

스마트폰을 이용한 HMD 사용자의 가상환경 상호작용 몰입 개선 방법 연구



지도교수 : 이명호

팀명: 땡호와

전기컴퓨터공학부 정보컴퓨터공학전공

201924401 강경찬

201924472 박예린

201624456 김진서

목 차

1. 과제 배경	1
1.1 과제 배경	
2. 기존 사항 분석	4
2.1 기존에 연구된 것	
2.2 작년 졸업과제의 한계점	
3. 요구 조건 분석	4
3.1 요구 조건	
4. 현실적 제약사항 및 대책	4
4.1 현실적 제약사항	
4.2 대책	
5. 연구 방향	4
5.1 기술 소개	
5.2 개발 환경	
5.3 시스템 구성도	
5.4 주요 기능	
6. 개발 일정 및 역할분담	4
6.1 개발 일정	
6.2 역할분담	
7. 참고 자료	4
7.1 참고 자료	

1. 과제 배경

1.1 과제 배경

최근 메타버스라는 단어를 필두로 VR/AR 에 대한 관심이 급부상하고 있습니다. 이는 가상공간에서의 자신의 아바타를 통해 다른 아바타와 소통하고 활동하는 것인데, 여러가지 활동을 하려면 현실의 “나”와 가상공간의 “아바타”가 서로 상호작용을 하여야 합니다.

현재 VR/AR 에서의 상호작용은 주로 VR Controller 혹은 Hand Tracking 으로 이루어지고 있습니다. VR Controller 는 정확하고 빠른 Tracking 기능을 제공하지만, 사용자가 할 수 있는 상호작용에 제한이 존재합니다. 그리고 Hand Tracking 은 별도의 Controller 가 필요 없다는 장점과 양손이 자유롭다는 장점이 존재하지만, 아직 정확도가 부족하며 할 수 있는 상호작용이 VR Controller 보다 제한적이라는 단점이 존재합니다. 이러한 문제를 해결하기 위해서 우리는 VR/AR 에서 스마트폰을 이용한 상호작용 방법을 제안합니다.

우리는 스마트폰을 이용할 경우, 사용자는 별도의 VR Controller 를 가지고 있을 필요도 없으며, 한 손이 자유롭다는 장점과 함께, 스마트폰의 다양한 센서 및 터치스크린을 이용해서 Hand Tracking 에서의 제한적인 상호작용 문제를 해결할 수 있을 것이라고 기대합니다.

이와 관련된 연구로 작년 이명호 교수님 연구실에서 진행된 스마트폰을 이용한 몰입형 가상현실 상호작용 기법 개발이라는 연구가 존재합니다. 해당 과제에서는 스마트폰을 이용해서 VR 에서의 사용 가능한 다양한 기능을 구현하였으나, 스마트폰의 위치정보를 대략적으로 얻을 수밖에 없다는 문제가 있었기에 사용성에 있어서 많은 제약이 존재하였습니다. 따라서 본 과제에서는 스마트폰의 위치 추적의 문제를 해결하기 위하여 스마트폰 카메라의 위치를 Tracking 하고, 여러가지 VR 안에서의 상호작용 기법들을 연구합니다.

2. 기존 사항 분석

2.1 기존에 연구된 것

현실 사람의 표정을 가상공간의 캐릭터로 표현하는 기술은 현재 페이스 Tracking으로 이미 연구된 바 있습니다. 하지만 스마트폰 카메라를 이용한, HMD를 착용한 상태의 페이스 Tracking 연구 결과는 많지 않습니다. 이는 눈을 포함한 얼굴의 윗부분은 가려진다는 점과 더불어 스마트폰 카메라를 활용한다는 점에서 기존의 페이스 Tracking 기술과 차이가 있습니다. 따라서 기존의 페이스 Tracking과 달리 HMD 기기를 착용한 상태에서의 얼굴 위치 Tracking과, 가려지지 않은 부분인 입술의 Motion을 Tracking해볼 것입니다.

2.2 작년 졸업과제의 한계점

작년에 이명호 교수님 연구실에서 진행하였던 저희의 선행 연구 격인 '스마트폰을 이용한 몰입형 가상현실 상호작용 기법 개발'이라는 연구에서 스마트폰을 이용한 상호작용을 이용해 다양한 기능 구현은 성공하였으나 스마트폰의 위치가 고정되어 있어야 하였고, 이에 스마트폰의 위치가 바뀌게 되면 VR 안에서 UI가 함께 움직이지 않으므로 많은 제약이 존재하였습니다. 따라서 본 과제에서 스마트폰 카메라의 위치를 Tracking하여 VR 내의 UI가 함께 움직일 수 있도록 할 것입니다.

