

상세 요구사항 명세서

컴퓨터공학 종합설계[202502-CSE4205-001]



그리마	팀장	정현민(12201392)
	팀원	최민석(12201812)
	팀원	박준희(12201833)

목차

I. 개요	4
1. 문서 목적	
2. 프로젝트 배경 및 필요성	
3. 프로젝트 목표	
4. 주요 차별성	
II. 시스템 전반 구조	5
1. 전체 아키텍처 개요	
2. 파이프라인 흐름도	
3. 주요 모듈 정의	
III. UI 및 기능 요구사항	8
1. 홈 화면	
2. 이미지 입력 화면	
3. 3D 미리보기 화면	
4. AR 배치 화면	
5. 최근 프로젝트 화면	
IV. 비기능 요구사항	13
1. 성능 (FPS, 처리 지연, AR 안정성)	
2. 확장성 (디바이스/플랫폼 확장, 타 AR 엔진 호환)	

3. 사용성 (UX 직관성, 반응성, 접근성)

V. 개발 환경 및 기술 스택 ----- 14

1. 개발 언어 및 프레임워크
2. 버전 관리 및 협업 도구
3. 외부 라이브러리/SDK

1. 개요

1.1 문서 목적

- 1.1.1 본 문서는 3DGS 기반 AR 렌더링 플랫폼의 상세 요구사항 명세서(SRS)이다.
사용자가 촬영한 2D 이미지를 입력 받아 3D Gaussian Splatting(이하 3DGS) 기법으로 재구성하고, AR 환경에서 자연스러운 조명·그림자·반사 효과와 함께 배치하여 현실과 어우러지는 결과물을 제공하는 시스템을 정의한다.

1.2 프로젝트 배경 및 필요성

- 1.2.1 기존 3D 재구성 기법은 다수의 이미지를 필요로 하며, 환경 조명 변화에 대한 적응력이 부족하다.
- 1.2.2 사용자가 소수의 이미지만으로도 실시간 3D 모델을 얻을 수 있는 경량화된 파이프라인이 요구된다.
- 1.2.3 AR 환경에서 가상 객체가 실제 환경과 부조화를 일으키는 문제(빛 반사, 그림자 부재, 위치 재배치 오류 등)가 빈번하게 발생한다.
- 1.2.4 따라서, **소수 프레임 기반 3DGS 재구성과 실시간 Lighting 동기화**를 지원하는 플랫폼의 필요성이 크다.

1.3 프로젝트 목표

- 1.3.1 **3D 재구성:** 소수의 2D 이미지를 입력으로 받아 Gaussian Splatting 기반 3D 모델을 재구성한다.
- 1.3.2 **AR 렌더링:** Unity 및 AR Foundation 을 이용해 현실 공간에 가상 모델을 배치한다.
- 1.3.3 **Interactive Lighting:** 환경광(빛, 그림자, 반사)을 동기화하여 현실감 있는 렌더링을 구현한다.

- 1.3.4 **사용성:** 사용자 친화적인 UI 를 제공하여 이미지 업로드, 미리보기, AR 배치, 저장/공유가 직관적으로 가능하도록 한다.

