

AR 기술을 활용한 소셜 네트워킹 서비스 개발 최종보고서



분과 : A

지도교수 이명호

메아리(meARy)

201924447 김진영

201924557 임석윤

201924613 허취원

지도교수 이명호

목 차

1. 서론.....	1
1.1. 연구 배경.....	1
1.2. 기존 서비스의 한계.....	1
1.3. 연구 목표.....	1
1.4. 개선 방안 및 구현 전략.....	2
1.4.1. 3D Model 생성 및 제어.....	2
1.4.2. AR 어플리케이션.....	2
1.5. 구성원별 역할 및 개발 일정.....	3
1.5.1. 구성원별 역할.....	3
1.5.2. 개발 일정.....	3
2. 배경 기술.....	4
2.1. 3D Model 생성 및 제어.....	4
2.1.1. Pose Detection (MediaPipe Blaze Pose).....	4
2.1.2. 3D Model 생성.....	4
2.1.3. Rigging.....	4
2.2. AR 어플리케이션.....	4
2.2.1. 평면 위치 추정 (AR Raycast).....	4
2.2.2. 지구 좌표 추정 (Google ARCore Geospatial API).....	5
2.2.3. 3D Model을 AR world에 표현 (AR Anchor).....	5
3. 연구 내용.....	6
3.1. System Architecture.....	6

3.1.1.	Work Flow.....	6
3.2.	Pose Retargeting	7
3.2.1.	Pose Landmark 추출 / Blender Scene 구성	7
3.2.2.	자동 스케일링 및 포즈 정렬	8
3.2.3.	GLB 파일 변환.....	9
3.3.	Unity ARFoundation / Google ARCore.....	9
3.3.1.	Human Detection	9
3.3.2.	AR world에 포스팅 3D Model 생성.....	10
4.	연구 결과 분석 및 평가	10
4.1.	결과물	10
4.2.	분석	11
4.2.1.	3D Model 및 Pose Retargeting.....	11
4.2.2.	AR 어플리케이션 및 지구 좌표 정확도	11
4.3.	평가	12
4.3.1.	연구 목표 달성도	12
4.3.2.	강점 및 혁신성.....	12
4.3.3.	한계.....	12
5.	결론 및 향후 연구 방향	13
5.1.1.	결론.....	13
5.1.2.	향후 연구 방향.....	13
6.	참고 문헌.....	14

1. 서론

1.1. 연구 배경

- 21세기 정보통신 기술의 발전으로 스마트폰은 현대인의 필수적인 기기로 자리잡았다. 스마트폰의 보편화는 모바일 인터넷 접속을 일상화시켰고, 이는 소셜 네트워킹 서비스(Social Networking Service, SNS)의 폭발적인 성장을 견인하는 핵심 동력이 되었다. 페이스북(Facebook), 인스타그램(Instagram), 틱톡(TikTok) 등 다양한 플랫폼들은 시공간의 제약 없이 타인과 소통하고 정보를 교류하는 핵심적인 소통 채널로 기능하고 있다.

1.2. 기존 서비스의 한계

- 현재 주류를 이루고 있는 SNS 플랫폼들은 주로 2차원의 평면적인 미디어, 즉 텍스트, 사진, 그리고 동영상을 기반으로 운영되고 있다. 이러한 형식은 정보를 효율적으로 전달하고 시각적 즐거움을 제공하지만, 사용자의 경험을 스크린이라는 평면적 공간 안에 가두는 본질적인 한계를 가진다. 사용자는 타인의 기록을 수동적으로 '관람'할 뿐, 기록이 담고 있는 공간과 시간에 깊이 몰입하거나 상호작용하기 어렵다. 이는 과거의 순간을 회상하고 추억을 공유하는 방식이 지극히 제한적이다.

1.3. 연구 목표

- 본 연구는 사용자가 생성한 영상 콘텐츠를 기반으로 사용자를 3D Model로 생성하고, 사용자의 SNS 게시물을 AR 환경에서 재현한다. 이를 통해 다른 SNS 사용자와 상호작용하는 경험을 극대화하여 제공하는 소셜 네트워킹 서비스를 개발하는 것을 최종 목표로 한다.