3. 요구 조건 분석

3.1 통신

- 스마트폰 카메라를 통해 찍은 사진을 Deep Learning PC 에 전달하기 위한 TCP 통신 가능케 하기
 - 전년도에 선행된, 본 졸업과제와 유사한 졸업과제에서 사용한 TCP 통신 코드 수정 및 활용
- Python Deep Learning Model 들(얼굴 위치 인식, Lip 모션 인식)에서 도출된 결과 정보를 Unity VR Project 로 전달

3.2 데이터 수집

- Quest2 착용 영상을 직접 찍어 수집
- 졸업과제 팀원들이 직접 Quest2 를 착용하고, 스마트폰 전면 카메라로 동영상을 찍어 30fps(초당 30 프레임)의 영상 얻기
- 이러한 영상들을 0.05~0.1 초 단위로 끊어 카메라 이미지 데이터 얻기
- 이를 OpenCV 혹은 다른 이미지 처리 프로그램을 통해, 모델을 학습시키기에 적합한 데이터 형태로 가공

3.3 카메라 접근

- Unity 로 스마트폰 APP 제작
- Realtime 으로 영상을 보내거나 0.05~0.1 초에 전면 카메라로 찍은 Quest2 HMD 기기를 착용한 사진 보내기
- 추후 추가적인 AR 스마트폰 상호작용 기능 구현을 위해 전/후방 카메라 동시 접근 가능한지 확인해보기

3.4 딥러닝/머신 러닝

- Quest2 착용 영상에서 얻은 이미지를 통해 Quest2 를 쓰고 있는 Face 위치 Tracking 하기
 - Tracking 방식은 HMD 기기 Tracking, HMD 기기를 착용한 얼굴 Tracking, HMD 기기를 착용한 얼굴의 입술 Tracking 으로 총 세가지 방식 사용
- Quest2 착용 영상에서 얻은 이미지를 통해 Quest2 를 쓰고 있는 사람의 Lip motion Tracking 하기

3.5 Unity 구현

- HMD 기기 위치 Tracking 정보를 바탕으로 계산하여 가상 환경에 스마트폰 구현하기
- 립 모션 Tracking 정보를 바탕으로 가상 환경에 립 모션 구현하기

3.6 스마트폰 AR 상호작용 기법 연구

- 위 3.5 에서 구현한 것을 바탕으로 아래 중에서 선택하여 추가적으로 AR 상호작용 기법 연구
 - 스마트폰 터치패드를 이용한 키보드 구현 (작년 구현된 기능의 개선)
 - 스마트폰 터치패드를 이용한 사용자 Handed-UI 구현 (VR 시나리오에 따라서 기능 선정 필요)
 - 스마트폰 카메라를 이용하여 사용자의 Lip Motion Tracking 이 가능한 캐릭터를 이용하여 사실감이 느껴지는 캐릭터 생성

4. 현실적 제약사항 및 대책

4.1 현실적 제약사항

1. 모델이 매 0.05 초~0.1 초마다 추론을 수행해야 하는데 이 추론과 Unity 가상환경을 한 PC 에서 진행하는 데 문제는 없을지
2. Realtime 카메라 이미지를 파이썬 PC 로 딜레이 없이 보낼 수 있을지
3. 안드로이드 버전에 따른 호환성 문제
4. 스마트폰 기울어짐에 따른 각도 차이
5. 스마트폰 사이즈 차이에 따른 구현 문제

4.2 대책

1. 모델 추론과 가상 환경 병행 문제 대책: 프로젝트를 진행하며 모델의 크기나 컴퓨터 자원 등을 고려하여 개발함으로써 해결
2. Realtime 카메라 이미지 전달 문제 대책: 프로젝트를 진행하며 많은 시간을 해당 부분의 성능 향상에 쏟음으로써 해결
3. 안드로이드 호환성 문제 대책: 최대한 많은 안드로이드 스마트폰에서 동작 가능한 안드로이드 버전으로 선택하여 개발을 진행
4. 스마트폰 기울어짐에 따른 각도 차이 대책: 스마트폰 기울기 센서를 통해 스마트폰이 얼마나 기울어졌는지에 대한 정보를 반영하여 보완
5. 스마트폰 사이즈 차이에 따른 구현 문제 대책: 스마트폰 사이즈 입력을 통해 조정이 가능하게끔 구현

5. 연구 방향

5.1 기술 소개

1. Oculus Integration

Unity 개발 환경에서 oculus 앱 개발에 도움을 준다.