1.4 주요 차별성

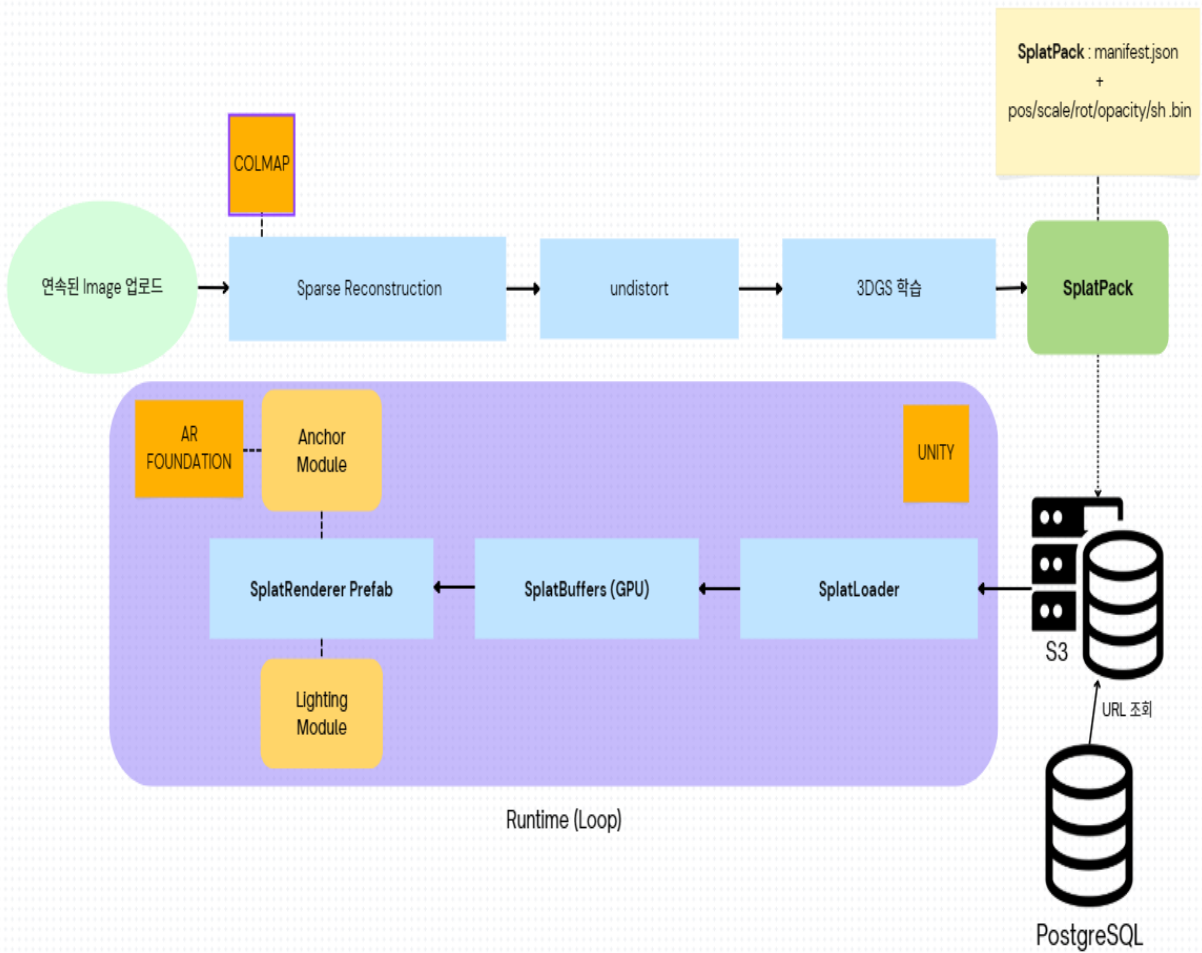
- 1.4.1 **소수 프레임 Reconstruction:** 기존 다수 이미지 기반 기법 대비 적은 입력으로도 빠른 3DGS 재구성 가능.
- 1.4.2 **Lighting 동기화:** Spherical Harmonics 기반 IBL-lite 기법으로 환경광을 반영한 동적 라이팅으로 현실감 강화.
- 1.4.3 **AR 안정성:** ARKit/ARCore 와 좌표 동기화를 통해 카메라 이동 후에도 객체가 사라지거나 재배치 오류가 발생하지 않음.
- 1.4.4 **결과물 활용성:** 생성된 3D 객체를 이미지 혹은 3D 모델 형태로 저장 및 공유 가능.

2. 시스템 전반 구조

2.1 전체 아키텍처 개요

- 2.1.1 **입력 모듈:** 사용자가 업로드한 2D 이미지를 수집 및 전처리(크기 정규화, 노이즈 제거)
- 2.1.2 **3DGS 재구성 모듈:** Colmap 을 사용한 Gaussian Splatting 기반으로 3D 모델 생성.
- 2.1.3 **AR 렌더링 모듈:** Unity 및 AR Foundation 을 이용한 AR 배치.
- 2.1.4 **Lighting 모듈:** IBL-lite, Spherical Harmonics 기반으로 환경광 반영.
- 2.1.5 **출력/저장 모듈:** 결과물 시각화 및 파일 저장/공유.

2.2 파이프라인 흐름도



SplatPack: 3DGS 학습 산출물(파일셋). 엔진 독립, 저장소에 두는 대상.

