



RADIANCEKIT

사용자 매뉴얼

Gaussian Splatting을 활용한
사실적인 3D 재구성

Version 1.5.0 · macOS 26.0+ · 2026년 5월

BJOERN KINDLER · KINDLER-DEV.DE

개요

소개 — 알아두어야 할 사항	3
RadianceKit이란?	3
Gaussian Splatting이란?	3
제 1 장 — 메뉴 바	5
File 메뉴	5
Mode 메뉴	9
Training 메뉴	10
Viewport 메뉴	14
Export 메뉴	19
Help 메뉴	25
참고: Edit 메뉴의 Cmd-Z	29
키보드 단축키 개요	30
제 2 장 — 인스펙터 (Expert View)	31
Look 섹션 (L1–L5)	33
Presets 섹션 (I1–I11)	36
Training Configuration 섹션 (I12–I22)	42
Enhancements 섹션 (I26–I29, I42–I44)	48
Metrics 섹션 (I30–I38)	55
Loss Chart 섹션 (I39–I41)	61
언제 인스펙터를 사용해야 하는가?	64
제 3 장 — 설정	66
General 탭	67
AI Helpers 탭	71
인스펙터 미리 설정	74
언제 무엇을?	75
제 4 장 — 보조 창	76
User Guide (W1–W4)	77
Keyboard Shortcuts (W5–W6)	80
Manage Storage (W7–W12)	82
Pareto Dashboard (W13–W22)	85
Holdout Analysis (W23–W29)	91
BayesOpt Console (W30–W39)	96
메인 윈도우: Loss 진행과 Gaussian Count (I39–I41, 상호 참조)	102
경험 법칙 상자	102
제 6 장 — 학습 구성	104
Iteration (T1–T2)	106
Learning Rates (T3–T10)	108

Densification — Classic (T11–T16)	113
Loss (T17–T20)	116
SH Degree Progression (T21)	119
Performance (T22–T25)	120
진단과 점 구름 준비 (T26–T30)	122
정규화 (T31–T37)	124
Refinement (T38–T44)	128
Sky-Dome (T45–T48)	132
Adam + LR 스케줄 (T49–T55)	134
Post-Processing + Apple AI (T56–T60)	137
MCMC Densification (T61–T73)	139
Mip-Splatting (Q1.5) (T74–T76)	146
Adaptive Densification (Q5) (T77–T79)	148
Curriculum (Q6) (T80–T81)	149
정적 프리셋 (TP1–TP9)	150
메서드: resolveMcmcMaxGaussians	153
어느 필드를 무엇을 위해? (치트 시트)	155
위험한 필드	155
제 7 장 — 내장 품질 프리셋	157
언제 어느 프리셋?	166
빠른 비교	167
자체 프리셋	167
제 8 장 — 내보내기 형식	169
어느 형식을 언제?	180
빠른 비교	181
제 9 장 — SfM 백엔드	182
어느 백엔드를 언제?	187
빠른 비교	188
제 10 장 — 초보자 모드	189
Z1 — 가져오기 (이미지와 프리셋 선택)	189
Z2 — 처리 (SfM + 학습)	198
Z3 — 미리보기 (3D 모델 회전)	206
Z4 — 내보내기 (형식 선택과 저장)	209
Expert 모드로 전환	214
자주 묻는 질문	214

이 매뉴얼을 읽는 방법

본 매뉴얼의 각 항목은 동일한 구성으로 작성되어 있습니다. 왼쪽에는 조작 경로와 기술적 세부 정보가, 오른쪽에는 따뜻한 색의 사이드바에 항상 간단한 설명이 표시됩니다. 줄 머리에 있는 작은 아이콘을 보면 곧이어 어떤 종류의 정보가 나오는지 한눈에 파악할 수 있습니다.

네 가지 아이콘



어디에 있습니까? 앱 내에서의 구체적인 클릭 경로입니다 — 메뉴 바, 인스펙터 섹션 또는 초보자 모드의 단계를 가리킵니다. 관련된 키보드 단축키도 여기에 함께 표시됩니다. 아이콘은 지도 위의 핀으로, 해당 기능이 사용자 인터페이스의 어느 위치에 있는지를 보여줍니다.



세부 정보. 기본값, 값 범위, 코드 경로입니다. 메뉴 항목이 아닌 숫자 매개변수인 학습 설정 항목에서 특히 자주 등장합니다. 아이콘은 작은 사양 카드를 나타냅니다.



기술적 설명. 해당 기능이 내부적으로 무엇을 수행하는지, 어떤 매개변수가 영향을 주는지, 무엇에 반응하는지, 그리고 어떤 부작용이 있는지를 설명합니다. 보닛 아래에서 무슨 일이 일어나고 있는지를 이해하고자 하는 독자를 위한 항목입니다. 아이콘은 슬라이더 블록으로, 내부 조정 장치를 상징적으로 나타냅니다.



간단히 말하면. 전문 용어나 코드 없이 핵심 내용을 명확한 언어로 전달합니다. 기능의 용도와 사용해야 할 시점을 빠르게 알고 싶다면 먼저 이 섹션을 읽으십시오. 아이콘은 말풍선으로, "요점만 정리하면"을 의미합니다. 이 영역은 항상 따뜻한 모래색 배경으로 칠해져 있어 눈에 잘 띕니다.

장별 색상

각 장에는 고유한 강조 색상이 있으며, 각 항목 제목 왼쪽에 있는 ID 마크 (예: **M1**) 와 그 앞에 있는 작은 아이콘으로 식별할 수 있습니다. 페이지를 넘길 때 현재 어느 장에 있는지 한눈에 알 수 있습니다.

- 1 메뉴
- 2 인스펙터
- 3 설정
- 4 보조 창
- 6 학습
- 7 프리셋
- 8 내보내기
- 9 SfM
- 10 초보자 모드

탐색 팁

빠른 시작. 조작 방법만 알고 싶다면 곧바로 **제 10 장 — 초보자 모드** 로 이동하십시오. 사전 지식이 전혀 필요 없는 4 단계 가이드 형식입니다.

깊이 있는 학습. 제 2 장 — 인스펙터 와 제 7 장 — 프리셋 에서는 전문가 모드에서 사용할 수 있는 조작 요소와 사전 설정된 품질 프로파일을 설명합니다.

참조. 목차와 PDF 전문 검색을 사용하면 특정 기능을 찾을 수 있습니다. 매뉴얼을 처음부터 끝까지 읽을 필요는 없습니다.



소개 — 알아두어야 할 사항

RadianceKit이란?

RadianceKit은 일반적인 사진 묶음 또는 동영상으로부터 걸어 다닐 수 있는 3D 재구성을 만들어 주는 네이티브 macOS 앱입니다. 입력 데이터는 예를 들어 대상 주위, 실내 공간 또는 풍경 위에서 촬영한 50장에서 500장 정도의 사진들입니다. 출력은 이른바 Gaussian Splatting 장면입니다 — Mac에서 모든 시점에서 실시간으로 볼 수 있고, 내보내어 웹사이트에 임베드할 수도 있으며, 주요한 측면에서 사실적으로 보이는 3D 모델입니다.

이 앱은 Mac에서 완전히 로컬로 실행됩니다 — 어떤 이미지도 클라우드에 업로드되지 않고, 로그인이 요구되지 않으며, 구독도 없습니다. Apple Silicon Mac (M 시리즈) 의 GPU를 집중적으로 활용합니다. 전체 학습은 장면과 프리셋에 따라 2분에서 수 시간이 걸릴 수 있습니다. 계산이 진행되는 동안에도 Mac에서 평소처럼 다른 작업을 계속할 수 있으며, RadianceKit은 백그라운드에서 실행되어 결과가 완성되면 알림을 보냅니다.

조작 모드는 두 가지입니다. 초보자 모드 (Simple Mode) 는 가져오기 → 프리셋 선택 → 학습 → 내보내기 워크플로를 4단계로 안내합니다. 전문가 모드 (Expert Mode) 에서는 모든 설정 항목을 갖춘 큰 인스펙터와 실시간 미리보기 창, 그리고 진단 차트가 열립니다. 모드 사이는 언제든지 전환할 수 있으며, 장면 데이터는 그대로 유지됩니다.

Gaussian Splatting이란?

Gaussian Splatting (흔히 줄여서 3DGS 또는 단순히 *Splatting*) 은 2023년 그라츠와 INRIA의 논문에서 발표된 비교적 새로운 사실적 3D 표현 기법입니다. 아이디어는 단순합니다. 장면을 고전적인 폴리곤 메시 (삼각형) 나 복셀 그리드로 모델링하는 대신, 수백만 개의 작고 부드러운 3D 구름으로 구성합니다 — 각 구름은 3D 가우시안 분포 (이름이 여기서 유래합니다) 이며 자체적인 위치, 크기, 형태, 색상, 투명도를 가집니다. 이 구름들은 입력 사진의 모든 시점에서 함께 보았을 때 올바른 이미지가 얻어지도록 학습됩니다.

실제로 말하자면 Gaussian Splatting은 반사, 하이라이트, 부드러운 잎, 머리카락, 커튼 같은 것을 고전적인 3D 모델링으로는 불가능하거나 막대한 비용을 들여야만 가능한 방식으로 표현할 수 있습니다. 그 대신, 결과는 고전적인 의미에서 편집할 수 있는 3D 모델은 아닙니다 — 벽 하나를 옮기거나 꽃병 하나를 교체하거나 하는 작업은 할 수 없습니다. 오히려 자유롭게 걸어 다닐 수 있는 공간의 얼어붙은 한 컷에 가깝습니다. 건축 시각화, 제품 프레젠테이션, 가상 투어, 법과학, 문화유산 보존과 같은 많은 용도에서는 바로 이런 점이 적합한 강점이 됩니다.

입력 이미지로부터 3D 장면을 만들기 위해서는 두 단계가 필요합니다. 먼저 앱은 *Structure-from-Motion (SfM)* 이라는 기법을 사용하여 각 사진을 촬영할 때 카메라가 있던 위치를 계산합니다. 그 과정에서 부산물로 장면의 거친 점 구름이 생성됩니다. 그런 다음 본격적인 Gaussian Splatting 학습이 시작됩니다. 이 거친 점 구름에서 출발하여 수백만 개의 3D 구름이 단계적으로 분포되고, 확대되고, 정교화되며, 위치와 색상이 미세 조정되어 모든 입력 시점에서 일치하는 이미지를 만들어내게 됩니다.

RadianceKit을 사용하기 위해 이러한 내부 동작을 반드시 알아야 하는 것은 아닙니다. 초보자 모드는 이 단계들을 완전히 숨겨 줍니다. 그러나 전문가 모드의 진단 수치 (Iteration, Loss, Gaussians, SSIM ...) 가 무엇을 의미하는지, 또는 왜 어떤 장면은 다른 장면보다 더 아름다워지는지 이해하고 싶다면, 매뉴얼 후반부의 장들에서 답을 찾을 수 있습니다.

2

제 1 장 — 메뉴 바

RadianceKit의 메뉴 바는 메인 윈도우나 인스펙터에 직접 노출되지 않은 모든 기능을 분류합니다. 여기에는 우선적으로 장면 전체에 작용하는 동작 (열기, 저장, 새 프로젝트), 학습 제어 (시작, 일시 정지, 계속), 뷰포트 조작 (자동 회전, 스크린샷, 배경색), 그리고 다양한 3D 형식 및 미디어 형식으로의 내보내기가 포함됩니다. 거기에 더해 모든 보조 창 (User Guide, Pareto Dashboard, Holdout Analysis, BayesOpt Console) 으로의 점프 포인트도 함께 모여 있습니다.

키보드 단축키는 각각 메뉴 항목 오른쪽에 표시됩니다. 규칙은 다음과 같습니다. ⌘ 는 Command 키 (애플 키) 를, ⇧ 는 Shift, ⌥ 는 Option (Alt), ⌘ 는 Control을 의미합니다. 예: ⌘⇧T 는 Shift+Command+T를 나타냅니다. 본 문서에 기록된 모든 단축키는 추가로 Help → Keyboard Shortcuts (⌘/) 의 별도 개요 창에서도 일람할 수 있습니다.

다음의 42개 항목은 인벤토리 순서 (M1–M42) 로 정리되어 있으며, 연관된 최상위 메뉴별로 그룹화되어 있습니다. 모든 항목은 현재 코드 상태 (175–477행) 와 대조하여 검증되었습니다. 인벤토리에 대비해 제거되거나 노후화된 항목은 없습니다. 새로 추가된 Edit 메뉴 항목 (Cmd-Z 의 “Remove Image”) 은 시스템의 NSUndoManager 프레임워크에 의해 자동으로 받아들여지므로 RadianceKitApp 코드에는 나타나지 않습니다 (장 마지막의 참고 사항 참조).

File 메뉴

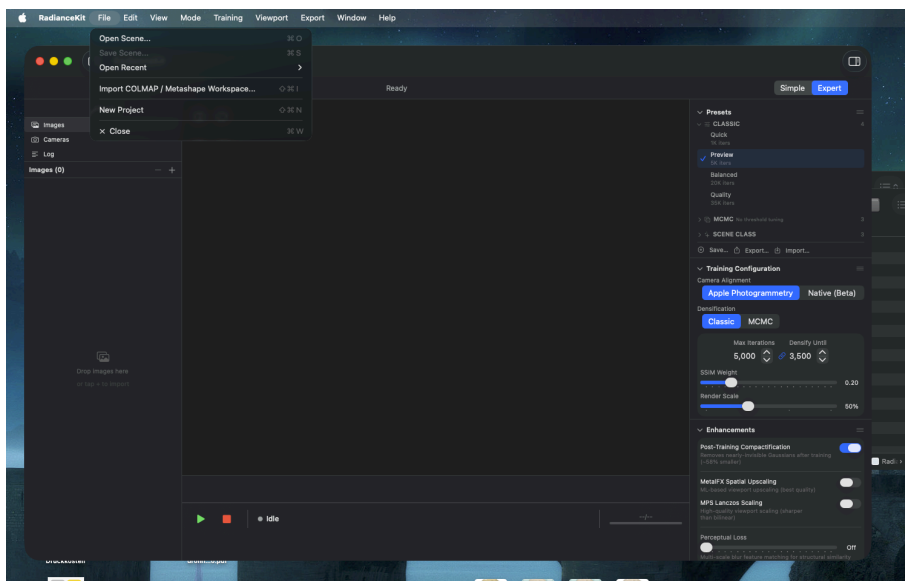


그림 1: File 메뉴 펼침 — 항목 M1부터 M6까지

File 메뉴는 Apple의 기본 "New Window" 항목을 프로젝트 관련 동작으로 대체합니다. 장면을 불러오기/저장하기, 동적 Recent 목록, Workspace 가져오기, 그리고 빈 상태로의 강제 초기화가 포함되어 있습니다.

M1 File > Open Scene...



메뉴 바 → File → Open Scene... (⌘O).



RadianceScene 번들, .ply, .splat, .spz 형식에 대한 파일 선택 대화창을 엽니다. 단일 선택이며, 파일과 디렉터리 (번들 형식의 경우) 를 모두 표시할 수 있습니다. 선택이 완료되면 경로가 Recent 목록에 등록되고 장면이 비동기적으로 로드됩니다 — 기존 장면은 대체되고 학습 파이프라인은 로드된 상태로 초기화됩니다. PLY/SPZ/Splat 파일은 각각의 형식 로더를 통해 읽고, .radiancecene 번들은 매니페스트, 클라우드 스냅샷, SfM 결과를 포함하는 디렉터리입니다.

간단히 말하면

?? ???? ???? ??.
?? ???? ???? ??.
RadianceKit ????, ??
Splatting ???? ????
?? ???? PLY, SPLAT, SPZ??
?????. ?: ?? ???? ?
????? ???? ???? ?
????? ???? ???? ?
??????. ?? ???? ?
????? ???? ???? ?
?, ?? ???? ???? ?
??????. ??? ????
? "Open Recent" (M3) ????
????? ???? ?
?????.

M2 File > Save Scene...



메뉴 바 → File → Save Scene... (⌘S).



Content-Type이 RadianceScene 번들이고 파일 이름이 scene.radiancecene으로 미리 채워진 파일 저장 대화창을 엽니다. manifest.json, 직렬화된 Gaussian Cloud (PLY 스냅샷), 그리고 SfM 결과 덤프를 포함하는 디렉터리 패키지를 작성하며, 다시 열었을 때 Continue Training도 동작하도록 합니다. Gaussian이 아직 존재하지 않는 동안에는 항목이 비활성화됩니다. 학습 로그 경로가 아니라 저장 대화창이 가리키는 위치 — 일반적으로 ~/Documents/ 아래 — 에 저장됩니다.

간단히 말하면

?? ???? ???? (???? ?
???? ???? ???? ??) ?
??????. ?? ???? ?
????? "Open Scene..." (M1) ?
???? ? ? ?????. ???
?? Gaussian Cloud? SfM ??
???? ???? ???? ?
Continue Training (M12-M14) ?
???? ? ? ?????. ???
?? ???? ? ? ???? ?
???? ???? ? ? ?????. ??
???? scene.radiancecene
??, ?? ???? ????
???? ???? ? ? ????.

M5 File > Import COLMAP / Metashape Workspace...



메뉴 바 → File → Import COLMAP / Metashape Workspace... (⇧⌨I).



폴더 선택기를 엽니다. COLMAP Workspace 레이아웃 (예: sparse/0/cameras.{bin,txt} 및 images/) 을 가진 폴더를 예상합니다. 선택 후 Workspace에 대한 사전 검사가 실행됩니다 — 이 검사는 세 가지 레이아웃 (sparse/0/, sparse/, 루트) 과, 재구성이 바이너리 (cameras.bin) 인지 ETH3D 텍스트 (cameras.txt) 인지를 판별합니다. 성공 시에는 Workspace를 가져오고, 실패 시에는 앱 로그에 경고만 표시됩니다. 자세한 파이프라인 로직은 제 9 장 "SfM 백엔드"의 Q6를 참고하십시오.

간단히 말하면

Metashape, COLMAP, RealityCapture ... RadianceKit SfM ... COLMAP (sparse/0/ cameras.* images/ ...) 9 "SfM ..."

M6 File > New Project



메뉴 바 → File → New Project (⇧⌨N).



저장되지 않은 작업이 있는지 확인합니다. 있는 경우, 어떤 것이 손실되기 전에 확인 대화창이 표시됩니다. 저장할 것이 없다면 초기화가 즉시 실행됩니다 — 가져온 이미지, SfM 결과, Gaussian Cloud, 학습 상태, 의존하는 모든 UI 표시 항목이 비워집니다. 주의: 사용자가 만든 프리셋 라이브러리는 앱 환경 설정에 저장되며 프로젝트 상태에는 속하지 않으므로 그대로 유지됩니다.

간단히 말하면

... — ... SfM, Gaussian Cloud, ...

Mode 메뉴

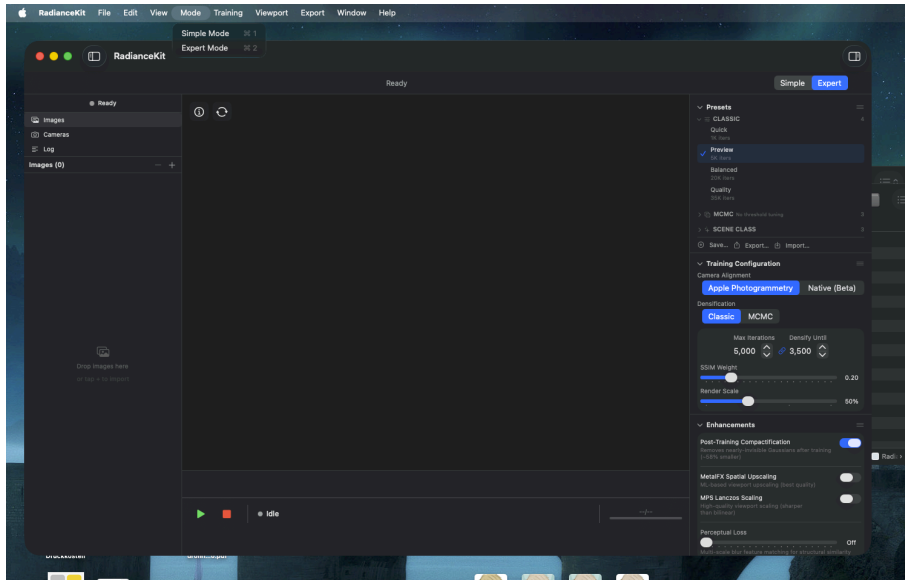


그림 2: Simple과 Expert 모드 스위치가 있는 Mode 메뉴

가이드형 Simple Mode (마법사 형식, 4단계) 와 모든 조절기가 있는 고전적 인스펙터 레이아웃의 전체 Expert Mode 사이를 전환하는 두 개의 간단한 스위치입니다.

M7 Mode > Simple Mode

위치

메뉴 바 → Mode → Simple Mode (⌘1).

기술적 설명

앱 상태를 Simple Mode로 전환합니다. 그러면 앱의 메인 영역에 Expert 레이아웃 대신 가이드형 워크플로가 표시됩니다. 모드 상태는 환경 설정에 저장됩니다 (제 3 장 Settings의 S1 "Default Mode" 참고).

간단히 말하면

? ? ? ? , ? ? , ? ? ? ? , ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? . ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? . ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 — ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? . ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? Expert Mode
 (M8) ? ? ? ? ? ? ? ? ? ? . ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? (? 3 ? , S1) ? ? ? ?
 ? ? ? ? ? ? .

M8 Mode > Expert Mode



메뉴 바 → Mode → Expert Mode (⌘2).



앱 상태를 Expert Mode로 전환합니다. 그 결과 모든 섹션 (Presets, TrainingConfig, Enhancements, Metrics, LossChart, ProjectNavigator) 을 포함하는 전체 인스펙터 레이아웃이 나타납니다. Expert Mode에서는 모든 학습 매개변수, COLMAP 선택기, Mid-Compact 토글, 진단 항목에 접근할 수 있습니다. 실시간 미리보기도 이 모드에서만 작동합니다.

간단히 말하면

?? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ?
 Loss ? ? ? ? ? ? , ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? , ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? , COLMAP ? ? ? , ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ?
 M7 ? ? ? Simple Mode ? ? ?
 ? ? ? ? ? — ? ? ? ? ? ? ? ?
 ? ? ? ? ? .

Training 메뉴

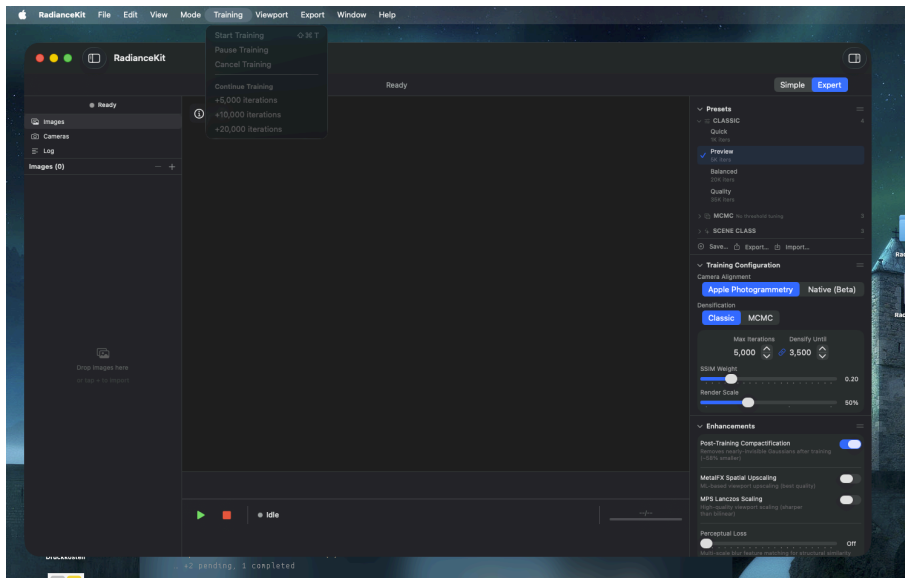


그림 3: Continue 서브 메뉴가 있는 Training 메뉴 — 항목 M9부터 M14까지

학습 실행과 관련된 네 가지 동작: 시작, 일시 정지, 취소, 그리고 정해진 반복 수만큼 연장. 세 개의 Continue 항목은 모두 IAP에 의해 게이트가 걸려 있습니다 (무료 체험판 버전에서는 클릭할 수 없음).

M9 Training > Start Training



메뉴 바 → Training → Start Training (⇧⌨T).



학습 파이프라인을 비동기적으로 시작합니다. 전제 조건: SfM 결과가 존재하고, 다른 파이프라인이 현재 실행 중이지 않아야 합니다. 두 조건 중 하나라도 만족하지 않으면 항목이 차단됩니다. 시작 시 현재 구성 값을 읽고, ~/Documents/RadianceKit/Logs/training_YYYY-MM-DD_HHm:ss.jsonl 경로에 새 JSONL 로그를 생성하며, 전략 선택에 따라 classic 경로 또는 MCMC 경로로 진행합니다. 학습 상태는 "idle"에서 "training"으로 전환됩니다.

간단히 말하면

?? ? ? ? ? ? ? ? — ?
?? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? Gaussian
Splatting ? ? ? ? ? ? ? ? . ?
?? ? ? ? ? ? ? ? ? ? . ? ?
?? ? ? ? ? 1? (Quick) ? ? ?
?? (MCMC Quality) ? ? ?
?? ? ? . SfM ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? . ? ? ? ? ? ? ? ?
? ~/Documents/RadianceKit/
Logs/? ? ? ? ? ? ? ? , ? ?
?? ? Pareto Dashboard (M40)
?? ? ? ? ? ? ? ? ? ? ? .

M10 Training > Pause Training



메뉴 바 → Training → Pause Training.



실행 중인 학습을 일시 정지합니다. 학습 상태가 "training"일 때에만 활성화됩니다. 일시 정지는 다음 안전 동기화 시점에서 반복 루프를 중단하고, 전체 GPU 상태 (Gaussian 버퍼, 옵티마이저 모멘트, 스케줄러 위치) 를 유지한 채 "paused"로 전환합니다. 재개는 동일 항목을 다시 누르면 됩니다 (항목 제목은 정적이며 — 앱은 내부적으로 실제로 Pause/Resume 사이를 전환합니다). 일시 정지된 학습은 앱 종료를 견디지 못합니다. 그 경우에는 장면을 저장한 뒤 나중에 Continue Training 항목 (M12-M14) 으로 연장하십시오.

간단히 말하면

?? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? . ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? . ? ? ? ? ? ?
?? ? ? ? ? ? . ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? — ? ? ?
?? ? ? ? ? ? ? ? ? ? ? Cancel
(M11) ? ? ? ? ? ? ? ? ? , Save
Scene (M2) ? ? ? ? ? ? ? ?
? Continue Training (M12-M14)
? ? ? ? ? ? ? ? . ? ? ? ? ?
? GPU? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? .

M13 Training > Continue Training > +10 000 iterations



메뉴 바 → Training → Continue Training → +10,000 iterations.



M12와 동일하나 추가 반복이 10 000회입니다. 동일한 전제 조건, 동일한 LR 스케줄러 경로입니다. 초기 학습을 중간 등급 프리셋으로 실행했고, 실행을 완전히 다시 시작하지 않고도 의미 있는 품질 향상을 보고 싶을 때 권장됩니다.

간단히 말하면

10 000 회 반복이 추가되어, 동일한 전제 조건, 동일한 LR 스케줄러 경로입니다. 초기 학습을 중간 등급 프리셋으로 실행했고, 실행을 완전히 다시 시작하지 않고도 의미 있는 품질 향상을 보고 싶을 때 권장됩니다.

M14 Training > Continue Training > +20 000 iterations



메뉴 바 → Training → Continue Training → +20,000 iterations.



M12 / M13과 동일하나 추가 반복이 20 000회입니다. 사전 정의된 Continue 점프 중 가장 큼니다. MCMC 학습에서는 종종 이것이 "그럭저럭 괜찮음"과 "벤치마크 등급" 사이를 가르는 차이를 만들어냅니다. 35-40K 이상의 Classic에서는 경험상 추가로 얻는 것이 많지 않습니다.

간단히 말하면

20 000 회 반복이 추가되어, 사전 정의된 Continue 점프 중 가장 큼니다. MCMC 학습에서는 종종 이것이 "그럭저럭 괜찮음"과 "벤치마크 등급" 사이를 가르는 차이를 만들어냅니다. 35-40K 이상의 Classic에서는 경험상 추가로 얻는 것이 많지 않습니다.

Viewport 메뉴

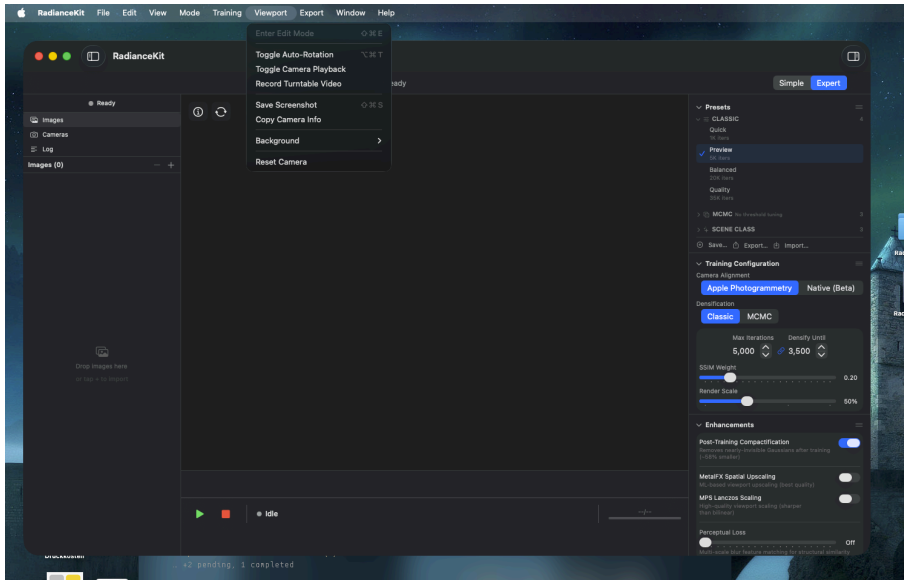


그림 4: Edit Mode, 카메라 제어, 배경 서브 메뉴가 있는 Viewport 메뉴

3D 뷰포트를 제어합니다. Gaussian 선택과 정리를 위한 Edit Mode, 카메라 제어 (자동 회전, 재생, 녹화), 스크린샷, 배경색, 그리고 리셋이 포함됩니다.

M15 Viewport > Enter/Exit Edit Mode

위치

메뉴 바 → Viewport → Enter Edit Mode (또는 상태에 따라 “Exit Edit Mode”). 순췡E.

기술적 설명

항목 제목은 동적이며 상태에 따라 “Exit Edit Mode” 또는 “Enter Edit Mode”를 표시합니다. 누르면 뷰포트 렌더러의 Edit Mode가 토글됩니다. Edit Mode를 빠져나갈 때에는 현재 선택이 추가로 초기화됩니다. Edit Mode는 Gaussian 클릭 선택, 박스 선택, 그리고 표시된 Gaussian의 삭제를 활성화합니다 (UI의 Editor 영역 참조). 뷰포트 렌더러가 연결되어 있지 않은 동안에는 비활성화됩니다.

간단히 말하면

?? 3D ??? Gaussian
 (? : ??? ? ? ? ? ? ? ?
 ?) ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ?
 ? — ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ?
 ?.


M16 Viewport > Toggle Auto-Rotation

 위치

메뉴 바 → Viewport → Toggle Auto-Rotation (⌘⇧T).

 기술적 설명

장면 중심을 통과하는 수직 축을 중심으로 한 뷰포트 카메라의 연속 회전을 켜거나 끕니다. 축과 속도는 카메라 제어 구성에서 나옵니다. 자동 회전은 순수한 뷰포트 효과이며 학습이나 녹화에 영향을 주지 않습니다 — Turntable 비디오 레코더 (M18) 를 병행 사용하는 경우 자동 회전은 레코더가 캡처할 정확한 경로를 제공합니다.

 간단히 말하면

?? ?? ?? ?? ?? ?? ?? ?? ??
?? ?? ?? ?? ?? ?? ?? ?? ??
?? ?? ?? ?? ?? ?? ?? ?? ??.
?? ?? ?? ?? ?? ?? ?? ?? ??
?. ?? ?? ?? ?? ?? ?? ?? ??
?? ?? ?? ?? ?? ?? ?? ?? ??
?? ?? ?? ?? ?? ?? ?? ?? ??
?? ?? ?? ?? ?? ?? ?? ?? ??
?? ?? ?? ?? ?? ?? ?? ?? ??
?? (M18) ?? ?? ?? ?? ?? ??
? ?? ?? ?? ?? ?? ?? ?? ??
?? ?? ?.


M17 Viewport > Toggle Camera Playback

 위치

메뉴 바 → Viewport → Toggle Camera Playback.

 기술적 설명

카메라 경로 재생을 토글합니다. 기록된 카메라 경로가 존재하는 경우 (예: 이전 녹화에서 또는 transforms.json 을 불러와 생성된 경로), 그 경로가 재생됩니다 — 즉 뷰포트 카메라는 더 이상 마우스/트랙패드 입력에 따라 움직이지 않고 프레임 단위로 궤적을 재현합니다. 다시 누르면 재생이 일시 정지됩니다.

 간단히 말하면

?? ?? ?? ?? ?? ?? ?? ?? ??
?? ?? ?? ?? ?? ?? ?? ?? ?
?? ?? ?? ?? ?? ?? ?? ?? ??
?? ?? ?? ?? ??, ?? ?? ?? ?? ??
?? ?? ?? ?? ?? ?? ?? ?? ??
?? ?? ?? ?? ?? ?? ?? ?? ??
?? ?? ?? ?? ?? ?? ?? ?? ??
?? ?? ?? ?? ?? ?? ?? ?? ?? — ??
?? ?? ?? ?? ?? ?? ?? ?? ??.
?? ?? ?? ?? ?? ?? ?? ?? ??
?? ?? ?? ?? ?? ?? ?? ?? ??
?? ?? ?? ?? ?? ?? ?? ?? ??
?? ?? ?? ?? ?? ?? ?? ?? ??
?? ?? ?? ?? ?? ?? ?? ?? ?.

M18 Viewport > Record Turntable Video



메뉴 바 → Viewport → Record Turntable Video.



뷰포트 녹화를 토글합니다. 처음 누르면 프레임 녹화가 임시 경로로 시작됩니다. 두 번째로 누르면 녹화가 종료되고 인코딩되어 MP4 경로로 저장됩니다 (경로는 저장 대화창으로 물어봅니다). 설정 가능한 시간의 고정 360° 경로를 생성하는 Export → Media → Orbit Video (M31) 와 달리, Turntable 레코더는 뷰포트에서 보이는 것을 실시간으로 기록합니다 — 즉 수동 카메라 이동도 녹화할 수 있습니다.

간단히 말하면

Turntable 레코더를 사용하여 뷰포트에서 보이는 것을 실시간으로 기록합니다. 이는 수동 카메라 이동도 포함하며, MP4 형식으로 인코딩된 비디오를 생성합니다. "Orbit Video" (M31)와 달리, Turntable 레코더는 뷰포트에서 보이는 것을 실시간으로 기록합니다. 즉, 수동 카메라 이동도 녹화할 수 있습니다.

M19 Viewport > Save Screenshot



메뉴 바 → Viewport → Save Screenshot (순췒S).



전체 렌더 해상도 (즉 윈도우 픽셀 레이아웃이 아닌 전체 렌더 타겟 내용) 의 뷰포트 프레임 하나를 PNG 파일로 캡처합니다. 경로는 저장 대화창으로 묻습니다. 배경색 (M21-M23) 은 함께 구워집니다. Enhancements (I27/I28 참조) 의 MetalFX/MPS 업스케일링 설정은 활성화된 경우 영향을 미칩니다 — 따라서 스크린샷은 업스케일된 출력을 보여 줍니다.

간단히 말하면

3D 뷰포트의 PNG 파일을 캡처합니다. 이는 배경색 (M21-M23) 을 포함하며, Enhancements (I27/I28 참조) 의 MetalFX/MPS 업스케일링 설정은 활성화된 경우 영향을 미칩니다. 따라서 스크린샷은 업스케일된 출력을 보여 줍니다.

M20 Viewport > Copy Camera Info

📍 위치

메뉴 바 → Viewport → Copy Camera Info.

🔧 기술적 설명

현재 뷰포트 카메라 포즈 (위치, Look-At 지점, Up 벡터) 와 FOV 값을 카메라 제어에서 읽어 여러 줄 텍스트로 클립보드에 기록합니다. 형식은 사람이 읽을 수 있도록 라벨 = 값 형태로 한 줄씩 표시되며, JSON은 아닙니다. 특정 보기를 디버깅용으로 재현하거나 지원팀과 공유할 때 편리합니다.

💬 간단히 말하면

?? ???? ???? ???? ????
?? ???? ???? ???? ????
?. ? : ???? ???? ???? ????
?? ???? ???? ???? ???? ????
?? ???? ???? ???? ???? ????
?? ???? ???? ???? ???? ????
???. ???? ???? ???? ????
? (?? ? ? ?) ???? ???? JSON
?? ?????. ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ?
??.

M21 Viewport > Background > Dark Gray

📍 위치

메뉴 바 → Viewport → Background → Dark Gray.

🔧 기술적 설명

뷰포트 배경색을 짙은 회색 (RGB 0.1/0.1/0.1) 으로 설정합니다. 렌더러는 이 색을 Gaussian이 합성되는 배경으로 사용합니다. 앱 시작 시의 기본 색상은 Settings 옵션 S3 "Default Viewport Background"가 제어합니다.

💬 간단히 말하면

3D ???? ???? ???? ????
?? ???? ?????. ???? ????
?? ? ? ? ? ? ? ? ? — ? ?
? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? Gaussian ?
?? ? Gaussian ? ? ? ? ? ?
?? ? ? ? ? ? ??. ? ? ? ? ? ?
? (M19) ? Orbit ? ? ? (M31) ?
? ? ? ? ? ??. Dark Gray ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? Black
(M22) ? ? ? White (M23) ? ? ?
? ? ? ? ? ??. ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? (S3) ? ?
?? ? ? ? ? ? ? ? ?.

M22 Viewport > Background > Black



메뉴 바 → Viewport → Background → Black.



뷰포트 배경색을 순수한 검정 (RGB 0/0/0) 으로 설정합니다. 장면에 밝은 플로터가 많고 식별하려는 경우, 또는 어둡고 분위기 있는 마케팅 자료에 적합합니다.

간단히 말하면

?? ? ? ? ? ? ? ? ? . ? ? ? ? ?
?? ? ? ? ? , Edit Mode ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ?
? Gaussian (? ? ?) ? ? ?
? ? ? ? ? ? ? ? ? ? . ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? . ? ? ? ? ? ? ? ? ?
Orbit ? ? ? ? ? ? ? ? ? ? — ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? .
? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
White (M23) ? ? ? ? ? ? ? ? ? .

M23 Viewport > Background > White



메뉴 바 → Viewport → Background → White.



뷰포트 배경색을 순수한 흰색 (RGB 1/1/1) 으로 설정합니다. 장면이 주로 어두운 내용으로 구성되어 있고 어두운 플로터 (전형적인 야외 배경 노이즈) 를 보고 싶을 때 유용합니다.

간단히 말하면

?? ? ? ? ? ? ? ? ? . ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? , ? ? Edit Mode
(M15) ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? . ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? . ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? .


M24 Viewport > Reset Camera

 위치

메뉴 바 → Viewport → Reset Camera.

 기술적 설명

뷰포트 카메라를 초기화하고, Training Camera 보기를 벗어나며, 자동 회전을 멈춥니다. 그 결과 카메라가 초기 위치 (일반적으로 장면 앞쪽, 약간 위에서 내려다보는 방향) 로 되돌아가고, 자동 회전은 꺼지며, 렌더러가 막 Training Camera (SfM 포즈 중 하나) 를 보여 주고 있었다면 Free Camera로 되돌아갑니다.

 간단히 말하면

????? ???? ???? ???? ?
 ??????. ???? ???? ?
 ?????? ???? ???? ?
 ?????? — ???? ???? ?
 ?????? ???? ???? ?
 ??????. ???? ???? ?
 ????? ???? ???? ?
 Training Camera ???? ?
 ?????? ???? ???? ?
 ?????? ???? ???? ?
 ??????.

Export 메뉴

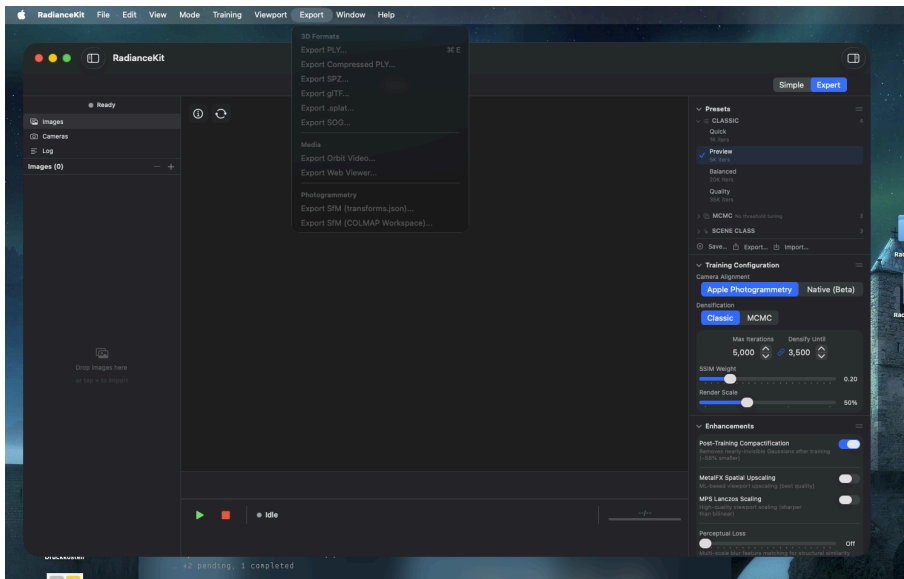


그림 5: 세 개의 서브 메뉴 그룹 (3D Formats, Media, Photogrammetry) 이 있는 Export 메뉴

여덟 가지 내보내기 대상과 두 가지 Photogrammetry 내보내기가 세 섹션 (3D Formats, Media, Photogrammetry) 으로 그룹화되어 있습니다. 처음 여섯 가지는 공통 헬퍼 루틴을 통해 구성되며, 각각 저장 대화창을 열고 형식 카탈로그에 내보내기를 등록합니다. Photogrammetry 항목은 개별 로직을 가집니다. 모든 Photogrammetry 항목과 일부 3D 내보내기는 정식 버전에서만 사용할 수 있습니다.

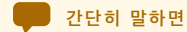
M25 Export > 3D Formats > Export PLY...



메뉴 바 → Export → 3D Formats → PLY (꺆E).



기본 파일 이름이 gaussians.ply 인 저장 대화창을 엽니다. 확인 시 현재 Gaussian Cloud가 표준 ASCII/Binary PLY 형식으로 기록됩니다 — SuperSplat, PolyCam, PlayCanvas 및 모든 일반적인 3DGS 뷰어와 호환됩니다. 완전한 SH 계수, 완전한 정밀도 (필드당 Float32). 파일 크기는 ≥ 500K Gaussian의 경우 종종 수백 MB에 이릅니다.



3D PLY — SuperSplat, PolyCam, PlayCanvas 및 모든 일반적인 3DGS 뷰어와 호환됩니다. 완전한 SH 계수, 완전한 정밀도 (필드당 Float32). 파일 크기는 ≥ 500K Gaussian의 경우 종종 수백 MB에 이릅니다. SPZ (M27) Compressed PLY (M26) —

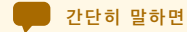
M26 Export > 3D Formats > Export Compressed PLY...



메뉴 바 → Export → 3D Formats → Compressed PLY.



위치, 스케일, 회전, SH 필드의 사용자 정의 양자화로 압축된 PLY 형식으로 Gaussian Cloud를 기록합니다. 시각적 손실이 최소인 상태에서 비압축 PLY (M25) 보다 5-10배 작은 파일입니다. Compressed PLY 표준을 읽는 SuperSplat 및 PlayCanvas와 호환됩니다. 기본 파일 이름은 gaussians_compressed.ply입니다.



PLY 5-10 PLY (M25) 보다 5-10배 작은 파일입니다. Compressed PLY 표준을 읽는 SuperSplat 및 PlayCanvas와 호환됩니다. 기본 파일 이름은 gaussians_compressed.ply입니다. SPZ (M27) Compressed PLY (M26) —

M27 Export > 3D Formats > Export SPZ...



위치

메뉴 바 → Export → 3D Formats → SPZ.



기술적 설명

Gaussian Cloud를 SPZ 형식으로 기록합니다 — Niantic 이 공개한 압축 Splat 형식으로 공격적인 양자화 (비압축 PLY 대비 약 90% 더 작음) 를 사용합니다. 특히 웹 뷰어와 모바일 앱에 최적화되어 있습니다. Niantic Splatt3R, gsplat.js, Niantic 브라우저 뷰어와 호환됩니다.



간단히 말하면

기억하세요, SPZ는 Niantic의 Splatt3R와 gsplat.js를 사용하여 생성된 3D 스plat의 압축 형식입니다. 이는 PLY보다 훨씬 작고, 웹 뷰어와 모바일 앱에서 더 빨리 로드됩니다. SPZ는 Niantic의 Splatt3R와 gsplat.js를 사용하여 생성된 3D 스plat의 압축 형식입니다. 이는 PLY보다 훨씬 작고, 웹 뷰어와 모바일 앱에서 더 빨리 로드됩니다.

M28 Export > 3D Formats > Export glTF...



위치

메뉴 바 → Export → 3D Formats → glTF.



기술적 설명

KHR_gaussian_splatting 확장이 포함된 .glb 파일 (Binary glTF) 을 기록합니다. 표준을 준수하며 KHR_gaussian_splatting 확장을 구현한 Babylon.js나 Three.js 같은 glTF 엔진을 사용하는 파이프라인에 적합합니다.



간단히 말하면

기억하세요, glTF는 3D 객체와 텍스처를 위한 표준화된 교환 형식입니다. KHR_gaussian_splatting 확장은 Gaussian Splatting을 glTF에 포함하는 데 사용됩니다. KHR_gaussian_splatting 확장을 구현한 Babylon.js나 Three.js 같은 glTF 엔진을 사용하는 파이프라인에 적합합니다. glb는 glTF의 바이너리 버전이며, KHR_gaussian_splatting 확장을 포함할 수 있습니다. glTF는 3D 객체와 텍스처를 위한 표준화된 교환 형식입니다. KHR_gaussian_splatting 확장은 Gaussian Splatting을 glTF에 포함하는 데 사용됩니다.

M29 Export > 3D Formats > Export .splat...



메뉴 바 → Export → 3D Formats → .splat.



Antimatter15의 .splat 형식을 기록합니다 — Gaussian 당 고정 32바이트 (위치 = 3× Float32, 스케일 = 3× Float32, 회전 = 4× Uint8 정규화 쿼터니언, RGB+Opacity = 4× Uint8). DC보다 높은 SH 계수는 없습니다. 브라우저 직접 호환성을 갖춘 최소 파일입니다. gsplat.js 및 antimatter15의 온라인 데모 뷰어용입니다.

간단히 말하면

Antimatter15의 .splat 형식을 기록합니다 — Gaussian 당 고정 32바이트 (위치 = 3× Float32, 스케일 = 3× Float32, 회전 = 4× Uint8 정규화 쿼터니언, RGB+Opacity = 4× Uint8). DC보다 높은 SH 계수는 없습니다. 브라우저 직접 호환성을 갖춘 최소 파일입니다. gsplat.js 및 antimatter15의 온라인 데모 뷰어용입니다.

M30 Export > 3D Formats > Export SOG...



메뉴 바 → Export → 3D Formats → SOG.



Gaussian Cloud를 SOG 형식으로 기록합니다. SOG ("Self-Organizing Gaussian")는 텍스처 아틀라스 레이아웃과 양자화된 데이터의 WebP 압축을 사용하는 PlayCanvas 형식입니다. PLY 대비 15–20배 더 좋은 크기 비율로 확장됩니다. 내보내기 시 내부적으로 외부 도구 cwebp를 호출합니다 — 따라서 Sandbox 버전 (App Store)에서는 잠재적으로 제한될 수 있습니다.

간단히 말하면

PlayCanvas는 Gaussian Cloud를 SOG 형식으로 기록합니다. SOG ("Self-Organizing Gaussian")는 텍스처 아틀라스 레이아웃과 양자화된 데이터의 WebP 압축을 사용하는 PlayCanvas 형식입니다. PLY 대비 15–20배 더 좋은 크기 비율로 확장됩니다. 내보내기 시 내부적으로 외부 도구 cwebp를 호출합니다 — 따라서 Sandbox 버전 (App Store)에서는 잠재적으로 제한될 수 있습니다.

M31 Export > Media > Export Orbit Video...



메뉴 바 → Export → Media → Orbit Video.



장면 중심을 도는 360° 궤도를 렌더링하여 MP4 (H.264) 또는 MOV (HEVC, 시스템 기본값에 따라) 로 인코딩합니다. M18 (실시간 녹화) 과 달리 경로는 여기서 고정됩니다 — 시간은 설정 또는 Simple Mode 내보내기 단계에서 선택합니다.

간단히 말하면

360° Orbit Video (M18) Record Turntable Video (M18) 360° Simple Mode H.264 MP4 HEVC MOV

M32 Export > Media > Export Web Viewer...



메뉴 바 → Export → Media → Web Viewer.



독립형 HTML 뷰어 (gsplat.js 기반) 와 Base64로 인코딩된 Gaussian 데이터를 단일 .html 파일로 패키징합니다. 이 파일은 모든 최신 브라우저에서 오프라인으로 실행됩니다 — 서버 의존성도, 외부 URL도 없습니다. 파일 크기는 Base64 오버헤드 때문에 SPZ 변형보다 약 1.3배 정도 큼니다.

간단히 말하면

3D Gaussian Splatting HTML Gaussian Splatting SPZ 3D 1

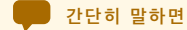
M33 Export > Photogrammetry > Export SfM (transforms.json)...



메뉴 바 → Export → Photogrammetry → Export SfM (transforms.json).



Gaussian Cloud가 아니라 SfM 결과를 내보내기 때문에 별도의 내보내기 경로 (공동 헬퍼 루틴이 아님) 입니다. 기본 값 transforms.json 및 Content-Type json 으로 저장 대화창을 엽니다. 확인 시 카메라 인트린식, 포즈 (NeRF 규약의 4x4 행렬), 프레임 경로를 포함하는 nerfstudio 호환 transforms.json 이 기록됩니다. UI의 도움말 텍스트는 학습 이미지가 자매 폴더 images/로 함께 복사되어야 한다고 안내합니다. SfM 결과가 존재하고 정식 버전이 잠금 해제된 경우에만 활성화됩니다.



SfM nerfstudio, Brush, gsplat OpenSplat images/ transforms.json images/ — SfM COLMAP Workspace M34

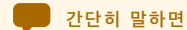
M34 Export > Photogrammetry > Export SfM (COLMAP Workspace)...



메뉴 바 → Export → Photogrammetry → Export SfM (COLMAP Workspace).



기본 이름이 colmap-workspace (폴더이므로 확장자 없음) 인 저장 대화창을 엽니다. sparse/0/ cameras.bin, images.bin, points3D.bin 을 포함하는 표준 COLMAP Workspace를 기록합니다. RadianceKit 에서 계산되었거나 가져온 SfM 재구성을 Postshot, Nerfstudio, Meshroom 같은 다른 도구에서 열거나, A/B 재 실행 시 미리 계산된 입력으로 RadianceKit 자체에서 (M5를 통해) 다시 로드하는 것을 허용합니다 — 계산 시간을 절약합니다. SfM 결과가 존재하고 정식 버전이 잠금 해제된 경우에만 활성화됩니다.



M33 nerfstudio COLMAP Postshot, Meshroom, Nerfstudio COLMAP sparse/0/ cameras.bin: images.bin M5 RadianceKit SfM COLMAP Workspace M34

Help 메뉴

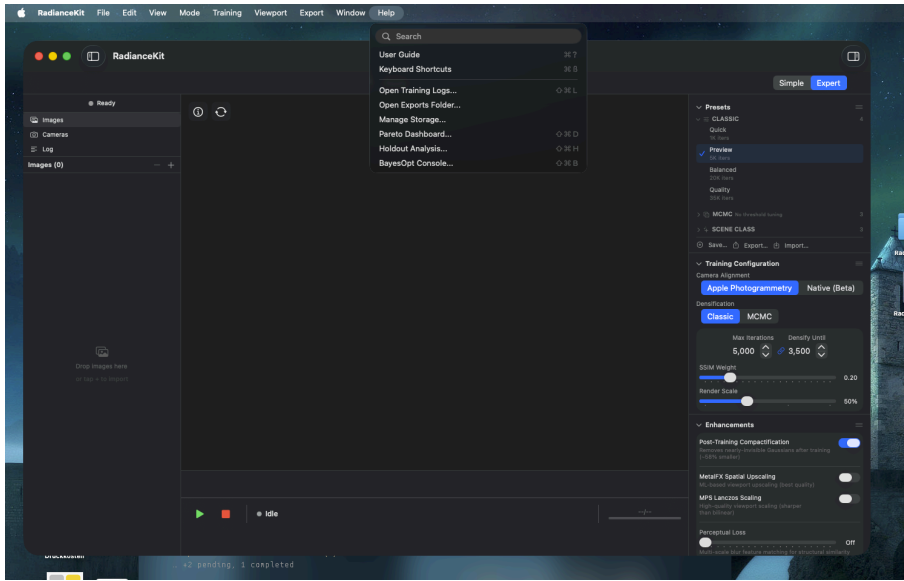


그림 6: 문서, 폴더, 분석 항목이 있는 Help 메뉴

일곱 항목: 두 문서 창 (User Guide, Keyboard Shortcuts), 세 폴더 바로 가기 (Training Logs, Exports, Storage), 그리고 세 분석 창 (Pareto Dashboard, Holdout Analysis, BayesOpt Console). Apple 스타일에 따라 Help 메뉴는 가장 오른쪽에 표시됩니다. 기본 Help 메뉴는 완전히 RadianceKit 고유 변형으로 대체됩니다.

M35 Help > User Guide

위치

메뉴 바 → Help → User Guide (꺾?).

기술적 설명

User Guide 창을 엽니다. 주제 사이드바와 스크롤 가능 세부 영역이 있는 내비게이션을 기본 크기 860x640으로 표시합니다. 콘텐츠는 정적으로 내장되어 있으며 (Markdown에서 파싱되지 않음).

간단히 말하면

?????????????. ??
 ??????????????????
 ??????????????????
 ??????????????????
 ??????????????????
 ??????????????????
 ??????????????????
 ??????????????????
 ??????????????????
 ??????????????????
 ??????????????????
 ??????????????????

M36 Help > Keyboard Shortcuts



메뉴 바 → Help → Keyboard Shortcuts (⌘/).



Keyboard Shortcuts 창을 엽니다 — 모든 앱 단축키가 최상위 메뉴별로 그룹화된 단순 스크롤 레이아웃입니다. 기본 크기 440x560. 콘텐츠도 정적으로 내장되어 있습니다.

간단히 말하면

Keyboard Shortcuts 창을 엽니다 — 모든 앱 단축키가 최상위 메뉴별로 그룹화된 단순 스크롤 레이아웃입니다. 기본 크기 440x560. 콘텐츠도 정적으로 내장되어 있습니다.

M37 Help > Open Training Logs...



메뉴 바 → Help → Open Training Logs... (⌘⌘L).



로그 폴더를 ~/Documents/RadianceKit/Logs로 계산하고, 필요한 경우 생성한 뒤 Finder에서 엽니다. 각 학습 실행은 거기에 고유한 JSONL 파일 training_YYYY-MM-DD_HHmss.jsonl을 기록합니다.

간단히 말하면

로그 폴더를 ~/Documents/RadianceKit/Logs로 계산하고, 필요한 경우 생성한 뒤 Finder에서 엽니다. 각 학습 실행은 거기에 고유한 JSONL 파일 training_YYYY-MM-DD_HHmss.jsonl을 기록합니다.

M38 Help > Open Exports Folder...



위치

메뉴 바 → Help → Open Exports Folder...



기술적 설명

M37과 유사하지만 ~/Documents/RadianceKit/Exports 를 사용합니다. 첫 번째 자동 테스트 실행 또는 첫 번째 클릭 시 생성됩니다. 이후 모든 자동 테스트 내보내기 (예: autotest_ <timestamp>.ply) 의 기본 경로가 거기에 저장됩니다. 저장 대화창을 통해 수동으로 선택된 내보내기는 반드시 여기에 들어 가지 않으며 사용자가 저장하는 위치에 저장됩니다 — 따라서 이 폴더는 주로 자동 테스트에 관심이 있습니다.



간단히 말하면

??? (???)
 (???)
 .???

M39 Help > Manage Storage...



위치

메뉴 바 → Help → Manage Storage...



기술적 설명

Storage 브라우저를 엽니다 (제 4 장 Auxiliary Windows의 ID W7–W12 참조). ~/Documents/RadianceKit/ 폴더의 모든 영구 장면, 학습 로그, 내보내기, 캐시를 크기와 함께 나열하고, 각 항목에 대해 Reveal-in-Finder와 휴지통 이동을 가능하게 합니다.



간단히 말하면

RadianceKit???,
 ???, ???

M40 Help > Pareto Dashboard...



메뉴 바 → Help → Pareto Dashboard... (순쵸D).



Pareto Dashboard를 엽니다 (제 4 장 ID W13-W22 참조). 대시보드는 ~/Documents/RadianceKit/Logs/에 있는 모든 JSONL 학습 로그를 로드하고, 장면과 프리셋별로 정렬한 뒤 Pareto 산점도 (기본값: Loss vs Gaussians, 선택적으로 Loss vs Wallclock 또는 PSNR vs Iterations) 를 그립니다.

간단히 말하면

Placeholder text for the summary of the Pareto Dashboard help page.

M41 Help > Holdout Analysis...



메뉴 바 → Help → Holdout Analysis... (순쵸H).



Holdout 분석 창을 엽니다 (제 4 장 ID W23-W29 참조). transforms.json 을 로드하고, 카메라를 3D 글로브로 그리며, Train/Test 폴드 분할 (각도 또는 선형, 2-8 폴드) 을 허용합니다. 출력은 fold-assignment.json이며, 학습은 해당 학습 구성에서 이를 테스트 세트로 사용할 수 있습니다.

간단히 말하면

Placeholder text for the summary of the Holdout Analysis help page.

M42 Help > BayesOpt Console...



메뉴 바 → Help → BayesOpt Console... (순열B).



BayesOpt 콘솔을 엽니다 (제 4 장 ID W30-W39 참조). 사전 정의된 탐색 공간 (예: "MCMC scale-reg + opacity-reg + ssim") 을 로드하고, Bayesian Optimization 시도를 비동기적으로 실행하며, 수렴 곡선과 시도 로그를 실시간으로 표시합니다.

간단히 말하면

?? ? ? ? ? ? ? ? ? ? ? . ?
?? ? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ? ?

참고: Edit 메뉴의 Cmd-Z

2026년 5월 이후 Expert Mode의 Project Navigator는 마이너스 버튼 또는 Backspace 키로 가져온 이미지의 삭제, 그리고 Cmd-Z 를 통한 되돌리기를 지원합니다. 이 Cmd-Z 동작은 삭제된 이미지를 복원할 수 있는 동안에는 "Undo Remove Image"로 macOS Edit 메뉴 (SwiftUI에서 제공) 에 나타납니다. 표준 NSUndoManager 시스템을 통해 등록되며, RadianceKitApp에는 등록되지 않습니다. 따라서 인벤토리에 별도의 M-ID 항목이 없습니다.