1.4. 개선 방안 및 구현 전략

1.4.1. 3D Model 생성 및 제어

- 기존의 3D Model 제어 방식은 한계가 명확하다. 미리 설정된 몇 가지 애니메이션을 선택하거나, 키보드와 마우스, 컨트롤러 등을 이용한 복잡한 조작은 사용자의 실제 움직임을 즉각적이고 다채롭게 반영하기 어렵다. 이는 사용자의 표현의 자유를 제한하고 서비스 몰입감을 저해한다.
- 본 연구에서는 보다 직관적이고 쉬운 3D Model 제어를 위해 사진 기반 Pose Retargeting을 이용하여 고가의 모션 캡처 장비나 다수의 카메라 없이 자신의 3D Model을 AR 환경에서 재현하고자 한다.

1.4.2. AR 어플리케이션

- AR 카메라로 공간을 인식하고, 이를 지리 좌표계로 추정하여 사용자가 지정한 위치에 AR Anchor를 생성하고 그 위에 특정 3D Model을 세우는 서비스는 이미 많이 접해볼 수 있다.
- 하지만 본 연구는 사진 하나만을 가지고 MediaPipe Pose로 사람의 자세를 인식, 지리 좌표 추정, Blender Retargeting을 거친 개인화 3D Model을 배치한다. 이를 통해 기존 서비스와 비교했을 때, 사용자로 하여금 기록이 담고 있는 공간과 공간에 깊이 몰입하는 경험을 선사하고자 한다.

1.5. 구성원별 역할 및 개발 일정

1.5.1. 구성원별 역할

구성원	담당 파트
김진영	- Unity AR Foundation, Google ARCore Extensions를 활용한 AR 기능 및 안드로이드 어플리케이션 개발
임석윤	- Unity AR Foundation, Google ARCore Extensions를 활용한 AR 기능 및 안드로이드 어플리케이션 개발 - 백엔드 개발
허취원	- 3D Model 생성 및 Blender를 활용한 Pose Retargeting Pipeline 개발

1.5.2. 개발 일정

개발구분	세부항목	3	4	5	6	7	8	9
3D Modeling	동영상 기반 3D Model 생성 기능 개발							
	3D Model Pose Retargeting 기능 개발							
	3D Model 생성 및 Pose Retargeting Pipeline 개발							
Android 앱 개발	3D Model AR world 배치 기능 개발							
	Human Detection 기능 개발							
	개인화 3D Model AR world 배치 기능 개발							
로컬 서버 개발	Flask 서버 개발							
	DB, Storage 개발							
	서버 앱 연동							
테스트 및 배포	보완사항 수정 및 배포							

2. 배경 기술

2.1. 3D Model 생성 및 제어

2.1.1. Pose Detection (MediaPipe Blaze Pose)

- 본 연구에서 사용되는 MediaPipe는 구글에서 만든 Cross-Platform Pipeline Framework로, Pose Landmark, Hands Landmark, Face Detection 등 여러 비전 솔루션을 포함하며 그 중 Pose Detection을 수행할 수 있는 MediaPipe Blaze Pose를 사용한다. MediaPipe Pose는 Blaze CNN (Light CNN Architecture)를 기반으로 하여 경량의 모바일 환경에서도 추론이 가능한 Framework이다.

2.1.2. 3D Model 생성

- 사용자를 3D Model로 변환하는 과정은 2차원 이미지를 기반으로 3차원 형상을 복원하는 Photogrammetry 기술을 기반으로 한다. 정확한 3D 모델 생성을 위해 대상의 모든 각도에서 촬영된 이미지가 필요하다. 본 과제에서는 사용자가 인물의 주변을 원형으로 돌며 동영상을 촬영하여 프레임 단위로 사진을 추출하는 방식으로 진행하였다. 사용가능한 50장의 사진을 선택하여 3DF Zephyr 프로그램을 이용하여 Mesh 형태의 3D Model를 생성하였다.