SplatLoader: SplatPack을 다운로드/검증/파싱하는 런타임 로더.

SplatBuffers: 로더가 만든 GPU 버퍼 상태(런타임 산물).

SplatRenderer Prefab: 셰이더/머티리얼이 미리 연결된 Unity 프리팹. 버퍼값만 전달

받아 렌더링.

2.3 주요 모듈 정의

2.3.1 입력 모듈

2.3.1.1 **역할:** 이미지 업로드, 기본 전처리(크기 정규화, 노이즈 제거) 수행.

2.3.1.2 **입출력:** (Input) 2D 이미지 파일 / (Output) 정규화된 이미지 데이터.

2.3.2 3DGS 재구성 모듈

2.3.2.1 **역할:** Gaussian Splatting 기법으로 포인트 클라우드 및 텍스 기반 3D 모델 생성.

2.3.2.2 **특징:** Colmap sfM, 시간 단축을 위한 2k 이내의 iteration, sh_degree = 2 사용.

2.3.3 AR 렌더링 모듈

2.3.3.1 **역할:** Unity 기반 AR 배치, ARKit / ARCore를 이용한 공간 매핑 및 좌표 추적.

2.3.3.2 **특징:** 카메라 이동 시에도 객체 위치를 안정적으로 유지.

2.3.4 Lighting & IBL 모듈

2.3.4.1 **역할:** Spherical Harmonics(2차) 기반 환경광 반영.

2.3.4.2 **기능:** 그림자, 반사, 빛의 강도·방향 자동 동기화.

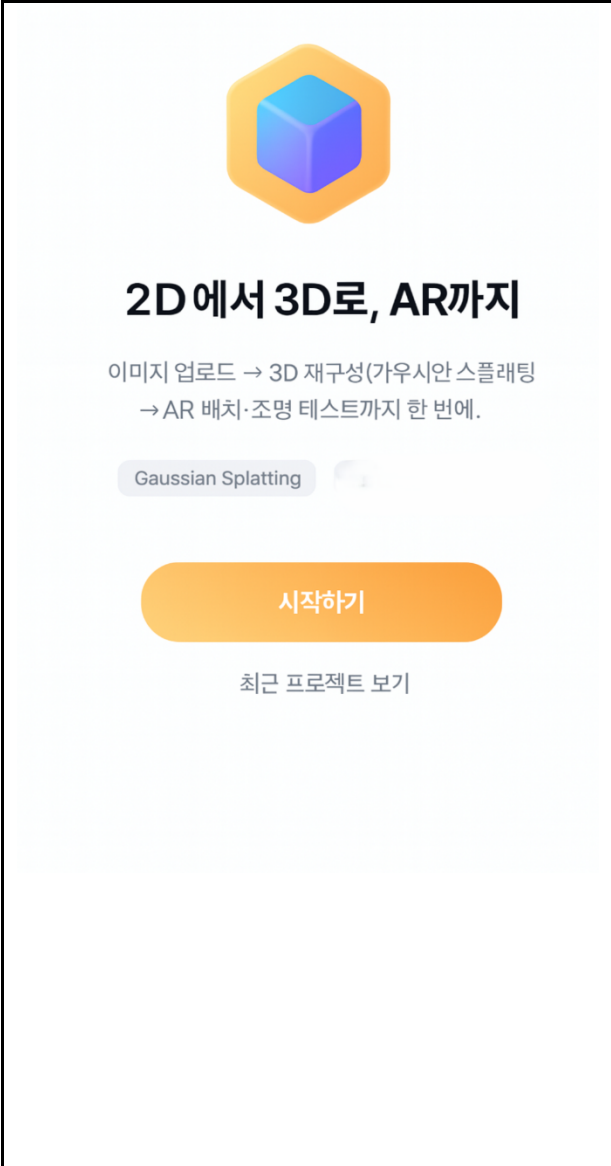
2.3.5 출력/저장 모듈

2.3.5.1 **역할:** 결과물 시각화 및 파일 저장/공유.

2.3.5.2 출력 포맷: SplatRender(AR composite) : 실시간 화면, splatPack / PLY: 재사용 가능한 모델 아티팩트.