키보드 단축키 개요

메뉴 항목	단축키
File > Open Scene...	⌘O
File > Save Scene...	⌘S
File > Import COLMAP / Metashape Workspace...	⇧⌘I
File > New Project	⇧⌘N
Mode > Simple Mode	⌘1
Mode > Expert Mode	⌘2
Training > Start Training	⇧⌘T
Viewport > Enter/Exit Edit Mode	⇧⌘E
Viewport > Toggle Auto-Rotation	⌘⇧T
Viewport > Save Screenshot	⇧⌘S
Export > 3D Formats > PLY	⌘E
Help > User Guide	⌘?
Help > Keyboard Shortcuts	⌘/
Help > Open Training Logs...	⇧⌘L
Help > Pareto Dashboard...	⇧⌘D
Help > Holdout Analysis...	⇧⌘H
Help > BayesOpt Console...	⇧⌘B

Edit 메뉴 (시스템 제공, Expert Mode에서 Project Navigator 선택이 활성화일 때):

동작	단축키
Undo Remove Image	⌘Z
Remove Selected Image	Backspace / Delete

2

제 2 장 — 인스펙터 (Expert View)

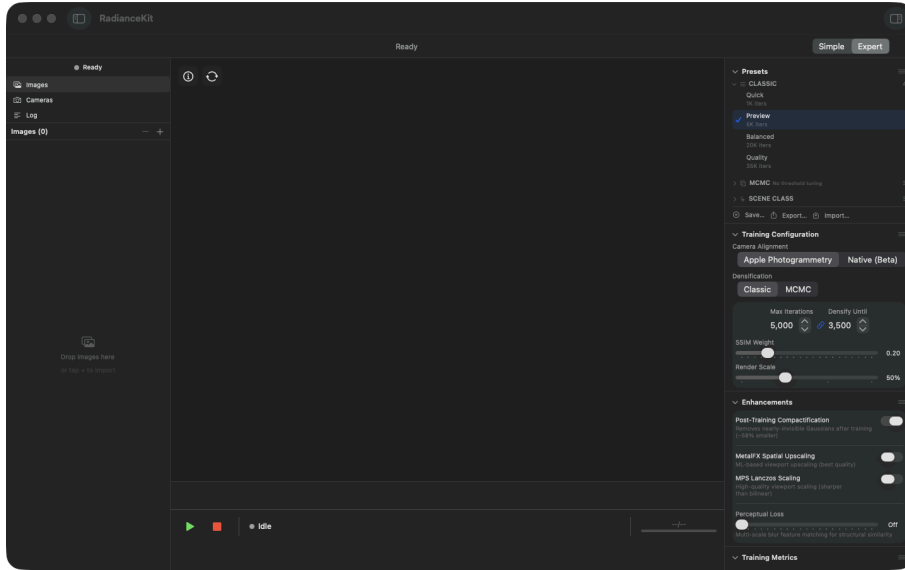


그림 7: Expert 모드 비어 있음 — 왼쪽에 Project Navigator (Images 0, Cameras, Log), 가운데 빈 뷰 포트, 오른쪽 인스펙터에 Presets/Training Configuration/Enhancements/Training Metrics 섹션

가져오기 전의 빈 인스펙터: 왼쪽 사이드바에 Images 카운터 0과 드롭 힌트 “Drop images here / or tap + to import”가 표시됩니다. 오른쪽 인스펙터는 완전히 작동하지만 프리셋은 정보용일 뿐입니다 (활성 학습 없음). 기본 프리셋 “Preview” (5K iters) 가 표시되어 있습니다. Camera Alignment는 Apple Photogrammetry, Denification은 Classic, SSIM Weight는 0.20, Render Scale은 50%입니다. Training Metrics (“Start training to see live metrics”) 와 Loss History (“Loss curve will appear during training”) 의 비어 있는 상태도 표시됩니다.

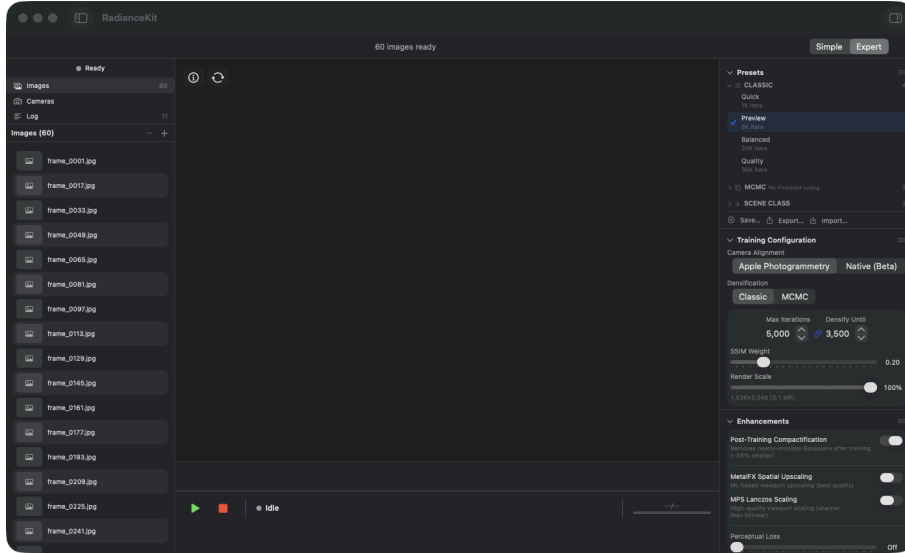


그림 8: 60개의 flowers 이미지를 로드한 인스펙터 — 이미지 사이드바가 frame_0001.jpg 이후의 파일 이름을 보여 주고 헤더에 “60 images ready”

가져오기 후의 인스펙터: 헤더 상태 “60 images ready”. 이미지 사이드바는 60개의 가져온 프레임 (frame_0001.jpg 에서 frame_0945.jpg 까지, 빠른 반복을 위한 부분 집합으로 960-Cam Bouquet 데이터셋의 16번째 프레임마다 하나) 을 모두 나열합니다. 자동 Render Scale 로직은 이미지 해상도 ($1536 \times 2048 = 3.1 \text{ MP}$) 를 검사하여 Render Scale을 그에 맞게 조정합니다. Play 버튼 (왼쪽 아래 녹색) 이 이제 활성화되어 있고 활성 프리셋으로 학습을 시작합니다.

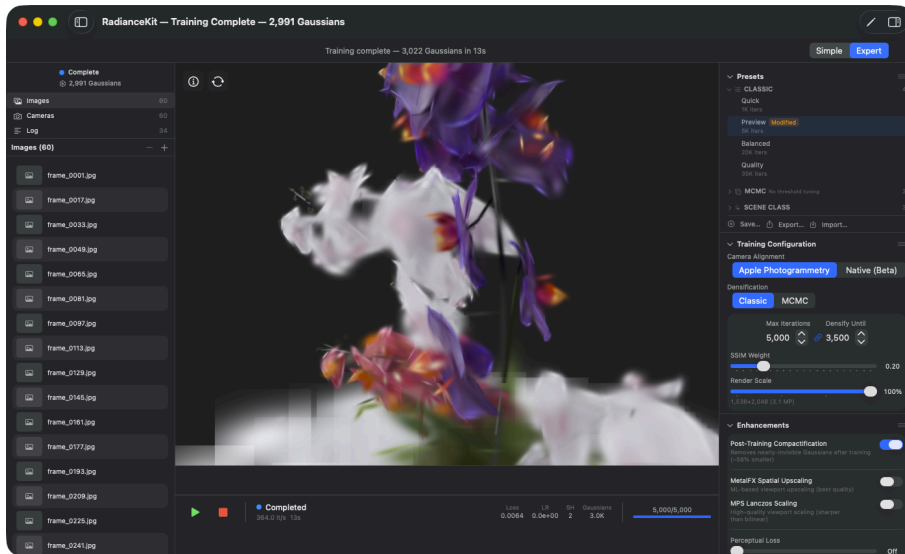


그림 9: 학습 중인 인스펙터 — 실시간 뷰포트가 flowers-Bouquet 재구성을 보여 줌, 하단에 메트릭 바 (Loss / LR / Gaussian-Count / Iterations), 매개변수가 조정된 경우 “Modified” 배지가 붙은 프리셋 카드 “Preview”

학습 중인 인스펙터: 제목 표시줄에 전체 진행률 “RadianceKit — Training NN %”가 표시됩니다. 뷰포트는 실행 중인 Gaussian 재구성을 실시간으로 렌더링합니다 (50회 반복마다 업데이트됨 — Live Preview 간격은 Settings → General → Training → Live Preview에서 설정 가능). 뷰포트 아래의 메트릭 바: 현재 Loss, Learning Rate, Gaussian Count, 그리고 반복 카운터 (예: Preview 프리셋의 경우 1,600/5,000). 인스펙터 프리셋 카드 “Preview”는 어떤 매개변수든 내장 기본값에서 벗어나면 “Modified” 배지를 표시합니다. 사이드바 “Log”는 SfM 및 학습 단계 이벤트를 수집합니다.

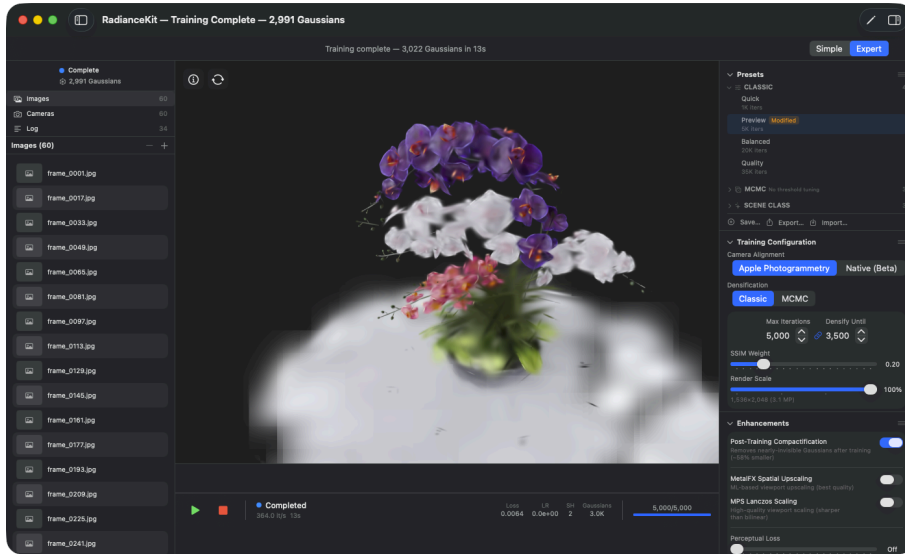


그림 10: 학습 완료 후 인스펙터 — 뷰포트가 완성된 flowers-Bouquet 재구성을 보여 줌 (5K 반복 후 2,991 Gaussians을 13초에 완료), 제목 표시줄 “Training Complete — 2,991 Gaussians”

학습 후의 인스펙터: 제목 표시줄에 최종 Gaussian 수 (여기서는 2 991 — 합성 Blender Bouquet 장면이 밝은 배경에 단순한 형상을 갖고 있어 매우 작음) 가 표시됩니다. 뷰포트는 완성된 점 구름을 보여 줍니다 — 궤도형 드래그 내비게이션 활성화 (장면 중심을 회전축으로). Training Metrics 섹션은 이제 최종 값들로 채워져 있고, Loss History 차트는 5 000회 반복 전체의 진행을 보여 줍니다. 아래의 Export 섹션이 이제 활성화되어 있습니다 (모든 형식 버튼이 활성화).

인스펙터는 Expert Mode (⌘2) 의 오른쪽 사이드바입니다. 학습 관련 모든 매개변수를 일곱 개의 접을 수 있는 섹션으로 묶어 둡니다. 첫 실행 시 위에서 아래로의 기본 순서는 다음과 같습니다: Look, Presets, Training config, Metrics, Loss chart, Enhancements, Export. “Look” 섹션 (post-Training 이미지 조정) 은 이전 “Finishing” 섹션의 실제 UI 이름 변경입니다 — 내부 Enum- rawValue 는 영속성 이유로 “Finishing” 으로 유지되지만, 표시되는 헤딩은 “Look”입니다. 각 섹션은 헤더를 클릭하면 접을 수 있고, 순서는 드래그 앤 드롭으로 재배열할 수 있습니다 (InspectorView.swift:81-97). 첫 실행 시 일곱 개 섹션이 모두 접혀 있습니다 (inspectorCollapsedSections의 기본값은 Set(InspectorSection.allCases)); 그 후 앱 상태는 접기 및 순서 설정을 앱 시작 간에도 유지합니다.

인스펙터의 일부 조작 요소는 거의 동일한 형태로 설정 (제 3 장) 에서도 나타납니다 — 일반적으로 SfM 백엔드, Sky Masking과 유사한 기본값들입니다. 분리는 의도적입니다. 설정은 새로 만든 프로젝트에 대한 앱 전역 템플릿을 제공하고, 인스펙터는 현재 열려 있는 프로젝트에 대해 이 값들을 재정의합니다. 한쪽의 조작 논리를 알면 다른 쪽도 익숙하게 사용할 수 있습니다.

Expert Mode의 왼쪽 열 — Project Navigator — 은 인스펙터에 속하지 않지만 바로 옆 이웃입니다. 거기서는 가져온 이미지를 클릭하여 선택하고, 스페이스 바로 Quick Look으로 미리 보고, 마이너스 버튼이나 Delete 키로 삭제할 수 있습니다 (Cmd-Z로 되돌리기). 인스펙터는 현재 사이드바 선택을 따라 컨텍스트 별 세부 정보를 표시하지만, 일곱 개의 주요 섹션은 항상 사용 가능한 상태로 유지됩니다.

Look 섹션 (L1-L5)

Look 섹션 (내부 rawValue 는 여전히 “Finishing”) 은 인스펙터의 최상단 섹션으로 **post-Training** 이미지 조정을 한곳에 모읍니다. 모든 슬라이더는 **비파괴적**으로 작동합니다. 각 슬라이더는 변경되지 않은

Pristine 스냅샷 (원본 DC 색상, Opacity, 스케일링) 에 `FinishingPass` 를 다시 적용합니다 — 따라서 조정은 누적되지 않고 **역동적**입니다. 결과는 **뷰포트에 실시간으로** 나타나며 (WYSIWYG, 이후 내보내기 와 정확히 동일) **모든 내보내기에 구워 넣어집니다**. 이 섹션은 **학습 실행이 완료된 후에야** 사용할 수 있습니다 (그 전에는 “Available after a training run completes.” 가 표시됨); 값은 **새 학습마다 재설정됩니다**. 내보내기가 실행되는 동안에는 모든 슬라이더가 **잠깁니다** — 잠금 힌트 “Locked while exporting — the file uses the current settings.” 가 나타나고 GroupBox가 비활성화됩니다.

L1 Saturation [?] [?] [?] [?]



위치

인스펙터 → Look 섹션 → GroupBox → Saturation.

기술적 설명

슬라이더 0.5–1.2, 두 자리 표시 (예: “1.00”). 각 Splat의 SH-DC 채도를 휘도 값에 대해 스케일합니다: 1.0 = 변경 없음, < 1.0 = 채도 감소 (색상이 그레이스케일로 당겨짐), > 1.0 = 더 강렬. 수학적으로 DC 색상은 Pristine 스냅샷에서 역산되므로 (`desaturateDC`) 반복적으로 밀어도 누적되지 않습니다. DJI 드론 소재 (Pensford Viadukt) 에서 검증되었는데, 이 소재는 과도하게 채색되는 경향이 있어 드론 기본값이 0.82 입니다. 색상 베이스 (SH 차수 0) 에만 작용하고 더 높은 SH 계수는 영향을 받지 않습니다.

간단히 말하면

[?] [?] [?] Splat [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?]. 1.00 [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]

L2 Splat length [?] [?] [?] [?]



위치

인스펙터 → Look 섹션 → GroupBox → Splat length.

기술적 설명

슬라이더 0.3–1.0, 두 자리 표시. 각 Gaussian의 세 스케일링 축을 로그 공간에서 평균값 쪽으로 당깁니다 (`shortenScale`, 계수 `alpha`): 1.0 = 변경 없음, 작은 값은 길쭉한 “바늘” Splat을 더 둥글게 만들고, 0이면 순수한 구가 됩니다. 전체 크기를 바꾸지 않고 바늘 모양의 과도하게 늘어진 Splat을 손보아 전형적인 “콘페티” 아티팩트를 줄입니다. Pristine 스냅샷 (원본 로그 스케일링) 에서 적용되므로 역동적입니다. 둘 다 로그 공간에서 작동하므로 Splat size (L3) 와 교환 가능합니다.

간단히 말하면

[?] [?] [?] [?] [?] [?] [?] Splat [?]
[?] [?] [?] [?] [?] [?] [?]. 1.00 [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]

L3 Splat size

위치

인스펙터 → Look 섹션 → GroupBox → Splat size.

기술적 설명

슬라이더 0.5–2.0, 두 자리 표시. 각 Gaussian을 세 축 모두에서 균일하게 스케일합니다 (`sizeScale`): 1.0 = 변경 없음, < 1.0 = 더 작고/조밀하고/선명, > 1.0 = 더 크고/“폭신” (Splat 사이의 틈을 채움). 스케일링이 로그 공간에 있으므로 곱셈은 가산적 `log(factor)` 오프셋으로 실현됩니다 — 상수 오프셋이 평균으로부터의 편차를 건드리지 않기 때문에 Splat length (L2) 와 교환 가능합니다. Pristine 스냅샷에서 적용되므로 먹등적입니다. 이 버전에서 새로 추가됨.

간단히 말하면

Splat size는 Gaussian의 세 축 모두에서 균일하게 스케일합니다. 1.0은 변경 없음, < 1.0은 더 작고/조밀하고/선명, > 1.0은 더 크고/“폭신” (Splat 사이의 틈을 채움). 스케일링이 로그 공간에 있으므로 곱셈은 가산적 `log(factor)` 오프셋으로 실현됩니다. 상수 오프셋이 평균으로부터의 편차를 건드리지 않기 때문에 Splat length (L2) 와 교환 가능합니다. Pristine 스냅샷에서 적용되므로 먹등적입니다. 이 버전에서 새로 추가됨.

L4 Fade far region

위치

인스펙터 → Look 섹션 → GroupBox → 토글 “Fade far region” 및 서브 슬라이더 “Fade start xradius” 와 “Fade floor”.

기술적 설명

카메라 중심으로부터의 거리에 따른 방사형 Opacity 감쇠를 활성화하는 토글입니다 — 배경에서 약하게 관측된 “Far 콘페티”가 흐려집니다. **궤도 촬영 전용:** `finishingContext.fadeEligible`가 `false`이면 (선형 비행, 카메라가 너무 적거나 퇴화) 토글이 비활성화되고, 서브 슬라이더 대신 “Far-fade applies only to orbit captures (not this scene).” 라는 힌트가 나타납니다. 적합성은 카메라 위치의 방위각 커버리지로 판정됩니다 (궤도는 중심을 돌며 많은 나침반 섹터를 채우고, 선형 비행은 ~2개만 채움). 두 서브 슬라이더가 형상을 제어합니다: **Fade start xradius** (1.0–3.0) 는 완전 Opacity가 적용되는 내부 반경을 궤도 반경의 배수로 설정하고; **Fade floor** (0.0–1.0) 는 Fade 반경 너머 멀리서의 Opacity 계수입니다. 중요: Fade는 **Sky-Dome 영역을 건너뛸니다** (인덱스 [0, frozenCount) 의 frozen Gaussian), 의도적으로 재구성된 배경 돔이 함께 어두워지지 않도록.

간단히 말하면

Fade far region은 거리에 따른 방사형 Opacity 감쇠를 활성화하는 토글입니다. “Far 콘페티”가 흐려집니다. 궤도 촬영 전용: `finishingContext.fadeEligible`가 `false`이면 (선형 비행, 카메라가 너무 적거나 퇴화) 토글이 비활성화되고, 서브 슬라이더 대신 “Far-fade applies only to orbit captures (not this scene).” 라는 힌트가 나타납니다. 적합성은 카메라 위치의 방위각 커버리지로 판정됩니다 (궤도는 중심을 돌며 많은 나침반 섹터를 채우고, 선형 비행은 ~2개만 채움). 두 서브 슬라이더가 형상을 제어합니다: Fade start xradius (1.0–3.0) 는 완전 Opacity가 적용되는 내부 반경을 궤도 반경의 배수로 설정하고; Fade floor (0.0–1.0) 는 Fade 반경 너머 멀리서의 Opacity 계수입니다. 중요: Fade는 Sky-Dome 영역을 건너뛸니다 (인덱스 [0, frozenCount) 의 frozen Gaussian), 의도적으로 재구성된 배경 돔이 함께 어두워지지 않도록.

L5 Reset finishing [?][?]

위치

인스펙터 → Look 섹션 → GroupBox → “Reset finishing” (하단, 작은 버튼).

기술적 설명

모든 Look 설정을 기본값으로 되돌리고 (FinishingPass.Settings() = Saturation 1.0, Fade 꺼짐, Splat length 1.0, Splat size 1.0) 즉시 Finishing 을 다시 트리거하여 뷰포트가 변경되지 않은 학습 상태로 되돌아가게 합니다. `controlSize(.small)`. 전체 Look 스택이 Pristine 스냅샷에서 역등적으로 계산하므로 “기본값으로 되돌리기”는 정확히 원래 학습 출력입니다 — 여러 번 앞뒤로 해도 품질 손실이 없습니다. 섹션의 모든 슬라이더와 마찬가지로 내보내기가 실행되는 동안 잠깁니다.

간단히 말하면

[?][?][?] [?][?][?] [?][?] Look [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 (Saturation 1.00, Fade [?][?], [?][?]) — [?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]

Presets 섹션 (I1–I11)

Presets 섹션은 검증된 구성을 가장 빠르게 적용하는 방법입니다. 내장 프리셋 (Capture Class, Classic, MCMC, Hybrid) 은 560+ 개의 문서화된 실험에서 나온 재현 가능한 시작점을 제공하고, 자체 프리셋은 저장, 내보내기, 가져오기, 공유할 수 있습니다. 목록은 카테고리 (Capture Class, Classic, MCMC, Hybrid, Custom) 별로 그룹화되어 있으며, 한 번에 둘 이상의 카테고리를 펼칠 수 있습니다. 컨텍스트 메뉴 (행 마우스 오른쪽 클릭) 를 통해 Export, Duplicate, 그리고 — 자체 프리셋의 경우 — Delete에 접근할 수 있습니다.

I3 Cancel (Save)

위치

Save 팝오버 → Cancel 버튼 (왼쪽).

기술적 설명

저장하지 않고 팝오버를 닫습니다. 텍스트 필드 내용을 폐기합니다 — 다음에 열 때 Save... 버튼 로직 (I1) 을 통해 다시 비워집니다. 표준 버튼 스타일, 확인 대화창 없음, 단축키 없음. 저장 경로가 실행되지 않았기 때문에 현재 TrainingConfig는 변경되지 않습니다.

간단히 말하면

Save 버튼을 클릭하면, 텍스트 필드 내용이 비워집니다. 다음에 열 때 Save... 버튼을 클릭하면, TrainingConfig가 저장됩니다. 현재 TrainingConfig는 변경되지 않습니다.

I4 Save (Save)

위치

Save 팝오버 → Save 버튼 (오른쪽, 강조된 스타일).

기술적 설명

실제 영구화를 트리거합니다. 비어 있지 않은 이름을 다시 한 번 검증하고 (방어적 점검) 현재 TrainingConfig를 JSON으로 앱 저장소에 기록합니다. 그 후 팝오버를 닫습니다. 파란색으로 강조되며, 텍스트 필드가 비어 있는 동안에는 회색입니다. 저장이 실패할 경우 (예: 앱 저장소가 가득 차서 — 매우 드뭄) 현재 표시되는 오류 대화창은 없습니다. 그 경우 프리셋은 다음 앱 시작 시 단순히 나타나지 않게 됩니다.

간단히 말하면

Save 버튼을 클릭하면, TrainingConfig가 저장됩니다. 현재 TrainingConfig는 변경되지 않습니다. Custom 버튼을 클릭하면, TrainingConfig가 저장됩니다. 현재 TrainingConfig는 변경되지 않습니다. (borderedProminent), Custom 버튼을 클릭하면, TrainingConfig가 저장됩니다. 현재 TrainingConfig는 변경되지 않습니다. (?: UserDefaults) 버튼을 클릭하면, TrainingConfig가 저장됩니다. 현재 TrainingConfig는 변경되지 않습니다. — Custom 버튼을 클릭하면, TrainingConfig가 저장됩니다. 현재 TrainingConfig는 변경되지 않습니다.

I5 Export... ????



인스펙터 → Presets 섹션 → 동작 표시줄 → Export... 버튼.



현재 선택된 프리셋을 `.radiancepreset` 파일 (내부는 JSON) 로 내보냅니다. 프리셋이 선택되지 않았다면 비활성화됩니다. 클릭 시 앱은 미리 정해진 파일 이름 (프리셋 이름 + `.radiancepreset` 확장자) 이 있는 저장 대화창을 엽니다. 저장된 형식은 전체 TrainingConfig와 메타데이터 (이름, 카테고리, ID, 내장 플래그) 를 포함합니다. Finder에서 더블 클릭하면 앱이 열리지만 **자동으로** 가져오기를 실행하지는 않습니다. 사용자는 가져오기 버튼 (I6) 을 사용해야 합니다.

간단히 말하면

```

?? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
Export ? ? ? ? ? ? ? ? — ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? : ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? Mac ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? Import... ?
? ? ? (I6) ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? Custom ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? : ? ? ? ? ? ? ? ? (I8) ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? — ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
  
```

I6 Import... ????



인스펙터 → Presets 섹션 → 동작 표시줄 → Import... 버튼.



`.radiancepreset` 파일만 허용하는 파일 대화창을 엽니다 (다중 선택 비활성화). 선택 시 JSON 파일이 로드되고 검증되어 Custom 카테고리에 삽입됩니다 — 내장 프리셋과의 충돌을 방지하기 위해 새 내부 ID로 등록됩니다. 가져오기는 내보낸 프리셋이 원래 예: 내장이었더라도 카테고리를 자동으로 Custom 으로 설정합니다. 손상된 파일이나 이전 스키마 버전과 호환되지 않는 파일은 오류 대화창 없이 조용히 거부됩니다 (콘솔 로그에는 정보가 제공됨).

간단히 말하면

```

? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
Mac ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
Custom ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
— ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
  
```

I7 (???) (???)

 위치

인스펙터 → Presets 섹션 → 각 카테고리의 각 프리셋 행.

 기술적 설명

프리셋 행을 클릭하면 TrainingConfig의 모든 필드가 프리셋의 값으로 대체되고, 활성화 프리셋의 ID가 기록되며, Modified 상태가 초기화됩니다. 행 앞의 활성화 체크 마크는 프리셋이 선택되고 동시에 수정되지 않은 경우에만 나타납니다. TrainingConfig의 어떤 값이 변경되는 즉시 (다른 인스펙터 섹션의 슬라이더, 스텝퍼, 토글), 이름 뒤에 주황색 "Modified" 배지가 나타납니다. 내장 프리셋은 덮어쓸 수 없습니다 — 수정 시 Save 버튼 (I1) 을 통해 자체 복사본을 만들어야 합니다.

 간단히 말하면

?? ???? ???? ?
 ??? ???? ???? ?
 ??? ???? ???? ?
 ?. ???? ???? ?
 ??? ???? ???? ?
 ??? ?????. ? ???? ?
 ?? ?????, ????
 ??? ???? ???? ?
 ? "Modified" ???? ?
 ? — ???? ???? ?
 ??? ?????. ???? ?
 ??? ?????. ??? ?
 ???? ???? Save... ?
 (I1) ???? ???? ??
 ??? ???? ? (I9) ??
 ?.

I8 (???) "Export..."

 위치

각 프리셋 행 마우스 오른쪽 클릭 → 첫 번째 항목 "Export..."

 기술적 설명

I5 (Export... 버튼) 와 기능이 동일하지만 더 편리하게 접근할 수 있습니다 — 프리셋을 먼저 선택할 필요 없이. 클릭된 행의 프리셋을 직접 내보냅니다. 모든 프리셋 카테고리 (내장 또는 Custom) 에 동일하게 작동하며 제한이 없습니다. 내보내기에는 내장 플래그와 원래 카테고리가 포함되지만, 다시 가져올 때는 I6에 설명된 대로 카테고리가 Custom으로 매핑됩니다.

 간단히 말하면

????? ???? ? — ??
 ? ???? ???? ????
 ?? ???? "Export..." ?
 ????. ??? ???? Export... ?
 ??? ???? ???? ????
 ???. ??? ???? ????
 ??? ???? ???? ????
 ???? ???? ???? ????
 ???? .radiancepreset ?
 ?? I5 ???? ????
 ??? ???? ???? Custom
 ???? ???? ????.

I12 Camera Alignment [?][?][?]

위치

인스펙터 → Training Configuration → Camera Alignment (상단 세그먼트 선택기).

기술적 설명

Apple Photogrammetry와 Native (Beta) 두 가지 옵션이 있는 세그먼트 선택기입니다. 선택은 다음 카메라 재구성에서 사용되는 SfM 백엔드를 결정합니다. 동시에 어떤 추가 인스펙터 요소가 표시될지에도 영향을 줍니다. Native는 EXIF가 없는 비디오 프레임에만 필요한 FOV 재정의 (I13) 를 추가로 표시합니다. 참고: 매우 큰 야외 촬영의 경우 외부 도구 (Metashape 또는 COLMAP) 의 결과를 Workspace 가져오기로 적용할 수 있습니다 — 제 1 장 (M5) 과 제 9 장 (Q3, Q6) 참고.

간단히 말하면

[?][?][?] [?][?][?] [?][?][?] [?][?][?] [?][?][?]
[?][?][?] [?][?][?] [?][?][?] — [?][?][?]
[?][?][?] [?][?][?] [?][?][?] [?][?][?]
[?] Apple Photogrammetry [?] [?]
[?] [?][?][?] [?][?][?] [?][?][?] [?]
[?] [?][?][?] [?][?][?]. Native (Beta)
[?] App Store [?] [?] [?] [?] [?]
[?], [?] [?] [?] [?] [?] [?] [?]
[?], EXIF [?] [?] [?] [?] [?]
[?] [?] FOV [?] [?] (I13) [?] [?]
[?] [?]. [?] [?] [?] [?] [?]
[?] [?] Metashape [?] COLMAP
[?] [?] [?] [?] [?] [?] [?]
[?] Workspace [?] [?] [?] [?] [?]
[?] [?] [?] [?]. [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] 9 [?]
[?] [?] [?] [?].

I13 FOV Override [?][?] (Native SfM)

위치

인스펙터 → Training Configuration → FOV Override (Camera Alignment = Native일 때만 표시).

기술적 설명

숫자 텍스트 필드 (범위 0-170°), 기본값 0 = EXIF 또는 휴리스틱으로부터의 자동 결정. 입력 이미지가 초점 거리 메타데이터가 없는 비디오에서 추출된 경우 수동 입력이 필요합니다. 일반적인 값: iPhone Wide ≈ 73°, DJI Mavic Wide-Crop ≈ 70°, 풀프레임 센서 드론 ≈ 84°. 값은 [0, 170] 으로 클램프됩니다 — 범위 외 값은 직접 잘립니다. Native SfM 파이프라인 (Q4/Q5) 에만 영향을 줍니다. Apple Photogrammetry 는 이 값을 완전히 무시합니다.

간단히 말하면

[?][?][?] EXIF [?] [?] [?] ([?] [?]
[?] [?] [?] [?] [?] [?] [?] [?]),
[?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?]. [?]
[?]: iPhone Wide ≈ 73°, DJI Mavic
Wide-Crop ≈ 70°, [?] [?] [?] [?]
[?] ≈ 84°. 0 [?] [?] [?] [?] [?]
[?] [?] — [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?]
[?]. 170° [?] [?] [?] [?] [?] [?]
[?] [?] [?]. [?] [?] [?] Camera
Alignment (I12) [?] Native [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?]
[?] — Apple Photogrammetry [?]
[?] [?] [?] [?] [?] [?].

I17 Auto-scale by scene



인스펙터 → Training Configuration → Auto-scale by scene (MCMC일 때만).

기술적 설명

활성화되면 효과적인 Max-Gaussians 상한이 SfM Init 포인트 수 x MCMC-Cap-Multiplier (기본 3.0) 와 함께 확장됩니다. 예: SfM이 250K 초기 점을 제공하고 베이스 Cap = 150K, Multiplier 3.0 → 효과적 상한 = max(150K, 750K) = 750K. 비활성화되면 베이스만 엄격히 적용됩니다. v1.4.5를 위해 도입되었습니다. 1000 프레임 이상과 상응하는 높은 SfM 점 밀도의 큰 야외 촬영이 경직된 150K 기본 Cap에 의해 Densification을 굼주리게 했기 때문입니다 — 잉여 점들은 남고 새 점들은 생길 수 없었습니다. Custom 프리셋에서는 OFF 기본, MCMC 내장에서는 ON. 학습 시에만 효과, 내보내기에는 영향 없음.

간단히 말하면

Gaussian (SfM Init) * Multiplier (기본 3.0) = Max-Gaussians 상한. 예: SfM이 250K 초기 점을 제공하고 베이스 Cap = 150K, Multiplier 3.0 → 효과적 상한 = max(150K, 750K) = 750K. 비활성화되면 베이스만 엄격히 적용됩니다. v1.4.5를 위해 도입되었습니다. 1000 프레임 이상과 상응하는 높은 SfM 점 밀도의 큰 야외 촬영이 경직된 150K 기본 Cap에 의해 Densification을 굼주리게 했기 때문입니다 — 잉여 점들은 남고 새 점들은 생길 수 없었습니다. Custom 프리셋에서는 OFF 기본, MCMC 내장에서는 ON. 학습 시에만 효과, 내보내기에는 영향 없음.

I18 Max Iterations



인스펙터 → Training Configuration → GroupBox → Max Iterations.

기술적 설명

범위 1 000-100 000, 단계 1 000인 스테퍼입니다. 옵티마이저 반복의 총 수를 결정합니다. 학습 시간과 선형 상관 (절반 = 약 50% 시간). 경험적 스위트 스폿: 20K (Classic Balanced, L1≈0.028), 40K (Classic Quality, L1≈0.023), 200K (MCMC Full, L1≈0.0246). Classic에서 40K 이상은 평균적으로 거의 개선을 가져오지 않습니다 — Diminishing Returns. 변경 시 Link 기능 (I19) 이 활성화되면 Densify Until이 비례적으로 함께 끌려갑니다 (기본 비율: 0.5, 즉 Densify-Until = Max/2).

간단히 말하면

Max Iterations 범위 1 000-100 000, 단계 1 000인 스테퍼입니다. 옵티마이저 반복의 총 수를 결정합니다. 학습 시간과 선형 상관 (절반 = 약 50% 시간). 경험적 스위트 스폿: 20K (Classic Balanced, L1≈0.028), 40K (Classic Quality, L1≈0.023), 200K (MCMC Full, L1≈0.0246). Classic에서 40K 이상은 평균적으로 거의 개선을 가져오지 않습니다 — Diminishing Returns. 변경 시 Link 기능 (I19) 이 활성화되면 Densify Until이 비례적으로 함께 끌려갑니다 (기본 비율: 0.5, 즉 Densify-Until = Max/2).

I19 Link/Unlink (Densify ↔ Iterations)

위치

인스펙터 → Training Configuration → GroupBox → Max Iterations와 Densify Until 사이의 작은 Link 버튼.

기술적 설명

Densify Until과 Max Iterations의 비율을 고정하는 토글 버튼입니다. 활성화 (Link 아이콘 강조) 일 때는 Max Iterations의 변경 시마다 Densify Until이 비례적으로 따라옵니다. Unlink (Link-Plus 아이콘) 일 때는 값이 독립적으로 유지됩니다. 기본은 linked입니다. 이것이 일반적인 상관관계를 반영하기 때문입니다 — 학습을 두 배 반복으로 늘리면 보통 Densification도 비례적으로 더 오래 실행하기를 원합니다. 비율은 Link 버튼 설정 시 현재 값에서 계산됩니다. 일반적인 비율은 0.5입니다 (Densify-Until = 반복 수의 절반).

간단히 말하면

Max Iterations Densify Until
 (Link) — Densify Until
 (link.badge.plus) linked.
 linked.
 Densification
 .99%
 .

I20 Densify Until

위치

인스펙터 → Training Configuration → GroupBox → Densify Until.

기술적 설명

범위 500–50 000, 단계 500인 스텝입니다. Clone/Split (Classic) 또는 Relocation (MCMC) 을 통해 새 Gaussian 이 더 이상 추가되지 않는 반복 인덱스를 결정합니다. 도달 후에는 위치와 색상만 정제됩니다. 값이 클수록 = Gaussian 더 많음 = 더 큰 파일, 더 긴 반복당 시간 (스텝당 GPU 시간 +30-60%). 일반적인 값: 15K (30K Max-Iter용), 20K (40K용), 100K (200K MCMC용). Link (I19) 가 활성화되면 자동으로 함께 확장됩니다. Classic과 MCMC에서 다르게 작용합니다: Classic은 성장을 완전히 중지하고, MCMC는 Relocation 로직을 중지하지만 Sample/Noise 적응은 계속 실행됩니다.

간단히 말하면

Gaussian
 — Classic
 Clone/Split, MCMC
 Relocation.
 =
 +30-60% GPU
 : 15K (30K Max-Iter),
 20K (40K), 100K (200K MCMC).
 Link (I19) Max
 Iterations —
 .

I21 SSIM Weight [?][?][?][?]



위치

인스펙터 → Training Configuration → GroupBox → SSIM Weight.



기술적 설명

0.05 단계로 0.0–1.0 슬라이더, “0.20”으로 표시됩니다. L1 Loss (0.0) 와 SSIM Loss (1.0) 를 혼합합니다. L1은 픽셀당 밝기를 조이고, SSIM은 구조적 유사성 (가장자리, 로컬 통계) 을 조입니다. 기본값 0.2는 원본 3DGS 논문 (Kerbl 2023) 의 값이며 수많은 세션에서 강력한 타협안으로 역설계되었습니다. 더 높은 값 (0.5+) 은 세부 보존을 선호하지만 로컬 밝기 오류를 무시할 수 있습니다. 더 낮은 값 (< 0.1) 은 선명한 가장자리에서 세부 손실로 이어집니다. SSIM 계산은 셰이더에서 11x11 Gaussian 윈도우로 실행됩니다. 성능: 0.0 (L1만) 에서 학습이 약 8-12% 더 빠릅니다. 셰이더에서 SSIM 계산이 건너뛰어지기 때문입니다.

간단히 말하면

[?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?]. 0.2[?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?]. [?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 (0.5+) — [?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?].

I22 Render Scale [?][?][?][?]



위치

인스펙터 → Training Configuration → GroupBox → Render Scale.



기술적 설명

0.25 단계로 0.25–1.0 슬라이더, “100%”로 표시됩니다. 원본 이미지 크기 대비 학습 렌더링 해상도를 확장합니다. 성능에서 가장 큰 레버: 50%는 GPU 시간을 약 75% 줄이고 (4배 적은 픽셀이므로), 25%는 약 94%. Gradient Threshold 는 자동으로 함께 확장됩니다. 슬라이더 아래에 MP 단위의 실시간 해상도 표시가 나타납니다 (예: “2304x1296 (3.0 MP)”). 현재 값이 권장값과 다른 경우 주황색 글자로 “— recommended: 50%”가 표시됩니다. 권장 사항은 ~3 MP 효과적 해상도를 목표로 합니다 — Apple Silicon GPU가 가장 효율적으로 처리할 수 있는 범위입니다. 예: 4K 원본 이미지는 자동으로 25%, FullHD 이미지는 100%가 권장됩니다. 변경 시 버퍼 재할당도 트리거됩니다.

간단히 말하면

[?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?] — [?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?]
 [?][?][?][?][?][?][?][?][?][?][?][?][?][?][?].

Enhancements 섹션 (I26–I29, I42–I44)

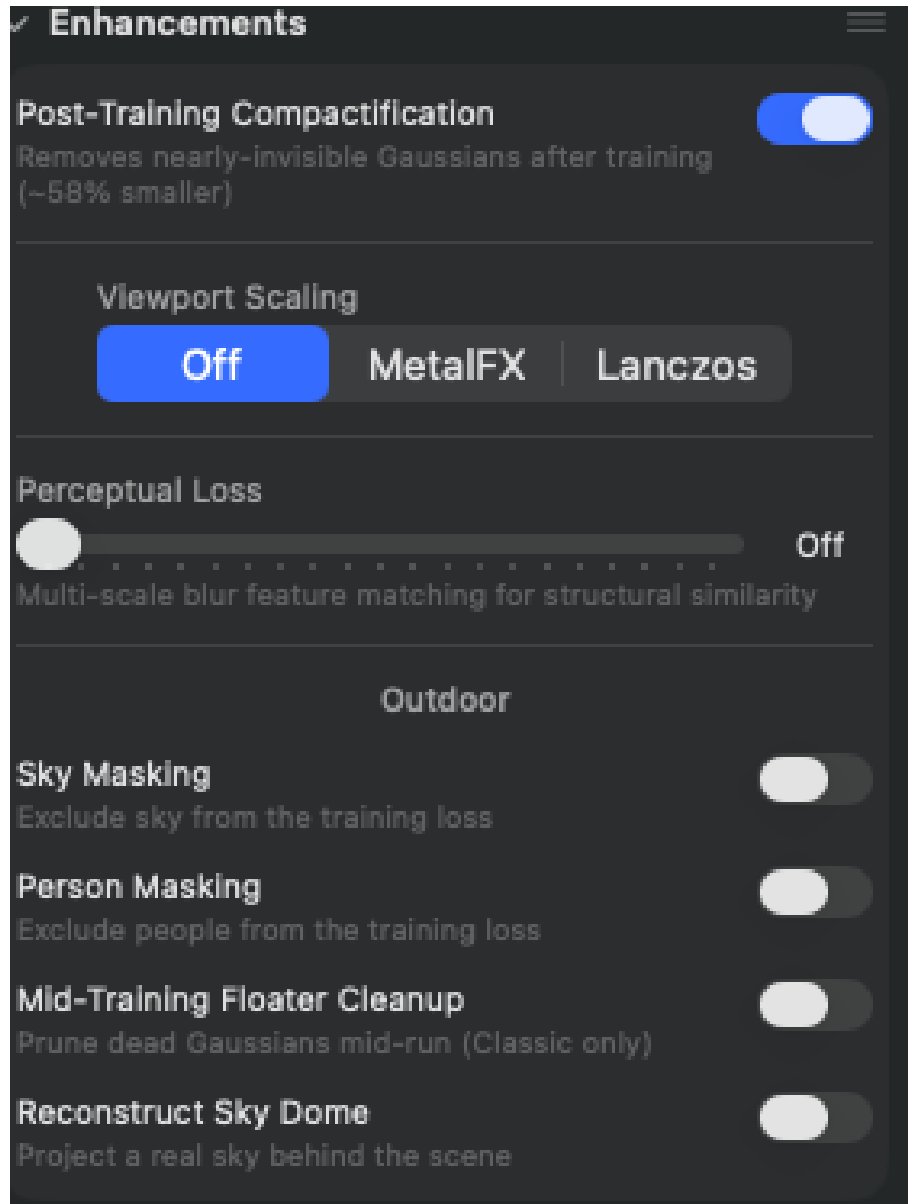


그림 12: *Enhancements* 섹션만 크롭 — 세 행: *Post-Training Compactification* (토글 켜짐), *Viewport Scaling* (세그먼트 선택기 *Off/MetalFX/Lanczos*), *Perceptual Loss* (슬라이더 “Off”). 각 행은 기능을 설명하는 부제목 포함

Enhancements 섹션은 핵심 학습 루프 자체를 변경하지 않고 이미지 품질을 개선하는 세 가지 기능을 그룹화합니다. 처음 두 가지 (I26–I27) 는 **Post-Training** 또는 **Viewport 단계** 입니다. Compactification은 학습 종료 후 정리하고, Viewport Scaling은 순수한 뷰포트 렌더러로 실행 중인 학습에 영향을 주지 않습니다. Perceptual Loss (I29) 는 섹션 소속에도 불구하고 학습 구성 요소입니다 — 학습 중에 추가 Loss 항목으로 활성화되므로 구분선을 통해 뷰포트 토크와 분리됩니다. v1.6부터 이 섹션에는 Outdoor 그룹 (I42–I44: Sky Masking, Mid-Training Floater Cleanup, Reconstruct Sky Dome) 도 있습니다 — 이전에 설정 창에 있다가 이제 프로젝트별로 여기에 자리 잡은, 하늘 플로터에 대응하는 학습 옵션입니다.

I26 Post-Training Compactification [?][?]

📍 위치

인스펙터 → Enhancements → Post-Training Compactification.

🔧 기술적 설명

V443 Post-Processing을 활성화합니다. 학습 반복 종료 후 Opacity 0.01 (1% 가시성) 미만의 Gaussian이 삭제됩니다. 경험적으로 이는 시각적 품질 손실 없이 파일 크기를 ~55-58% 줄입니다 — 이러한 Gaussian은 어쨌든 시각적으로 기여하지 않기 때문입니다. Compactification은 GPU 컴팩트 패스로 실행되며 Gaussian 수에 따라 1초 미만에서 몇 초까지 걸립니다. 학습 성능에는 영향을 주지 않습니다. 이 토크를 이 꺼져 있으면 보이지 않는 Gaussian도 내보내집니다 — 추가 학습 단계 (Continue Training) 용으로 형식을 사용하려는 경우에만 관련성이 있고, 그렇지 않으면 저장 공간 낭비입니다.

💬 간단히 말하면

[?][?] [?][?] [?][?] [?][?] [?][?] [?][?]
Gaussian (Opacity 1% [?][?]) [?][?]
[?][?] [?][?]. [?][?] [?][?] [?][?] [?][?]
[?][?] [?][?] [?][?] [?][?] [?][?] [?][?]
[?][?] [?][?] (55-58% [?][?] [?][?]
[?]). [?][?] [?][?] [?][?] GPU [?]
[?][?] [?][?] [?][?] 1[?] [?][?] [?][?] [?][?]
[?][?] [?][?] [?][?]. [?][?] [?][?] [?][?]
[?][?] [?][?] [?][?] — [?][?] [?][?]
[?][?] [?][?] [?][?] [?][?] Continue
Training[?][?] [?][?] [?][?] [?][?]
[?][?] [?][?] [?][?] Gaussian[?][?]
[?][?] [?][?] [?][?] [?][?] [?][?]. [?][?] [?]
[?][?] [?][?] [?][?] [?][?] [?][?] [?][?] [?]
[?][?] [?][?] [?][?] [?][?] [?][?].

I27 Viewport Scaling [?][?][?]



인스펙터 → Enhancements → Viewport Scaling (세 가지 옵션 Off, MetalFX, Lanczos가 있는 세그먼트 선택기).

기술적 설명

뷰포트 업스케일러를 선택하는 단일 세그먼트 선택기로 — 세 옵션은 상호 배타적입니다. 학습 해상도 (122 Render Scale 을 통해) 가 뷰포트 크기보다 낮은 경우 선택된 모드가 렌더링 된 프레임의 디스플레이 크기로 업스케일합니다. Off = 단순한 바이리니어 확대. MetalFX = Apple의 ML 기반 Spatial Upscaler로, 가장 선명한 옵션 (ML 모델이 선명한 가장자리에 최적화됨), M3 GPU에서 프레임당 약 1-2ms 오버헤드. Lanczos = Apple의 Metal Performance Shaders 와 8-Tap Sinc 재샘플링으로, ML이 없는 고전적 방식, 최소 오버헤드 (< 0.5ms), 품질은 MetalFX보다 낮지만 ML 특유의 미세한 선 구조 “부드럽게 다리기”가 없습니다. 렌더러 파이프라인은 전환 시 실시간으로 재구성됩니다 — 재시작 없이 즉시 보입니다. 배경: 이전에는 두 개의 별도 토클 (MetalFX + Lanczos) 이었고 동시에 켜질 수 있었습니다 — MetalFX가 조용히 Lanczos를 덮어쓰는 모순된 상태였습니다. 선택기는 이 상태를 제거합니다; 오래된 세션에서 상속된 “둘 다 켜짐” 상태는 다음 전환 시 스스로 MetalFX로 치유됩니다. 실시간 뷰포트에만 작용하고, 렌더링된 내보내기 (Orbit 비디오, 스크린샷) 에는 작용하지 않습니다 — 그것들은 전체 원본 해상도로 렌더링됩니다.

간단히 말하면

Viewport Scaling (Render Scale 50%, 122)가 켜져 있으면, 선택된 옵션이 렌더링된 프레임의 디스플레이 크기로 업스케일합니다. Off = 단순한 바이리니어 확대. MetalFX = Apple의 ML 기반 Spatial Upscaler로, 가장 선명한 옵션 (ML 모델이 선명한 가장자리에 최적화됨), M3 GPU에서 프레임당 약 1-2ms 오버헤드. Lanczos = Apple의 Metal Performance Shaders 와 8-Tap Sinc 재샘플링으로, ML이 없는 고전적 방식, 최소 오버헤드 (< 0.5ms), 품질은 MetalFX보다 낮지만 ML 특유의 미세한 선 구조 “부드럽게 다리기”가 없습니다. 렌더러 파이프라인은 전환 시 실시간으로 재구성됩니다 — 재시작 없이 즉시 보입니다. 배경: 이전에는 두 개의 별도 토클 (MetalFX + Lanczos) 이었고 동시에 켜질 수 있었습니다 — MetalFX가 조용히 Lanczos를 덮어쓰는 모순된 상태였습니다. 선택기는 이 상태를 제거합니다; 오래된 세션에서 상속된 “둘 다 켜짐” 상태는 다음 전환 시 스스로 MetalFX로 치유됩니다. 실시간 뷰포트에만 작용하고, 렌더링된 내보내기 (Orbit 비디오, 스크린샷) 에는 작용하지 않습니다 — 그것들은 전체 원본 해상도로 렌더링됩니다.

I29 Perceptual Loss [?][?][?][?]

📍 위치

인스펙터 → Enhancements → Perceptual Loss.

🔧 기술적 설명

0.01 단계로 0.0-0.2 슬라이더, 0.0에서 "Off"로, 그렇지 않으면 "0.05" 등으로 표시됩니다. 렌더링의 다중 스케일 Gaussian Blur를 Ground Truth 이미지와 비교하는 추가 Loss 항을 활성화합니다 (3 Blur 스케일). L1+SSIM만으로는 인식하지 못하는 구조적 차이를 포착합니다. V460 구현. 경험적으로 0.05-0.1의 값은 세션의 L1 점수를 몇 퍼센트 개선하지만 ~5% 학습 시간이 듭니다 (Blur 커널을 통한 추가 Forward Pass). 0.15 이상에서는 학습이 불안정해지고 L1이 다시 악화됩니다 (Loss 항이 최적화를 지배). 학습 **중에** 작용하고 Post-Processing에는 작용하지 않습니다 — "Enhancements" 섹션의 위치에도 불구하고 순수한 후속 업그레이드는 아닙니다.

💬 간단히 말하면

[?][?][?] [?][?][?][?][?][?][?][?][?]
 [?][?][?] [?][?][?][?] [?][?][?][?] [?][?]
 Loss [?][?][?][?][?]. [?][?][?][?], [?
 [?][?] [?][?] [?][?] [?][?] [?][?][?][?]
 [?][?][?][?] [?][?] [?][?][?] [?][?]
 [?]. L1+SSIM[?][?][?][?] [?][?] [?][?]
 [?] [?][?][?] [?][?][?][?] [?][?][?][?]
 [?]. [?][?] [?][?] [?] [?][?][?][?][?] —
 0.05[?][?] 0.1[?] [?][?][?] [?][?][?]
 [?], 0.15 [?][?][?][?] [?][?][?] [?][?]
 [?][?][?][?] Loss[?] [?][?] [?][?][?][?]
 [?]. 0 (Off) [?][?][?] [?][?][?] [?][?]
 [?] [?][?] [?][?] [?][?][?] [?][?][?][?].
 [?][?][?] Blur [?][?][?] [?][?] [?][?]
 Forward Pass[?] [?] 5% [?][?] [?][?]
 [?] [?][?][?][?][?]. "Enhancements"
 [?][?][?][?] [?][?][?][?] [?][?] [?][?] [?]
 [?] [?][?][?][?] Post-Processing[?]
 [?][?] [?][?][?][?] [?][?][?][?].

I 42 Sky Masking



인스펙터 → Enhancements (Outdoor 그룹) → "Sky Masking" 토크. 연결: AppState.trainingConfig.skyMaskingEnabled (프로젝트별, @DefaultFalse). 기본값: false .

기술적 설명

학습 전 Apple Vision 기반 하늘 픽셀 세분화를 활성화합니다. 학습 시작 전 각 입력 카메라의 하늘 영역이 Apple Vision Foreground Mask (Sky = Background) 를 통해 추출되어 해당 카메라의 픽셀별 마스크로 할당됩니다. 학습 중 픽셀당 Loss 기여는 하늘 마스크의 보수와 곱해집니다 — 하늘 픽셀은 그래디언트에 0으로 기여하므로 하늘로 투영되는 Gaussian은 최적화 신호를 받지 않고 따라서 "더 밀도가 높아지거나" "더 밝아지지" 않습니다. 야외/드론 장면의 플로터 (하늘의 어두운 덩어리) 를 상당히 줄입니다. 고전적 40K 학습에서 ~3% L1 회귀 비용이 발생합니다 (memory/dev_outdoor-floater-reduction.md 참고). 명확하게 인식 가능한 하늘이 있는 야외 장면에서만 의미가 있습니다. 실내 장면 또는 흰색 배경에서는 하늘 세분화가 잘못된 영역을 식별하고 유효한 Loss 신호를 차단합니다. 이 값은 이제 프로젝트별로 저장되며 (더 이상 앱 전역이 아님) 프리셋 / 장면 파일을 따릅니다.

간단히 말하면

???? ???? ???? ????
???? ???? ???? ????
???? ???? ???? — ??
? "???"?. ???? ????
???? ???? ???? ????
???? ???? ???? "???" ?
????"?? ?????. ???
???? ???? ???? ????
?????. ??? ???? ??
???? ???? ???? ????
???? — ???? ???? ??
???? ???? ?????. ???
?: memory/dev_outdoor-floater-reduction.md.

I 43 Mid-Training Floater Cleanup



인스펙터 → Enhancements (Outdoor 그룹) → “Mid-Training Floater Cleanup” 토글. 연결: AppState.trainingConfig.floaterCleanupEnabled (프로젝트별, @DefaultFalse). 기본값: false .

기술적 설명

Classic 40K 학습 (프리셋 “P4 Quality”) 에서 두 가지 추가 Density Control 패스를 켭니다: 반복 20 000과 30 000. 두 패스는 세 가지 기준으로 모든 Gaussian을 검색합니다: (a) 매우 낮은 Opacity (표준 0.005), (b) 작은 Screen Space 크기, (c) 지난 1000회 반복 동안 Loss 기여 없음. 세 조건을 모두 충족하는 Gaussian은 정리됩니다. 효과: 학습 종료 시 ~5-15% 적은 Gaussian, 드론/야외 장면의 하늘에서 어두운 덩어리가 눈에 띄게 적음. 클로즈업 실내 장면에서 ~1-3% L1 회귀 비용이 발생하므로 기본값으로 활성화되지 않습니다. 값은 재시작 간에 기억됩니다 (S7과 달리). 두 정리 반복 (20K, 30K) 은 단단히 정의되어 있으며 현재 UI를 통해 변경할 수 없습니다. 더 짧은 학습 (예: P2 Preview 5K) 에서는 토글이 반복 표시에 도달하지 않으므로 효과가 없습니다. **신규:** 이 토글은 이제 활성 프리셋이 **Classic Densifier**를 사용할 때만 조작할 수 있습니다 (densificationStrategy == .classic). MCMC나 Hybrid에서는 **비활성화**되고 인라인 힌트가 나타납니다. 이러한 전략은 죽은 Gaussian을 어차피 자체적으로 처리하기 때문입니다 (MCMC는 Relocation, Hybrid는 결합된 Reloc-/Noise 로직) — 그곳에서는 수동 Cleanup Pass가 효과가 없거나 역효과를 냅니다. 코드 참조: RadianceKitApp.swift, General 탭. 자세한 내용: memory/dev_outdoor-floater-reduction.md.

간단히 말하면

?? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? “???” Gauss ? ?
? ? ? ? ? . ? ? ? ? ? ? ? ?
? ? ? (20K? 30K ? ? ? ? ?)
? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?
— MCMC? Hybrid ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? (?
? ? ? ? ? ?), ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

I 44 Reconstruct Sky Dome



인스펙터 → Enhancements (Outdoor 그룹)
 → "Reconstruct Sky Dome" 토글. 연결:
 AppState.trainingConfig.skyDomeEnabled (프로젝트
 별, @DefaultFalse). 기본값: false .

기술적 설명

학습 전 하늘 돔 투영 (V549e MVP) 을 활성화합니다. SfM
 후, 학습 시작 전에 각 입력 카메라에 대해 S7과 공유된 Apple
 Vision 하늘 마스크가 이미지에서 추출되고, 하늘 픽셀은 카메
 라 인트린식과 함께 가상 구체 표면 (표준 반경 8x 장면 반경)
 으로 역투영됩니다. 이 구체에서 ~5000개의 새 Gaussian
 이 투영된 하늘 픽셀의 평균 색상값, 매우 큰 스케일 (장면 단
 위 1.0), 초기 Opacity 0.95로 초기화됩니다. 이 5000개의
 Gaussian은 고전적 의미의 Sky Mask가 아닙니다 — 다른 것
 들과 마찬가지로 학습되지만 높은 초기 Opacity를 통해 얇은
 껍질로 남습니다. 결과: 야외/드론 장면의 360° 새 시점에서 어
 두운 컨페티 덩어리 대신 실제 하늘 색상과 구름 구조가 나타납
 니다. 값은 재시작 간에 기억됩니다. 최소 360° 카메라 커버리
 지가 있는 야외 장면에서만 의미가 있습니다. 하늘 뷰가 없는 순
 수 객체 캡처에서는 효과가 없습니다. 상태: 실험적, 추가 야외
 세트에 대한 더 폭넓은 A/B 검증이 아직 남아 있습니다.