2.1.3. Rigging

- 3DF Zephyr를 통해 생성된 3D Model은 정적인 상태이다. Model에 Pose를 적용하기 위해서는 내부에 뼈대 역할을 하는 Armature를 심는 Rigging이 필수적이다. 수동 리깅은 전문 지식이 필요하기에 Adobe Mixamo를 활용하여 진행하였다. 인체의 주요 관절(턱, 손목, 무릎 등) 위치를 자동으로 인식하여 표준적인 인간형 (humanoid) Armature를 생성하고 Mesh와 연결된다. 이 과정을 통해 단시간에 포즈 적용이 가능한 동적 3D Model을 확보할 수 있다.

2.2. AR 어플리케이션

2.2.1. 평면 위치 추정 (AR Raycast)

- MediaPipe Pose를 사용하여 사진에서 사람이 서 있는 위치를 알기 위해서는 AR Raycast를 사용해야 한다. AR Raycast란, 카메라 기준의 광선(Ray)을 화면의 한 지점에 쏘아서 AR이 인식한 실제 표면과 교차하는 위치를 찾아오는 과정이다. 간단히 말해서 화면에서 보이는 특정 지점이 AR world 좌표로 어디에 있는지 판단하여 추정해주는 기능이다.

2.2.2. 지구 좌표 추정 (Google ARCore Geospatial API)

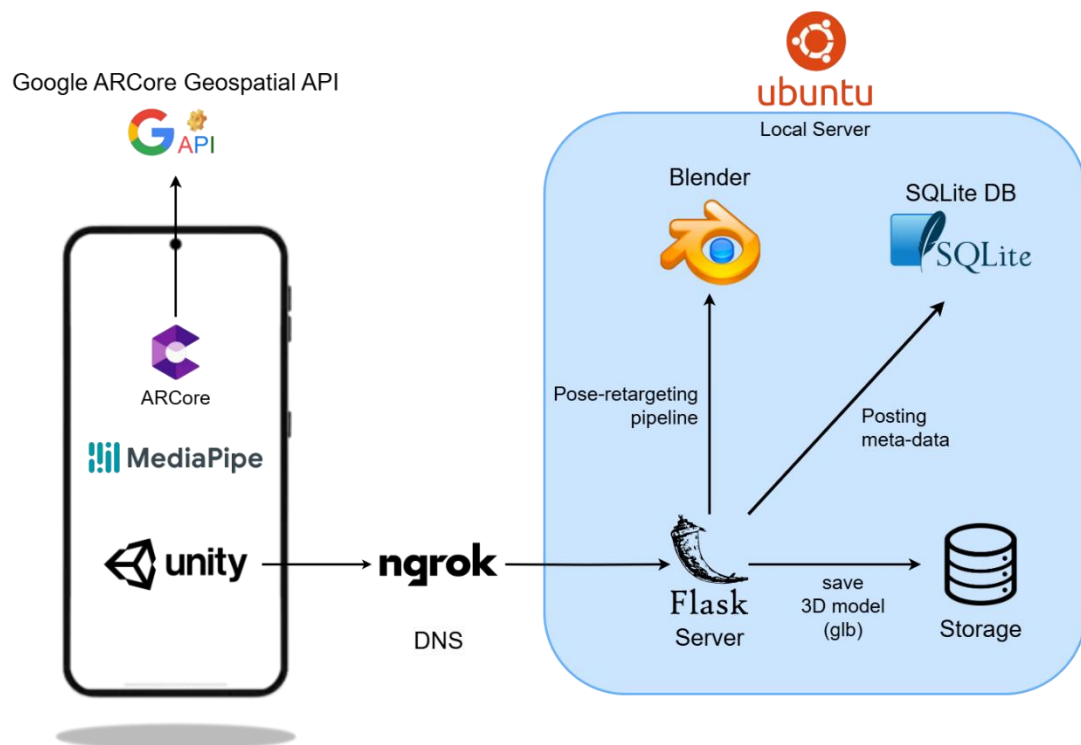
- AR Raycast로 얻은 AR world 좌표는 해당 AR Session이 끝나면 다시 좌표를 찾을 수 없다. 우리가 목표로 하는 기능을 구현하기 위해서는 AR world 좌표인 Local Pose를 지구 좌표인 AR Geospatial Pose로 변환, 이를 데이터화 하여 관리되어야 한다. 이를 구현하기 위해 Google ARCore의 Geospatial API를 사용한다.
- Geospatial Pose는 카메라의 지구 좌표인 (Camera Geospatial Pose)와 ARCore가 만든 Local ENU 좌표계(East-North-Up)을 사용하여 특정 위치의 Geospatial Pose를 계산한다. 이러한 기본 기술을 바탕으로 정확도를 더욱 높이기 위해 VPS라는 기술을 활용한다. VPS (Visual Positioning System)란 카메라 영상과 사전에 구축된 Google Street View, Google Earth 3D 데이터를 함께 활용하여 위치 정확도를 더욱 높이는 것이다. 하지만 야외에서 촬영된 Google Street View의 특성상 실내에서는 정확도가 줄어드는 것이 불가피하다.
- VPS Coverage 지역은 오차범위가 1~5m로 보고되어 있으며, VPS Coverage 지역이 아니면 10~20m의 오차를 가진다. 따라서 Google ARCore Geospatial API의 핵심기능은 VPS를 활용한 지구 좌표 계산이라고 할 수 있다.

