

AR 기술을 활용한 소셜 네트워킹 서비스 개발 중간 보고서



메아리(meARy)

201924447 김진영

201924557 임석윤

201924613 허취원

1 목차

2	요구조건 및 제약사항 분석에 대한 수정사항.....	3
2.1	기존의 요구 조건.....	3
2.2	요구조건에 대한 수정 사항.....	4
2.3	수정된 요구조건에 따른 제약사항 분석.....	5
3	설계 상세화 및 변경 내역.....	5
3.1	전체 설계도.....	5
3.2	설계 상세화.....	6
3.3	설계 변경 내역.....	7
4	갱신된 과제 추진 계획.....	8
5	구성원별 진척도.....	8
6	보고 시점까지의 과제 수행 내용 및 중간 결과.....	9
6.1	AR 앱.....	9
6.2	3D 모델링.....	10

2 요구조건 및 제약사항 분석에 대한 수정사항

2.1 기존의 요구 조건

- I. 플랫폼: iOS, Android 양쪽 플랫폼 모두 지원
- II. 사용자 기능
 - ① 영상 업로드 및 3D 모델 생성 요청
 - T자 자세로 서 있는 사용자를 기준으로 촬영자가 원을 그리며 360도 회전하여 사용자의 모습 영상 촬영
 - 360도 회전 영상을 기반으로 3D 모델링 요청
 - ② 포스트 생성 (위치 정보 포함)
 - 사용자의 모습이 담긴 사진 촬영
 - 사진과 촬영 장소 위치를 기반으로 AR 포스트 생성 요청
 - ③ 다른 사용자의 AR 포스트 보기
 - 사용자의 장소, 위치 정보를 기반으로 다른 사용자의 AR 포스팅 열람
- III. 서버 기능
 - ① 360도 영상 처리 및 3D 모델링
 - 사용자로부터 수신한 사용자의 360도 회전 영상을 50개의 사진 Frame으로 나누어 3D Zephyr(또는 3DF)를 사용하여 3D 모델 생성
 - Mixamo Auto Rigging(또는 Blender Rigify)을 사용하여 3D 모델에 관절 정보 매핑.
 - ② 사진을 기반으로 3D 모델로 AR 포스팅 생성
 - 사용자로부터 수신한 사용자의 포스팅 사진에서 MediaPipe를 사용하여 관절 정보 추출
 - 3D 모델에 관절 정보를 적용
 - 사용자의 위치를 기반으로 GeoSpatial Anchor를 생성하여 모델을 Anchor에 올려서 고정
 - ③ 사용자 정보, 포스트, 모델 저장
 - 사용자의 계정 정보, 포스팅 위치, 3D 모델의 자세를 서버에 저장

IV. AR 기능

① Unity 기반 앱 개발 (AR Foundation 사용)

- 자세가 적용된 3D 모델과 Anchor 정보를 서버로부터 받아와서 이를 AR로 표현

2.2 요구조건에 대한 수정 사항

I. 플랫폼

- 환경적 제약 (iOS 개발용 Mac 및 테스트 디바이스 부족 등)
- 제한적인 프로젝트 기한과 인원
- **수정 사항** : Android로 플랫폼을 축소

II. 사용자 기능

- 영상 업로드를 통해 3D 모델을 생성하는 것은 아래의 '서버 기능'에서 설명하듯이 많은 제약점과 한계를 가짐
- **수정 사항** : 사용자의 3D 모델링에 사용자의 모습이 촬영된 360도 회전 영상을 사용하는 것이 아닌, 포스팅으로 사용되는 사진으로 SMPL 3D 모델 생성 요청

III. 서버 기능

- T자 자세로 서 있는 사용자를 기준으로 촬영자가 360도 회전 영상을 기반으로 3D 모델을 생성하는 것에 있어서, 사용자의 촬영 환경과 촬영 장비의 기술 사양, 촬영자의 회전 속도와 능력이 3D 모델 생성에 많은 영향을 미침
- 이는 Rigging 기능이 정상적으로 동작하지 않을 수 있고, 3D 모델의 퀄리티가 매우 낮아질 수 있으며 결과적으로 서비스 자체가 제대로 작동하지 않을 수 있음
- **수정 사항** : 2D 사진을 토대로 3D 메쉬 모델과 자세 정보를 가지고 있는 SMPL 3D 메쉬 모델을 생성해주는 PARE 모델 사용으로 연구 방향 변경