간단히 말하면

???? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? " ? ? ? " ? ? ? ? ? ? ? ?
 ? ? ? ? (? ? ? ? ? ? ? ? ? ?),
 RadianceKit ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? 360° ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? . ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

Metrics 섹션 (I30–I38)

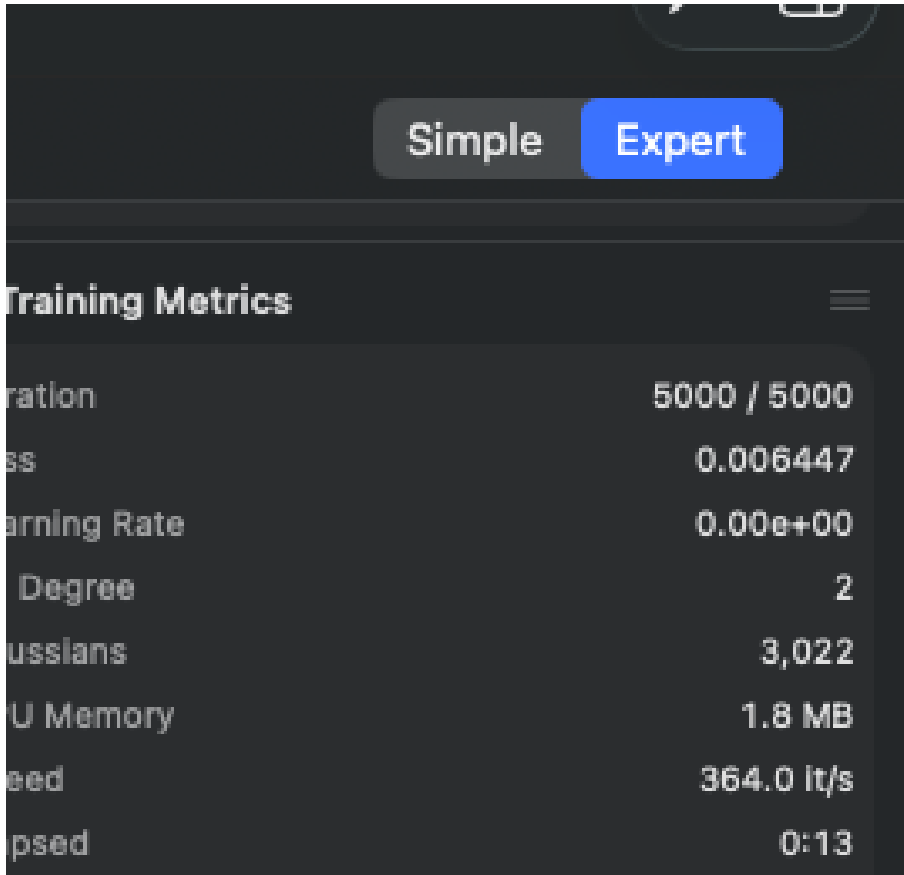


그림 13: Bouquet에서 학습 완료 후 Training Metrics 섹션만 크롭 (5K 반복, 최종 2 991 Gaussians) — 학습 메트릭 테이블 (Iteration, Loss, SSIM Loss, Combined Loss, Gaussian Count, Learning Rate, Elapsed, ETA)

학습이 실행되는 동안 Metrics 섹션은 학습 루프의 아홉 가지 실시간 값을 표시합니다. 학습 시작 전에는 섹션이 비어 있습니다 (“Start training to see live metrics”). 모든 값은 ~30회 반복마다 업데이트됩니다 (AsyncStream 업데이트 빈도). 섹션은 읽기 전용입니다 — 어떤 요소도 클릭하거나 변경할 수 없습니다. 더 깊은 분석을 위해서는 ~/Documents/RadianceKit/Logs/ 아래의 JSONL 학습 로그를 참고하십시오 (스크립트 `python3 scripts/analyze_logs.py best 5`).

I30 Iteration



인스펙터 → Metrics → Iteration. 읽기 전용.

기술적 설명

"4523 / 40000" 형식으로 표시됩니다 — 현재 반복과 계획된 총 반복. 학습 루프와 동기적으로 카운트되며, 루프는 ~30 Iter마다 값을 푸시합니다. 두 번째 숫자는 시작 시점의 Max Iterations 값에 해당합니다. 사용자가 그 후 스텝퍼를 조정해도 변경되지 않습니다 — 실행 중인 실행은 자체 스냅샷 복사본을 사용합니다. 학습 메뉴를 통해 앱이 반복을 추가하면 (Continue Training +5K/+10K/+20K) 분모가 증가합니다.

간단히 말하면

4523 / 40000. "4523 / 40000" 40 000 4523 11% 4523 / 40000 GPU Max Iterations (18) Continue Training (+5K/+10K/+20K)

I31 Loss



인스펙터 → Metrics → Loss. 읽기 전용.

기술적 설명

소수점 6자리의 Float 값 (예: "0.024385"). 결합된 L1+SSIM Loss (I21 SSIM Weight를 통해 혼합 제어) 와 선택적으로 Perceptual Loss (I29) 및 기타 정규화 항을 측정합니다. 스케일은 절대적이지 않고 장면별이며 — 대부분의 비교에 동일한 데이터셋이 필요합니다. 좋은 구성의 일반적인 최종 값: - Classic Quality 40K iters: 0.022-0.025 (Horse, Truck, Garden) - MCMC Full 200K iters: 0.024-0.028 - Outdoor Drone 30K: 0.030-0.060 (형상 때문에 더 나쁨) - Indoor 아파트: 0.018-0.025 5K 반복 후 0.10 초과 값은 SfM 문제 (나쁜 카메라 포즈) 를 시사합니다 — 중단하고 SfM을 다시 계산하십시오.

간단히 말하면

L1, SSIM, Perceptual Loss 0.03 0.05 0.03-0.06 0.10 Gradient Explosion

I32 Learning Rate



인스펙터 → Metrics → Learning Rate. 읽기 전용.

기술적 설명

과학적 표기법 표시 (예: "1.60e-04"). 위치 매개변수에 대한 현재 학습률 (3DGS에는 위치, SH-DC, SH-Rest, Opacity, Scale, Rotation에 대한 여섯 개의 독립적인 LR이 있으며 — 여기에는 위치 LR이 대표값으로 표시됨). 기본 시작값은 1.6e-4이며, 지수 감쇠를 통해 학습 종료 시 ~1.6e-6까지 떨어집니다. 감쇠는 학습 구성의 LR Schedule 필드 (제 6 장 T 필드) 를 통해 조정할 수 있습니다. LR이 비정상적으로 높게 유지되면 (예: 10K 반복 후 1e-3 이상) 잘못 로드된 구성을 시사할 수 있습니다.

간단히 말하면

Learning Rate (LR)은 모델이 학습하는 속도를 조절하는 값입니다. 너무 높으면 학습이 불안정해지고, 너무 낮으면 학습이 느립니다. 여기서는 위치 매개변수(위치, SH-DC, SH-Rest, Opacity, Scale, Rotation)에 대한 여섯 개의 독립적인 LR이 있으며, 기본 시작값은 1.6e-4입니다. 학습이 진행됨에 따라 LR은 지수적으로 감쇠하여 학습 종료 시 약 1.6e-6까지 떨어집니다. LR이 비정상적으로 높게 유지되면 (예: 10K 반복 후 1e-3 이상) 잘못 로드된 구성을 시사할 수 있습니다.

I33 SH Degree



인스펙터 → Metrics → SH Degree. 읽기 전용.

기술적 설명

정수 0-3. 색상 표현에 대한 Spherical Harmonics 차수. 0 (DC 성분만, 즉 Gaussian당 방향 독립적 색상 — 즉 RGB 상수 하나만) 에서 시작하여 학습 진행 과정에서 점진적으로 3까지 상승합니다. 표준 스케줄은 1000/2000/3000 반복에서 차수를 1씩 올립니다. SH-3은 Gaussian당 48개의 색상 계수에 해당합니다 (RGB 채널 3개 × SH 기저 함수 16개). 더 높은 SH 차수 = 더 많은 방향 의존성 반사 (광택 표면이 다른 시점에서 올바르게 다르게 보임), 그러나 더 많은 메모리와 더 느린 학습.

간단히 말하면

SH Degree는 Spherical Harmonics의 차수를 나타내며, 0부터 3까지의 정수입니다. 0은 방향 독립적인 색상(즉, RGB 상수 하나)을 의미하며, 3은 방향 의존적인 반사(광택 표면)를 나타냅니다. 더 높은 SH 차수는 더 많은 방향 의존성 반사를 의미하지만, 더 많은 메모리와 더 느린 학습을 초래합니다. 표준 스케줄은 1000, 2000, 3000 반복에서 차수를 1씩 올립니다. SH-3은 Gaussian당 48개의 색상 계수에 해당합니다 (RGB 채널 3개 × SH 기저 함수 16개).

I34 Gaussians



인스펙터 → Metrics → Gaussians. 읽기 전용.

기술적 설명

현재 모델의 Gaussian 수로, Locale 구분자로 형식화됩니다 (예: "524.318"). 성장: - Classic: SfM 초기 점 (일반적으로 50K-300K) 에서 시작하여 Clone/Split을 통해 Densify Until 직전까지 성장한 뒤 학습 종료까지 정적 (Pruning 제외) - MCMC: MCMC Cap까지 샘플 포인트 추가, 그 후 Relocation만

건강한 최종 값: - Classic Quality: 400K-700K (Horse 524K, Garden 800K) - MCMC Full: 정확히 Cap 위 (기본 150K, Auto-Scale Multiplier × SfM Count로 장면에 따라 500K-1.5M)

MCMC에서 숫자가 Cap의 < 60%로 떨어지면 → 이상 (Collapse 지표, 너무 공격적인 정규화 항을 시사).

간단히 말하면

3D Gaussian Densify Until (I20) = 500.000 Gaussian Classic Quality, MCMC Full Auto-Scale (I17) 500K 1.5M MCMC Cap 60% Collapse — Render Scale (I22) Densify Until (I20)

I35 GPU Memory



인스펙터 → Metrics → GPU Memory. 읽기 전용.

기술적 설명

Gaussian 수 × 616 바이트 (Memory Style로 형식화) 로 Gaussian 버퍼 메모리 사용량의 추정치. 616 바이트는 완전히 갖춰진 Gaussian의 경험적 크기입니다 (위치, 스케일, 회전, Opacity, SH 계수 차수 3, Gradient Accumulator). 표시는 렌더러 오버헤드 (Tile Buffer, Sort Buffer, Backward Buffer) 를 포함하지 않습니다 — 실제 GPU 메모리 요구량은 일반적으로 이 값의 2-3배입니다. 500K Gaussian에서: 표시 ~290 MB, 실제 ~700 MB. 1.5M Gaussian에서: 표시 ~880 MB, 실제 ~2.5 GB. 64+ GB 통합 메모리의 M3 Max에서는 문제가 없지만, 18 GB의 M3 Pro에서는 이미 한계입니다.

간단히 말하면

Gaussian GPU — 616 GPU 2-3 Tile, Sort, Backward Buffer 16-18 GB MacBook 500K Gaussian M3 Max Studio (64+ GB) 1.5M M3 Pro — Render Scale (I22) Densify Until (I20)

I36 Speed



인스펙터 → Metrics → Speed. 읽기 전용.



소수점 한 자리의 초당 반복 수 ("24.3 it/s"). 트레이너에 의해 최근 ~100 반복에 대한 슬라이딩 평균으로 계산됩니다. 일반적인 값: - Quick Preset (1K iters): 80-120 it/s (짧음, Steady State 없음) - Classic 20K @ 1.0 Render Scale (Truck 장면, M3 Max): 25-35 it/s - Classic 20K @ 0.5 Render Scale: 80-120 it/s - MCMC 200K @ 0.5 Render Scale: 25-50 it/s (Relocation 때문에 더 느림) - 1M+ Gaussian과 전체 해상도: < 10 it/s

학습 진행 과정에서 Speed 감소는 정상입니다 — 더 많은 Gaussian = 반복당 더 많은 컴퓨팅. 갑작스러운 하락 (예: 30 → 5 it/s) 은 GPU 열 스로틀링이나 경쟁 앱을 시사합니다.

간단히 말하면

20-50 it/s, Render Scale (50%) 80-120 it/s. Gaussian = 30 → 5 it/s GPU, Time Machine, Photos, 1M+ Gaussian 10 it/s.

I37 Elapsed



인스펙터 → Metrics → Elapsed. 읽기 전용.



이미 경과한 시간으로 "4:23" (m:ss) 또는 "1:23:45" (h:mm:ss). 1시간부터 형식이 전환됩니다. 순수한 학습 시간만 측정합니다. 선행 단계 (SfM 계산, 이미지 가져오기) 는 포함되지 않습니다. Pause/Resume 시 시계는 계속 흐릅니다 — 즉 Wall Clock 시간이며 CPU 시간이 아닙니다.

간단히 말하면

Wall Clock, "m:ss", "h:mm:ss", "CPU", "SfM", ETA, Elapsed, ETA.

I38 ETA



인스펙터 → Metrics → ETA. 읽기 전용.



추정 남은 시간으로 "17:42" 또는 "1:12:35"로 표시됩니다.
 계산: (Max Iterations – 현재 Iteration) / 초당 반복 수.
 Speed가 현재 0인 경우 (가장 처음 또는 일시 정지 시)
 "-"를 표시합니다. 추정치는 학습 종료 시의 일반적인 감속에
적응하지 않습니다 — 특히 MCMC와 큰 Densify Until 값
 의 Classic은 더 많은 Gaussian이 이미지에 들어오면서 느
 려지는 경향이 있습니다. 실제로는 일반적으로 초기 ETA보다
 10-20% 정도 더 오래 걸립니다.

간단히 말하면

ETA는 현재 남은 학습 시간의 추정치입니다. 이는 현재 Speed (초당 반복 수)를 기반으로 계산됩니다. Speed가 0이면 ETA는 "-"로 표시되며, 이는 현재 Speed가 0인 경우를 나타냅니다. ETA는 일반적으로 초기 ETA보다 10-20% 정도 더 오래 걸립니다. 이는 특히 MCMC와 큰 Densify Until 값을 사용하는 경우 더 많은 Gaussian이 이미지에 들어오면서 느려지는 경향이 있기 때문입니다. ETA는 현재 Speed (초당 반복 수)를 기반으로 계산되며, Speed가 0이면 ETA는 "-"로 표시됩니다.

Loss Chart 섹션 (I39-I41)



그림 14: 학습 완료 후 Loss History 섹션만 크롭 — Current 0.0064, Min 0.0035 (녹색), 파란색 진행 0.027 (Iteration 1) 에서 0.0035 (Iteration 5K) 로, Iter 200 즈음의 특징적 꺾임 포함, 아래에 주황색 Gaussian Count 차트

Loss Chart 섹션은 학습 진행을 시간에 따라 시각화합니다. 두 차트로 구성됩니다. 큰 Loss Curve 차트 (위, 파란색) 와 작은 Gaussian Count 차트 (아래, 주황색) 입니다. 둘 다 학습 중에 실시간으로 구축되며 다음 학습 시작까지 유지됩니다. 첫 학습 전에는 영역이 비어 있습니다 ("Loss curve will appear during training"). 차트는 순수 SwiftUI Path 드로잉 (Swift Charts 프레임워크 없음) 이므로 100K+ 포인트에서도 매끄럽게 렌더링됩니다.

언제 인스펙터를 사용해야 하는가?

빠른 참조: 일반적인 사용 사례별로 어떤 섹션 + 어떤 컨트롤?

일반 작업	섹션	컨트롤 ID
완성된 Splat의 색상 채도 낮추기	Look	L1 (Saturation)
바늘/콘페티 Splat 둥글게 만들기	Look	L2 (Splat length)
구멍 난 구름 채우기 / Splat 키우기	Look	L3 (Splat size)
퀘도에서 먼 "Far 콘페티" 흐려지게 하기	Look	L4 (Fade far region)
Look 조정 버리기	Look	L5 (Reset finishing)
사전 만들어진 설정 로드	Presets	I7 (행 클릭)
자체 설정 저장	Presets	I1 → I2 → I4
동료와 설정 공유	Presets	I5 (Export) 또는 I6 (Import)
SfM 백엔드 전환 (예: Apple PG가 너무 불안정해서)	Training Configuration	I12 (제 9 장 참고)
EXIF 초점 거리가 없는 비디오 프레임 처리	Training Configuration	I13 (FOV Override)
COLMAP 성능: Classic 대신 GLOMAP	Training Configuration	I14
Classic에서 MCMC로 전환	Training Configuration	I15
학습을 더 오래 실행	Training Configuration	I18 (Max Iter) 1. I20 (Densify Until) — I19로 연결
GPU 시간 절반으로 줄이기	Training Configuration	I22 (Render Scale 50%)
학습 품질 +6% (MCMC)	Training Configuration	I16 (MCMC Quality)
SfM 점이 많은 야외 장면	Training Configuration	I17 (Auto-scale by scene)
COLMAP 경로 설정 / 전환	Training Configuration	I23 / I24 / I25
내보내기 파일 더 작게	Enhancements	I26 (항상 켜 둘 것)
학습 시간 증가 없이 뷰포트 선명하게	Enhancements	I27 (Viewport Scaling → MetalFX)
MetalFX가 너무 매끄럽게 만들 → 대안	Enhancements	I27 (Viewport Scaling → Lanczos)
미세 구조에서 마지막 한 방울의 세부 정보	Enhancements	I29 (Perceptual Loss 0.05-0.1)
학습 모니터링	Metrics	I30 (진행), I36 (속도), I38 (남은 시간)
품질 워킹 세트 Densify until 학습 완료 인식	Metrics	I31 (5K 후 Loss ≤ 0.05를 줄여 40% 이상 개선) 또는 I32 (5K 후 SfM 재계산)

②

제 3 장 — 설정

설정 창은 RadianceKit → ??... 또는 표준 단축키 **⌘,** 를 통해 열립니다. 두 개의 탭이 있습니다.

General 과 **AI Helpers** 입니다. 제 2 장의 인스펙터 값과 달리 이 창의 설정은 **앱 전역으로** (모든 프로젝트에 걸쳐) 적용됩니다 — UserDefaults에 영구화되어 앱 재시작을 견딥니다. General 탭은 세 가지 콘텐츠 섹션으로 그룹화되어 있습니다: Interface, Viewport, Training. AI Helpers 탭은 SfM 및 학습 전처리를 위한 온디바이스 머신 러닝 헬퍼 (Vision, CoreML) 를 컵니다. (이전에 여기에 있던 세 개의 야외 플로터 토클 — Sky Masking, Mid-Training Floater Cleanup, Reconstruct Sky Dome — 은 v1.6부터 Expert 인스펙터의 Enhancements 섹션으로 이동했으며, 이제 프로젝트별로 저장됩니다. 제 2 장의 I42-I44를 참고하십시오.)

모든 AI Helpers를 일괄 활성화하거나 비활성화하는 이전의 조작 요소는 현재 버전에서는 더 이상 존재하지 않습니다 — 따라서 여기에 문서화되지 않습니다. 아직 출시되지 않은 헬퍼를 위한 이전의 “Coming Soon” 섹션도 제거되었으며 여기에 참조되지 않습니다.

General 탭

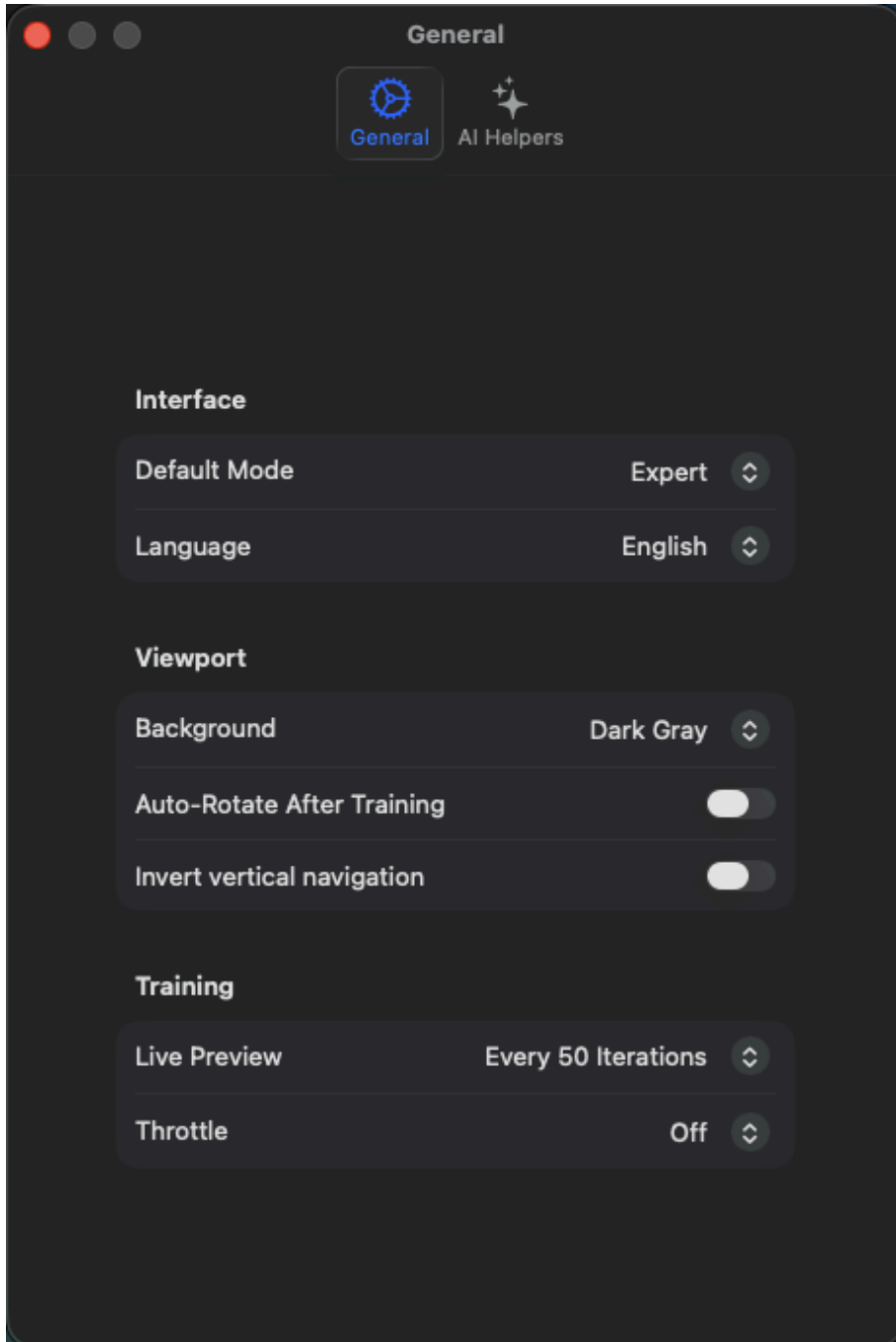


그림 15: 설정 → General 탭, 인터페이스, 뷰포트, 학습, 실험 섹션 포함

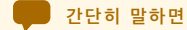
S1 Default Mode



Settings → General → Interface → Default Mode 선택기. 연결: AppState.defaultMode. 기본값: `.simple`.



다음 시작 후 앱이 두 UI 모드 중 어느 것으로 열릴지를 제어합니다. "Simple Mode"는 4단계 가이드형 마법사 워크플로 (가져오기 → 처리 → 미리보기 → 내보내기, 제 10 장 Z1-Z4 에 문서화) 이고, "Expert Mode"는 제 2 장의 Navigator, 3D 뷰포트, Expert 인스펙터가 있는 고전적인 3 패널 레이아웃입니다. 값은 재시작 간에 기억됩니다. 메뉴 Mode → Simple Mode (⌘1) / Mode → Expert Mode (⌘2) 와 효과가 동일하지만, 메뉴는 실행 중인 세션을 전환하는 반면 이 선택기는 향후 세션의 기본값을 설정합니다. 두 모드 모두 동일한 프로젝트 상태에 접근합니다 — 모드 전환 시 프로젝트, 카메라, 학습 구성은 유지됩니다. 모드별 툴바 버튼은 즉시 다시 렌더링됩니다.



??? RadianceKit? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? . "Simple Mode" ? ?
? ? ? ? ? ? ? ? ? . ? ? ? ? ? ?
? ? ? ? , ? ? ? ? ? ? ? ? ? ? , ?
? ? ? ? ? ? ? . "Expert Mode" ?
? 2 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

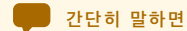
S2 Language



Settings → General → Interface → Language 선택기. 연결: AppState.language. 기본값: `.system` (macOS 언어를 따름).



macOS 시스템 언어와 독립적으로 전체 앱 UI의 표시 언어를 선택합니다. RadianceKit은 17개 언어 (`de` , `en` , `pl` , `en-AU` , `ar-SA` 와 추가 12개) 로 현지화되어 있습니다. "System"의 경우 앱은 macOS 언어를 따릅니다. 명시적 선택의 경우 언어 설정은 재시작 간에 기억됩니다. 완전한 적용은 보통 앱 재시작이 필요합니다. 현지화 번들은 시작 시에만 로드되기 때문입니다. 프로젝트에 문서화된 298개의 현지화 키는 모두 고려됩니다. 모든 하위 보기와 도움말 도구 설명의 텍스트가 포함됩니다.



Mac? ? ? ? ? ? ? ? ? ? ?
?? RadianceKit ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? (? ? ? ? ? ? ?) , ?
? ? ? ? ? ? ? ? ? ? . ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? . ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

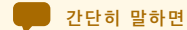
S3 Viewport Background



Settings → General → Viewport → Background 선택기. 연결: AppState.viewportBackground. 기본값: `.darkGray` (RGB 0.1, 0.1, 0.1).



3D 뷰포트의 기본 배경색을 설정합니다. 세 가지 옵션: "Dark Gray" (RGB 0.1, 0.1, 0.1 — 기본값), "Black" (0, 0, 0), "White" (1, 1, 1). 설정은 재시작 전에 새 프로젝트와 세션의 기본값을 영구화하면서 실행 중인 Metal 렌더러도 즉시 업데이트합니다. 동일한 옵션이 메뉴 Viewport → Background (M21, M22, M23) 에 있지만, Settings 선택기는 기본값을 설정하는 반면 메뉴는 실행 중인 표시를 전환합니다. 스크린샷과 데모 비디오에 중요: 흰색 배경은 녹색/파란색 플로터를 더 강하게 강조하고, 어두운 배경은 깨끗한 렌더 촬영에 더 좋습니다.



3D 뷰포트의 기본 배경색을 설정합니다. 세 가지 옵션: "Dark Gray" (RGB 0.1, 0.1, 0.1 — 기본값), "Black" (0, 0, 0), "White" (1, 1, 1). 설정은 재시작 전에 새 프로젝트와 세션의 기본값을 영구화하면서 실행 중인 Metal 렌더러도 즉시 업데이트합니다. 동일한 옵션이 메뉴 Viewport → Background (M21, M22, M23) 에 있지만, Settings 선택기는 기본값을 설정하는 반면 메뉴는 실행 중인 표시를 전환합니다. 스크린샷과 데모 비디오에 중요: 흰색 배경은 녹색/파란색 플로터를 더 강하게 강조하고, 어두운 배경은 깨끗한 렌더 촬영에 더 좋습니다.

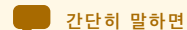
S4 Auto-Rotate After Training



Settings → General → Viewport → "Auto-Rotate After Training" 토글. 연결: AppState.autoRotateAfterTraining. 기본값: `false`.



학습 종료 직후 장면 무게 중심을 중심으로 뷰포트 카메라의 연속 턴테이블 회전 (표준 회전을 ~0.3 rad/s) 을 시작합니다. 데모 세션, A/B 비교, 그리고 360° 시점에서 장면 가장자리에 "플로터"가 생성되었는지 직접 평가하는 데 유용합니다. 효과는 메뉴 Viewport → Toggle Auto-Rotation (M16, ⌘T) 과 시각적으로 동일하지만, 여기 토글은 수동 대신 학습 종료 후 자동으로 동작을 트리거합니다. 나중에 언제든지 메뉴를 통해 또는 뷰포트 클릭 (회전 일시 정지) 으로 중단할 수 있습니다. 학습 성능에 영향을 주지 않습니다 — 회전은 학습이 완료된 후에만 실행됩니다.



학습 종료 직후 장면 무게 중심을 중심으로 뷰포트 카메라의 연속 턴테이블 회전 (표준 회전을 ~0.3 rad/s) 을 시작합니다. 데모 세션, A/B 비교, 그리고 360° 시점에서 장면 가장자리에 "플로터"가 생성되었는지 직접 평가하는 데 유용합니다. 효과는 메뉴 Viewport → Toggle Auto-Rotation (M16, ⌘T) 과 시각적으로 동일하지만, 여기 토글은 수동 대신 학습 종료 후 자동으로 동작을 트리거합니다. 나중에 언제든지 메뉴를 통해 또는 뷰포트 클릭 (회전 일시 정지) 으로 중단할 수 있습니다. 학습 성능에 영향을 주지 않습니다 — 회전은 학습이 완료된 후에만 실행됩니다.

S5 Live Preview Interval



Settings → General → Training
→ Live Preview 선택기. 연결:

AppState.trainingConfig.livePreviewInterval. 기본값: 0 (Off).



실행 중인 학습 스냅샷이 3D 뷰포트에 렌더링되는 반복 간격을 결정합니다. 네 가지 이산 값: 0 ("Off"), 50, 250, 1000 반복. Live Preview 활성화 시 트레이너는 Gaussian 버퍼를 GPU에서 별도의 렌더 버퍼로 복사하고 뷰포트 재그리기를 트리거합니다. "Off"의 경우 뷰포트는 학습 완료 후에만 업데이트됩니다. 성능 비용: M3 Ultra에서 50회 반복마다 ~5-10% 느려짐, 250회 반복마다 ~1-2% 느려짐, 1000회 반복마다 측정 불가. 스냅샷 버퍼의 메모리 오버헤드는 간격과 관계없이 일정 ~2 GB. 값은 새 학습의 기본값으로 사용됩니다. 학습 시작 후에는 Training 인스펙터가 이 학습의 실제 실시간 값을 표시합니다. 간격 50에서 시각적 인상은 점 구름의 매끄러운 "성장"이고, 1000에서는 끊기게 느껴집니다.

간단히 말하면

3D 뷰포트에 렌더링되는 스냅샷의 반복 간격을 결정합니다. 값은 0 ("Off"), 50, 250, 1000 중 하나입니다. "Off"는 학습 완료 후 뷰포트를 업데이트합니다. 성능 비용은 M3 Ultra에서 50회 반복마다 ~5-10% 느려짐, 250회 반복마다 ~1-2% 느려짐, 1000회 반복마다 측정 불가. 스냅샷 버퍼의 메모리 오버헤드는 간격과 관계없이 일정 ~2 GB. 값은 새 학습의 기본값으로 사용됩니다. 학습 시작 후에는 Training 인스펙터가 이 학습의 실제 실시간 값을 표시합니다. 간격 50에서 시각적 인상은 점 구름의 매끄러운 "성장"이고, 1000에서는 끊기게 느껴집니다.

S6 Throttle Delay



Settings → General → Training → Throttle 선택기. 연결: AppState.trainingConfig.throttleDelayMs. 기본값: 0 (Off).



학습 반복 사이에 인위적인 지연을 밀리초 단위로 삽입합니다. 네 가지 이산 값: 0 ("Off"), 2 ("Light"), 5 ("Moderate"), 10 ("Eco"). 의미: 더 긴 학습 (몇 시간)의 경우 GPU가 그렇게 않으면 100% 사용됩니다. 이는 눈에 띄게 느린 시스템 UI를 야기합니다 (마우스 포인터가 끊김, 다른 앱이 느려짐). Throttle 지연은 다른 작업이 실행될 수 있는 GPU 휴식을 제공합니다. 성능 비용은 상당합니다. 5 ms Throttle에서 일반적인 40K 학습은 Throttle 없을 때보다 약 50-80% 더 오래 걸립니다. 성능 모드 "Eco" (10 ms)에서는 반복당 지연이 반복 자체보다 더 깁니다 — 2-3배 느림. Throttle 활성화 시 선택기 아래에 "Throttle is on. Training will be slower than usual."이라는 힌트가 나타납니다. 앱 자체는 눈에 띄게 더 잘 반응하지 않습니다 — 다른 앱만 혜택을 봅니다.

간단히 말하면

MacBook에서 GPU 휴식을 제공하는 지연을 삽입합니다. 값은 0 ("Off"), 2 ("Light"), 5 ("Moderate"), 10 ("Eco") 중 하나입니다. "Eco"는 GPU 휴식을 제공하는 지연을 삽입합니다. 성능 비용은 상당합니다. 5 ms Throttle에서 일반적인 40K 학습은 Throttle 없을 때보다 약 50-80% 더 오래 걸립니다. 성능 모드 "Eco" (10 ms)에서는 반복당 지연이 반복 자체보다 더 깁니다 — 2-3배 느림. Throttle 활성화 시 선택기 아래에 "Throttle is on. Training will be slower than usual."이라는 힌트가 나타납니다. 앱 자체는 눈에 띄게 더 잘 반응하지 않습니다 — 다른 앱만 혜택을 봅니다.

AI Helpers 탭

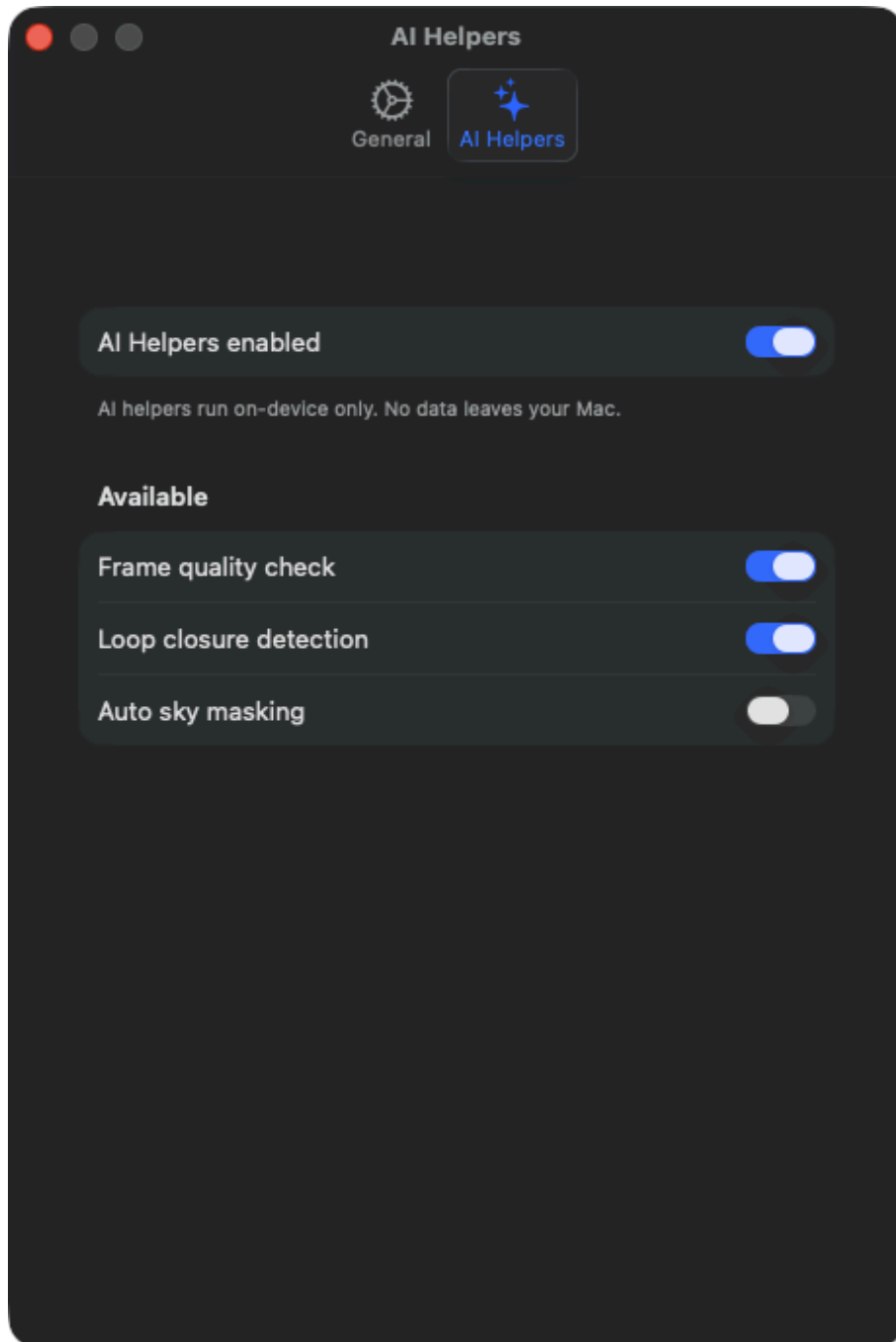


그림 16: 설정 → AI Helpers 탭, 마스터 스위치와 하위 토글 포함


S11 AI Helpers enabled (Master)

 위치

Settings → AI Helpers → 첫 번째 섹션 → “AI Helpers enabled” 토글. 연결: AppState.aiHelpersEnabled. 기본 값: true .

 기술적 설명

파이프라인의 모든 AI Helpers 기능에 대한 마스터 스위치. 꺼져 있으면 가져오기 및 SfM 파이프라인이 모든 ML 기반 전 처리 단계를 완전히 건너뛸니다 — Apple Vision 호출 없음, CoreML 모델 로드 없음, NPU 깨우기 없음. 켜져 있으면 개별 하위 토글 (S12–S13) 이 참조됩니다. 값은 재시작 간에 기억됩니다. 다음 단계에 영향을 줍니다: (a) SfM 전 프레임 품질 사전 점검 (S12), (b) 루프 폐쇄 감지 (S13). 중요: 꺼져 있으면 두 하위 토글이 비활성화되고 시각적으로 회색이 됩니다. 푸터 힌트는 모든 AI Helpers가 엄격하게 온디바이스에서 실행됨을 강조합니다 — 이미지 업로드 없음, 클라우드 처리 없음. 개인 정보 보호 보장은 Apple Vision 프레임워크 (Neural Engine 의 로컬) 와 앱 번들에 직접 위치한 CoreML 모델을 독점적으로 사용함으로써 달성됩니다.

 간단히 말하면

AI/?? ???? ???? ????
?? ???? ???? ????
??. ??? “?””???. ?
?? ???? Mac? ???? ?
?? ???? ???? ???? ?
??. ??? ???? ???? (?:
?? ???? Mac? NPU
? ? ? ?), ? ? ? ?
— ??? ???? ? ? ? ?
?? ? ? ? ? ? ? ? ?
?? ? ? ? ?.

S12 Frame quality check



Settings → AI Helpers → Available 섹션 → “Frame quality check” 토글. 연결: AppState.aiHelpersFrameQualityCheck. 기본값: true .



SfM 호출 전 각 가져온 프레임을 분석하는 Frame Quality Screener (Phase 3.11) 를 활성화합니다. 프레임당 파이프라인 단계: (a) Apple Vision의 Laplacian Variance 필터 (블러 감지 — 임계값 ~150), (b) 히스토그램 기반 과/저노출 점검 (임계값: 픽셀의 >5%가 0 또는 255), (c) 빈 프레임 감지 (모든 픽셀에 대한 표준 편차 < 5). 세 점검을 모두 통과한 프레임은 곧바로 통과합니다. 적어도 하나의 점검을 실패한 프레임은 문제가 있는 각 프레임을 썸네일과 이유와 함께 나열하고 제거할지 묻는 모달 확인 대화창을 트리거합니다. 중요: 자동 삭제 없음 — 대화창이 항상 필요하며, 사용자가 최종 결정을 유지합니다. 성능: M3 Ultra에서 프레임당 ~50 ms, 병렬 실행. 꺼져 있으면 모든 프레임이 점검 없이 SfM에 전달됩니다. 마스터 (S11) 가 비활성화되면 이 토글은 시각적으로 회색이고 효과가 없습니다. 메모리에 따른 출시 상태: SHIPPED 2026-05-23.

간단히 말하면

Placeholder text consisting of multiple lines of question marks.

S13 Loop closure detection



Settings → AI Helpers → Available 섹션 → "Loop closure detection" 토글. 연결: AppState.aiHelpersLoopClosure. 기본값: true .

기술적 설명

Apple Vision Feature Print 기반 루프 폐쇄 감지를 활성화합니다. 각 가져온 프레임에 대해 이미지 콘텐츠의 신경 임베딩을 나타내는 ~768차원 특징 벡터가 계산됩니다. 그 후 모든 Feature Print는 Cosine Similarity로 쌍별 비교됩니다. Similarity > 0.85이고 프레임 인덱스 거리 > 50 인 쌍 (즉 비인접 프레임) 은 "루프 폐쇄 후보"로 식별되어 프로젝트 폴더의 사이드카 JSONL 파일에 기록됩니다. 정보용만 — 가져온 이미지 시퀀스는 수정되지 않습니다. 의의: SfM 솔버 (특히 COLMAP) 에 이 프레임들이 3D 공간에서 함께 클러스터에 속한다는 힌트를 제공합니다. Native SfM의 경우 사이드카 정보는 현재 문서용일 뿐입니다. COLMAP은 커스텀 matches 파일을 통해 내부적으로 힌트를 사용합니다 (수동 통합 가능, 자동으로 연결되지 않음). 성능: M3 Ultra에서 프레임당 ~200 ms, 병렬 실행. 꺼져 있으면 Feature Print가 생성되지 않습니다. 마스터 (S11) 가 비활성화되면 시각적으로 회색입니다.

간단히 말하면

?? ???? ???? ???? ????
????? ???? ???? ???? ????
???, ?? ???? ???? ?
???? ???? ???? ???? ????.
????? ???? ???? ???? "??
???? ??"? ???? ????
????? ???? ???? ???? ?
?? ???? ???? ???? ???? SfM ?
? (?? COLMAP) ? ? ????
????? ? ???? ? 3D ????
????? ? ???? ??. ?? ?
"??"???. ???? ????
????? ???? ???? ???? ?
????? ???? ???? ???? ?
?.

인스펙터 미리 설정

인벤토리 테이블의 나머지 설정 항목 (S17-S33) 은 Expert 인스펙터에서 미러링된 것이며 제 2 장 (인스펙터 컨트롤 112-129) 에 문서화되어 있습니다. 설정 창에 물리적으로 나타나지 않으며, UserDefaults를 통해 영구화되는 TrainingConfig 속성을 통해 실행되기 때문에 인벤토리에만 나열되어 있고 형식적으로 설정 성격을 가집니다. 콘텐츠 설명은 거기를 참조하십시오.

언제 무엇을?

설정	범위	지속성
S1 Default Mode	앱 전역	앱 재시작
S2 Language	앱 전역	앱 재시작
S3 Viewport Background	앱 전역 (기본값) + 런타임	앱 재시작
S4 Auto-Rotate After Training	앱 전역	앱 재시작
S5 Live Preview Interval	새 학습의 기본값	앱 재시작
S6 Throttle Delay	새 학습의 기본값	앱 재시작
S11 AI Helpers Master	앱 전역	앱 재시작
S12 Frame quality check	앱 전역	앱 재시작
S13 Loop closure detection	앱 전역	앱 재시작

앱 전역 = 모든 프로젝트에 적용. 새 학습의 기본값 = 다음에 생성되는 학습에만 적용, 실행 중인 세션은 변경되지 않음. 현재 학습 = 실행 중인 학습 구성에 즉시 적용되지만 명시적인 재가져오기 없이는 영구화되지 않음.

2

제 4 장 — 보조 창

메인 윈도우 (3D 뷰포트와 인스펙터) 외에도 RadianceKit은 일곱 개의 추가 창을 관리하며, 모두 Help 메뉴를 통해 열립니다. 위에서 아래로의 목록: User Guide (⌘?), Keyboard Shortcuts (⌘/), Open Training Logs... (앱 창을 열지 않고 Finder를 엽니다. 따라서 여기서는 더 다루지 않음), Manage Storage..., Pareto Dashboard... (⌘D), Holdout Analysis... (⌘H), BayesOpt Console... (⌘B). 그 중 세 가지 — Dashboard, Holdout, BayesOpt — 는 독립적인 분석 도구입니다. 각각 자체 View Model 스택을 가지며 디스크의 JSON 파일을 읽거나 쓰고, 앱 시작 시 창을 특정 파일에 바로 가리키게 할 수 있는 CLI 인수가 있습니다 (`--dashboard-dir` , `--holdout-file` , `--bayesopt-autorun`).

네 개의 단순한 창 (User Guide, Keyboard Shortcuts, Manage Storage, 그리고 하위 메뉴 항목 Open Training Logs / Open Exports Folder) 은 조작 요소당 짧은 항목을 받습니다. 세 분석 창은 더 자세히 문서화되어 있습니다 — 각각 창에서 무엇을 보는지, 언제 열어야 하는지, 표시된 이미지를 어떻게 해석할지 설명하는 도입부와 함께.

장 끝에는 메인 윈도우의 인스펙터에 대한 상호 참조 섹션이 있습니다. 실행 중인 학습에서 실시간 Loss 차트와 Gaussian Count 표시에서 무엇을 의미 있게 읽을 수 있는지에 대한 것입니다.

User Guide (W1-W4)

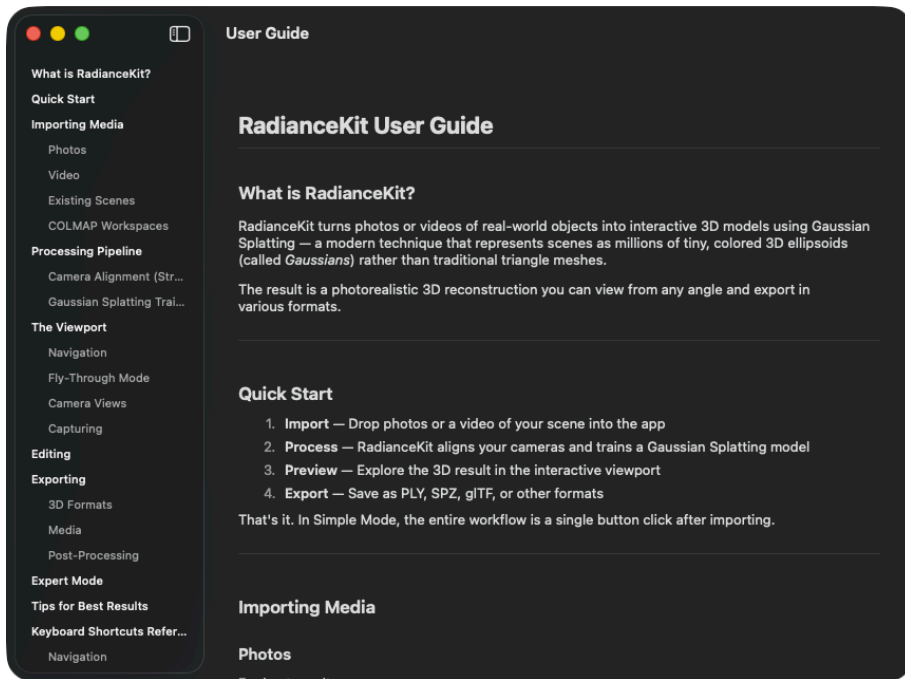


그림 17: User Guide 창, 왼쪽 사이드바와 오른쪽 렌더링된 Markdown 콘텐츠

그것이 무엇인지: 앱에 동봉된 `guide_<?>.md` 를 렌더링하는 내장 도움말 창입니다. 언어는 설정 (General 탭 → Language) 에서 또는 거기에 “System”이 있으면 macOS 언어 환경 설정에서 파생됩니다. 레이아웃은 고전적입니다: 왼쪽에 모든 헤딩이 있는 사이드바, 오른쪽에 흐름 텍스트.

언제 열까 하나의 항목에 대해 빠른 상기가 필요할 때 — 즉 키워드 대체로. 자세한 참조는 이 매뉴얼입니다. 내장 도움말 창은 명령줄의 `--help` 와 비슷합니다. 각 앱 릴리스마다 업데이트되지만 콘텐츠 면에서는 더 표면적으로 유지됩니다.

W1 NavigationSplitView (?? + ??)

위치

Help → User Guide (???)..

기술적 설명

콘텐츠 트리를 위한 좁은 사이드바 (최소 180 pt 폭) 와 실제 Markdown 콘텐츠를 위한 스크롤 가능한 디테일 영역의 2 열 레이아웃입니다. 창은 최소 크기 700 x 500 pt를 가집니다. 처음 열 때 창은 앱 번들에서 적절한 `guide_<lang>.md` 를 로드하고 (대체로 `guide_en.md`), 블록 레코드 (H1-H4 헤딩, 단락, 목록, 테이블, 구분선) 로 파싱하며, 사이드바를 위한 헤딩 구조를 별도로 추출합니다. 인라인 형식 (Bold, Italic, Code Span) 은 내장 Markdown 엔진을 통해 렌더링됩니다. 언어는 앱 설정에서 읽히며, 중국어 (zh-Hans) 와 브라질 포르투갈어 (pt-BR) 의 특수 경우에는 이 변형이 zh 또는 pt와 다르기 때문에 전체 Locale 태그로 유지됩니다.

간단히 말하면

?? ???? ???? , ???? ?
???, ???? ???? . ??
???? ???? ???? ????
?????. ???? ???? ??
??? ???? ???? ????
?? — ??? ???? ????
??.

W2 List (?? ? ? ? ? ?)

📍 위치

User Guide 창의 왼쪽 열..

🔧 기술적 설명

현재 Markdown 문서의 모든 H2 및 H3 헤딩에 대한 목록입니다. H2 항목은 들여쓰기 없이 Medium 글자 굵기로 나타나고, H3 항목은 왼쪽에 16 pt 들여쓰기와 축소된 Foreground 스타일로 표시됩니다. H4 이하는 무시됩니다. 그 이상의 깊이는 사이드바를 혼잡하게 만들기 때문입니다. 앵커 ID는 헤딩 텍스트의 슬러그화 (소문자 + 공백을 대시로

1. 글자/숫자/대시로 필터링) 로 생성됩니다 — GitHub가 Markdown 앵커에

사용하는 것과 동일한 알고리즘이므로 문서의 외부 URL도 잠재적으로 같은 앵커에 도달할 수 있습니다. 목록은 네이티브 macOS 스타일을 사용합니다.

💬 간단히 말하면

?? ? ? ? ? ? ? ? ? ? ? ? ?
?. ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?

W3 Button (?? → ? ? ? ?)

📍 위치

사이드바 행당 하나의 버튼..

🔧 기술적 설명

각 사이드바 항목은 현재 앵커를 설정하는 버튼이지만 시각적으로 목록 항목처럼 보입니다. 관찰자 변수는 그런 다음 0.3초에 걸친 부드러운 애니메이션으로 해당 앵커에 대한 스크롤 점프를 트리거합니다. 점프 후 앵커 값은 재설정됩니다 — 그렇지 않으면 동일한 앵커의 다음 클릭이 다시 발사되지 않을 것입니다 (값이 변경되지 않았기 때문에 관찰자가 다시 트리거되지 않음).

💬 간단히 말하면

?? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?

W4 ScrollView (???)

위치

오른쪽 열..

기술적 설명

Lazy 렌더링이 있는 스크롤 가능한 수직 스택 콘텐츠 영역. 더 긴 가이드는 쉽게 200개 이상의 Markdown 블록을 가질 수 있기 때문입니다 — 비 lazy 변형은 모두 동시에 인스턴스화하게 됩니다. 각 블록은 자체 ID를 받습니다. 헤딩 앵커 (점프 가능한 H1-H3용) 또는 인덱스 자리표시자. 최대 폭은 720 pt, 패딩 32 수평 / 24 수직이므로 긴 줄도 잘 읽을 수 있는 레이아웃을 유지합니다. 테이블은 수평 스택과 구분선과 함께 셀별로 렌더링되며, 인라인 코드는 내장 Markdown 엔진을 통해 렌더링됩니다. 실제 코드 블록은 현재 단락으로 처리됩니다 — Help 창의 알려진 제한입니다.

간단히 말하면

?? ???? ?????. ??
???, ???? ???? , ??
??????.

Keyboard Shortcuts (W5–W6)

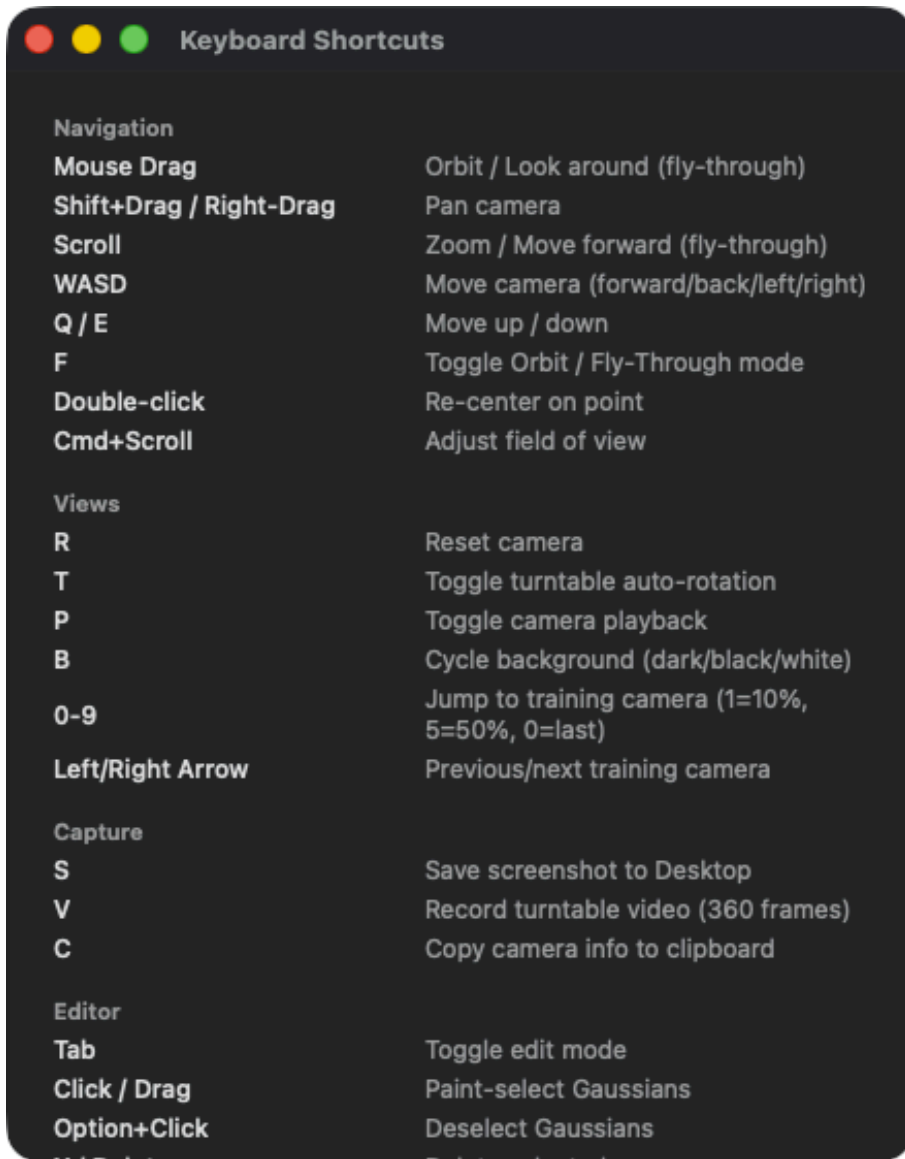


그림 18: Keyboard Shortcuts 창 — Navigation/Views/Capture/Editor/Training 5개 그룹, 왼쪽에 단축키 열, 오른쪽에 설명

이미지에 보이는 것 다섯 섹션의 정적 참조 목록입니다. **Navigation:** Mouse Drag (Orbit/Fly), Shift+Drag/Right-Drag (Pan), Scroll (Zoom), WASD (Fly Through 이동), Q/E (Up/Down), F (Toggle Orbit/Fly), Double-click (Re-center), Cmd+Scroll (FoV 조정). **Views:** R (Reset Camera), T (Auto-Rotation), P (Camera Playback), B (Background-Cycle), 0-9 (Training Cam으로 점프 1=10%/5=50%/0=last), Left/Right Arrow (Prev/Next Cam). **Capture:** S (Screenshot to Desktop), V (Turntable Video), C (Copy Camera Info). **Editor:** Tab (Edit Mode), Click/Drag (Paint-Select), Option+Click (Deselect), X / Delete (선택 삭제), Cmd-Z (마지막 삭제 되돌리기), [/] (브러시 크기 작게/크게), Esc (선택 해제). **Training:** Start, Pause/Resume, Cancel, Continue +5K/+10K/+20K는 M9-M14의 메뉴 단축키를 통해.

그것이 무엇인지: 모든 단축키의 단순한 정적 개요 — Navigation, Views, Capture, Editor, Training. 콘텐츠는 하드 코딩되어 있으며, Markdown 로딩은 없습니다.

언제 열까 뷰포트에서 어떤 것을 가장 빠르게 하는 방법을 찾을 때. WASD Fly Through, R로 Camera Reset, B로 Background Cycling — 모두 여기에 있습니다.


W5 ScrollView (???) (???)

 위치

Help → Keyboard Shortcuts (⌘)..

 기술적 설명

수직 목록을 포함하는 단순한 스크롤 영역입니다. 주위에 패딩 20, 사이드바 내비게이션 트리 없음 (목록이 충분히 짧음). 콘텐츠는 다섯 섹션 (Navigation, Views, Capture, Editor, Training) 으로 그룹화되어 있습니다. 키 조합당 두 열의 번역 가능한 텍스트가 있는 하나의 줄. 왼쪽 열 (키 코드) 은 180 pt 폭으로 고정되어 오른쪽 설명이 수직 정렬되어 유지됩니다. 스크롤 외에 상호작용 없음 — 줄 클릭은 아무것도 트리거하지 않습니다. 단축키는 메뉴와 뷰포트의 실제 키보드 수정자입니다.

 간단히 말하면

?? ?????. ??
?? ????.


W6 VStack (???) (???)

 위치

ScrollView 내부..

 기술적 설명

16 pt 간격으로 왼쪽 정렬된 스택형 섹션. 다섯 섹션 내에 각각 헤딩 + 줄 시퀀스. 헤딩은 보조 Subheadline 스타일을 사용합니다 — 의도적으로 Title 형식이 아닙니다. 섹션이 내비게이션 가능할 필요가 없기 때문입니다. 콘텐츠는 의도적으로 평평합니다 (Disclosure 없음, Search 없음, Filter 없음). 모든 macOS 버전에서 변경 없이 컴포넌트가 실행되고 파일이 읽을 수 있도록 유지됩니다.

 간단히 말하면

?? ? ? ? (Navigation, Views, Editor ?).

Manage Storage (W7-W12)

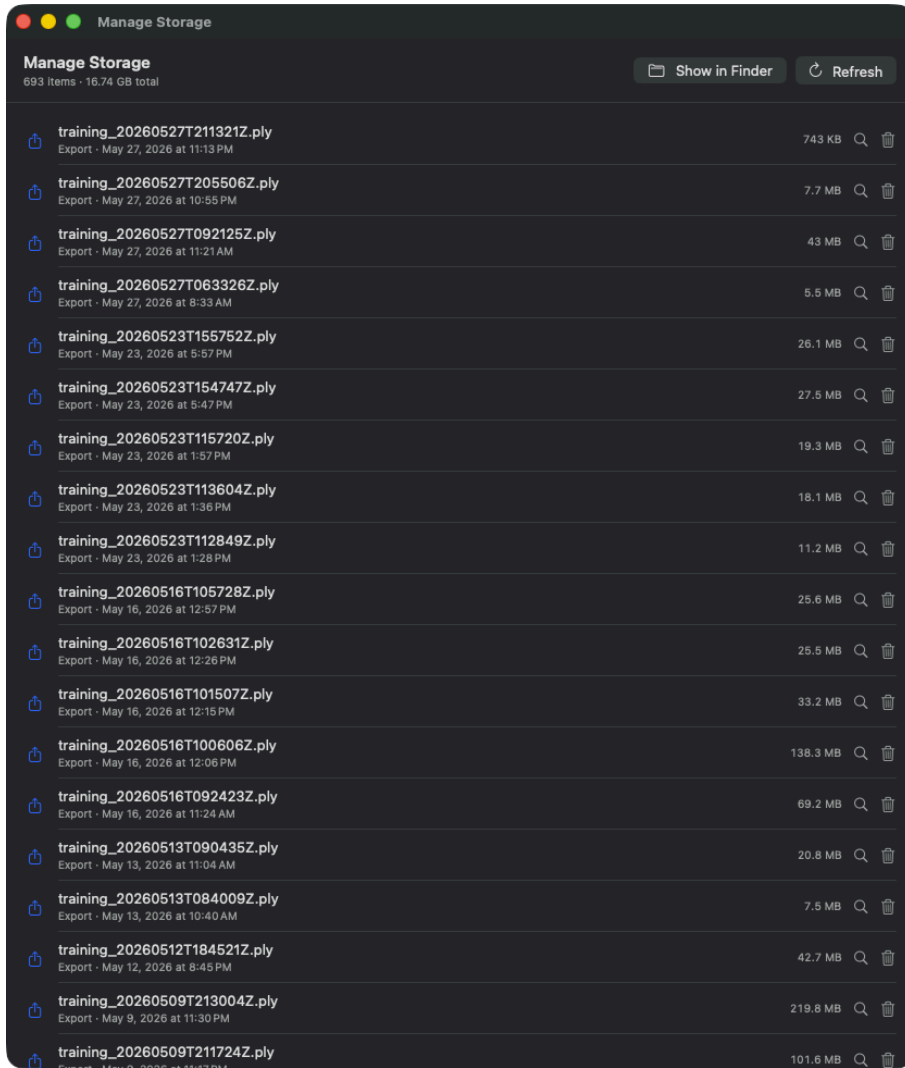


그림 19: Manage Storage 창 — 헤더 “693 items · 16.74 GB total”, 날짜순 정렬된 Export PLY 파일 테이블, 각각 포맷 핀 + 파일 이름 + 크기 + 날짜

이미지에 보이는 것 RadianceKit이 관리하는 모든 파일의 테이블 뷰. 헤더는 693개 항목, 16.74 GB 총 크기를 계산합니다. 위 툴바: “Show in Finder” + “Refresh”. 각 줄: PLY 아이콘, 파일 이름 (예: training_20260527T211321Z.ply), 내보내기 날짜, 크기 (7 KB에서 218 MB까지 변동), 돋보기 아이콘 (Reveal), 휴지통 아이콘 (Move to Trash). 파일은 날짜별로 정렬되며, 최신이 위입니다. 이 데모 캡처에서는 `--benchmark` 로 많이 작업했기 때문에 PLY 내보내기가 지배적입니다.