2.2.3. 3D Model을 AR world에 표현 (AR Anchor)

- 3D Model을 AR world에 좌표 기반으로 바로 생성한다면, 카메라 화면이 움직임에 따라 3D Model도 미세하게 움직이며 위치 정확도가 떨어진다. 이를 해결하기 위한 기술이 AR Anchor이다.
- AR Anchor는 카메라가 움직이더라도 3D Model이 현실의 특정 위치에 고정되도록 해주는 좌표계 기준점이다. ARCore 엔진은 시간에 따라 추적 품질이 바뀌면 Anchor 위치를 미세 조정하여 더욱 정확한 위치에 3D Model이 위치할 수 있도록 한다. 예를 들어, 처음 3D Model을 세웠을 때는 공간에 대한 정보가 불충분하여 부정확한 위치에 생성될 수 있다. 하지만 AR Anchor 위에 3D Model을 세웠다면, 시간에 따라 공간에 대한 정보 품질이 상승할수록 더욱 정확한 위치에 3D Model을 위치시키는 미세 조정을 수행한다. 이로써 좌표 기반으로 3D Model을 생성하는 방법보다 정확하게 위치시킬 수 있다.

3. 연구 내용

3.1. System Architecture



[그림 3.1.1] System Architecture

3.1.1. Work Flow

3.1.1.1. 기능 1 (AR 포스팅 Upload)

- ① 사용자가 사진을 촬영.
- ② Light Mediapipe Pose(C# plugin)을 사용하여 사진속에 사람이 존재하는지 여부 확인.
- ③ 사람이 존재한다면, 화면상의 픽셀 기준으로 사진 속 서 있는 사람의 발 위치 계산.
- ④ 사람의 발이 있는 화면의 픽셀을 기준으로 AR Raycast를 통해 AR world 좌표 획득.

-
- ⑤ Google ARCore Geospatial API를 사용하여 AR world 좌표(Local Pose)를 지구 좌표(Geospatial Pose)로 변환.
 - ⑥ Preview 화면으로 전환하여 사용자가 촬영한 사진을 확인.
 - ⑦ 포스팅 여부에 따라서 서버에 사진과 지구 좌표(Geospatial Pose) 전송.
 - ⑧ 서버에서 Unity 어플리케이션으로부터 받은 사진을 바탕으로 Full MediaPipe Pose(Python)로 사용자의 Pose Detection.
 - ⑨ Pose Detection정보를 사용자 3D Model에 Pose Retargeting
 - ⑩ 로컬 서버 Storage에 Retargeted 3D Model 저장, 로컬 DB에 포스팅 위치 정보 등의 Meta Data 저장.