3. UI 및 기능 요구사항

3.1 홈 화면

홈 화면	
와이어프레임	요구사항
 <p>2D에서 3D로, AR까지</p> <p>이미지 업로드 → 3D 재구성(가우시안 스플래팅) → AR 배치·조명 테스트까지 한 번에.</p> <p>Gaussian Splatting</p> <p>시작하기</p> <p>최근 프로젝트 보기</p>	<p>기능 설명</p> <p>앱 실행 직후, 사용자는 새 프로젝트 시작(이미지 업로드) 또는 최근 프로젝트 이어하기 중 하나를 선택할 수 있어야 한다.</p> <p>상세 설명</p> <ol style="list-style-type: none"> 1. 앱 소개 표시 홈 화면 상단 중앙에 앱 로고와 앱 소개 문구를 표시한다. 2. 시작하기(이미지 업로드) 버튼 홈 화면 중앙에 "시작하기" 버튼을 배치한다. 클릭 시, 기기 파일 탐색기를 실행한다. 허용 파일 형식: .jpg, .png 3. 최근 프로젝트 이어하기 버튼 클릭 시, 기기 내 저장된 프로젝트 목록 창을 표시한다. 4. 로딩 상태 표시 "시작하기" 또는 "최근 프로젝트 이

	<p>어하기" 실행 시, 로딩 인디케이터 (회전 아이콘)와 "불러오는 중..." 텍스트를 화면 중앙에 표시한다.</p>
--	--

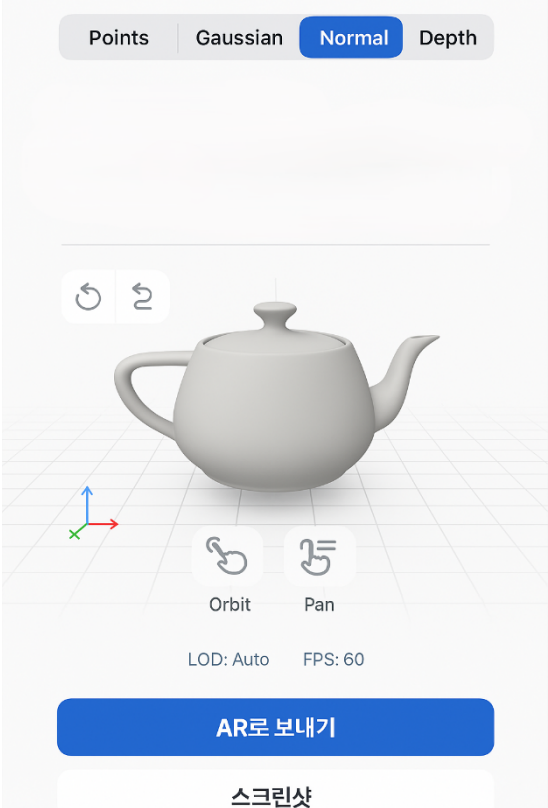
3.2 이미지 업로드 화면

이미지 입력 화면	
와이어 프레임	상세 요구사항

	<h3>기능 설명</h3> <p>사용자는 3D 재구성에 필요한 4장의 이미지를 입력(촬영 또는 업로드) 해야 하며, 각 이미지는 서로 다른 각도에서 촬영된 것이어야 한다. 입력된 이미지는 형식 및 용량 기준을 검증하고, 검증이 완료되면 다음 단계로 진행할 수 있다.</p> <h3>상세 설명</h3> <ol style="list-style-type: none"> 1. 이미지 촬영 / 업로드 화면 중앙에 "이미지 추가" 버튼을 배치한다. 버튼 클릭 시, 사용자는 다음 중 하나를 선택할 수 있어야 한다. (카메라 촬영 / 기기 파일 탐색기에서 업로드) 2. 이미지 권장 각도 안내 각 이미지 업로드 슬롯(총 4개)에 권장 각도를 시각적으로 안내한다. 3. 용량 및 형식 체크 (필요시 재조정) 기준을 만족하지 못하면 업로드 실패 처리 후, "이미지를 다시 선택하거나 크기를 줄이세요." 안내 4. 진행도 표시 화면 하단에 진행도 바 배치 5. 입력 기준 만족 시, 다음으로 진행
--	--