그것이 무엇인지: RadianceKit이 `~/Documents/RadianceKit/` 아래에 저장하는 모든 것에 대한 디스크 사용량 개요 — Logs, Exports, Scenes, Capture 번들 (iOS 동반자에서), Imports (입력 이미지의 스테이징 복사본). 항목당 바이트 크기와 두 버튼: “Finder에서 보기”와 “휴지통으로 이동”. 자동 정리는 **아님** — 앱은 스스로 아무것도 삭제하지 않습니다. 항목별로 결정합니다.

연제 열까 디스크가 가득 찰 때. 특히 로그가 쌓입니다 (학습 시도당 하나의 JSONL, 그리고 `_qualityMetrics.json`). 내보내기도 마찬가지입니다 (PLY 100% 원시 데이터, 내보내기당 하나). 크래시 후에도 유용합니다. Imports 스테이징 디렉터리에 여전히 입력 이미지의 오래된 복사본이 있을 경우 (“Disk-pressure incident” 참고: [dev_v549f-needle-reduction.md](#)).

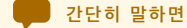
W7 **“Show in Finder”**



Storage 브라우저 창 위 오른쪽 헤더..



전체 RadianceKit 디렉터리 (~/Documents/RadianceKit/) 를 Finder에서 열어 폴더 구조를 직접 보고 Finder 자체로 도 조작할 수 있게 합니다. 동작은 새 Finder 창을 열고 앱 Sandbox 컨테이너로 전환하지 않습니다 — ~/Documents/RadianceKit/는 앱에 정기적으로 접근 가능한 Documents 도메인이며, Sandboxed 컨테이너 경로가 아닙니다.



Finder [?][?] [?][?][?][?] [?][?][?] [?][?][?][?][?][?][?][?].

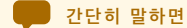
W8 **“Refresh”**



헤더, Finder 버튼 옆..



사용자가 초기화한 비동기 작업에서 실행되는 백그라운드 스캔을 트리거합니다. 큰 디렉터리 트리를 스캔하는 것이 UI를 차단하지 않도록 합니다. 실제 워크는 모든 알려진 하위 폴더 (Logs, Exports, Scenes, Captures, Imports) 를 통과하여 각 직접 자식마다 Storage 항목을 생성합니다. 항목당 재귀적 크기가 결정됩니다 — 실제 디스크 사용량 (APFS Hardlink 공유 포함) 우선, 논리적 파일 크기를 대체로 사용.



[?] [?] [?] [?] Finder [?][?] [?][?][?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?].

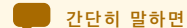
W9 **List (Storage [?][?])**



헤더 아래 메인 콘텐츠..



이 레이아웃의 줄당 목록: 카테고리별 SF Symbol 아이콘 (로그용 문서, 내보내기용 업로드 화살표, Scene용 큐브, Imports용 Tray), 이름 + 부제목 (Kind 레이블 + 형식화된 수정 날짜), 오른쪽 바이트 카운터 (오른쪽 정렬, monospaced), Reveal 버튼 (돋보기 기호), Trash 버튼 (휴지통). 정렬: 주로 Kind 별 (Scene 먼저, 그 다음 Exports, Logs, Captures, Imports, Other), 부차적으로 수정 날짜 내림차순 (최신이 위). 스캔이 아직 실행 중이면 그 위치에 “Scanning...” 진행이 대신 표시됩니다. 아무것도 발견되지 않으면 Tray 아이콘과 함께 비어 있는 상태 표시.



[?][?] RadianceKit [?][?][?][?] [?][?], [?][?][?] [?][?][?][?] [?][?][?][?][?]. [?][?][?] [?][?][?] [?] [?] [?] [?] [?] [?][?][?] [?] [?][?][?][?].

Pareto Dashboard (W13–W22)

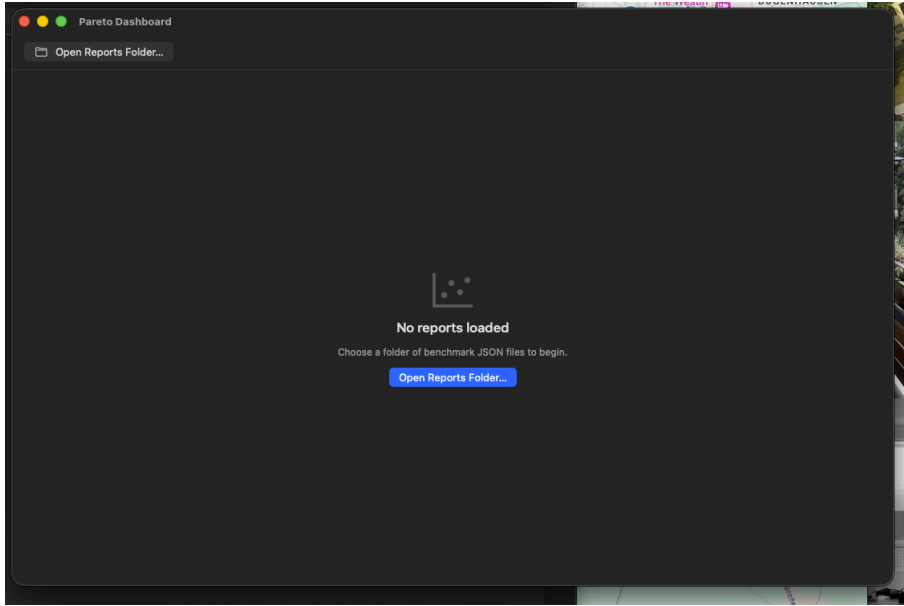


그림 20: Pareto Dashboard — 리포트 가져오기 전 비어 있는 상태

비어 있는 상태 (처음 열린 후) — “Open Reports Folder...” 호출 동작이 있는 비어 있는 상태. 학습 리포트가 로드되면 데이터 포인트가 나타납니다. 다음 캡처 참고.

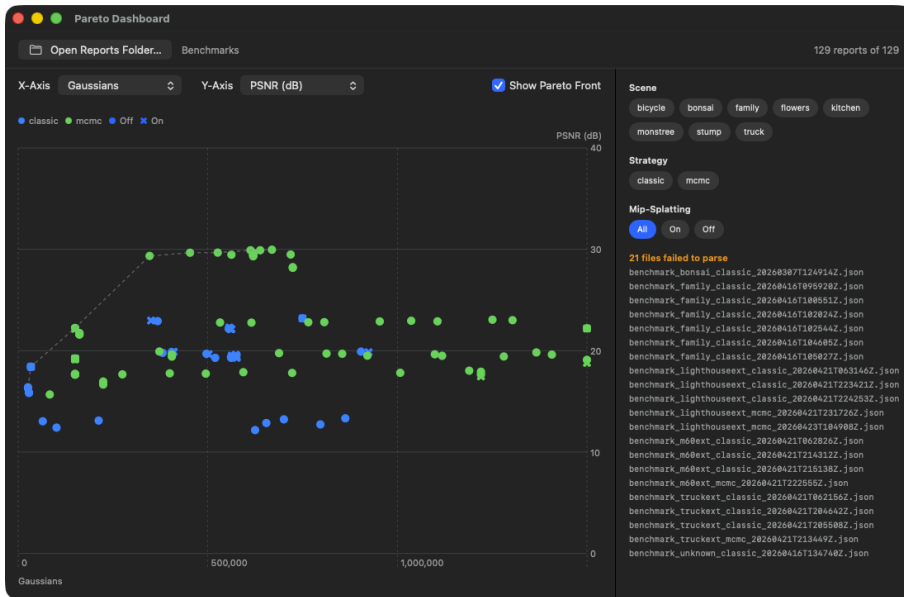


그림 21: 129개 벤치마크 리포트가 로드된 Pareto Dashboard — Gaussians vs PSNR, Pareto Front, Scene/Strategy/Mip 필터

이미지에 보이는 것 헤더 툴바에 “129 reports of 129” 표시 (선택한 폴더의 모든 리포트가 성공적으로 파싱됨 — 추가 21개 파일은 더 오래된 형식 때문에 파싱되지 않을 수 있음, 오른쪽 힌트 목록 참고). 축: X축 선택기 Gaussians, Y축 선택기 PSNR (dB). 산점도: 녹색 점 = Classic 전략, 파란색 점 = MCMC. 점선 Pareto Front 선은 가장 좋은 PSNR 값을 따라 진행하며 약 500K Gaussian부터 PSNR≈30 dB에서 안정화됩니다. 오른쪽 필터 칩: 7개의 Scene (bicycle, bonsai, family, flowers, kitchen,

stump, truck), 2개의 Strategy (classic, mcmc), 3개의 Mip Splatting 옵션 (All, On, Off). 현재 모든 필터가 열려 있으므로 조밀한 점 클러스터입니다.

그것이 무엇인지: 다중 실행 비교 도구입니다. 과거에 여러 장면 또는 다른 프리셋으로 동일한 장면을 학습했습니다 — 각 학습 실행은 (`--benchmark` 를 함께 제공했거나 Benchmark 기능을 통해 호출한 경우) JSON 리포트 파일을 생성합니다. 그것은 무엇보다 최종 PSNR, SSIM, LPIPS, Gaussian Count, Wallclock 시간을 포함합니다. Dashboard는 이러한 리포트의 전체 폴더를 동시에 읽고, 선택 가능한 축이 있는 2D 산점도로 플로팅합니다. 또한 Pareto Front (비지배된 점의 집합) 가 점선으로 표시됩니다.

언제 열까 최소한 세 개 또는 네 개의 학습 리포트를 만든 후. 점이 적으면 Frontier 선이 의미가 없습니다. 일반적인 사용 사례: 야외 장면을 재구성하려 시도했고 차례로 P3 Balanced (Classic), P4 Quality (Classic), P7 MCMC Quality, P9 Outdoor (tuned) 를 시도했습니다 — 이제 어느 구성이 학습 시간 초당 최고의 PSNR을 제공하는지, 또는 어느 것이 주어진 PSNR에 대해 가장 적은 Gaussian을 필요로 하는지 알고 싶습니다.

어떻게 해석할까 두 축 모두 자유롭게 선택할 수 있습니다 (X축: GaussianCount, trainingTime, psnr, ssim, lpips, ...; Y축 동일). ParetoFront2D.indices의 Pareto Front 로직은 각 메트릭에 대해 “작을수록 좋음” (예: LPIPS, Loss, Time) 또는 “클수록 좋음” (PSNR, SSIM) 을 알고 있습니다 — 따라서 선은 축 선택에 따라 왼쪽 아래에서 오른쪽 위로 또는 왼쪽 위에서 오른쪽 아래로 항상 도달한 최선의 조합을 따라 진행합니다. 다른 점이 두 차원 모두에서 최소한 같지 않으면 점은 Pareto 최적입니다 (즉, 다른 어떤 것도 그것을 지배하지 않음). Pareto 최적 점은 선 위에 있고, 다른 점은 그 오른쪽/위 (축 방향에 따라) 에 있습니다. 선 위의 점은 “최고의 프리셋”에 대한 진정한 후보입니다. 선에서 멀리 떨어진 점은 낭비된 학습 시간입니다.

필터 칩 선택을 특정 장면으로 제한할 수 있고 (예: 야외 실행만 비교하려는 경우), 특정 전략 (Classic 또는 MCMC) 으로 제한하거나 Mip-Splatting on/off로 제한할 수 있습니다 (Phase Q1.5 후 관련 — Mip이 opt-in 고급 플래그로 남음).

예제 워크플로 ~/Documents/RadianceKit/Reports/에 “truck” 장면용 세 개의 리포트가 있습니다: Run A (P4 Quality, 40K iter, 524K Gs, 105 s, PSNR 23.4), Run B (P7 MCMC, 200K iter, 150K Gs, 693 s, PSNR 24.6), Run C (P9 Outdoor, 100K iter, 1.25M Gs, 312 s, PSNR 25.8). X축을 trainingTime, Y축을 PSNR로 설정합니다. Run B는 오른쪽 위, Run C는 더 오른쪽 위, Run A는 왼쪽 아래에 있습니다. Pareto Front는 A와 C를 연결합니다 — 둘 다 비지배. Run B는 “lost”입니다 (C가 Time과 PSNR 모두에서 더 좋음). 통찰: “truck”의 경우 MCMC 기본값은 가치가 없습니다. 빠름+ok (A) 또는 길다+매우 좋음 (C). C의 구성을 자체 프리셋으로 저장 (인스펙터 → 11 Save Preset).

다음 동작: 최고 구성을 프리셋으로 저장. 구체적으로: Pareto 점을 살펴봅니다 (호버는 도구 설명에서 PSNR/SSIM/LPIPS/Gs/Time을 보여 줍니다), 시간 vs 품질 트레이드오프가 가장 잘 맞는 것을 결정하고, 관련 리포트를 열고 (파일 이름에 실행 타임스탬프 포함), 새 실행에서 학습 구성을 복사하거나 다음 학습 세션 후 인스펙터를 통해 프리셋으로 저장합니다.

W13 [?] [?] "Open Reports Folder..."

📍 위치

툴바 위 왼쪽..

🔧 기술적 설명

"Select a folder containing benchmark .json reports" 요청과 함께 폴더 선택 대화 상자를 엽니다. 확인 후 백그라운드 작업이 폴더의 모든 .json 파일을 순차적으로 파싱합니다. 잘못된 리포트 (손상된 JSON, 잘못된 스키마) 는 수집되어 사이드바 아래에 "N file failed to parse"로 표시됩니다 — 크래시 없음. 첫 번째 로드가 아직 실행되는 동안 두 번째 클릭이 발생하면 이전 작업이 중단되어 두 결과가 동시에 상태에 쓰이지 않도록 합니다.

CLI를 통해서도: --dashboard-dir /path/to/reports는 앱 시작 시 폴더를 바로 로드합니다.

🗨 간단히 말하면

```
? ? ? ? ? ? ? ? ? ? ? ? ? ?  
? ? ? ? ? ? ? ? ? ? . ? ? ? ? ?  
? ~/Documents/RadianceKit/  
Reports/? ? ? ? . ? ? ? ? ? ?  
? JSON ? ? ? ? ? ? ? ? ? ? ? .
```

W14 [?] [?] [?] "X-Axis"

📍 위치

차트 위, 왼쪽..

🔧 기술적 설명

Dashboard 모듈의 사용 가능한 모든 메트릭 축 (PSNR, SSIM, LPIPS, Gaussian Count, 학습 시간 등) 이 있는 메뉴 선택기. 기본값은 Gaussian Count. 전환 시 호버된 점이 재설정됩니다. 이전에 하이라이트된 위치가 새 축 좌표계에서는 의미가 없기 때문입니다. 선택기는 콘텐츠 폭으로 제한되어 전체 폭에 걸쳐지지 않습니다.

🗨 간단히 말하면

```
? ? ? ? ? ? ? ? ? ? ? ? ? ?  
? ? ? ? ? ? ? ? ? ? " ? ? ? ? ? ? " ? ? ?  
"Gaussian ? ?" — ? ? ? ? ? ? ? ? ? " ?  
? " ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? .
```

W15 [?] [?] [?] "Y-Axis"

📍 위치

차트 위, X-Axis 옆..

🔧 기술적 설명

W14와 동일하지만 기본값이 PSNR입니다. 축 선택은 독립적으로 저장되므로 사용자가 무의미한 조합 (X=PSNR, Y=PSNR — 모든 점을 대각선에 던짐) 도 선택할 수 있습니다. 그러나 그러한 조합은 포착되지 않습니다. 의도적인 결정입니다. "SSIM vs PSNR" 비교가 메트릭이 얼마나 일관되게 동작하는지 보기 위해 흥미롭기 때문입니다.

🗨 간단히 말하면

```
? ? ? ? ? ? ? ? ? ? ? ? ? ?  
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? "PSNR"  
? ? ? "SSIM".
```

W16 [?] [?] "Show Pareto Front"

위치
축 선택기 오른쪽..

기술적 설명

표준 macOS 토글입니다. 활성이면 Pareto 차트에 점 구름 외에 계산된 2D Pareto Front가 있는 선이 그려집니다. 스타일: 점선 (대시 패턴 4-4), 회색 반투명, 선 두께 1.5 pt. Pareto 계산은 메인 스투드에서 실행됩니다 — 일반적인 리포트 수 (≤ ~50) 의 경우 문제없이 빠릅니다. 토글이 꺼져 있으면 선이 생략되어 벗은 점만 표시됩니다.

간단히 말하면

"[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]" [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]

W17 [?] "Scene" [?] [?]

위치
Dashboard 창 오른쪽 사이드바..

기술적 설명

로드된 리포트에서 발생하는 각 장면에 대한 필터 칩. 칩이 소진되면 자동으로 칩을 여러 줄로 다시 패키징하는 자체 Flow Layout. 활성 칩은 강조 배경을 받고, 비활성 칩은 중립 표준 Material 배경을 받습니다. 다중 선택 가능 (Set 의미론). 칩이 선택되지 않은 경우 모든 장면이 "통과"되는 것으로 적용됩니다 — 즉 Set 로직은 "빈 선택 = 모두"이며, "빈 선택 = 없음"이 아닙니다.

간단히 말하면

[?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?]

W18 [?] "Strategy" [?] [?]

위치
사이드바의 Scene 필터 아래..

기술적 설명

W17과 정확히 같지만 학습 전략용 — 일반적으로 두 값 "classic"과 "mcmc". 벤치마크 리포트 JSON의 Strategy 필드에서 파생됩니다. 두 전략의 리포트를 혼합했고 한 종류만 보고 싶을 때 (예: "Classic을 이미 제외했기 때문에 MCMC 실행만 표시") 유용합니다.

간단히 말하면

Classic [?] [?] MCMC [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]

W19 ? "Mip-Splatting" ? ?

📍 위치

사이드바의 Strategy 필터 아래..

🔧 기술적 설명

3값 필터 (W17/W18처럼 Set이 아님): "All" / "On" / "Off".
배경: Mip Splatting은 Phase Q1.5에서 실험적 다중 스케일 개선으로 평가되었고 최종 verdict는 "내내 좋은 win 없음, opt-in 플래그로 유지"였습니다. Mip on/off 비교를 하는 경우 종종 매우 선명하게 분리할 수 있기를 원합니다. 따라서 "모두 통과", "Mip on만", "Mip off만" 상태가 있는 전용 3진 필터가 있습니다. 사이드바 섹션은 데이터 집합에 최소 하나의 Mip 리포트와 최소 하나의 비 Mip 리포트가 있을 때에만 표시됩니다 (그렇지 않으면 필터링이 의미가 없음).

💬 간단히 말하면

Mip Splatting on/off ? ? ? ? ?
? ? ? ? ? 3 ? ? ? ? . ? ? ? ? ?
? ? ? ? ?.

W20 ChipButton (? ? ? ? , all/on/off)

📍 위치

Helper 컴포넌트, W17/W18/W19에서 사용됨..

🔧 기술적 설명

최소한의 버튼 래퍼입니다. 콘텐츠: Caption 글자 크기와 패딩 10 수평 / 5 수직의 레이블 텍스트. 배경은 조건적: 활성화일 때 → 흰색 텍스트가 있는 앱 강조 색상. 그렇지 않으면 검은색 텍스트가 있는 중립 표준 Material 배경. 모양은 Capsule (알약 같은). Plain 버튼 스타일로 Capsule Material이 시스템 테두리에 의해 덮이지 않도록 합니다.

💬 간단히 말하면

? ? ? ? ? ? ? ? . ? ? ? ? ?
? iOS ? ? ? ? ? ? ? ? .

W21 Chart (Pareto Scatter)


 위치

Dashboard의 중앙 영역..

 기술적 설명

두 레이어가 있는 Swift Charts 다이어그램: 1. 리포트당 하나의 점 — 선택된 X 및 Y 메트릭에서의 위치, Strategy별 색상, Mip 상태별 기호. 기호 크기는 보통 80, 강조 표시되었을 때 200 (ID가 현재 호버된 리포트와 일치하는 경우). 2. Pareto Front용 선, 토클이 켜져 있을 때만.

차트 오버레이: 투명 직사각형이 마우스 움직임을 등록합니다. 프레임당 플롯 프레임에서 유클리드 가장 가까운 점 위치가 결정되고 거리가 24 px 미만이면 호버된 리포트가 업데이트됩니다 (그렇지 않으면 재설정). 따라서 클릭하지 않고도 도구 설명을 받을 수 있습니다 — 호버로 충분합니다.

 간단히 말하면

?? ? ? ? ? . ? ? ? ? ? ? ? ?
?? ? ? . ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? .


W22 Tooltip (?? ? ? ? ?)

 위치

차트 아래, 호버 시 표시..

 기술적 설명

수평 스택: 장면 이름 (Headline), Strategy 태그 (Caption), 구분선, 그런 다음 PSNR/SSIM/LPIPS/Gs/Time 메트릭을 각각 작은 수직 그룹 (레이블 + monospaced 값) 으로. Mip이 활성화되었다면 추가로 강조 색상의 "Mip" Capsule 태그. 배경은 반투명 블러, 8 pt 반경의 둥근 직사각형. 마우스가 실제로 점 위에 있을 때에만 표시됩니다. 떠나면 자동으로 사라집니다.

 간단히 말하면

?? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? . ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? .

Holdout Analysis (W23–W29)

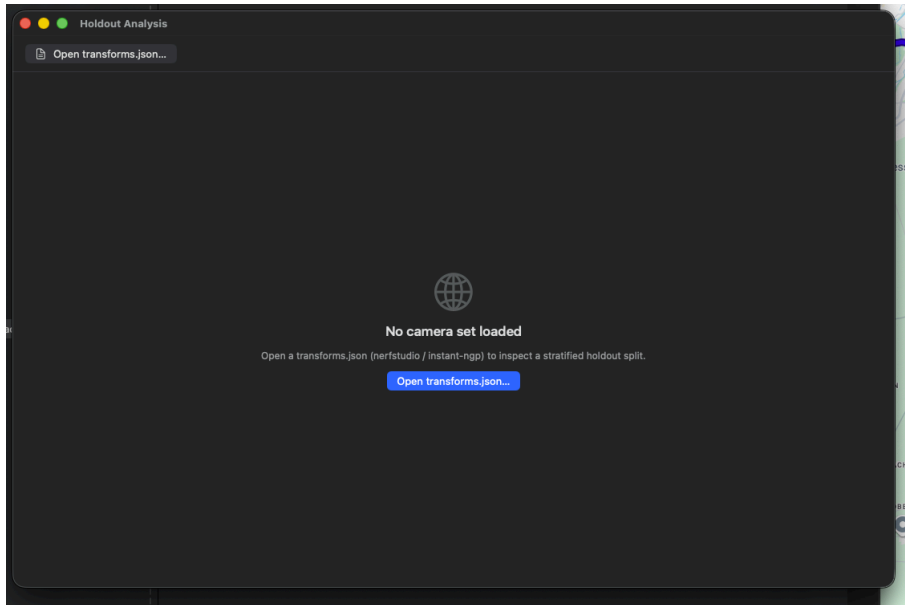


그림 22: Holdout Analysis — transforms.json 로드 전 비어 있는 상태

“Open transforms.json...” 호출 동작이 있는 비어 있는 상태와 빈 상태. NeRF Studio 및 Instant NGP 형식을 수용합니다.

비어 있는 상태 (처음 열린 후) — transforms.json 이 로드되면 카메라 마커가 나타납니다. 다음 캡처 참고.

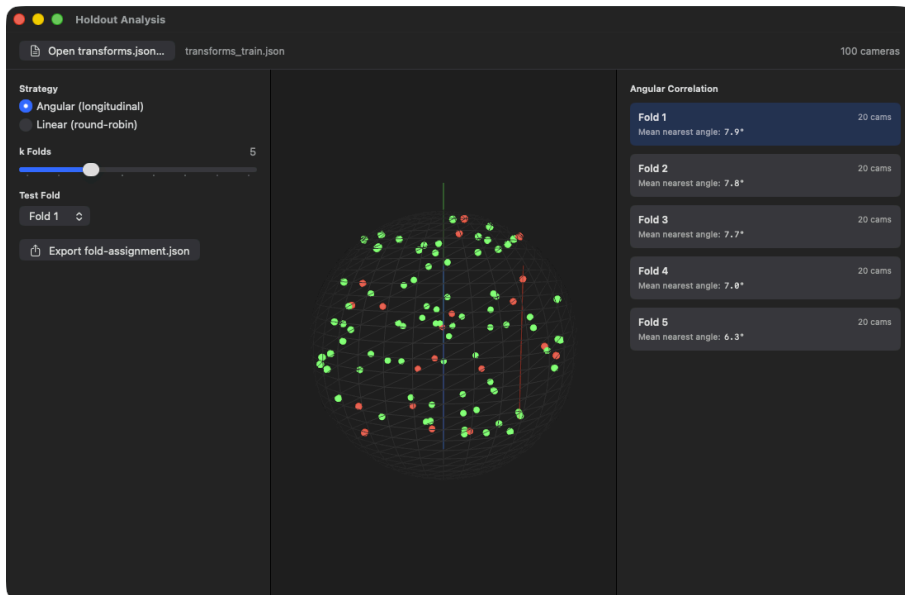


그림 23: 100개의 NeRF Blender Mic 카메라가 있는 Holdout 글로브, 5개 폴드 × 20개 카메라, Angular 전략 활성화

이미지에 보이는 것 헤더는 로드된 파일 (transforms_train.json) 과 Cam Count (“100 cameras”) 를 표시합니다. 왼쪽 사이드바: 두 옵션이 있는 Strategy 선택기 — Angular (longitudinal) 활성화 (구체의 경도/위도 섹터에 따라 폴드를 정렬하므로 각 테스트 폴드가 기하학적으로 조밀함) vs Linear (round-robin) (순서 기반, k번째 프레임마다 테스트 세트). k 폴

드 슬라이더는 5, 테스트 폴드 선택기는 Fold 1. Export 버튼은 Nerfstudio/Instant NGP용 `fold-assignment.json`을 생성합니다. 중앙 패널: 모든 100개 카메라의 3D 글로벌 투영 — 녹색 점 = Train, 빨간색 점 = 현재 테스트 폴드 (20개 카메라가 있는 Fold 1). 오른쪽 사이드바 (Angular Correlation): 폴드당 20개 Cams + Mean Nearest Angle (Fold 1: 7.9°, Fold 2: 7.8°, Fold 3: 7.7°, Fold 4: 7.0°, Fold 5: 6.3°) — 작은 값은 이 폴드 내의 카메라가 서로 가까이 있다는 것을 의미하므로 Holdout 분할이 공간적으로 일관됩니다.

그것이 무엇인지: Cross Validation 로직이 있는 카메라 배치를 위한 3D 시각화 도구.

`transforms.json` (Nerfstudio / Instant NGP의 카메라 포즈용 표준 형식) 을 로드하면 앱은 모든 카메라를 읽고 그 시선 방향을 단위 구체에 투영하고 가상 글로벌에 작은 구체 마커로 표시합니다. 그런 다음 카메라를 (선택된 전략에 따라: angular 또는 linear) k 폴드로 나누고, 학습 부분을 녹색으로, 테스트 부분 (Holdout) 을 빨간색으로 표시하고, 테스트 폴드가 시선 각도 공간에서 학습 폴드로부터 얼마나 멀리 떨어져 있는지를 알려 주는 Angular Correlation 점수를 폴드당 계산합니다.

언제 열까 Holdout 평가를 하고 싶을 때 — 즉: 모델이 보이지 않는 시점에서 얼마나 잘 일반화되는가? 학습의 표준은 “every-8th view를 Holdout으로” (Mip-NeRF360 규약) 이지만 이는 매우 선형적인 분할입니다. 예를 들어 이미지가 시간적으로 클러스터링된다면 (먼저 객체의 한 쪽, 그 다음 다른 쪽), “every-8th”는 대표적이지 않습니다 — 무작위 시퀀스 위치가 테스트에 들어가지만 모든 이웃은 학습에 있어 너무 쉽습니다. “angular”로는 대신 시선 각도 공간에 걸쳐 계층화됩니다: 각 폴드는 궤도의 모든 영역에서 카메라를 포함하므로 테스트가 실제로 일반화 격차를 테스트합니다.

어떻게 해석할까 Angular vs Linear: - Angular (표준): 카메라를 경도 각도 (Y축 주위의 ϕ 좌표) 에 따라 k 개의 동일한 섹터로 나눕니다. Fold 0은 $\phi \in [0^\circ, 360/k^\circ)$ 의 카메라, Fold 1은 그 다음, 등입니다. 장점: 각 폴드가 궤도의 일부를 다루며, 테스트 폴드는 공간적으로 콤팩트하지만 전체 데이터셋에 걸쳐 넓습니다. 고전적 궤도 촬영에 좋습니다. - Linear (Round-Robin): Fold-Index = (image_index modulo k). 단순한 “every-k-th” 분할입니다. 이미지 순서가 공간적 편향이 없을 때 동작합니다 (예: 무작위로 정렬된 드론 촬영). 이미지가 시간적으로 클러스터링되면 잘 동작하지 않습니다.

3D 글로벌에서 즉시 볼 수 있습니다: 녹색 점 (Training) 과 빨간색 점 (Test). 빨간색 점이 모두 한 구석에 클러스터되면 Holdout이 나뉩니다 (좋은 일반화 테스트가 아님). 녹색 사이에 균등하게 있으면 좋습니다. 폴드당 Angular Correlation 점수 (오른쪽 사이드바, 도 단위) 는 추가로 말합니다: 작은 값 = 테스트가 학습에 가까움 (각 테스트 카메라가 가까운 학습 카메라를 가짐, 쉬운 테스트). 큰 값 = 테스트가 학습에서 멀리 떨어짐 (더 어려운 일반화).

예제 워크플로 Truck 장면을 251개의 이미지로 촬영했고 메뉴 항목 M33 (Export SfM `transforms.json`) 을 통해 nerfstudio 파일을 내보냈습니다. Holdout 창 (순서H) 을 열고 “Open transforms.json...”을 통해 JSON을 로드하고 글로벌을 살펴봅니다. $k=5$ (기본값) 는 5개의 폴드를 제공합니다. “Fold 3”을 클릭하고 — 빨간색 마커가 어느 정도 균등한지 확인합니다. 그렇다면: “Export fold-assignment.json”, 내보낸 파일을 Reports 폴더에 두고, 다음 학습 실행에서 `--benchmark` (또는 해당 인스펙터 설정) 와 함께 정확히 이 폴드 분할이 테스트 Holdout으로 사용됩니다 — 표준 “every-8th” 대신.

W23 **“Open transforms.json...”**

위치
툴바 위 왼쪽..

기술적 설명

JSON 파일로 제한된 파일 선택 대화 상자를 엽니다. 확인 후 Holdout 모듈이 파일을 로드합니다. 로더는 nerfstudio 형식 (카메라 인트린식과 이미지 경로 및 변환 행렬이 있는 프레임 목록) 과 instant-ngp 형식 (같은 구조) 을 모두 파싱합니다. 프레임당 시선 방향이 변환 행렬에서 추출되고 (카메라 로컬 베이스의 z축) 저장됩니다. 파싱이 실패하면 상태 영역에 오류 메시지가 표시됩니다.

CLI를 통해서도: --holdout-file /path/to/transforms.json은 로드된 파일과 함께 창을 직접 시작합니다.

간단히 말하면

JSON Nerfstudio Instant NGP. RadianceKit Menu → Export → SfM transforms.json.

W24 **“Strategy” (angular/linear)**

위치
왼쪽 사이드바, 위..

기술적 설명

두 옵션이 있는 라디오 선택기: Angular와 Linear. 전략 변경은 폴드의 자동 재계산을 트리거합니다. 시선 방향은 구체의 3D 단위 벡터 목록입니다. Angular 전략은 그것들을 경도 각도 ϕ 에 투영하고 정렬합니다. Linear 전략은 단순히 프레임 인덱스에 대한 모듈로 분할을 수행합니다.

간단히 말하면

Angular (Angular), Linear.

W25 **“k Folds”**

위치
왼쪽 사이드바, 중간..

기술적 설명

3에서 10까지의 슬라이더, 단계 1. 변경 시 폴드 계산이 자동으로 다시 시작되어 폴드 목록, 학습/테스트 인덱스, 폴드 별 점수가 즉시 다시 계산됩니다. 선택된 값은 레이블 옆에 monospaced 자릿수 텍스트로 표시됩니다.

경험 법칙: k=5가 표준 (폴드당 20% 테스트 제공, Cross Validation에 일반적). 데이터가 매우 많고 통계적 신뢰성을 위해 더 많은 폴드가 필요하면 k=10. 데이터가 적으면 k=3.

간단히 말하면

5.

W26 [?] [?] [?] "Test Fold"

위치

왼쪽 사이드바, k 슬라이더 아래..

기술적 설명

메뉴 선택기. 옵션은 동적으로 $0..<k$, 레이블 "Fold 1" 에서 "Fold N"까지 (UI에서 1 인덱스, 내부에서 0 인덱스). 이전에 선택된 인덱스가 $\geq k$ 이면 (예: k를 10에서 5로 줄였기 때문에) 자동으로 0으로 재설정됩니다. 선택된 테스트 폴드는 글로브에서 빨간색으로 표시되고 다른 모든 폴드는 녹색으로 표시됩니다.

간단히 말하면

[?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
?. [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?].

W27 [?] [?] "Export fold-assignment.json"

위치

왼쪽 사이드바, 아래..

기술적 설명

기본 파일 이름 fold-assignment.json이 있는 저장 대화 상자를 엽니다. 확인 후 Holdout 모듈이 현재 분할을 JSON 스키마로 인코딩합니다 (프레임별 폴드 할당과 Strategy 메타 블록). 이 파일은 다음 학습에서 --benchmark 와 함께 제공될 수 있어, 동일한 Holdout이 최종 메트릭 평가에 사용됩니다. 쓰기 오류는 오류 텍스트로 표시됩니다. 성공은 녹색 텍스트로 "Saved to (filename)"로 표시됩니다.

간단히 말하면

[?] [?] Train/Test [?] [?] [?] JSON [?]
[?] [?] [?] [?] [?] [?]. [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?], [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?].

W28 SCNView (3D Camera Globe)



Holdout 창의 중앙 패널..



SceneKit 글로브 뷰. 장면은 다음으로 구성됩니다: 와이어프레임 구체 (반경 1.0, 36 세그먼트, 짙은 회색), 세 개의 색이 있는 축 stub (X/Y/Z용 빨강/녹색/파랑, 각 1.2 길이), 그리고 카메라 당 단위 구체의 해당 시선 방향 위치 (와이어프레임 구체 안으로 사라지지 않도록 약간 외부) 의 작은 마커 구체 (반경 0.03). 마커는 각 폴드 변경 시 재구성되지 **않습니다** — 재구성은 프레임 목록이 변경될 때 (즉 새 JSON이 로드될 때)에만 필요합니다. 대신 업데이트마다 재료 색상의 인플레이스 업데이트가 실행됩니다: 테스트 인덱스용 빨강, 학습용 녹색, 둘 다 아닌 경우 밝은 회색. 따라서 N > 1000 카메라에서도 슬라이더 킵이 성능적으로 유지됩니다.

카메라 제어가 활성화되어 있습니다 — 마우스로 글로브를 회전하고 줌하고 팬할 수 있습니다. 조명은 마커가 평평하게 보이지 않도록 합니다. 배경은 짙은 회색입니다.

간단히 말하면

3D
 = Training, = Test,
 = (,
).
 ?
 ?

W29 FoldCard ()



오른쪽 사이드바, "Angular Correlation" 섹션..



폴드당 하나의 카드 뷰 — 6 pt 반경의 둥근 직사각형, 패딩 10, 두 줄의 수직 레이아웃 (위 "Fold N" + 카메라 수, 아래 "Mean nearest angle:" + 도 단위 값). 배경색은 조건적: 활성 폴드 = 반투명 강조 색상, 비활성 = 중립 표준 Material. 탭 하면 폴드가 선택되고 글로브가 실시간으로 다시 칠해집니다.

"Mean nearest angle" 점수는 가장 가까운 학습 카메라에 대한 테스트 카메라당 평균 가장 작은 각도입니다 (내부적으로 라디안으로 계산, UI에서 도 단위로 표시).

간단히 말하면

?
 ?
 ?
 ?
 ?

BayesOpt Console (W30-W39)

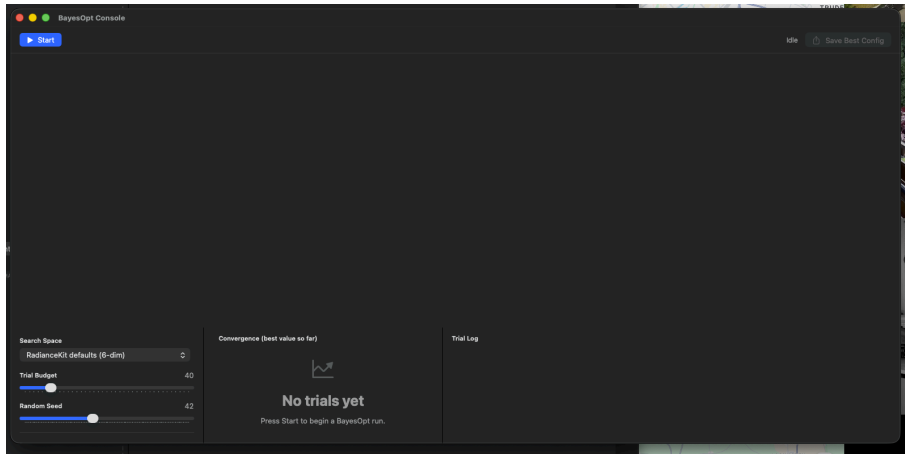


그림 24: BayesOpt 콘솔 — 시행 시작 전 비어 있는 상태

탐색 공간 선택기 (RadianceKit defaults (6-dim)), 시행 예산 슬라이더 (기본 40), 무작위 Seed (42), 그리고 Convergence 차트, Trial Log, 탐색 공간 매개변수 목록을 위한 세 개의 비어 있는 패널이 있는 비어 있는 상태.

비어 있는 상태 (처음 열린 후) — 실행이 시작되면 Convergence 차트와 시행 테이블이 채워집니다. 다음 캡처 참고.



그림 25: 40번의 시행 후 BayesOpt 콘솔 — Convergence 차트는 시행 15까지 가파르게 상승, Best Value 0.9943, Trial Log에 init/bo/restart 태그

이미지에 보이는 것 위 오른쪽 상태 "Finished — best 0.9943 after 40 trials". 왼쪽 사이드바: 탐색 공간 선택기 RadianceKit defaults (6-dim), 시행 예산 40, 무작위 Seed 42. 매개변수 목록은 튜닝할 여섯 하이퍼파라미터를 값 범위와 함께 표시합니다:

mipSmoothing3DScale [0.05, 0.5], mipFilter2DVariance [0.1, 0.6], densifyGradThreshold [5e-07, 5e-06], ssimWeight [0.05, 0.5], mcmcNoiseScale [1e-05, 0.0001], mcmcRelocationInterval [50, 200]. 중앙: Convergence 차트 (X = 시행 인덱스 1-40, Y = 목표 값 0-1) — 회색 점 = 초기 샘플 (LHS), 파란색 점 = BayesOpt Acquisition, 주황색 점 = Restart 시행 (#22와 #31). Best Value 선은 시행 ~7까지 가파르게 상승, 그 후 시행 15까지 미미한 개선만, 그 이후 0.99+에서 평평한 plateau. 오른쪽 사이드바: 시행 로그 #1-#34, 점수 + 태그 (init/bo/restart). 오른쪽 위 Save Best Config 버튼은 bayesopt-best.json 을 작성합니다.

그것이 무엇인지: 하이퍼파라미터 검색을 위한 베이지안 최적화 콘솔. 베이지안 옵트는 가능한 적은 실험으로 알 수 없는 함수의 최적 점을 찾으려고 시도하는 자동 절차입니다 — 일반적으로: “mcmcMaxGaussians, capMultiplier, ssimWeight, gradThreshold의 어떤 조합이 내 장면 클래스에 대해 최고의 PSNR을 제공합니까?” $6^4 = 1296$ 개의 시행으로 구성된 격자 대신 베이지안 옵트는 약 40–100개의 정보를 가진 시행을 시도하고 그것으로 최적에 가까이 다가갑니다.

중요: 현재 앱에 출시된 버전은 실제 학습 실행에 대해 최적화를 실행하지 않습니다 (그것은 며칠이 걸릴 것입니다). 대신 합성 데모 목표 — Hill Climbing 특성과 약간의 노이즈가 있는 다중 모달 풍경 — 에 대해 실행됩니다. 이는 의도적인 것입니다. 창은 옵티마이저의 동작 (수렴 진행, 샘플 포인트, Best-So-Far) 을 보여 주고 탐색 공간 정의를 이해할 수 있도록 합니다. 실제 학습 기반 BayesOpt 실행 (Scene Class 프리셋에 대한 Phase Q7에서처럼) 의 경우 별도의 오프라인 CLI 워크플로가 사용됩니다. 창은 실시간 UI 변형입니다.

언제 열까 세 가지 사용 사례: 1. BayesOpt가 어떻게 작동하는지 이해하고 싶다 — 데모 실행을 시작하고 Convergence 차트를 관찰. 2. 내장된 10개 프리셋이 완벽하게 맞지 않는 새 장면 클래스 (예: “수족관” 또는 “골동품 가구”) 를 계획. 정신적으로 탐색 공간을 정의하고 여기서 “Bowl demo” 또는 “Densify” 프리셋으로 테스트한 다음 Best Config를 JSON으로 내보내고 실제 학습 실행의 시작점으로 사용. 3. RKBayesOpt 패키지에 정의된 기본 탐색 공간 (Mip Subset, RadianceKit Defaults) 을 검사하고 싶다 — 왼쪽 사이드바의 매개변수 패널에 나열됩니다.

어떻게 해석할까 - **Convergence 차트** (중앙 열): Y = 지금까지 달성된 최고의 목표 함수 값. X = 시행 인덱스. 처음에는 가파르게 상승 (BayesOpt가 초기 샘플을 무작위로 시도하고, 그 중 일부가 운이 좋음), 그 다음 최적에 가까운 영역이 소진되었으므로 점차 평평해집니다. 선이 20+ 시행 동안 평평하게 유지되면 실행을 중지할 수 있습니다 — 추가 시행은 더 이상 가져오지 않습니다. 차트의 개별 점은 개별 시행 값 (즉 “best so far”가 아님) 이며 단계별로 색상이 지정됩니다: 회색 = 초기 샘플, 파랑 = BayesOpt Acquisition, 주황 = Restart. - **시행 테이블** (오른쪽 열): #1, #2, #3, ... 각각 값과 단계 태그. 지금까지의 최고 시행은 노란색 별로 표시됩니다. 테이블에서 Best 시행을 식별하고 나중에 내보내기 시 매개변수 값을 볼 수 있습니다. - **탐색 공간 인스펙터** (왼쪽 사이드바): 선택된 프리셋에 대해 모든 매개변수 이름과 탐색 범위 $[l_0, h_i]$ 를 표시합니다. “RadianceKit defaults (6-dim)” 프리셋에 있으면 예: “densifyGradThreshold [5e-7, 5e-6]”를 봅니다 — 즉 이 두 값 사이의 로그 균등.

예제 워크플로 프리셋 “RadianceKit defaults (6-dim)”, 시행 예산 40, Seed 42를 선택합니다. “Start”를 클릭합니다. 관찰: 처음 8개 시행은 회색 (초기 샘플, LHS-Latin Hypercube), 그 다음은 파란색 (BayesOpt Acquisition). Convergence 차트는 시행 ~15까지 가파르게 진행되며, 그 후 평평해 집니다. 시행 ~30–40에서 Best 값이 안정화됩니다. “Save Best Config”를 클릭하면 프리셋 이름, 시행 인덱스, 값, 디코딩된 매개변수 값과 함께 bayesopt-best.json 이 저장됩니다. 이 JSON을 그런 다음 프리셋 정의에 수동으로 적용할 수 있습니다.

W30 [?] [?] "Start"


 위치

툴바 왼쪽, Idle/Finished 상태에서..

 기술적 설명

시행 목록을 재설정하고 Running 상태로 전환하고 새 실행 ID를 생성하고 (여러 Start 클릭의 stale 감지를 위해) 새 Pause Gate를 생성합니다. 그런 다음 옵티마이저를 비동기 스트림으로 실행하는 백그라운드 작업을 시작합니다. 초기 샘플 크기는 $\min(8, \text{budget} / 4 + 1)$ 에서 발생합니다 — 즉 예산 ≥ 28 의 경우 일반적으로 8개의 Latin Hypercube 샘플, 작은 예산의 경우 더 적음. 시행 업데이트는 점진적으로 수신되어 목록에 추가됩니다. Stale 실행 보호: 그 사이에 두 번째 Start 클릭이 실행 ID를 다시 설정하면 이전 실행에서의 업데이트가 폐기됩니다.

두드러진 버튼 모양을 위한 Primary Action 스타일.

 간단히 말하면

[?] [?] [?] [?] [?] [?], [?] [?], Seed [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?].


W31 [?] [?] "Pause"

 위치

툴바 왼쪽, Running 상태에서..

 기술적 설명

Pause Gate를 활성화하고 Paused 상태로 전환합니다. 실제 효과: Runner는 다음 목표 함수를 평가하기 전에 50 ms 폴링 루프에서 기다립니다. 즉, 현재 실행 중인 시행이 끝까지 진행되지만 (합성이며 마이크로초만 걸림), 더 이상 시행이 시작되지 않습니다. Resume이 실행되는 즉시 중단된 곳에서 계속됩니다.

 간단히 말하면

[?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?].


W32 [?] [?] "Stop"

 위치

툴바 왼쪽, Running 및 Paused 상태에서..

 기술적 설명

Runner 작업을 중단하고 참조를 null로 설정하고 Pause Gate를 해제하고 (paused였다면) Finished 상태로 (시행이 존재할 때) 또는 Idle 상태로 (없을 때) 전환합니다. 이미 계산된 시행은 목록에서 보이는 채로 남아 있습니다 — Stop은 그것들을 삭제하지 않습니다. 파괴적 버튼 역할은 실행을 중단하기 때문에 버튼을 빨간색으로 표시합니다.

 간단히 말하면

[?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] Best Config [?] [?] [?] [?] [?] [?]
[?] [?] [?].

W33 [?] [?] "Resume"

위치

툴바 왼쪽, Paused 상태에서..

기술적 설명

Pause Gate를 해제하고 Running 상태로 다시 전환합니다. Runner 작업은 이미 실행 중입니다 (폴링 루프에서 기다림). 루프가 일시 정지가 해제되었다는 것을 알아차리는 즉시 계속 실행되고 다음 시행을 시작합니다.

간단히 말하면

[?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?].

W34 [?] [?] "Save Best Config"

위치

툴바 오른쪽, 항상 표시 (그러나 bestTrial이 없으면 비활성화)..

기술적 설명

기본 파일 이름 bayesopt-best.json 이 있는 저장 대화 상자를 엽니다. JSON으로 제한됩니다. 확인 후 Payload 사전이 구축됩니다: 프리셋 이름, 시행 인덱스, 값 (목표 점수), 매개변수 (디코딩된 매개변수 이름 → 값의 사전). 디코딩은 정규화된 탐색 공간 좌표 $[0,1]^d$ 를 원래 값 범위로 다시 투영합니다 (log-uniform/linear/integer 스케일에 따라). JSON 출력은 pretty printed이며 정렬된 키가 있습니다. 쓰기 오류 시 (현재 데모 버전에서) 조용히 무시됩니다 — 오류 UI 없음, 데모 경로이기 때문에.

시행이 실행되지 않은 동안에는 버튼이 회색입니다.

간단히 말하면

[?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
 [?] [?] JSON [?] [?] [?] [?] [?] [?].
 [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
 [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
 [?].

W35 ? ? ? "Search Space" ? ? ?

위치

왼쪽 사이드바, 위..

기술적 설명

네 프리셋 옵션이 있는 메뉴 선택기: - "RadianceKit defaults (6-dim)" — 모든 Q7 하이퍼파라미터가 있는 전체 표준 탐색 공간. - "Mip subset (2-dim)" — mipSmoothing3DScale [0.05, 0.5] log-uniform과 mipFilter2DVariance [0.1, 0.6] linear만. 장면 클래스에 대해 Mip Splatting을 튜닝하려는 경우 유용. - "densify-until + ssim-weight

- 1. grad-thresh" — 세 개의 Densify 관련 매개변수 (densifyGradThreshold

log-uniform, ssimWeight linear, densifyUntilIter integer). - "Bowl demo (1-dim)" — "BayesOpt가 이렇게 작동" 데모를 위한 교육용 단일 매개변수 탐색 공간.

실행이 활성화된 동안 탐색 공간은 변경할 수 없습니다 (옵티마이저를 혼란시킬 것임).

간단히 말하면

BayesOpt ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? "RadianceKit defaults" ? ?
 ? ? ? Mip ? ? ? ? ? ? ? ? ? ?
 "Mip subset". BayesOpt ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? "Bowl demo".

W36 ? ? ? ? "Trial Budget"

위치

왼쪽 사이드바, Search Space 선택기 아래..

기술적 설명

10에서 200까지 슬라이더, 단계 5. 기본 40. 즉: BayesOpt는 최대 N 시행을 할 수 있습니다. 그 중 처음 몇 개는 초기 샘플 (Latin Hypercube), 나머지는 실제 BayesOpt 시행입니다. 실용을 위한 경험 법칙: d 차원의 탐색 공간은 좋은 최적을 위해 약 10d에서 20d 시행이 필요합니다. 6차원 기본값의 경우 60-120, 2차원 Mip Subset의 경우 20-40, 1차원 Bowl Demo의 경우 10-20.

실행 중에는 슬라이더가 비활성화됩니다.

간단히 말하면

? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? = ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? . 40 ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ? .

W37 ??? **"Random Seed"**



왼쪽 사이드바, 예산 슬라이더 아래..



1에서 100까지 슬라이더, 단계 1. 기본 42. Seed는 초기 Latin Hypercube 샘플과 데모 목표의 노이즈 구성 요소 모두에 전달됩니다. 재현성: 같은 Seed + 같은 탐색 공간 + 같은 예산은 정확히 동일한 시행 시퀀스를 제공합니다. "모든 동료가 데모를 재구축할 때 같은 실행을 얻을 수 있는가?"에 유용합니다. 실행 중에는 비활성화됩니다.

간단히 말하면

?? ???? ?????. ??
Seed = ?? ?? — ???? ?
?.

W38 **Chart (Convergence)**



창의 중앙 열..



두 레이어가 있는 Swift Charts 다이어그램: 1. 시행당 "best-value-so-far" 선 — 강조 색상의 단조 증가 또는 동일한 곡선. 2. 개별 목표 값이 있는 시행당 하나의 점, 단계별로 색상 지정. 기호 크기 40. 세 개의 단계 레이블: "init" (회색), "bo" (파랑), "restart" (주황).

작은 범례는 위 왼쪽에 단계 색상을 표시합니다. 시행 목록이 비어 있으면 (첫 Start 전), 대신 Chart 아이콘과 힌트 "Press Start to begin a BayesOpt run."이 있는 빈 상태 표시가 표시됩니다.

간단히 말하면

?? ??. ??? "???
?? ? ? ? ? ? ? ? ? ? ? ? ? ? ?"??
?. ?? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
BayesOpt?? ? ? ? ? ? ? ? ? ?
?.

W39 **Table (Trial Log)**



창의 오른쪽 열..



Lazy 스택형 시행 줄이 있는 스크롤 영역. 줄당 수평 스택: 시행 번호 (3자리 monospaced, 왼쪽), 값 (monospaced, 오른쪽 정렬, 70 pt 폭), 단계 태그 (Capsule, 25% Opacity에서 단계 색상으로 채워짐), 이 시행이 현재 Best일 경우 선택적으로 노란색 별. 자동 스크롤 메커니즘은 새 시행이 추가되는 즉시 자동으로 끝으로 점프합니다 — 따라서 자체적으로 스크롤하지 않고 화면 하단에서 실시간 진행을 함께 읽을 수 있습니다.

간단히 말하면

?? ? ? ? ? ? ? ? ? ? ? ? ? ? ?, ?? ? ?,
Best?? ? ?. ?? ? ? ? ? ? ? ? ? ? ?
?, ? ? ? ? ? ? ? ? ? ? ? ? ? ?
?.

메인 윈도우: Loss 진행과 Gaussian Count (I39-I41, 상호 참조)

메인 윈도우의 인스펙터 표시 세 가지는 실행 중인 학습 동안 끊임없이 보이고 진행이 건강해 보이는지에 대한 중요한 경험 법칙이 있기 때문에 별도의 설명이 필요합니다. 표시는 인스펙터의 "Loss Chart" 섹션 (제 2 장 — Inspector 참고) 에 있으며, 위 보조 창의 Holdout 분석을 보완합니다.

Loss 곡선이 언제 건강한가? 건강한 Loss 곡선은 세 단계를 보여 줍니다: (1) **Warmup** — 처음 200–500 반복에서 Loss가 높음 (장면에 따라 L1+SSIM 결합으로 일반적으로 0.15–0.25) 에서 약 절반으로 가파르게 떨어집니다. 이 단계에서 Loss가 떨어지지 않으면 입력이 보통 잘못된 것입니다 (이미지 손상, SfM 포즈가 나쁨, 초기 Gaussian 수가 너무 적음). (2) **Densification** — ~500과 densifyUntilIteration (고전적으로 15K, MCMC는 20K 또는 25K) 사이에 Loss가 계속 떨어지며, 종종 Densify 동작이 새 Gaussian을 삽입하고 옵티마이저가 그것을 활용할 때 작은 하향 점프가 있습니다. Gaussian Count는 이 단계에서 증가합니다. (3) **Refinement** — 그 후 Loss는 평평해지는 tail로 실행됩니다. 일반적인 최종 값: P4 Quality가 있는 Tanks & Temples Truck은 L1 \approx 0.023, V546과 함께 Full Classic이 있는 Horse는 L1 \approx 0.0230, 야외 Mip-NeRF360 장면은 종종 더 나쁨 (0.04–0.07).

Plateau는 무엇을 의미하는가? Plateau (Loss 곡선이 수천 회 반복 동안 수평으로 진행) 는 두 가지 해석을 가집니다: (a) 모델이 수렴되어 추가 학습이 더 가져오지 않음 — 좋은 경우. (b) 모델이 막힘 (지역 최소값, 나쁜 그래디언트 정보, 버퍼 한도의 Cap) — 나쁜 경우. 둘 다 차트에서 동일하게 보입니다. 구별: Gaussian Count를 봅니다. 그것도 평평하고 MCMC Cap에 가깝다면 (예: `.fullMCMC` 에서 150K/150K), 한도에 있습니다 — 그러면 Cap을 늘리거나 Plateau를 수용합니다. Gaussian Count가 여전히 증가하지만 Loss가 떨어지지 않으면 막힌 것입니다.

언제 중단할지 또는 계속 학습할지? 경험 법칙: 10K 반복 동안 Min Loss의 개선 없음 → 중단, 추가 반복은 낭비입니다. 그 전에: 한계선 개선이 보이면 Cmd+T (Training 메뉴 → Continue Training → +5K iterations) 를 통해 연장을 추가할 수 있습니다. 주의: MCMC에서는 Plateau가 종종 진짜입니다 — Cap이 자연스러운 한계입니다.

Gaussian Count Plateau는 "완료" 신호가 아닙니다. MCMC가 Cap에 도달했거나 Classic Densification이 소진되었다는 것만 의미합니다. 실제 "완료" 질문은 Holdout 분석만이 묻습니다 — 독립적인 테스트 세트의 PSNR/SSIM/LPIPS, Holdout 창 (W23–W29) 또는 `--benchmark` 플래그를 통해 평가됨.

PSNR/Holdout이 진실, Loss는 단지 프록시. Loss는 상대적 메트릭입니다: 모델이 학습 뷰에 적응하는 동안 떨어집니다. 그러나 낮은 Loss가 자동으로 좋은 모델을 의미하지는 않습니다 — 모델이 학습 이미지를 암기했다면 (Overfitting) Loss는 작지만 보이지 않은 뷰 (Holdout) 에서의 PSNR은 나쁠 것입니다. 따라서: 최종 품질 평가를 위해 항상 Holdout 메트릭을 보고 최종 Loss만 보지 마십시오.

경험 법칙 상자

- User Guide와 Keyboard Shortcuts는 정적 도움말입니다 — 키워드 질문에는 빠르지만 깊이를 위해서는 여기 이 매뉴얼을 사용하십시오.
- 디스크 여유 공간이 10% 미만으로 떨어지면 Manage Storage를 엽니다. 로그와 Imports 스테이징이 일반적인 원인입니다.
- Pareto Dashboard는 최소 세 개 또는 네 개의 학습 리포트 후에만 의미가 있습니다. X축 = 비용 (Time / Gs), Y축 = 품질 (PSNR / SSIM). Pareto Front가 효율적인 조합을 보여 줍니다.
- PSNR 벤치마크를 다른 사람들과 공개하기 전에 Holdout Analysis를 사용하십시오 — 테스트 세트가 실제로 대표적인지 확인합니다.

- BayesOpt Console은 주로 탐색 공간 정의를 위한 학습 및 검사 도구입니다. 실제 학습 기반 하이퍼파라미터 튜닝의 경우 오프라인 CLI 워크플로를 사용하십시오.
- Loss Plateau와 Gaussian Count Plateau는 별도로 해석되어야 합니다. Cap 한도는 "완료" 신호가 아닙니다. 실제 품질은 Holdout PSNR만이 측정합니다.
- Min Loss 개선 없이 10K 반복 → 학습 중지.