3.1.1.2. 기능 2 (AR 포스팅 조회)

- ① Google ARCore Geospatial API를 사용하여 사용자의 지구좌표(Camera Geospatial Pose) 획득.
- ② 사용자 지구 좌표, Unity Web Request를 사용하여 Flask 서버에 주변 포스팅 정보 요청.
- ③ Flask 서버에서 사용자 위치와 가까운 Retargeted 3D Model(.glb), 지구 좌표(Geospatial Pose)를 Unity 어플리케이션에 Response로 전달.
- ④ 서버로부터 받은 GLB 파일을 GameObject Prefab으로 생성.
- ⑤ Unity 어플리케이션에서 지구 좌표를 기준으로 AR Anchor를 생성, Retargeted 3D Model Prefab을 Anchor에 올려서 AR world에 표현.

3.2. Pose Retargeting

- 사용자의 2D 이미지를 입력받아, 이미지 속 인물의 자세를 추출하고 이를 사전에 생성된 3D Model에 적용하여 AR world에서 즉시 사용할 수 있는 GLB 파일로 변환하는 과정이다. 전체 과정은 Pose Landmark 추출, Pose Retargeting, GLB Export로 진행된다.

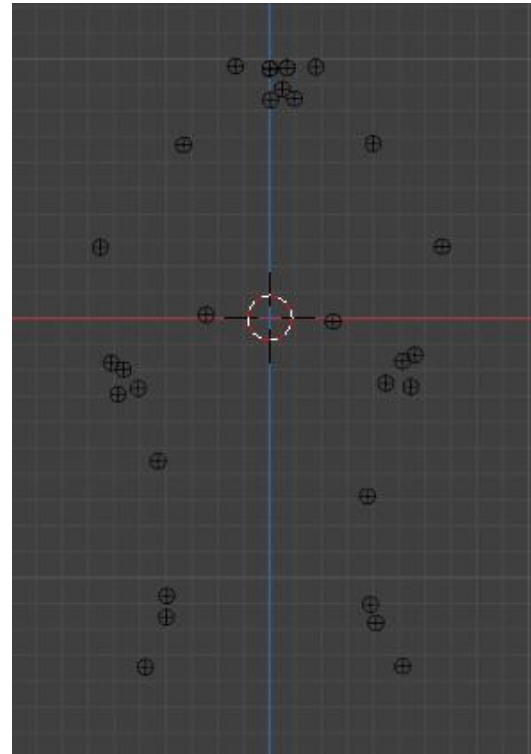
3.2.1. Pose Landmark 추출 / Blender Scene 구성

- MediaPipe의 Pose Landmarker를 활용하여 입력된 2D 이미지에서 인물의 33개 주요 관절점(Pose Landmark)을 감지하고 json파일 형태로 저장한다. 이후 Blender에서 사전에 생성된 3D Model, Pose Landmark를 import 한다. 변환된 33개의 Landmark 좌표는 Blender 씬 내부에 'Empty' Object로 생성되어 시각적으