2.3 수정된 요구조건에 따른 제약사항 분석

I. 입력 사진 품질

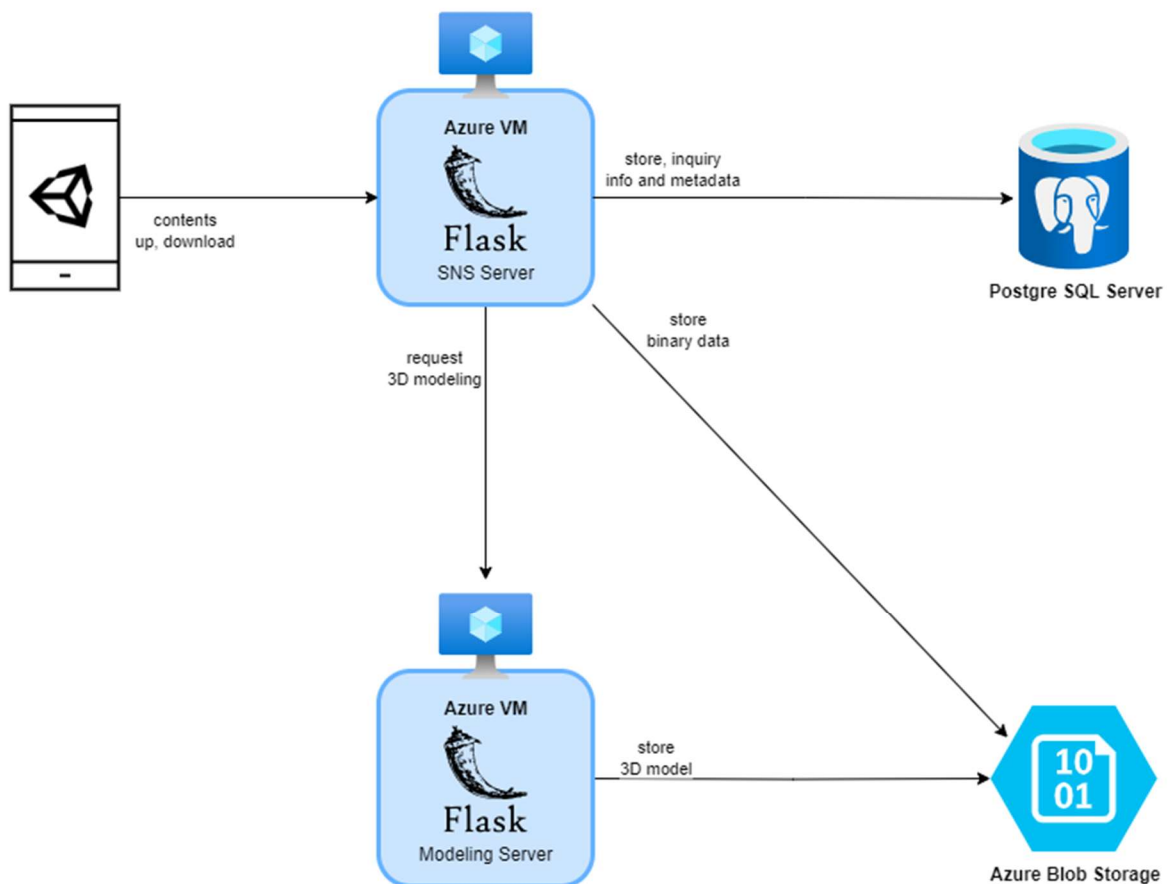
- PARE를 사용하여 SMPL 3D 모델을 생성하는데 있어서, 전신이 잘 보이는 이미지가 필요. 또한 사람이 한 명만 있는 사진에서 가장 잘 동작 하므로 사용자는 이러한 제약사항에 따라주어야 함