2

제 6 장 — 학습 구성

```

[?] preview-preset.json ~
{
  "id": "00000000-0000-0000-0000-000000000002",
  "name": "Preview",
  "category": "classic",
  "version": 1,
  "createdAt": "2026-05-27T22:54:00Z",
  "description": "Fast preview training - 5K iterations, 50% render scale, classic densification.",
  "trainingConfig": {
    "maxIterations": 5000,
    "densifyUntilIteration": 3500,
    "ssimWeight": 0.20,
    "renderScale": 0.50,
    "strategy": "classic",
    "cameraAlignment": "applePhotogrammetry",
    "densifyGradThreshold": 2.0e-06,
    "opacityResetInterval": 3000,
    "minOpacity": 0.005,
    "postCompactification": true,
    "perceptualLoss": 0.0,
    "metalFXUpscaling": false,
    "mpsLanczosScaling": false,
    "skyMasking": false,
    "midTrainingFloaterCleanup": true,
    "scaleRegularization": false
  }
}

```

그림 26: Preview 프리셋을 JSON으로 내보내고 `TextEdit`에 표시 — `id/name/category/version/createdAt/description` 필드, 모든 관련 매개변수가 있는 `trainingConfig` (`maxIterations` 5000, `densifyUntilIteration` 3500, `ssimWeight` 0.20, `renderScale` 0.50, `strategy` classic, `cameraAlignment` applePhotogrammetry, `densifyGradThreshold` 2.0e-06, `opacityResetInterval` 3000, `minOpacity` 0.005, 여섯 개의 `Bool` 토큰)

이미지에 보이는 것 일반적인 프리셋 JSON 내보내기입니다. 최상위 필드: `id` (UUID), `name`, 카테고리 (`classic` | `mcmc` | `sceneClass` | `custom`), 버전 (스키마 버전), `createdAt` (타임스탬프), `description` (자유 텍스트). 중첩된 `trainingConfig` 객체는 재현성에 결정적인 매개변수를 포함합니다 — 가져오기 시 전체 블록이 `TrainingConfig` 구조로 역직렬화되고 앱 버전의 기본값이 JSON에 없는 필드를 채웁니다 (예: 앱 업데이트 후). 다른 Mac에 프리셋을 전달하려는 사람은 단순히 이 JSON 파일을 보내면 됩니다.

`TrainingConfig` 구조는 RadianceKit의 모든 학습 실행의 핵심입니다. 학습에 영향을 미치는 모든 매개변수를 모아 둡니다 — 최대 반복 수에서 여덟 학습률, MCMC, Mip-Splatting, Curriculum, 장면 인식 Cap 로직의 특수 필드까지. Training Configuration 섹션 (Expert View) 의 사이드바에서 편집하고, 프리셋으로 저장하거나 JSON 내보내기로 다른 Mac에 전달합니다. 학습 시 정확히 이 객체가 동결되어 GPU 백엔드에 전달됩니다.

이 장은 파워 유저와 스크립트 작성자를 위한 참조 자료입니다. 81개의 모든 공개 필드, 9개의 정적 프리셋, 그리고 하나의 공개 메서드를 나열합니다. 소스 파일은 `TrainingConfig.swift`입니다 — 의심이 있다면 거기에 저장된 Doc Comment와 이니셜라이저 기본값이 Source-of-Truth 입니다.

참고 · UI VS. 프리셋/CLI

81개 필드 중 12개만이 Inspector (Sandboxed App Store 빌드) 에 직접 슬라이더, 토글 또는 선택기가 있습니다: **T1, T2, T17, T20, T22, T38, T56–T58, T60, T61, T73**. 나머지 69개 필드는 선택된 프리셋 (제 7 장) 을 통해 설정되며 **CLI 플래그** (제 5 장 참고) 를 통해서만 직접 재정의할 수 있습니다. 이 분리는 의도적입니다. 기본값은 안정적이고 프로덕션 검증된 상태로 유지되며, 파워 유저는 그래도 탈출구를 가집니다. 어떤 필드가 특히 흥미롭다면 먼저 제 2 장 (Inspector) 과 제 5 장 (CLI) 을 확인하여 JSON 조작 없이 도달할 수 있는지 살펴보십시오.

목차:

1. Iteration (T1–T2)
2. Learning Rates (T3–T10)
3. Densification — Classic (T11–T16)
4. Loss (T17–T20)
5. SH-Degree-Progression (T21)
6. Performance (T22–T25)
7. 진단과 점 구름 준비 (T26–T30)
8. 정규화 (T31–T37)
9. Refinement (T38–T44)
10. Sky-Dome (T45–T48)
11. Adam + LR 스케줄 (T49–T55)
12. Post-Processing + Apple AI (T56–T60)
13. MCMC Densification (T61–T73)
14. Mip-Splatting (Q1.5) (T74–T76)
15. Adaptive Densification (Q5) (T77–T79)
16. Curriculum (Q6) (T80–T81)
17. 정적 프리셋 (TP1–TP9)
18. 메서드: `resolveMcmcMaxGaussians`
19. 어느 필드를 무엇을 위해? (치트 시트)
20. 위험한 필드

Iteration (T1-T2)

T1 maxIterations

세부 정보

기본값: 30 000 (이니셜라이저), 35 000 (.full), 200 000 (.fullMCMC) 범위: 1 000 – 500 000 (UI 슬라이더), 로직에 하드 상한 없음 정의 위치: TrainingConfig.swift

기술적 설명

백엔드가 통과하는 총 학습 반복 수. 한 반복은 단일 학습 카메라의 Forward Render, 모든 Loss 구성 요소 (L1 + SSIM + 선택적 정규화

1. Sky Mask) 에 대한 Backward Pass, 그리고 하나의 Adam 옵티마이저 단계를

의미합니다. 이 숫자는 다른 스케줄에 직접 영향을 줍니다: Position 학습률은 0에서 T1 자체 또는 T49 positionLRScheduleEndIteration까지의 Cosine Annealing 곡선을 따르고, Densification은 T2 densifyUntilIteration에서 멈추며, MCMC 노이즈 감소는 T69 mcmcNoiseDecayEnd에서 종료되고, SH-Degree 업그레이드는 T21 에 정의된 세 표시에서 발생합니다. 고전적 Densification의 경우 경험적으로 결정된 스위트 스폿은 20 000-35 000 반복 (Sessions 1-32, V546 테스트), MCMC는 60 000-200 000 (V534) 입니다. 프리셋에 저장된 값을 훨씬 초과하여 극단적으로 늘리면 추가 품질이 거의 나오지 않습니다 — Adam Momentum이 포화되고, LR Decay 종료 없이는 Loss가 정체됩니다. 반대로 ~5 000 미만으로 떨어지면 완전히 수렴되지 않은 형상으로 이어집니다 (Density Control이 클론/스플릿할 시간이 너무 적음).

간단히 말하면

? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? = ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? : Quick 1
 000, Preview 5 000, Balanced
 20 000, Quality 35 000, MCMC
 Quality 200 000. ? ? ? ?
 ? ? : MCMC ? ? ? ? ? ? ? ?
 (100 000-200 000), Classic ? ?
 ? 40 000 ? ? ? ? ? ? ? ? —
 ? ? ? ? ? ? ? ? ? ? ? ? ? ? .

T2 densifyUntilIteration

☰ 세부 정보

기본값: 15 000 (이니셜라이저), 5 000 (.full), 160 000 (.fullMCMC) 범위: 0 – maxIterations 정의 위치: TrainingConfig.swift

🔧 기술적 설명

Densification이 중지되는 반복. 이때까지 Gaussian은 T11--T16 (Classic) 또는 T67--T70 (MCMC) 에 매개변수화된 규칙을 통해 복제, 스플릿, 정리됩니다. 그 후 Gaussian 수는 일정하게 유지되며 위치, 회전, 스케일, Opacity, SH 계수만 최적화됩니다 (Refinement 단계). 3DGS 원본 논문에서는 값이 T1 의 50%이지만 RadianceKit의 .full 프리셋에서는 약 14% (35 000 중 5 000) — V310/V338 실험의 결과로, 5 000 반복 후 추가 Densification이 결과를 악화시키는 경향이 있음을 보여 주었습니다 (더 많은 플로터, 더 많은 메모리 사용량, 품질 이득 없음). 반면 MCMC는 Relocation을 T1 의 80%까지 실행합니다 (V504b). MCMC는 해로운 플로터를 생성하지 않기 때문입니다. T2 를 너무 작게 (< 1 000) 선택하면 너무 적은 Gaussian이 생성됩니다. Classic에서 너무 크면 (> T1 의 50%) Overgrowth와 RGB-Saturation 아웃라이어로 이어 집니다 (Outdoor Overtraining Findings 참고).

💬 간단히 말하면

?? ?? Gaussian ??
?? ???. ?? ?? ??
?? ?? ?? ?? ???. 35
000 ?? ?? ?? ?? ??
5 000 ?? ?? ?? ?? — ?
?? ?? ?? ?? ?? ??
???. MCMC ?? ?? ??
80% ?? (? 200 000 ??
? 160 000). Quality ?? ??
?? ?? ? ?? ?? ?? ??
?? ?? ?? ?.

Learning Rates (T3–T10)

T3 positionLearningRate

☰ 세부 정보

기본값: 0.00016 범위: 1e-7 – 1e-3 (권장) 정의 위치:
TrainingConfig.swift

🔧 기술적 설명

학습 시작 시 (Iteration 0) 각 Gaussian의 XYZ 위치에 대한 Adam 학습률. Cosine Annealing 곡선을 따르고 학습 진행 과정에서 T4 positionLearningRateFinal로 감소합니다. 기본값 0.00016은 3DGS 원본 논문 (Kerbl 외 2023) 에서 가져왔으며 이미지 해상도를 늘려도 RadianceKit에서 스케일링할 필요가 없습니다 — 위치는 픽셀 공간이 아닌 세계 좌표계에서 움직입니다. 상당히 높으면 (> 0.0005) Gaussian이 긴 거리에 걸쳐 점프하고 Loss가 불안정해집니다. 훨씬 낮은 값 (< 0.00005) 은 잘못 초기화된 점 구름이 자리를 찾지 못하게 합니다. V414는 Init 값의 두 배를 테스트했습니다 → 16.8% 더 나쁜 L1 Loss. V544a 튜닝은 논문 기본값을 최적으로 확인했습니다. 참고: `.fullMCMC`에서는 의도적으로 이 값을 기본값으로 둡니다 — MCMC는 Relocation 로직에 일정한 학습률이 필요하므로 여기서 튜닝해도 가져오는 것이 없습니다.

💬 간단히 말하면

```
Splat [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] Splat [?] "[?] [?] [?] [?]"
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
```

T4 positionLearningRateFinal

세부 정보

기본값: 0.0000016 (이니셜라이저 + 논문), 0.000016
 (.full, .fullMCMC — 10배 더 높음) **범위:** 0 –
 positionLearningRate **정의 위치:** TrainingConfig.swift

기술적 설명

Position LR Cosine Annealing 곡선의 최

종 값. T_1 maxIterations 또는 설정된 경우
 T49 positionLRScheduleEndIteration에서 도달합니다.
 RadianceKit .full 프리셋은 0.000016을 사용합니다 —
 논문 기본값 0.0000016보다 10배 더 높습니다. V420 실험에서는 최종 값의 0.5배 (0.000008) 가 Loss를 6.4% 악화시킨다는 것을 보여 주었고, V414는 Init 값의 2배가 Loss를 16.8% 악화시킨다는 것을 보여 주었습니다. 높은 최종 값은 트레이드오프가 아니라 의식적인 선택입니다: Decay가 너무 강하면 Gaussian이 Refinement 단계 동안 새로 추가된 Densification 후보에 적응하는 능력을 잃습니다. V431/V433 확장을 통해 스케줄 단계를 단축할 수 있어 ($T_{49} < T_1$), T4가 학습 종료 전에 이미 도달하고 학습의 나머지가 일정한 미니 LR로 실행됩니다 — 일반적인 구성: $T_1 =$
 성: $T_{49} = 20\ 000, 35\ 000$, 따라서 Refinement는 15 000 반복 동안 0.000016에서.

간단히 말하면

Position
 .full, .fullMCMC — 10배 더 높음
 positionLearningRate
 Splat
 Splat = Splat
 Splat = Splat

T5 shDCLearningRate

세부 정보

기본값: 0.0025 (이니셜라이저 + 논문), 0.005 (.full 과 모든 MCMC 프리셋 — 2배) **범위:** 0.0001 – 0.05 **정의 위치:** TrainingConfig.swift

기술적 설명

Spherical Harmonic 색상의 DC 부분 (degree 0, 즉 일정한 Albedo) 에 대한 Adam 학습률. SH-DC는 Gaussian의 방향 독립적 기본 톤에 해당하며, 어떻게 보면 "기본 색상"입니다. V176 및 V188 실험에서 논문 기본값보다 2배 더 높은 것이 최적임을 발견했습니다 — 더 빠른 색상 수렴, 특히 짧은 학습 (예: 5 000 반복) 에서는 그렇지 않으면 SH-DC가 정상으로 들어오지 않기 때문입니다. 기하학적 LR과 달리 SH-DC는 Decay가 없습니다. 학습률은 모든 반복에 걸쳐 일정하게 유지됩니다 (또는 T51의 선택적 Extended Phase Decay만 따름). V416은 4배 0.01로 증가를 테스트했습니다 → beta2=0.99 Adam에서 6.4% 더 나쁜 Loss.

간단히 말하면

Splat
 .full, .fullMCMC — 2배
 shDCLearningRate
 Splat = Splat
 Splat = Splat

T6 shRestLearningRate

세부 정보

기본값: 0.000125 (이니셜라이저 + 논문), 0.00025 (.full 과 MCMC — 2배) 범위: 0.000001 – 0.005 정의 위치: TrainingConfig.swift

기술적 설명

더 높은 차수의 SH 계수 (Degree 1, 2, 3 — 즉 하이라이트, 반사, 부드러운 음영을 담당하는 View Direction 의존 색상 구성 요소) 에 대한 Adam 학습률. 논문 규약에 따라 T5 보다 20배 작습니다. 이 계수가 수에서 이차적으로 증가하기 때문입니다 (Degree 1의 경우 3, Degree 2의 경우 5, Degree 3의 경우 7 → Gaussian당 총 15 Float) 더 작은 학습률 없이는 이미지가 과포화될 것입니다. 두 단계로 잠금 해제됩니다 — T21 shDegreeUpgradeIterations의 첫 번째 마크까지는 Degree 0만 활성화 (즉 T5 만), 그 후 1, 그 다음 2, 마지막으로 3. 디퓨즈 조명이 많은 장면에서 여기에 낮은 값이 특히 중요합니다. 매우 광택이 있는 표면 (자동차 페인트, 물) 에서는 조정해도 가치가 없습니다 — SH 표현 자체가 제한적입니다.

간단히 말하면

?? ???? ???? (??, ?) ? ???? ?????.
????? ?????. ??
????? ???? ?????.
?. ?? ???? ???? ?????.
?? — ???? ???? ????
???? ???? ???? LR?? MCMC
?? ???? ???? ????
?? ? ?????.

T7 opacityLearningRate

세부 정보

기본값: 0.05 (이니셜라이저 + 논문), 0.1 (.full , MCMC — 2배) 범위: 0.001 – 1.0 정의 위치: TrainingConfig.swift

기술적 설명

각 Gaussian의 logit Opacity에 대한 Adam 학습률. 오픈 Opacity를 제한되지 않은 Float 값으로 저장하고 Sigmoid 로 [0, 1] 로 변환합니다. LR은 Logit Space에서 작용합니다. V50 테스트 (Best Single-Run L1 0.1664) 후 논문 기본값 0.05가 복원되었습니다. V71은 V67의 0.025를 revert했습니다. V188의 0.1로의 두 배는 Pruning을 더 효율적으로 만듭니다 — 죽은 Gaussian이 T14 pruneOpacityThreshold 아래로 더 빨리 떨어집니다. V418은 다음을 보여 주었습니다: beta2=0.99 Adam의 0.05는 0.1보다 7.1% 나뉩니다 — Adam 구성과의 상호작용은 사소하지 않습니다. 낮은 값 (< 0.01) 은 “죽은” Gaussian이 영원히 주위에 누워 있고 메모리를 소비하게 합니다. 너무 높은 값 (> 0.5) 은 Opacity 폭 발로 이어질 수 있으므로 Logit 값이 옵티마이저에서 [-15, 3] 으로 클램프됩니다 (CLAUDE.md의 “Opacity Explosion Prevention” 참고).

간단히 말하면

Splat? ???? ???? ????
????? ?????. ?? ????
?? — ???? ???? Splat?
???? ???? ???? ????
????? ?????. ?? ????
??, ???? ???? ????.

T8 opacityLearningRateFinal

세부 정보

기본값: 0.0 (= "Decay 없음") 범위: 0 또는 0.001 – opacityLearningRate 정의 위치: TrainingConfig.swift

기술적 설명

Opacity LR의 선택적 Cosine Decay 최종 값 (V427). 0.0 이면 Decay가 비활성화되고 Opacity LR이 전체 학습에 걸쳐 T7 에서 일정하게 유지됩니다. V427은 Decay 0.1 → 0.01 을 테스트했습니다 — 결과 11.5% 나쁜 Loss. revert되었으므로 기본값은 "꺼짐". 필드 뒤의 가설: Refinement 단계에서 일정한 Opacity LR이 진동으로 이어질 수 있어, 이미 올바른 투명도 수준에 도달한 Splat이 무작위 그래디언트 변동에 의해 다시 이동될 수 있다는 것. 경험적으로 확인되지 않습니다 — Logit 클램핑 로직이 어쨌든 이를 잡습니다. 필드는 향후 실험을 위해 사용 가능한 채로 유지됩니다. 매우 긴 MCMC 실행 (> 500K 반복) 도 이로부터 혜택을 받을 수 있습니다.

간단히 말하면

Opacity $\frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2}$
 $\frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2}$
 $\frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2}$
 $\frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2}$

T9 scaleLearningRate

세부 정보

기본값: 0.005 (이니셜라이저 + 논문), 0.01 (.full , MCMC — 2배) 범위: 0.0001 – 0.1 정의 위치: TrainingConfig.swift

기술적 설명

Log Space의 각 Gaussian의 세 스케일 구성 요소에 대한 Adam 학습률 (RadianceKit은 스케일이 양수로 유지 되도록 log(scale) 을 저장합니다). 논문 기본값 0.005, RadianceKit에서는 최적화된 학습률 구성에서 더 나은 스케일 수렴을 위해 0.01로 두 배. V423 실험: beta2=0.99 Adam 의 0.005 → 18.7% 더 나쁜 Loss와 눈에 띄게 너무 적은 Gaussian (Density Control이 클론할 수 없었음, 스케일 업데이트가 너무 느렸기 때문에). 스케일은 각 Gaussian의 확장을 제어합니다 — 너무 빠른 학습은 "Needle" Gaussian (극도로 길고 얇은 Splat, T34 scaleRatioPruneThreshold 참고) 으로 이어지고, 너무 느린 학습은 Splat이 너무 콤팩트하게 유지되어 Density Control이 너무 자주 스플릿해야 합니다.

간단히 말하면

Splat $\frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2}$
 $\frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2}$
 $\frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2}$
 $\frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2}$

T10 rotationLearningRate

세부 정보

기본값: 0.001 (이니셜라이저 + 논문), 0.002 (`.full` ,
MCMC — 2배) 범위: 0.0001 – 0.05 정의 위치:
TrainingConfig.swift

기술적 설명

각 Gaussian의 네 쿼터니언 구성 요소에 대한 Adam 학습률. 쿼터니언은 각 옵티마이저 단계에서 Adam 업데이트 후 다시 정규화됩니다 (L2 Norm = 1) — 그렇지 않으면 공분산 행렬이 퇴화될 것입니다. RadianceKit은 Quality 프리셋에서 논문 기본값을 두 배로 늘립니다. 스케일/위치에 비해 회전이 더 작은 절대 그래디언트 크기를 가지기 때문 (단위 구체에서 각 단계가 짧게 유지됨) 이고 2배 없이는 회전이 35 000 반복 윈도우에서 상당히 under-converged 될 것이기 때문입니다. V188에 문서화되어 있습니다. NeRF Blender 장면 (Lego, Chair)에서는 회전이 특히 영향을 미칩니다 — 객체의 가장자리는 5 000–10 000 반복 후에야 올바르게 정렬됩니다.

간단히 말하면

Splat[? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? — ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? . ?
? ? ? ? ? ? ? ? . ? ? ? ? ? ? ? ?
?]: Splat[? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?

Densification — Classic (T11–T16)


T11 densifyGradThreshold

 세부 정보

기본값: 0.000002 (이니셜라이저, 0.5배 해상도용으로 보정), 0.0000011 (`.full` , 1.0배용으로 보정), 0.000004 (`.quickTest` , 0.25배용으로 보정), $2e-7$ (`.fullClassicPaper`) **범위:** $1e-8 - 1e-3$ (해상도 의존)
정의 위치: TrainingConfig.swift

 기술적 설명

Screen Space에 투영된 그래디언트 `dMean2D` 의 L2 Norm 임계값. 이를 초과하면 Gaussian이 클론 또는 스플릿용으로 표시됩니다. 절대값은 학습 해상도에 직접 의존합니다 — `dMean2D` 는 대략 $1/\text{해상도}^2$ 처럼 스케일됩니다 (더 많은 픽셀 = 더 작은 픽셀당 그래디언트). 따라서 각 T22 `trainingRenderScale` 단계는 보정된 임계값을 필요로 합니다: $0.25x \rightarrow 4e-6$, $0.5x \rightarrow 2e-6$, $1.0x \rightarrow 5e-8 \dots 1.1e-6$ (`.full`). 논문 기본값 0.0002는 NDC 정규화되어 있고 RadianceKit의 세계 공간 파이프라인에서 직접 비교할 수 없습니다. V440에서 활성화된 T52 `adaptiveDensifyThreshold` 플래그를 사용하면 값을 현재 그래디언트 분포의 p98에서 런타임에 계산할 수 있지만 — V440은 실제 장면에서 이를 테스트하고 63 K Gaussian을 생성했습니다 (재앙적인 Pruning 손실). 플래그는 꺼진 채로 유지됩니다. Q5 (T77–T79) 는 Rolling Median을 통한 대체 Adaptive 로직을 제공합니다. **이 필드는 위험합니다** — 절반으로 줄이면 2–4배 더 많은 Gaussian이 생성됩니다 (메모리 압력, OOM 위험). 두 배로 늘리면 장면이 under-densify 될 수 있습니다.

 간단히 말하면

Splat(???)?? ?
 ?????? ???? ? ?
 ?????? ?????. ?? ? =
 ??? = ? ?? Splat. ? ?
 ? = ? ? ? Splat. ? ? ? ?
 ?????? ???? ? ? ? ? ?
 ??: ?? ???? Mac? ???
 Splat(??)?? ???? ? ?
 ?????? ?????. ?? ? ?
 ?? ???? 10% ???? ? ?
 ???)??.

T12 densifyFromIteration

세부 정보

기본값: 500 범위: 100 – 5 000 정의 위치: TrainingConfig.swift

기술적 설명

Densification이 활성화되는 첫 번째 반복. 그 전에는 초기 SfM 점 구름에서 “벗은” 학습만 발생하고 새 Gaussian이 생성되지 않습니다. 기본값 500은 3DGS 논문에서 왔으며 초기화에 안정화될 시간을 제공합니다 — Iteration 0부터 이미 Densification이 시작되면 잘못 배치된 SfM 점이 올바른 위치를 찾기 전에 여러 번 복제됩니다. V349는 1000을 테스트했습니다 → 약간 더 나쁜 Loss. 기본값이 최적입니다.

간단히 말하면

?? ???? Splat? ??
? ?????. ? ??
?? ???? ? ? ?
?. 500? ???? — ?
? ? ? ? ? ? ? ?
? orientation? ? ? ? ?
?? ? ? ? ? ?.

T13 densifyInterval

세부 정보

기본값: 100 (이니셜라이저, MCMC), 200 (.full) 범위: 50 – 1 000 정의 위치: TrainingConfig.swift

기술적 설명

두 Densification 단계 사이의 반복 수. 논문 기본값 100 — 매 100회 반복마다 Densify 후보 목록이 평가되어 클론/스플릿되고 동시에 Prune 후보 목록 (sigmoid(opacity) < T14 pruneOpacityThreshold) 이 제거됩니다. V112 테스트는 .full 의 경우 200이 최적임을 발견했습니다 — 더 적은 재구성 패스가 실행되므로 GPU 부하가 줄어들고, 각 Gaussian에게 Clone 작업 후 안정화할 더 많은 시간을 제공합니다. V417은 beta2=0.99에서 100을 테스트했습니다 → 5.8% 더 나쁨 (957 K Gaussian, 과 Densification). MCMC에서는 동일한 필드가 Relocation Interval로 해석됩니다. MCMC 특정 로직은 T67 mcmcRelocationInterval을 참고하십시오.

간단히 말하면

?? ???? ? Splat? ?
??. 100 = ??, 200 = ??. ?
???: ? Splat? ? ? ?
? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ?
? ? ? ? GPU? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ?.

T14 pruneOpacityThreshold

 세부 정보

기본값: 0.005 (이니셜라이저, 논문, MCMC), 0.001 (.full)
범위: 0.0001 – 0.1 **정의 위치:** TrainingConfig.swift

 기술적 설명

다음 Densification 단계에서 Gaussian이 삭제되는 Sigmoid Opacity 임계값. T7 opacityLearningRate 및 옵티마이저의 Logit Clamp 로직과 함께 작용합니다. V393은 .full 에서 기본값을 0.005에서 0.001로 낮추었습니다 — 결과: 이국적인 시점에서만 역할을 하는 Splat이 더 오래 보존되고 SH 세부 정보에 기여합니다. V394는 0.0001을 테스트했습니다 → 약간 더 나뭇 (너무 적게 정리됨, 메모리 낭비). 중요: Density Control은 항상 정리해야 합니다. 다른 조치로 버퍼 용량이 이미 가득 차 있더라도 (“Density Control Must Always Prune” 참고 CLAUDE.md) — 그렇지 않으면 죽은 Gaussian이 누적되고 카운트가 멈춥니다.

 간단히 말하면

```
Splat [0.005] "0.005" [0.005] [0.005].
Quality [0.001] [0.001]
— Splat [0.001] [0.001] [0.001] [0.001] [0.001] [0.001] [0.001] [0.001] [0.001] [0.001] (0.01) Splat [0.001] [0.001] [0.001] [0.001] [0.001] [0.001] [0.001] [0.001] [0.001] [0.001].
```

T15 opacityResetInterval

 세부 정보

기본값: 3 000 (이니셜라이저 + 논문), 100 000 (.full = 효과적으로 비활성화), 200 000 (.fullMCMC = 비활성화) **범위:** 1 000 – 100 000+ **정의 위치:** TrainingConfig.swift

 기술적 설명

모든 Gaussian의 Opacity가 낮은 값 (~0.01) 으로 재설정 되는 간격 (반복 수) — “동결된” Splat을 재평가하기 위한 3DGS 논문의 조치. V194는 RadianceKit의 Warmup + 확률적 학습 설정 + 2배 학습률에서 Opacity Reset이 5.5% 품질이 들고 Logit Clamp가 이미 Reset 기능을 다룬다는 것을 보여 주었습니다. 따라서 .full 에서는 실질적으로 비활성화 됩니다 (100 000 > 35 000 = 절대 트리거되지 않음). V421 은 beta2=0.99로 3 000마다 Reset을 테스트했습니다 → 4.9% 더 나뭇. revert됨. .fullClassicPaper (Q1.5-A, 논문 충실 테스트) 에서는 의도적으로 다시 3 000으로 설정되어 있습니다 — 논문 규모 Gaussian 예산을 달성하기 위한 레버 중 하나였습니다.

 간단히 말하면

```
[0.001] Splat [0.001] "0.001" [0.001] [0.001] [0.001] (0.01) — Opacity Reset [0.001] [0.001] [0.001] [0.001] [0.001] [0.001] [0.001] [0.001] [0.001] [0.001] (0.01) [0.001] [0.001] [0.001] [0.001] [0.001] [0.001] [0.001] [0.001] [0.001] [0.001].
```

T16 maxScreenSize

세부 정보

기본값: 0.0 (= 비활성화) **범위:** 0 (off) 또는 > 0 정의 위치: TrainingConfig.swift

기술적 설명

Gaussian이 강제로 스플릿되기 전에 도달할 수 있는 Screen Space 최대 크기 (투영된 픽셀 단위). 값은 0으로 설정됩니다 (V48 테스트 및 revert) — RadianceKit의 Density Control은 대신 dMean2D 로직의 세계 공간 스케일 임계값을 사용합니다. 향후 Mip-Splatting (T74-T76) 이나 장면별 Splatting 전략 실험이 이로부터 혜택을 받을 수 있기 때문에 필드 카탈로그에 남아 있습니다. 활성화 (값 > 0, 예: 20) 는 매우 커진 Splat이 화면에서 강제로 분할되도록 합니다 — 단일 거대 Splat이 너무 적은 세부 정보를 제공하는 크고 매끄러운 벽 표면에 관련이 있습니다.

간단히 말하면

?? Splat? ???? ????
?? ? ???? ? ? ? ? ?
????? ? ? ? ? ? ? ? ? ?
???? ???? ???? Splat (? :
??) ? ???? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ?

Loss (T17-T20)

T17 ssimWeight

세부 정보

기본값: 0.2 (이니셜라이저 + 논문 + .full), 0.05 (모든 MCMC 프리셋) **범위:** 0.0 - 1.0 정의 위치: TrainingConfig.swift

기술적 설명

결합된 Loss 함수에서 D-SSIM 부분의 가중치 $loss = (1 - \lambda) * L1 + \lambda * D-SSIM$, 여기서 $\lambda = T17$. 3DGS 논문 기본값 0.2는 Classic Densification에 최적입니다 — V383은 0.3을 테스트했습니다 → 28.9% 더 나쁨. V373b는 0.2를 스위트 스폿으로 확인했습니다. MCMC의 경우 V521b/V534에서 독립적으로 0.05가 최적임이 밝혀졌습니다. MCMC가 확률적 탐색을 통해 더 강한 L1 신호 부분을 필요로 하기 때문입니다 — 더 높은 SSIM 가중치는 Relocation 결정을 희석할 것입니다. SSIM은 L1보다 계산 비용이 훨씬 비쌉니다 (전체 이미지에 걸친 로컬 11x11 윈도우). RadianceKit은 1080p 이미지당 1 ms 미만으로 유지되는 MPS 가속 구현을 사용합니다. Q7-BayesOpt Sweep은 0.05 (.outdoorPreset : 0.082) 와 0.171 (.indoorPreset) 사이의 장면별 최적값을 발견했습니다.

간단히 말하면

"????? ?????" ? ? ?
? "???? ?????" ? ? ? ?
????? ?????. 0.2? ? ?
?? ? ? ? ? ? ? ? ? ? ? ?
? ? ? = ? ? ? ? ? ? ? ? ?
???? ???? ? ? ? ? ? ? ?
?. ? ? ? = ? ? ? ? ? ? ? ?
????? ? ? ? ? ? ? ? ? ?
??????. ? ? ? ? ? ? ? ? ?
?????

T18 ssimWeightRefinement

☰ 세부 정보

기본값: 0.0 (= "전환 없음, ssimWeight 유지") **범위:** 0 또는 0 – 1.0 **정의 위치:** TrainingConfig.swift

⚙ 기술적 설명

T2 densifyUntilIteration 후 Refinement 단계의 선택적 SSIM 값. V428은 Refinement에서 0.2 → 0.3을 테스트했습니다 → 16% 더 나쁜 Loss (L1과 SSIM 모두 악화). revert됨, 따라서 기본값 0.0. 필드 뒤의 가설은 Densification 후 — 더 이상 새 Gaussian이 생성되지 않을 때 — 더 강한 SSIM 부분이 구조적 선명도를 최대화할 것이라는 것이었습니다. 경험적으로 잘못됨: SSIM 가중치를 늘리는 것은 간접적으로 L1 가중치를 낮추는 것을 의미하고, L1은 최종 Refinement 단계에서 훨씬 더 의미 있는 신호입니다. 필드는 Perceptual Loss (T60) 또는 Edge Loss (T19) 의 향후 실험을 위해 사용 가능한 채로 유지되며, 거기서는 Refinement 특정 Loss 구성이 의미가 있을 수 있습니다.

💬 간단히 말하면

???(?) (Splat ???
???) ?????. 0.0??:
???? SSIM ????. ?
????????????????
????????????.

T19 edgeLossWeight

☰ 세부 정보

기본값: 0.0 (= 비활성화) **범위:** 0 또는 0.001 – 1.0 **정의 위치:** TrainingConfig.swift

⚙ 기술적 설명

V437 실험적 Loss: L1+SSIM에 추가로 이미지 가장자리를 직접 비교하는 Sobel Gradient Domain L1 Loss의 가중치 (Ground Truth Sobel vs Render Sobel). 가설: 가장자리 정보가 이미지 품질의 인식적 초석이며 명시적 향이 Gaussian 이 가장자리를 더 잘 맞추도록 장려해야 합니다. 테스트 결과: 가중치 0.1 → 11% 더 나쁜 Loss, 0.01 → 품질 중립이지만 10% 더 느림. Sobel 패스는 Ground Truth와 Render에 추가 MPS Forward 비용이 듭니다. 따라서 영구 비활성화. 향후 사용 사례: 단단한 인공 가장자리 (건축, 가구, 렌더링) 가 있는 장면이 혜택을 받을 수 있습니다 — Q7-Scene-Class 프리셋은 이를 선택하지 않고 대신 SSIM 가중치를 스케일 조정했습니다.

💬 간단히 말하면

????? ???? ???? ?
????? ??. ??????
????? ? ?????. ??
?????????.


T20 skyMaskingEnabled

 세부 정보

기본값: false (이니셜라이저와 모든 프리셋) **범위:** boolean
정의 위치: TrainingConfig.swift

 기술적 설명

Sky Masking을 켭니다. 각 이미지에서 Apple Vision 프레임워크 (VNGenerateForegroundInstanceMaskRequest)를 통해 하늘 영역이 마스크되고 이 영역의 Loss가 0으로 설정됩니다. 의의: 야외 장면은 종종 파란색/회색/흰색 하늘 픽셀이 앱이 정확히 거기에 Gaussian을 배치하도록 하는 것으로 고통받습니다 — 이는 “플로터”로 인식됩니다. Sky Mask 없이는 이 영역의 Loss가 절대 0이 되지 않을 것입니다. 이미지의 하늘이 약간 변하고 앱이 Splat으로 영원히 다시 만들려고 시도하기 때문입니다. Vision 마스크는 학습 전에 카메라당 한 번 계산되어 RAM에 보관됩니다. 일반적으로 T45 skyDomeEnabled와 함께 활성화됩니다 (Settings View의 UI 로직). 실내 장면이나 합성 렌더링에서는 비활성화 상태로 두십시오 — 마스크가 거기서 잘못 천장이나 벽을 “Sky”로 인식할 것입니다.

 간단히 말하면

?? ? ? ? ? ? ? ? ? ? ?
?: ? ? ? ? ? ? ? ? ? ?
Splat ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ?
? 3D ? ? ? ? ? ? ? ? ? ?
?.

SH Degree Progression (T21)

T21 shDegreeUpgradeliterations

세부 정보

기본값: [1_000, 2_000, 3_000] (이니셜라이저), [2_000, 5_000, 8_000] (.full , MCMC), [1_000, 2_000] (.preview — Degree 3 건너뛴) **범위:** [Int], 각 값이 [0, maxIterations] 안에, 단조 증가 **정의 위치:** TrainingConfig.swift

기술적 설명

활성 SH Degree가 0→1, 1→2, 2→3으로 업그레이드되는 반복. 첫 마크 전에는 DC 구성 요소만 활성 (즉 T5 shDCLearningRate), 첫 마크 후 DC + 3 Degree-1 계수, 두 번째 마크 후 + 5 Degree-2 계수, 세 번째 마크 후 모든 15 계수. Gaussian당 메모리 요구량이 단계로 증가합니다 — 4 Float → 16 Float → 36 Float → 64 Float. Quality 프리셋은 이니셜라이저 기본값에 비해 업그레이드를 지연합니다 (V228). 색상 디테일이 더 높은 주파수로 추가되기 전에 형상이 먼저 안정화되어야 하기 때문입니다. V384는 .full 에 대해 [1K, 2K, 3K] 를 테스트했습니다 → 9.3% 더 나뭇 — 지연을 확인합니다. .preview 는 Degree 2에서 자릅니다. Degree 3은 5 000 반복에서 수렴하지 않고 옵티마이저 용량만 소비하기 때문입니다. Q6 (T80–T81) 는 이 목록을 동적으로 재정의하는 대체 Curriculum 로직을 제공합니다.

간단히 말하면

???? ???? ???? ????
 ???? ???? ???? ????
 ???? ???? ???? ???? (??
 ????), ??). ?? — ?? ?
 ?? ??, ? ???? ??. ?? ?
 ???? ???? ???? ????
 ?????. ???? ???? ????
 ???? ???? ???? ??.

Performance (T22–T25)

T22 trainingRenderScale

☰ 세부 정보

기본값: 1.0 (이니셜라이저, `.full`, MCMC, Scene-Class), 0.5 (`.preview`), 0.25 (`.quickTest`) **범위:** 0.05 – 2.0 (일반적으로 0.25, 0.5, 1.0) **정의 위치:** TrainingConfig.swift

🔧 기술적 설명

학습 이미지의 원본 해상도에 상대적인 학습 시 렌더 해상도. 0.5에서 각 이미지가 50% 폭 × 50% 높이 (즉 픽셀의 25%)로 다운샘플링되고 Gaussian 렌더링이 이 더 작은 해상도에서 발생합니다. 메모리와 계산 오버헤드를 모두 이차적으로 줄입니다. **중요:** T11 `densifyGradThreshold`가 선택된 해상도와 일치해야 합니다 — 그래디언트 크기가 1/해상도²로 스케일되므로 `.quickTest` (0.25x)는 `.full` (1.0x, 1.1e-6)보다 훨씬 더 높은 임계값 (4e-6)을 가집니다. RadianceKit은 매우 큰 이미지에 경고하고 자동으로 조정합니다 — 3 MP 목표 해상도. 극단적인 4K 입력 이미지에서는 0.5 또는 심지어 0.25가 의미가 있습니다. 그렇지 않으면 모든 Mac이 CPU Compaction에서 실행됩니다.

💬 간단히 말하면

?? ? ? ? ? ? ? ? ? ? ? ? ?
 1.0 = ??, 0.5 = ? ? ? ? ? ? ? ?
 ?? = ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? (12 ? ? ? ? ? ? ? ?) ? ? ? ?
 ? ? ? ? ? ? ? ? ? ?

T23 resolutionWarmupScale

☰ 세부 정보

기본값: 0.0 (= 비활성화) **범위:** 0 또는 0.1 – trainingRenderScale **정의 위치:** TrainingConfig.swift

🔧 기술적 설명

V133 최적화: Refinement 단계보다 낮은 해상도에서 Densification 단계 (Iter 0에서 T2 까지)를 학습합니다. V308은 `.full`을 다시 비활성화했습니다. T22 = 1.0 과 Cosine Annealing에서 Time-Win이 미미하고 품질이 최소로 들었기 때문입니다. 4K 입력과 긴 학습 실행에서 다시 의미가 있을 수 있기 때문에 필드 카탈로그에 유지됩니다 — Q6 Curriculum (T80)이 비슷한 로직을 채택했고 거기서는 LR 스케줄과 결합되어 있습니다. 활성화되고 T80 `curriculumResolutionRamp`도 true이면 Q6가 우선하고 이 값을 덮어씁니다.

💬 간단히 말하면

?? ? ? ? : ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? , ? ? ? ? ? ?
 ? ? ? ? ? . ? ? ? ? ? ? ? ? ?
 ? Q6 ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?

T24 tileSize

세부 정보

기본값: 16 범위: 8, 16, 32 정의 위치: TrainingConfig.swift

기술적 설명

픽셀 단위의 래스터화 타일 크기. Gaussian Splatting 렌더링은 타일 기반입니다. 이미지가 16x16 픽셀 타일로 분해되고 각 타일이 관련 Gaussian을 모아 깊이별로 정렬한 다음 블렌딩합니다. 16은 실질적으로 모든 3DGS 구현이 사용하는 표준이며 RadianceKit Metal 커널에서 하드코딩되어 있습니다. 이 값을 변경하면 셰이더의 재컴파일이 필요하며 현재 상태에서는 효과적이지 않습니다. 향후 엔진 버전이 Tile Size를 동적으로 지원할 경우를 위해 필드로 남아 있습니다.

간단히 말하면

타일 크기는 16x16 픽셀입니다. 이는 Gaussian을 모으고 블렌딩하는 데 사용됩니다.

T25 throttleDelayMs

세부 정보

기본값: 0 (이니셜라이저, .full, MCMC, Scene-Class), 0 (.preview) 범위: 0 – 100 정의 위치: TrainingConfig.swift

기술적 설명

학습 반복 사이의 인위적인 지연 (밀리초). 0 = 풀 스피드 (표준). 더 높은 값은 GPU/CPU가 정기적으로 호흡할 휴식을 받게 하여 학습 중 Mac을 "사용 가능"하게 만듭니다 — 다른 앱의 사용성은 증가하지만 학습 시간은 Delay와 선형적으로 증가합니다. 일반적인 값: 1-2 ms ("가벼운" Throttling, +5% 학습 시간, Mac이 더 반응적으로 느껴짐), 5 ms ("중간", +15% 학습 시간), 10+ ms ("Eco", 잠재적으로 두 배 학습 시간). Inspector의 "Performance"에서 제공되지만 표준 보기에는 없습니다 — dev_ux-backlog.md 를 참고하십시오. 잘못 이해되면 학습 시간을 극적으로 연장하기 때문에 Expert View에서 제거하자고 제안합니다.

간단히 말하면

이 설정은 학습 속도 조절을 위한 지연 시간입니다. 0은 풀 스피드입니다. 더 높은 값은 GPU/CPU가 정기적으로 호흡할 휴식을 받게 하여 학습 중 Mac을 "사용 가능"하게 만듭니다. 일반적인 값: 1-2 ms ("가벼운" Throttling, +5% 학습 시간, Mac이 더 반응적으로 느껴짐), 5 ms ("중간", +15% 학습 시간), 10+ ms ("Eco", 잠재적으로 두 배 학습 시간).

진단과 점 구름 준비 (T26–T30)


T26 depthDistortionWeight

 세부 정보

기본값: 0.0 (= 비활성화) 범위: 0 또는 0.0001 – 0.05 정의 위치: TrainingConfig.swift

 기술적 설명

V366 실험적: Depth Distortion 정규화 Loss의 가중치. 렌더 광선을 따라 깊게 배치되어 있지만 개념적으로 같은 표면에 속하는 Gaussian을 처벌합니다 — 집중된 깊이 분포를 장려하고 플로터를 줄입니다. 테스트: 0.01 → 4.5% 더 나뭇, 0.001 → 8.1% 더 나뭇. 이론적 이점 — Multi-View 일관성을 개선 — 은 L1 Loss에 반영되지 않습니다. 가설이 SfM 형상이 올바르게 암묵적으로 가정하고 Gaussian이 단지 “쌍일” 필요가 있기 때문입니다. 실제로는 SfM 점 구름이 일반적으로 가장 약한 구성 요소이지 적재가 아닙니다. 특히 깨끗한 포즈가 있는 Multi-View 데이터셋 (합성, Ground Truth가 있는 Mip-NeRF 360) 에 사용 가능한 채로 유지됩니다.

 간단히 말하면

?? ? ? ? ? Splat ? ?
 ? ? ? ? ? ? ? ? ? ? ? ?
 ?. ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?

T27 singleViewOverfit

 세부 정보

기본값: false 범위: boolean 정의 위치: TrainingConfig.swift

 기술적 설명

진단 플래그: true인 경우 각 학습 반복에서 카메라 풀에서 무작위로 선택되는 대신 카메라 인덱스 0이 강제로 사용됩니다. 의의: 모델이 단일 뷰조차 Overfit할 수 없다면 (즉 View 0의 Loss가 10 000 반복 후에도 0에 가까이 가지 않으면) Forward/Backward Pass에 근본적인 버그가 있는 것입니다. 이 스위치는 Metal 셰이더와 Differentiable Rasterizer 커널의 개발 중에 집중적으로 사용되었습니다 — V42–V47 단계. 오늘날은 누군가 백엔드 코드를 수정했고 회귀 테스트를 하고 싶을 때 Sanity Check로만 사용 가능합니다. CLI에서는 `--single-view`.

 간단히 말하면

?? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

T28 maxCameras

세부 정보

기본값: 0 (= “모든 카메라 사용”) **범위:** 0 또는 1 – N **정의 위치:** TrainingConfig.swift

기술적 설명

V43의 진단 한도: 처음 N개의 카메라로만 학습하고 나머지는 무시합니다. 원래 의의: 너무 많은 카메라가 그래디언트 충돌을 생성한다는 가설을 테스트 (같은 Gaussian에 대한 너무 많은 모순적 Loss 신호). 테스트 결과: 인위적 제한에서 체계적 이점 없음 — 더 많은 프레임은 실질적으로 항상 더 많은 품질을 가져옵니다. 대상 실험을 위한 CLI 플래그 (--max-cameras N) 로 유지됩니다 (예: “1 500 이미지 드론 비행의 처음 100 이미지에서 학습이 작동합니까?”). UI에는 노출되지 않습니다.

간단히 말하면

????? ???? ???? —
?? N?? ???? ????
????? ?????. ?
????? ???? ???? ,
?? 0 = ???? ?
?. ? ? ? = ?
?? ? (feedback_more-
frames-better.md ??).

T29 maxInitialPoints

세부 정보

기본값: 0 (= “모든 SfM 점 사용”) **범위:** 0 또는 1 000 – 200 000+ **정의 위치:** TrainingConfig.swift

기술적 설명

V54 보호: 학습이 시작되는 초기 SfM 점의 수를 제한합니다. 조밀한 COLMAP 재구성은 > 60 000 점을 생산할 수 있으며 큰 초기 스케일에서 픽셀 오버랩당 200–300 Gaussian으로 이어집니다 — 학습이 수렴하지 않는 “안개 필드”를 만듭니다. ~16 000 점으로 서브샘플링 (학습 엔진의 Hard Cap 로직) 은 초기 밀도를 참조 3DGS가 사용하는 수준으로 가져오고 오버랩을 극적으로 줄입니다. 매우 조밀한 SfM에서 자동으로 설정됩니다. CLI에서는 --max-points N .

간단히 말하면

???? ???? ???? ??
?? ? ? ? ? ? ? ? ? ? ?
????? (60 000 ??) ? ? ?
?? ? ? ? ? ? 16 000 ? ? ?
???? — ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ?
???? ? ? ? ? ? ? ? ? ? ?

T30 cameraClusterOutlierMultiplier

세부 정보

기본값: 10.0 (모든 프리셋 — 절대 덮어쓰지 않음) **범위:** 1.0 – 100.0 **정의 위치:** TrainingConfig.swift

기술적 설명

Phase 3.10 A.1에서 도입된 카메라 클러스터 아웃라이어 필터의 승수. 학습 전 학습 엔진은 모든 카메라 위치의 중심과 카메라의 중심으로부터의 최대 거리를 계산합니다. 중심으로부터의 거리가 multiplier × maxCameraDistance 를 초과하는 SfM 점은 아웃라이어로 폐기됩니다. 기본값 10x는 Phase 3.10 이전의 동작을 보존합니다. 미묘한 버그: 더 타이트한 SfM (카메라가 더 가깝게) → 더 작은 maxCameraDistance → 더 작은 임계값 → 더 많은 점이 아웃라이어로 폐기됨. 더 느슨한 SfM → 더 큰 임계값 → 더 적은 점이 폐기. 이는 Phase 3.9 Funnel vs Training 반상관관계의 원인 중 하나입니다: 더 나은 SfM이 다운스트림에서 더 나쁜 학습으로 이어질 수 있습니다. 너무 많은 초기 점이 죽기 때문입니다. 필드는 A.3 Sweep을 위해 CLI 재정의 (--camera-cluster-outlier-multiplier) 로 있습니다. UI에는 노출되지 않습니다. 5 미만의 값은 일반적으로 너무 제한적이고, 20 이상은 효과가 없습니다.

간단히 말하면

?? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?
???. 10 = ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? (? ? ? ? ?) ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? — ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?

정규화 (T31–T37)

T31 coarseToFineBlurRadius

세부 정보

기본값: 0 (= 비활성화) **범위:** 0 또는 1 – 10 **정의 위치:** TrainingConfig.swift

기술적 설명

V369 실험적: Densification 단계 시작 시 Ground Truth 이미지에 적용되고 Densification 종료 (T2) 까지 0으로 선형 감소되는 Box Blur 반경. 가설: Coarse-to-Fine 학습 — 먼저 거친 구조를 학습한 다음 세부 정보 — 이 더 안정적인 형상을 제공해야 합니다. 테스트: r=3 → 9.6% 더 나뭇, r=1 → 5.1% 더 나뭇. 실패 이유: Densification은 이미지 도메인 그라디언트를 기반으로 결정하며, 불러는 “여기서 클론해야 함”에 중요한 신호를 정확히 줄입니다. 다른 Density Control 스킴으로 향후 테스트를 위해 필드 카탈로그에 남아 있습니다.

간단히 말하면

?? ? " ? ? ? ? ? ? , ? ? ? ? ?
? ? " ? ? . ? ? ? ? ? ? ? ? ? ?
? ? ? ? . ? ? ? ? ? ? ? ? ? ? ? ?

T32 scaleRegWeight

세부 정보

기본값: 0.0 (= 비활성화) **범위:** 0 또는 0.0001 – 0.05 **정의 위치:** TrainingConfig.swift

기술적 설명

V370 실험적: 세계 공간 스케일에 대한 L1 정규화. 너무 커지는 Gaussian을 처벌합니다 — 전체 벽 표면을 하나의 Gaussian으로 덮는 "Mega Splat"을 방지합니다. 테스트: 0.01 → 200% 더 나쁜 Loss (2 M Gaussian, 완전한 폭발), 0.001 → 214% 더 나쁨. 이유: 스케일 정규화는 Density Control과 충돌합니다 — 더 작은 스케일은 더 많은 Gaussian이 필요하다는 것을 의미합니다. 따라서 Density Control이 더 자주 스플릿하고 이는 더 많은 그래디언트 비용을 의미합니다. 비활성화되었지만 Mip-Splatting 실험 (T74)을 위해 문서화됨: 이 컨텍스트에서는 스케일 하한이 의미를 있을 수 있습니다.

간단히 말하면

Splat ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? Splat ?
 ? (? ? ? Splat) ? ? ? ? ? ? ?
 ? . ? ? ? ? ? ? ? ? ? .

T33 anisotropyRegWeight

세부 정보

기본값: 0.0 (= 비활성화) **범위:** 0 또는 0.0001 – 0.05 **정의 위치:** TrainingConfig.swift

기술적 설명

V445 실험적: max(scale)/min(scale) 비율에 대한 페널티. 플로터로 인식되는 극도로 늘어난 "Needle" Gaussian을 방지해야 합니다. 테스트: 0.01 → 69% 더 나쁨, 0.001 → 15% 더 나쁨. 이유: 정규화가 Splat을 "둥근" 모양 방향으로 강제하는데, 평평한 표면 (벽, 테이블, 바닥)에서는 정확히 잘못된 것입니다 — 거기서는 평평하고 넓은 Gaussian이 구형보다 더 효율적입니다. 비활성화됨. V549f는 T34 scaleRatioPruneThreshold로 대안적이고 더 표적화된 접근 방식을 제공했고 마찬가지로 revert되었습니다.

간단히 말하면

? ? ? ? ? ? Splat ? ? ? ?
 ? ? ? ? ? . ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? .
 ? ? ? .


T36 opacityDecayFactor

 세부 정보

기본값: 0.0 (이니셜라이저 = 비활성화), 0.9995 (`.full` , `.classicBalanced` — HTGS 표준) **범위:** 0 (off) 또는 0.95 – 1.0 **정의 위치:** TrainingConfig.swift

 기술적 설명

HTGS 스킴 (Hierarchical Time-Gating, Eurographics 2025) 의 V546 구현: 매 T37 `opacityDecayInterval` 반복마다 각 Gaussian의 sigmoid Opacity가 이 인수로 곱해집니다. 0.9995 × 100회 적용은 Densification 단계당 ~95% 잔존을 산출합니다 — 모든 Opacity에 대한 약하지만 꾸준한 하향 압력으로, 약하게 기여하는 Gaussian이 T14 `pruneOpacityThreshold`에 안정적으로 가라앉도록 합니다. 결과: V438 (Decay 없음) 에 비해 Horse Full 에서 14% 더 나은 L1 Loss (3-trial 평균 V546). Densification 단계 동안만 활성화 (T2 까지), 그 후 학습은 Decay 없이 계속 실행되어 Refinement에서 확립된 Opacity가 안정적으로 유지됩니다. MCMC에서는 사용되지 않습니다 (MCMC는 T67 `mcmcRelocationInterval` + T68 `mcmcDeadOpacityThreshold`를 통해 자체 메커니즘이 있음).

 간단히 말하면

```

[?] [?] [?] [?] [?] [?] Splat [?]
"[?] [?] [?] [?] [?] [?]". [?] [?] Splat
[?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?]
[?]: 14% [?] [?]. Quality [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?]
    
```


T37 opacityDecayInterval

 세부 정보

기본값: 50 **범위:** 10 – 500 **정의 위치:** TrainingConfig.swift

 기술적 설명

T36 `opacityDecayFactor`가 적용되는 반복 간격. HTGS 논문 기본값 50, `.full` 에서 유지. 긴 간격 (>200) 은 효과를 부분적으로 상쇄합니다. 두 적용 사이에 Opacity가 다시 증가할 만큼 충분한 그래디언트 업데이트가 발생하기 때문입니다. 짧은 간격 (<20) 은 Decay를 너무 공격적으로 만듭니다. Densification 단계에서만 활성화.

 간단히 말하면

```

"[?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?]
    
```

Refinement (T38–T44)

T38 gradientAccumulationSteps

세부 정보

기본값: 1 (= “Adam 단계당 하나의 뷰”) **범위:** 1 – 8 **정의 위치:** TrainingConfig.swift

기술적 설명

V424 기능: Adam 업데이트가 실행되기 전에 그래디언트가 누적되는 뷰의 수. > 1 에서 앱은 그래디언트를 별도의 버퍼에 합하는 별도의 “unfused” Backward Project 경로에서 실행됩니다. 최종 적용은 크기를 일정하게 유지하기 위해 1/N로 스케일됩니다. V424는 2-뷰를 테스트했습니다 → 품질 중립이지만 10% 더 느림 (unfused 경로가 fused 경로보다 비싸기 때문에). .full 에 대해 revert되었지만 MCMC에 대해서는 의도적으로 사용 — .fullMCMC 는 함께 실행되지만 V544a 테스트는 Classic과의 품질 격차가 11% 대신 5%로 줄어든다는 것을 보여 주었습니다. 이니셜라이저 기본값은 1, 현재 프리셋도 1, CLI 플래그 (--accum-steps N) 로 유지됩니다.

간단히 말하면

Splat [? ? ? ? ? ? ? ? ? ?]
[? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?]. 1
= [? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?]. [? ?]
[? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?]
[? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?]. [? ? ? ?]
[? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?].
MCMC [? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?]
[? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?].

T39 testViewIndices

세부 정보

기본값: [] (= 비어 있음, 모든 뷰가 학습에 사용) **범위:** Set<Int>, 카메라 인덱스의 임의 하위 집합 **정의 위치:** TrainingConfig.swift

기술적 설명

V546 기능: 학습에 사용되지 않고 PSNR/SSIM/LPIPS 평가용 Holdout으로 저장되는 카메라 인덱스 집합. --benchmark CLI 플래그가 활성화일 때 자동으로 설정됩니다: 그러면 인덱스 0부터 시작하여 매 8번째 뷰 (LLFF 표준, Mip-NeRF 360 및 3DGS 논문 규약과 동일). benchmark 없이는 비어 있음 — 학습이 모든 뷰를 사용합니다. **주의:** 인덱스 이해 없이 이 필드를 수동으로 설정하면 벤치마크가 사용 불가능해질 수 있습니다 (예: N-50 뷰만 있는데 모든 인덱스가 N 이상으로 설정됨 → Holdout 없음 → 평가 없음). 자체 프리셋 내보내기 시 testViewIndices는 영구화되지 않습니다. 장면 의존적이며 그렇지 않으면 다른 데이터셋 사이에 무의미한 값을 남길 것이기 때문입니다.

간단히 말하면

[? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?]
[? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?]
[? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?].
--benchmark [? ? ? ? ? ? ? ?]
[? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?] (? 8 ? ? ? ? ? ?)
[? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?]
[? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?]
[? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?].


T40 refinementPruneInterval

 세부 정보

기본값: 0 (= 비활성화) **범위:** 0 또는 100 – 5 000 **정의 위치:** TrainingConfig.swift

 기술적 설명

V425 기능: Refinement 단계 (T2 후) 동안 매 N 반복마다 sigmoid(opacity) < T41 refinementPruneOpacityThreshold의 Gaussian을 제거하는 추가 Prune 패스가 실행됩니다. 의의: Densification 동안에는 정기적인 Density Control 호출이 있지만 그 후에는 없습니다 — Opacity가 계속 감소하는 Gaussian은 버퍼에 남습니다. V425는 테스트하고 revert했습니다: 추가 Pruning은 V426 (Two-Phase Densification, 마찬가지로 0 Gaussian 캐스케이드 실패로 중단) 과 상관관계가 있었습니다. 비활성화. 실험을 위한 CLI 플래그 사용 가능. 활성화되면 1 000 또는 2 000이 합리적인 값입니다.

 간단히 말하면

Refinement [?] [?] [?] [?] [?] [?] [?] [?]
 [?]. [?] [?] [?] [?] [?] [?] [?] [?], [?] [?]
 [?] [?] [?] [?] [?] [?] [?].


T41 refinementPruneOpacityThreshold

 세부 정보

기본값: 0.0 (= "T14 사용") **범위:** 0 또는 0.001 – 0.1 **정의 위치:** TrainingConfig.swift

 기술적 설명

V425b: Refinement Pruning을 위한 별도의 Opacity 임계값. Densification 후 대부분의 Gaussian은 상당히 더 높은 Opacity (> 0.001) 에 도달하므로 표준 T14 pruneOpacityThreshold가 너무 느슨합니다. T40 이 활성화되면 이 필드가 자체 임계값을 결정합니다. 0.0에서는 T14 가 계속 사용됩니다. T40 > 0일 때만 관련이 있습니다.

 간단히 말하면

[?] [?] Refinement [?] [?] (T40 [?] [?])
 [?] [?] [?] [?]. [?] [?] [?] [?] [?] [?] [?]
 [?] [?] [?] [?] [?] [?] [?].

T42 midTrainingCompactificationIterations

☰ 세부 정보

기본값: [1] (= 비활성화) **범위:** [Int], 값이 (densifyUntilIteration, maxIterations) 안에 정의
위치: TrainingConfig.swift

⚙ 기술적 설명

V549 기능: Refinement 단계 동안 Compactification 패스가 실행되는 명시적 반복 지점 (sigmoid(opacity) < 0.01 + 아웃라이어 스케일 Gaussian을 제거, T56 postTrainingCompactification과 같은 로직). 의의: 긴 Refinement 단계가 SH가 부별 아티팩트에 Overfit하는 Confetti/Floater 누적을 보여 줄 수 있습니다. 활성화된 경우 일반적인 구성: 40K Classic용 [10000, 20000, 30000]. 그러나: Family 데이터셋의 V549 A/B 테스트는 모든 구성에서 더 나쁜 L1을 보여 주었습니다: [10K, 20K, 30K]\@0.01 → -48% Count이지만 +36% L1. [20K, 30K]\@0.005 → -44% Count이지만 +45% L1. [20K, 30K]\@0.001 → -17% Count이지만 +87% L1. 따라서 비활성화. CLI 플래그 --mid-compact "10000,20000"은 시각적 플로터 트레이드 오프 (뷰포트에서 더 적은 Confetti) 를 Loss 회귀보다 선호하는 경우 사용 가능합니다.