로 표현된다



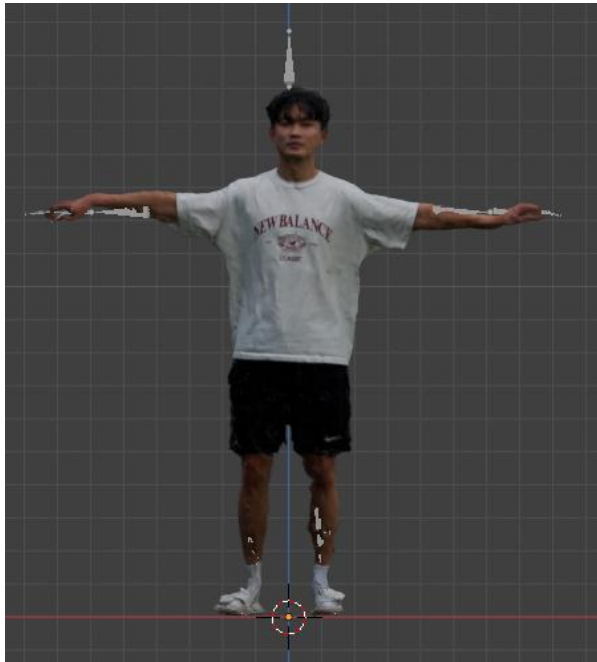
[그림 3.2.1.1] Pose



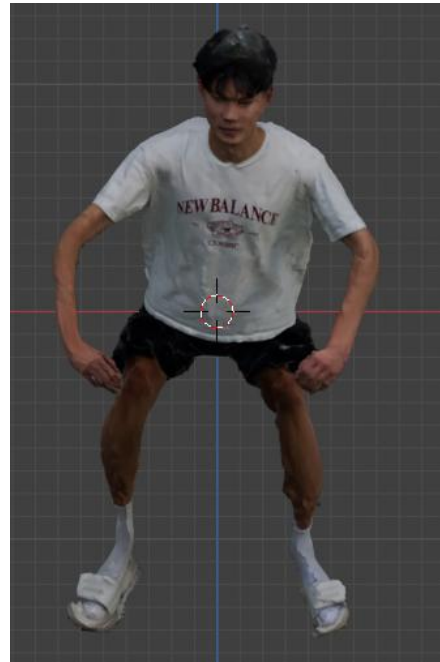
[그림 3.2.1.2] Pose Landmarks 시각화

3.2.2. 자동 스케일링 및 포즈 정렬

- MediaPipe Pose로 추출된 Pose의 신체 비율과 3D Model의 신체 비율이 다를 수 있다. `compute_auto_scale` 함수는 두 Skeleton의 주요 부위(골반, 머리, 발) 간의 상대적 길이를 비교하여 3D Model의 크기를 자동으로 조절하는 Scale Factor를 계산하고 적용한다. 이를 통해 Pose 적용 시 발생할 수 있는 왜곡을 최소화한다. 단순히 Landmark 위치를 1:1로 매칭하는 것이 아니라, `lerp`(선형 보간)와 `center_of`(중심점 계산) 같은 유틸리티 함수를 사용하여 척추, 목, 어깨 등 여러 Landmark의 관계를 고려한 자연스러운 목표 지점을 생성한다



[그림3.2.2.1] Pose Retargeting 전



[그림 3.2.2.2] Pose Retarget 후

3.2.3. GLB 파일 변환

- 제약 조건에 의해 임시로 적용된 Pose를 3D Model의 아마추어(Armature)에 영구적으로 적용하는 '베이킹' 과정을 거친다. Pose가 적용된 Mesh, 아마추어를 선택하여 Unity 환경에서 사용가능한 GLB 파일로 Retargeted 3D Model을 내보낸다.

3.3. Unity ARFoundation / Google ARCore

3.3.1. Human Detection

- Apple의 IOS에서 동작하는 ARKit는 Human Detection이라는 기능이 있다. 하지만 본 연구에서는 Android를 플랫폼으로 채택함에 따라 ARKit는 사용하지 못하고 Google ARCore를 사용한다. 따라서 Human Detection기능을 직접 구현했다.
- workflow에서 설명했듯이, 사용자가 사진을 촬영하면 해당 사진에 사람이 존재하는지 아닌지를 판단해야 한다. 이를 위해 MediaPipe Pose를 사용해 Pose Detection을 수행했다. MediaPipe Pose는 Unity ARFoundation에서 사용하는 언어인 C#을 지원하지 않는다. 따라서 Open Source MediaPipe Unity Plugin(C#)을 도입하여 우리 서비스에 맞게 적용했다. 기존의 MediaPipe Plugin에는 Pose Detection, Hand Detection, Face Detection 등 여러 기능을 지원하지만 휴대폰 어플리케이션 특성상 메모리 최적화를 위해 Pose Detection기능만을 수행하도록 Open Source를 경량화 하였다.