II. 3D 모델링을 위한 고성능 GPU 필요

- PARE 모델 사용에 있어서 추론은 CPU만으로 가능하긴 하지만 속도가 매우 느리고, 대량 처리를 위해서는 GPU 사용이 불가피함

3 설계 상세화 및 변경 내역

3.1 전체 설계도



3.2 설계 상세화

I. 사용자

- 사용자는 포스팅 사진을 AR SNS Server로 전송함으로써 AR 포스팅 생성 요청
- 사용자의 위치를 기반으로 다른 사람의 AR 포스팅을 SNS Server에게 요청
- 사용자의 기기에서 SNS Server를 통해 받은 모델과 Google GeoSpatial Anchor 정보를 토대로 다른 사람의 포스팅 AR로 표현
- 사용자는 AR Foundation (Google ARCore 기능을 사용)으로 AR 앱을 구현

II. SNS Server

- 사용자로부터 포스팅 사진과 Google GeoSpatial Pose 정보, 사용자의 위치 정보를 수신
- 포스팅 사진을 Storage에 저장
- Modeling Server에 모델링을 요청함으로써 Modeling Server가 Storage의 사진을 조회하여 모델링 수행 후 Storage에 저장하도록 함
- GeoSpatial Pose 정보와 사용자 Meta Data를 SQL Server에 저장
- 사용자의 위치 정보를 토대로 SQL Server에서 근처 포스팅의 Meta Data를 받아오고 이를 바탕으로 Storage에서 3D 모델을 조회하여 사용자에게 반환

III. Modeling Server

- SNS Server로부터 모델링 요청을 받으면, Storage에서 해당 사진을 조회하여 모델링 후 SMPL 3D Model을 Storage에 저장

IV. Postgre SQL Server

- 포스팅 Meta Data (사용자 정보, GeoSpatial Pose 등) 을 저장하고 있음

V. Azure Blob Storage

- 포스팅 사진과 SMPL 3D 모델을 저장하고 있음

3.3 설계 변경 내역

I. Modeling Server

- 자문 기업의 자문을 토대로 구현의 편의성과 각 서버간 상호 호환성을 위해 FastAPI기반의 서버에서 Flask기반의 서버로 변경
- Modeling 기법을 360도 회전 영상 기반의 3D 모델링, Auto Rigging, 포스팅 사진의 Pose Detection을 통한 3D 모델 구현 방식에서 2D 사진 기반으로 SMPL 3D 모델 생성 방식으로 변경

II. 사용자

- 자문 기업의 자문과 프로젝트 시행착오를 토대로 360도 회전 영상을 SNS Server에 송신함으로써 3D 모델링을 요청하는 방식에서 포스팅하는 2D 사진을 SNS서버로 송신하여 3D 모델링과 AR 포스팅을 한 번에 요청하는 방식으로 변경

4 갱신된 과제 추진 계획

개발구분	세부항목	7	8	9	10
핵심 기능 개발	SMPL 3D 모델링				
	3D 모델 생성 파이프라인 개발				
Azure DB, 스토리지 구축	DB, 스토리지 설계 및 구축				
Unity 개발	Unity 앱 개발				
서버 개발	Flask 서버 개발				
	서버 앱 연동				
테스트 및 배포	보완사항 수정 및 배포				

5 구성원별 진척도

구성원	담당 파트	진척도
김진영	- Unity AR Foundation, google ARCore Extensions 를 활용한 AR 기능 및 안드로이드 앱 개발 - 3D 모델 생성 및 파이프라인 개발	40%
임석윤	- Unity AR Foundation, google ARCore Extensions 를 활용한 AR 기능 및 안드로이드 앱 개발 - 백엔드 개발	50%
허취원	- PARE을 이용한 SMPL 3D 모델링 - 3D 모델 생성 및 파이프라인 개발	40%

6 보고 시점까지의 과제 수행 내용 및 중간 결과

6.1 AR 앱

I. 평면 인식 후 오브젝트 비치

- Unity AR Foundation을 활용해 평면을 인식하고 터치한 위치가 평면이면 오브젝트를 비치하는 기능을 구현했다.
- 추후 서버에 저장되어 있는 좌표를 불러와 평면(바닥)을 인식하고 오브젝트를 비치하는 기능으로 확장 예정이다.