💬 간단히 말하면

?? ? ? ? ? ? ? ? ? . ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? (? ? ? ? ?
?? ? ? ? ? ? ? ? ?) . ? ? , ?
?? ? ? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ? ? ?

T43 frustumCullEnabled

세부 정보

기본값: false **범위:** boolean **정의 위치:** TrainingConfig.swift

기술적 설명

V549b 기능: 학습 후 모든 학습 카메라 Frustum의 합집합 밖에 있는 모든 Gaussian이 제거됩니다. 이러한 Gaussian은 Loss 신호에 의해 결코 제약되지 않았으며 항상 플로터입니다. Novel View가 카메라 경로 뒤나 옆에 있는 장면 (예: 선형 드론 비행의 뒷면) 에 특히 효과적입니다 — 거기의 플로터는 학습 단계에서 결코 보이지 않지만 나중에 3D 뷰어에서 움직일 때 매우 보입니다. V549b A/B는 드론 비행에서 긍정적인 결과를 보여 주었으므로 Opt-In으로 사용 가능합니다. 기본값은 false. 전체 궤도 커버리지가 있는 객체 캡처에서는 Frustum Union이 전체 장면을 포함하고 기능이 아무것도 제거하지 않기 때문입니다 — Settings의 "Floater Reduction"에서 제공되며 Q9 Outdoor 프리셋에서 T44 frustumCullExpansion 을 통해 암묵적으로 테스트됩니다 (Q7-BayesOpt는 이를 활성화하지 않았습니다. Outdoor Sky Dome이 같은 문제를 더 잘 해결하기 때문에).

간단히 말하면

FrustumCullEnabled는 학습 후 모든 학습 카메라 Frustum의 합집합 밖에 있는 모든 Gaussian이 제거됩니다. 이러한 Gaussian은 Loss 신호에 의해 결코 제약되지 않았으며 항상 플로터입니다. Novel View가 카메라 경로 뒤나 옆에 있는 장면 (예: 선형 드론 비행의 뒷면) 에 특히 효과적입니다 — 거기의 플로터는 학습 단계에서 결코 보이지 않지만 나중에 3D 뷰어에서 움직일 때 매우 보입니다. V549b A/B는 드론 비행에서 긍정적인 결과를 보여 주었으므로 Opt-In으로 사용 가능합니다. 기본값은 false. 전체 궤도 커버리지가 있는 객체 캡처에서는 Frustum Union이 전체 장면을 포함하고 기능이 아무것도 제거하지 않기 때문입니다 — Settings의 "Floater Reduction"에서 제공되며 Q9 Outdoor 프리셋에서 T44 frustumCullExpansion 을 통해 암묵적으로 테스트됩니다 (Q7-BayesOpt는 이를 활성화하지 않았습니다. Outdoor Sky Dome이 같은 문제를 더 잘 해결하기 때문에).

T44 frustumCullExpansion

세부 정보

기본값: 1.1 **범위:** 1.0 – 2.0 **정의 위치:** TrainingConfig.swift

기술적 설명

T43 frustumCullEnabled에 대한 NDC 마진. 1.0은 이미지 가장자리에서 정확히 잘릴 것입니다. 이는 가장자리의 흔들리는 Splat을 너무 많이 자를 것입니다. 1.1 = 정확한 카메라 프레임 너머 10% 패딩 — 약간 오프셋된 Novel View에서 보일 수 있는 가장자리 픽셀에 대한 약간의 허용 오차를 제공합니다. > 1.2 값은 Cull을 실질적으로 비효과적으로 만듭니다. 확장된 Frustum이 훨씬 더 많은 공간을 포함하기 때문입니다.

간단히 말하면

FrustumCullExpansion은 T43 frustumCullEnabled에 대한 NDC 마진입니다. 1.0은 이미지 가장자리에서 정확히 잘릴 것입니다. 이는 가장자리의 흔들리는 Splat을 너무 많이 자를 것입니다. 1.1 = 정확한 카메라 프레임 너머 10% 패딩 — 약간 오프셋된 Novel View에서 보일 수 있는 가장자리 픽셀에 대한 약간의 허용 오차를 제공합니다. > 1.2 값은 Cull을 실질적으로 비효과적으로 만듭니다. 확장된 Frustum이 훨씬 더 많은 공간을 포함하기 때문입니다.

Sky-Dome (T45–T48)

T45 skyDomeEnabled


 세부 정보

기본값: false (이니셜라이저 + P9 Outdoor를 제외한 모든 프리셋) **범위:** boolean **정의 위치:** TrainingConfig.swift

 기술적 설명

V549e 기능: 학습 시작 전에 구형 점 구름이 생성되고 (T46 샘플 포인트가 있는 Fibonacci Sphere), 장면 중심 주위에 T47

skyDomeRadiusMultiplier × scene_extent 반경으로 배치되며, 모든 학습 카메라의 Sky Masked 픽셀의 색상으로 초기화됩니다 (T20 skyMaskingEnabled 참고). 이 Sky Dome Gaussian은 Gaussian 버퍼의 시작에 삽입되고 학습 중 “동결”됩니다 (Position/Scale/Rotation Gradient = 0, SH와 Opacity만 최적화 가능). 효과: 멀리 있는 검은 “Confetti” 영역 대신 사용자가 Novel View에서 실제 하늘을 봅니다. V549e MVP는 드론 및 풍경 장면에서 매우 잘 작동합니다. P9 Outdoor 프리셋에서 기본적으로 켜져 있습니다. 실내 장면에서는 꺼 두십시오 — 구체가 무의미하게 방 밖에 매달릴 것입니다.

 간단히 말하면

?? ???? ? "?? ?" ? ?
 ?? . ?? ???? ? ? ? ? ?
 ? ? ? ? ? : ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? . ? ? ? ? ? ?
 ? ? ? ? ? , ? ? ? ? ? ? ? ?
 ? ? .


T46 skyDomeSampleCount

 세부 정보

기본값: 5 000 **범위:** 1 000 – 50 000 (일반적으로 2 000 – 10 000) **정의 위치:** TrainingConfig.swift

 기술적 설명

Sky Dome 구체의 Fibonacci Sphere 샘플 포인트 수. 더 높은 값 → 더 조밀한 Sky Dome (큰 해상도와 많이 보이는 하늘에 더 좋음), 그러나 더 많은 메모리 요구량. 5 000은 4K 렌더링용 스위트 스폿입니다. 더 낮은 해상도에서는 2 000–3 000으로 충분합니다. 적은 해당 Sky Masked 픽셀로 각 학습 카메라 View 벡터에 대한 Cosine Distance에 따라 초기화됩니다 — View Cone에 카메라가 보이지 않는 샘플 포인트는 낮은 Opacity 초기값으로 뒤에 남지만 학습에서 변경되지 않습니다 (동결).

 간단히 말하면

?? ???? ???? ???? ? . 5
 000 ???? ? ? ? ? ? ? ? ? ? .
 ? ? ? = ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? .

T47 skyDomeRadiusMultiplier

☰ 세부 정보

기본값: 30.0 (이니셜라이저 + 대부분의 프리셋), 59.0 (P9 Outdoor, Q7 BayesOpt 최적) **범위:** 5.0 – 200.0 **정의 위치:** TrainingConfig.swift

☞ 기술적 설명

장면 범위 (= 카메라 위치 사이의 평균 거리) 에 상대적인 Sky Dome 구체 반경. 30 = 구체가 카메라 클라우드의 30배 직경을 가집니다. 너무 작음 (< 5) → Sky Dome이 장면 자체와 간섭 (예: Sky Dome Splat이 전경에 떨어짐). 너무 큼 (> 100) → Sky Dome 위치의 float32 정밀도 손실로 거리에서 렌더 글리치 트리거. Q7 BayesOpt는 Bicycle (Mip-NeRF 360) 에서 야외용 장면별 최적값으로 59.0을 발견했습니다 — 이는 표준 30.0이 깊은 풍경에 너무 작고 Sky Dome 픽셀이 이미지 가장자리 영역에서 “벽”으로 보이게 렌더링됨을 시사합니다.

💬 간단히 말하면

?? ? ? ? ? ? ? ? ? ? ?
???. 30 = ? ? ?. ? ?
?? ? ? 50–60 ? ? ? ? ? ?
(Outdoor ? ? ? ? ? ? ? ? ?
?). ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ?
??.

T48 frozenGaussianCount

☰ 세부 정보

기본값: 0 (= 동결된 Gaussian 없음) **범위:** 0 또는 1 – T46 **정의 위치:** TrainingConfig.swift

☞ 기술적 설명

옵티마이저에서 Position/Scale/Rotation Gradient가 0으로 설정되는 버퍼 시작 부분의 Gaussian 수 — 전체 학습에 걸쳐 공간적으로 경직된 상태로 유지됩니다. Density Control은 이를 클론, 스플릿 또는 Prune해서는 안 됩니다. Sky Dome Injection에 사용됩니다 (T45 참고): Sky Dome이 켜져 있으면 이 필드가 자동으로 T46 skyDomeSampleCount로 설정됩니다. 수동 설정이 가능합니다 (예: LiDAR 스캔의 사전 배치된 점 구름을 동결하기 위해) 만 UI에서 직접 액세스할 수 없습니다. 중요: 버퍼의 처음 N Gaussian은 항상 동결됩니다 — 명시적 인덱스가 아니라 버퍼의 순서가 결정합니다.

💬 간단히 말하면

?? ? ? ? ? ? ? Splat ? ?
?? ? ? ? ? ? ? ? ? ? ? . Sky
Dome ? ? ? ? ? ? ? ? ? ?
Sky Dome ? ? ? ? ? ? ? ? . ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? .

Adam + LR 스케줄 (T49–T55)

T49 adamResetIteration

세부 정보

기본값: 0 (= 비활성화) 범위: 0 또는 100 – maxIterations
정의 위치: TrainingConfig.swift

기술적 설명

V430 기능: Adam 옵티마이저 Momentum Accumulator (m1, m2) 가 0으로 재설정되는 반복. 그 후 Bias 보정은 iter 대신 (iter - adamResetIteration)로 실행됩니다. V430은 5 000에서 Reset을 테스트했습니다 (Densification 종료 후) → 12.8% 더 나쁜 Loss. 이유: Densification 동안 구축된 Adam Momentum이 일반적인 그래디언트 크기에 대한 정보를 가지고 있고 Refinement 단계를 가속화합니다. 이를 버리면 Refinement의 처음 ~500 반복의 수렴 비용이 듭니다. 비활성화. 연구 실험을 위한 CLI 플래그.

간단히 말하면

Adam "Adam" Reset. . . ,

T50 positionLRScheduleEndIteration

세부 정보

기본값: 0 (이니셜라이저 = "maxIterations 사용"), 20 000 (.full — maxIter=35K에도 불구하고 Cosine이 20K에서 종료), 30 000 (.fullClassicPaper) 범위: 0 또는 1 000 – maxIterations 정의 위치: TrainingConfig.swift

기술적 설명

V431 기능: Position LR Cosine Annealing 곡선이 최솟값에 도달하는 반복. 0이면 T1 maxIterations 와 동일합니다. > 0이면 스케줄이 이 값까지 실행되고 그 후 T4 positionLearningRateFinal에서 일정하게 유지됩니다. 이는 최소이지만 일정한 학습률로 "확장된 Refinement 단계"를 허용합니다 — 새 Decay 없이 위치를 천천히 정제합니다. .full 은 이를 수행합니다 (스케줄 종료가 20K, 학습이 35K까지 실행). V434c/V434d는 다음을 확인했습니다: 15K와 25K가 거의 같음, 20K가 미세하게 최적. 확장 단계에서 비 Position LR도 수정하기 위해 T51 과 함께 사용됩니다.

간단히 말하면

Position "Position" Quality


T51 extendedPhaseLRDecay

 세부 정보

기본값: 0.0 (= 비활성화, 일정한 LR) 범위: 0 또는 0.01 – 1.0 정의 위치: TrainingConfig.swift

 기술적 설명

V433 기능: “확장 단계”에서 비 Position LR (스케일, 회전, Opacity, SH) 의 최소 승수 — 즉: T_{50} 이 도달되어 Position LR이 이미 T_4 에 있을 후. 0.1이면 스케일/회전/Opacity/SH가 1.0 (= 표준 LR) 에서 표준의 $0.1 \times$ 로 cosine-decay됩니다. 0.0 (기본값) 이면 일정하게 유지됩니다. V457은 전체 Decay (0.0 = 0까지 Decay) 를 no-decay에 대해 테스트했고 다음을 발견했습니다: 평균 0.0400 (2 Run) = Decay 없는 V438과 같은 Loss. Decay로 동작이 더 깨끗하지만 측정 가능하게 더 나음은 아닙니다. 따라서 비활성화. CLI에 `--nonpos-lr-scale F`로 유지됩니다.

 간단히 말하면

?? ? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ? ? ?
?? “? ? ? ? ?” ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ? ? ? ?


T52 adaptiveDensifyThreshold

 세부 정보

기본값: false 범위: boolean 정의 위치: TrainingConfig.swift

 기술적 설명

V440 실험적: true인 경우 앱은 각 Densification 단계에서 현재 그래디언트 분포의 p98을 계산하고 동적 임계값으로 사용합니다 (구성된 T_{11} 값의 최소 $0.5 \times$ 에 클램프되어 너무 많이 벗어나지 않도록 함). 가설: 현재 장면 단계에 대한 자동 적응이 Density Control을 더 강건하게 만들 것 — 예: 처음에는 더 엄격한 Pruning, 나중에는 더 느슨하게 또는 그 반대. V440은 테스트하고 revert했습니다: 63 K Gaussian으로의 재앙적 하락 (Mass Pruning, 처음 반복에서 p98이 극도로 높고 그 다음 거의 아무것도 임계값을 초과하지 않기 때문에). 고정된 임계값이 이미 잘 보정되어 있고 동적 적응이 도움이 되기보다 해를 끼칩니다. Q5 (T77) 는 Rolling Median을 통한 대체 Adaptive 로직을 제공하여 문제를 우회합니다.

 간단히 말하면

Densify ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? (Splat ? ? ? 63K
? ? ? ? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? ? ? ? ?


T53 mergeAfterDensification

 세부 정보

기본값: false (이니셜라이저), true (`.full` , `.classicBalanced` , `.fullClassicPaper`) **범위:** boolean **정의 위치:** TrainingConfig.swift

 기술적 설명

V438 기능: Densification 단계 종료 시 (Iter T2) 유사한 스케일과 색상으로 가까이 있는 Gaussian을 결합하는 일회성 Merge 패스가 수행됩니다. 가시적 품질 손실 없이 Gaussian 수를 일반적으로 5-15% 줄입니다. 의의: 강한 Clone 후 거의 동일한 Gaussian 클러스터가 발생하며 이는 새로운 것을 기여하지 않습니다 — Merging은 다른 영역에 옵티마이저 용량을 해제합니다. Classic Quality 프리셋의 표준. MCMC에서는 사용되지 않습니다. MCMC가 Relocation 로직을 통해 이러한 클러스터가 처음부터 발생하지 않게 하기 때문입니다.

 간단히 말하면

Splat `??????????`
`??????????????.??`
`??????????????`
`???.Quality??????`
`??????????`.


T54 densifyPhase2FromIteration

 세부 정보

기본값: 0 (= 비활성화) **범위:** 0 또는 T2 - T1 **정의 위치:** TrainingConfig.swift

 기술적 설명

V426 실험적: Refinement 일시 정지 후 이 반복에서 시작하여 T55 까지 실행되는 두 번째 Densification 단계를 가능하게 합니다. 가설: Refinement 단계 후 그래디언트 Accumulator가 더 안정적인 크기를 가지고 추가 Gaussian이 필요한 영역을 더 정확하게 말할 수 있을 것입니다. V426은 테스트하고 revert했습니다: Two-Phase Densification이 0 Gaussian 캐스케이드 실패에 빠짐 (V425 Refinement Pruning과 결합하여 버퍼를 파괴함). 비활성화. 실험을 위한 CLI 플래그 사용 가능.

 간단히 말하면

`??????????????`
`???.???????? Splat????`
`?????????.??`.


T55 densifyPhase2Untilliteration

 세부 정보

기본값: 0 **범위:** 0 또는 T54 - T1 **정의 위치:** TrainingConfig.swift

 기술적 설명

V426 Two-Phase Densification의 종료. T54 > 0 일 때만 관련 있음. 두 필드 모두 비활성화.

 간단히 말하면

`????????????`
`(T54??).????`.

Post-Processing + Apple AI (T56–T60)


T56 postTrainingCompactification

 세부 정보

기본값: true (모든 Production 프리셋), false (.quickTest, .preview) **범위:** boolean **정의 위치:** TrainingConfig.swift

 기술적 설명

V443 기능: 학습 종료 후 sigmoid(opacity) < 0.01의 Gaussian이 하드 제거됩니다 (실질적으로 더 이상 이미지에 기여하지 않음). 가시적 품질 손실 없이 Gaussian Count를 일반적으로 58%, 내보내기 파일 크기를 55% 줄입니다. Production 프리셋에서 표준으로 활성화 — 최종 결과를 가능한 한 콤팩트하게 제공해야 합니다. .quickTest에서는 꺼짐. 진단 실행이 어쨌든 내보내지지 않기 때문입니다. T42 midTrainingCompactificationIterations (V549)와 달리 Compactification은 끝에서만 발생합니다 — Refinement는 그때까지 모든 Gaussian을 사용할 수 있습니다.

 간단히 말하면

?? ? ??: ?? ? ? ? ?
 ? Splat? ? ? ? ? ? . ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ,
 ? ? ? ? ? ? ? ? ? ? ? ? ? .


T57 metalFXUpscaling

 세부 정보

기본값: false **범위:** boolean **정의 위치:** TrainingConfig.swift

 기술적 설명

V444 기능: 3D Viewer 출력에서 Bilinear Interpolation 대신 Apple의 MetalFX Spatial Upscaler를 활성화합니다. 학습 해상도 < 뷰포트 크기 (예: 0.5x 학습, 전체 해상도 뷰포트 표시)인 경우 MetalFX는 훨씬 더 선명한 이미지를 제공할 수 있습니다. 뷰포트에서 실시간으로 변경됩니다. 재학습 필요 없음. T58 mpsLanczosScaling과 상호 배타적 — MetalFX가 우선합니다. 권장: 이미지가 예상 세부 정보에 비해 뷰어에서 "흐릿"하게 나타나면 켜십시오.

 간단히 말하면

3D Viewer? Apple ML ? ? ? ?
 ? ? ? ? ? . ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? . ? ?
 ? ? ? , ? ? ? ? ? ? ? ? .


T58 mpsLanczosScaling

 세부 정보

기본값: false **범위:** boolean **정의 위치:** TrainingConfig.swift

 기술적 설명

V444 기능: Bilinear Interpolation 대신 뷰포트 스케일링용 MPSImageLanczosScale. Lanczos는 Bilinear보다 훨씬 더 선명한 결과를 최소한의 오버헤드로 제공하는 고전적 Sinc 기반 재샘플링 절차입니다. 실시간 토글. 둘 다 켜져 있으면 T57 에 의해 덮어쓰여집니다.

 간단히 말하면

3D Viewer `mpsLanczosScaling` (Lanczos). MetalFX (T57) `mpsLanczosScaling`. Lanczos `mpsLanczosScaling`.


T59 livePreviewInterval

 세부 정보

기본값: 50 (이니셜라이저와 대부분의 프리셋) **범위:** 0 (off) 또는 10 – 5 000 **정의 위치:** TrainingConfig.swift

 기술적 설명

학습 중 3D Viewer가 현재 Gaussian으로 업데이트되는 빈도. 50 = 50회 반복마다 Viewer에 새 Render — 학습을 느끼게 하지 않고 진행을 관찰하기에 충분히 좋음. 0 = Viewer가 전혀 업데이트되지 않음 (백그라운드 학습, 최대 속도). 일반적인 조정: `.quickTest` 에서 10으로 낮춤 (모든 단계를 보고 싶음), 긴 MCMC 실행에서 500–2000으로 높임 (Update 오버헤드가 합계로 눈에 띈).

 간단히 말하면

3D Viewer `livePreviewInterval`. 50 = 50회 반복마다 `livePreviewInterval` = `livePreviewInterval`. 0 = `livePreviewInterval` (Viewer가 전혀 업데이트되지 않음).


T60 perceptualLossWeight

 세부 정보

기본값: 0.0 (= 비활성화) **범위:** 0 또는 0.001 – 0.5 **정의 위치:** TrainingConfig.swift

 기술적 설명

V444 미래 기능: MPSGraph (VGG와 비슷한 작은 네트워크)를 통한 Perceptual Loss 항의 가중치. L1+SSIM보다 더 높은 의미 수준에서 구조적 및 텍스처 유사성을 포착할 것입니다 — “Pixel-Perfect”가 “사실적으로 보임”보다 덜 중요한 연구 파이프라인에서 일반적입니다. 구현은 아직 보류 중 (Code Stub 존재하지만 Forward Pass는 구현되지 않음). 기본값 0.0. 향후 활성화를 위해 필드 카탈로그에 유지됩니다. CLI 플래그 `--percep-weight F` 예약.

 간단히 말하면

AI `perceptualLossWeight` “Pixel-Perfect” (Pixel-Perfect). `perceptualLossWeight`.

MCMC Densification (T61–T73)

T61 densificationStrategy

☰ 세부 정보

기본값: `.classic` (이니셜라이저 + Classic 프리셋), `.mcmc` (모든 MCMC 프리셋 + Scene-Class) **범위:** `.classic` 또는 `.mcmc` **정의 위치:** TrainingConfig.swift

⚙️ 기술적 설명

Classic Densification (Clone/Split/Prune, Kerbl 외 2023) 과 MCMC Densification (Relocation을 사용한 Stochastic Gradient Langevin Dynamics, Kheradmand 외 NeurIPS 2024) 사이의 선택.

`.classic`에서는 T11--T16 이 평가되고, `.mcmc`에서는 T62--T73 이 평가됩니다. 전환 시 주의: Classic 기본값과 MCMC 기본값은 완전히 다르게 보정되어 있습니다 — Expert View의 선택기를 Flip하는 사람이 적절한 프리셋을 로드하지 않으면 1.4.3 버그 스타일 대량 멸종 (한 반복에서 460 K → 5) 의 위험이 있습니다. MCMC OpacityReg 0.01이 Classic Opacity를 죽이기 때문에. 따라서 MCMC Init 기본값은 의도적으로 "부드럽게" (모든 Reg 값 0.0).

💬 간단히 말하면

Splat [?][?][?][?] [?][?] [?][?][?][?][?] [?][?][?][?][?]. Classic = [?][?] [?][?] ([?][?], [?][?] Splat). MCMC = [?][?] [?] ([?][?] [?][?], [?][?] [?][?] Splat, [?] [?] [?][?][?][?]). [?][?][?][?] [?][?][?] [?][?][?][?][?][?]. [?][?][?][?][?] [?][?] [?][?][?] [?][?][?] (P5–P7 [?][?] P8–P10) [?] [?][?][?] [?][?] [?][?][?] [?].

T62 mcmcMaxGaussians

세부 정보

기본값: 150 000 (이니셜라이저 + `.fullMCMC + .mcmcBalanced`), 100 000 (`.mcmcPreview`), 1 500 000 (`.fullMCMCMip - 10x` 예산이 있는 Mip-Splatting 변형), 1.19 M (`.renderPreset`), 1.25 M (`.outdoorPreset`), 670 K (`.indoorPreset`) **범위:** 0 (= "버퍼 용량 사용") 또는 10 000 - 5 000 000 **정의 위치:** TrainingConfig.swift

기술적 설명

MCMC 전략에서 Gaussian 수의 하드 상한. 수는 이 Cap까지 Relocation 단계당 `T70 mcmcGrowthRate` (일반적으로 5%) 로 점진적으로 증가합니다. V473/V531은 150 K를 스위트 스폿으로 발견했습니다 — 200 K 이상에서는 Splat 품질이 희석됩니다 (너무 많은 작은, 중복된 Gaussian). 100 K 미만에서는 장면이 under-densify 상태로 유지됩니다. 매우 큰 장면 (예: 158 K SfM 초기화의 1 545 사진 드론 비행) 에서는 150 K가 너무 낮습니다 — 따라서 1.4.5 확장 `T72 mcmcCapMultiplier + T73 mcmcAutoScaleByScene`. Q7 BayesOpt는 670 K (Indoor) 에서 1.25 M (Outdoor) 사이의 장면별 최적값을 발견했습니다. 값 0에서 엔진은 전체 버퍼 용량을 Cap으로 사용합니다.

간단히 말하면

MCMC `?? ??` Splat `?. 150 000?? ???? ???? ?`
`?? ?????. Outdoor ?`
Render `???` (P8, P9) `?? ?`
`?? ? ? ? ? ? ? ? ?`
`100?+? ???? ??. ??? ?`
`?? ? ? ? ? ? ? ? ?`
`?? ??. ? ? ? ? ? ? ? ?`
`?? ? ? ? ? ? ? ? ?.`

T63 mcmcNoiseScale

세부 정보

기본값: 0.00005 (5e-5 = 논문 기본값) **범위:** 1e-6 - 1e-3 **정의 위치:** TrainingConfig.swift

기술적 설명

각 MCMC 반복에서 각 Gaussian의 위치에 추가되는 Gaussian 노이즈의 승수 (SGLD 로직). 더 높음 = 더 많은 탐색 (Gaussian이 더 많이 방황, 잠재적으로 더 나은 자리를 찾음), 더 낮음 = 더 많은 이용 (Gaussian이 이미 좋은 곳에 머무름). V467과 V536은 5e-5를 최적으로 확인했습니다 — 1e-5/2e-5는 너무 적은 탐색, 1e-4는 너무 많음 (Splat이 흩어짐). 학습 시간에 걸쳐 `T69 mcmcNoiseDecayEnd`까지 cosine-decay됩니다 — Decay 범위 종료 시 노이즈가 효과적으로 0이고 Gaussian이 수렴합니다.

간단히 말하면

Splat `?? ? ? ? ? ? ? ? ?`
`?? ? ? ? ? ? ? ? ? ?`
`?? ? ? ? ? "???" ? ? ? ? ?`
`?. ? ? ? ? ? ? ? ? ? ? ? ?`
`???. ?? ? ? ? ? Splat ? ? ?`
`?? ? ? ? ?.`


T64 mcmcOpacityRegWeight

 세부 정보

기본값: 0.0 (= RadianceKit 기본값에서 비활성화, 논문: 0.01) **범위:** 0 또는 0.001 – 0.05 **정의 위치:** TrainingConfig.swift

 기술적 설명

Opacity에 대한 MCMC 특정 L1 페널티. 논문 기본값 0.01 (사용되지 않은 Gaussian을 0으로 누르고, Relocation에 사용할 수 있게 만듭니다). 그러나 V464b는 RadianceKit에서 Reg 없이 측정 가능하게 더 나음을 보여 주었습니다 (Session 28 확인). 이유: T68 mcmcDeadOpacityThreshold로 정의된 Pruning 기준만으로 충분합니다 — 추가 L1 페널티는 가치 있는, 낮은 Opacity Gaussian도 죽도록 강제합니다. 따라서 기본값 0. **주의:** 1.4.3 베타 빌드에서 이니셜라이저 기본값이 잘못 0.01이었고 대량 멸종 버그를 초래했습니다 (T61 설명 참고). 1.4.4부터 0.0으로 고정.

 간단히 말하면

MCMC [?] [?] [?] [?] [?]. [?] MCMC [?] [?] [?] [?] (T68 [?] [?] [?] [?]) [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?].


T65 mcmcScaleRegWeight

 세부 정보

기본값: 0.0 (= 비활성화, 논문: 0.01) **범위:** 0 또는 0.001 – 0.05 **정의 위치:** TrainingConfig.swift

 기술적 설명

스케일 고유값에 대한 MCMC 특정 L1 페널티. 논문 기본값 0.01. V464b: Reg 없이 더 나음, T64와 같은 이유. 모든 RadianceKit MCMC 프리셋에서 비활성화. T64와 같은 주의: 1.4.3 버그.

 간단히 말하면

T64 [?] [?] [?] [?] [?] Splat [?] [?] [?]. [?] [?].


T66 mcmcRelocationInterval

 세부 정보

기본값: 100 (이니셜라이저 + 모든 MCMC 프리셋, 논문 표준), 155 (P9 Outdoor — Q7 BayesOpt 최적) **범위:** 50 – 500 **정의 위치:** TrainingConfig.swift

 기술적 설명

MCMC가 죽은 Gaussian (sigmoid(opacity) < T68 mcmcDeadOpacityThreshold) 을 새 위치로 재배포하는 반복 간격. V537은 50을 테스트했고 (너무 파괴적, Loss 변동) 200을 테스트했습니다 (미세하게 더 나쁨, MCMC가 반응성을 잃음). 100이 최적입니다. Q7 BayesOpt는 Bicycle에서 야외용 장면별 최적값으로 155를 발견했습니다 — 약간 더 긴 간격은 Adam이 다음 Reloc 이벤트가 압력을 가하기 전에 새 로 배치된 Gaussian을 통합할 더 많은 시간을 줍니다.

 간단히 말하면

MCMC Splat 100 — Outdoor
Splat 100. Outdoor
Splat 100. Outdoor


T67 mcmcWarmupIterations

 세부 정보

기본값: 500 **범위:** 100 – 5 000 **정의 위치:** TrainingConfig.swift

 기술적 설명

아직 MCMC Relocation이 발생하지 않는 초기 반복 수. 이 Warmup 후에만 Reloc 로직이 시작됩니다. 의의: 처음 반복에서 Opacity 값이 아직 안정화되지 않았습니다 — Reloc로 바로 시작되면 Gaussian이 잘못된 위치에 배치되어 바로 다시 이동되어야 하며 이는 Adam Momentum을 파괴합니다. 논문 기본값 500. RadianceKit은 V464b가 강건함을 보여 주었기 때문에 이 값을 채택합니다.

 간단히 말하면

MCMC Splat 500
MCMC Splat 500
MCMC Splat 500

T68 mcmcDeadOpacityThreshold

☰ 세부 정보

기본값: 0.005 (이니셜라이저, 논문 표준), 0.01
(.fullMCMC 와 모든 MCMC 프리셋 — V535 최적) **범위:**
0.001 – 0.05 **정의 위치:** TrainingConfig.swift

⚙ 기술적 설명

sigmoid(Opacity) 임계값. 이 아래에서 Gaussian이 “죽음”으로 간주되어 Relocation 대상이 됩니다. V535는 0.01을 최적으로 발견했습니다 (0.005는 미미함, 0.02는 더 나쁨). 더 높음 = 더 공격적인 Reloc (더 많은 Gaussian이 이동됨), 더 낮음 = 더 신중함. 0.01은 대략 “0.5% 시각적 가시성”에 해당합니다. P10 Indoor는 Q7 BayesOpt를 통해 0.0142를 최적으로 사용합니다.

💬 간단히 말하면

Splat “ ”
MCMC
. 0.01
.
.
.

T69 mcmcNoiseDecayEnd

☰ 세부 정보

기본값: 0 (이니셜라이저 = “Decay 없음”), 160 000 (.fullMCMC = 200K의 80%), 96 000 (.mcmcBalanced = 120K의 80%), 40 000 (.mcmcPreview) **범위:** 0 또는 1 000 – maxIterations
정의 위치: TrainingConfig.swift

⚙ 기술적 설명

T63 mcmcNoiseScale 노이즈가 완전히 0으로 감소되는 반복 (Iter 0부터 여기까지 Cosine Decay). V497c/V502는 maxIterations의 80%를 최적으로 발견했습니다 — MCMC에 충분한 탐색 시간을 주지만 마지막 20%를 노이즈 없이 수렴에 남깁니다. 0 = 모든 반복에 걸쳐 일정한 노이즈 (거의 의미가 없음, MCMC가 수렴할 수 없음).

💬 간단히 말하면

Splat “ ”
MCMC
80% — ,
.
.

T70 mcmcGrowthRate

☰ 세부 정보

기본값: 0.05 (논문 표준 = 5%) **범위:** 0.01 – 0.2 정의 위치: TrainingConfig.swift

⚙️ 기술적 설명

Relocation 단계당 MCMC 인구 대상의 성장률. 로직: 각 Reloc 이벤트에서 대상 인구 크기가 $(1 + \text{growthRate})$ 만큼 증가하여 T62 mcmcMaxGaussians (또는 T72/T73을 통해 스케일된 변형) 에 도달합니다. V512/V522는 0.05를 최적으로 발견했습니다 — 더 높은 값은 너무 빠른 성장으로 이어집니다 (Adam Momentum이 통합할 수 있기 전에 Gaussian이 삽입됨), 더 낮은 값은 종료 시 under-densified 장면으로 이어집니다.

💬 간단히 말하면

MCMC `?? Splat ?? ???? ?
?? ?????. ?? ?%?
???. ?? ???? ?????.`

T71 mcmcSigmoidK

☰ 세부 정보

기본값: 100.0 **범위:** 10.0 – 500.0 정의 위치: TrainingConfig.swift

⚙️ 기술적 설명

MCMC 노이즈 감쇠를 위한 Sigmoid Sharpness 매개변수. SGLD 단계에서 Gaussian당 노이즈는 $\text{sigmoid}(K \times \text{logit_opacity})$ 로 감쇠됩니다 — 높은 Opacity Gaussian (Logit이 양수) 은 낮은 Opacity보다 기하급수적으로 적은 노이즈를 받습니다. $K = 100$ 은 선명하며, 즉 Opacity 0.5 주위에서 "풀 노이즈"에서 "노이즈 없음"으로의 전환이 매우 빠르게 발생합니다. V484–V487은 $K = 100$ 을 최적으로 발견했습니다 — 더 작은 값 (10–50) 은 높은 Opacity Gaussian도 흔들리게 합니다 (수렴된 Gaussian 파괴), 더 큰 값 (> 500) 은 전환을 인위적으로 단단하게 만들고 죽은 Gaussian이 전혀 이동되지 않습니다.

💬 간단히 말하면

MCMC `"???? ???? ?
?" "???, ?????"
???? ???? ????
???? ?????. ???
???. ???? ????.`

T72 mcmcCapMultiplier

☰ 세부 정보

기본값: 3.0 (이니셜라이저 + `.fullMCMC`), 2.0 (`.mcmcPreview`), 2.5 (`.mcmcBalanced`), 2.98 (P8 Render), 5.32 (P9 Outdoor), 1.76 (P10 Indoor)
범위: 0 (= 비활성화) 또는 1.0 – 10.0 **정의 위치:** TrainingConfig.swift

⚙️ 기술적 설명

1.4.5 기능: 장면 적응 Cap 스케일링. T73 `mcmcAutoScaleByScene`이 true이면 효과적 Cap이 $\max(T62, \text{initialPointCount} \times T72)$ 로 계산됩니다 (버퍼 용량에 클램프). 배경: 큰 장면에서 (예: 1 545 사진 드론 비행 → 158 K SfM 초기화) $T62 = 150\ 000$ 은 너무 낮습니다 — Density Control이 전혀 성장할 수 없을 것입니다. Multiplier 3.0으로 이 예에서 Cap이 474 K로 스케일됩니다 ($158\ K \times 3.0$). Q7 BayesOpt는 장면별 최적값을 발견했습니다: Outdoor는 높은 Multiplier (5.32 → 156 K bicycle 초기화에서 ~830 K Cap) 로부터 혜택, Indoor는 1.76으로 만족 (벽이 어쨌든 더 빨리 포화). Cap의 전체 해상도는 `resolveMcmcMaxGaussians` 메서드 참고.

💬 간단히 말하면

Splat Cap \approx $\frac{\text{initialPointCount} \times T72}{\text{Density Control}}$ 로 계산됩니다. Outdoor는 높은 Multiplier (5.32 → 156 K bicycle 초기화에서 ~830 K Cap) 로부터 혜택, Indoor는 1.76으로 만족 (벽이 어쨌든 더 빨리 포화). Cap의 전체 해상도는 `resolveMcmcMaxGaussians` 메서드 참고.

T73 mcmcAutoScaleByScene

☰ 세부 정보

기본값: true (이니셜라이저 + 모든 MCMC 프리셋) **범위:** boolean **정의 위치:** TrainingConfig.swift

⚙️ 기술적 설명

1.4.5 기능: 장면 인식 Cap 로직의 마스터 스위치 (T72 1. `resolveMcmcMaxGaussians` 참고). false이면 T62 `mcmcMaxGaussians`만 Cap으로 사용됩니다 (1.4.4 동작으로 되돌림). 1.4.3의 큰 장면에서 대량 멸종 문제가 그렇지 않으면 반복되기 때문에 기본적으로 켜져 있습니다. 명시적으로 하드 Cap을 설정하려는 경우에만 수동으로 비활성화 — 예: 종료 크기가 계획 가능한 150 K 변형을 학습하기 위해.

💬 간단히 말하면

Splat Cap \approx $\frac{\text{initialPointCount} \times T72}{\text{Density Control}}$ 로 계산됩니다. Splat Cap은 장면 인식 Cap 로직의 마스터 스위치 (T72 1. `resolveMcmcMaxGaussians` 참고)에 의해 제어됩니다.

Mip-Splatting (Q1.5) (T74-T76)

상태: Q1.5는 14번의 자율 반복 + Overnight 1.5M Confidence Check 후 2026-05-25에 "closed no-win"으로 폐기되었습니다 (max $\Delta@2\times = +0.27$ dB, 원래 Gate는 0.5x/2x에 대해 평균 $\geq +1.5$ dB을 요구했는데 0/11 Pair Scene에서 FAIL). 필드는 연구 실험을 위해 **opt-in**으로 유지됩니다. 모든 Production 프리셋은 useMipSplatting = false입니다. Verdict 참고: docs/plans/2026-05-25-phase-q1.5-final-verdict.md.

T74 useMipSplatting

세부 정보

기본값: false (모든 Production 프리셋), true (.fullMCMCMip — 연구 형제) **범위:** boolean **정의 위치:** TrainingConfig.swift

기술적 설명

Mip-Splatting (Yu 외 CVPR 2024) 을 활성화합니다: 가장 조밀한 학습 카메라 샘플링 비율의 Nyquist 한계로 Gaussian 당 주파수를 제한하는 3D Smoothing 필터 + 2D 필터 + α 보상. 이론적 목표: 학습 외 스케일 (0.5x 또는 2x 학습 해상도) 에서의 렌더링 시 Aliasing 제거. Preprocess 및 Backward Projection 셰이더에서 활성화되었으며 Q1.5-D 테스트에서 기능적으로 정확하다고 검증됨. 그러나: 원래 수용 Gate ($\Delta@1\times \geq +0.3$ dB 그리고 $\text{avg}(\Delta@0.5\times, \Delta@2\times) \geq +1.5$ dB) 가 11개의 Pair Scene 중 어느 것에서도 달성되지 않았습니다. 최대 관찰: family 750K classic $\Delta@2\times = +0.270$ dB. 야외 장면 (Truck, Flowers) 은 1x와 0.5x에서 악화를 보였습니다. 가설: 3D Smoothing이 high-Gs에서 MCMC Relocation과 경쟁합니다. 필드는 올바른 Mip-NeRF-360 방법론을 사용한 향후 Multi-Scale 재평가를 위해 유지됩니다 (벤치마크 경로의 O3 백로그 참고).

간단히 말하면

2024 Aliasing. Aliasing 제거를 위한 3D Smoothing 필터와 2D 필터를 결합하여 학습 해상도 외의 스케일에서 렌더링 시 발생하는 Aliasing을 제거합니다. 그러나, 원래 Gate를 달성하지 못했습니다. 특히, Truck와 Flowers와 같은 야외 장면에서 1x와 0.5x 해상도에서 악화를 보였습니다. 가설: 3D Smoothing이 high-Gs에서 MCMC Relocation과 경쟁합니다. 필드는 올바른 Mip-NeRF-360 방법론을 사용한 향후 Multi-Scale 재평가를 위해 유지됩니다.


T75 mipSmoothing3DScale

 세부 정보

기본값: 0.2 (논문 기본값) **범위:** 0.05 – 1.0 **정의 위치:** TrainingConfig.swift

 기술적 설명

3D Smoothing 스케일 매개변수 (Yu 외 §3.3, 논문 기본값 0.2). 더 큼 = Gaussian당 더 많은 세계 공간 평활화 (= 더 많은 Anti-Aliasing, 그러나 기본 스케일에서 더 많은 흐림), 더 작음 = 더 선명하지만 Aliasing에 더 취약. T74 useMipSplatting = true일 때만 참조됩니다. Q1.5 테스트에서 더 이상 최적화되지 않았습니다 — A/B Gate가 이미 논문 기본값 0.2로 졌습니다, 추가 Sweep은 무의미할 것입니다.

 간단히 말하면

Mip-Splatting . Mip .


T76 mipFilter2DVariance

 세부 정보

기본값: 0.3 (= 정확히 V242 Legacy 동작) **범위:** 0.1 – 1.0 **정의 위치:** TrainingConfig.swift

 기술적 설명

Σ _2D 대각선에 추가되는 2D Mip 필터 분산 (분산 직접, 제공 아님). 0.3은 Mip-Splatting 전 커널에 하드코딩되어 있던 정확히 V242 Legacy 값입니다. T74 useMipSplatting = false이면 커널이 이 값을 완전히 무시하고 하드코딩된 0.3을 작성합니다 — 따라서 Baseline이 회귀할 수 없습니다 (Codex Round 1 S3-1 보증). useMipSplatting = true이면 여기서 설정된 값이 사용됩니다. Mip Sweep을 위해 필드 카탈로그에 유지됩니다.

 간단히 말하면

Mip-Splatting . Mip : .

Adaptive Densification (Q5) (T77–T79)

T77 adaptiveDensification

☰ 세부 정보

기본값: false 범위: boolean 정의 위치: TrainingConfig.swift

⚙️ 기술적 설명

Q5 기능: 고정 T11 densifyGradThreshold의 대안으로 Rolling Median 추적기. true이면 각 Densify 단계에서 현재 임계값이 median(last N avgGrad samples) × T79 adaptiveDensifyMultiplier로 덮어쓰여집니다. N = T78 adaptiveWindow . V440 p98 (재앙적 63 K Pruning 함정) 보다 엄격함, 중앙값 + 2×는 Steady State 에서 그래디언트 분포의 p70–p80에 있습니다. Q5 테스트: 독립적으로 FAIL 0/3 장면 — Q5+Q6 번들은 2026-05-25에 opt-in으로 통과되었으며 CLI --adaptive-densify 를 통해 활성화할 수 있습니다. 거기서 Q6가 품질 이득의 “Carrier”이고 Q5는 안정성에 기여합니다.

💬 간단히 말하면

Q5는 Densify 단계에서 현재 임계값을 Rolling Median 추적기(중간값 + 2배)로 대체하여, Steady State 에서 그래디언트 분포의 p70-p80에 있습니다. Q5 테스트는 독립적으로 실패했지만, Q5+Q6 번들은 2026-05-25에 opt-in으로 통과되었으며 CLI --adaptive-densify 를 통해 활성화할 수 있습니다. 거기서 Q6가 품질 이득의 “Carrier”이고 Q5는 안정성에 기여합니다.

T78 adaptiveWindow

☰ 세부 정보

기본값: 1 000 범위: 100 – 10 000 정의 위치: TrainingConfig.swift

⚙️ 기술적 설명

Densification 이벤트의 Rolling Median 윈도우 (반복이 아님 — 각 T13 densifyInterval 단계가 하나의 샘플을 제공). 기본값 1 000 — densifyInterval = 100에서 이는 마지막 100 000 학습 반복이 중앙값에 기여한다는 의미이며 일반적으로 여기까지의 전체 학습 기록입니다. 초기 단계 (T78 샘플 전): 추적기는 nil 반환 → 고정 임계값 T11 로 대체. adaptiveDensification = true일 때만 관련 있음.

💬 간단히 말하면

T78은 Densify 단계에서 Rolling Median 윈도우(반복이 아님)를 사용하여, 초기 단계 (T78 샘플 전)에는 고정 임계값 T11로 대체됩니다. adaptiveDensification = true일 때만 관련 있음.


T79 adaptiveDensifyMultiplier

 세부 정보

기본값: 2.0 범위: 1.0 – 4.0 정의 위치:
TrainingConfig.swift

 기술적 설명

적응 임계값에 대한 Rolling Median 승수. 기본값 2.0은 대략 일반적인 그래디언트 분포의 p70–p80에 해당합니다. 더 낮음: 더 공격적인 성장 (더 많은 클론), 더 높음: 더 엄격함 (더 적은 클론). Q5 테스트는 1.5–3.0 범위에서 — 2.0 최고 기본값. adaptiveDensification = true일 때만 관련 있음.

 간단히 말하면

T77/T78 [?] [?] [?]. [?] [?] 2.0 = [?] [?]
[?] [?] [?] [?] [?] [?] [?]. [?] [?] [?] [?]
[?] [?].

Curriculum (Q6) (T80–T81)


T80 curriculumResolutionRamp

 세부 정보

기본값: false 범위: boolean 정의 위치:
TrainingConfig.swift

 기술적 설명

Q6 기능: 학습 해상도가 0.5x에서 시작하여 T50 positionLRScheduleEndIteration / 2 (T50이 설정되지 않은 경우 또는 T1 maxIterations / 2) 에서 T22 trainingRenderScale 로 전환됩니다. Q1.5.1 에서 개발된 resize/restoreImageBuffers 인프라를 사용합니다. 활성화되면 T23 resolutionWarmupScale 을 덮어씁니다. Q6는 Q5+Q6 번들에서 “품질 이득의 Carrier”로 통과되었습니다 (T77 참고) — 단계별 해상도 증가는 앱에 미세한 세부 작업으로 전환하기 전에 더 낮은 해상도에서 거친 형상을 찾을 시간을 줍니다. CLI를 통해: --curriculum-resolution.

 간단히 말하면

[?] [?] [?] [?] [?] [?] “ [?] [?] [?] [?] [?], [?]
[?] [?] [?] [?] [?] [?] “. [?] [?] [?] [?] [?]
[?] [?] [?], [?] [?] [?] [?] [?] [?] [?]. [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
—
T81 [?] [?] [?] [?] [?] [?] [?] [?] [?].


T81 curriculumSHProgression

 세부 정보

기본값: false 범위: boolean 정의 위치:
TrainingConfig.swift

 기술적 설명

Q6 기능: T21 shDegreeUpgradeIterations를 [maxIter/4, maxIter/2, maxIter*3/4]로 덮어씁니다. 즉 SH 업그레이드를 Front Load 대신 학습 시간에 걸쳐 균등하게 분배합니다. 가설: 안정적인 형상은 색상 세부 정보 폭발 전에 확립되며, 이는 View Direction 의존 광택 효과를 더 정확하게 위치시킵니다. Q5+Q6 함께 PASS 1/3 장면, Q6가 이득의 Carrier (Q5 단독 FAIL). CLI를 통해: --curriculum-sh .

 간단히 말하면

“???, ??” — ?
?? ? ? ? ? ? ? ? ?
?? ? ? Splat ? ? ? ? ? ?
?? ? ? ? ? ? ? . T80 ? ? ? ?
?? ? ? ? ? . ? ? ? ? ? ? ? ?
?? ? ? ? ? ? ? ? ? ? ? .


정적 프리셋 (TP1-TP9)

여기서는 이니셜라이저 기본값과의 구조적 차이만. 11개 UI 프리셋 P1-P11의 전체 마케팅 설명은 제 7 장에서 찾을 수 있습니다.

TP1 .preview

 세부 정보

≥ 10 GB RAM 시스템용 진단/미리보기 프리셋. 이니셜라이저에 대한 재정의를: - maxIterations 30 000 → 5 000 - densifyUntillteration 15 000 → 3 500 (maxIter의 70%) - positionLearningRateFinal 1.6e-6 → 1.6e-5 (10배 더 높음, 덜 공격적인 Decay) - shDC/shRest/opacity/scale/rotation LRs 각 2배 (V176) - opacityResetInterval 3 000 → 100 000 (효과적으로 꺼짐, V172: Reset이 짧은 학습 파괴) - [1K, shDegreeUpgradeliterations 2K, 3K] → [1K, 2K] (V182: Degree 3이 2K Iter에서 수렴하지 않음) - trainingRenderScale 1.0 → 0.5


 간단히 말하면

?? ? ? ? ? ? ? ? ? ? ?
?? ? ? ? ? — 2-3 ? ? ? ?
?, ? ? ? ? “Quality ? ? ? ? ?
? ? ? ? ?” ? ? ? ? ? ? ? ?
?? ? ? ? ? .

TP2 .full

 세부 정보

Production-Quality Classic. 재정의: - maxIterations 30 000 → 35 000 (V550: 40K 테스트 Truck 과학 습 +10.7% Gs에 -1.3% L1) - densifyUntilliteration 15 000 → 5 000 (V310 스위트 스폿, V338 7K 더 나뭇) - 모든 LR 2배 (V188) - positionLearningRateFinal 1.6e-6 → 1.6e-5 (V45 10배) - densifyGradThreshold 2e-6 → 1.1e-6 (V335) - densifyInterval 100 → 200 (V112) - pruneOpacityThreshold 0.005 → 0.001 (V393) - opacityResetInterval 3 000 → 100 000 (V194 비활성화, V421 확인) - shDegreeUpgradeliterations [2K, 5K, [1K, 2K, 3K] → 8K] (V228 지연) - opacityDecayFactor 0.0 → 0.9995 (V546 HTGS, 14% 개선) - opacityDecayInterval 50 (변경되지 않음, V546) - mergeAfterDensification false → true (V438) - positionLRScheduleEndIteration 0 → 20 000 (V431) - postTrainingCompactification true (V443, .full에 대해 이미 이니셜라이저 기본값)

 간단히 말하면


< 500 [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] ([?] [?], [?] [?], [?] [?] [?]). V546 [?] [?] [?] [?] V438 [?] [?] 14% Loss [?] [?] [?] Horse Full [?] [?] 3-trial [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?].

TP3 .fullClassicPaper

 세부 정보

TP2의 Q1.5-A 테스트 형제, 논문 충실 Classic. TP2에 대한 재정의:

- maxIterations 35 000 → 30 000 (논문 표준) - densifyUntilliteration 5 000 → 15 000 (논문: maxIter의 50%) - positionLearningRateFinal 1.6e-5 → 1.6e-6 (논문 기본값) - opacity/scale/rotation LR 논문 기본값으로 복귀 (0.05, 0.005, 0.001) - densifyGradThreshold 1.1e-6 → 2e-7 (Bicycle에서 ~1-2M Gs용으로 보정) - densifyInterval 200 → 100 (논문)
- pruneOpacityThreshold 0.001 → 0.005 (논문 기본값) - opacityResetInterval 100 000 → 3 000 (논문 §5.2, 위험 — V194 회귀를 트리거할 수 있음) - opacityDecayFactor 0.9995 → 0.0 (논문에 Decay 없음)
- positionLRScheduleEndIteration 20 000 → 30 000 (Cosine이 maxIter의 100%에서 실행)

 간단히 말하면

Mip-Splatting [?] [?] [?] [?] [?] [?] [?] [?] [?] Gaussian [?] [?] (1-2 M) [?] [?] [?] Q1.5 [?] [?] [?] [?]. Q1.5 "closed no-win" Verdict [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] Production [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?].

TP4 .fullMCMC

 세부 정보


Production-Quality MCMC. 이니셜라이저에 대한 재정의: - maxIterations 30 000 → 200 000 (V534, MCMC가 Classic보다 5배 더 많은 Iter 필요) - densifyUntilliteration 15 000 → 160 000 (V504b maxIter의 80%) - positionLearningRateFinal 1.6e-6 → 1.6e-5 - LR 스케줄 TP2처럼 (모두 2배) - ssimWeight 0.2 → 0.05 (V521b/V534: MCMC에 더 강한 L1 신호 필요)

- shDegreeUpgradeliterations [1K, 2K, 3K] → [2K, 5K, 8K] -

densificationStrategy .classic → .mcmc - mcmcMaxGaussians 150 000 (이니셜라이저에 이미 있음, 프리셋에서 확인) - mcmcNoiseScale 5e-5 (V467/V536 최적) - mcmcDeadOpacityThreshold 0.005 → 0.01 (V535 최적)

- mcmcNoiseDecayEnd 0 → 160 000 (maxIter의 80%, V497c/V502) -

mcmcCapMultiplier 3.0 (이니셜라이저에 이미 있음) - mcmcAutoScaleByScene true (이니셜라이저에 이미 있음) - opacityResetInterval 3 000 → 200 000 (효과적으로 꺼짐, MCMC는 Reset 대신 Reloc 사용)


 간단히 말하면

???, ?? ? ? ? ? ? ?
 ??, ?? ? ? ? ? (P9 Outdoor ?
 ? ? ? ? ? ? ? ?). Classic ? ?
 71% ? ? Gaussian ? ? ? ? ?
 L1.

TP5 .fullMCMCMip

 세부 정보

Mip-Splatting + 논문 규모 MCMC 예산이 있는 TP4의 Q1.5-D 테스트 형제. TP4에 대한 재정의: - mcmcMaxGaussians 150 000 → 1 500 000 (10배, 논문 규모) - useMipSplatting false → true (Mip 켜짐)


 간단히 말하면

?? ? ? ? ? TP4 ? ?
 ?. Q1.5 D-PASS ? Bicycle
 2026-05-24 ? ? (12-Iter Multi-Scale FAIL ? ? ? ?). Q1.5 ?
 ? ? ? 2026-05-25 ? ? ? ?
 closed-no-win — 11 ? ? Pair
 Scene ? ? ? Mip-Splatting ? ?
 ? ? ? ? ? ? ? ? ? ? opt-
 in ? ? ? ? ? ? ? ?.

TP6 .classicBalanced

 세부 정보

Mid-Tier Classic. TP2에 대한 재정의: - maxIterations 35 000 → 20 000 (V149: 20K = 30K, 33% 적은 시간으로) - positionLRScheduleEndIteration 20 000 → 0 (Cosine이 maxIter = 20K에서 실행, 확장 단계 없음)


 간단히 말하면

??? ? ? ? ? ? ? ? ? ? ? ?
V149? ? ? ? ? ? ? ? ? ? ? ?

TP7 .mcmcPreview

 세부 정보

MCMC 진단. TP4에 대한 재정의: - maxIterations 200 000 → 60 000 (V494b) - densifyUntilIteration 160 000 → 48 000 (80%) - mcmcMaxGaussians 150 000 → 100 000 (V473b) - mcmcNoiseDecayEnd 160 000 → 40 000 (V494b) - mcmcCapMultiplier 3.0 → 2.0 (1.4.5: Preview = 가벼운 스케일링)


 간단히 말하면

??? MCMC ? ? ? ? ? ? TP4
?? Scene-Class ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ?

TP8 .mcmcBalanced

 세부 정보

Mid-Tier MCMC. TP4에 대한 재정의: - maxIterations 200 000 → 120 000 (V518) - densifyUntilIteration 160 000 → 96 000 (80%) - mcmcNoiseDecayEnd 160 000 → 96 000 (80%) - mcmcCapMultiplier 3.0 → 2.5 (Preview 2.0과 Full 3.0 사이)


 간단히 말하면

?? 200K ? ? ? ? MCMC.
~120 K ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ?

TP9 .quickTest

 세부 정보

순수 기능 테스트. 이니셜라이저에 대한 재정의: - maxIterations 30 000 → 1 000 - densifyUntilIteration 15 000 → 500 - densifyGradThreshold 2e-6 → 4e-6 (0.25배 해상도용으로 보정) - densifyInterval 100 → 50 - opacityResetInterval 3 000 → 100 000 (꺼짐, 너무 짧기 때문에) - trainingRenderScale 1.0 → 0.25

 간단히 말하면

Sanity Check "?? ? ? ? ?
? ? ? ? ? ? ? ? ? ?". M3 Ultra
?? ? ? ? ? ? ? < 30 s. ? ? ? ? ?
??

메서드: resolveMcmcMaxGaussians

시그니처: public func resolveMcmcMaxGaussians(initialPointCount: Int, bufferCapacity: Int) -> Int **정의 위치:** TrainingConfig.swift

기술적 “MCMC가 최대 얼마나 많은 Gaussian으로 성장하도록 허용해야 하는가?”라는 질문에 대한 유일한 Source-of-Truth. 세 입력에서 계산됩니다: 구성된 T62 mcmcMaxGaussians (0이면 대량 멸종 Floor 150 000), initialPointCount (SfM 초기화 점의 수), bufferCapacity (사전 할당된 Gaussian 버퍼 크기). 로직:

1. `base = T62 > 0 ? T62 : 150_000` (대량 멸종 Floor가 1.4.3 대량 멸종 사건 같은 이니셜라이저 기본값 버그로부터 보호)
2. 만약 `T73 mcmcAutoScaleByScene && initialPointCount > 0 && T72 mcmcCapMultiplier > 0`:
 - `scaled = max(base, ceil(initialPointCount × T72))` 그렇지 않으면
 - `scaled = base`
3. 만약 `bufferCapacity > 0`: `return min(scaled, bufferCapacity)`
4. 그렇지 않으면 `return scaled`

예: Bicycle (Mip-NeRF 360, 194 사진 프레임) → SfM 초기화 ~156 K 점, `T62 = 150 000`, `T72 = 5.32`, 버퍼 용량 8 M. Resolved Cap = `min(8M, max(150K, ceil(156K × 5.32)))` = `min(8M, 830K)` = 830 K. 이는 MCMC Relocation 로직이 준수하는 효과적인 성장 Cap입니다.

간단히 말하면 MCMC에서 실제 최대 Splat 수를 계산합니다. 설정 (mcmcMaxGaussians) 을 가져오고, 장면이 초기에 몇 개의 점을 가지고 있는지 살펴보고, 자동 적응이 켜져 있는 경우 Multiplier로 스케일합니다. 따라서 Cap이 장면에 적응하며, 작은 장면과 거대한 장면에 같은 값을 강제하는 대신. 메서드를 직접 호출할 필요가 없습니다 — 학습이 내부적으로 사용합니다.

