



RADIANCEKIT

ユーザーマニュアル

Gaussian Splatting による
フォトリアリスティック 3D 再構築

Version 1.5.0 · macOS 26.0+ · 2026 年 5 月

BJOERN KINDLER · KINDLER-DEV.DE

概要

| | |
|---|-----|
| はじめに — 知っておくべきこと | 3 |
| RadianceKit とは? | 3 |
| Gaussian Splatting とは? | 3 |
| 第1章 — メニューバー | 5 |
| File メニュー | 6 |
| Mode メニュー | 9 |
| Training メニュー | 11 |
| Viewport メニュー | 14 |
| Export メニュー | 20 |
| Help メニュー | 25 |
| 注意: Edit メニューの Cmd-Z | 29 |
| キーボードショートカット概要 | 30 |
| 第2章 — Inspector (エキスパートビュー) | 31 |
| Look セクション (L1~L5) | 34 |
| Presets セクション (I1~I11) | 37 |
| トレーニング構成セクション (I12~I22) | 43 |
| Enhancements セクション (I26~I29、I42~I44) | 50 |
| メトリックセクション (I30~I38) | 57 |
| Loss チャートセクション (I39~I41) | 63 |
| いつ Inspector を使うか? | 66 |
| 第3章 — 設定 | 68 |
| General タブ | 69 |
| AI Helpers タブ | 74 |
| Inspector ミラー設定 | 77 |
| いつ何を? | 78 |
| 第4章 — 補助ウィンドウ | 79 |
| User Guide (W1~W4) | 80 |
| Keyboard Shortcuts (W5~W6) | 83 |
| Manage Storage (W7~W12) | 85 |
| Pareto Dashboard (W13~W22) | 89 |
| Holdout Analysis (W23~W29) | 96 |
| BayesOpt Console (W30~W39) | 101 |
| メインウィンドウ: Loss 進行と Gaussian Count (I39~I41、クロスリファレンス) | 107 |
| 経験則ボックス | 108 |
| 第6章 — トレーニング構成 | 110 |

| | |
|---|-----|
| 反復 (T1~T2) | 112 |
| Learning Rate (T3~T10) | 114 |
| Densification – Classic (T11~T16) | 119 |
| Loss (T17~T20) | 123 |
| SH 次数進行 (T21) | 126 |
| パフォーマンス (T22~T25) | 127 |
| 診断と点群準備 (T26~T30) | 130 |
| 正則化 (T31~T37) | 133 |
| Refinement (T38~T44) | 136 |
| Sky-Dome (T45~T48) | 140 |
| Adam + LR スケジュール (T49~T55) | 142 |
| ポスト処理 + Apple AI (T56~T60) | 145 |
| MCMC Densification (T61~T73) | 148 |
| Mip-Splatting (Q1.5) (T74~T76) | 155 |
| 適応 Densification (Q5) (T77~T79) | 157 |
| Curriculum (Q6) (T80~T81) | 159 |
| 静的プリセット (TP1~TP9) | 159 |
| メソッド: | 162 |
| どのフィールドが何のためか? (チートシート) | 163 |
| 危険なフィールド | 164 |
| 第7章 – 内蔵品質プリセット | 165 |
| どのプリセットをいつ? | 175 |
| クイック比較 | 176 |
| 独自プリセット | 177 |
| 第8章 – エクスポート形式 | 178 |
| どの形式をいつ? | 192 |
| クイック比較 | 193 |
| 第9章 – SfM バックエンド | 194 |
| どのバックエンドをいつ? | 200 |
| クイック比較 | 201 |
| 第10章 – 初心者モード | 202 |
| Z1 – インポート (画像とプリセットの選択) | 202 |
| Z2 – 処理 (SfM + トレーニング) | 211 |
| Z3 – プレビュー (3D モデルを回転) | 217 |
| Z4 – Export (形式選択 & 保存) | 220 |
| Expert モードへの切り替え | 225 |
| よくある質問 | 225 |

このマニュアルの読み方

本マニュアルの各項目は同じ構成になっています。左側には操作経路と 技術的な詳細が、右側には常に温かい色のサイドバーで簡単な説明が 記載されています。行頭の小さなアイコンを見れば、これからどのような 種類の情報が続くかを一目で把握できます。

4 つのアイコン



どこにあるか？ アプリ内の具体的なクリック経路 — メニューバー、Inspector のセクション、または初心者モードのステップです。関連する キーボードショートカットもここに記載されています。アイコンは 地図のピンで、この機能がユーザーインターフェイスのどこにあるかを 示しています。



詳細。 既定値、値の範囲、コードパスです。特にメニュー項目ではなく 数値パラメータであるトレーニング設定でよく登場します。アイコンは 小さな仕様カードです。



技術詳細。 この機能が内部で何を行うか、どのパラメータが影響し、 何に反応し、どのような副作用があるかを説明します。背後で何が 起きているかを理解したい読者向けです。アイコンはスライダー ブロックで、ボンネットの下にあるつまみを象徴しています。



簡単に言うと。 専門用語やコードなしで、主旨を明確な言葉で 伝えます。機能の用途や使うべきタイミングを素早く知りたい場合は、 まずこのセクションから読んでください。アイコンは吹き出しで 「要点をまとめると」を意味します。この欄は常に温かい砂色で 塗られているため、目がすぐに見つけられます。

章の色

各章には独自のアクセントカラーがあり、ID マーク（例えば **M1**）が各項目タイトルの左に、また小さなアイコンの前にあるので識別できます。 ページをめくるとき、現在どの章にいるかが一目でわかります。

- 1 メニュー
- 2 インスペクター
- 3 設定
- 4 補助ウィンドウ
- 6 トレーニング
- 7 プリセット
- 8 エクスポート
- 9 SfM
- 10 初心者モード

ナビゲーションのヒント

クイックスタート。 操作方法だけに興味がある場合は、直接 **第 10 章 — 初心者モード** に進んでください。これは 4 ステップのガイド版で、事前知識は一切必要ありません。

より深く学ぶ。 **第 2 章 — インспекター** と **第 7 章 — プリセット** では、エキスパートモードで使用できる操作要素と事前設定された品質 プロファイルを説明しています。

参照する。 目次と PDF 全文検索を使えば、特定の機能を見つけられます。マニュアルを最初から最後まで読む必要はありません。

はじめに — 知っておくべきこと

RadianceKit とは？

RadianceKit はネイティブ macOS アプリで、通常の写真群またはビデオから歩き回れる 3D 再構築を作成します。入力例えば、被写体の周り、室内、あるいは風景の上を撮影した 50 ~ 500 枚のショットです。出力は Gaussian Splatting シーンと呼ばれるもの — Mac 上でリアルタイムにあらゆる視点から見られる 3D モデルで、エクスポートして Web サイトに埋め込むことも可能で、主要な側面においてフォトリアルに見えます。

このアプリは Mac 上で完全にローカルで動作します — 画像は一切クラウドにアップロードされず、ログインも要求されず、サブスクリプションもありません。Apple Silicon Mac (M シリーズ) の GPU を集中的に活用します。完全なトレーニングはシーンとプリセットに応じて 2 分から数時間かかることがあります。計算中も Mac で通常通り作業を続けられ、RadianceKit はバックグラウンドで実行され、結果が完成すると通知します。

操作モードは 2 種類あります。初心者モード (Simple Mode) は、インポート → プリセット選択 → トレーニング → エクスポートのワークフローを 4 ステップで案内します。エキスパートモード (Expert Mode) では、全ての設定項目を備えた大きな Inspector、ライブプレビューウィンドウ、診断チャートが開きます。モード間はいつでも切り替え可能で、シーンのデータは保持されます。

Gaussian Splatting とは？

Gaussian Splatting (しばしば略して 3DGS または単に *Splatting*) は、2023 年にグラーツと INRIA の論文で発表された 比較的新しいフォトリアリスティックな 3D 表現手法です。アイデアはシンプルです。シーンを古典的なポリゴンメッシュ (三角形) や ボクセルグリッドとしてモデル化する代わりに、何百万もの小さく柔らかい 3D 雲から構成します — 一つひとつの雲は 3D ガウス分布 (それが名前の由来) で、位置、サイズ、形状、色、透明度を独自に持っています。これらの雲は、入力写真のあらゆる視角から組み合わせざったときに正しい画像が得られるように学習されます。

実際に言うと、Gaussian Splatting は反射、ハイライト、柔らかな葉、髪の毛、カーテンといったものを、古典的な 3D モデリングでは不可能、もしくは多大なコストでしか実現できない方法で表現できます。その代わりに、結果は古典的な意味で編集可能な 3D モデルでは

ありません — 単一の壁を動かしたり花瓶を置き換えたりすることはできません。むしろ、自由に歩き回れる空間の凍結されたショットのようなものです。建築ビジュアライゼーション、製品プレゼンテーション、バーチャルツアー、鑑識、文化遺産といった多くの用途では、まさにこれが適切な強みです。

入力画像から 3D シーンを作成するには、2 つのステップが必要です。まず、アプリは *Structure-from-Motion (SfM)* と呼ばれる手法で、各写真撮影時のカメラ位置を計算します。その過程で、シーンの粗い点群が副産物として生成されます。次に、実際の Gaussian Splatting トレーニングが開始されます。この粗い点群から出発して、何百万もの 3D 雲が徐々に分布、拡大、洗練され、位置と色が調整されて、すべての入力視角から一致する画像が得られるようになります。

RadianceKit を使用するにあたって、これらの仕組みを知っている必要はありません。初心者モードはこれらのステップを完全に隠します。しかし、エキスパートモードの診断数値 (Iteration、Loss、Gaussians、SSIM ...) が何を意味するか、なぜ一部のシーンが他より美しくなるかを理解したい場合は、マニュアルの後半の章で答えが見つかります。

章

第 1 章 — メニューバー

RadianceKit のメニューバーは、メインウィンドウや Inspector に直接ないすべての機能を整理します。主にシーン全体に影響するアクション (開く、保存する、新規プロジェクト)、トレーニングを制御するアクション (開始、一時停止、再開)、ビューポートを操作するアクション (自動回転、スクリーンショット、背景色)、さまざまな 3D およびメディア形式へのエクスポートをトリガーするアクションです。また、すべての補助ウィンドウ (User Guide、Pareto Dashboard、Holdout Analysis、BayesOpt Console) へのジャンプポイントも含まれます。

キーボードショートカットは、各メニュー項目の右側に表示されます。規約: **⌘** はコマンドキー (Apple キー)、**⇧** は Shift、**⌥** は Option (Alt)、**⌘** は Control を意味します。例えば、**⇧⌘T** は Shift+Command+T を表します。ここに記載されたすべてのショートカットは、Help → Keyboard Shortcuts (**⌘/**) 経由で独自の概要ウィンドウにも追加でリストされます。

以下の 42 のエントリは、インベントリの順序 (M1~M42) で記載され、関連するトップレベルメニューごとにグループ化されています。すべてのエントリは現在のコード状態 (175~477 行) に対して検証されています。インベントリから削除または時代遅れになったエントリはありません。新しい Edit メニュー項目 (Cmd-Z による「Remove Image」) は、システム NSUndoManager フレームワーク経由で記録され、そのため RadianceKitApp コードには表示されません (章末の注を参照)。

File メニュー

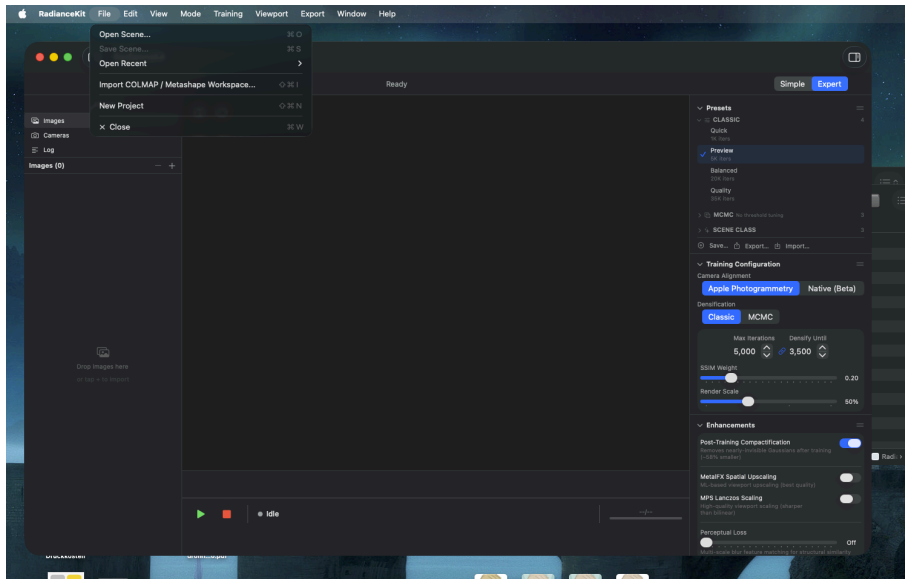


図 1: File メニューが展開された状態 — エントリ M1 から M6

File メニューは Apple の標準「New Window」 エントリをプロジェクト固有のアクションで置き換えます。シーンの読み込み/保存、動的な最近のリスト、Workspace インポート、空の状態へのハードリセットを含みます。

M1 File > Open Scene...

場所

メニューバー → File → Open Scene... (⌘O)。

技術詳細

RadianceScene バンドル、.ply、.splat、.spz 形式のファイルダイアログを開きます。単一選択で、ファイルとディレクトリ (バンドル形式用) の両方を表示できます。選択が成功した後、パスは最近のリストに登録され、シーンは非同期で読み込まれます — 前のシーンは置き換えられ、トレーニングパイプラインは読み込まれた状態で初期化されます。PLY/SPZ/Splat ファイルは、それぞれの形式ローダーで読み込まれます。.radiancescene バンドルは、マニフェスト、Cloud スナップショット、SfM 結果を含むディレクトリです。

簡単に言うと

これは、すでにトレーニング済みのシーンをアプリに再度読み込む方法です。RadianceKit 独自の形式、および他の Splatting プログラムが生成する標準形式 PLY, SPLAT, SPZ で機能します。例えば、夜通しトレーニングしたシーンを翌日に続行したり エクスポートしたいときに使用します。開くと、メインウィンドウの以前の状態が置き換えられます — 現在のシーンがまだ重要な場合は事前に保存してください。パスは自動的に「Open Recent」(M3) に保存され、次回より早くアクセスできます。

M2 File > Save Scene...

場所

メニューバー → File → Save Scene... (⌘S)。

技術詳細

コンテンツタイプ `RadianceScene` バンドルと事前入力されたファイル名 `scene.radiance_scene` を持つファイル保存ダイアログを開きます。`manifest.json`、シリアル化された Gaussian Cloud (PLY スナップショット)、SfM 結果のダンプを含むディレクトリパッケージを書き込み、再度開いた後も Continue トレーニングが機能します。Gaussian がまだ存在しない場合、このエントリは無効化されています。保存先はトレーニングログパスではなく、保存ダイアログが示す場所 — 通常は `~/Documents/` 以下です。

簡単に言うと

現在のシーンをファイルとして保存します (正確には、ファイルのように見えるパッケージフォルダ)。これを行ってからのみ、後で「Open Scene...」(M1) でこのシーンを再度開けます。パッケージには Gaussian Cloud と SfM 結果の両方が含まれるため、後で Continue トレーニング (M12~M14) を追加することもできます。トレーニングをまだ完了していない限り、エントリはグレーアウトされます。標準名は `scene.radiance_scene` ですが、保存ダイアログで独自の名前を指定できます。

M3 File > Open Recent > [シーン名]

場所

メニューバー → File → Open Recent → (リスト)。

技術詳細

最近開いたパスのリスト (設定に保存) から生成される動的サブメニュー。各リストエントリはファイル名で名付けられ、クリックで読み込まれます。リストが空の場合、無効化されたラベル「No Recent Scenes」が代わりに表示されます。Apple 風に、リストは最近開いた N 個のシーンを保持します — 制限は、設定への書き込み時に行われ、メニュービルダー自体ではありません。

簡単に言うと

ここでは最近開いたシーンが表示され、ファイルダイアログを経由せずにクリックして再び開けます。最近始めたばかりの場合、リストは空でメニューにはグレーで表示されます。「Open Scene...」(M1) で開く各シーンは、自動的にこのリストに保存されます。リストが多すぎる、またはプライバシー上の理由でクリアしたい場合は、「Clear Recent」(M4) を使用してください。

M4 File > Open Recent > Clear Recent

場所

メニューバー → File → Open Recent → Clear Recent。



技術詳細

設定の最近のリストをクリアします。確認ダイアログなしですぐに動作します。このエントリは、最近のリストにエントリが存在する場合にのみサブメニューに表示されます (パスの後、区切り線の下にあります)。

簡単に言うと

最近開いたシーンのリストを削除します。テストデータセットで遊んでいた後、パスを表示したくない場合に便利です。シーンファイル自体は削除されません — メニュー内のリンクのみです。アクションは確認なしですぐに動作します。その後、サブメニューには「No Recent Scenes」が表示されます。このエントリは、リストにシーンが存在する場合にのみ表示されます — リストが空の場合は見えません。

M5 File > Import COLMAP / Metashape Workspace...

場所

メニューバー → File → Import COLMAP / Metashape Workspace... (🔗)。



技術詳細

フォルダピッカーを開きます。COLMAP Workspace レイアウト (例えば `sparse/0/cameras.{bin,txt} + images/`) を持つフォルダを期待します。選択後、Workspace の事前チェックが実行されます — これは3つのレイアウト (`sparse/0/`、`sparse/`、ルート) と、再構築がバイナリ (`cameras.bin`) または ETH3D テキスト (`cameras.txt`) として存在するかどうかを検出します。成功した場合 Workspace がインポートされます。それ以外の場合はアプリログに警告のみが表示されます。完全なパイプラインロジックについては第9章「SfM バックエンド」Q6 も参照してください。

簡単に言うと

Metashape, COLMAP, RealityCapture, または同様のカメラ再構築用ソフトウェアを使用し、エクスポートを持っている場合、フォルダをここで読み込みます。RadianceKit は SfM ステージをスキップして、直接トレーニングを開始します — 大きなシーンで時間を節約します。メインウィンドウへのドラッグ&ドロップも同じように機能します。COLMAP レイアウトを持つフォルダが期待されます (つまり `cameras.*` を持つ `sparse/0/` と `images/` フォルダ)。サポートされているレイアウトとワークフローの詳細は、第9章「SfM バックエンド」にあります。

M6 File > New Project



場所

メニューバー → File → New Project (⇧⌘N)。



技術詳細

未保存の作業があるかどうかをチェックします。ある場合、何かが失われる前に確認ダイアログが表示されます。保存するものがない場合、リセットは直接実行されます — インポートされた画像、SfM 結果、Gaussian Cloud、トレーニング状態、すべての依存 UI インジケータをクリアします。注意: ユーザーが作成したプリセット ライブラリは保持されます。これはアプリ設定にあり、プロジェクト状態にはないからです。

簡単に言うと

すべてを空のスタートにリセットします — まるでアプリを開いたばかりのように。未保存の作業がまだある場合、アプリは事前に尋ねます。まったく異なるシーンで開始したい場合に使用します。インポートされた画像、SfM 結果、Gaussian Cloud、トレーニング状態は完全にクリアされます。しかし、独自のプリセットはアプリ設定にあるため保持され、シーンに属しません。