3.3 3D 미리보기 화면

3D 미리보기 화면




와이어 프레임	상세 요구사항
	<p>기능 설명</p> <p>사용자는 3D 재구성된 결과물을 모바일 화면에서 실시간으로 확인할 수 있으며, 카메라 조작을 통해 다양한 각도에서 모델을 관찰할 수 있다. 또한, 모델 품질 정보를 확인하고, 결과물을 AR 환경으로 전송하거나 스크린샷으로 저장할 수 있어야 한다.</p> <p>상세 설명</p> <ol style="list-style-type: none"> 1. 모델 로드 및 렌더링 3D 재구성이 완료되면 자동으로 3D 미리보기 화면으로 전환된다. 2. 카메라 조작(관찰 기능) 터치 제스처를 통해 모델을 조작할 수 있어야 한다. 3. 모델 정보 및 품질 표시 모델의 Point, Gaussian, Normal, Depth 정보를 표시한다. 4. 스크린샷 및 AR로 보내기 스크린샷 클릭 시, 현재 뷰포트 이미지를 저장하여 기기 앨범에 저장 AR 보내기 클릭 시, 현재 모델을 AR 환경에 배치한다.

3.4 AR 배치 화면

AR 배치 화면

와이어 프레임	상세 요구사항
	<p>기능 설명</p> <p>사용자는 3D 모델을 AR 환경에 배치할 수 있다. 모델은 탐지된 평면 위에 고정되며, 제스처로 위치·크기·회전을 조정할 수 있다. 실제 환경의 조명과 그림자가 모델에 실시간 반영되어야 하며, 사용자는 화면 캡처 및 조명 설정 기능을 이용할 수 있다.</p> <p>상세 설명</p> <ol style="list-style-type: none"> 1. 평면 탐지 / 앵커 고정, 좌표 동기화 앱 실행 시 AR SDK를 통해 바닥 및 테이블 등의 평면을 자동 탐지한다. 2. 배치 제스처 (탭 배치 / 드래그 이동 / 스케일 / 회전) 3. 실시간 조명 모델은 카메라와 센서를 통해 측정된 환경광과 방향광에 따라 실시간으로 밝기와 색상을 조정한다. 4. 캡처 및 조명 설정 조명 설정 변경 사항은 즉시 3D 모델과 그림자에 반영되어야 한다.

3.5 최근 프로젝트 화면

최근 프로젝트 화면	
와이어 프레임	상세 요구사항
<div style="text-align: center;"> <h2>최근 프로젝트</h2> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin-bottom: 10px;"> Q 검색: 이름, 태그, 형식... </div> <div style="display: flex; justify-content: space-around; margin-bottom: 10px;"> 전체 AR 준비됨 즐거찾기 </div> <div style="display: grid; grid-template-columns: 1fr 1fr; gap: 10px;"> <!-- Project 1 --> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; text-align: center;">  <p>프로젝트 1 2024. 03.22 · 2.3 MB</p> <p style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 5px;">AR 준비됨</p> ⋮ </div> <!-- Project 2 --> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; text-align: center;">  <p>프로젝트 2 2024. 03.20 · 1,3 MB</p> <p style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 5px;">AR 준비됨</p> ⋮ </div> <!-- Project 3 --> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; text-align: center;">  <p>프로젝트 3 2024. 03.22 · 3,7 MB</p> <p style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 5px;">AR 준비됨</p> ⋮ </div> <!-- Project 4 --> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; text-align: center;">  <p>프로젝트 4 2024. 03.22 · 420 MB</p> <p style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 5px;">AR 준비됨</p> ⋮ </div> </div> <div style="display: flex; justify-content: space-around; margin-top: 20px;"> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px 20px; background-color: #f9f9f9;">가져오기(외부)</div> <div style="background-color: #007bff; color: white; padding: 10px 20px; border-radius: 10px;">+ 새 프로젝트</div> </div> </div>	<p>기능 설명</p> <p>사용자는 저장된 3D 프로젝트 결과물을 리스트 형태로 확인하고, 검색·정렬·필터 기능을 통해 원하는 프로젝트를 빠르게 찾을 수 있다. 각 프로젝트는 썸네일 미리보기를 통해 확인 가능하며, 선택 시 프로젝트를 불러와 재사용할 수 있다.</p> <p>상세 설명</p> <ol style="list-style-type: none"> 저장된 프로젝트 리스트 표시 화면에 저장된 프로젝트가 카드 형태로 나열된다. 검색/정렬/필터 (AR 준비됨, 즐겨찾기 등) 화면 상단에 검색 창 배치하여 프로젝트명 기준으로 실시간 검색이 가능하다. 미리보기 및 가져오기 기능 제공 프로젝트 카드 선택 시, 전체화면 미리보기 창이 열린다.