어느 필드를 무엇을 위해? (치트 시트)

목표	조정할 필드
거리에 더 많은 세부 정보	T62 mcmcMaxGaussians 높음, T72 mcmcCapMultiplier 5+
일반적으로 더 많은 세부 정보 (Classic)	T1 maxIterations 높음 ($\leq 40K$), T2 densifyUntilIteration T1 의 $\leq 14\%$
드론 비행에서 플로터 감소	T43 frustumCullEnabled 켜짐, T20 skyMaskingEnabled 켜짐, T45 skyDomeEnabled 켜짐
야외 장면에서 멋진 하늘	T45 skyDomeEnabled 켜짐, T47 skyDomeRadiusMultiplier 30-60
더 작은 내보내기 파일	전략 .mcmc (T61), T56 postTrainingCompactification 켜짐, T62 mcmcMaxGaussians $\leq 200K$
더 빠른 학습	T22 trainingRenderScale 0.5, T1 maxIterations 절반 — 그러나 둘 다는 아님!
더 나은 하이라이트	T21 shDegreeUpgradeIterations를 [2K, 5K, 8K] 로 (Early Front Load 없음), MCMC + 200K iter
Mac 반응적 유지	T25 throttleDelayMs 5-10 (~15% 학습 시간 비용)
실시간 미리보기 더 자주	T59 livePreviewInterval 10-20으로 낮춤
그림자의 더 부드러운 전환	T17 ssimWeight 약간 높음 (0.15-0.25), 그러나 0.3 이상은 아님
실내 콤팩트하게 유지	P10 Indoor 프리셋 (mcmcMaxGaussians = 670 K, T72 = 1.76)

위험한 필드

이러한 필드는 잘못된 구성 시 OOM, 앱 충돌, Gaussian의 대량 멸종 또는 사용 불가능한 벤치마크 데이터로 이어질 수 있습니다. 주의해서 다루십시오:

- T11 densifyGradThreshold — 절반으로 줄이면 2-4배 더 많은 Gaussian이 생성되어 GPU 메모리를 빠르게 폭발시킵니다. 또한 주의: T22 trainingRenderScale과 일치해야 합니다 (1.0x \rightarrow 1e-6, 0.5x \rightarrow 2e-6, 0.25x \rightarrow 4e-6).
- T72 mcmcCapMultiplier \rightarrow 200 K SfM 초기화 점이 있는 큰 장면과 Multiplier > 5에서 수백만 Gaussian의 Resolved Cap이 생성됩니다. 36GB RAM Mac에서 OOM 가능. Outdoor 프리셋 5.32는 Mip-NeRF 360 Bicycle이 156 K init 점을 가지기 때문에만 작동합니다 \rightarrow 830 K Cap.
- T39 testViewIndices — 수동 설정은 벤치마크를 사용 불가능하게 만들 수 있습니다 (모든 인덱스 > N \rightarrow Holdout 없음). `--benchmark` 플래그가 이를 설정하도록 두십시오.
- T64 mcmcOpacityRegWeight와 T65 mcmcScaleRegWeight — 1.4.3 베타에서 0.01로 설정되어 대량 멸종으로 이어졌습니다 (한 반복에서 460 K \rightarrow 5 Gaussian). 1.4.4부터 0.0으로 고정되었지만 수동 증가는 문제를 재현할 수 있습니다.

- T15 `opacityResetInterval` — 100 000+ (효과적으로 꺼짐) 이 아니고 학습이 10 000 반복 미만이면 `Reset`이 수렴을 파괴합니다. `.preview` 가 `maxIterations = 5 000`에도 불구하고 100 000 인 이유입니다.
- T54/T55 `densifyPhase2*` — `Two-Phase Densification`은 테스트에서 0 Gaussian 캐스케이드에서 중단되었습니다. 둘 다 0으로 두십시오.
- T74 `useMipSplatting` — Q1.5 `closed-no-win 2026-05-25`, 일부 야외 장면에서는 PSNR을 악화시킬 수도 있습니다. 기본값 `off`, 연구용 `opt-in`.

이 목록에 있는 필드를 변경하려면 먼저 현재 프리셋의 백업 (JSON으로 내보내기) 을 만들고 결과를 재현 가능하게 측정할 수 있는지 고려하십시오 — 그렇지 않으면 나중에 개선을 가져왔는지 악화시켰는지 알 수 없습니다.

②

제 7 장 — 내장 품질 프리셋

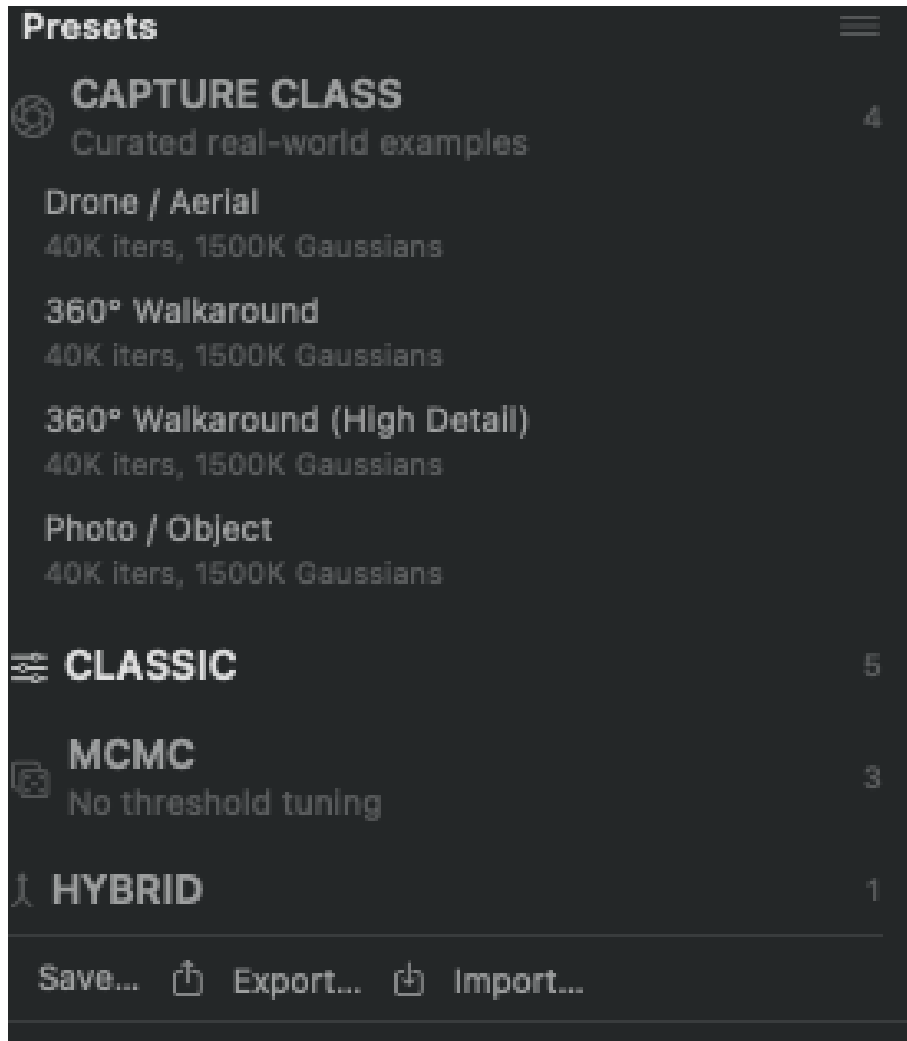


그림 27: 네 그룹이 모두 펼쳐진 Presets 섹션 — CAPTURE CLASS (4 프리셋: Drone/Aerial, 360° Walkaround, 360° Walkaround (High Detail), Photo/Object), CLASSIC (5 프리셋: Quick/Preview/Balanced/Quality/Ultra Detail), MCMC (3 프리셋, “No threshold tuning” 안내), HYBRID (1 프리셋: Balanced (Hybrid))

이미지에 보이는 것 인스펙터의 Presets 섹션, 네 그룹이 모두 펼쳐져 있습니다. CAPTURE CLASS에는 네 개의 큐레이션된 실제 촬영용 프리셋 (Drone / Aerial, 360° Walkaround, 360° Walkaround (High Detail), Photo / Object) 이 있습니다 — 이것이 기본 그룹이며 Simple Mode에서는 유일하게 보이는 그룹입니다. CLASSIC에는 Quick (1K iters), Preview (5K iters, 파란색 체크 마크와 함께 활성 선택), Balanced (20K iters), Quality (35K iters), Ultra Detail (35K iters)이 있습니다. MCMC에는 부제목 “No threshold tuning”이 붙어 있습니다 — MCMC는 Densify-Until 임계

값이 필요하지 않습니다: Preview (60K iters, 100K 가우시안), Balanced (120K, 150K), Quality (200K, 150K). HYBRID에는 Balanced (Hybrid) (20K iters, 150K 가우시안) 가 있습니다. 푸터 동작 행: Save..., Export..., Import...

프리셋은 학습을 위한 사전 준비된 구성입니다. RadianceKit은 네 그룹에 걸쳐 열세 개의 내장 프리셋을 제공합니다: 네 개의 **Capture-Class** 프리셋 (P9-P12) — 실제 커뮤니티 자료에 대해 육안으로 검증한, 실제 촬영 방식 (드론, 360° 산책, 사진 객체) 을 위한 큐레이션된 레시피이며 v1.6 이후의 기본 축 — 다섯 개의 Classic 프리셋 (P1-P5: Quick/Preview/Balanced/Quality/Ultra Detail), 세 개의 MCMC 프리셋 (P6-P8), 그리고 Classic과 MCMC 전략을 결합한 한 개의 하이브리드 프리셋 (P13). 이전의 "Scene-Class" 프리셋 (Render/3D, Outdoor, Indoor, Phase Q7에서 Mip-NeRF 360 및 NeRF Blender 장면에 대해 학술적으로 튜닝됨) 은 v1.6에서 보이는 그룹으로는 철회되었습니다 — 실제 자료에 대해 육안으로 검증된 Capture-Class가 이제 기본 축이며, Q7 튜닝된 구성은 내부적으로만 유지됩니다. 사이드바의 **Presets** 영역에서 또는 Simple Mode의 가져오기 시 프리셋을 선택합니다. + 버튼은 자체 프리셋을 옆에 만드는 대화창을 엽니다 — 열세 개의 내장은 삭제할 수 없지만 복제할 수 있습니다.

Expert View에서는 프리셋이 촬영 방식과 전략별로 (Capture Class / Classic / MCMC / 하이브리드) 그룹화되어 표시됩니다. 항목 클릭은 저장된 학습 구성을 현재 상태에 작성합니다. 이는 스냅샷이 아닙니다 — 그 후 슬라이더를 조정하면 상태가 변경되지만 프리셋 자체는 변경되지 않습니다. 그러면 색깔 있는 힌트가 "modified"를 표시합니다.

어느 프리셋이 언제 적절한지는 주로 장면 유형과 하드웨어에 달려 있습니다. 장 끝의 세 표 형식 개요가 이를 요약합니다.

P1 — Quick



위치

인스펙터 → Presets 섹션 → "Classic" 그룹 → "Quick" 항목. UUID 접미사 ...001 .



기술적 설명

1 000 반복, 고전적 (적응) Densification 전략, 0.25배 학습 해상도 스케일링 (입력 이미지가 학습 전에 25%로 축소됨) 이 있는 진단 프리셋. 장면을 제공하는 것이 아니라 설정 (카메라 포즈, 점 구름, 이미지 시리즈) 이 Loss 값에서 의미 있는 움직임을 보여 주는지 빠르게 확인해야 합니다. M3 Ultra에서 일반적으로 50-200 이미지에서 30초 미만. 작은 해상도는 실제 이미지 품질을 가리지만 메모리 요구량과 렌더 비용을 매우 낮게 유지합니다. 시스템에 10 GB RAM 미만이 있는 경우 첫 시각 시 자동으로 기본값으로 선택됩니다.

간단히 말하면

?? ? ? ? ? . ? ? ? ? ,
30 ? ? ? ? , ? ? ? ?
?? ? ? ? ? ? ? . Viewer ?
? ? ? ? ? ? ? ? ? ?
?? — ? ? ? ? , ? ? ? ?
?. ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? (? 9 ? ? ?).
? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? P2 ? P3 ? ? ? ? ? ? .

P4 — Quality (Classic)



인스펙터 → Presets 섹션 → "Classic" 그룹 → "Quality" 항목. UUID 접미사 ...003 .



V546 "Opacity Decay" (HTGS, Eurographics 2025) 가 있는 35 000 반복 Classic Densification: 각 Density Cycle 후 모든 기존 Gaussian의 Opacity가 < 1.0 인수로 곱해집니다. 이는 비활성화된 Gaussian을 Pruning에서 안정적으로 제거하고 따라서 동일한 Iter 수에서 고전적 35 000 실행보다 14% 더 나은 L1 Loss를 달성합니다. SSIM Loss 가 활성화됨 (ssimWeight=0.05). M3 Ultra에서 일반적으로 200 이미지에서 2-4분. NeRF Blender (Lego, Chair, Drums) 에서 최종 L1 ≈ 0.023 제공 — 560+ 문서화된 실험에서 최고의 Classic 변형. 참고: ~3-5 GB GPU 메모리 필요. 8 GB 시스템에서는 P3가 안전한 선택입니다.

간단히 말하면

Quality Classic Preset (P4) details and performance notes.

P5 — Ultra Detail (Classic)



인스펙터 → Presets 섹션 → "Classic" 그룹 → "Ultra Detail" 항목. UUID 접미사 ...008 .



약 35 000 반복 Classic Densification — Quality 매트릭스의 Held-out 실행 (2026-06-10) 우승자. 테스트된 세 Mip-NeRF 360 장면 모두에서 Ultra Detail은 비슷한 Wall-Clock 시간에 내장 MCMC "Quality" 프리셋 (P8) 을 약 +0.94 dB PSNR 차이로 능가합니다. 따라서 Classic 그룹에서 가장 강력한 Quality 프리셋이자 RadianceKit이 제공하는 가장 선명한 Classic 변형입니다. M3 Ultra에서 일반적으로 P4 Quality와 같은 시간 범위 (200 이미지에서 2-5분) 이지만 GPU 메모리가 약간 더 필요합니다. 8 GB 시스템에서는 P3가 안전한 선택으로 남습니다.

간단히 말하면

Ultra Detail Classic Preset (P5) details and performance notes.

P6 — Preview (MCMC)



인스펙터 → Presets 섹션 → “MCMC” 그룹 → “Preview” 항목. UUID 접미사 ...006 .



100 000 Gaussian Cap에서 60 000 반복 MCMC Densification (3DGS-MCMC, NeurIPS 2024). MCMC 는 휴리스틱 Clone/Split 로직을 Markov Chain Monte Carlo Relocation으로 대체합니다: 작은 Gaussian이 Sigmoid 가중 샘플링 깊이를 통해 새로 배치되어 제어되고 재현 가능한 Gaussian 수를 산출합니다. Cap은 최대 수를 100K에서 하드 제한합니다 — 이는 메모리와 렌더 시간을 절약하지만 세부 정보가 듭니다. M3 Ultra에서 일반적으로 200 이미지에서 5–8분. “MCMC 기능 테스트”로 적합 — P7 또는 P8에 더 많은 시간을 투자하기 전에 Classic에서 MCMC로 전환이 의미가 있는지 평가하는 데 도움이 됩니다.

간단히 말하면

P2 Preview [?] [?] [?] [?] [?] [?] [?]
MCMC [?] [?] [?]. [?] [?] Classic [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] Splat [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] 5–8[?]
[?]. [?]
[?]
[?]
P7 (Balanced) [?] [?] P8 (Quality MCMC) [?] [?] [?].

P7 — Balanced (MCMC)



인스펙터 → Presets 섹션 → “MCMC” 그룹 → “Balanced” 항목. UUID 접미사 ...007 .



150 000 Gaussian Cap에서 120 000 반복 MCMC. 중간 MCMC 단계 — P8 Quality의 거의 최종 Gaussian 수이지만 반복은 60%만. 경험적으로 L1 Loss는 문서화된 학습 세션에서 Horse Full에서 0.026–0.028, P8의 0.0246에 비해 — 즉 약 7% 더 높지만 대기 시간은 절반. M3 Ultra에서 일반적으로 200 이미지에서 8–15분. 입력 SfM 점 구름의 점 밀도에 효과적인 Gaussian Cap을 스케일링하는 절차를 사용합니다 (제 6 장 T75 참고).

간단히 말하면

[?]
MCMC, [?] [?] [?] P8 [?]
[?]
[?]. [?]
[?]
[?] MCMC [?]
[?]. [?]
GB [?] [?] [?] M [?]
[?] [?] [?] [?] [?] — P8 [?] [?] [?] [?] GPU [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?].

| P8 — Quality (MCMC)

위치

인스펙터 → Presets 섹션 → “MCMC” 그룹 → “Quality” 항목. UUID 접미사 ...004 .

기술적 설명

150 000 Gaussian Cap에서 200 000 반복 MCMC, SSIM Loss 0.05, 반복의 80% 이상에 걸친 MCMC Noise Decay. 560+ 실험에서 Best Single Run L1: Horse Full 에서 0.0238, 3 trial 평균 0.0246 (같은 장면의 P4 0.0230 대비). MCMC는 71% 적은 Gaussian (150K vs ~524K) 을 제공합니다 — 더 작은 클라우드가 상당히 작은 내보내기 파 일을 생성하므로 결과를 웹에 제공하려는 경우 결정적입니다. M3 Ultra에서 학습 시간은 일반적으로 200 이미지에서 20–35분. 1 000+ 이미지 세트에서는 1–2 시간. 최소 종료 크기에 서 최대 이미지 품질이 원하는 경우 최선의 선택.

간단히 말하면

?? MCMC ?. ?? ?
?? ???? Splat ????
???? — ??? ????
3D Viewer? ???? ?
? ???? ? ????
(??? ???? ? ????
P4 Quality?? ? ?). ???
??? ???? — ? ??
?? ? ???? ? ? ??. ?
?? “?? ??”?? ????
??.

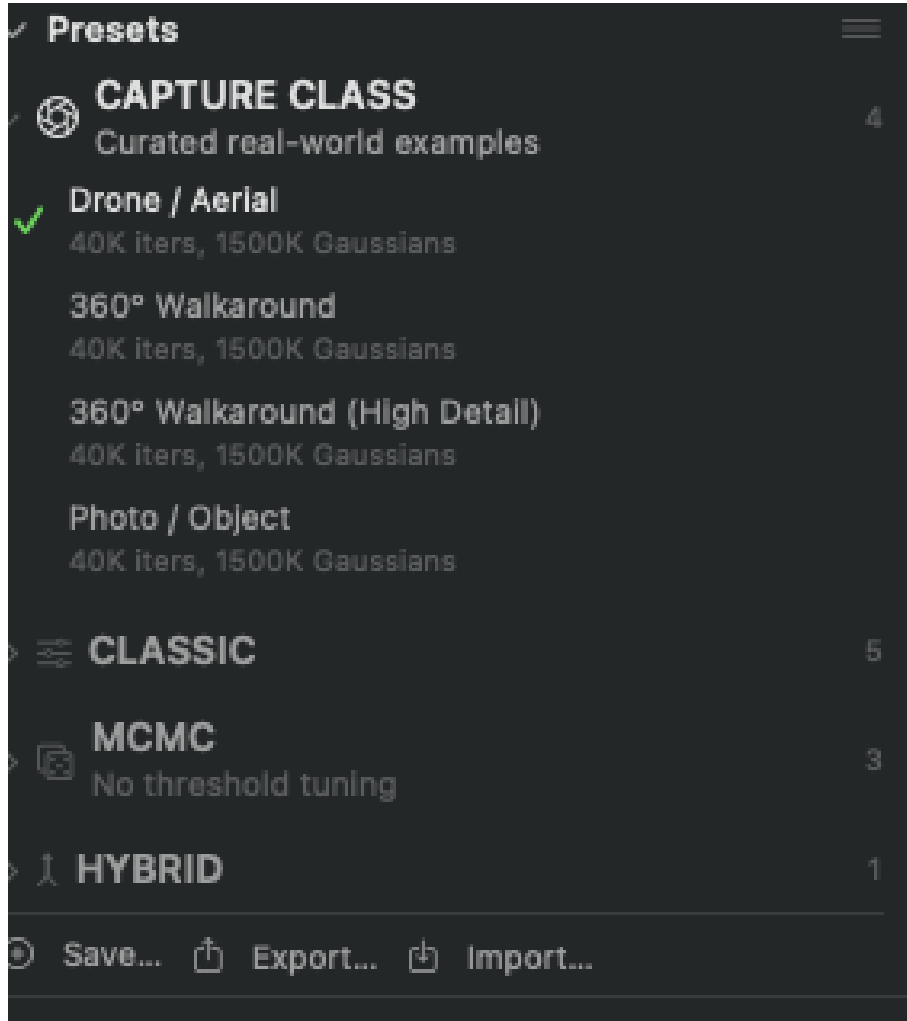


그림 28: 네 개의 큐레이션된 실제 촬영용 프리셋이 모두 펼쳐진 CAPTURE CLASS 그룹 — Drone / Aerial (MCMC, 40K iters, Cap 1.5 M), 360° Walkaround (MCMC, 40K, Cap 1.5 M), 360° Walkaround (High Detail) (Hybrid, 40K, Cap 1.5 M, opt-in) 및 Photo / Object (Hybrid, 40K, Cap 1.5 M). 이 그룹은 맨 위에 있으며 Simple Mode에서는 유일하게 보이는 그룹입니다.

이미지에 보이는 것 CAPTURE CLASS 그룹이 펼쳐진 인스펙터 — v1.6 이후의 기본 프리셋 그룹이며 Simple Mode에서는 유일하게 표시되는 그룹입니다. 각 항목은 실제 커뮤니티 자료에 대해 육안으로 검증한, 구체적인 촬영 방식 (드론, 360° 산책, 사진 객체) 을 위한 레시피이며, 학술적 테스트 세트에 대해 최적화된 값이 아닙니다. 클릭으로 선택하면 저장된 학습 구성이 현재 상태에 작성됩니다.

| P9 — Drone / Aerial



인스펙터 → Presets 섹션 → "Capture Class" 그룹 → "Drone / Aerial" 항목. UUID 접미사 ...010 .



건물과 풍경의 항공 및 드론 궤도용 Capture-Class 프리셋. MCMC Densifier, 40 000 반복, Cap 150만 Gaussian, SSIM Loss 0.5 더하기 Edge-Aware 항 0.1. 결정적인 것은 비율 임계값 6에서 가중치 0.003의 이방성 페널티입니다 — 드론 자료가 생성하는 전형적인 바늘 모양 아티팩트에 맞서는 "스파게티 킬러"입니다. Pensford 고가교 위를 비행한 실제 DJI 4K 드론 비행으로 검증됨 (메트릭뿐 아니라 육안으로 점검).

간단히 말하면

?? ? ? ? ? ? ? ? ? — ? ? ? ? ? ?
?? ? ? ? , ? ? ? ? , ? ? ? ? ?
?. ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? . ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? Photo / Object ? Classic
? ? ? ? ? ? ? ? ? ? ? ? .

| P10 — 360° Walkaround



인스펙터 → Presets 섹션 → "Capture Class" 그룹 → "360° Walkaround" 항목. UUID 접미사 ...011 .



360° 산책 영상용 Capture-Class 프리셋. MCMC Densifier, 40 000 반복, Cap 150만 Gaussian, SSIM Loss 0.5 더하기 Edge-Aware 항 0.1, 부드러운 이방성 페널티 (비율 임계값 15에서 가중치 0.001). 인물 및 하늘 마스크가 활성화. 이 프리셋은 360° Equirect 영상을 기대하며, 학습이 시작되기 전에 내부적으로 약 90° 폭의 원근 Crop으로 다시 투영됩니다. 셀카봉을 이용한 8K 360° 산책으로 검증됨 (Monument 장면, 육안으로 점검).

간단히 말하면

360° ? ? ? ? ? — 360° ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? .
RadianceKit ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? . ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? High Detail ? ? (P11) ? ? ?
? ? ? ? ? ? ? ? ? ? ? .

P11 — 360° Walkaround (High Detail)



위치

인스펙터 → Presets 섹션 → "Capture Class" 그룹 → "360° Walkaround (High Detail)" 항목. UUID 접미사 ...013 (Opt-in).



기술적 설명

최대 디테일을 위한 360° 산책 영상용 Opt-in Capture-Class 프리셋. 하이브리드 Densifier (고전적 Abs-Gradient Clone/Split + MCMC Noise

1. Relocation), 40 000 반복, Cap 150만 Gaussian, 비율 임계값 15에서

이방성 페널티 0.0015, SSIM Loss 0.2, Edge-Aware 항 0 — 잠긴 "r50" Screen-Split 레시피. 360° 자료에서 표준 MCMC 프리셋 "360° Walkaround" (P10) 을 PSNR, LPIPS, 가시적 컨페티에서 능가하며, 그것도 약 3분의 1의 Splat 수로. 더 많은 장면에서 검증될 때까지 의도적으로 표준 360 프리셋 옆에 opt-in으로 놓여 있습니다.

간단히 말하면

360 (P10)
 : , ,
 , .
 —
 . 360°
 .
 P10 .

P12 — Photo / Object



위치

인스펙터 → Presets 섹션 → "Capture Class" 그룹 → "Photo / Object" 항목. UUID 접미사 ...012 .



기술적 설명

선명한 개별 사진 (영상 아님) 의 객체 궤도용 Capture-Class 프리셋. 하이브리드-t1 Densifier (Relocation 포함), 40 000 반복, Cap 150만 Gaussian, SSIM Loss 0.5 더 하기 Edge-Aware 항 0.1, 부드러운 이방성 페널티 (비율 임계값 15에서 가중치 0.001), 50 반복마다 Opacity Decay 0.9995, 마스크 없음. 해골의 163장 고해상도 41 MP 사진으로 검증됨 (육안으로 점검). 적은 뷰 (약 600까지) 는 하이브리드 Collapse 임계값 아래에 머뭙니다.

간단히 말하면

— , ,
 .
 . 360°
 Drone
 .

P13 — (??) (?????)



위치

인스펙터 → Presets 섹션 → “하이브리드” 그룹 → “균형 (하이브리드)” 항목. UUID 접미사 ...009 .



기술적 설명

하이브리드 Densification 전략의 20 000 반복: 고전적 그라디언트 기반 Clone/Split이 Loss가 필요로 하는 곳에 용량을 배치하고, MCMC SGLD 노이즈가 계속 탐색하며, 죽은 Gaussian은 Pruning으로 손실되는 대신 재배치됩니다. Opacity Decay (V546) 가 Opacity 리셋을 대체합니다; 이 방성 페널티 (가중치 0.001, 비율 임계값 15) 가 바늘 모양 Splat을 억제합니다. Gaussian Cap은 장면에 따라 스케일됩니다 (150K 기본, scene-aware x3.0). 같은 예산의 순수 MCMC에 대해 5개 장면에서 검증됨: 평균 +0.45 dB PSNR, 20–30 % 적은 Gaussian (stonehenge +1.23, family +0.82, garden +0.47 dB). M3 Ultra에서 일반적으로 200 이미지에서 5–10분.

간단히 말하면

????? ???? ???? ? ?
 ?]: MCMC ???? ???? ? ?
 ???? ???? ? ???? ?
 ? ???? ? P8 ? ? ? ? ?
 ???? ? ???? ? ? ? ?
 ? ???? ? ???? ? ? ? ?
 Capture ???? ? ? ? ?
 ???? ? ???? ???? ????
 ? ? ? ? ? ? ? ? ? ? Classic
 ???? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ?
 ???? ? ? ? Capture-Class ? ?
 ? (P9–P12) ? ? ? ? ? ? ? ?

언제 어느 프리셋?

시나리오	첫 테스트	본 실행
새 이미지 기능 테스트, < 30초	P1 Quick	—
선명한 개별 사진의 객체 궤도	P2 Preview	P12 Photo / Object
단일 객체 스캔 (영상), < 500 사진	P2 Preview	P4 Quality 또는 P8 Quality MCMC
360° 산책 영상	P6 Preview MCMC	P10 360° Walkaround (선명: P11 High Detail)
항공/드론 궤도, 풍경	P6 Preview MCMC	P9 Drone / Aerial
웹 배포 (작고 콤팩트)	P2	P8 Quality MCMC (전체 품질에서 가장 작은 파일)
짧은 시간에 선명한 세부 정보, 콤팩트한 내보내기	P2 또는 P6	P13 균형 (하이브리드)
최대 디테일, Classic 전략	P3 또는 P6	P5 Ultra Detail
인쇄, 마케팅, 전체 세부 정보	P3 또는 P6	P4 Quality (Classic) 또는 P5 Ultra Detail

빠른 비교

프리셋	전략	Iters	Max-Gs	렌더 스케일	일반적인 시간 (200 이미지, M3 Ultra)	Q Sweep
P1 Quick	Classic	1 000	∞	0.25x	~30 s	—
P2 Preview	Classic	5 000	∞	0.5x	2–3 min	—
P3 Balanced	Classic	20 000	∞	1.0x	30–60 s	—
P4 Quality	Classic	35 000	∞	1.0x	2–4 min	V546 HTGS
P5 Ultra Detail	Classic	~35 000	∞	1.0x	2–5 min	Matrix Δ+0.94 dB
P6 Preview MCMC	MCMC	60 000	100 K	1.0x	5–8 min	—
P7 Balanced MCMC	MCMC	120 000	150 K	1.0x	8–15 min	—
P8 Quality MCMC	MCMC	200 000	150 K	1.0x	20–35 min	V544a
P9 Drone / Aerial	MCMC	40 000	1.5 M	1.0x	10–25 min	육안 / Viadukt
P10 360° Walkaround	MCMC	40 000	1.5 M	1.0x	10–25 min	육안 / Monument
P11 360° Walkaround (High Detail)	하이브리드	40 000	1.5 M	1.0x	10–25 min	육안 (opt-in)
P12 Photo / Object	하이브리드	40 000	1.5 M	1.0x	10–25 min	육안 / 해골
P13 균형 (하이브리드)	하이브리드	20 000	150 K	1.0x	5–10 min	Matrix Δ+0.45 dB

자체 프리셋

Presets 섹션의 **Save...** 버튼 (제 2 장 11) 을 통해 현재 학습 구성을 자체 프리셋으로 저장합니다. 자체 프리셋은 "Built-in"이 아니며 이름을 바꾸고, 내보내고 (JSON으로), 드래그 앤 드롭으로 공유하고, 복제하고, 삭제할 수 있습니다. 열세 개의 내장 프리셋 P1–P13은 삭제 버튼의 영향을 받지 않습니다.

경험 법칙: 더 자주 필요할 프리셋의 무언가를 변경하는 경우 — Sky-Dome 켜짐, 특정 장면 클래스를 위한 더 높은 SSIM 가중치, 다른 Iter 수 — 변형을 자체 프리셋으로 저장하십시오. 그러면 다음 실행에서 표준과 다른 구성임을 즉시 알 수 있습니다.

2

제 8 장 — 내보내기 형식

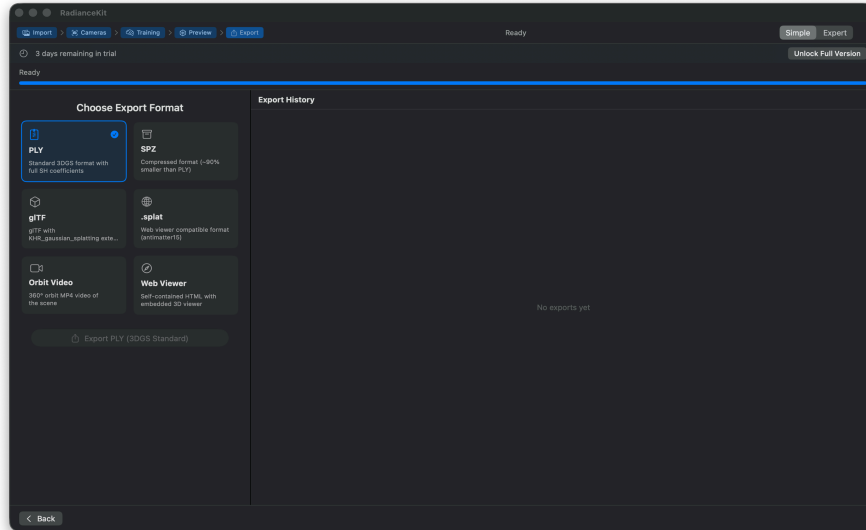


그림 29: Simple Mode의 내보내기 형식 선택 — 6개의 형식 카드

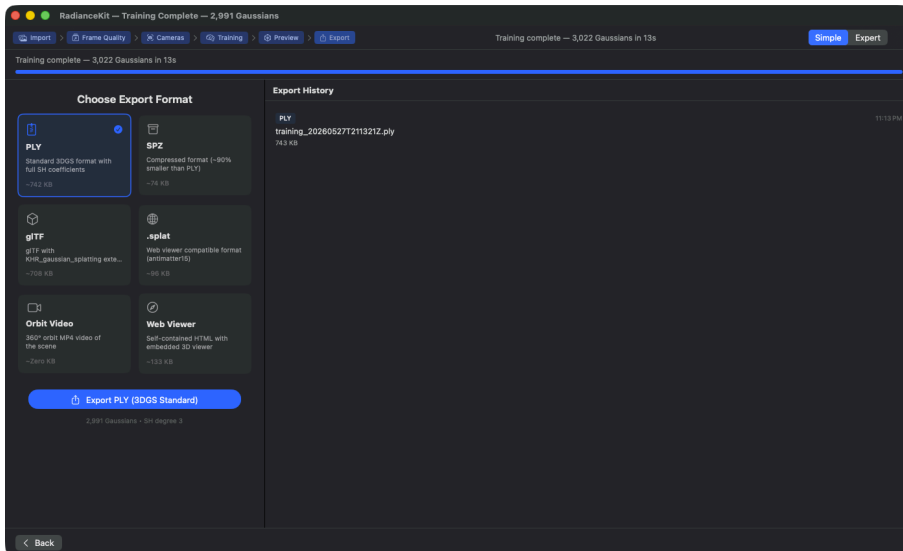


그림 30: flowers-Bouquet에서 5K-Iter 학습 후 실시간 내보내기 형식 그리드 — 동적 크기 계산이 있는 모든 6개 카드 (PLY 742 KB 선택, SPZ 74 KB, glTF 708 KB, .splat 96 KB, Orbit Video ~Zero KB, Web Viewer 133 KB), 오른쪽에 이미 저장된 PLY가 있는 Export History

이미지에 보이는 것 (IP-clean 테스트 세트르 Bjoern의 합성 Blender Bouquet의 2 991 Gaussian, SH degree 3): 각 형식 카드 아래의 크기 정보는 현재 Gaussian Count와 형식 오버헤드에서 실시간으로 계산됩니다 — 하드코딩되지 않습니다. 2 991 Gaussian (SH degree 3) 에서 742

KB PLY, 74 KB SPZ (양자화를 통해 ~10x 작음), 708 KB glTF (KHR_gaussian_splatting 확장
과 함께 거의 PLY 동등), 96 KB .splat (압축된 Gaussian당 24 바이트 형식) 가 생성됩니다. Orbit
Video는 “~Zero KB”를 표시합니다. 크기는 MP4 인코딩 후에야 알려지기 때문입니다. Web Viewer
(133 KB) 는 내장된 WebGL Viewer와 압축된 Splat 데이터가 있는 독립형 HTML 파일을 묶습니
다 — Viewer 오버헤드 때문에 순수 .splat보다 큼니다. 오른쪽 Export History는 형식 편과 Reveal
in Finder 동작이 있는 이미 완료된 PLY 내보내기 (“training_20260527T211321Z.ply, 743 KB,
23:13”) 를 나열합니다.

완료된 학습은 Gaussian Cloud를 제공합니다 — 함께 장면을 재구성하는 수십만에서 수백만의 3D
Gaussian 분포의 모음. RadianceKit은 이 클라우드를 디스크에 쓰는 10가지 방법을 알고 있습니다. 그
중 6개는 순수 3D 데이터 형식 (PLY, Compressed PLY, SPZ, SOG, glTF, .splat) 이고, 1개는 클
라우드를 완성된 HTML Viewer와 함께 묶고 (Web Viewer), 1개는 Orbit 카메라 이동에서 MP4 파
일을 렌더링하고 (Orbit Video), 2개는 Gaussian 콘텐츠를 내보내지 않고 단지 다른 학습 파이프라인
에서 재사용할 SfM 결과 (카메라 포즈와 거친 점 구름) 만 내보냅니다 (transforms.json + COLMAP
Workspace).

어느 형식이 언제 적절한지는 목표에 따라 다릅니다. 품질 손실 없이 전체 데이터를 보관하려면 PLY를 사용
합니다. 자체 사이트의 Web Viewer에는 보통 .splat이나 내장 Web Viewer로 충분합니다. 파일이 최소
여야 한다면 SPZ나 SOG가 가치가 있습니다. Nerfstudio, Postshot 또는 Brush에서 SfM 결과를 재
사용하려면 transforms.json과 COLMAP Workspace가 올바른 경로입니다.

모든 내보내기 기능은 “Export” 메뉴와 Simple Mode의 마지막 마법사 단계에 있습니다. 대부분의 형식
은 완전히 Sandbox 호환이며 App Store 버전에서 작동합니다. SOG만이 외부 바이너리 (cwebp) 가
필요하며 App Store 빌드에 반드시 존재하지는 않습니다 — 자세한 내용은 E4 참고.

E1 — PLY (.ply)

📍 위치

메뉴 바 → Export → 3D Formats → Export PLY... (☞E).
 Simple Mode: 마법사 단계 Export → 형식 카드 "PLY".
 크기: 일반적으로 100% (참조 값). 호환: SuperSplat,
 PolyCam, 모든 3DGS Viewer.

🛠 기술적 설명

PLY는 3D Gaussian Splatting의 표준 저장 형식입니다. RadianceKit은 표준화된 3DGS 속성 레이아웃이 있는 바이너리 Little-Endian 파일을 작성합니다: Gaussian당 3 성분 위치, 항상 0으로 설정된 3 노멀, 기본 RGB 색상용 3 DC SH 계수 (`f_dc_0..2`), 그 후 Kerbl 2023 논문에 정의된 transposed Channel-Major 배열 (먼저 모든 R 채널 계수, 그 다음 모든 G, 그 다음 모든 B) 의 최대 45개 추가 SH 계수 (`f_rest_0..44`), 그 다음 Logit Opacity (원본 사전 Sigmoid 값), 3개의 Log Space 스케일, 하나의 wxyz 쿼터니언 회전. 최대 내보낸 SH 정도는 사용자 희망과 실제로 학습된 정도의 최솟값으로 클램프됩니다. 기본값은 3 (45개의 Rest 계수). 쓰기 전에 Payload 크기가 64비트 정수로 계산되어 극도로 큰 Cloud에서 오버플로를 잡습니다. 파일은 원자적으로 작성되며, 큰 Cloud에서는 잠시 두 배의 디스크 공간을 점유합니다.

💬 간단히 말하면

"? ? ? ?" ? ? ? . ? ? ? ?
 ? , ? ? ? ? ? ? ? , ? ? ? ? .
 ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? PLY ? ? ? ? ? ? —
 ? ? ? ? 3DGS ? ? ? ? ? ?
 ? ? . 100 ? Gaussian ? ? ? SH
 ? ? ? ? ? ? 200 ? ? 800 MB. ?
 ? ? ? ? ? ? ? ? E2 (? ? PLY)
 ? E3 (SPZ) ? ? ? ? ? ? ? ? .

E2 — Compressed PLY (.ply)

위치

메뉴 바 → Export → 3D Formats → Export Compressed PLY... Simple Mode: 형식 카드 "Compressed PLY". 크기: PLY 대비 약 10–20% (5 10 배 압축). 호환: SuperSplat, PlayCanvas Engine, 웹 기반 Viewer.

기술적 설명

Chunked 양자화가 있는 PLY 형식의 PlayCanvas 변형. Gaussian이 256개 청크로 그룹화됩니다. 청크당 Position, Scale, Color의 Min/Max Bound가 헤더에 별도로 저장됩니다. 개별 Gaussian은 이러한 Bound에 대해 상대적으로 값을 참조하고 각각 32비트로 압축됩니다: Position과 Scale은 11-10-11 비트 패킹으로, Rotation은 2-10-10-10 비트 "Smallest-Three" 쿼터니언으로, Color는 8-8-8-8 RGBA로. 더 높은 SH 계수는 구성 요소당 8비트로만 양자화됩니다 (Gaussian당 `shCoeffCount * 3 uchar`). 형식 자체는 여전히 ASCII Header PLY이며 따라서 기본적으로 PLY 도구로 검증 가능하지만 Vertex Properties는 `uint` 필드로 선언됩니다. 압축을 최대화하기 위해 SH 정도는 기본적으로 0입니다 (Rest 계수 없음) — 더 높은 SH 정도는 명시적으로 선택할 수 있습니다.

간단히 말하면

```

ply
  header
    element vertex 510
    property float position
    property float rotation
    property float color
  end_header
  vertex
    position 1.0 1.0 1.0
    rotation 0.0 0.0 0.0
    color 0.0 0.0 0.0
  end
end
    
```


E4 — SOG (.sog)



위치

메뉴 바 → Export → 3D Formats → Export SOG...
Simple Mode: 형식 카드 "SOG". 크기: PLY 대비 약 5 6% (15 20배 압축 — 가장 작은 옵션). 호환: PlayCanvas Engine, SuperSplat Editor.



기술적 설명

"Spatially Ordered Gaussians" — 여러 무손실 WebP 이미지에 Cloud를 GPU 준비 상태로 저장하는 PlayCanvas 형식. 먼저 모든 Gaussian이 3D Morton Code (30비트 Z Order, 축당 10비트) 로 공간적으로 정렬되어 이미지가 렌더러에서 나중에 Cache Locality를 가지게 합니다. 그 다음 Position이 대칭 Log 변환 (더 나은 동적 범위를 위해) 으로 16비트 값으로 양자화되어 두 RGBA 이미지로 분할됩니다 (means_l.webp 는 하위 8비트, means_u.webp 는 상위). Rotation은 RGBA 이미지의 2비트 Mode

1. 3x8비트의 Smallest-Three로 인코딩됩니다 (Mode는 252 + largest 로

Alpha에 들어감). Scale과 DC SH는 각각 256 항목 코드북으로 양자화됩니다 (모든 값에 대해 백분위 기반 분포). 인덱스는 scales.webp 와 sh0.webp 에 저장됩니다. 다섯 이미지와 Codebook 및 Bound가 있는 meta.json 은 ZIP 파일로 패키징되고 (사용자 정의 인코더, Sandbox가 시스템 zip 을 차단하기 때문) .sog 확장자로 저장됩니다.

Sandbox 주의: SOG는 외부 바이너리가 필요한 유일한 형식 옵션입니다. WebP 인코더 단계는 /usr/local/bin/cwebp 또는 /opt/homebrew/bin/cwebp에서 cwebp 를 호출합니다. cwebp 바이너리가 발견되지 않으면 코드는 원시 PNG 인코딩으로 대체됩니다 — 그러나: **PNG 대체는 SuperSplat에서 작동하지 않습니다.** App Store 버전에서는 빌드 변형에 따라 가용성을 평가하십시오. Developer 변형에서는 cwebp가 Homebrew (brew install webp) 를 통해 설치되어 있어야 합니다.

간단히 말하면

```

?? ?? 3DGS ??? SPZ
?? ????. ????:
RadianceKit?? ????
???? ? ???? Mac
? cwebp ????.
Homebrew?? ????
?? ?????. App Store
????? ???? ?
?? ???? — ????
WebP ?? PNG? ????
SuperSplat??? ????
??. Homebrew ?? ????
??? ? SPZ (E3) ? ?
??.
```

E5 — glTF (.glb)

📍 위치

메뉴 바 → Export → 3D Formats → Export glTF...
 Simple Mode: 형식 카드 "glTF". 크기: PLY와 비슷. 호환:
 KHR_gaussian_splatting 확장 (Khronos Draft 표준) 이
 있는 glTF Viewer.

🔧 기술적 설명

KHR_gaussian_splatting 확장 사양에 따라 자체 포함된
 .glb 바이너리 파일 (별도 Bin 파일 첨부 없음) 을 작성
 합니다. Position은 일반 glTF POSITION Vertex 데이터
 (float3) 로 저장되고, 다른 모든 속성 (Rotation은 float4,
 Scale은 float3, Opacity는 float, SH 계수는 float3 ×
 shCoeffCount) 은 추가 Vertex 속성에 있고 확장을 통
 해 참조됩니다. 중요: glTF는 오른손 Y up 좌표계를 사용하
 고, COLMAP/3DGS는 Y down/Z forward로 작업합니다.
 따라서 내보내기는 X축 주위로 180도 회전을 적용합니다 —
 Position은 (x, -y, -z) 로 다시 작성되고, Quaternion
 은 (w, x, -y, -z) 로 조정됩니다. 이는 glTF Viewer에서
 기하학적으로 올바르게 핸드 (거울이 아닌) 표현을 산출합니다.
 GLB 표준이 요구하는 대로 JSON과 바이너리 청크는 4바이트
 정렬로 패드됩니다.

💬 간단히 말하면

3D [?] [?] [?] [?] Khronos [?] [?]
 [?] [?], Gaussian Splat [?] [?] [?]
 [?] [?] [?] [?]. [?]: glTF [?]
 [?] [?] 3D [?] (Babylon.js,
 Three.js, Unity, Unreal) [?]
 [?] [?] [?] [?]. [?]: [?] [?]
 2026 [?] [?] [?] Draft [?] [?]
 [?] [?] Viewer [?] [?] [?] [?]
 [?] [?] [?]. [?] Splat [?] [?]
 [?] glTF [?] [?] [?] [?]
 [?] [?] glTF [?] [?] Viewer [?]
 [?] [?] [?] [?] [?] [?] [?].

E6 — Splat (.splat)

📍 위치

메뉴 바 → Export → 3D Formats → Export .splat...
 Simple Mode: 형식 카드 ".splat". 크기: Gaussian 당 정확히 32 바이트. 호환: gsplat.js, 웹 기반 Viewer (antimatter15 참조, 대부분의 브라우저 3DGS 데모.

🔧 기술적 설명

antimatter15 .splat 형식 — Gaussian당 32 바이트, 헤더 없음, 간접 참조 없음. 항목당 레이아웃: 3 × float32 Position (세계 좌표), 3 × float32 Scale (내부 버퍼의 Log Space에서 exp 변환), 4 × uint8 RGBA 색상 (SH_C0 = 0.282... 로 스케일된 DC SH 계수, [0,255] 에 클램프, 4 × uint8 쿼터니언 (w,x,y,z, 정규화되고 바이트 범위에 + 128*q 로 인코딩). DC SH만 저장 — 더 높은 SH 밴드는 폐기됩니다. 이는 형식을 극도로 콤팩트하게 만들지만 반사나 Specular Highlight에서 발생하는 View 의존 색상 변경 비용이 듭니다. 쓰기 순서는 정확히 Cloud의 인덱스 순서입니다 (공간 정렬 없음), gsplat.js 같은 웹 Viewer가 이를 기반으로 렌더링합니다.

💬 간단히 말하면

```
gsplat.js Viewer
Splat Splat
(Gaussian 32) SH —
View View
```

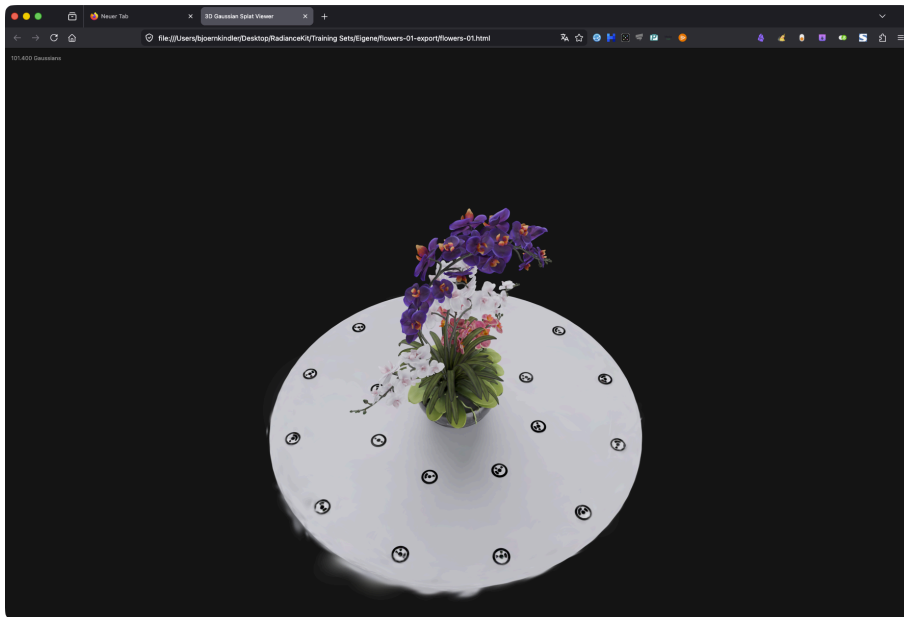


그림 31: Firefox에서 열린 Web Viewer — 주변 카메라 마커 구체와 함께 렌더링된 Bjoern의 Bouquet Splat, 위에 보이는 브라우저 탭 바, CDN/서버 설정 필요 없음. Finder에서 더블 클릭으로 기본 브라우저에서 직접 연 독립형 flowers-01.html — 내장된 WebGL2 프로그램이 네트워크나 서버 없이 Gaussian Cloud를 즉시 렌더링합니다. Bouquet 주위의 검은 마커는 학습 카메라이며 선택적으로 표시 가능합니다. 마우스 드래그로 회전, 스크롤로 줌.

E7 — Web Viewer (.html)

📍 위치

메뉴 바 → Export → Media → Export Web Viewer...
Simple Mode: 형식 카드 "Web Viewer". 크기: Splat 데이터 base64 인코딩 (≈ 4/3 오버헤드) + 약 5 KB HTML/JS 셀.
호환: WebGL2가 있는 모든 최신 브라우저 (모든 데스크톱, iOS 15+, Android 5+).

🔧 기술적 설명

완전히 인라인으로 작성된 WebGL2 렌더러와 함께 Gaussian Cloud를 단일 .html 파일로 묶습니다. CDN 의존성이 없고, WASM이 없고, 두 번째 파일이 없습니다. Cloud는 먼저 .splat 바이너리로 인코딩되고 (E6과 같은 32 바이트 로직), 그 다음 base64로 임베드되고, 그 다음 브라우저에서 atob 로 디코딩됩니다. 내장 렌더러는 자체 WebGL2 정렬, 마우스 Orbit 제어, 프레임당 CPU 정렬을 수행합니다. 전체 JS 코드 (셰이더, 수학, 루프) 가 출력 HTML 에서 볼 수 있습니다. 저장-렌더러 경계의 축 규약은 E5와 정확히 동일합니다: Position (x, -y, -z), Quaternion (w, x, -y, -z). 선택적으로 브랜딩 오버레이를 표시할 수 있습니다 (Free Tier 스위치). 모든 것이 인라인이기 때문에 파일은 file:// 프로토콜에서도 직접 작동합니다 — 테스트에 로컬 웹 서버 필요 없음.

💬 간단히 말하면

```
????? ???? ????
????? ???? ????
? HTML ??. Finder???
????? ???? ????
????? ???? ?????.
????? ???? ??, ?
?? ?? ??, ?? ???. ??
??????, ?????, ??
? ???? ?????. ??:
base64 ???? ???? ?
? .splat ?? ? 1/3 ???
? — ???? ???? ????
? .splat ??? Viewer
??? ???? ????
???? ????.
```

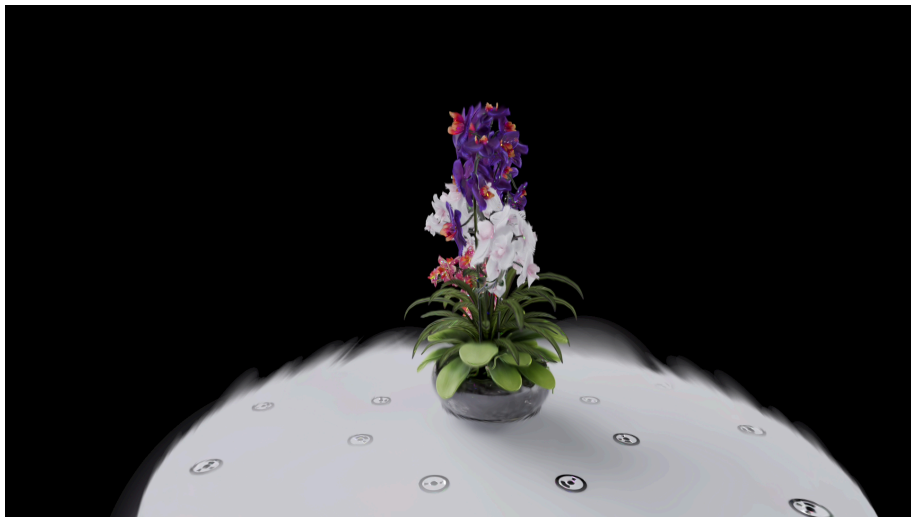


그림 32: flowers-01.mp4에서 추출한 단일 프레임 — 프로필 렌더에서 Bjoern의 Bouquet, 카메라 마커가 있는 보이는 흰색 플랫폼, 검은색 배경 (기본 뷰포트 배경, Settings에서 변경 가능). 카메라는 파라메트릭 궤도에서 장면을 둘러쌉니다 (Elevation + 거리 고정, Yaw 회전), 일반적으로 30 또는 60 fps에서 6-10초 지속 시간. 프레임 해상도는 VideoPreset을 통해 480p에서 8K까지 스케일 가능합니다.

E8 — Orbit Video (.mp4/.mov)

📍 위치

메뉴 바 → Viewport → Record Turntable Video 또는
메뉴 바 → Export → Media → Export Orbit Video....
Simple Mode: 3–30 s 지속 시간 슬라이더가 있는 형식 카
드 "Orbit Video". 크기: 지속 시간, 해상도, 비트레이트에 따
라 다름. 호환: 모든 플랫폼 (H.264과 HEVC가 Apple 표준).

🔧 기술적 설명

파라메트릭 Orbit 카메라 이동을 따라 Gaussian Cloud를
렌더링하고 AVAssetWriter를 통해 각 프레임을 MP4 또는
MOV 파일로 인코딩합니다. Orbit 구성은 회전 속도 (회전
수), 거리, Elevation, FOV, 지속 시간, Ease In/Out 인수
를 제어합니다. Orbit 비디오 내보내기는 전체 SH 평가와 함께
RadianceKit 자체의 ForwardPass를 거칩니다 — 인앱 뷰포
트와 픽셀 단위로 동일합니다 (WYSIWYG). 프레임당 World
적용 매트릭스 (내부 좌표를 Y up Orbit 세계로 회전시키기
위해 렌더러가 계산) 가 카메라와 곁해진 후 카메라 규약 뒤집
기 (camFlip: Orbit Y up → COLMAP Y down) 가 적용
됩니다. 오프스크린 렌더 타겟은 인코더용 CVPixelBuffer에
IOSurface를 통해 끌어내립니다. 인코더는 H.264와 HEVC,
구성 가능한 비트레이트 및 480p에서 8K까지의 해상도를 지
원합니다. 첫 프레임 전에 렌더러는 초기 Splat 정렬이 완료되
도록 200 ms를 기다립니다. 이 내보내기는 GPU 제한입니다
— 8K와 수백만 Gaussian에서 프레임당 렌더 시간이 몇 초이
며, 따라서 6초 비디오에 총 렌더 시간 10–30분 가능.

💬 간단히 말하면

?? ? ? ? ? ? ? ? ?
MP4 ???. ? ? ? ? ? , ? ?
? , ? ? ? ? ? ? ? ? ? ? ? ? . ? ?
? ? (3–30?), ? ? ? ? , ? ?
? ? ? ? ? ? ? ? ? ? . ? ? ?
YouTube, Instagram, PowerPoint
? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? . ? ? ? ? ? ? ?
? . ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? — 8K
? ? ? ? ? ? ? ? Gaussian ? ? ?
? 5 ? ? ? ? 30 ? ? ? ? ? ? ? ? ?
? ? ? ? .

E9 — SfM Transforms (transforms.json)

📍 위치

메뉴 바 → Export → Photogrammetry → Export SfM (transforms.json).... 크기: 일반적으로 1-10 KB (Pose + Intrinsic만, 이미지 없음, Gaussian 없음). 호환: nerfstudio, Brush, gsplat, OpenSplat, Meshroom, 모든 현대 Feed Forward 3DGS Trainer.

🔧 기술적 설명

공유 Intrinsic이 있는 카메라 포즈 목록이 있는 nerfstudio transforms.json 형식을 작성합니다. 카메라당 View 매트릭스 (RadianceKit 내부: COLMAP 규약의 World to Camera) 가 반전된 다음 카메라 로컬 Y와 Z 베이스 벡터가 미러링되어 nerfstudio 규약 (OpenGL 스타일, 카메라가 -z를 따라 보고, +Y 가 위) 으로 변환됩니다. 최종 4x4 매트릭스는 각 프레임의 transform_matrix 필드의 row-major nested array of Doubles로 저장됩니다. Intrinsic은 최상위에 저장됩니다 (초점 거리 x/y, 주점 x/y, 이미지 폭/높이, camera_model = "OPENCV", Distortion Coefficient k1, k2, p1, p2) — 내보내기가 여러 다른 Intrinsic 세트를 감지하지 않는 한, 그 경우 프레임당 작성됩니다. 이미지 경로는 JSON 파일에 상대적인 images/<filename>으로 작성됩니다. 사용자는 학습 사진이 있는 형제 images/ 폴더를 만들어야 합니다.

💬 간단히 말하면

? JSON ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? — ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ?
RadianceKit SfM ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ?.

E10 — COLMAP Workspace (sparse/0/)

📍 위치

메뉴 바 → Export → Photogrammetry → Export SfM (COLMAP Workspace).... **크기:** 세 바이너리 파일 함께 일반적으로 4~8 MB — `points3D.bin` 이 지배 (Sparse Cloud의 3D 점당 한 줄), `images.bin` 과 `cameras.bin` 은 각각 100 KB 미만. **호환:** COLMAP 자체, Nerfstudio, Postshot, Meshroom, COLMAP `sparse/` 디렉터리를 예상하는 모든 도구.

🔧 기술적 설명

세 바이너리 파일이 있는 표준 COLMAP `sparse/0/` 레이아웃을 작성합니다: `cameras.bin`, `images.bin`, `points3D.bin`. 형식 참조는 공식 COLMAP 문서입니다. `cameras.bin` 은 중복 제거된 Intrinsic 목록 (동일한 Intrinsic + 이미지 크기를 가진 카메라가 단일 항목으로 결합됨) 을 포함합니다. 사용된 Camera Model은 `OPENCV` (Model 4) 이며 `fx/fy/cx/cy` + 네 개의 Distortion Coefficient `k1/k2/p1/p2`가 있습니다. `images.bin` 은 이미지당 `wxyz` 쿼터니언으로 Pose + Translation, 그 다음 Camera ID와 파일 이름을 나열합니다. 2D-3D 대응은 저장되지 않습니다. `points3D.bin` 은 Position, Color (0-255 RGB), Reprojection과 Track Length의 기본값이 있는 SfM 점 구름을 포함합니다. 모두 Little-Endian으로 작성됩니다. RadianceKit으로의 재가져오기는 File 메뉴 → "Import COLMAP/Metashape Workspace..."를 통해 작동합니다 (SfM 백엔드 장의 Q3 참고).

💬 간단히 말하면

[?] COLMAP [?]. [?]
 Postshot, Nerfstudio [?] [?]
 COLMAP [?] [?] [?] [?]
 [?] [?] [?] [?] [?] [?].
 [?] [?] [?] [?] [?],
 [?] [?] [?] [?] [?] [?]
 COLMAP [?] [?] [?] [?]
 [?] [?] [?] [?] [?] [?]
 [?]. transforms.json [?] (E9) [?]
 [?] [?] [?] [?] [?] [?] [?]
 [?] [?], [?] [?] [?] [?]
 [?] [?] [?] [?] [?] [?] [?]
 [?] [?] [?].

어느 형식을 언제?