Mode メニュー

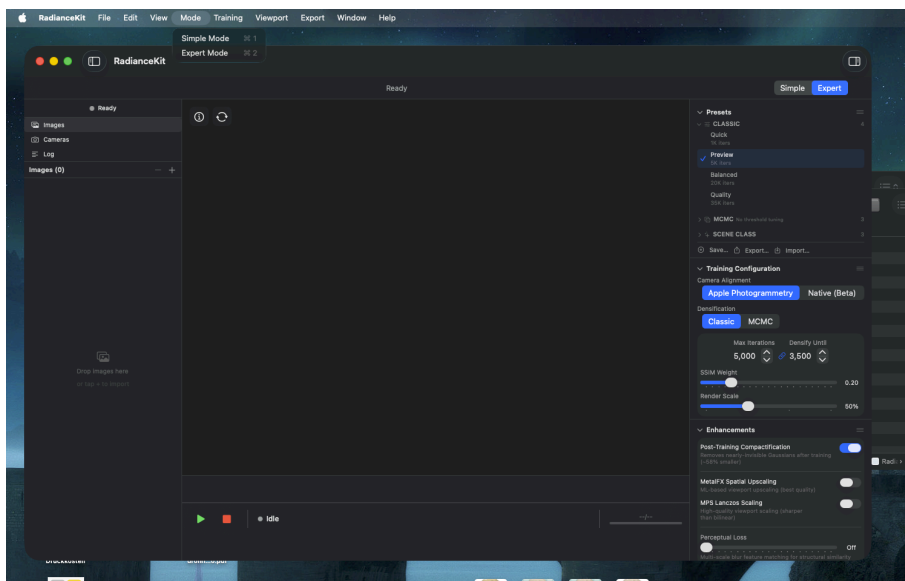


図 2: Mode メニュー、Simple モードと Expert モードのスイッチ付き

ガイド付き Simple モード (ウィザード風、4 ステップ) とフル Expert モード (すべてのコントロールを持つ古典的な Inspector レイアウト) の間のシンプルな 2 つのスイッチです。

M7 Mode > Simple Mode

場所

メニューバー → Mode → Simple Mode (⌘1)。



技術詳細

アプリ状態を Simple モードに切り替えます。アプリのメイン領域は Expert レイアウトの代わりにガイド付きワークフローを表示します。モード状態は設定に保存されます (第3章 Settings の S1 「Default Mode」参照)。

簡単に言うと

インポート、処理、プレビュー、エクスポートにアプリがあなたをガイドするステップバイステップのバリエーションに切り替えます。始めたばかり、または素早く結果が必要な場合に推奨されます。ほとんどの詳細コントロールは非表示で、適切なプリセットで作業します。後でより深く入りたい場合は、Expert モード (M8) に切り替えるだけです。アプリ起動時にどのモードがアクティブかは、設定 (第3章、S1) で設定できます。

M8 Mode > Expert Mode

場所

メニューバー → Mode → Expert Mode (⌘2)。



技術詳細

アプリ状態を Expert モードに切り替えます。これにより、すべてのセクション (Presets、TrainingConfig、Enhancements、Metrics、LossChart、ProjectNavigator) を持つフル Inspector レイアウトが表示されます。Expert モードでは、すべてのトレーニングパラメータ、COLMAP ピッカー、Mid-Compact トグル、診断にアクセスできます。ライブプレビューもこのモードでのみ機能します。

簡単に言うと

すべてのコントロールを持つフルプレビューに切り替えます。ここでは Loss チャートをリアルタイムで見、すべてのパラメータを微調整し、プリセット経由で複数の比較構成を並行して管理できます。トレーニングが内部で何をしているかを理解したい、または的を絞った実験をしたい場合に推奨されます。ライブプレビュー、COLMAP ピッカー、診断もここでのみアクセス可能です。圧倒されていると感じたら、M7 経由で Simple モードに戻ってください — シーンは保持されます。

Training メニュー

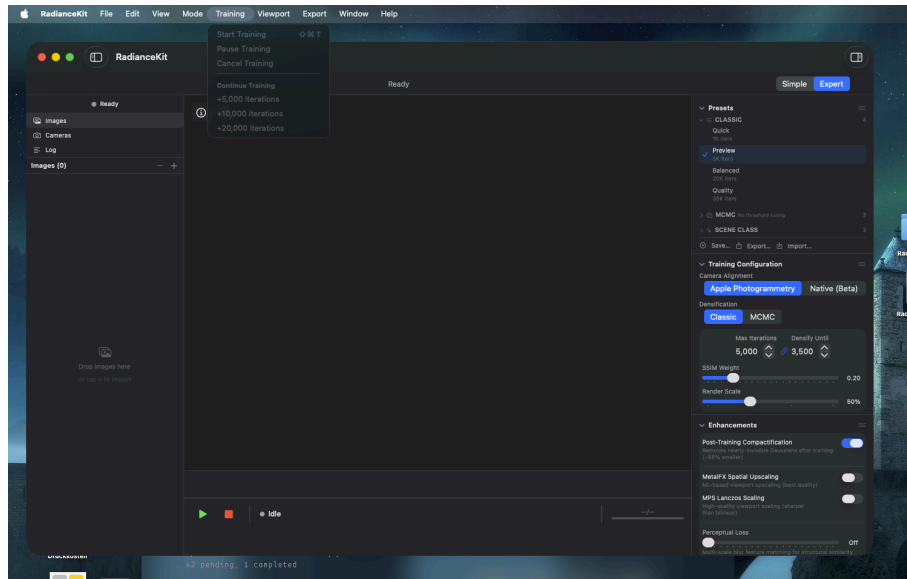


図 3: Continue サブメニュー付き Training メニュー — エントリ M9 から M14

トレーニング実行に関する 4 つのアクション: 開始、一時停止、キャンセル、指定した反復数だけ延長。3 つの Continue エントリはすべて IAP でゲートされています (Free Trial バージョンではクリック不可)。

M9 Training > Start Training



場所

メニューバー → Training → Start Training (⌘T)。



技術詳細

トレーニングパイプラインを非同期で開始します。前提条件: SfM 結果が存在し、他のパイプラインが現在実行されていないこと。両方の条件は、満たされない場合にエントリをブロックします。開始時に、現在の構成値が読み取られ、新しい JSONL ログが `~/Documents/RadianceKit/Logs/training_YYYY-MM-DD_HHmss.jsonl` に作成され、戦略選択に応じてクラシックまたは MCMC パスが実行されます。トレーニング状態は「idle」から「training」に変わります。

簡単に言うと

大きな緑のボタンを押します — 写真をインポートしてカメラ再構築が完了するとすぐに、実際の Gaussian Splatting トレーニングが開始されます。アプリを動作させ続けてください。プリセットに応じて、1分 (Quick) から数時間 (MCMC Quality) かかります。SfM 結果がまだ存在しない、または他のパイプラインが現在実行されている限り、エントリはグレーのままです。各実行は同時に `~/Documents/RadianceKit/Logs/` にログを書き込みます。これは後で Pareto Dashboard (M40) 経由で評価できます。

M10 Training > Pause Training

場所

メニューバー → Training → Pause Training。



技術詳細

実行中のトレーニングを一時停止します。トレーニング状態が「training」の場合にのみ有効になります。一時停止は反復ループを次の安全同期ポイントで停止し、完全な GPU 状態 (Gaussian バッファ、Optimizer モーメント、Scheduler 位置) を保持し、「paused」に切り替えます。再開は再度押すことで行われます (エントリタイトルは静的ですが、アプリは実際のロジックで Pause/Resume を切り替えます)。一時停止されたトレーニングはアプリ終了を生き残りません。その場合は代わりにシーンを保存し、後で Continue トレーニングエントリ (M12~M14) で拡張します。

簡単に言うと

進行を失うことなく、トレーニングを短期間停止します。コンピューターをより重要なことに短期間必要な場合に便利です。もう一度クリックすると再開します。アプリ再起動を超えて機能しません — 本当に後で続行したい場合は、Cancel (M11) でトレーニングを終了し、Save Scene (M2) でシーンを保存し、その後 Continue Training (M12~M14) を使用してください。一時停止中、GPU は完全に休止します。ただし、メモリは占有されたままです。

M11 Training > Cancel Training

場所

メニューバー → Training → Cancel Training。



技術詳細

実行中のトレーニングを中止します。トレーニング状態が「idle」でない場合にアクティブです。トレーニングエンジンのキャンセルフラグを設定します。これは次の同期ポイントで反復ループをクリーンに終了し、最終サマリーエントリを JSONL ログに書き込み、状態を「idle」にリセットします。これまでトレーニングされた Cloud は保持されます (保存またはエクスポート可能) が、「cancelled」とマークされます。

簡単に言うと

実行中のトレーニングを最終的に中止します。これまでの状態は保持されます — 数千の反復後にすでに見せられる結果がある場合、それでもエクスポートできます。短期間だけ中断したい場合は、代わりに Pause (M10) を使用してください。トレーニングログでは、実行が「cancelled」とマークされ、最終 Loss 値は同様に書き込まれます。中止されたシーンは、アプリがその間に終了されていない限り、Continue Training (M12~M14) 経由で後で続行することもできます。

M12 Training > Continue Training > +5 000 iterations

場所

メニューバー → Training → Continue Training → +5,000 iterations。



技術詳細

トレーニングを 5 000 反復続行します。完了したトレーニングが続行可能で、フルバージョンがアンロックされている場合にアクティブです。続行可能性は、完了したトレーニングが存在し、完全な Optimizer 状態がまだメモリにある場合に有効です。Continue では、Adam モーメントと LR スケジューラが続行され、続行は新規開始の代わりに、連続的な 25K/45K/60K 実行のように動作します。JSONL ログは、増分セットアップを含む新しい構成エントリを取得します。フルバージョンでのみ利用可能。

簡単に言うと

さらに 5 000 のトレーニングステップを追加します。最初の実行後に結果が近いがまだ完全に鮮明でない場合に使用してください。有料のフルバージョンでのみ機能します。完全に新しい実行とは異なり、Optimizer の状態は保持されるため、続行は連続的な実行のように感じられます。5 000 ステップ以上必要な場合は、直接 M13 (+10 000) または M14 (+20 000) を使用してください。

M13 Training > Continue Training > +10 000 iterations

場所

メニューバー → Training → Continue Training → +10,000 iterations。



技術詳細

M12 と同じですが、10 000 の追加反復を持ちます。同じ前提条件、同じ LR スケジューラパス。初期トレーニングが中位のプリセットで実行され、実行を完全に新規開始することなく重要な品質向上を見たい場合に推奨されます。

簡単に言うと

トレーニングを 10 000 ステップ延長します — 利用可能な 3 つの Continue 値の中間値。最初の実行は問題なかったが、明らかにより良くしたい場合に良い選択です。M12 と M14 と同様に、新規開始の代わりに学習率の進行がシームレスに続行されます。フルバージョンでのみ利用可能。

M14 Training > Continue Training > +20 000 iterations

場所

メニューバー → Training → Continue Training → +20,000 iterations。

技術詳細

M12 / M13 と同じですが、20 000 の追加反復を持ちます。最大の指定された Continue ジャンプ。MCMC トレーニングでは、これは「フィット」と「ベンチマーク 対応」の違いをもたらすことが多いです。Classic では 35~40K から、経験的にほとんど追加されません。

簡単に言うと

さらに 20 000 のトレーニングステップを追加します、最大の Continue 値。本当に品質の最後の一滴を絞り出したい場合に使用してください。クラシックトレーニングでは 40 000 ステップ後に大した違いはありません—MCMC では逆に、収束が遅れて始まるため、しばしば価値がありません。シーンに応じて顕著な追加実行時間を見込んでください。M12 と M13 と同様に、このエントリもフルバージョンでのみ利用可能です。

Viewport メニュー

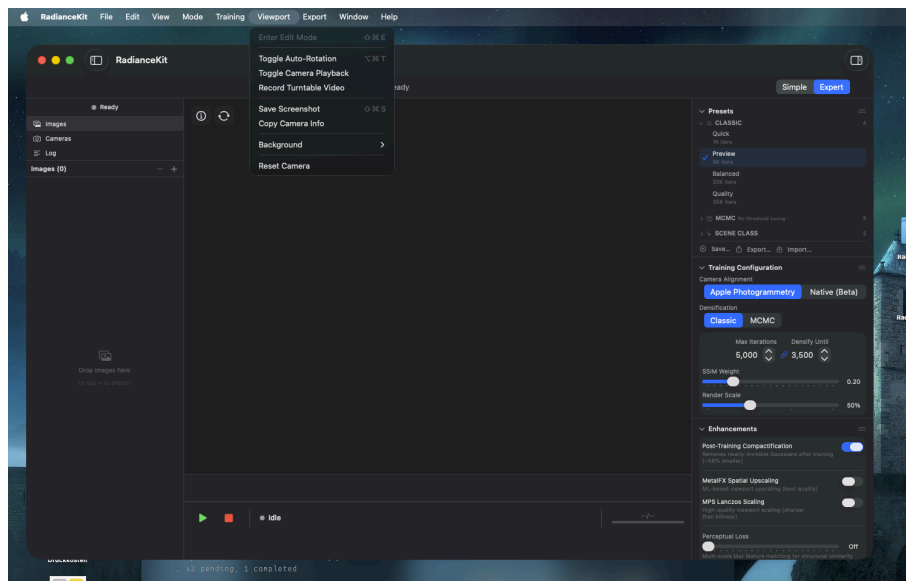


図 4: Edit モード、カメラコントロール、Background サブメニュー付き Viewport メニュー

3D ビューポートを制御します: Gaussian 選択とクリーンアップ用の Edit モード、カメラコントロール (自動回転、再生、録画)、スクリーンショット、背景色、リセット。

M15 Viewport > Enter/Exit Edit Mode

場所

メニューバー → Viewport → Enter Edit Mode (または「Exit Edit Mode」、状態に応じて)。⌘E。



技術詳細

エンタイトルは動的で、状態に応じて「Exit Edit Mode」または「Enter Edit Mode」を表示します。押すと、Edit モードが Viewport レンダラーで切り替わります。Edit モードを終了するとき、現在の選択も追加でリセットされます。Edit モードは、Gaussian のクリック選択、Box 選択、マークされた Gaussian の削除を有効にします (UI の Editor 領域参照)。Viewport レンダラーが接続されていない限り無効化されます。

簡単に言うと

通常の 3D ビューと、個々の Gaussian (例えば Floater や背景の外れ値) をマークして削除できる編集モードを切り替えます。終了時、選択は自動的にリセットされます。Viewport にまだシーンが表示されていない限り、エンタイトルはグレーのままです。ラベルは状態に応じて「Enter Edit Mode」と「Exit Edit Mode」の間で切り替わります — したがって、現在のモードかが常にわかります。

M16 Viewport > Toggle Auto-Rotation

場所

メニューバー → Viewport → Toggle Auto-Rotation (⌘T)。



技術詳細

シーン中心を通る垂直軸を中心としたビューポートカメラの連続回転を オン/オフします。軸と速度はカメラコントロール構成から取得されます。Auto-Rotation は 純粋なビューポート効果で、トレーニングや録画には影響しません — Turntable Video レコーダー (M18) を並行して使用する場合、Auto-Rotation はレコーダーがキャプチャする正確なパスを提供します。

簡単に言うと

マウスをドラッグせずにすべての側面からシーンを見られるように、カメラをゆつくりと連続的にシーンの周りに回転させます。もう一度クリックすると回転が停止します。完成したトレーニング済みシーンを評価したり、ライブデモのバックグラウンドアニメーションとして実用的です。並行してビデオを録画する (M18) 場合、Auto-Rotation はレコーダーがキャプチャする正確な動きを提供します。

M17 Viewport > Toggle Camera Playback

場所

メニューバー → Viewport → Toggle Camera Playback。



技術詳細

カメラパス再生を切り替えます。記録されたカメラパス (例えば以前の録画から、または `transforms.json` が読み込まれたため) が存在する場合、パスが実行されます — ビューポートカメラはマウス/トラックパッド入力で移動しなくなり、フレームごとに軌跡を再現します。再度押すと再生が一時停止します。

簡単に言うと

以前に記録またはインポートされたカメラ走行を実行します。これにより、シーンが記録された元のパスをたどったり、ビデオエクスポート前に計画された **Orbit** の動きを確認したりできます。再生中、マウスとトラックパッド入力は無効化されます — カメラは厳密にパスに従います。再度クリックすると再生が一時停止します。カメラパスを読み込んだり記録したりしていない場合、何も起こりません。

M18 Viewport > Record Turntable Video

場所

メニューバー → Viewport → Record Turntable Video。



技術詳細

ビューポート録画を切り替えます。最初に押すと一時パスへのフレーム録画が開始されます。2回目に押すと録画が終了され、エンコードされて MP4 パスに書き込まれます (パスは保存ダイアログ経由で尋ねられます)。Export → Media → Orbit Video (M31) が固定 360° パスを設定可能な長さで生成するのは異なり、Turntable レコーダーはビューポートで見ているものをライブで録画します — したがって、手動カメラ走行も録画できます。

簡単に言うと

Viewport で直接ビデオを録画します。カメラが自動的に回転するか、マウスで自分で動かすかに関係なく — 見るすべてが MP4 ファイルに保存されます。「**Orbit Video**」エクスポート (M31) とは異なり、カメラ走行を自分で指定します。最初のクリックで録画が開始され、2回目のクリックで終了され、保存場所が尋ねられません。例えば、固い **Orbit** の動きでは不可能な特定の詳細パンを示したい場合に便利です。

M19 Viewport > Save Screenshot

場所

メニューバー → Viewport → Save Screenshot (⌘S)。



技術詳細

フルレンダリング解像度の単一ビューポートフレーム (つまり、ウィンドウ ピクセルレイアウトではなく、フルレンダーターゲットコンテンツ) を PNG ファイルとしてキャプチャします。パスは保存ダイアログ経由で尋ねられます。背景色 (M21~M23) は焼き込まれます。Enhancements (I27/I28 参照) の MetalFX/MPS アップスケーリング 設定は、アクティブな場合に影響します — スクリーンショットはアップスケールされた出力を表示します。

簡単に言うと

現在の 3D ビューのスナップショットを PNG 画像として保存します。マーケティング資料や素早い比較に実用的です。注意: 背景は画像の一部です — 透明度が必要な場合は、シーンファイルをエクスポートしてください。解像度はウィンドウサイズではなく内部レンダーターゲットに対応します — 画像はしばしばウィンドウで見るよりも鮮明です。アップスケーリング設定 (Inspector → Enhancements) も含まれます。

M20 Viewport > Copy Camera Info

場所

メニューバー → Viewport → Copy Camera Info。



技術詳細

現在のビューポートカメラのポーズ (位置、Look-At ポイント、Up ベクトル) と FOV 値をカメラコントロールから読み取り、クリップボードに複数行テキストとして書き込みます。形式は人間が読める形式 (行ごとにラベル = 値) で、JSON ではありません。デバッグ目的で特定のビューを再現したり、サポートと共有したりするのに 実用的です。

簡単に言うと

現在のカメラ位置と視線方向をテキストとしてクリップボードにコピーします。例えば、共同開発者にシーンのどの位置から奇妙に見えるかを示したい場合、テキストをメールやチャットウィンドウに簡単に貼り付けます。形式は人間が読める形式 (値ごとに 1 行) で、JSON ではありません。主にバグレポートやサポートリクエスト用に意図されています。

M21 Viewport > Background > Dark Gray

場所

メニューバー → Viewport → Background → Dark Gray。



技術詳細

ビューポート背景色を暗いグレー (RGB 0.1/0.1/0.1) に設定します。レンダラーは、Gaussian がコンポジットされる前の背景としてこの色を使用します。アプリ起動時のデフォルト色は、Settings オプション S3 「Default Viewport Background」で制御されます。

簡単に言うと

3D ビューポートの背景を暗いグレーに着色します。ほとんどのシーン用の標準選択 — 明るい Gaussian と暗い Gaussian の両方に対して良いコントラストを提供し、目が純粋な黒や白の表面に固定されません。色はスクリーンショット (M19) と Orbit ビデオ (M31) にも引き継がれます。Dark Gray が地味すぎる場合は、比較のため Black (M22) または White (M23) も試してください。アプリ起動時にどの色がアクティブかは、設定 (S3) で設定できます。

M22 Viewport > Background > Black

場所

メニューバー → Viewport → Background → Black。



技術詳細

ビューポート背景色を純粋な黒 (RGB 0/0/0) に設定します。シーンに多くの明るい Floater があり、それらを識別したい場合、または暗いルックアンドフィールのマーケティング資料に役立ちます。

簡単に言うと

黒い背景。非常に明るいシーンに適しており、または Edit モードを見て、グレーで埋もれる小さな明るい Gaussian (Floater) を探したい場合。暗くドラマチックなルックのマーケティング資料にも理想的です。色はスクリーンショットと Orbit ビデオに焼き込まれます — 後でコンポジットするための透明度が必要な場合、黒は最悪の選択です。暗い Floater には、反対方向の White (M23) に再度切り替えてください。

M23 Viewport > Background > White

メニューバー → Viewport → Background → White。



ビューポート背景色を純粋な白 (RGB 1/1/1) に設定します。シーンが主に暗いコンテンツを持ち、暗い Floater (典型的な屋外背景ノイズ) を見たい場合に便利です。