3.3.2. AR world에 포스팅 3D Model 생성

- AR world에 포스팅 3D Model을 생성하기 위해서는 3D Model (Prefab)와 지구 좌표(Geospatial Pose)가 필요하다. 먼저, 지구 좌표를 기준으로 AR Anchor를 생성한다. 이 AR Anchor는 AR world에서 어느 부분에 3D Model를 생성할지 정해놓은 위치이다. 이 Anchor에 3D Model을 올리는 식으로 포스팅 3D Model을 생성한다. 하지만 지구 좌표에서 특히 고도 정보를 통해 AR world에 표현했을 때 오차로 인해 3D Model가 공중에 떠 있는 문제가 발생한다.
- 이를 해결하기 위해 휴대폰의 고도와 Plane Detection결과 고도의 차이를 계산하여 3D Model의 고도를 직접 조정해주었다. 이를 통해 3D Model이 공중에 떠서 생성되는 현상을 해결했다.

4. 연구 결과 분석 및 평가

4.1. 결과물



[그림 4.1.1] 사진 촬영 화면



[그림 4.1.2] Model 배치 화면

-
- 그림 4.1.1과 같이 workflow 기능 1을 성공적으로 구현했다.
 - 그림 4.1.2와 같이 workflow 기능 2를 성공적으로 구현했다.

4.2. 분석

4.2.1. 3D Model 및 Pose Retargeting

- 사진측량 기반 모델링
3D Model을 생성하는 초기 단계는 3DF Zephyr와 같은 사진측량 툴을 사용하여 수동으로 처리하는 오프라인 작업에 의존한다. 이 과정 결과물의 품질은 사용자의 숙련도에 크게 좌우되며 서비스 확장성을 저해하는 가장 큰 제약 요인이다.
- Pose Retargeting의 정확도
Mediapipe를 통해 추출한 포즈를 3D Model에 정렬하는 pipeline은 저성능 CPU 만으로도 충분히 수행할 수 있지만 복잡하거나 일반적이지 않은 자세에서는 관절의 부자연스러운 회전이나 정확하지 않은 관절의 위치와 같은 오차가 관찰될 수 있다.

4.2.2. AR 어플리케이션 및 지구 좌표 정확도

- 온디바이스 성능 최적화
초기 인물 감지를 위해 무거운 라이브러리 대신 경량화된 C# 플러그인을 채택함으로써 모바일 기기의 제한된 리소스를 효율적으로 관리할 수 있었다. 그러나 다수의 복잡한 GLB Model을 동시에 렌더링할 경우, AR 세션의 프레임 저하 및 배터리 소모량 증가는 잠재적인 성능 문제로 남는다.
- 지구 좌표 정확도 및 환경 의존성
실외(VPS 지원 영역) 환경에서는 오차가 1~5m 수준의 높은 정확도를 보였다. 이는 사용자의 3D Model을 본래 의도했던 위치와 매우 근접하게 배치하기에 충분했으며, 서비스의 핵심 가치를 성공적으로 구현했다. 하지만 실내(VPS 미지원 영역) 환경에서는 오차가 10~20m 수준으로 급격히 저하되는 치명적인 약점이 있었다.
- 고도 보정 로직
GPS 기반 고도 정보는 오차가 크기 때문에, 이를 그대로 사용할 경우 3D Model이 공중에 뜨거나 바닥에 파묻히는 현상이 빈번히 발생한다. 이를 해결하기 위해 평면 인식(Plane Detection) 결과를 바탕으로 고도를 동적으로 조정하는 로직을

구현하였고 이는 오차 문제를 효과적으로 해결하는 필수적이고 성공적인 보완책이었다.

4.3. 평가

4.3.1. 연구 목표 달성도

- 2D 이미지를 개인화된 3D Model로 변환하고, 이를 지리적 위치에 고정시키는 기능적 프로토타입을 개발함으로써 핵심 기술 목표를 성공적으로 달성했다. 이는 공간 인지형 소셜 미디어라는 새로운 패러다임의 가능성을 입증하는 강력한 개념 증명(Proof-of-Concept)으로 기능한다.

