented Reality Contents Using Unity Engine," *The Magazine of Kiice*, vol 16, no 1, pp.54-61, Jun. 2016.
- [4] Y. M. Lim, "Virtual Reality Content and Technology," *Information And Communications Magazine*, vol. 33, no. 12, pp.49-55, Nov. 2016.
- [5] W. B. Yoon, J. H. Han, "Relation between jittering from compensation for latency and VR sickness," *Journal of the Korea Computer Graphics Society*, vol. 23, no. 1, pp.1-8, March 2017.
- [6] Y. A. Ahn, K.H. Ryu, "Management of Uncertain Position Information of Moving Objects," *Journal of the Research Institute for Computer and Information Communication*, vol. 9, no. 1, pp.57-74, May 2001.
- [7] K. H. Park, H. J. Lee, Y. H. Kim, "A Study on the VR Synchronization System using Wearable Input Device," *Proceedings of KIIT*, pp.807-810, June 2009.



이현섭(Hyoun-Sup Lee)

2009년 2월 : 동의대학교 컴퓨터공학과 박사 수료  
2014년 4월 ~ 현재 : 동의대학교 멀티미디어 공학과 조교수  
※관심분야 : IPS, 데이터베이스, 스마트 애플리케이션, 모바일 웹



유은재(Eun-Jae You)

2016년 2월 : 동의대학교 컴퓨터공학과 졸업  
2017년 3월 ~ 현재 : 동의대학교 컴퓨터공학과 석사과정  
※관심분야 : 데이터베이스, 스마트 애플리케이션



김진덕(Jin-deog Kim)

2000년 8월 : 부산대학교 컴퓨터공학과 졸업(공학박사)  
2001년 ~ 현재 : 동의대학교 컴퓨터공학과 교수  
※관심분야 : 데이터베이스, GIS, 모바일 시스템, LBS, 빅데이터