簡単に言うと

白い背景。被写体が明るい背景でより良く表示される、または後で **Edit** モード (M15) で削除できる暗い外れ値を見つけるために実用的です。屋外シーンでは、典型的な屋外 Floater がむしろ暗いため、白がしばしば黒より便利です。他の背景 オプションと同様に、色はスクリーンショットとビデオに引き継がれます。

M24 Viewport > Reset Camera

メニューバー → Viewport → Reset Camera。



ビューポートカメラをリセットし、Training Camera ビューを離れ、Auto-Rotation を停止します。これにより、カメラは初期位置 (典型的: シーンの前、わずかに上から見下ろす) に戻り、Auto-Rotation はオフになり、レンダラーが現在 Training Camera (SfM ポーズの 1 つ) を表示していた場合、Free Camera に戻ります。

簡単に言うと

ビューポートカメラを開始位置に戻します。回転中に迷ったり、シーンを画像から押し出したりした場合 — ここを一度クリックすると、見るべきものが再び見えます。同時に **Auto-Rotation** がオフになり (実行中の場合)、固定された **Training** カメラから自由ビューに戻ります。これにより、どのような場合でも、ビューのクリーンな再起動が得られます。

Export メニュー

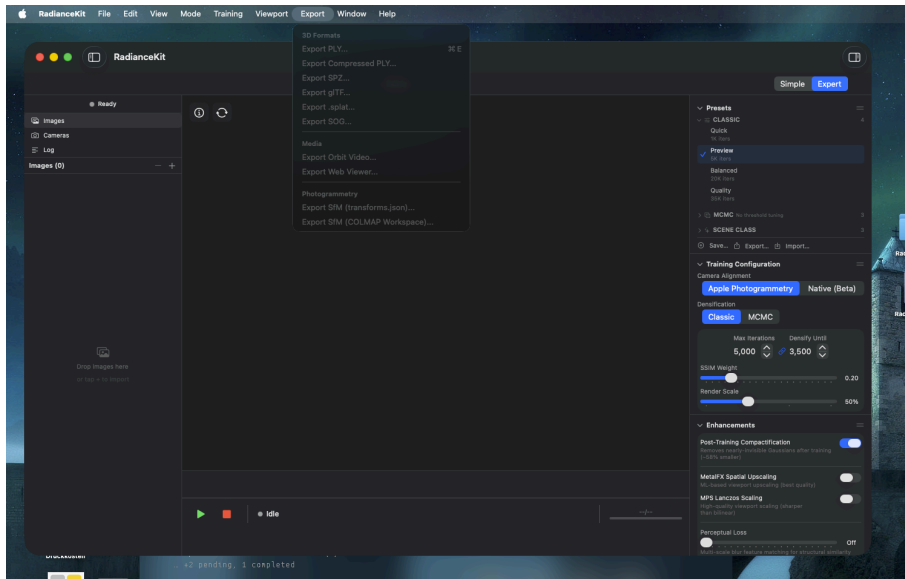


図 5: 3つのサブメニューグループ付き Export メニュー — 3D Formats、Media、Photogrammetry

3つのセクション (3D Formats、Media、Photogrammetry) にグループ化された 8つのエクスポート先と 2つの Photogrammetry エクスポート。最初の 6つは、それぞれ保存ダイアログを開いて形式カタログでエクスポートを登録する共通ヘルパールーチンを紹介して構築されます。Photogrammetry エントリは個別のロジックを持ちます。すべての Photogrammetry エクスポートと一部の 3D エクスポートは、フルバージョンでのみ利用可能です。

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

場所

メニューバー → Export → 3D Formats → PLY (⌘E)。

技術詳細

事前指定ファイル名 `gaussians.ply` を持つ保存ダイアログを開きます。OK で、現在の Gaussian Cloud が標準化された ASCII/Binary PLY 形式に書き込まれます — SuperSplat、PolyCam、PlayCanvas、すべての一般的な 3DGS ビューアーと互換性があります。フル SH 係数、フル精度 (フィールドごとに Float32)。≥ 500K Gaussian でファイルサイズはしばしば数百 MB。

簡単に言うと

3D シーンを標準 PLY ファイルとして保存します。これは最も普遍的な形式です — SuperSplat, PolyCam, PlayCanvas など、ほぼすべてのソフトウェアがこれを読み込みます。しかし、ファイルは大きくなり、しばしば数百メガバイトになります。フル品質で続行またはアーカイブしたい場合は PLY を使用してください。Web 経由でシーンを共有したい場合は、代わりに SPZ (M27) または Compressed PLY (M26) をご覧ください — これらは大幅に小さいです。

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

場所

メニューバー → Export → 3D Formats →
Compressed PLY。



技術詳細

位置、スケール、回転、SH フィールドのカスタム量子化を持つ Compressed PLY 形式で Gaussian Cloud を書き込みます。圧縮されていない PLY (M25) より 5~10 倍小さいファイル、最小限の視覚的損失。SuperSplat (Compressed PLY 標準を読み込む) と PlayCanvas と互換性があります。標準ファイル名 `gaussians_compressed.ply`。

簡単に言うと

通常の PLY と同じですが、5~10 倍小さくなります。品質はほぼ同じままです。オンラインで共有したり、メールで送信したりするためのファイルが必要な場合に使用してください。SuperSplat と PlayCanvas で直接機能します。ただし、ターゲットシステムがさらに小さいファイルを必要とする場合 (モバイル、ブラウザデモ)、代わりに SPZ (M27) を使用してください — これはさらに積極的に圧縮されます。フル編集品質には、圧縮されていない PLY (M25) を使用してください。

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

場所

メニューバー → Export → 3D Formats → SPZ。



技術詳細

SPZ 形式で Gaussian Cloud を書き込みます — Niantic がリリースした、積極的な量子化を持つ圧縮された Splat 形式 (圧縮されていない PLY より ~90% 小さい)。主に Web ビューアーとモバイルアプリ用に最適化されています。Niantic Splatt3R、gsplat.js、Niantic ブラウザビューアーと互換性があります。

簡単に言うと

最小の形式の1つ。通常の PLY より約 10 倍小さい。特にブラウザでシーンを表示する、または携帯アプリで表示したい場合に使用してください。最大品質には PLY がより良い選択です。SPZ は Niantic が開発し、gsplat.js、Splatt3R、Niantic Web ビューアーで直接機能します。強い圧縮のため、SPZ ファイルをそのまま再トレーニングすることはできません — 編集には PLY を使用してください。

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

場所

メニューバー → Export → 3D Formats → glTF。



技術詳細

KHR_gaussian_splatting 拡張付きの `.glb` ファイル (Binary glTF) を書き込みます。標準準拠で、`Babylon.js` や `Three.js` などの glTF エンジンを使用し、`KHR_gaussian_splatting` 拡張を実装するパイプラインに適しています。

簡単に言うと

多くの 3D プログラムや Web エンジンが理解する glTF 形式でシーンを保存します — ただし、`Gaussian Splatting` 拡張をサポートする場合に限りです。特定の 3D パイプライン (例えば `Three.js` または `Babylon.js`) がそれを理解する場合、これはあなたの形式です。ファイルはバイナリ `.glb` として出力されます — すべてを含む単一のパッケージ。古典的な `Splatting` ワークフローには、より多くのツールがそれを直接理解するため、通常 `PLY` や `SPZ` がより良い選択です。

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

場所

メニューバー → Export → 3D Formats → .splat。



技術詳細

`Antimatter15` `.splat` 形式を書き込みます — `Gaussian` あたり固定サイズ 32 バイト (位置は $3 \times \text{Float32}$ 、スケールは $3 \times \text{Float32}$ 、回転は $4 \times \text{Uint8}$ 正規化 クォータニオン、`RGB+Opacity` は $4 \times \text{Uint8}$)。DC より高い SH 係数はありません。ブラウザの直接互換性を持つ最小ファイル。 `gsplat.js` と `antimatter15` のオンライン デモビューアー用。

簡単に言うと

最もシンプルな Web ビューアー形式。小さく、すぐにどのブラウザでも表示可能。ただし、ディテール照明を失います (より高い SH 係数が失われます — 光に反応する代わりに、`Splat` はすべての視点から同じに見えます)。最大の Web パフォーマンスには良いですが、フォトリアリズムには `SPZ` または `PLY` の方が適しています。`antimatter15` オンラインビューアーと `gsplat.js` で機能します。各 `Gaussian` は固定 32 バイトを占有し、形式をシンプルかつ互換性のあるものにします — ただし、ディテールの深さを犠牲にして。

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

場所

メニューバー → Export → 3D Formats → SOG。



技術詳細

SOG 形式で Gaussian Cloud を書き込みます。SOG (「Self-Organizing Gaussian」) は、Texture Atlas レイアウトと量子化されたデータの WebP 圧縮を持つ PlayCanvas 形式です。PLY より 15~20 倍優れたサイズ比率でスケールリングします。エクスポートは内部で `cwebp` を外部ツールとして呼び出します — そのため、Sandbox バリエーション (App Store) では潜在的に制限されます。

簡単に言うと

PlayCanvas ワークフロー用の非常に小さい形式。データが Texture Atlas レイアウトにパックされて WebP 圧縮されるため、PLY より約 15~20 倍小さくなります。PlayCanvas ワークフローがない場合は、SPZ または Compressed PLY が通常より良い選択です。エクスポートは内部で `cwebp` を外部ツールとして呼び出します — App Store バージョン (Sandbox) では、このステップが制限される可能性があります。

M31 Export > Media > Export Orbit Video...

場所

メニューバー → Export → Media → Orbit Video。



技術詳細

シーン中心の周りに 360° Orbit をレンダリングし、MP4 (H.264) または MOV (HEVC、システムデフォルトによる) としてエンコードします。M18 (ライブ録画) と異なり、パスはここで固定されています — 長さは Settings または Simple モード エクスポートステップで選択されます。

簡単に言うと

シーンの周りに自動的に回転ビデオを生成します。手動移動は不要です。ソーシャルメディアや素早いデモに適しています。カメラを自分で制御したい場合は、代わりに Record Turntable Video (M18) を使用してください。パスは固定: シーン中心の周りに完全な 360° Orbit、長さは設定または Simple モードエクスポート ステップで選択します。ビデオはシステムに応じて H.264-MP4 または HEVC-MOV として出力されます。

M32 Export > Media > Export Web Viewer...

場所

メニューバー → Export → Media → Web Viewer。



技術詳細

スタンドアロン HTML ビューアー (gsplat.js ベース) と base64 エンコードされた Gaussian データを単一の `.html` ファイルにパッケージ化します。このファイルは最新のブラウザでオフラインで実行されます — サーバー依存性なし、外部 URL なし。ファイルサイズは SPZ バリエーションの約 1.3 倍 (base64 オーバーヘッドのため)。

簡単に言うと

シーンを自己起動 Web ページとして保存します。HTML ファイルをダブルクリック → ブラウザが開く → 完成したインタラクティブな 3D シーン。インターネットなしで機能し、メールで送信可能で、結果を友人やクライアントと共有する最も簡単な方法です。ファイルには完全な `gsplat.js` ビューアーと Gaussian データが単一のドキュメントに含まれています — Web から何もリロードされません。ファイルサイズは SPZ エクスポートの約 1/3 大きいです。受信者側で追加のソフトウェアは不要です。

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

場所

メニューバー → Export → Photogrammetry → Export SfM (transforms.json)。



技術詳細

独自のエクスポートパス (共通ヘルパールーチンを介していません)。Gaussian Cloud ではなく SfM 結果がエクスポートされるためです。事前指定として `transforms.json` とコンテンツタイプ `json` を持つ保存ダイアログを開きます。OK で、カメラ内部パラメータ、ポーズ (NeRF 規約の 4x4 マトリクスとして)、フレームパスを持つ `nerfstudio` 互換 `transforms.json` が書き込まれます。UI のヘルプテキストは、トレーニング画像を兄弟フォルダ `images/` として一緒にコピーする必要があることを指摘しています。SfM 結果が存在し、フルバージョンがアンロックされている場合にのみアクティブです。

簡単に言うと

SfM 結果を `nerfstudio`, `Brush`, `gsplat`, `OpenSplat` などの別のソフトウェアで再利用したい場合、ここでカメラ位置をエクスポートします。トレーニング画像を `transforms.json` ファイルの横の `images/` フォルダに追加で配置してください — そうしないと、ターゲットプログラムは画像を割り当てられません。SfM 結果がまだ存在しない限り、エントリはグレーアウトされ、`Free Trial` バージョンでロックされます。`COLMAP Workspace` ワークフローには、代わりに M34 を使用してください。

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



場所

メニューバー → Export → Photogrammetry → Export SfM (COLMAP Workspace)。



技術詳細

事前指定名 `colmap-workspace` (フォルダのため拡張子なし) を持つ保存ダイアログを開きます。 `sparse/0/cameras.bin`、 `images.bin`、 `points3D.bin` を持つ標準 COLMAP Workspace を書き込みます。 RadianceKit で計算またはインポートされた SfM 再構築を Postshot、Nerfstudio、Meshroom などの他のツールで開いたり、A/B 再実行で RadianceKit 自体に既に計算された入力として (M5 経由で) 再度読み込んだりすることができます — 計算時間を節約します。 SfM 結果が存在し、フルバージョンがアンロックされている場合にのみアクティブです。

簡単に言うと

M33 と同じですが、nerfstudio の代わりに COLMAP 形式です。 Postshot, Meshroom, Nerfstudio, または COLMAP ワークフローを持つ他のツールを使用する場合、これがあなたのエクスポートです。 実用的な副作用: このフォルダを後で M5 経由で RadianceKit に再度読み込んで、次の実行で SfM 計算時間を節約できます — 特に大きなシーンでは数時間の時間節約。 M33 と同様に、SfM 結果が存在し、Free Trial バージョンでロックされていない場合にのみ利用可能です。

Help メニュー

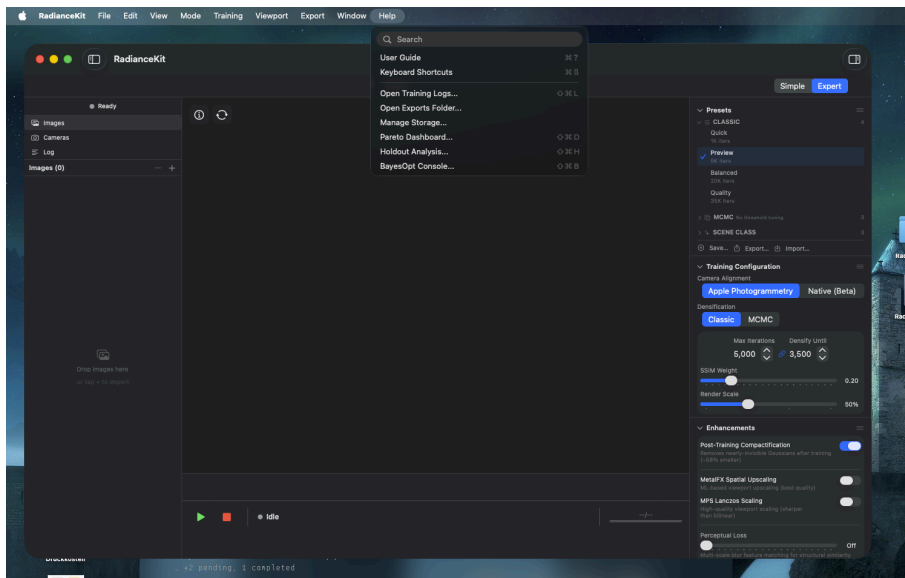


図 6: ドキュメント、フォルダ、分析エントリ付き Help メニュー

7つのエントリ: 2つのドキュメントウィンドウ (User Guide、Keyboard Shortcuts)、3つのフォルダショートカット (Training Logs、Exports、Storage)、3つの分析ウィンドウ (Pareto Dashboard、Holdout Analysis、BayesOpt Console)。Apple 風に、Help メニューは右端に表示されます。標準 Help メニューは、RadianceKit 独自のバリエーションで完全に置き換えられています。

M35 Help > User Guide

メニューバー → Help → User Guide (⌘?)。



User Guide ウィンドウを開きます。トピックサイドバーとスクロール詳細領域を持つナビゲーションを、デフォルトサイズ 860×640 で表示します。コンテンツは静的に保存されています (Markdown から解析されていません)。

簡単に言うと

アプリ内部のガイドを開きます。このマニュアル全体を読みたくない場合、そこでプログラム内に最も重要なステップが見つかります。ガイドは、トピックサイドバーを持つ独自のウィンドウとして構築されています。個々のトピックに的を絞ってジャンプできます。コンテンツはこのハンドブックより短く、最も一般的なワークフローに焦点を当てています。

M36 Help > Keyboard Shortcuts

メニューバー → Help → Keyboard Shortcuts (⌘/)



Keyboard Shortcuts ウィンドウを開きます — すべてのアプリショートカットをトップレベルメニューごとにグループ化したシンプルなスクロールレイアウト。デフォルト サイズ 440×560。コンテンツも静的に保存されています。

簡単に言うと

すべてのキーボードショートカットの完全なリストを持つウィンドウを開きます。例えば、どのキーでトレーニングを開始するか思い出せない場合、そこを見てください。要約はこの章の最後にもあります。リストはトップレベルメニューごとにグループ化されており、適切なエリアに素早くジャンプできます。マウススタイルからキーボードスタイルに移行している場合に役立ちます。

M37 Help > Open Training Logs...

場所

メニューバー → Help → Open Training Logs...
(⇧⌘L)。



技術詳細

ログフォルダを `~/Documents/RadianceKit/Logs` として計算し、必要に応じて作成し、Finder で開きます。各トレーニング実行は、独自の JSONL ファイル `training_YYYY-MM-DD_HHmms.jsonl` をそこに書き込みます。

簡単に言うと

これまでのトレーニングプロトコルすべてを含むフォルダを Finder で開きます。何かがうまくいかなかった、またはトレーニングが正確にいつどの値に収束したかを確認したい場合、ここで JSONL ファイルとして見つけられます。トレーニング実行ごとに、タイムスタンプ付きで正確に1つのファイルが作成されます — これを他のツールに読み込んだり、サポートにメール送信したりすることもできます。グラフィカルな評価が必要な場合は、**Pareto Dashboard (M40)** がより良い入り口です。

M38 Help > Open Exports Folder...

場所

メニューバー → Help → Open Exports Folder...



技術詳細

M37 と類似していますが、`~/Documents/RadianceKit/Exports` を使用します。最初の Auto-Test 実行または最初のクリックで作成されます。その後、すべての Auto-Test エクスポートの標準パス (例えば `autotest_<timestamp>.ply`) がそこに配置されます。保存ダイアログ経由で手動で選択されたエクスポートは、必ずしもここに行くのではなく、ユーザーが保存する場所に行きます — そのため、このフォルダは主に Auto-Test に興味深いです。

簡単に言うと

アプリが独自のエクスポートを配置するフォルダ (主に **Auto-Test** 実行) を開きます。保存ダイアログで手動でエクスポートを別の場所に置いた場合、それはそこにあり、このフォルダにはありません。クリーンアップしたり、以前のテストエクスポートがどれくらいのスペースを占有しているかを確認したりするのに実用的です。ログとシーンバンドルを含む完全な概要が必要な場合は、代わりに **Manage Storage (M39)** を使用してください。

M39 Help > Manage Storage...

場所

メニューバー → Help → Manage Storage...



技術詳細

Storage ブラウザを開きます (第 4 章 Auxiliary Windows、IDs W7~W12 参照)。~/Documents/RadianceKit/ フォルダ内のすべての永続化されたシーン、トレーニング ログ、エクスポート、キャッシュをサイズとともにリストし、エントリごとに Reveal-in-Finder と Move-to-Trash を可能にします。

簡単に言うと

ディスク上で RadianceKit がどれくらいのスペースを占有しているかを — シーン、ログ、エクスポートごとに — 示すウィンドウ ブラウザを開きます。Finder に移動することなく、個別のものを直接削除できます。長時間使用后、ディスクがいっぱいになるときに実用的です — 以前のログと Auto-Test エクスポートは数ギガバイトに合計される可能性があります。Reveal-in-Finder 経由で、古典的なビューにいつでもアクセスできます。

M40 Help > Pareto Dashboard...