목표	형식
자체 사이트의 Web Viewer	E7 Web Viewer (.html)
<code>gsplat.js</code> 가 있는 Web Viewer	E6 Splat (.splat)
Postshot / Nerfstudio의 파이프라인 재사용	E9 transforms.json + E10 COLMAP Workspace
SuperSplat 편집	E1 PLY 또는 E2 Compressed PLY
Niantic Scaniverse / Spatial Fields	E3 SPZ
최대 압축	E4 SOG (cwebp 필요)
마케팅/소셜 비디오	E8 Orbit Video

빠른 비교

형식	확장자	Sandbox	크기 (1M Gauss)	Best Use
E1 PLY	.ply	예	~250 MB	아카이브, 최고 호환성
E2 Compressed PLY	.ply	예	~40 MB	Web + SuperSplat
E3 SPZ	.spz	예 (gzip-Spawn)	~40 MB	Niantic + 모바일
E4 SOG	.sog	조건부 (cwebp)	~20 MB	최대 압축
E5 glTF	.glb	예	~250 MB	Khronos 파이프라인
E6 Splat	.splat	예	~32 MB	gsplat.js Web Viewer
E7 Web Viewer	.html	예	~45 MB	독립형 브라우저 파일
E8 Orbit Video	.mp4 / .mov	예	가변	소셜/마케팅
E9 SfM Transforms	.json	예	~5 KB	Pose 전달
E10 COLMAP Workspace	디렉터리	예	~4-8 MB	Pose 전달 바이너리

크기 열은 SH 정도 3의 100만 Gaussian에 대한 대략적인 기준값입니다. 실제 값은 장면의 압축성에 따라 다릅니다. SH 정도 0은 PLY/glTF를 4배 줄입니다.

2

제 9 장 — SfM 백엔드

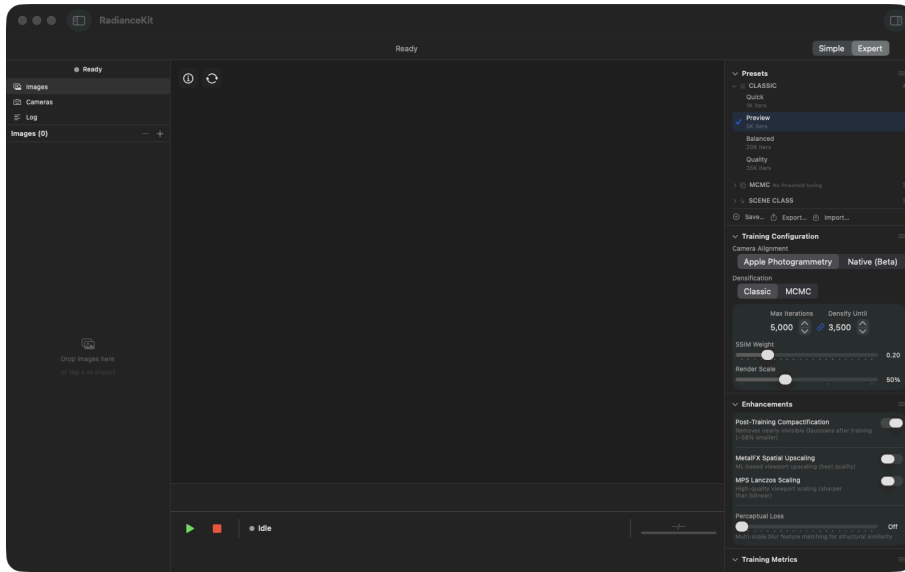


그림 33: 인스펙터의 Camera Alignment 선택기가 있는 Expert 모드 (Apple Photogrammetry / Native (Beta))

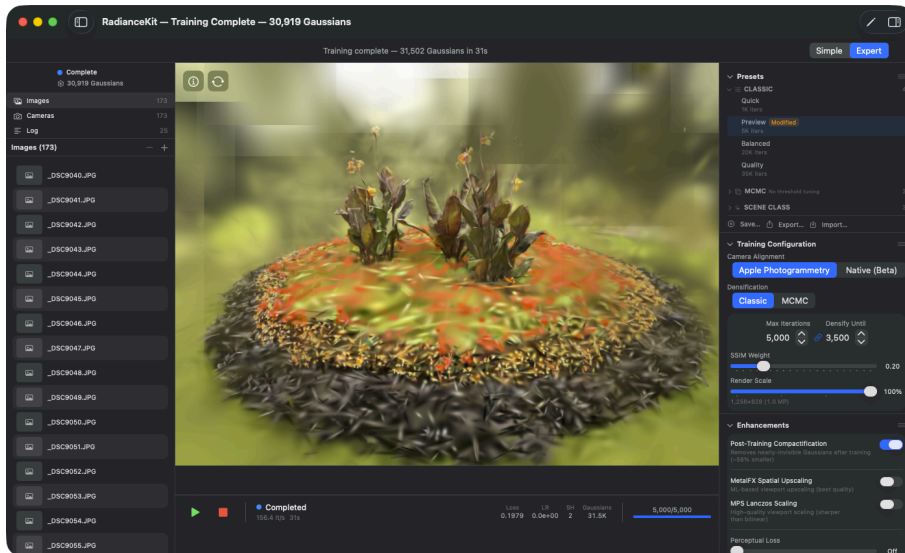


그림 34: Native (Beta) 가 활성화된 인스펙터 — Camera Alignment 선택기에서 두 번째 옵션이 선택되었고 다른 모든 학습 구성 매개변수는 변경되지 않음

이미지에 보이는 것 인스펙터의 Camera Alignment 선택기는 두 옵션이 있는 세그먼트 컨트롤입니다 — Apple Photogrammetry (App Store 빌드의 기본값, 완전 Sandbox 호환) 와 Native (Beta) (RadianceKit 자체의 FAST+BRIEF+GLOMAP 파이프라인 백엔드, Phase 3.8/3.9 개발, 2026-05)

기준). Native (Beta) 는 궤도 전용으로 검증되었고 $\geq 1\,000$ 프레임에서 Apple Photogrammetry보다 빠르지만 Phase 3 §5 품질 게이트 ($\text{finalLoss} \leq 0.0115$) 를 아직 충족하지 못합니다 — 따라서 Beta 태그. Metashape, COLMAP 또는 다른 Photogrammetry 소프트웨어에서의 외부 SfM 결과는 추가로 File 메뉴를 통해 가져올 수 있습니다 (Q3 COLMAP 텍스트 형식, Q6 Workspace 가져오기) — 선택기가 전환되지 않지만 가져온 포즈는 SfM 결과를 대체합니다.

SfM은 **Structure from Motion** 을 의미합니다. 겹치는 사진 세트에서 소프트웨어는 각 이미지에 대해 공유 3D 좌표계에서 카메라의 위치와 시선 방향을 재구성합니다. 그 과정에서 거친 3D 점 구름이 생성되며 Gaussian Splatting의 학습을 초기화합니다. SfM 결과는 실제 학습의 입력이며 이후 이미지 품질을 결정적으로 좌우합니다.

RadianceKit은 5개의 SfM 경로를 제공합니다: 앱에 내장된 두 백엔드 (Q1 Apple Photogrammetry, Q4/Q5 Native), 외부 도구에서의 두 가져오기 경로 (Q3 COLMAP 텍스트 형식, Q6 바이너리 Workspace 가져오기), 그리고 App Store 외부의 Developer 빌드에서만 사용 가능한 Q2 COLMAP Binary. 어느 것이 적절한지는 장면 유형 (객체 주위 궤도, 실내 공간, 드론 비행) 과 외부 소프트웨어가 이미 재구성을 제공하는지에 따라 다릅니다.

Q1 — Apple Photogrammetry



위치

Expert View → 인스펙터 → Training Configuration
→ Camera Alignment 선택기, 항목 "Apple Photogrammetry".



기술적 설명

원래 Object Capture용으로 개발된 Apple의 내장 Photogrammetry 프레임워크를 래핑합니다. Apple은 내부적으로 독점 파이프라인 (단계는 공개적으로 문서화되지 않음) 으로 특징을 추출하고, Multi View 매칭을 통해 검증하고, Apple Silicon Neural Engine + GPU에서 Bundle Adjustment를 해결합니다. 백엔드는 완전히 App Store 호환 (외부 바이너리 없음, Sandbox=true, on-device) 이지만 카메라 포즈와 거친 점 구름만 제공합니다 — Track Length나 Reprojection Error 같은 진단 메트릭은 없습니다. Apple의 권장에 따라 수백 이미지까지 확장됩니다. 선형 드론 비행이나 큰 야외 장면에서 ~500 프레임을 초과하면 재현 가능한 충돌이나 개별 카메라의 조용한 폐기가 관찰되었습니다.

간단히 말하면

Apple은 Apple의 내장 Photogrammetry 프레임워크를 래핑합니다. Apple은 내부적으로 독점 파이프라인 (단계는 공개적으로 문서화되지 않음) 으로 특징을 추출하고, Multi View 매칭을 통해 검증하고, Apple Silicon Neural Engine + GPU에서 Bundle Adjustment를 해결합니다. 백엔드는 완전히 App Store 호환 (외부 바이너리 없음, Sandbox=true, on-device) 이지만 카메라 포즈와 거친 점 구름만 제공합니다 — Track Length나 Reprojection Error 같은 진단 메트릭은 없습니다. Apple의 권장에 따라 수백 이미지까지 확장됩니다. 선형 드론 비행이나 큰 야외 장면에서 ~500 프레임을 초과하면 재현 가능한 충돌이나 개별 카메라의 조용한 폐기가 관찰되었습니다.

POWER-USER

Q2 COLMAP Binary — 외부 COLMAP 프로그램을 서브 프로세스로 Spawn하므로 App Store 버전 (Sandbox) 에서는 **사용할 수 없습니다**. App Store 외부의 Developer 빌드에서만 작동합니다. COLMAP이 제공하는 품질을 위해 App Store 버전에서는 Workspace 가져오기 (Q3 또는 Q6) 가 있습니다: COLMAP이나 Metashape에서 외부적으로 SfM을 계산하고 결과를 로드하십시오.

Q3 — COLMAP [?] [?] [?] [?] (Metashape / ETH3D)

위치

메뉴 "File → Import COLMAP / Metashape Workspace..." (Cmd+Shift+I) 또는 `sparse/0/cameras.txt`가 있는 폴더의 드래그 앤 드롭.

기술적 설명

표준화된 COLMAP 텍스트 내보내기를 읽습니다 — `sparse/0/` 하위 폴더의 세 텍스트 파일 `cameras.txt`, `images.txt`, `points3D.txt` — 그리고 내부 SfM 결과 모델로 변환합니다. COLMAP 바이너리 내보내기와 동일한 형식 정의이며, 바이너리 대신 ASCII로 작성됩니다. Agisoft Metashape, RealityCapture, PolyCam, ETH3D 벤치마크가 정확히 이 레이아웃으로 출력합니다. 파서는 Binary 파서와 카메라 모델 감지를 공유합니다 (모든 일반적인 모델: SIMPLE_PINHOLE, PINHOLE, OPENCV, OPENCV_FISHEYE, FULL_OPENCV). 주석 줄과 빈 줄에 강건. 테스트에서 ~1 400 카메라 (ETH3D Tunnel) 까지 문제없이 확장됩니다.

간단히 말하면

[?] [?] Metashape, RealityCapture
[?] [?] [?] [?] [?] Photo 3D [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] — [?] [?] [?] [?] [?]
RadianceKit[?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?].
[?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?]
[?] [?] [?] [?] [?] [?] [?] [?] [?].

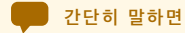
Q4 — Native SfM (???)



위치
Expert View → 인스펙터 → Training Configuration → Camera Alignment 선택기, 항목 "Native (Beta)". 증분이 이 백엔드의 기본 모드 — 인스펙터에 별도의 매퍼 선택기는 없습니다. CLI를 통해 모드를 명시적으로 설정하려면 `--native-sfm` 또는 `--sfm-mapper incremental`을 사용합니다.



기술적 설명
전체 SfM 파이프라인의 자체 GPU 가속 구현: FAST+BRIEF 특징량 또는 CoreML을 통한 SuperPoint+LightGlue (`--coreml-features` 포함), 그 다음 Hamming KNN 매칭, RANSAC 기본 행렬, Track 구축, 초기 페어 선택, 2 View Bootstrap ($F \rightarrow E + DLT$), PnP 등록과 Multi View 삼각측량을 사용하는 Greedy 증분 매퍼, 마지막으로 Huber Loss와 해석적 Jacobian을 Cholesky Solve 로 사용하는 Schur 축소 Levenberg-Marquardt Bundle Adjustment. 완전 App Store 호환: 외부 바이너리 없음, Sandbox=true. Phase 3.10에서 제공된 R2 Collapse Detector 포함: 애플이 입력 프레임의 60% 미만을 등록하거나 카메라당 점 비율이 13 아래로 떨어지면 자동으로 글로벌 매퍼 (Q5) 로 폴백합니다. 경험적으로 궤도/턴테이블 장면에서 깨끗합니다. 더 일반적인 움직임 (드론 비행, 복잡한 형상의 실내 공간)에서는 성공률이 더 낮습니다 — 그러나 Detector가 이러한 케이스를 잡습니다. ~200 카메라까지 안정적으로 확장되며, 그 이상은 상당히 더 긴 실행 시간으로.



간단히 말하면
Apple (App Store), (???)
 (???) (???) (???) (???) (???) (???)
 (???) (???) (???) (???) (???) (???)
 (???) (???) (???) (???) (???) (???)
 (???) (???) (???) (???) (???) (???)
 (???) (???) (???) (???) (???) (???)
 (???) (???) (???) (???) (???) (???)
 (???) (???) (???) (???) (???) (???)
 (???) (???) (???) (???) (???) (???)
 (???) (???) (???) (???) (???) (???)
 (???) (???) (???) (???) (???) (???)
 (???) (???) (???) (???) (???) (???)
 (???) (???) (???) (???) (???) (???)
 (???) (???) (???) (???) (???) (???)
 (???) (???) (???) (???) (???) (???)
 (???) (???) (???) (???) (???) (???)
 (???) (???) (???) (???) (???) (???)
 (???) (???) (???) (???) (???) (???)
 (???) (???) (???) (???) (???) (???)
 (???) (???) (???) (???) (???) (???)
 (Q3 (???) Q6) (???) (???) (???)

Q5 — Native SfM (???)

📍 위치

증분 매퍼 (Q4) 가 Collapse Detector를 트리거한 경우 (입력 프레임의 60% 미만이 등록되거나 카메라당 점 비율이 13 미만) 자동으로 호출됩니다. CLI `--sfm-mapper global`을 통해서만 수동으로 강제 가능합니다. 인스펙터에서 글로벌 방법은 별도의 선택기를 통해 접근할 수 없습니다 — 앱이 전환 시점을 자체적으로 결정합니다.

⚙️ 기술적 설명

네이티브 파이프라인의 글로벌 변형. 먼저 Q4와 같이 특징 추출 + 매칭, 그 다음 모든 검증된 페어의 상대 Pose 추정, 그 다음 Rotation Averaging (세계 좌표계 내 모든 카메라 회전을 동기화) 과 Translation Averaging (큰 카메라 수에서 정수 오버플로를 피하기 위해 행렬 자유 Sparse Formulation의 LSQR 기반). 원칙적으로 ~5 000 카메라까지 확장되지만 실제로는 수백 카메라를 초과하면 품질이 저하됩니다 — K-1351에서의 Phase 3.8 §5 수용 게이트 측정은 목표 0.0115 대신 finalLoss 0.07을 산출했습니다. “Fallback Tier”로 처리됩니다: 증분 매퍼가 퇴화될 때 등장하지만 자체적으로는 품질이 다시 확인되지 않습니다.

💬 간단히 말하면

????? Plan B ???.
?? ? ? ? ? ? ? ? ? ? ?
???? ? ? ? ? ? ? ? ? ? ?
???? ? ? ? ? ? ? ? ? ? ?
???? ? ? ? ? ? ? ? ? ? ?
???? ? ? ? ? ? ? ? ? ? ?
Metashape?? ? ? ? COLMAP ?
???? ? ? ? ? ? ? ? ? ? ?
???? ? ? ? ? ? ? ? ? ? ?
???? ? ? ? ? ? ? ? ? ? ?
? Workspace ? ? ? ? ? (Q3 ? ?
Q6) ? ? ? ? ? ? ? ? ? ?.

Q6 — Metashape / COLMAP Workspace (Phase Q7)



File 메뉴 -> "Import COLMAP / Metashape Workspace..." (Cmd+Shift+I). sparse/0/cameras. {bin,txt} 와 images/가 있는 폴더의 드래그 앤 드롭.



기술적 설명
드래그 앤 드롭 또는 Open 패널에서 선택된 폴더가 세 COLMAP Workspace 레이아웃 (sparse/0/, sparse/, 또는 루트) 중 하나에 해당하는지, 그리고 재구성 바이너리 (cameras.bin) 인지 텍스트 (cameras.txt) 인지를 자동 감지합니다. 바이너리 경로는 COLMAP 바이너리 파서를 사용하고, 텍스트 경로는 ETH3D 로더를 사용합니다 - 둘 다 동일한 SfM 결과 모델을 생성하며 파이프라인의 나머지 (이미지 가져오기, MCMC 학습 시작) 는 소스에 무관합니다. 이미지는 앱 Sandbox 북마크 시스템을 통해 Security Scoped로 열리므로 가져오기는 App Store 버전에서도 작동합니다. 특히 "재구성 없이 Metashape 내보내기를 다시 계산" 케이스용으로 의도되었습니다. File 메뉴 항목에서 언급된 감지는 선택된 폴더가 인식 가능한 Workspace가 아닐 경우 앱 로그에 경고합니다.

간단히 말하면

Metashape RealityCapture
Metashape RealityCapture
SfM
RadianceKit SfM
Eth3D

어느 백엔드를 언제?

Table with 2 columns: 시나리오 (Scenario) and 권장 백엔드 (Recommended Backend). Rows include scenarios like '객체 스캔, 50-200 사진', '큰 야외 / 드론 / >500 이미지', and 'Metashape/RealityCapture 내보내기 존재' with their corresponding backends like 'Q1 Apple Photogrammetry', 'Q6 Workspace 가져오기', etc.

빠른 비교

백엔드	App Store	Sandbox	외부 바이너리	Best Use	최대 ~Cams
Q1 Apple PG	✓	✓	—	궤도 객체	~300
Q2 COLMAP Binary	✗ (Developer 빌드만)	—	colmap/glomap	야외 대형	~5 000
Q3 COLMAP 텍스트 가져오기	✓	✓	—	Bench rigs	~1 500
Q4 Native 증분	✓	✓	—	궤도 객체	~200
Q5 Native 글로벌	✓	✓	—	Q4 폴백	~1 351
Q6 Workspace 가져오기	✓	✓	—	Metashape 재사용	소스 별

2

제 10 장 — 초보자 모드

초보자 모드 (영어 Simple Mode, Cmd+1) 는 3D Gaussian Splatting 장면을 처음 재구성하는 모든 사람을 위한 가이드형 워크플로입니다. 인스펙터 필드로 가득한 사이드바를 표시하는 대신 앱이 네 단계를 안내합니다: 먼저 이미지나 비디오를 가져오고 품질 프리셋을 선택, 그 다음 처리가 실행 (SfM + 학습), 그 다음 완성된 장면을 3D 미리보기에서 평가할 수 있고, 마지막으로 원하는 형식으로 내보냅니다. 창 상단 가장자리의 좁은 진행률 표시줄은 항상 어느 단계에 있는지 표시합니다.

모든 컨트롤 필드를 동시에 보여 주는 Expert 모드 (Cmd+2) 와 비교하여 초보자 모드는 사용되지 않는 옵션을 숨기고, 이미지가 너무 적거나 나쁘면 검증 경고를 제공하며, 각 단계에서 현재 상태에서 의미 있는 버튼만 제공합니다. 언제든지 초보자 모드와 Expert 모드 사이를 전환할 수 있고 (Cmd+1 / Cmd+2), 전체 상태 — 가져온 이미지, 선택된 프리셋, 현재 실행 중인 학습, 완성된 점 구름 — 가 유지되며 각 다른 모드에서 즉시 사용할 수 있습니다.

Z1 — 가져오기 (이미지와 프리셋 선택)

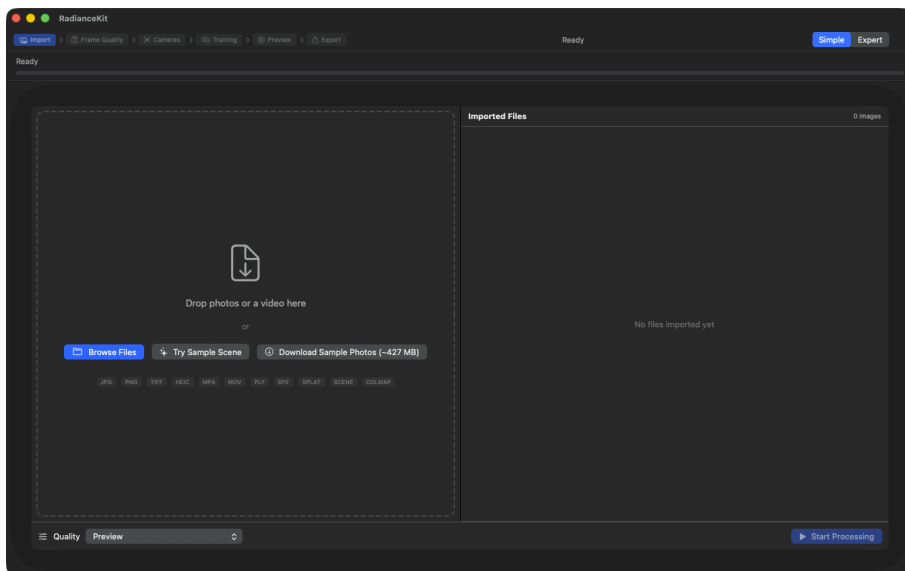


그림 35: 초보자 모드 1단계 — 이미지 가져오기 전 빈 드롭 영역, 위에 Crumb Trail (Import → Frame Quality → Cameras → Training → Preview → Export), 형식 핀 JPG/PNG/TIFF/HEIC/MP4/MOV/PLY/SPZ/SPLAT/SCENE/COLMAP

이미지에 보이는 것 Crumb Trail (Import 활성화) 이 4단계 워크플로를 보여 줍니다. 세 CTA가 있는 왼쪽 드롭 영역: "Browse Files" (NSOpenPanel), "Try Sample Scene" (번들된 데모), "Download Sample Photos (~427 MB)" (Mip-NeRF360 flowers 하위 집합). 그 아래의 형식 핀은 모든 허용된 파일 유형을 나열합니다. 오른쪽에 카운터 "0 images"와 빈 상태 "No files imported yet"가 있는

“Imported Files”. 아래에 Quality 선택기 (기본값: Preview) 와 “Start Processing” (이미지가 없는 동안 비활성).

첫 번째 단계는 앱에 이미지 자료를 제공하는 것입니다. 중간의 큰 점선 필드로 드래그 앤 드롭, “Browse Files” 버튼을 통해, 또는 동봉된 Sample 장면을 클릭하여. 오른쪽에 해상도와 파일 크기가 있는 모 든 가져온 이미지 목록이 나타납니다. 아래의 떠 있는 도구 모음에서 품질 프리셋을 선택하고 “Start Processing”으로 파이프라인을 시작합니다. 검증 경고 (< 3 또는 < 10 이미지에서 빨강, 10–19 이미지 에서 주황) 는 앱이 의미 있는 재구성을 기대하는지 여부를 나타냅니다.

C-01 ProgressIndicator (???)

위치

워크플로 위, 항상 표시.

기술적 설명

Stage Allocation이 있는 전체 파이프라인 (Frame-Quality → SfM → Training) 에 걸친 수평 진행률 표시줄을 표시합 니다: Frame-Quality는 0–5% (Phase 3.11, 매우 짧음) 를 차지하고, SfM은 표시줄의 0–30%, Training은 30–100%. 그 옆에 상태 텍스트와 단계 이름 백분율 표시 (“SfM 41%”, “Training 12 500/20 000”) 가 있어 사용자가 “41% SfM → 25% Training” 의 명백한 후퇴를 오류로 읽지 않도록 합니 다 — 표시줄은 전체 파이프라인 진행을 보여 주며 하위 단계가 아닙니다. ETA 계산은 충분한 학습 속도가 측정된 즉시 시작됩 니다 (일반적으로 처음 100 반복 후). 같은 표시는 Expert 모 드의 인스펙터 위에서도 사용됩니다.

간단히 말하면

ProgressIndicator는 전체 파이프라인 (Frame-Quality → SfM → Training) 에 걸친 수평 진행률 표시줄을 표시합니다. Frame-Quality는 0–5% (Phase 3.11, 매우 짧음) 를 차지하고, SfM은 표시줄의 0–30%, Training은 30–100%. 그 옆에 상태 텍스트와 단계 이름 백분율 표시 (“SfM 41%”, “Training 12 500/20 000”) 가 있어 사용자가 “41% SfM → 25% Training” 의 명백한 후퇴를 오류로 읽지 않도록 합니다. ETA 계산은 충분한 학습 속도가 측정된 즉시 시작됩니다 (일반적으로 처음 100 반복 후). 같은 표시는 Expert 모드의 인스펙터 위에서도 사용됩니다.


C-13 File List ForEach (?? ? ? ? ? ? ?)

 **위치**

오른쪽 이미지 목록, 각 항목.

 **기술적 설명**

Swipe to Delete가 있는 가져온 이미지 목록. 이미지당 아이콘, 파일 이름, 해상도 ("1920 x 1080"), 파일 크기 (KB/MB로 형식화) 가 있는 한 줄. 해상도는 인터페이스를 차단하지 않도록 이미지 헤더에서 비동기적으로 채워지는 메타데이터 캐시에서 옵니다. 삭제 동작은 macOS 스타일 Swipe Delete (줄에서 트랙패드 스와이프 왼쪽) 와 선택된 줄에서 키보드 Delete를 제공합니다. 참고: 명시적 마이너스 버튼, Backspace, Cmd-Z로 되돌리기가 있는 확장된 이미지 삭제 경로는 Expert 모드만 의 Project Navigator에 추가되었습니다 — 초보자 모드에서는 Swipe Delete로 남아 있습니다.

 **간단히 말하면**

?? ? ? ? ? ? ? ? ? ? ?
 ?? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? — ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? . ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? — iOS Mail ? ?
 — ? ? ? ? ? ? ? ? Delete ? ?
 ? ? ? ? ? . ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? . ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ? . ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ? Cmd-Z ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ? Expert ? ?
 (Cmd+2) ? ? ? ? ? ? ? ? . ? ? ?
 ? ? ? ? ? Project Navigator ? ? ? ? ?
 ? . ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ?

C-15 Validation Warnings (3?? ??)

위치

이미지 목록 아래, Clear All 버튼 위.

기술적 설명

가져온 이미지 수 (이미지가 있고 비디오가 없을 때만 활성)를 기반으로 한 세 가지 연속 임계값: - < 3 이미지: 빨간색 배너 (red octagon), 텍스트 "At least 3 images are required. Camera alignment cannot be computed from fewer images." - 3-9 이미지: 빨간색 배너, 텍스트 "With fewer than 10 images, SfM often fails and the trained scene tends to overfit [...]. 15-20 images minimum recommended; 30+ for object captures." - 10-19 이미지: 주황색 배너 (warning triangle), 텍스트 "Workable, but quality usually improves with 20+ images and good coverage around the scene." 20개 이미지부터 배너가 사라집니다. 임계값은 하드코딩되어 있고 경험적 560+ 학습 실험을 기반으로 합니다.

간단히 말하면

?? ? ?? ???? ???? ???? ?
 ??? ? ? ? ? ? ? ? ? ? ?
 ????. ??? ? ? ? ? ? ? ? : ?
 ? ? ? ? ? ? ? ? ? ? ? — SfM
 ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? Overfit
 ? ? ? ? ? ? ? ? ? ? ? ? : ?
 ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 : ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?

C-16 COLMAP Workspace

위치

폴더 드롭 시 — 보이는 버튼이 아니라 감지 로직.

기술적 설명

디렉터리 드롭 시 세 표준 Workspace 레이아웃 중 하나가 포함되어 있는지 확인됩니다: sparse/0/cameras.bin, sparse/cameras.bin 또는 루트에 직접 cameras.bin . 해당되면 표준 이미지 열거가 중단되고 대신 사용자에게 기존 재구성을 사용할지 또는 이미지를 Apple Photogrammetry 를 통해 다시 보낼지 묻는 모달 경고가 열립니다. Text Format Workspace (cameras.txt) 와 ETH3D 내보내기에도 같은 경로. 자세한 내용은 제 9 장 Backend Q6 참고. Expert 모드 와 마찬가지로 초보자 모드에서도 작동합니다.

간단히 말하면

Metashape, RealityCapture COLMAP sparse/0/cameras.bin sparse/cameras.bin 또는 루트에 직접 cameras.bin . RadianceKit COLMAP Workspace (sparse/0, cameras.bin) ETH3D Text Format Workspace (cameras.txt) Expert Sfm 9 Backend Q6

언제 다음 단계로?

(a) 최소 하나의 이미지나 비디오가 가져와졌고 (b) 검증 배너가 주황색이거나 사라졌으면 Start Processing을 클릭할 수 있습니다. 빨간 배너에서는 앱이 어쨌든 시작하게 하지만 높은 확률로 처리를 즉시 다시 중단할 수 있습니다. 권장: 최소 20개의 선명한 이미지, 연속된 촬영 사이에 명확한 오버랩이 있고 모두 모티프로부터 대략 같은 거리에서. 시작 전에 시간 예산에 맞는 프리셋을 선택하십시오 — 30개 이미지와 Quick 프리셋이라면 몇 분 만에 완료됩니다. Quality에서는 1-2 시간이 걸립니다.

Z2 — 처리 (SfM + 학습)

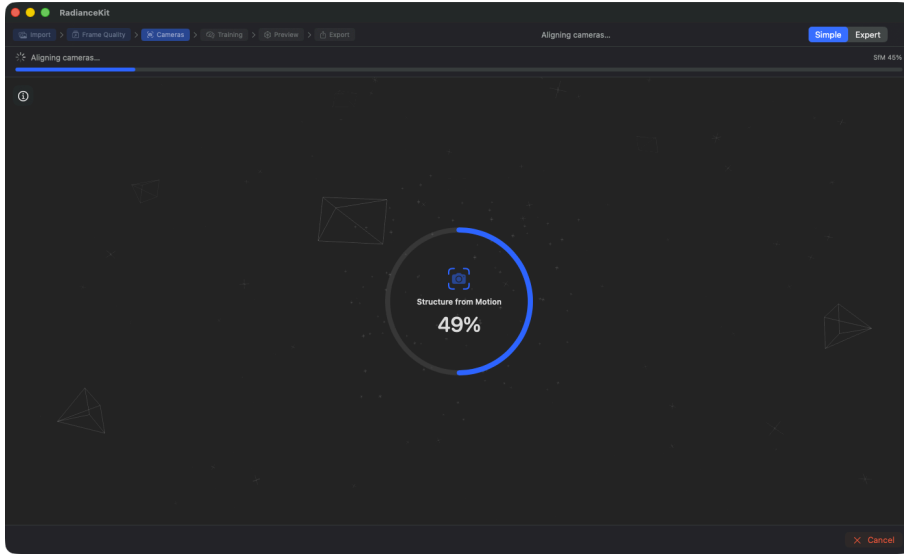


그림 36: Z2 SfM 단계 — 큰 원에 41%가 있는 Stage 아이콘 “Structure from Motion”, 위 상태 표시줄 “SfM 25%”, 오른쪽 아래 Cancel 버튼

SfM 단계 (카메라가 정렬됨): 큰 진행 원이 하위 단계 진행을 보여 줍니다 (여기서는 실행 중인 Apple Photogrammetry 세션의 41%). 왼쪽 위 상태 텍스트 “Aligning cameras...”. Crumb Trail은 “Cameras”를 활성 단계로 표시합니다. 위 상태 표시줄은 파이프라인 전체 진행을 보여 줍니다 (25%) — SfM이 표시줄의 첫 절반을 차지합니다. 배경의 떠 있는 와이어프레임 카메라는 포즈가 추정되고 있음을 나타냅니다.

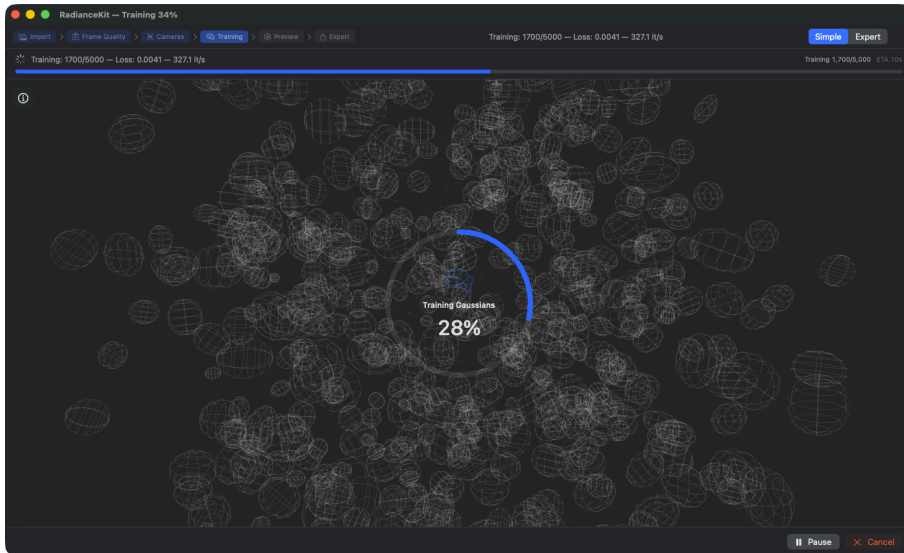


그림 37: Z2 학습 단계 — 6%가 있는 Stage 아이콘 “Training Gaussians”, 위 라이브 메트릭 (Training: 400/5000 — Loss: 0.1642 — 138.7 it/s), ETA 33s, 아래 Pause/Cancel

학습 단계 (Gaussian이 최적화됨): 하위 단계 아이콘이 “Training Gaussians”로 전환되고 백분율이 선택된 프리셋의 반복을 셉니다 (여기서는 Preview 프리셋의 경우 400 / 5 000 = 단계의 8%). 라이브 메트릭 줄은 Loss 값 (0.1642), 초당 반복 (138.7 it/s), ETA (33 s) 를 표시합니다. 파이프라인 전체 진행은 이 단계 동안 50%에서 100%로 올라갑니다. Pause 버튼 (SfM 단계의 Cancel Only 대신) 은 나중에 Resume을 허용합니다. Cancel은 학습 결과를 폐기하고 Z1로 돌아갑니다.

C-27 Retry

위치

하단 내비게이션 표시줄, 파이프라인이 실패한 경우 표시 (SfM 상태가 "SfM failed"로 시작하거나 학습이 오류 상태).

기술적 설명

강조 버튼. 전체 파이프라인을 새로 시작합니다. 시작 전에 가져온 이미지/비디오가 아직 있는지 확인됩니다. 이전 오류 로그는 JSONL 디렉터리에 유지됩니다. 새 실행은 현재 타임스탬프가 있는 새 로그 파일을 작성합니다.

간단히 말하면

SfM... (RANSAC, Densification) ... JSONL ... SfM ... Back ... Training Logs ... (Help -> Open Training Logs) ...

C-28 Inline Loss Chart

 위치

Info Panel, 오른쪽 열, 비어 있지 않은 진행 기록이 있는 학습 동안에만 표시.

 기술적 설명

컴팩트한 드로잉 영역 (40 픽셀 높이), Loss History를 강조 색상의 1픽셀 선으로 그립니다. 데이터는 유한 값에 대해 필터링됩니다 (불안정한 학습의 NaN 보호). Min/Max는 전체 History에 대해 계산됩니다 — 따라서 차트는 값 범위에 자동 줌됩니다. 마지막 Loss 값은 차트 위 오른쪽 위에 있습니다. History 자체는 각 학습 틱에서 앱 상태에서 구축됩니다 (일반적으로 100 반복마다).

 간단히 말하면

???"???"?? (?? ?
?? ? ? ?) ? ? ? ? ?
?? ? ? ? ? ? (?? ? ? ? ? ?
? ? ?) ? ? ? ? ? ? ? ?
? Loss ??. ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? — ?
? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ?
? — ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ?
? Info Panel ? ? ? ? ? ? ?
??.

Z3 — 미리보기 (3D 모델 회전)

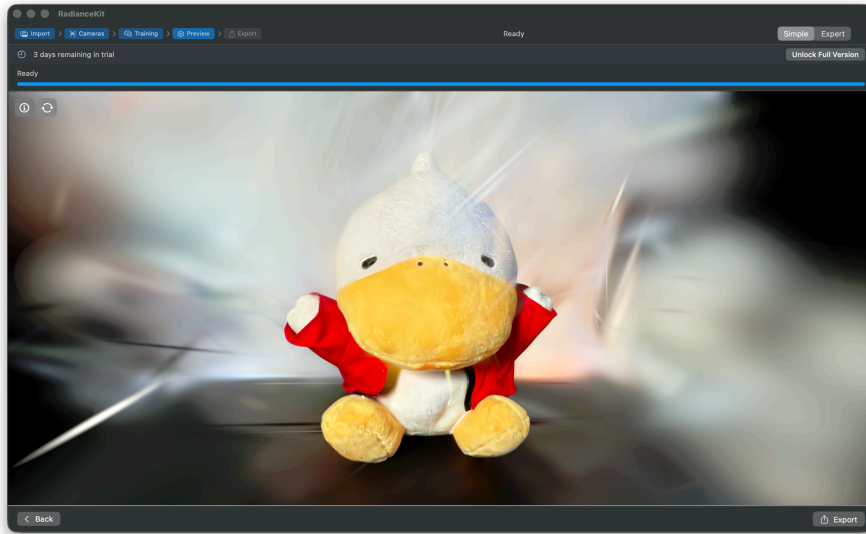


그림 38: 3D Viewer가 있는 초보자 모드 Preview 단계

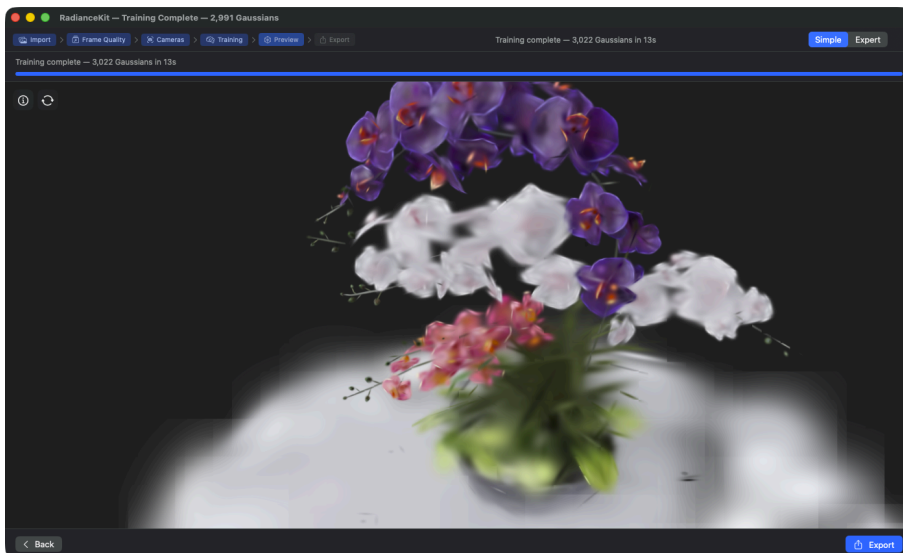


그림 39: 학습 완료 후 Z3 Preview — Bjoern의 Blender Bouquet 재구성, 헤더가 "Training complete — 3,022 Gaussians in 13s" 표시, 아래에 Back과 Export 버튼

이미지에 보이는 것 Crumb Trail이 "Preview"를 활성화 단계로 표시합니다. 전체 화면 3D 뷰포트가 완성된 학습된 Bouquet 장면을 렌더링합니다 (Bjoern의 합성 Blender 테스트 세트, 960 반구체 Cam에서 60 프레임 하위 집합). 헤더 상태 표시줄: "Training complete — 3 022 Gaussians in 13 s" — 최종 Gaussian 수와 학습 시간을 제공합니다. 뷰포트의 드래그는 카메라를 회전합니다 (Yaw/Pitch). 스크롤 휠은 View Direction을 따라 줌합니다. "Back" 버튼 (왼쪽 아래) 은 Resume이나 Re-Run을 위해 Z2로 돌아갑니다. "Export" 버튼 (오른쪽 아래, primary) 은 Z4로 더 이동합니다.

학습 완료 후 앱이 자동으로 미리보기에 도착합니다. 여기서 완성된 Gaussian Splatting 모델을 전체 화면 Metal 뷰에서 보고 마우스와 트랙패드로 회전, 줌, 팬할 수 있습니다. 뷰포트 위쪽에 카메라 제어와 정보가 있는 작은 오버레이가 있습니다 — 자동 회전, 학습 통계, 리셋 버튼. 다음 단계 (내보내기) 전에 다양한 각도에서 모델을 확인하여 재구성이 깨끗한지 확인하는 것이 좋습니다.

C-36 SplatViewportView (3D ? ? ?)

📍 위치

Preview 단계의 전체 화면 배경.

🔧 기술적 설명

완성된 점 구름을 렌더링하는 Metal 기반 3D 뷰포트. 렌더러는 RadianceKit 자체의 ForwardPass 래스터라이저 — 학습 중에 이미 Splat을 렌더링하는 바로 그것 — 이므로 진정한 WYSIWYG입니다 (학습된 것이 정확히 동일하게 표시되고 내 보내집니다). Order Independent Transparency를 사용하는 타일 기반 렌더링 파이프라인. 렌더러를 초기화할 수 없는 경우 (예: 시스템에서 Metal을 사용할 수 없음) 대신 "Metal not available" 텍스트가 있는 검은색 배경이 나타납니다. 뷰는 Safe Area를 무시하므로 모델이 창 가장자리까지 도달합니다.

💬 간단히 말하면

3D 뷰포트. Metal 기반 3D 뷰포트 GPU 렌더러. ForwardPass 래스터라이저. WYSIWYG입니다. Order Independent Transparency를 사용하는 타일 기반 렌더링 파이프라인. 렌더러를 초기화할 수 없는 경우 (예: 시스템에서 Metal을 사용할 수 없음) 대신 "Metal not available" 텍스트가 있는 검은색 배경이 나타납니다. 뷰는 Safe Area를 무시하므로 모델이 창 가장자리까지 도달합니다. Mac Metal GPU 렌더러. ForwardPass 래스터라이저. WYSIWYG입니다. Order Independent Transparency를 사용하는 타일 기반 렌더링 파이프라인. 렌더러를 초기화할 수 없는 경우 (예: 시스템에서 Metal을 사용할 수 없음) 대신 "Metal not available" 텍스트가 있는 검은색 배경이 나타납니다. 뷰는 Safe Area를 무시하므로 모델이 창 가장자리까지 도달합니다.

C-37 CameraControlsOverlay (? ? ? ? ? ?)

📍 위치

뷰포트 위, 떠 있음.

🔧 기술적 설명

Auto-Rotation (Turntable), Reset-Camera, 배경 선택 (Gray/Black/White), Save-Screenshot, Toggle-Info-Panel 버튼이 있는 콤팩트한 UI 오버레이. 카메라 매개변수 (거리, 방위각, Elevation, Target, FOV) 에 바인딩하고 자동 Turntable을 제어합니다. 학습 중 (사용자가 Expert 모드에서 뷰포트가 실행되는 것을 보고 싶을 때) 오버레이는 추가로 콤팩트한 학습 상태 줄을 표시합니다.

💬 간단히 말하면

Auto-Rotation (Turntable), Reset-Camera, 배경 선택 (Gray/Black/White), Save-Screenshot, Toggle-Info-Panel 버튼이 있는 콤팩트한 UI 오버레이. 카메라 매개변수 (거리, 방위각, Elevation, Target, FOV) 에 바인딩하고 자동 Turntable을 제어합니다. 학습 중 (사용자가 Expert 모드에서 뷰포트가 실행되는 것을 보고 싶을 때) 오버레이는 추가로 콤팩트한 학습 상태 줄을 표시합니다.


C-38 Export () () () () () () () ()

 위치

Z3의 하단 내비게이션 표시줄.

 기술적 설명

레이블 "Export"와 Share 아이콘이 있는 강조 버튼. 클릭은 Z4로의 전환을 트리거합니다. 그 전에 상위 보기는 정식 버전이 잠금 해제되었는지 확인합니다 — 그렇지 않으면 Export 단계 대신 잠금 보기가 표시됩니다 (U-06 참고).

 간단히 말하면

Export Export
 Premium Premium.
 Back Back

언제 다음 단계로?

내보내기 전에 모델을 완전히 한 번 회전시키고 확인하십시오: 입력 이미지에서 다른 모든 영역이 존재합니까? 떠 있는 "Floater" (공기 중에 자유롭게 떠 있는 Gauss Splat 클라우드) 가 있습니까? 배경/하늘이 깨끗하게 또는 번져 보입니까? 심각한 문제는 재학습으로만 수정할 수 있습니다 — 더 많은 이미지, 다른 프리셋 또는 Expert 모드의 Floater Reduction 설정으로.

Z4 — 내보내기 (형식 선택과 저장)

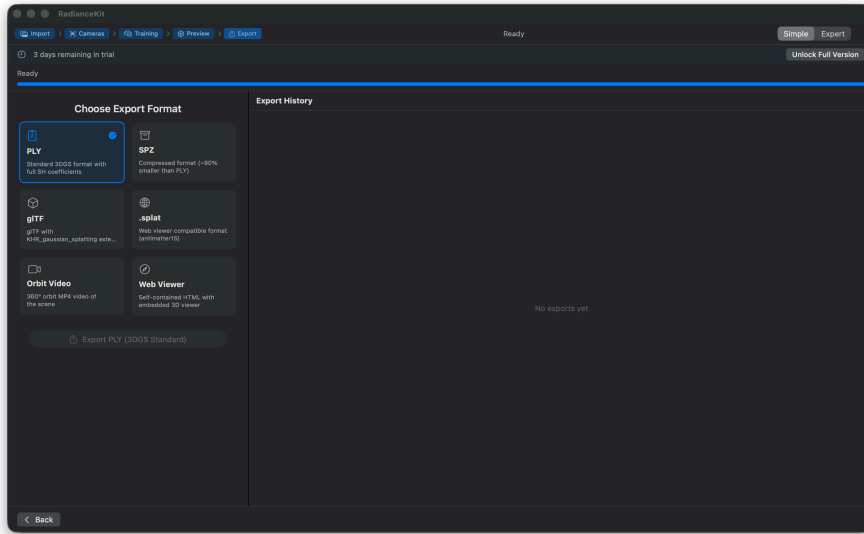


그림 40: 형식 카드가 있는 초보자 모드 Export 단계

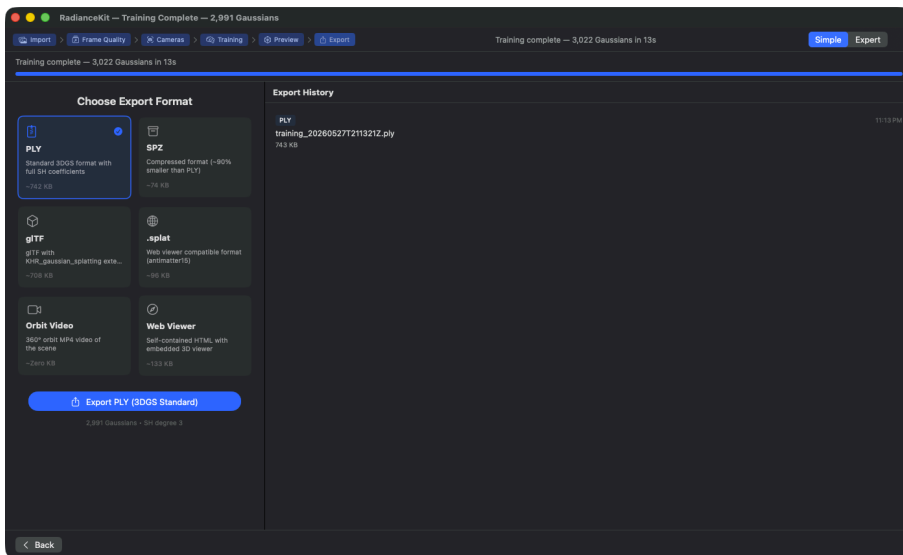


그림 41: Z4 Export 카드 — 6 형식 (PLY 742 KB 선택, SPZ 74 KB, glTF 708 KB, .splat 96 KB, Orbit Video, Web Viewer 133 KB), 이미 내보낸 PLY가 있는 오른쪽 Export History 사이드바

이미지에 보이는 것 Crumb Trail이 “Export”를 활성 단계로 표시합니다. 모든 6개 옵션이 있는 왼쪽 카드 그리드 “Choose Export Format”: PLY (표준 3DGS, 742 KB, 전체 SH 계수와 함께 — 여기서 파란색 체크 마크와 함께 미리 선택됨), SPZ (압축 3DGS 형식, PLY보다 ~90% 작음, 74 KB), glTF (KHR_gaussian_splatting 확장과 함께, 708 KB), .splat (antimatter15을 통한 Web Viewer 호환, 96 KB), Orbit Video (장면의 360° MP4, 실시간 크기 계산), Web Viewer (내장된 3D Viewer가 있는 독립형 HTML, 133 KB). 크기 정보는 현재 Gaussian Count와 형식 오버헤드에서 실시간으로 계산됩니다. 오른쪽 “Export History”는 형식 핀, 파일 이름, 타임스탬프와 함께 이미 완료된 내보내기를 나열합니다 — 클릭은 Finder에서 표시합니다. 왼쪽 아래 Primary CTA: “Export PLY (3DGS Standard)”와 Gaussian 부제목 “2,991 Gaussians · SH degree 3”.

마지막 단계에서 2열 카드 그리드를 통해 6개의 내보내기 형식 (PLY, SPZ, glTF, .splat, Orbit Video, Web Viewer) 에서 선택하고 Export를 클릭하고 macOS 대화 상자에서 저장 위치를 선택합니다. 오른쪽에 이전 모든 내보내기의 History가 실행됩니다 — 카드 선택에서 각 카드 아래에 추정 파일 크기가 즉시 표시되므로 예: 웹으로 가려면 SPZ를 선호하고 (작음) 다른 소프트웨어 (SuperSplat, Postshot, 플러그인을 통한 Blender) 로 가져오려면 PLY를 선호합니다 (크고 완전).

C-39 2 ? ? ? ? ? ?

📍 위치

Export 단계의 왼쪽 메인 페이지.

🔧 기술적 설명

두 개의 유연한 열과 12 포인트 간격이 있는 카드 그리드. 초보자 모드에서 제공되는 형식 위에서 반복합니다 — 가장 중요한 6개 형식만 포함하는 전체 형식 목록의 필터링된 하위 집합: PLY, SPZ, glTF, .splat, Orbit Video, Web Viewer. Compressed PLY와 SOG는 Expert 모드에서만 제공됩니다.

💬 간단히 말하면

? ? ? ? ? ? ? ? ? ? ? ? 6 ?
? ? ? ? ? ? ? ? ? ? ? ? : PLY (?
? ? 3D ? ? ? ? ? ? ? ? ?), SPZ (?
? ? ? ? ? ? ? ?), glTF (? ? ? Web3D
? ? ?), .splat (antimatter15 Web
Viewer ?), Orbit Video (? ? ?
? ? ? ? ? ? ? ? MP4), Web
Viewer (? ? ? 3D Player ? ? ? ?
? ? ? HTML ? ? ?). ? ? ? ? ? ? ?
? ? ? 90% ? ? ? ? ? ? . ? ? ?
? ? ? ? ? ? (Compressed PLY ? ?
? ? ? ? ? ? SOG) ? ? ? ? ? ?
? ? ? ? Expert ? ? ? ? ? ? ? ?
? ? ? . ? ? ? ? ? ? ? ? 8 ? ? ? ? ?
? ? ? ? ? ? ? ? . ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? .


C-40 **Format Card** [?][?]

 위치

그리드의 각 카드.

 기술적 설명

카드 레이아웃이 있는 단순한 버튼: 위에 아이콘 (예: PLY용 Document Zipper, SPZ용 Archive Box, Orbit Video 용 Video Icon), Format Name을 Headline으로, 설명 캡션 (2줄로 단축), 그 아래에 추정 파일 크기 (형식, Gaussian Count, SH Degree에서 실시간으로 계산되고 KB/MB로 형식화). 클릭 시 형식이 선택됩니다. 선택된 카드는 강조 배경, 강조 테두리, 오른쪽 위의 체크 마크 아이콘을 받습니다. 도구 설명은 형식 설명입니다.

 간단히 말하면

[?][?][?] [?][?][?] [?][?]. [?][?][?] [?]
 [?][?][?] [?][?] [?][?] [?][?] [?][?]
 [?] [?][?] [?][?] [?] [?][?] [?] [?] Export
 [?][?] [?] [?][?] [?] [?][?] [?] [?]
 ("Export PLY", "Export SPZ" [?]).
 [?] [?][?] [?] [?] [?] [?][?], [?][?], [?]
 [?] [?] [?][?], [?][?] [?] [?][?]
 [?] [?] [?] [?] [?] [?] [?] [?] [?]
 [?]. [?] [?] [?] [?] [?] [?] [?] [?]
 [?] [?] [?] [?] [?] — [?] [?] [?]
 [?] [?] [?] [?] [?] [?] [?] [?] [?]
 [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
 [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
 [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
 [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
 [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
 [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
 [?] [?] [?] [?] [?] [?] [?] [?] [?] [?].
 [?] [?] [?] [?] [?] [?] [?] [?] [?] [?].


C-41 **Video Duration** [?][?][?][?]

 위치

Format Grid 아래, 비디오 형식 (Orbit Video 또는 Social Video) 이 선택된 경우에만 표시.

 기술적 설명

3-30초 슬라이더, 1초 단계, 앱 상태의 Video Length에 바인딩. 최대 폭 300 픽셀. 비디오 형식이 선택된 경우에만 표시됩니다. 비디오가 아닌 형식에서는 슬라이더가 보기에서 완전히 제거됩니다 — 죽은 공간 없음.

 간단히 말하면

[?][?][?] [?] [?] [?] [?] Orbit Video [?] [?]
 [?] [?] [?] [?] [?] [?] [?] [?] [?]
 [?] [?] [?] [?]. 3[?] = [?] [?] [?] [?]
 [?], 30[?] = [?] [?] [?] [?] [?] [?]
 [?] [?] [?] [?]. [?] [?] [?] [?] Reel
 (Instagram, TikTok) [?] [?] [?] [?]
 6-10[?] [?] [?] [?] [?] — [?]
 [?] [?] [?] [?] [?] [?] [?] [?]
 [?] [?] [?] [?] [?] [?] [?] [?]
 [?]. [?] [?] [?] [?] [?] [?] [?] [?]
 [?] [?] [?] [?] [?] 15-20[?] [?] [?]
 [?] [?]. [?] [?] [?] [?] [?] [?] [?]
 [?] [?] [?] [?] [?] [?] [?] [?] [?].
 PLY [?] SPZ [?] [?] [?] [?] [?] [?]
 [?] [?] [?] [?] [?] [?] [?] [?].


C-44 Export Error Display

 위치

Progress Bar 아래, 마지막 내보내기에서 오류가 발생한 경우에만 표시.

 기술적 설명

Warning Icon과 오류 텍스트가 있는 빨간색 줄. 빨간색 8% 배경 Opacity, 둥근 모서리. 최대 폭 400 픽셀. 일반적인 오류 원인: SOG는 시스템 PATH에 cwebp 를 기대 (App Store 호환 아님). 가득 찬 디스크 공간에서의 쓰기 오류. 허용된 영역 외부의 저장 대상에서의 Sandbox 오류.

 간단히 말하면

Progress Bar 아래, 마지막 내보내기에서 오류가 발생한 경우에만 표시. Warning Icon과 오류 텍스트가 있는 빨간색 줄. 빨간색 8% 배경 Opacity, 둥근 모서리. 최대 폭 400 픽셀. 일반적인 오류 원인: SOG는 시스템 PATH에 cwebp 를 기대 (App Store 호환 아님). 가득 찬 디스크 공간에서의 쓰기 오류. 허용된 영역 외부의 저장 대상에서의 Sandbox 오류.


C-46 Export History List

 위치

Export 단계의 오른쪽.

 기술적 설명

내보내기 기록 (UserDefaults에 JSON으로 영구적으로 저장되고 각 성공적인 내보내기 후 유지) 에 대한 목록. 각 줄은 형식 배지 (작음, 강조 색상), 타임스탬프 (HH:mm), 파일 이름 (1줄로 단축), 형식화된 파일 크기를 표시합니다. 줄 클릭은 Finder를 선택된 파일과 함께 엽니다. 빈 상태: "No exports yet".

 간단히 말하면

Export 단계의 오른쪽. 빈 상태: "No exports yet".

이미지나 비디오를 아직 가져오지 않았습니다. 최소한 하나의 파일을 드롭 영역에 끌거나 “Browse Files”를 사용하십시오. 오른쪽 이미지 목록에 최소한 하나의 항목이 포함되는 즉시 버튼이 활성화됩니다. (1-2 이미지에서만 시작되지만 SfM이 오류로 직접 중단됩니다 — 빨간색 검증 배너 참고.)

왜 Export 버튼이 잠겨 있습니까?

초보자 모드에는 두 단계가 있습니다: (a) 학습 파이프라인이 아직 완료되지 않았고 가지고 있는 gaussianCloud가 없다면 버튼이 비활성화됩니다 — 먼저 Z2를 완료해야 합니다. (b) 정식 버전을 아직 구매하지 않은 경우 (`PurchaseManager.hasAccess == false`) Export 단계 대신 자물쇠 아이콘과 구매 시트를 여는 “Unlock Full Version” 버튼이 있는 잠금 보기를 봅니다. Quick과 Preview 프리셋은 무료 학습을 허용하지만 Export는 Premium입니다.

왜 프리셋을 선택할 수 없습니까?

선택할 수 있습니다 — 그러나 정식 버전을 구매하지 않고 Premium 프리셋 (Balanced, Quality, MCMC 변형) 을 탭하면 선택기가 자동으로 Preview로 다시 점프하고 Purchase Sheet가 열립니다. Quick과 Preview는 무료로 사용 가능한 유일한 프리셋입니다.

왜 이미지를 끌어넣어도 드롭 영역이 비어 있고 점선 회색이 유지됩니까?

아마도 UTI 유형 불일치입니다. 오픈 JPG, PNG, TIFF, HEIC, MP4, MOV와 앱 자체 Splat 형식을 받아들입니다. 다른 이미지 형식 (BMP, GIF, WebP, RAW 형식) 은 인식되지 않습니다. 이미지 유형이 있어야 한다고 확신한다면 파일 이름 확장자를 확인하십시오 — 앱은 주로 확장자에 따라 가며 파일 내용에 따라 가지 않습니다.

왜 30개 이미지만 가지고 있는데 SfM이 그렇게 오래 걸립니까?

Apple Photogrammetry는 선형으로 확장되지 않습니다 — 일부 이미지 구성에서 (복잡한 텍스처가 있는 실내 공간, 움직임 흐림, 나쁜 빛) 이미지 수가 시사하는 것보다 상당히 더 오래 걸립니다. 30 이미지에서 10+ 분 후에도 SfM이 여전히 멈춰 있다면 중단하고 더 나은 자료로 다시 시도하거나 Expert 모드로 전환하고 COLMAP/Native SfM을 시도하십시오 (Cmd+2 → 인스펙터 → Camera Alignment).

학습 로그는 어디에서 찾습니까?

Help → Open Training Logs (Cmd+⌘+L). 이는 ~/Documents/RadianceKit/Logs/를 엽니다. 각 학습 세션은 파일 이름에 타임스탬프가 있는 자체 JSONL 파일을 작성합니다 — 첫 줄은 구성이며, 그 다음 100 반복마다 진행 줄이 따르고, 마지막 줄은 Final Loss와 Success Flag가 있는 요약입니다.



판권

Set in SF Pro · Code in SF Mono · Typst 0.14 ·
2026년 06월 22일

© 2026 Bjoern Kindler · Bischofshofener Str. 9, 82008 Unterhaching, 독일

Made with ♥ in Unterhaching