4.3.2. 강점 및 혁신성

- 엔드투엔드 시스템 통합
모바일 AR, 컴퓨터 비전(Mediapipe), 3D 모델링(Blender), 백엔드 서비스(Flask) 등 이질적인 기술 스택을 성공적으로 통합하여 하나의 유기적인 시스템으로 구현한 것이 본 프로젝트의 가장 큰 기술적 성과이다.
- 새로운 사용자 경험
'meARy'의 핵심 개념, 즉 물리적 공간에 자신의 3D '잔상'을 남겨 다른 사람이 발견하게 하는 경험은 기존의 평면적이고 휘발적인 SNS와 차별화되는 새롭고 매력적인 상호작용 모델을 제시한다.

4.3.3. 한계

- 확장성 문제
부족한 컴퓨팅 자원과 예산으로 인해 채택하게 된 수동적인 사진측량 및 리깅 과정은 실제 소셜 애플리케이션에 적용하기에는 비현실적이다.
- 정적인 콘텐츠
최종 결과물은 단일 포즈의 정적인 3D Model이다. 이는 AR 경험의 '생동감'과 표현의 잠재력을 근본적으로 제한한다. 사용자의 기억은 동적인 순간이 아닌, 정지된 하나의 찰나로 박제된다.
- 환경적 제약
실내에서의 위치 정확도 저하는 서비스의 사용성을 실외 VPS 지원 영역으로 국한시키는 운영상의 한계이다.

5. 결론 및 향후 연구 방향

5.1.1. 결론

- 본 연구 '메아리(meARy)'는 증강 현실과 지구 공간 정보 기술을 융합하여, 사용자의 사진을 실제 공간에 고정되는 개인화된 3D Model로 변환하는 새로운 소셜 네트워킹 패러다임의 실현 가능성을 성공적으로 입증하였다. 이 시스템은 기존의 2D 콘텐츠 중심의 소셜 미디어와 차별화되는 몰입형 경험의 청사진을 제시했다. 그러나 분석 과정에서 확장성, 콘텐츠의 역동성, 그리고 실내 환경에서의 작동 신뢰성 등 몇 가지 한계점들이 식별되었다. 본 프로젝트는 이러한 한계에도 불구하고, 미래의 공간 기반 소셜 플랫폼 연구를 위한 기초 프레임워크를 구축하고 핵심 사용자 경험을 검증했다는 점에서 의의를 가진다.

5.1.2. 향후 연구 방향

- 현재 시스템을 한계를 극복하고 발전시키기 위해 다음과 같은 연구 방향을 제안한다.

5.1.2.1. 3D Model 생성 방식 변화: SMPL + Texturing

- SMPL(Skinned Multi-Person Linear) 모델을 활용해 다양한 신체 비율과 포즈를 정밀하게 재현하고 Texturing을 통해 실제 사용자 이미지로부터 질감을 추출하고 3D Model에 매핑하는 방식을 연구해 기존 3D Model 생성 방식의 한계를 해결한다.

5.1.2.2. 환경적 제약 극복: 하이브리드 실내 측위 시스템 도입

- ARCore의 시각-관성 SLAM(Visual-Inertial SLAM) 기술과 블루투스 저전력(BLE) 비콘과 같은 영구적인 위치 인식 기술을 융합한 하이브리드 실내 측위 시스템을 개발해 신뢰도 높은 실내 경험을 제공한다.

6. 참고 문헌

- [1] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "BlazePose: On-device Real-time Body Pose Tracking," arXiv preprint arXiv:2006.10204, Jun. 2020.
- [2] Google Research. (2020, Aug 13). On-Device, Real-time Body Tracking with MediaPipe BlazePose. [Online]. Available: <https://research.google/blog/on-device-real-time-body-pose-tracking-with-mediapipe-blazepose> (Accessed 2025, Sep. 18)
- [3] Unity Technologies. "AR Foundation (Package: com.unity.xr.arfoundation) Manual," [Online]. Available: <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@6.1/manual/index.html> (Accessed 2025, Sep. 18)
- [4] Google ARCore. "ARCore Extensions for AR Foundation (Package: Google.XR.ARCoreExtensions) Document" [Online]. Available: <https://developers.google.com/ar/develop> (Accessed 2025, Sep. 18)