場所

メニューバー → Help → Pareto Dashboard... (⇧⌘D)。



技術詳細

Pareto Dashboard を開きます (第 4 章、IDs W13~W22 参照)。Dashboard は ~/Documents/RadianceKit/Logs/ からすべての JSONL トレーニングログを読み込み、シーンとプリセットごとにソートし、Pareto Scatter Plot (デフォルト: Loss vs Gaussian、オプションで Loss vs Wallclock または PSNR vs 反復) を描画します。

簡単に言うと

これまでのトレーニング実行すべての概要を図として開きます。どの実行が品質とサイズの最良のバランスを提供したかがすぐにわかります。さまざまなプリセットを互いに比較したい場合に実用的です。デフォルトでは、図は Loss 対 Gaussian 数を表示します — ただし、Wallclock 時間または PSNR に切り替えることもできます。データは JSONL トレーニングログ (M37) から来ます。実行が多いほど、評価がより意味のあるものになります。

M41 Help > Holdout Analysis...

場所

メニューバー → Help → Holdout Analysis... (⇧⌘H)。



技術詳細

Holdout 分析ウィンドウを開きます (第 4 章、IDs W23~W29 参照)。 `transforms.json` を読み込み、カメラを 3D Globe として描画し、Train/Test fold 分割 (角度または線形、2~8 fold) を許可します。出力は `fold-assignment.json` で、トレーニングはそれぞれのトレーニング構成でテストセットとして使用できます。

簡単に言うと

カメラ撮影をトレーニングセットとテストセットに分割するのに役立ちます — シーンの品質を客観的に測定できます (トレーニングが見ていない画像で)。むしろ研究およびベンチマークツールです。カメラは 3D Globe として表示されます。2 から 8 fold の間で、角度で均等または順序にわたって線形に選択できます。結果は小さな JSON ファイルで、トレーニングはこれをテストセットとして使用します。

M42 Help > BayesOpt Console...

場所

メニューバー → Help → BayesOpt Console... (⇧⌘B)。



技術詳細

BayesOpt Console を開きます (第 4 章、IDs W30~W39 参照)。事前定義された 検索空間 (例えば「MCMC scale-reg + opacity-reg + ssim」) を読み込み、ベイズ最適化トライアルを非同期で実行し、収束曲線とトライアルログをライブで表示します。

簡単に言うと

組み込み自動チューナーコンソール。さまざまなパラメータを手動で試す代わりに、アプリは夜通し自分で実行し、最後にシーンに最適な値を提案できます。非常に高度なツール — ほとんどのワークフローでは良いプリセット (第 7 章参照) で十分です。事前定義された検索空間 (例えば「MCMC scale-reg + opacity-reg + ssim」) を選択し、収束曲線とトライアルログをライブで見ます。セットアップに応じて、数時間 から数日を計画してください。

注意: Edit メニューの Cmd-Z

2026 年 5 月以降、Expert モードの Project Navigator は、マイナスポタンまたはバックスペースキーによるインポート画像の削除、および `Cmd-Z` 経由の取り消しをサポートします。この `Cmd-Z` アクションは、削除された画像がまだ復元可能である限り、macOS Edit メニュー (SwiftUI から提供) に「Undo Remove Image」として表示されます。これは標準標準の `NSUndoManager` システム経由で登録され、`RadianceKitApp` にはありません。したがって、インベントリに独自の M-ID エントリはありません。

キーボードショートカット概要

| メニュー項目 | ショートカット |
|---|---------|
| 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 メニュー (システム提供、Project Navigator 選択がアクティブな Expert モード):

| アクション | ショートカット |
|-----------------------|--------------------|
| Undo Remove Image | ⌘Z |
| Remove Selected Image | Backspace / Delete |

章

第2章 — Inspector (エキスパートビュー)

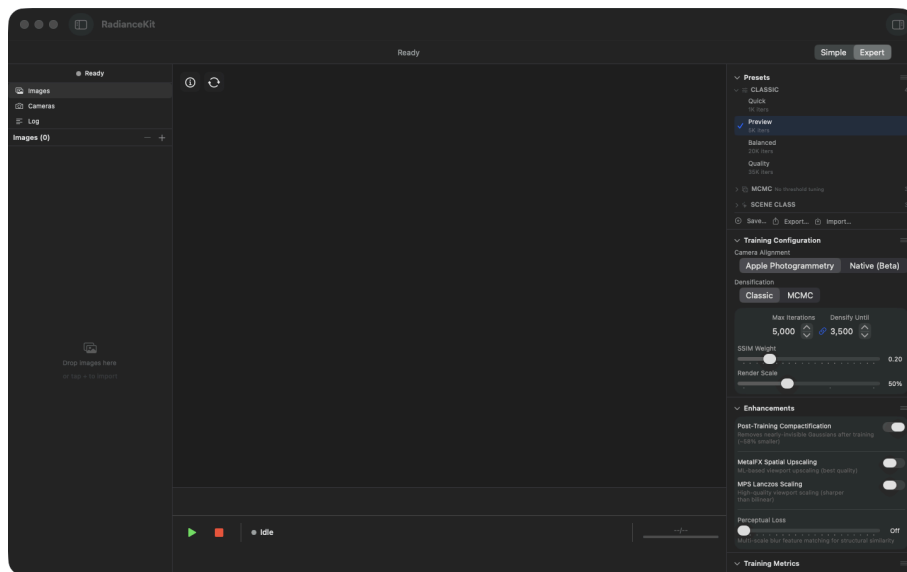


図7: エキスパートモード空 — 左に Project Navigator (Images 0、Cameras、Log)、中央に空のビューポート、右に Inspector (Presets/Training Configuration/Enhancements/Training Metrics セクション)

インポート前の空の Inspector: 左サイドバーには Images カウンター 0 と、ドロップヒント「Drop images here / or tap + to import」が表示されます。右の Inspector は完全に機能しますが、プリセットは情報提供のみ (アクティブなトレーニングはありません)。デフォルトプリセット「Preview」(5K 反復) がマークされています。Camera-Alignment は Apple Photogrammetry、Densification は Classic、SSIM Weight 0.20、Render Scale 50 %。Training Metrics の空状態 (「Start training to see live metrics」) と Loss History (「Loss curve will appear during training」) も表示されます。

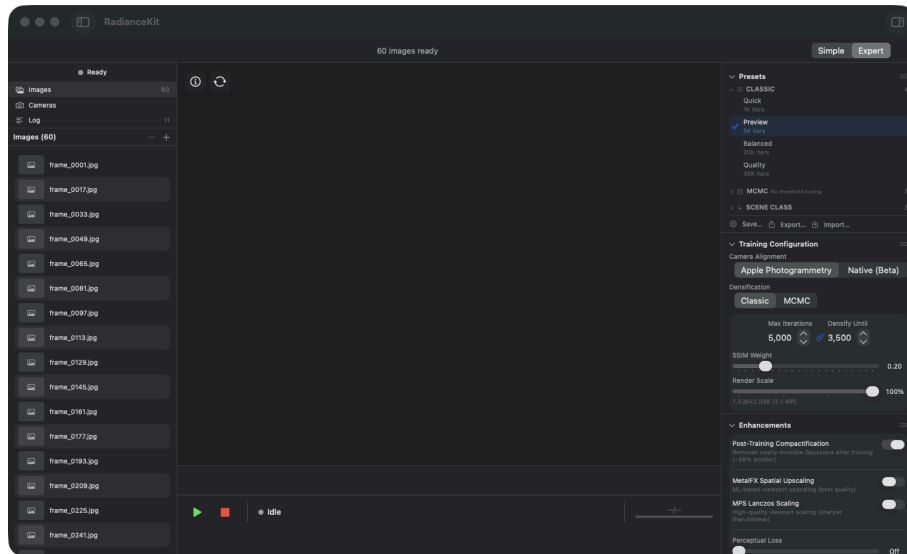


図 8: 60 枚の flowers 画像をロードした Inspector — Image サイドバーに 最初のファイル名 `frame_0001.jpg` ff、ヘッダー「60 images ready」

インポート後の Inspector: ヘッダーステータス「60 images ready」。Image サイドバーにインポートされた 60 枚のフレームすべて (`frame_0001.jpg` から `frame_0945.jpg` まで、960 カメラのブーケデータセットから 16 フレームごとのサブセットを高速反復用にしたもの) がリストされます。Auto-Render-Scale ロジックは 画像解像度 ($1536 \times 2048 = 3.1 \text{ MP}$) を確認し、それに応じて Render Scale を調整します。Play ボタン (緑色、左下) はアクティブになり、アクティブなプリセットでトレーニングを開始します。

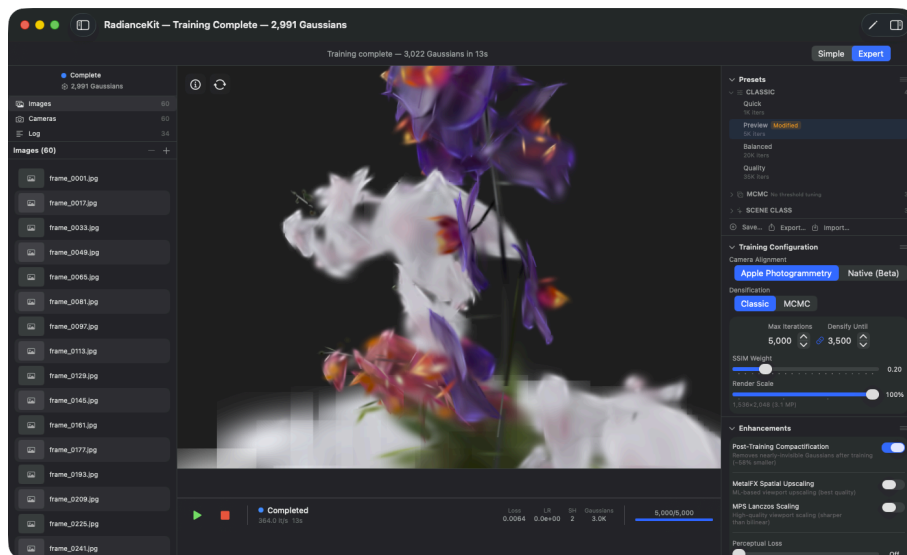


図 9: トレーニング中の Inspector — ライブビューポートに flowers ブーケの再構築、下のメトリックバー (Loss / LR / Gaussian カウント / 反復)、パラメータが調整された場合「Modified」バッジ付きの「Preview」プリセットカード

トレーニング中の Inspector: タイトルバーには全体進捗「RadianceKit — Training NN %」が表示されます。Viewport は実行中の Gaussian 再構築をリアルタイムでレンダリングします (50 反復ごとに更新 — Live Preview 間隔は Settings → General → Training → Live Preview で調整可能)。Viewport の下のメトリックバー: 現在の Loss、Learning

Rate、Gaussian カウント、反復カウンター (例えば Preview プリセットで 1,600/5,000)。Inspector プリセットカード「Preview」は、組み込みデフォルトからパラメータが逸脱するとすぐに「Modified」バッジが付きます。サイドバー「Log」は SfM とトレーニングステージのイベントを収集します。

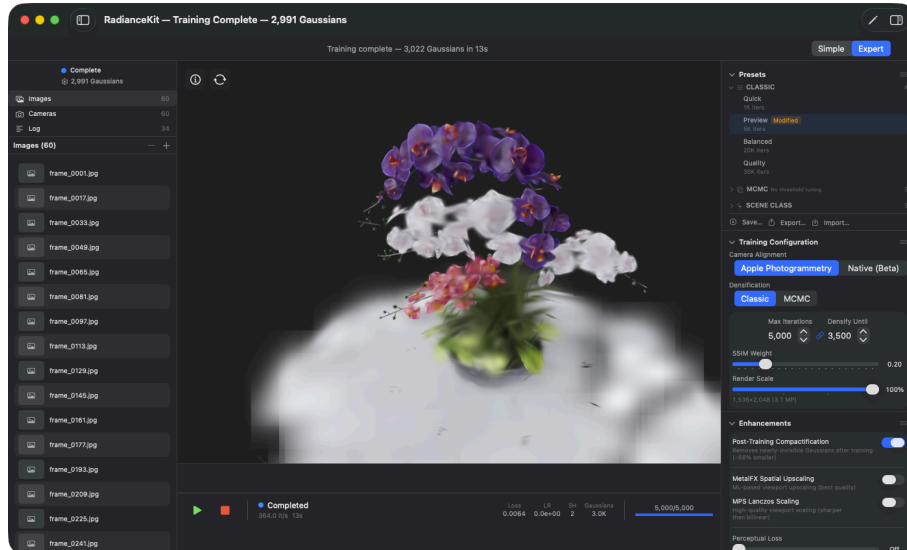


図 10: トレーニング完了後の Inspector — Viewport に完成した flowers ブーケ再構築 (5K 反復後 2,991 Gaussian、13 秒)、タイトルバー「Training Complete — 2,991 Gaussians」

トレーニング後の Inspector: タイトルバーには最終 Gaussian 数 (ここでは 2 991 — 非常にコンパクト。合成 Blender ブーケシーンが明るい背景上の単純な形状を持つため) が表示されます。Viewport は完成した点群を表示します — オービット型ドラッグナビゲーションがアクティブです (シーン中心の周りに回転)。Training Metrics セクションは最終値で満たされ、Loss History チャートは 5 000 反復全体の経過を示します。下のエクスポートセクションがアクティブになります (すべての形式ボタンが有効化)。

Inspector はエキスパートモード (⌘2) の右サイドバーです。すべてのトレーニング関連パラメータを 7 つの折りたたみ可能なセクションにまとめます。初回起動時の上から下へのデフォルト順序は次のとおりです: Look、Presets、トレーニング構成、メトリック、Loss チャート、Enhancements、Export。「Look」セクション (トレーニング後の画像調整) は、以前の「Finishing」セクションの実際の UI 上の改称です — その内部 Enum の `rawValue` は永続性の理由から「Finishing」のままで、表示される見出しは「Look」です。各セクションはヘッダーをクリックして折りたたみ、順序はドラッグ & ドロップで並べ替え可能です (InspectorView.swift:81-97)。**初回起動時はすべての 7 つのセクションが折りたたまれています** (inspectorCollapsedSections のデフォルトは `Set(InspectorSection.allCases)`)。アプリ状態はその後、折りたたみと順序の設定をアプリ起動を超えて保存します。

Inspector の操作要素の一部は、設定 (第 3 章) にもほぼ同じ形で表示されます — 典型的には SfM バックエンド、Sky-Masking、および同様のデフォルトです。分離は意図的です: 設定は新しいプロジェクト用のアプリ全体テンプレートを提供し、Inspector は現在開いているプロジェクトのこれらの値を上書きします。一度どちらかの操作ロジックを知れば、もう一方は目をつぶって使えます。

エキスパートモードの左カラム — Project Navigator — は Inspector の一部ではありませんが、その直接の隣にあります。そこではインポートされた画像をクリックで選択し、スペースキーでクイックルックで表示し、マイナスボタンまたは Delete キーで削除 (Cmd-Z で取り消し可能) できます。Inspector は現在のサイドバー選択に従って コンテキスト固有の詳細情報を表示しますが、7つのメインセクションは常に 利用可能です。

Look セクション (L1~L5)

Look セクション (内部 `rawValue` は引き続き「Finishing」) は最上部の Inspector セクションで、**トレーニング後の**画像調整を一箇所に 集約します。すべてのスライダーは**非破壊的に**動作します: 各スライダーは `FinishingPass` を変更されていない Pristine スナップショット (オリジナルの DC カラー、Opacity、スケーリング) に対して改めて適用します — この調整はしたがって**冪等 (idempotent)** であり、累積しません。結果は **Viewport にライブ**で (WYSIWYG、後のエクスポートとまったく同じ) 表示され、**すべてのエクスポートに焼き込まれます**。このセクションは **トレーニング実行の完了後にのみ**利用可能です (それ以前は「Available after a training run completes.」と表示)。その値は **新しいトレーニングのたびにリセット**されます。エクスポートが実行されている間、すべてのスライダーは**ロック**されます — ロックヒント「Locked while exporting — the file uses the current settings.」が表示され、GroupBox は無効化されます。

L1 Saturation スライダー



場所

Inspector → Look セクション → GroupBox → Saturation。



技術詳細

スライダー 0.5~1.2、表示は小数 2 桁 (例えば「1.00」)。各 Splat の SH-DC クロマを輝度値の周りでスケーリングします: 1.0 = 変更なし、< 1.0 = 彩度を下げる (色をグレースケールへ引き寄せる)、> 1.0 = より鮮やか。数学的には DC カラーは Pristine スナップショットから逆算されるため (`desaturateDC`)、繰り返しスライドしても 累積しません。DJI ドローン素材 (Pensford 高架橋) で検証済みで、これは色が過剰になりがちです — ドローンのデフォルトは 0.82 です。色の基礎 (SH 次数 0) にのみ作用し、より高次の SH 係数は手付かずのままです。

簡単に言うと

完成した Splat の色がどれだけ鮮やかかを決めます。1.00 はすべてを トレーニングされたままにし、それ未満の値は色をグレー方向へ引き寄せます — しばしば過飽和で出力されるドローンやビデオ素材に最適です。1.0 を超える値はより鮮やかにします。アプリは常に変更されていないオリジナルの状態から計算し直すため、何か「増幅」することなく自由に行き来できます。Viewport にライブで表示され、エクスポートでもまったく同じです。

L2 Splat length スライダー



場所

Inspector → Look セクション → GroupBox → Splat length。



技術詳細

スライダー 0.3~1.0、表示は小数2桁。各 Gaussian の3つのスケール軸を対数空間で平均値へ引き寄せます (shortenScale、係数 alpha): 1.0 = 変更なし、より小さい値は細長い「針」状の Splat をより丸くし、0 で純粋な球になります。針状の過度に引き伸ばされた Splat に作用し、全体のサイズを変えずに典型的な「コンフェッティ」アーティファクトを減らします。Pristine スナップショット (オリジナルの対数スケール) から適用されるため冪等です。両方とも対数空間で動作するため Splat size (L3) と可換です。

簡単に言うと

長すぎる、ささくれ立った Splat をより丸くします。1.00 は形状をトレーニングされたままにし、より低い値は引き伸ばされた「針」をより丸い斑点に押し縮めます — これがコンフェッティアーティファクトに悩まされた粒状の再構築を落ち着かせます。全体のサイズは同じままで、長さだけが対象です。Splat size (L3) と安全に組み合わせられます。

L3 Splat size スライダー



場所

Inspector → Look セクション → GroupBox → Splat size。



技術詳細

スライダー 0.5~2.0、表示は小数2桁。各 Gaussian を **3つすべての軸で一様にスケール**します (sizeScale): 1.0 = 変更なし、< 1.0 = より小さく/密に/鮮明に、> 1.0 = より大きく/「ふわふわ」に (Splat 間の隙間を埋める)。スケールは対数空間にあるため、乗算は加算的な $\log(\text{factor})$ オフセットとして実現されます — 一定のオフセットは平均からの偏差を手付けずのままにするため、これは Splat length (L2) と可換です。Pristine スナップショットから適用されるため冪等です。このバージョンで新規。

簡単に言うと

すべての Splat を一様により大きくまたは小さくスケールします。1.00 はトレーニングされた状態で、それ未満の値は点群をより密で鮮明にし、それを超える値は Splat 間の隙間を覆います (より柔らかく/「ふわふわ」に見えます)。穴だらけの再構築を視覚的に閉じる、あるいは逆により多くの詳細を引き出すのに便利です。Splat length (L2) と問題なく共存します — 両方のスライダーは互いに影響しません。

L4 Fade far region (サブスライダー付き)



場所

Inspector → Look セクション → GroupBox → トグル「Fade far region」 およびサブスライダー「Fade start xradius」と「Fade floor」。



技術詳細

カメラ重心からの距離とともに放射状の Opacity 減衰を有効化する トグルです — 背景の弱く観測された「Far コンフェッティ」をフェードアウトさせます。 **オービット撮影専用:** `finishingContext.fadeEligible` が `false` の場合 (直線飛行、カメラが少なすぎるまたは退化している)、トグルは無効化されます。 その場合、サブスライダーの代わりにヒント「Far-fade applies only to orbit captures (not this scene).」が表示されます。 適格性はカメラ位置の方位角カバレッジで判定されます (オービットは重心を周回し多くのコンパスセクターを埋め、直線飛行は ~2 つだけ)。 2 つのサブスライダーがジオメトリを制御します: **Fade start xradius** (1.0~3.0) はオービット半径の倍数として内側半径を設定し、その内側ではフル Opacity が適用されます。 **Fade floor** (0.0~1.0) はフェード半径のはるか彼方での Opacity 係数です。 **重要:** フェードは **Sky-Dome 領域をスキップします** (インデックス [0, frozenCount) の frozen Gaussian)。これにより意図的な背景ドームと一緒に暗くならないようにします。