Advanced rendering, social, platform, audio 그리고 Avatars development support for Oculus VR 등 여러가지 중요한 SDK 패키지를 제공한다.

2. TensorFlow

google 에서 딥러닝 프로그램을 쉽게 구현할 수 있도록 도움을 주는 라이브러리이다.

필기 숫자 판별, 이미지 인식, 단어 Embedding, 반복 신경망 등이 가능하다.

tensor board 를 활용하여 딥러닝 학습 과정을 추적하여 딥러닝 시각화에 도움이 된다.

3. Open CV

OpenCV(Open Source Computer Vision)는 영상 처리에 사용할 수 있는 오픈 소스 라이브러리이다. 컴퓨터가 사람의 눈처럼 인식을 할 수 있도록 도와준다. 우리가 사용하는 많은 카메라 App 들이 Open CV 를 활용하여 개발된다.

4. Unity

유니티는 3D 및 2D 게임 엔진이자 3D 애니메이션, 가상 현실 등 콘텐츠 제작을 위한 통합 개발 도구이다.

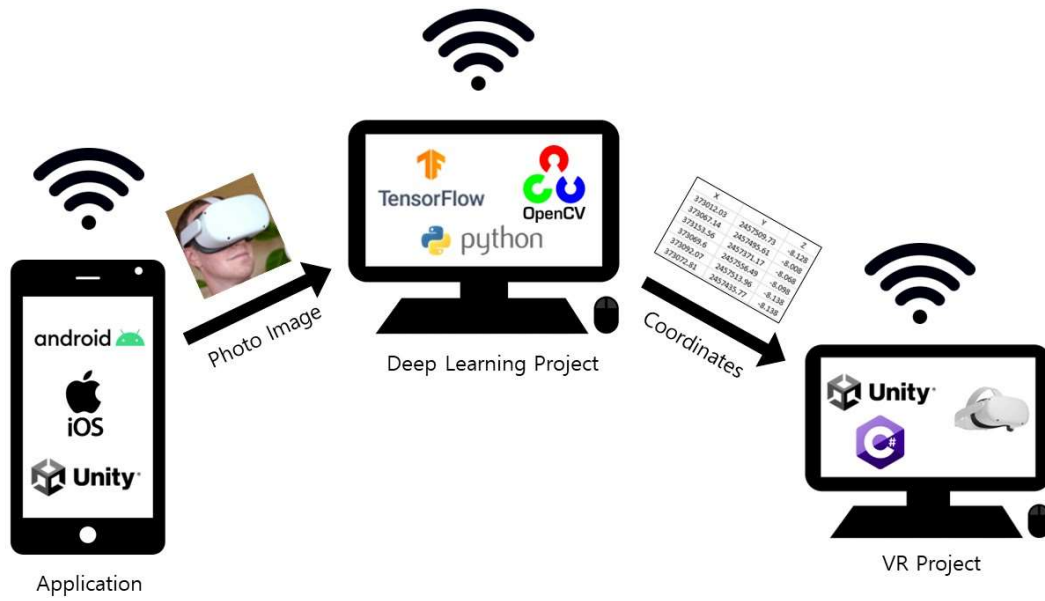
5.2 개발 환경

1. 개발언어: C#, python

2. 개발도구: Unity Studio, Jupyter Note

3. 개발장비: Oculus Quest 2, 스마트폰

5.3 시스템 구성도



6. 개발 일정 및 역할분담

6.1 개발 일정

업무/일정	5 월				6 월				7 월				8 월				9 월			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
착수보고서																				
카메라 접근																				
통신 연결																				
HMD 위치 tracking																				
Lip motion tracking																				
스마트폰 AR 상호작용 기법 추가 연구																				
테스트 및 디버깅																				
최종보고서 및 발표 준비																				

6.2 역할 분담

이름	역할
박예린	HMD 착용자 Lip motion Tracking
강경찬	HMD VR device 용 Unity 프로그램 개발
김진서	HMD 착용자 얼굴 위치 Tracking
공통	카메라 접근, 통신 연결, 스마트폰 AR 상호작용 기법 추가 연구

7. 참고 자료

7.1 참고 자료

1. 본 과제 초안 notion: <https://sprinkle-weather-c5c.notion.site/0190d4ecc29e4d30b5ddcb678d2d5c25>
2. 작년 졸업과제 notion: <https://nosy-spike-9c8.notion.site/2021-afb85d2b405e428ea56b86c3220e2a35>