4. 비기능 요구사항

4.1 성능

4.1.1 AR 렌더링 FPS

4.1.1.1 목표(Goal): ≥ 30 FPS(안정 시연 환경, 스마트폰(iOS/Android) 기준).

4.1.1.2 최소 허용(Acceptable): ≥ 24 FPS

4.1.2 지연 (Latency)

4.1.2.1 모델 재구성: 4장 소수 프레임 입력 \rightarrow 3DGS 모델 생성: 목표 ≤ 5 분

4.1.2.2 로컬 라이트 추정 & 적용: 카메라 프레임 수신 \rightarrow SH 계수 업데이트 세이더 반영: ≤ 200 ms (프레임당 실시간 적용 범위).

4.1.2.3 업로드/응답: 이미지 업로드 \rightarrow 서버 응답(수락 확인): ≤ 2 s

4.1.3 안정성 / 내구성

4.1.3.1 연속 시연(30분) 동안 크래시 없음(99% 안정성).

4.1.3.2 재구성 파이프라인 실패율: $< 5\%$ (정상 입력 조건에서).

4.2 확장성

4.2.1 플랫폼 확장성: AR Foundation(유니티) 사용으로 iOS/Android 동시 지원. (단, 시연은 Android 기기로만 진행함)

모듈(렌더러, 라이트 추정, 앵커관리)은 인터페이스로 분리하여 다른 엔진(예: Native ARKit UI)로 교체 가능하나 본 프로젝트에서는 교체하지 않음.

4.2.2 데이터 포맷 호환성: .glb/.obj 및 Gaussian Splat 전용 포맷(엔진 스페셜 자산) 모두를 지원하도록 파이프라인 설계.

4.3 사용성

4.3.1 **반응성**: UI 버튼 클릭 → 피드백(로딩/상태 표시)은 $\leq 200\text{ ms}$ 내에 표시. 오래 걸리는 작업은 진행도(%) 제공.

4.3.2 **온보딩**: 신규 사용자 온보딩 흐름은 초당 1 화면, 전체 최초 사용 플로우(설명 포함) ≤ 60 초.

4.3.3 **접근성**: 텍스트 대비, 버튼 최소 크기(44-48 px), 키보드/스크린리더 고려.

4.3.4 **언어**: 한국어 고정

4.3.5 **오류 메시지**: 사용자가 취할 수 있는 조치(예: 재시도, 이미지 재촬영 권장)를 포함한 친절한 오류 메시지.

5. 개발 환경 및 기술 스택

5.1 개발 언어 및 프레임워크

5.1.1 **Unity (C#)** : 앱 전체 UI (온보딩, 업로드, 미리보기, 설정 등), AR Foundation 으로 AR 기능 (배치, 트래킹, 라이트 추정 반영), Shader/HLSL 로 조명, 그림자 효과

5.1.2 **Python** : 2D → 3DGS 재구성 파이프라인, Unity 앱에서 REST API 로 업로드/결과 다운로드

5.2 버전 관리 및 협업 도구 : Git, Github

5.3 외부 라이브러리 / SDK / 툴

5.3.1 **3DGS / NeRF 계열**: 기존 공개 구현(학습/참고용) - 파이프라인 최적화 후 사용.

5.3.2 **COLMAP**: 카메라 파라미터/초기 SfM(필요시).

5.3.3 **PyTorch**: 학습/미세조정(필요 시).

5.3.4 **OpenCV**: 특징 추출(ORB), 매칭, RANSAC, Homography.

5.3.5 **Unity AR Foundation, ARKit, ARCore**: AR 핵심 툴.

5.3.6 **Database/Storage**: PostgreSQL, S3 호환 스토리지(모델/asset).