簡単に言うと

シーンの外縁にあるぼやけた残りをフェードアウトさせます — まさに全周撮影で遠く後方に浮かぶ「Far コンフェッティ」の塊です。本物のオービット/周回撮影でのみ機能します。直線的なドローン飛行やカメラが少なすぎる場合、スイッチはグレーアウトし、理由を説明するヒントが表示されます。アクティブな場合、2 つの微調整スライダーが追加されます: 「Fade start xradius」はどの距離から (周回半径の倍数として) フェードが始まるかを設定し、「Fade floor」は遠くの Splat が最終的にどれだけ見えたままになるかを設定します (0 = 完全に消える、1 = 変更なし)。意図的に再構築された Sky-Dome (I44) はその際決して手を付けられません — 空は保持されます。

L5 Reset finishing ボタン



場所

Inspector → Look セクション → GroupBox → 「Reset finishing」 (下の 小さいボタン)。



技術詳細

すべての Look 設定をデフォルトにリセットし (FinishingPass.Settings() = Saturation 1.0、 Fade オフ、 Splat length 1.0、 Splat size 1.0)、ただちに再 Finishing をトリガーするため、Viewport は変更されていないトレーニング済みの状態に戻ります。 controlSize(.small)。Look スタック全体が Pristine スナップショット から幕等に計算するため、「デフォルトに戻す」はまさに元のトレーニング出力です — 何度も行き来しても品質劣化はありません。セクションのすべてのスライダーと同様に、実行中のエクスポート中はロックされます。

簡単に言うと

ワンクリックですべての Look スライダーを標準に戻します (Saturation 1.00, Fade オフ、両方の Splat スライダーを 1.00)。その後 Viewport は再び新しくトレーニングされた結果をまったくそのまま表示します。いじりすぎてきれいにゼロから始めたいときに便利です。アプリは常にオリジナルの状態から計算するため、その際に品質劣化はありません。エクスポートが実行されている間、ボタンは (スライダーと同様に) ロックされます。

Presets セクション (I1~I11)

Presets セクションは、テストされた構成を適用する最も速い方法です。組み込み プリセット (Capture Class、Classic、MCMC、Hybrid) は、560+ の文書化された実験から再現可能な 開始点を提供します。独自のプリセットは保存、エクスポート、インポート、共有が可能です。リストはカテゴリ別 (Capture Class、Classic、MCMC、Hybrid、Custom) にグループ化され、複数のカテゴリを同時に展開できます。コンテキストメニュー機構 (行を右クリック) を介して、Export、Duplicate、— 独自プリセットの場合 — Delete にアクセスできます。

I1 Save... ボタン

場所

Inspector → Presets セクション → Save... ボタン (下のアクションバー)。



技術詳細

テキストフィールドと Save/Cancel ボタンを持つポップオーバーを開きます。現在の TrainingConfig 状態が新しいユーザー定義プリセットとして永続化されます (JSON エンコード、アプリ全体に保存)。Save 操作は、現在の Densification 戦略を含む 81 のトレーニングパラメータすべてをコピーします。プリセットは、派生元の 組み込みプリセットに関係なく、自動的に Custom カテゴリに配置されます。空の名前と 純粋な空白入力は破棄されます。すでに存在する名前は拒否されません — 各プリセットには独自の内部 ID があり、重複名は技術的に許可されていますが、実際には混乱を招きます。

簡単に言うと

現在の構成を再利用可能なプリセットとして保存します。ボタンを押して、ポップオーバーに名前を入力し、Save をクリックします — Densification 戦略を含む 81 のパラメータすべてが、選択された名前が Custom カテゴリに配置されます。苦勞して、次のプロジェクトで再びゼロから調整したくない場合に必要です。「ドローン 4K」や「室内 高速」などの繰り返しのセットアップに特に実用的です。重複名は技術的に許可されていますが、実際には混乱を招きます — 何か分かりやすい名前を選んでください。

I2 Preset Name テキストフィールド

場所

Save ポップオーバー → テキストフィールド「Preset Name」。



技術詳細

角丸境界線を持つシンプルなテキストフィールド、幅広い形状。値は Save ボタンクリック時にプリセット名として引き継がれます。UI には長さ制限はありませんが、保存される名前は JSON エンコード可能で、UI リストで表示可能である必要があります — 絵文字とウムラウトは機能します。コンテンツは ポップオーバーが開かれるときに自動的に空の文字列にリセットされます。Save ボタンは、トリム後にフィールドが空である限り無効です。自動候補もアクティブなプリセット名の 事前入力はありません。

簡単に言うと

ここでプリセットの名前を入力します。Custom カテゴリで後で見つけやすいように、「ドローン 4K 30fps」や「室内 高速」など何か分かりやすい名前を選んでください。絵文字とウムラウトが許可されており、厳しい長さ制限はありません。フィールドが空、または空白のみで構成されている限り、Save ボタンはグレーアウト されます。ポップオーバーを再度開くとフィールドは再び空になります — アクティブなプリセット名による事前入力は ありません。

I3 Cancel ボタン (Save ダイアログ)



場所

Save ポップオーバー → Cancel ボタン (左)。



技術詳細

保存せずにポップオーバーを閉じます。テキストフィールドのコンテンツを破棄します — 次回開くと、Save...ボタンロジック (I1) で再び空にリセットされます。標準ボタンスタイル、確認ダイアログなし、ホットキーなし。Save パスが実行されなかったため、現在の TrainingConfig は変更されません。

簡単に言うと

何も保存せずに Save ポップオーバーを閉じます。考え直したり、入力ミスをしたり、ダイアログを誤って開いた場合 — 単純に Cancel をクリックしてください。何もまだ書き込まれていないため、現在のトレーニング構成は変更されません。ポップオーバーを再度開くと、名前フィールドは再び空で開始されます。確認なし、ホットキーなし — クリックして消えます。

I4 Save ボタン (Save ダイアログ)



場所

Save ポップオーバー → Save ボタン (右、目立つスタイル)。



技術詳細

実際の永続化をトリガーします。空でない名前を再度検証 (防御的チェック) してから、現在の TrainingConfig を JSON としてアプリストレージに書き込みます。その後ポップオーバーを閉じます。青で強調表示され、テキストフィールドが空である限りグレーアウトされます。保存が失敗した場合 (例えばアプリストレージがいっぱい — 非常にまれ)、現在表示されるエラーダイアログはありません。プリセットは次回アプリ起動時に単に表示されなくなります。

簡単に言うと

Save をクリックすると、名前が引き継がれ、現在のセットアップが新しいプリセットとして書き出されます。ポップオーバーが閉じ、プリセットは Preset リストの Custom カテゴリにすぐに表示され、これからクリックでアクティブ化できます。ボタンは青で強調表示され (borderedProminent), 名前フィールドが空である限りグレーアウトされたままです。保存が失敗した場合 (例えば UserDefaults がいっぱい), 表示されるエラーダイアログはありません — プリセットは次回アプリ起動時に単に表示されなくなります。

I5 Export... ボタン



場所

Inspector → Presets セクション → アクションバー → Export... ボタン。



技術詳細

現在選択されているプリセットを `.radiancepreset` ファイル (内部 JSON) として エクスポートします。プリセットが選択されていない場合は無効化されます。クリックすると、アプリは事前指定ファイル名 (プリセット名 + `.radiancepreset` 拡張子) を持つ Save ダイアログを開きます。保存される形式には、完全な TrainingConfig と メタデータ (名前、カテゴリ、ID、組み込みフラグ) が含まれます。Finder でダブルクリックするとアプリが開きますが、自動的にインポートは **しません**。ユーザーは Import ボタン (I6) を使用する必要があります。

簡単に言うと

リストでプリセットを選択して Export をクリックします — その後、`.radiancepreset` ファイルとして保存し、同僚に送信したり、2 台目の Mac に転送したりできます。受信者は Import... ボタン (I6) で再度ロードします。組み込みプリセットと独自の Custom プリセットの両方で同じように機能します。リストで何もクリックされていない限り、ボタンはグレーアウトされます。ヒント: コンテキストメニュー (I8) 経由ではさらに高速です — プリセットを最初に選択する必要はありません。

I6 Import... ボタン



場所

Inspector → Presets セクション → アクションバー → Import... ボタン。



技術詳細

`.radiancepreset` ファイルのみを許可するファイルダイアログを開きます (複数選択無効)。選択時、JSON ファイルが読み込まれ、検証され、Custom カテゴリに新しい内部 ID で挿入されます — 組み込みとの衝突を防ぐためです。インポートは、エクスポートされたプリセットが元々組み込みであった場合でも、カテゴリを自動的に Custom に設定します。破損したファイル、または古いスキーマバージョンと互換性のないファイルは、エラーダイアログなしで暗黙的に拒否されます (ただし コンソールログには情報が表示されます)。

簡単に言うと

ディスクから `.radiancepreset` ファイルを読み込みます。誰かがテスト済みのセットアップを送ってきた場合や、複数の Mac で自分のお気に入りのプリセットを同期したい場合に便利です。インポートされたプリセットは、元々が組み込みからエクスポートされたものであっても、常に Custom カテゴリに配置されます。破損したファイルや古いファイルは暗黙的に無視されます。コンソールログに理由が表示されます。ダイアログでの複数選択は無効化されているため、クリックごとに 1 つのファイルです。

I7 プリセット行 (クリックアクティブ化)



場所

Inspector → Presets セクション → 各カテゴリの各プリセット行。



技術詳細

プリセット行をクリックすると、TrainingConfig のすべてのフィールドがプリセットの値で置き換えられ、アクティブなプリセットの ID が記憶され、Modified ステータスがリセットされます。行の前のアクティブチェックマークは、プリセットが選択されており、変更されていない場合にのみ表示されます。他の Inspector セクションの TrainingConfig の値 (スライダー、ステッパー、トグル) が変更されるとすぐに、名前の上にオレンジの「Modified」バッジが表示されます。組み込みプリセットは上書きできません — 変更の場合は Save ボタン (I1) で独自のコピーを作成する必要があります。

簡単に言うと

行をクリックするとプリセットがアクティブになり、そこに保存されているすべての値が現在のトレーニング設定に引き継がれます。名前の上のチェックマークは、現在どのプリセットがアクティブかを示します。その後、他のセクションでスライダー、ステッパー、トグルを調整するとすぐに、名前の後にオレンジの「Modified」バッジが表示されます — セットアップがプリセットから逸脱したからです。組み込みプリセットは上書きできません。変更を保持したい場合は、Save... ボタン (I1) で独自のコピーを作成するか、プリセットを複製 (I9) してください。

I8 コンテキストメニュー「Export...」



場所

各プリセット行を右クリック → 最初のエントリ「Export...」。



技術詳細

I5 (Export...ボタン) と同じ機能ですが、より便利にアクセスできます — プリセットを事前に選択する必要はありません。行で直接クリックされたプリセットをエクスポートします。すべてのプリセットカテゴリ (組み込みまたは Custom) で同じように機能し、制限はありません。エクスポートには組み込みフラグと元のカテゴリが含まれますが、再インポート時には I6 で説明したようにカテゴリは Custom にマッピングされます。

簡単に言うと

素早くエクスポートする方法 — 希望のプリセットを右クリックして「Export...」を選択します。事前にクリックして Export... ボタンを押す回り道を節約します。すべてのカテゴリで同じように機能し、組み込みでも同じです。生成された `.radiancepreset` ファイルは I5 のものと同じです。後で再インポートすると自動的に Custom カテゴリに配置されます。

I9 コンテキストメニュー「Duplicate」

場所

各プリセット行を右クリック → 2 番目のエントリ「Duplicate」。



技術詳細

プリセットを Custom カテゴリに複製します。新しい内部 ID を生成し、名前に「Copy」を追加し、コピーを保存します。組み込みプリセットでも機能します。クローンは編集可能になります。オリジナルはそのままです。TrainingConfig は値ごとにコピーされ (JSON ラウンドトリップ)、オリジナルとコピー間に参照バイディングは存在しません。

簡単に言うと

Custom カテゴリにプリセットの編集可能なコピーを作成します。例えば、組み込み「Quality」プリセットを出発点として、SSIM スライダーを少しだけ動かしたい場合に実用的です。ワークフロー: 複製、名前変更 (コンテキストメニューまたは新規 Save... 実行)、調整完了。オリジナルはそのままです。いつでも戻れます。組み込みでも機能します。これは、その値をベースとして引き継ぎながら同時に編集可能にする唯一の方法です。

I10 コンテキストメニュー「Delete」

場所

独自プリセット行を右クリック → 最後のエントリ「Delete」(赤、破壊的)。



技術詳細

Custom プリセットでのみ表示されます。組み込みは削除できません。エントリは破壊的とマークされ、コンテキストメニューに赤で表示され、誤ってクリックしないように区切り線の後に配置されています。確認ダイアログは **ありません**。クリックでプリセットがすぐに削除されます。削除されたプリセットは復元できません (Cmd-Z はここでは機能しません。現在のビルドでは Undo は画像リストでのみ機能し、プリセット操作では機能しません)。削除されたプリセットが現在アクティブだった場合、現在の TrainingConfig は変更されず、アクティブなプリセット選択のみがクリアされます。

簡単に言うと

独自プリセットを削除します。組み込み (Quick, Preview, Balanced, Quality, Ultra Detail, Drone / Aerial, 360° Walkaround, Photo / Object など) では、「Delete」がそもそも表示されません。誤って kill することはできません。注意: 確認ダイアログや Undo はなく、クリックでプリセットが消えます。確信がない場合は、事前に Export... (I5/I8) でディスクに安全コピーを保存してください。いつでも再インポートできます。プリセットがアクティブだった場合、TrainingConfig は変更されず、チェックマークのみが消えます。

I11 カテゴリヘッダー (展開/折りたたみ)



場所

Inspector → Presets セクション → 各カテゴリヘッダー (Capture Class、Classic、MCMC、Hybrid、Custom)。



技術詳細

異なるデフォルトを持つカテゴリごとの折りたたみ状態: キュレーションされた Capture Class グループは **展開** で開始し、Classic、MCMC、Hybrid、Custom は **折りたたみ** で開始します。状態は永続化されません — アプリ再起動時にすべてのカテゴリは デフォルト状態に戻ります。Chevron 矢印はアニメーション付きで回転します。ヘッダーの右側の数字は、このカテゴリのプリセットの数を示します。クリック ヒット領域はヘッダー全体に及びます。

簡単に言うと

プリセットリストを見やすく保つために、カテゴリを展開および折りたたみます。アプリ起動時、Capture Class グループは開いており、Classic、MCMC、Hybrid、Custom は閉じています。ヘッダー (全領域がクリック可能) をクリックすると、リストが短い Chevron アニメーションで開閉します。右側の小さな数字は、カテゴリに含まれるプリセットの数を示します。アプリの再起動後、デフォルト状態が再び表示されます — アプリはこの折りたたみ設定を意図的に保存しません。

トレーニング構成セクション (I12~I22)

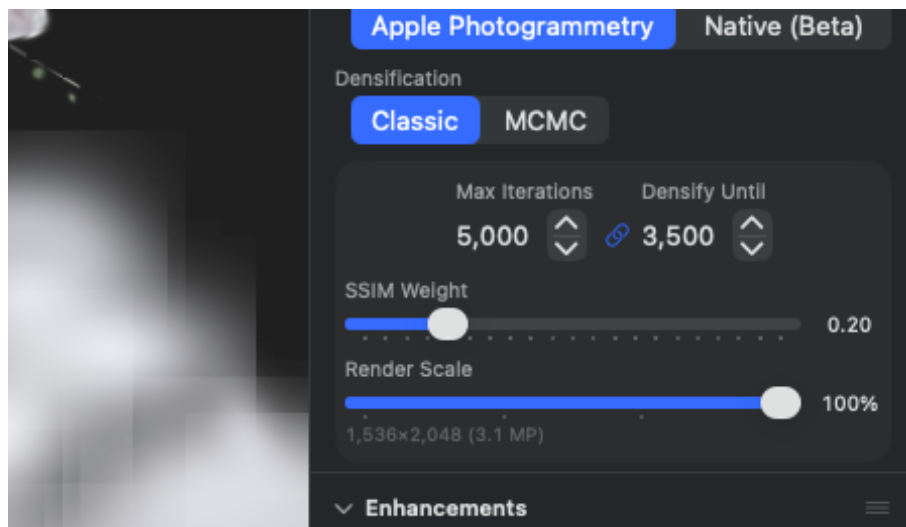


図 11: トレーニング構成セクションのみのクローズ — Camera Alignment (Apple Photogrammetry アクティブ、Native (Beta) 非アクティブ)、Densification (Classic アクティブ)、Link シンボル付き Max Iterations 5,000 / Densify Until 3,500、SSIM Weight スライダー 0.20、Render Scale スライダー 100% (1,536×2,048 = 3.1 MP)

ここに中心的な制御要素があります: どの SfM バックエンドが計算するか、Densification の動作、反復数、SSIM 重みです。MCMC 戦略では、Classic モードで非表示になる 2 つの追加トグル (「MCMC Quality」と「Auto-scale by scene」) が表示されます。Native SfM バックエンドでは、EXIF 焦点距離のないビデオフレーム用の FOV オーバーライド フィールドが追加されます。

I12 Camera Alignment ピッカー

場所

Inspector → トレーニング構成 → Camera Alignment (上部のセグメントドピッカー)。



技術詳細

2 つのオプションを持つセグメントドピッカー: Apple Photogrammetry と Native (Beta)。選択は次のカメラ再構築で使用される SfM バックエンドを決定します。同時に、他のどの Inspector 要素が表示されるかにも影響します: Native は、EXIF のないビデオフレームでのみ必要な FOV オーバーライド (I13) を追加で表示します。ヒント: 非常に大きな屋外撮影では、外部ツール (Metashape または COLMAP) の結果を Workspace インポート経由で読み込めます — 第 1 章 (M5) と第 9 章 (Q3、Q6) 参照。

簡単に言うと

ここでカメラ位置がどのように再構築されるかを選択します — 最終品質の最も重要なスイッチ。Apple Photogrammetry は高速な標準で、ほとんどのオブジェクトスキャンに完全に十分です。Native (Beta) は App Store 準拠の独自開発で、オービットやターンテーブルシーンに適しており、EXIF のないビデオフレームでは FOV オーバーライド (I13) が必要です。非常に大きな屋外セットでは、Metashape や COLMAP でカメラを代替計算し、結果を Workspace インポート経由でロードできます。シーンタイプ別の詳細と推奨事項は第 9 章にあります。

I13 FOV Override フィールド (Native SfM)

場所

Inspector → トレーニング構成 → FOV Override (Camera Alignment = Native の場合にのみ表示)。



技術詳細

数値テキストフィールド (Range 0-170°)、デフォルト 0 = EXIF または ヒューリスティックからの自動決定。手動入力、焦点距離メタデータを含まないビデオから抽出された入力画像の場合に必要です。典型値: iPhone Wide ≈ 73°、DJI Mavic Wide-Crop ≈ 70°、フルフォーマットセンサー付きドローン ≈ 84°。値は [0, 170] にクランプされます — 範囲外の値は直接戻されます。ネイティブ SfM パイプライン (Q4/Q5) のみに影響します。Apple Photogrammetry はこの値を完全に無視します。

簡単に言うと

画像に EXIF がない場合 (抽出されたビデオフレームに典型的)、ここにカメラの水平視野を度単位で入力します。経験値: iPhone Wide ≈ 73°, DJI Mavic Wide-Crop ≈ 70°, フルフォーマットセンサー付きドローン ≈ 84°。0 はアプリ自身に推測させます — しばしばうまくいきますが、まれなレンズではうまくいかないことがあります。170° を超える値は自動的に戻されます。フィールドは Native を Camera Alignment (I12) として選択した場合にのみ表示され、有効です — Apple Photogrammetry は完全に無視します。