II. Geospatial API 활용

- 카메라 화면에서 터치한 위치를 Geospatial API를 활용해 지구 위도, 경도, 고도 변환에 성공했다.
- 앱 내에서 위도, 경도, 고도를 입력하여 Anchor를 생성해 정육면체 오브젝트를 부착함으로 시각화 및 API 성능 테스트를 진행했다.
- 실외에서 테스트한 경우 터치한 위치와 변환된 지구 좌표의 오차가 평균적으로 3m 가량 발생하고 고도 오차로 인해 앵커가 평면(바닥)에 부착되지 않고 공중에 뜨는 상황 발생했다.
- 오차 줄이기는 한계가 있다.
- 평면 부착은 위도, 경도는 고정하고 고도를 클라이언트 단에서 수정해가며 평면을 인식한 고도에 앵커를 세우는 방안으로 해결 가능성 보인다.

6.2 3D 모델링

- I. 사용자 360도 영상 기반 3D Mesh 모델 생성
 - 사용자를 기준으로 촬영자가 360도 원을 그리며 회전을 한 동영상에서 50개의 사진 Frame을 만들어냈다.
 - 3DF, 3D Zephyr, RealityScan 이렇게 세개의 모델에 Frame들을 넣고 모델링을 시도했다.



(그림1)



(그림2)

- 그림1은 3D Zephyr로 생성된 3D Mesh 모델이고, 그림2는 RealityScan으로 생성된 모델이다.
- 3DF는 성능이 다른 모델보다 좋지 않았기에 자료를 첨부하지 않았다.
- 두 모델 전부 Skin이 조금 왜곡되고 표현력이 좋지 않다. 이는 촬영 영상의 퀄리티 때문이라고 판단된다.

II. 3D Mesh 모델 Rigging



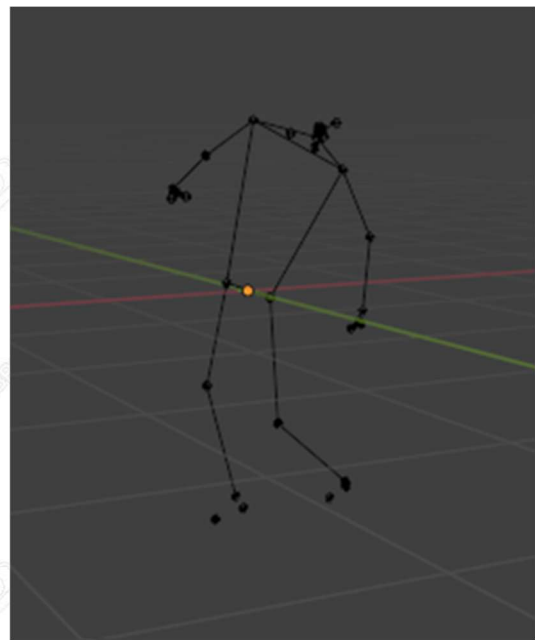
(그림3)

- 그림3은 3D Zephyr로 만들어진 3D Mesh 모델을 Mixamo Auto Rigging으로 관절 정보를 매핑하였다.

III. 사진 Pose Detection



(그림 4)



(그림 5)

- 그림 4 를 MediaPipe 로 PoseDetection 하여 그림 5 와 같은 시각화 정보를 얻었다.
- 그림 5 에서의 관절 위치가 그림 4 의 원본사진과는 조금씩 차이가 나는 것을 눈으로 확인할 수 있다.

IV. SMPL 3D 모델 생성

- 3D 모델링을 Rigging하여 PoseDetection결과를 적용하는 방식에서 2D 사진 기반으로 SMPL 3D모델을 생성하는 것으로 구형 방향을 바꾸었다.
- 현재까지는 PARE 기술을 사용할 계획으로 연구 중이다.