I15 **Densification** ピッカー



場所

Inspector → トレーニング構成 → Densification (セグメンテッドピッカー、常に表示)。



技術詳細

2 つの Densification 戦略間で切り替えます: Classic (オリジナル 3DGS プロシージャ、Clone/Split/Prune と勾配閾値付き) と MCMC (Stochastic Gradient Langevin Dynamics with Relocation、NeurIPS 2024)。Classic から MCMC への切り替え時、アプリは MCMC 固有のフィールドを自動的にテスト済みのデフォルト値 (Reg-Weights = 0、MCMC-Cap-Multiplier 3.0、Sample/Noise スケジュール) に設定します。この自動初期化なしでは、古いプリセットのセッションが 1.4.4 MCMC-Collapse バグ (460K → 5 Gaussian、ウォッチドッグキル) に苦しみました。ピッカーの選択は、表示される Inspector 要素も追加で決定します — MCMC では I16/I17 が表示されます。詳細なフィールド効果は 第 6 章 T11~T16 (Classic) と T61~T73 (MCMC)。

簡単に言うと

Gaussian 数の成長のための中心的な戦略選択。Classic は 459 の実験からよくチューニングされ、高速で高品質な結果を生成し、MCMC フィールドを知る必要はありません。MCMC は新しいアプローチ (NeurIPS 2024) より再現可能で、手動の閾値調整を不要にします — 代わりに、同等の品質で約 6 倍長く計算します。MCMC への切り替え時、アプリは自動的に安全なデフォルトを設定し、トレーニングが 1.4.4 Collapse に陥らないようにします。戦略フィールドの詳細は第 6 章 (T11~T16 Classic, T61~T73 MCMC) にあります。

I16 **MCMC Quality** トグル



場所

Inspector → トレーニング構成 → MCMC Quality (Densification = MCMC の場合のみ)。



技術詳細

勾配累積を 2 ステップ (アクティブ) または 1 ステップ (非アクティブ) に切り替えます。Optimizer ステップが実行される前に、連続する 2 つのカメラビューからの勾配を累積します。経験的に (Session 33、V544a)、最終 L1 エラーが約 6% 減少します (Horse-Full-MCMC の 3 試行平均で Quality 付き 0.0246 vs Quality なし 0.0261)。コスト: 倍のトレーニング時間。非常に長いトレーニング (200K 反復) では、これにより 10+ 分の追加待ち時間が発生します — 最後の数パーセントの品質が本当に必要な場合にのみ値があります。トレーニングのみに影響し、エクスポート形式や Viewport 表示には影響しません。

簡単に言うと

2 ビューにわたる勾配累積付き MCMC の Quality モード。経験的に最終結果が約 6% 良くなり (Horse テストで L1 0.0261 の代わりに 0.0246)、その代わり倍の時間がかかります。すでに 200K MCMC トレーニング (簡単に 2 時間) を実行している場合、さらに 1 時間弱が追加されます。最終ショーケースレンダリングまたは Quality Sweep セッションの終わりに値がありますが、日々のワークフローではむしろそうではありません。Densification が MCMC (I15) に設定されている場合にのみ表示されます。

I17 Auto-scale by scene トグル

場所

Inspector → トレーニング構成 → Auto-scale by scene (MCMC の場合のみ)。



技術詳細

アクティブな場合、実効最大 Gaussian 上限は SfM 初期点数 × MCMC-Cap Multiplier (デフォルト 3.0) でスケールアップされます。例: SfM は 250K 初期点を提供し、基本上限 = 150K、Multiplier 3.0 → 実効上限 = $\max(150K, 750K) = 750K$ 。非アクティブの場合、基本のみが厳密に適用されます。1000 フレーム以上で対応する高い SfM 点密度を持つ大規模な屋外撮影が、固定 150K キャップデフォルトで Densification を飢えさせていたため、v1.4.5 で導入されました — 余分な点が残りに、新しい点は発生できませんでした。Custom プリセットでデフォルト OFF、MCMC 組み込みで ON。トレーニング時にのみ影響し、エクスポートには影響しません。

簡単に言うと

Gaussian の最大数をシーンサイズに合わせて成長させます (より正確には: SfM 初期点数とともに)。小さなシーンではほとんど違いに気づきませんが、大きな屋外シーンでは品質にしばしば決定的です — そうでなければ、デフォルトの上限 150K がシーンに対して低すぎるため、トレーニングが「窒息」します。非常に大きな屋外セット (1000 フレーム以上) が目に見えてキャップに引っかかった後、v1.4.5 用に特別に導入されました。MCMC 組み込みプリセットではすでに事前に有効化されています。独自のプリセットではデフォルトでオフです。

I18 Max Iterations ステツパー

場所

Inspector → トレーニング構成 → GroupBox → Max Iterations。



技術詳細

Range 1 000~100 000、ステップ幅 1 000 のステツパー。Optimizer 反復の総数を決定します。トレーニング時間と線形に相関 (半減 = 約 50% 時間)。経験的 스위트スポット: 20K (Classic Balanced、 $L1 \approx 0.028$)、40K (Classic Quality、 $L1 \approx 0.023$)、200K (MCMC Full、 $L1 \approx 0.0246$)。40K を超える Classic は平均してほとんど改善をもたらしません — 収益逡減。変更時、Link 機能 (I19) がアクティブな場合、Densify Until が比例して引き寄せられます (デフォルト比: 0.5、つまり Densify-Until = Max/2)。

簡単に言うと

何回のトレーニングステップが実行されるか — 多いほど良いが、線形に時間もかかります。経験則: 良い品質には 20 000, Classic 戦略の最適値には 40 000 (それを超えると平均してほとんど何ももたらしません)。MCMC はかなり多くを必要とし、200 000 がここでの標準です。反復を倍にすると、トレーニング時間がほぼ倍になります。アクティブな Link ボタン (I19) では、Densify Until が比例して引き寄せられます — ほとんど常にあなたが望むものです。

I19 Link/Unlink ボタン (Densify ↔ Iterations)



場所

Inspector → トレーニング構成 → GroupBox → Max Iterations と Densify Until の間の小さな Link ボタン。



技術詳細

Densify Until と Max Iterations の比率を固定するトグルボタン。アクティブ (Link アイコンが強調表示) では、Max Iterations の変更ごとに Densify Until が比例して引き寄せられます。Unlink (Link-Plus アイコン) では、値は独立したままです。これが典型的な相関を反映するため、デフォルトはリンクされています — 反復を倍にすると、通常 Densification も比例して長く実行したいです。比率は Link ボタンの 設定時に現在の値から計算されます。典型的な比率は 0.5 (Densify-Until = 反復数の半分) です。

簡単に言うと

Max Iterations と Densify Until の間の小さなクランプスイッチ。アクティブ (Link アイコンが強調表示) の場合、2つの値は一緒に移動します — Iterations を 倍にすると、Densify Until も同じ比率で倍になります。そうでない場合 (link.badge.plus アイコン), 独立して設定できます。これが典型的な相関を反映するため、標準ではリンクされています — 長いトレーニングは通常、長い Densification 段階も必要とします。99% のケースでは、ロックされたままにしてください。

I20 Densify Until ステツパー



場所

Inspector → トレーニング構成 → GroupBox → Densify Until。



技術詳細

Range 500~50 000、ステップ幅 500 のステツパー。Clone/Split (Classic) や Relocation (MCMC) を介して新しい Gaussian が追加されなくなる反復インデックスを 決定します。到達後、位置と色のみが洗練されます。高い値 = より多くの Gaussian = より大きなファイル、長い反復あたりの時間 (ステップあたり +30~60% GPU 時間)。典型値: 15K (30K Max-Iter の場合)、20K (40K の場合)、100K (200K MCMC の場合)。アクティブな Link (I19) では自動的にスケーリングされます。Classic vs MCMC で異なる効果: Classic は成長を完全に停止し、MCMC は Relocation ロジックを停止しますが、Sample/Noise 適応は続行されます。

簡単に言うと

どの反復まで新しい Gaussian が追加されるか — Classic では Clone/Split で、MCMC では Relocation で。その後は、既存の点の色と形の改善のみ。高い = より多くの詳細、しかしファイルサイズも大きく、ステップあたり +30~60% GPU 時間。典型値: 15K (30K Max-Iter の場合), 20K (40K の場合), 100K (200K MCMC の場合)。通常は Link (I19) 経由で Max Iterations に依存します — 手動で切り離すことはほとんど意味がありません。

I21 SSIM Weight スライダー

場所

Inspector → トレーニング構成 → GroupBox → SSIM Weight.



技術詳細

スライダー 0.0~1.0、0.05 ステップ、「0.20」として表示。L1 Loss (0.0) と SSIM Loss (1.0) をミックスします。L1 はピクセルごとの明るさを引き締め、SSIM は構造的類似性 (エッジ、ローカル統計) を引き締めます。デフォルト 0.2 はオリジナル 3DGS 論文 (Kerbl 2023) の値で、多くのセッションで堅牢な妥協点としてリベース エンジニアリングされました。高い値 (0.5+) は詳細保持を優先しますが、ローカルの明るさエラーを無視する可能性があります。低い値 (< 0.1) は鋭いエッジでの詳細 損失を引き起こします。SSIM 計算はシェーダーで 11x11 ガウシアンウィンドウで実行されます。パフォーマンス: 0.0 (L1 のみ) では、SSIM 計算がシェーダーでスキップされるため、トレーニングは約 8~12% 高速になります。

簡単に言うと

純粋な明るさ比較に対して、構造的画像類似性 (エッジ、ローカルパターン) がどの程度重み付けされるか。0.2 はオリジナル 3DGS 論文の標準で、ほぼすべてのシーンで十分です。髪、毛皮、植生などの細かい構造には高く (0.5+) — そこではより多くの構造重みが役立ちます。低く (0.0) するとトレーニングが約 8~12% 高速になります。シェーダーで SSIM 計算がスキップされるためですが、鋭いエッジでの詳細を犠牲にします。変更する正当な理由がない人は、0.2 のままにしてください。

I22 Render Scale スライダー

場所

Inspector → トレーニング構成 → GroupBox → Render Scale。



技術詳細

スライダー 0.25~1.0、0.25 ステップ、「100%」として表示。ソース画像 サイズに対するトレーニングレンダリング解像度をスケールします。パフォーマンスへの最大のレバー: 50% は GPU 時間を約 75% 削減し (ピクセルが 4 倍少ないため)、25% は約 94% 削減します。Gradient Threshold は自動的にスケールされます。スライダーの下に、ライブ解像度表示が MP で表示されます (例えば 「2304×1296 (3.0 MP)」)。現在の値が推奨値から逸脱している場合、オレンジの文字で 「— recommended: 50%」 が表示されます。推奨は ~3 MP の実効解像度を目指します — Apple Silicon GPU が最も効率的に処理できる領域です。例えば、4K ソース画像には自動的に 25% が推奨され、FullHD 画像には 100% が推奨されます。変更により Buffer の再割り当てもトリガーされます。

簡単に言うと

トレーニングがどの解像度でレンダリングするか — 最大のパフォーマンス レバーの 1 つ。フル (100%) は最高品質を提供しますが、大きな画像では多くの GPU 時間を消費します。半分 (50%) はピクセルが 4 倍少なく計算されるため、約 75% の GPU 時間を節約します — 4K ソースに最適。スライダーの下には、メガピクセル単位の実効解像度が表示されます。アプリは約 3 MP を目指します。これは Apple Silicon で最も効率的に実行されるためです。値がこれから逸脱する場合、アプリはオレンジの「recommended」ヒントを表示します — ほとんどの場合、それに従うことが価値があります。

Enhancements セクション (I26~I29、I42~I44)

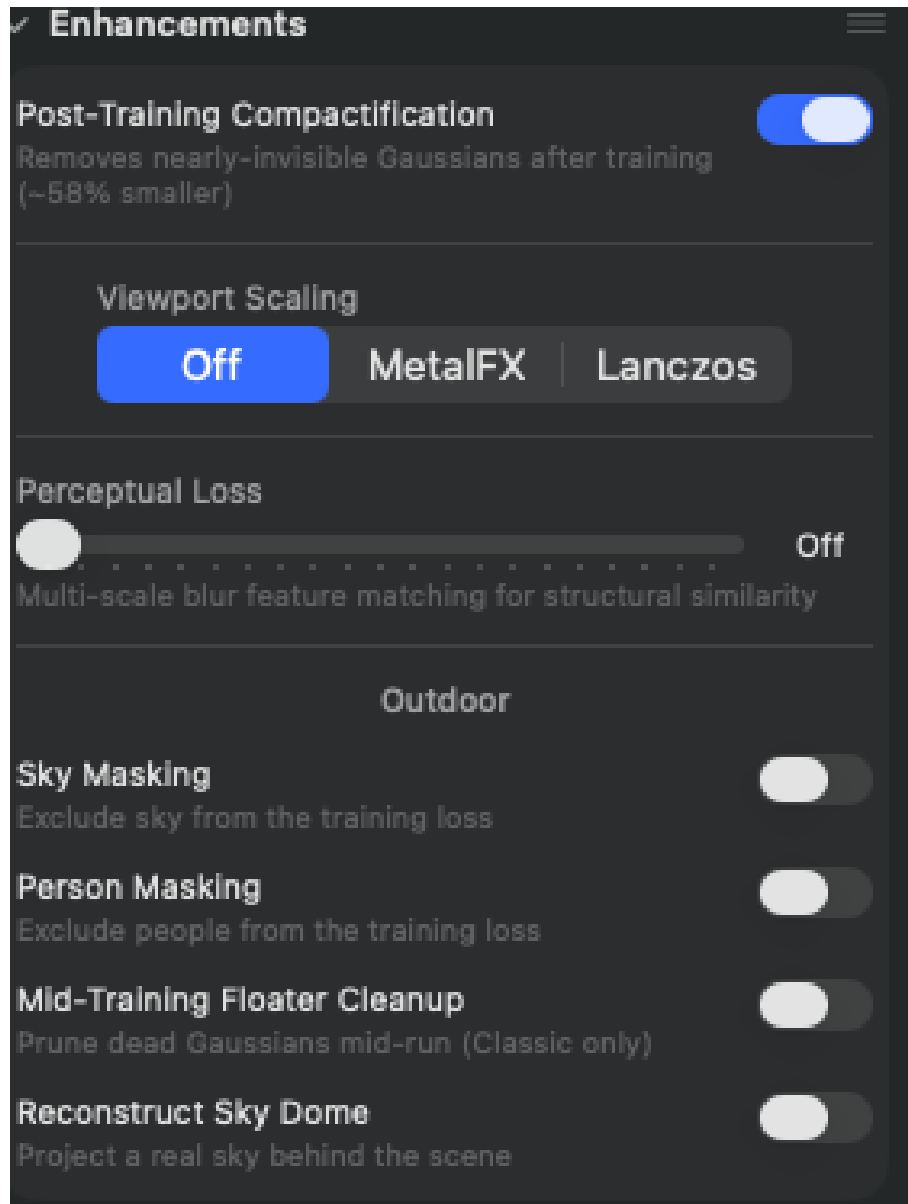


図 12: Enhancements セクションのみのクローズ — 3 行: Post-Training Compactification (トグルオン)、Viewport Scaling (セグメント化ピッカー Off/MetalFX/Lanczos)、Perceptual Loss (スライダー「Off」)。各行にサブタイトルで機能を説明

Enhancements セクションは、コアトレーニンググループ自体を変更することなく画像品質を改善する 3 つの機能をグループ化します。最初の 2 つ (I26-I27) は **ポストトレーニング** または **ビューポートステージ** です: Compactification はトレーニング終了後にクリーンアップし、Viewport Scaling は実行中のトレーニングに影響しない純粋なビューポートレンダラーです。Perceptual Loss (I29) はセクション所属にもかかわらずトレーニングコンポーネントです — トレーニング中に追加の Loss 項として有効化されるため、ビューポートトグルとは区切り線で分離されています。v1.6 以降、このセクションには Outdoor グループ (I42~I44: Sky Masking、Mid-Training Floater Cleanup、Reconstruct Sky Dome) もあり

ます — 以前は 設定ウィンドウにあった、空の Floater に対するトレーニングオプションで、現在はプロジェクトごとにここに置かれています。

I26 Post-Training Compactification トグル



場所

Inspector → Enhancements → Post-Training Compactification。



技術詳細

V443 ポスト処理を有効化します: トレーニング反復完了後、Opacity が 0.01 (1% 可視性) 未満の Gaussian は削除されます。経験的に、ファイルサイズが ~55-58% 削減され、視覚的品質損失はゼロです — これらの Gaussian は視覚的にとにかく寄与しないからです。Compactification は GPU Compact パスとして実行され、Gaussian カウントに応じて秒の何分の 1 か、または数秒かかります。トレーニングパフォーマンスには 影響しません。このトグルがオフの場合、非表示の Gaussian もエクスポートされます — 形式を別のトレーニングステージ (Continue Training) で使用したい場合にのみ関連します。そうでなければストレージの無駄です。

簡単に言うと

トレーニング後、とにかく見えない Gaussian (Opacity 1% 未満) をクリーンアップします。エクスポートファイルを約半分小さくし (~55-58% のサイズ削減), 視覚的品質損失はありません。最後の反復後に短い GPU パスとして実行され、秒の何分の 1 から数秒しかかかりません。実質的に常にオンにすべきです — これをオフにする唯一の理由は、後で Continue Training でトレーニングを続行し、見えない Gaussian も保持する必要がある場合です。通常のエクスポートワークフローでは、そのままにしてください。

I27 Viewport Scaling ピッカー



場所

Inspector → Enhancements → Viewport Scaling (3つのオプションを持つセグメント化ピッカー: Off、MetalFX、Lanczos)。



技術詳細

Viewport アップスケーラーを選択する単一のセグメント化ピッカーです — 3つのオプションは**相互排他的**です。トレーニング解像度 (I22 Render Scale 経由) がビューポートサイズより低い場合、選択されたモードがレンダリングされたフレームを表示サイズにスケールアップします。**Off** = 単純な双線形引き伸ばし。**MetalFX** = Apple の ML ベースの Spatial Upscaler、最も鮮明なオプション (ML モデルは鋭いエッジ用に最適化)、M3 GPU でフレームあたり約 1~2 ms のオーバーヘッド。**Lanczos** = Apple の Metal Performance Shaders に 8 タップ Sinc リサンプリングを使った、ML なしの古典的手法、最小オーバーヘッド (< 0.5ms)、品質は MetalFX を下回りますが、細かい線構造の ML 特有の「平滑化」がありません。レンダラーパイプラインは切り替え時にライブで再構成されます — 再起動なしですぐに表示されます。**背景:** 以前はこれが 2 つの別々のトグル (MetalFX + Lanczos) で、同時にオンにできました — MetalFX が静かに Lanczos を上書きする矛盾した状態でした。ピッカーはこの状態を排除します。古いセッションから継承された可能性のある「両方オン」状態は、次の切り替え時に自動的に MetalFX へ自己修復します。Live Viewport に**のみ**作用し、レンダリングされたエクスポート (Orbit ビデオ、スクリーンショット) には影響しません — これらはフルソース解像度でレンダリングされます。

簡単に言うと

Viewport のライブ画像を鮮明にアップスケールします — 特に低いトレーニング解像度 (Render Scale 50%, I22 参照) で作業している場合に便利です。常に 1 つだけがアクティブになる 3 段階: 「Off」は単純にピクセルを引き伸ばし、「MetalFX」は Apple の機械学習を使い最も鋭いエッジを提供 (実質的に常に最良の選択)、「Lanczos」は ML なしの古典的フィルター — MetalFX があるシーンで線を平滑化したりアーティファクトを示したりする場合のフォールバックとして使ってください。再起動なしでライブに作用します。Live Viewport にのみ作用し、エクスポートされた Orbit ビデオやスクリーンショットには影響しません — これらはフルソース解像度でレンダリングされます。以前と異なり、誤って 2 つのモードを同時に選ぶことはもうできません。

I29 Perceptual Loss スライダー

場所

Inspector → Enhancements → Perceptual Loss。



技術詳細

スライダー 0.0~0.2、0.01 ステップ、0.0 で「Off」として表示、それ以外は「0.05」などとして表示。レンダリングのマルチスケールガウシアンブラーを Ground Truth 画像と比較する追加の Loss 項を有効化します (3 ブラスケール)。L1+SSIM だけでは認識できない構造的差異をキャプチャします。V460 実装。経験的に、0.05~0.1 の値はセッションで L1 スコアを数パーセント改善しますが、~5% のトレーニング時間 (ブラー カーネルを通る追加の順方向パス) を消費します。0.15 を超えると、トレーニングが不安定になり、L1 が再び悪化します (Loss 項が最適化を支配)。「Enhancements」セクション内の位置にもかかわらず、トレーニング **中** に効果があり、後処理では効きません — 純粋な事後アップグレードではありません。

簡単に言うと

3つの異なる不鮮明レベルにわたって構造的画像類似性をチェックする追加の Loss 部分。L1+SSIM だけでは見えないパターンをキャプチャするため、髪、布、植生などの細かい構造を持つシーンに特に役立ちます。小さい値の方が安全です — 0.05~0.1 がスイートスポット、0.15 を超えるとトレーニングが不安定になり、Loss が再び悪化します。0 (Off) では機能は完全にオフで、コストはかかりません。アクティブだと、ブラーカーネルを介する追加の順方向パスのために約 5% のトレーニング時間を消費します。「Enhancements」セクションにもかかわらず、後処理ではなくトレーニング中に直接効果があります。

I42 Sky Masking



場所

Inspector → Enhancements → Outdoor グループ → トグル「Sky Masking」。バインド:

`AppState.trainingConfig.skyMaskingEnabled` (プロジェクトごと、`@DefaultFalse`)。デフォルト:
`false`。



技術詳細

pre-training Apple Vision ベースの空ピクセル セグメンテーションを有効化します。トレーニング開始前に、各入力カメラについて、Apple Vision フォアグラウンドマスク (空 = 背景) を介して空領域が抽出され、該当カメラのピクセルごとのマスクとして関連付けられます。トレーニング中、ピクセルごとの Loss 寄与は空マスクの補集合と乗算されます — 空ピクセルは勾配に 0 を寄与するため、空に投影される Gaussian は最適化信号を受け取らず、したがって「より密」または「より明るく」なりません。屋外/ドローンシーンでの Floater (空の暗い小さな塊) を大幅に削減します。クラシック 40K トレーニングで ~3% L1 リグレッションのコスト ([memory/dev_outdoor-floater-reduction.md](#) 参照)。明確に識別可能な空がある屋外シーンでのみ意味があります。室内シーンや白い背景では、空セグメンテーションは誤った領域を識別し、有効な Loss 信号をブロックします。値は現在プロジェクトごとに保存され (もはやアプリ全体ではありません)、プリセット / シーンファイルに従います。

簡単に言うと

画像内に空がある屋外撮影では、しばしば空に黒や色付きの小さな塊が発生します — いわゆる「Floater」です。このオプションは自動的に空がどこにあるかを検出し、トレーニングに「空をそのままにしておいて」と伝えます。ドローン飛行や風景シーンに非常に適しています。室内や暗い背景では画像が悪化することがあります — したがって、本物の空が見える場合にのみ有効にしてください。詳細: [memory/dev_outdoor-floater-reduction.md](#)。

I43 Mid-Training Floater Cleanup

 場所

Inspector → Enhancements → Outdoor グループ → トグル「Mid-Training Floater Cleanup」。バインド: `AppState.trainingConfig.floaterCleanupEnabled` (プロジェクトごと、`@DefaultFalse`)。デフォルト: `false`。

 技術詳細

クラシック 40K トレーニング (プリセット「P4 Quality」) で 2 つの追加 Density Control パスを有効化します: 反復 20,000 と反復 30,000 で。両パスは 3 つの基準ですべての Gaussian を検索します: (a) 非常に低い Opacity (標準 0.005)、(b) 極小スクリーン空間サイズ、(c) 直近 1000 反復で Loss 寄与なし。3 つの条件すべてを満たす Gaussian は purge されます。効果: トレーニング終了時に ~5~15% 少ない Gaussian、ドローン/屋外シーンの空で暗い小さな塊が目に見えて減少。クローズアップ室内シーンで ~1~3% L1 リグレーションのコストのため、デフォルトでは有効になりません。値は再起動を超えて記憶されます (S7 とは異なります)。2 つのクリーンアップ反復 (20K、30K) はハードコードされており、現在 UI 経由で変更できません。より短いトレーニング (例えば P2 Preview 5K) では、反復マークに到達しないため、トグルは効果がありません。**新規:** このトグルは、アクティブなプリセットが **Classic** デンシファイアを使用する場合 (`densificationStrategy == .classic`) にのみ操作可能になりました。MCMC または Hybrid では **無効化** され、インラインヒントが表示されます。これらの戦略は死んだ Gaussian をそもそも自前で処理するため (MCMC は Relocation 経由、Hybrid は組み合わせた Reloc/Noise ロジック経由)、手動のクリーンアップパスはそこでは効果がない、あるいは逆効果になるからです。コード参照: `RadianceKitApp.swift`、General タブ。詳細: `memory/dev_outdoor-floater-reduction.md`。

 簡単に言うと

トレーニング中、画像品質に貢献しなくなったがメモリを占有する「死んだ」ガウス点が時々発生します。このオプションは長いトレーニング中に 2 回 (20K と 30K 反復で) クリーンアップし、これらの死骸を削除します。そこに最も多くの Floater が集まるため、空のある屋外シーンでは特に意味があります。小さなトレーニングや家具の接写では、必要ない場合が多いです。このスイッチは、プリセットが **Classic** デンシファイアを使用する場合にのみオンにできます — MCMC または Hybrid プリセットではグレーアウトされます (短い説明付き)。それらは自前で死んだ点を片付けるからです。

I 44 Reconstruct Sky Dome

📍 場所

Inspector → Enhancements → Outdoor グループ
→ トグル「Reconstruct Sky Dome」。バインド:
AppState.trainingConfig.skyDomeEnabled (プロジェクトごと、@DefaultFalse)。デフォルト: false。

🔧 技術詳細

pre-training の Sky-Dome 投影を有効化します (V549e MVP)。SfM 後、トレーニング開始前に、各入力カメラについて、S7 で共有される Apple Vision 空マスクが画像から抽出され、空ピクセルはカメラ内部パラメータを使用して 仮想球面 (標準半径 8x シーン半径) に逆投影されます。この球面上に ~5000 の新しい Gaussian が、投影された空ピクセルからの色の平均値、非常に大きなスケールリング (シーン単位で 1.0)、初期 Opacity 0.95 で初期化されます。これらの 5000 Gaussian は 古典的な意味での空マスクではありません — 他のすべてと同様にトレーニングされますが、高い初期 Opacity により薄い殻に残されます。結果: 屋外/ドローンシーンでの 360° 新規ビューでは、暗いコンフェッティの小さな塊の代わりに実際の空の色と雲の構造が表示されます。値は再起動を超えて記憶されます。少なくとも 360° カメラ カバレッジを持つ屋外シーンでのみ意味があります。空が見えない純粋なオブジェクト キャプチャでは効果がありません。ステータス: 実験的、追加の屋外セットでの より広範な A/B 検証はまだ保留中。

💬 簡単に言うと

トレーニングが空をいくつかの見えるピクセルから「推測」しようとして Floater になる代わりに、RadianceKit はトレーニング開始前に空のピクセルをシーンの周りの仮想球面に直接投影します。完成したシーンを 360° で回転させると、黒い塊の代わりに本物の空が見えます。画像に実際に空が見える屋外撮影でのみ機能します。リビングルームスキャンやスタジオショットでは何もしません。

メトリックセクション (I30~I38)

| Training Metrics | |
|------------------|-------------|
| Iteration | 5000 / 5000 |
| Loss | 0.006447 |
| Learning Rate | 0.00e+00 |
| Degree | 2 |
| Gaussians | 3,022 |
| GPU Memory | 1.8 MB |
| Speed | 364.0 It/s |
| Elapsed | 0:13 |

図 13: ブレークでのトレーニング完了後 (5K 反復、最終 2 991 Gaussian) の Training Metrics セクションのみのクロープ。トレーニングメトリックのテーブル (Iteration、Loss、SSIM Loss、Combined Loss、Gaussian Count、Learning Rate、Elapsed、ETA)

トレーニングが実行されている間、メトリックセクションはトレーニンググループから 9 つのライブ値を表示します。トレーニングの開始前、セクションは空です (「Start training to see live metrics」)。すべての値は ~30 反復ごとに更新されます (ストリームの更新頻度)。セクションは読み取り専用です – 要素はクリック可能でも変更可能でもありません。より深い分析には、~/Documents/RadianceKit/Logs/ の JSONL トレーニングログを使用してください (スクリプト `python3 scripts/analyze_logs.py best 5`)。

I30 Iteration



場所

Inspector → メトリック → Iteration。読み取り専用。



技術詳細

「4523 / 40000」形式の表示 — 計画された総反復に対する現在の反復。値を ~30 反復ごとにプッシュするトレーニンググループと同期してカウントします。2 番目の数字は、開始時の Max-Iterations 値に対応します。ユーザーが後でステッパーを調整しても変更されません — 実行中の実行は独自のスナップショットコピーを使用します。アプリが Training メニュー経由で反復を追加する場合 (Continue Training +5K/+10K/+20K)、分母が増加します。

簡単に言うと

トレーニングが現在どこにあるか。「4523 / 40000」は、40 000 ステップのうち 4523 が完了し、つまり約 11% 完了していることを意味します。左の数字は秒単位でカウントアップします。数分間止まっている場合、トレーニングが停止しています — ほとんどの場合、GPU スロットリングや競合アプリのヒントです。右の数字はトレーニング開始時の Max-Iterations 値 (118) に対応し、ステッパーを後で調整しても変更されません。Continue Training (+5K/+10K/+20K) では、追加のステップ分増加します。

I31 Loss



場所

Inspector → メトリック → Loss。読み取り専用。



技術詳細

小数点以下 6 桁の Float 値 (例えば「0.024385」)。結合 L1+SSIM Loss (I21 SSIM Weight で制御されるミックス) と、オプションで Perceptual Loss (I29) と他の正則化を測定します。スケールは絶対的ではなく、シーン依存 — ほとんどの比較に同じデータセットが必要です。良好な構成での典型的な終値:
 - Classic Quality 40K 反復: 0.022~0.025 (Horse、Truck、Garden) - MCMC Full 200K 反復: 0.024~0.028 - Outdoor ドローン 30K: 0.030~0.060 (幾何学的理由で悪い) - Indoor アパート: 0.018~0.025
 5K 反復後の 0.10 を超える値は SfM 問題 (悪いカメラポーズ) を示します — 中止して SfM を再計算してください。

簡単に言うと

レンダリングされた画像がオリジナルからどれだけ離れているか — L1,SSIM, オプションで Perceptual Loss から結合。小さいほど良い。0.03 未満はほとんど非常に良く、0.05 未満はまだ OK, 屋外シーンは幾何学的理由で 0.03~0.06 にあります。数千の反復後に 0.10 を超えるのは警告サイン — ほとんどの場合、カメラ再構築 (SfM がうまく機能しなかった) のせいです。スケールは絶対的ではなく、シーン依存です。同じデータセット内の比較のみを行ってください。値が突然上にジャンプする場合、ほとんどの場合 Gradient-Explosion イベントが発生しました。

I32 Learning Rate



場所

Inspector → メトリック → Learning Rate。読み取り専用。



技術詳細

科学表記表示 (例えば「1.60e-04」)。Position パラメータの現在の学習率 (3DGS には Position、SH-DC、SH-Rest、Opacity、Scale、Rotation の 6 つの独立した LR があります — ここでは代表値として Position LR が表示されます)。デフォルトの開始値 1.6e-4 で、Exponential Decay を介してトレーニング終了時に $\sim 1.6e-6$ まで減少します。減衰はトレーニング構成の LR-Schedule フィールド (第 6 章の T フィールド) で調整可能です。LR が異常に高いままの場合 (例えば 10K 反復後の $1e-3$ 以上)、これは誤って読み込まれた構成を示す可能性があります。

簡単に言うと

現在の最適化ステップの大きさ — 具体的には Gaussian 位置の学習率。1.60e-04 で始まり、トレーニング終了に向けて約 1.60e-06 まで指数関数的に減少 (「1.60e-06」= 0.0000016)。進行は自動的に実行され、ここで調整する必要はありません。値が 10 000+ 反復後にまだ $1e-3$ より大きい場合、おそらく誤った構成が読み込まれました — トレーニングを中止してプリセットを再選択してください。内部的には、3DGS は 6 つの独立した学習率 (Position, SH-DC, SH-Rest, Opacity, Scale, Rotation) を持ちます。ここでは代理として Position LR のみが表示されます。

I33 SH Degree



場所

Inspector → メトリック → SH Degree。読み取り専用。



技術詳細

0~3 の整数。色表現の球面調和次数。0 (DC コンポーネントのみ、つまり Gaussian ごとに方向に依存しない色 — 単一の RGB 定数) で開始し、トレーニング中に段階的に 3 まで上昇します。標準スケジュールは、1000/2000/3000 反復で次数を 1 ずつ上げます。SH-3 は Gaussian あたり 48 の色係数 (3 RGB チャンネル × 16 SH 基底関数) に対応します。より高い SH 次数 = より多くの方向依存反射 (光沢のある表面が異なる視点から正しく異なる外観) ですが、より多くのメモリと遅いトレーニング。

簡単に言うと

現在 Gaussian あたりの色表現がどれだけ複雑か。0 (点ごとに方向に依存しない色のみ) で始まり、段階的に 3 まで引き上げられます — 通常、反復 1000, 2000, 3000 で。レベル 3 は Gaussian あたり 48 の色係数を意味し、方向依存反射を可能にします。つまり、光沢のある表面が異なる視点から正しく異なる外観になります。アクティブに触れる必要はありません。スケジュールは自動的に実行されます。より高い次数はより多くのメモリを消費し、トレーニングをわずかに遅くします — しかし、これが現実的なハイライトの代償です。

I34 Gaussians



場所

Inspector → メトリック → Gaussians。読み取り専用。



技術詳細

モデル内の Gaussian の現在の数、ロケール区切り文字でフォーマット (例えば「524.318」)。成長:

- Classic: SfM 初期点 (通常 50K-300K) で開始し、Densify Until 直前まで Clone/Split を通じて成長し、その後トレーニング終了まで 静的 (Pruning を除く)
- MCMC: Sample 点が MCMC キャップまで追加され、その後 Relocation のみ

健全な終値: - Classic Quality: 400K-700K (Horse 524K、Garden 800K) - MCMC Full: キャップで正確に (デフォルト 150K、Auto-Scale Multiplier × SfM-Count でシーンに応じて 500K-1.5M)

MCMC で数がキャップの < 60% まで落ちる → 異常 (Collapse インジケータ、過度に 積極的な正則化を示唆)。

簡単に言うと

3D モデルが現在持つ Gaussian 点の数。トレーニング中に成長し、Densify Until (I20) に達するまで。その後、数は実質的に一定のままです。より多くの点 = より多くの詳細、しかしファイルサイズも大きく、Viewport でのレンダリングも遅くなります。500.000 Gaussian は中程度のシーンでの Classic Quality の典型的な平均値です。MCMC Full は、Auto-Scale (I17) に応じて 500K から 1.5M の間に着地します。MCMC で数が突然キャップの 60% を下回る場合、これは Collapse インジケータです — ほとんどの場合、過度に積極的な正則化。

I35 GPU Memory



場所

Inspector → メトリック → GPU Memory。読み取り専用。



技術詳細

Gaussian バッファメモリ消費量の推定値、Gaussian カウント × 616 バイト (Memory スタイルでフォーマット) として。616 バイトは、完全装備の Gaussian (位置、スケーリング、回転、Opacity、SH 係数次数 3、Gradient Accumulator) の経験的サイズです。表示にはレンダラーのオーバーヘッド (Tile バッファ、Sort バッファ、Backward バッファ) は **含まれません** — 実際の GPU メモリ需要は通常この値の 2-3 倍です。500K Gaussian で: 表示 ~290 MB、実際 ~700 MB。1.5M Gaussian で: 表示 ~880 MB、実際 ~2.5 GB。M3 Max では 64+ GB Unified Memory で問題ありませんが、M3 Pro では 18 GB ですすでに制限です。

簡単に言うと

Gaussian 自体が占有する GPU メモリ量の推定値 — 点あたり約 616 バイト。レンダラーが独自の Tile, Sort, Backward バッファを追加するため、実際の GPU 消費は表示の 2-3 倍です。16-18 GB の Unified Memory を持つ MacBook では、500K Gaussian 未満に留まるべきです。M3 Max または Studio (64+ GB) では、楽に 1.5M 以上を実行できます。トレーニングが突然クラッシュしたり、システムがスワップしたりする場合、ほとんどの場合、ここで制限に達しています — Render Scale (I22) を下げるか、Densify Until (I20) を減らします。

I36 Speed



場所

Inspector → メトリック → Speed。読み取り専用。

技術詳細

小数点以下 1 桁の秒あたり反復 (「24.3 it/s」)。直近 ~100 反復にわたる 移動平均としてトレーナーによって計算されます。典型値: - Quick プリセット (1K 反復): 80-120 it/s (短く、定常状態なし) - Classic 20K @ 1.0 Render Scale (Truck シーン、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 のため遅い) - 1M+ Gaussian とフル解像度で: < 10 it/s

トレーニング中の Speed の減少は正常です — より多くの Gaussian = 反復あたりより多くの計算。突然の落ち込み (例えば 30 → 5 it/s) は GPU 熱スロットリングまたは 競合アプリを示します。

簡単に言うと

トレーニングがどれくらい速く実行されているか、秒あたりの反復で。通常 20-50 it/s で、低い Render Scale (50%) と小さなシーンでは 80-120 it/s になることもあります。トレーニング中に自然に減少します。より多くの Gaussian = 反復あたりより多くの作業のため。突然の落ち込み (例えば 30 → 5 it/s) は GPU 熱スロットリングや競合アプリを示します — ビデオ付きブラウザタブ、Time Machine バックアップ、Photos のインデックス作成。アプリを前面に保ち、バックグラウンドプログラムを閉じることがしばしば役立ちます。1M+ Gaussian とフル解像度では、10 it/s 未満が正常です。

I37 Elapsed



場所

Inspector → メトリック → Elapsed。読み取り専用。

技術詳細

「4:23」(m:ss) または 「1:23:45」(h:mm:ss) としての経過時間。1 時間から 形式が切り替わります。純粋なトレーニング時間のみを測定し、前置段階 (SfM 計算、画像インポート) は含まれません。Pause/Resume では時計が続行されます — つまり Wall-Clock であり、CPU 時間ではありません。

簡単に言うと

トレーニングがすでにどれくらい実行されているか、純粋なストップウォッチ (Wall-Clock 時間) として。形式は 1 時間まで 「m:ss」、その後 「h:mm:ss」。「CPU 時間」ではなく「すでにどれくらい待っているか」 — つまり、Pause 時間も含まれます。純粋なトレーニング段階のみを測定し、前置 SfM 計算や画像インポートは含まれません。ETA (I38) との比較に役立ちます — Elapsed が元の ETA を大きく超えて飛び出した場合、トレーニングはどこかで計画より遅くなりました。

I38 ETA

Inspector → メトリック → ETA。読み取り専用。

 技術詳細

「17:42」または「1:12:35」としての推定残り時間。計算: (Max Iterations – 現在の反復) / 秒あたりの反復。Speed がちょうどゼロの場合 (最初または Pause) は「-」を表示します。推定は、トレーニング終了に向けた典型的な減速に **適応されません** — 特に MCMC と Densify-Until 値の大きい Classic では、より多くの Gaussian が画像に入るため、トレーニングは遅くなる傾向があります。実際には、通常初期 ETA より 10-20% 上に留まります。

 簡単に言うと

推定された残り待ち時間 — 残りの反復と現在の Speed (I36) から計算。大まかな推定: Mac が突然遅くなった場合 (Densify フェーズから多くの Gaussian, 熱スロットリング、他のアプリ), 表示より長くかかる可能性があります。アプリはトレーニング終了に向けた典型的な減速を含めません。そのため、実際の終わりは通常、初期 ETA を 10-20% 超えます。15% を計算に入れると、ほとんどの場合うまく適合します。Speed が現在 0 の場合 (トレーニングの開始または Pause), 「-」が表示されません。

Loss チャートセクション (I39~I41)



図 14: トレーニング完了後の Loss History セクションのみのクロップ – Current 0.0064、Min 0.0035 (緑)、0.027 (反復 1) から 0.0035 (反復 5K) への青の経過、反復 200 周辺の特徴的なキック付き、その下にオレンジの Gaussian Count チャート

Loss チャートセクションは、時間にわたるトレーニングの進行を可視化します。2つのチャートで構成されます: Loss Curve チャート (大、上、青) と Gaussian Count チャート (小、下、オレンジ)。両方ともトレーニング中にライブで構築され、次のトレーニング開始まで永続します。最初のトレーニング前、領域は空です (「Loss curve will appear during training」)。チャートは純粋な SwiftUI Path 描画です (Swift Charts フレームワークなし) – 100K+ ポイントでもスムーズにレンダリングします。

I39 Current Loss (表示)



場所

Inspector → Loss チャート → 左ラベル領域
「Current: 0.0287」。読み取り専用。



技術詳細

最後の Loss サンプルポイントの Float 値、小数点以下 4 桁でフォーマット。I31 (Metrics セクションの Loss) と同じ、ここではよりコンパクトにフォーマットされているだけ。ソースは Loss History — ~30 反復ごとに 1 エントリを取得する リスト。有限値のみがリストに追加されます — NaN/Infinity (非常にまれ、Gradient-Explosion バグの場合) はフィルタリングされません。

簡単に言うと

Metrics セクションよりも短い表記の現在の Loss 値 (小数点以下 4 桁)。内容的には I31 と同じですが、ここでは表示が Loss チャートの直接に座っており、曲線を観察するときに正確な数値が得られます。すべてのライブメトリックと同様に、~30 反復ごとに更新されます。NaN または Infinity 値 (Gradient-Explosion バグの非常にまれなケース) は、アプリが自動的にフィルタリングします。図を見ながら他のセクションにジャンプする必要がないので便利です。

I40 Min Loss (表示)



場所

Inspector → Loss チャート → 右ラベル領域 「Min: 0.0245」 (緑)。読み取り専用。



技術詳細

現在のトレーニング実行のこれまでに見られたすべての Loss 値の最小値。Loss History からライブで再計算されます — 別の永続性なし。緑のフォントで表示されます。「Min」 = 「これまでで最高」だからです。チャート下部の緑の破線は、この Y 位置を視覚的にマークします。Continue Training セッションでは、最小追跡が新規開始されます — 古い履歴は UI で新しいものに置き換えられます (追加されません)。現在のトレーニングが前のものより悪く実行されている場合、Min 表示は前の最終結果よりも大きい可能性があります。

簡単に言うと

このトレーニングがこれまでに見た最低の Loss 値 — 緑で表示、「これまでで最高」だからです。チャート下部の緑の破線も、この位置を視覚的にマークします。現在の曲線がそれをはるかに上回る場合、運があれば改善があります。ほとんどの場合、Min は後で関心のある最終結果の指標です。Continue Training セッションでは、UI で古い履歴が新しいものに置き換えられるため、Min 追跡が新規開始されます — Min 値は前の最終結果よりも悪く見える可能性があります。

I41 Gaussian Count チャート



場所

Inspector → Loss チャート → その下の 2 番目のチャート (オレンジ)。読み取り専用。



技術詳細

トレーニング反復にわたる Gaussian カウントの線グラフ。ソース: Gaussian Count 履歴 (トレーナーから ~30 反復ごとに埋められる (Iter、Count) ペアのリスト)。Y スケールは履歴の最小値と最大値の間で動的。Classic 戦略では、曲線は通常次のように見えます: Densify Until まで安定的に上昇、その後フラット (小さな Pruning 変動付き)。MCMC では: キャップまでの急上昇、その後水平線 (Relocation が数を一定に保つ)。アクティブなトレーニングにもかかわらず曲線が **落ちる** 場合、Densification が過度に積極的に Pruning しています — 誤ったデフォルトまたは既知の MCMC-Collapse バグ (v1.4.4 ホットフィックステーマ) の指標。

簡単に言うと

トレーニング時間にわたる Gaussian の数がどのように発展するか — Loss 曲線の下小さなオレンジのチャート。Classic 戦略では、線は Densify Until (I20) に達するまで安定的に上昇し、その後小さな Pruning 変動でフラットなままです。MCMC では、急にキャップまで上昇し、その後水平のままです。Relocation が数を一定に保つからです。アクティブなトレーニングにもかかわらず曲線が突然下に折れる場合、Densification が Pruning で過度に積極的です — v1.4.4 の MCMC-Collapse バグの典型的な兆候。アプリの更新または Classic への切り替えが役立ちます。

Loss 曲線の読み方

Loss チャートは Inspector の最も重要な診断ツールです — トレーニングが有用に進んでいるか停止しているかを、これほど直接示す他のインジケータはありません。典型的な健康的な形状は、最初の 1000~3000 反復での速い低下 (~0.15 から ~0.05 へ)、続いてトレーニング終了までの遅く均一な低下 (0.020~0.030 へ) です。対数的には、曲線はスムーズな対角線のように見えます。

Loss のプラトーは何を意味するか? 曲線が数千の反復にわたってフラットなままの場合、2 つの可能な解釈があります: (a) トレーニングが「収束」した — モデルが与えられたデータと設定でなれる限り良いため、Loss はもはや有意に低下できません。これは望ましいです。これが「完了」です。(b) トレーニングが「停止」している — Loss は実際にはまだ低下できませんが、最適化が停滞しています (ローカル最小値、学習率が小さすぎる、Densification オフ)。区別: Loss 値が典型的に良い範囲 (Indoor/Object で 0.020~0.030、Outdoor で 0.040~0.060) にあり、曲線が 5K 反復から平坦である場合、収束しています。値が同等のシーンより大幅に高い場合 (例えば 0.08)、停止しています。

注意 Gaussian プラトー ≠ Loss プラトー。 Gaussian カウントのプラトーは「トレーニングが完了」を意味 **しません**。Densification が新しい点の追加を停止したことを意味するだけです — (Classic で) に達したため、または MCMC キャップがいっぱいになったためです。トレーニングはその後も継続し、既存の点のみを洗練します。実際の「完了」信号は、ここではなく Loss 曲線と Iteration 表示 (I30) で読みます。

中止の経験則: Loss 曲線が 5000+ 反復後に 0.08 を超え、もはやほとんど低下しない場合、高い確率で SfM 再構築が間違っています。トレーニングを中止し、第 9 章で選択され

た SfM バックエンドがシーンに合っているか確認し、必要に応じて COLMAP/Native に切り替えてから、再起動してください。悪いカメラ整列で 2 時間のトレーニングよりも、より良い SfM に 10 分投資する方がよいです。

いつ Inspector を使うか?

クイックリファレンス: 各典型的なユースケースにどのセクションとコントロール?

| 共通タスク | セクション | コントロール ID |
|--------------------------------------|--------------|---|
| 完成した Splat の色の彩度を下げる | Look | L1 (Saturation) |
| 針状/コンフェッティ Splat を丸める | Look | L2 (Splat length) |
| 穴だらけの雲を埋める / Splat を大きくする | Look | L3 (Splat size) |
| オービットで遠い「Far コンフェッティ」をフェードアウト | Look | L4 (Fade far region) |
| Look 調整を破棄 | Look | L5 (Reset finishing) |
| 事前準備されたセットアップのロード | Presets | I7 (行をクリック) |
| 独自セットアップの保存 | Presets | I1 → I2 → I4 |
| 同僚とセットアップを共有 | Presets | I5 (Export) または I6 (Import) |
| SfM バックエンドの変更 (例えば Apple-PG が不安定すぎる) | トレーニング構成 | I12 (第 9 章参照) |
| EXIF 焦点距離なしのビデオフレーム 処理 | トレーニング構成 | I13 (FOV Override) |
| COLMAP パフォーマンス: 古典の代わりに GLOMAP | トレーニング構成 | I14 |
| Classic から MCMC への切り替え | トレーニング構成 | I15 |
| トレーニングを長く実行 | トレーニング構成 | I18 (Max Iter) + I20 (Densify Until) – I19 でリンク |
| GPU 時間を半分にする | トレーニング構成 | I22 (Render Scale を 50%) |
| トレーニング品質 +6% (MCMC) | トレーニング構成 | I16 (MCMC Quality) |
| SfM 点が多い屋外 シーン | トレーニング構成 | I17 (Auto-scale by scene) |
| COLMAP パスの設定/変更 | トレーニング構成 | I23 / I24 / I25 |
| エクスポートファイルを小さくする | Enhancements | I26 (常にオン のまま) |
| トレーニング時間を増やさず Viewport を鮮明にする | Enhancements | I27 (Viewport Scaling → MetalFX) |
| MetalFX が過度に平滑化 → 代替 | Enhancements | I27 (Viewport Scaling → Lanczos) |
| 細かい構造での最後の詳細の一滴 | Enhancements | I29 (Perceptual Loss 0.05-0.1) |
| トレーニングを監視 | メトリック | I30 (進捗)、I36 (速度)、I38 (残り時間) |
| 収束時間と問題の認識 | メトリック | I31 (0.20 以下で 100% 未満の SfM 再計算) |
| 損失関数の認識 | メトリック | I32 (0.20 以下で 100% 未満の SfM 再計算) |
| リソース使用量の認識 | メトリック | I33 (0.20 以下で 100% 未満の SfM 再計算) |
| メモリ使用量の認識 | メトリック | I34 (0.20 以下で 100% 未満の SfM 再計算) |
| スプラッシュの認識 | メトリック | I35 (0.20 以下で 100% 未満の SfM 再計算) |

章

第 3 章 — 設定

設定ウィンドウは RadianceKit → 設定... または標準ショートカット **⌘,** から開きます。2 つのタブを含みます: **General** と **AI Helpers**。第 2 章の Inspector 値と異なり、このウィンドウの設定は **アプリ全体** (すべてのプロジェクトにわたって) 適用されます — これらは永続化され、アプリ再起動を超えて保持されます。General タブは 3 つの内容領域をグループ化しています: Interface、Viewport、Training。(以前ここにあった 3 つの屋外 Floater トグル — Sky Masking、Mid-Training Floater Cleanup、Reconstruct Sky Dome — は v1.6 でエキスパート Inspector の Enhancements セクションに移動し、現在はプロジェクトごとに保存されます。第 2 章 I42~I44 を参照してください。) AI Helpers タブは、SfM とトレーニングの前処理のための オンデバイス機械学習ヘルパー (Vision、CoreML) を有効化します。すべての AI Helpers をまとめて有効化/無効化する以前の操作要素は、現在のバージョンには存在しなくなりました — それに応じてここでは記載しません。未提供のヘルパー用の以前の「Coming Soon」エリアも削除され、ここでは参照されていません。

General タブ

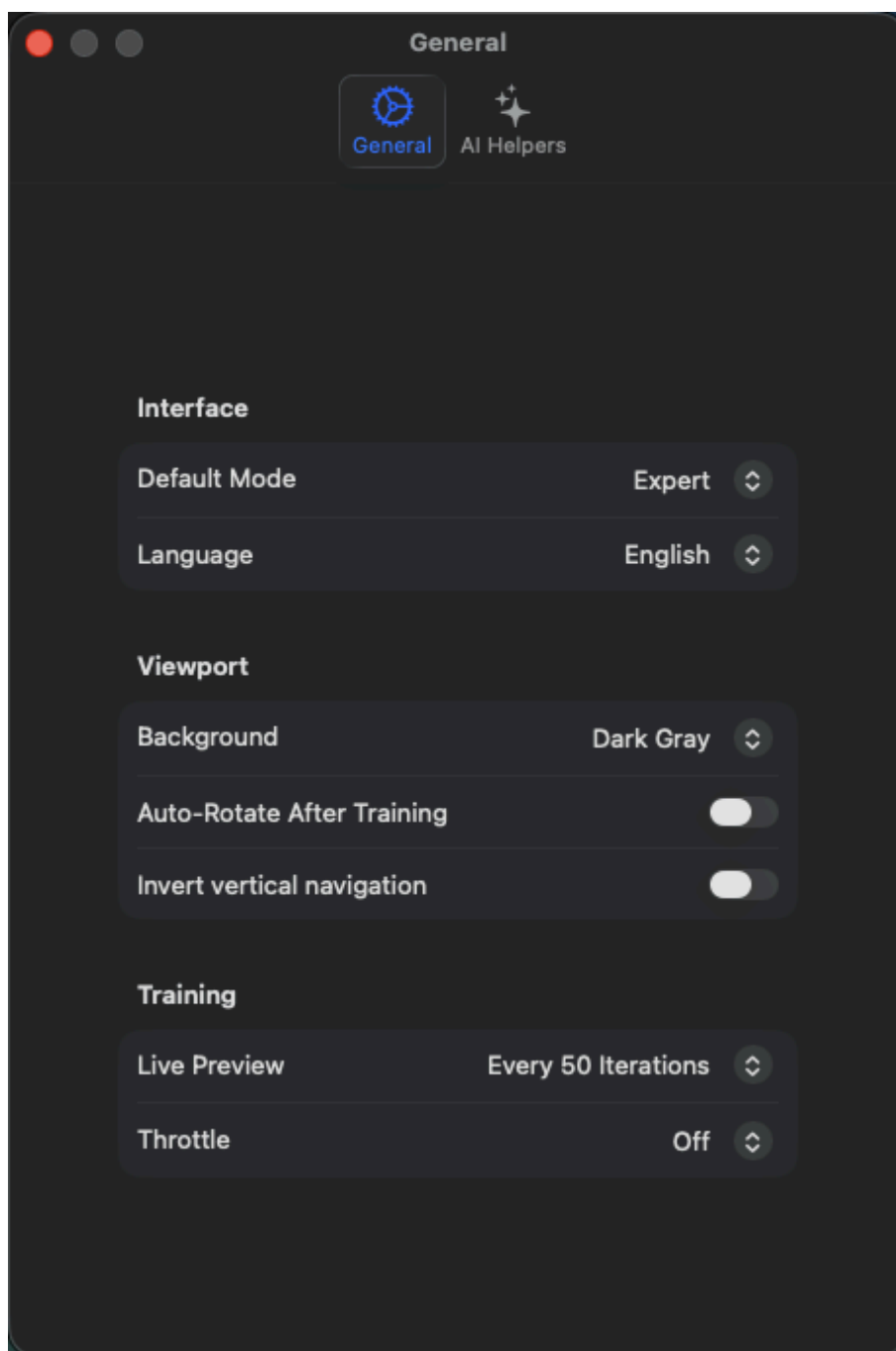


図 15: 設定 → General タブ、Interface、Viewport、Training、Experimental セクションを含む

S1 Default Mode

場所

Settings → General → Interface → Default Mode ピッカー。バインド済み。デフォルト: `.simple`。

技術詳細

次回起動時にアプリが2つのUIモードのどちらで開くかを制御します。「Simple Mode」は4ステップのガイド付きウィザードワークフロー (インポート → 処理 → プレビュー → エクスポート、第10章 Z1~Z4に記載)、「Expert Mode」は第2章のナビゲーター、3Dビューポート、エキスパート Inspector を持つ古典的な3パネルレイアウトです。値は再起動を超えて記憶されます。メニュー Mode → Simple Mode (⌘1) / Mode → Expert Mode (⌘2) と同じ効果ですが、メニューは現在のセッションを切り替えるのに対し、このピッカーは将来のセッションのデフォルトを設定します。両モードは同じプロジェクト状態にアクセスします。プロジェクト、カメラ、トレーニング構成はモード切り替え時に保持されます。モード固有のツールバーボタンは即座に再レンダリングされます。

簡単に言うと

ここで、次回起動時に RadianceKit が開くインターフェイスを選択します。「Simple Mode」は初心者モードです: 4つの明確なステップ、事前設定されたプリセット、ほとんどオプションなし。「Expert Mode」は第2章で見るすべてのコントロールを持つフルツールボックスレイアウトです。画像やトレーニングの進行を失うことなく、メニュー「Mode」を介していつでも切り替えられます。

S2 Language

場所

Settings → General → Interface → Language ピッカー。バインド済み。デフォルト: `.system` (macOSの言語に従う)。

技術詳細

macOS システム言語とは独立して、アプリ UI 全体の表示言語を選択します。RadianceKit は 17 言語にローカライズされています (`de`、`en`、`pl`、`en-AU`、`ar-SA`、加えて 12 言語)。「System」では、アプリは macOS の言語に従います。明示的な選択の場合、言語設定は再起動を超えて記憶されます。完全な反映には通常、アプリの再起動が必要です。なぜなら、ローカライゼーションバンドルは起動時にのみ読み込まれるからです。プロジェクト内の 298 の文書化されたローカライゼーションキーがすべて考慮され、サブビューやヘルプツールチップのすべてのテキストを含みます。

簡単に言うと

Mac が英語で動作していて、ドイツ語の RadianceKit インターフェイスを好む場合 (または逆)、ここで設定します。ほとんどのテキストはすぐに切り替わります。一部のダイアログは、アプリ再起動後に新しい言語で表示されます。

S3 Viewport Background



場所

Settings → General → Viewport → Background ピッカー。バインド済み。デフォルト: `.darkGray` (RGB 0.1, 0.1, 0.1)。



技術詳細

3D ビューポートの標準背景色を設定します。3つのオプション: 「Dark Gray」(RGB 0.1, 0.1, 0.1 — デフォルト)、「Black」(0, 0, 0)、「White」(1, 1, 1)。この設定は、新規プロジェクトとセッションのデフォルトを再起動を超えて永続化し、同時に実行中の Metal レンダラーを即座に更新します。同じオプションがメニュー Viewport → Background (M21, M22, M23) にもありますが、Settings ピッカーはデフォルトを設定し、メニューは現在の表示を切り替えます。スクリーンショットとデモビデオに重要: 白い背景は緑/青の Floater をより強く強調し、暗い背景はクリーンなレンダリングショットに適しています。

簡単に言うと

プレビューウィンドウ内の 3D モデルの後ろの色。ダークグレーが標準で、ほとんどのシーンに適しています。白はスクリーンショットに適しており、黒はレンダリングショットでより上品に見えます。メニュー「Viewport → Background」からいつでも現在のシーンの色を切り替えられます — この設定は次回開いたときにどの色がアクティブになるかを決定するだけです。

S4 Auto-Rotate After Training



場所

Settings → General → Viewport → トグル「Auto-Rotate After Training」。バインド済み。デフォルト: `false`。



技術詳細

トレーニング終了直後、シーン重心の周りでビューポートカメラの連続ターンテーブル回転を開始します (標準回転速度 ~0.3 rad/秒)。デモセッション、A/B 比較、360° ビューから直接「Floater」がシーン境界で発生したかを判断するのに 実用的に便利です。視覚的にはメニュー Viewport → Toggle Auto-Rotation (M16, ⌘⇧T) と同じ効果ですが、こここのトグルは手動ではなくトレーニング終了後に自動的に動作をトリガーします。後でメニューやビューポート内のクリック (回転を一時停止) を介して、いつでも中断できます。トレーニングパフォーマンスには影響しません — 回転はトレーニング完了後にのみ実行されます。

簡単に言うと

有効にすると、トレーニング完了直後に 3D シーンが自動的に回転します — 回転木馬のようです。夜通しトレーニング後、自分でクリックすることなく朝に結果が動いているのを見られると素敵です。トレーニングを監視するだけの長いセッションでは、オフにしておく方が良いでしょう。

S5 Live Preview Interval



場所

Settings → General → Training → Live Preview ピッカー。バインド:

AppState.trainingConfig.livePreviewInterval。
デフォルト: 0 (Off)。



技術詳細

実行中のトレーニングスナップショットを 3D ビューポートにレンダリングする反復間隔を決定します。4 つの離散値: 0 (「Off」)、50、250、1000 反復。Live Preview がアクティブな場合、トレーナーは GPU から別のレンダリングバッファに Gaussian バッファをコピーし、ビューポートの再描画をトリガーします。「Off」では、ビューポートはトレーニング完了後にのみ更新されます。パフォーマンスコスト: 50 反復ごとは M3 Ultra で ~5~10% 遅く、250 反復ごとは ~1~2% 遅く、1000 反復ごとは測定不能です。メモリアーバーヘッドは間隔に関係なく、スナップショットバッファ用に ~2 GB で一定。値は新規トレーニング用のデフォルトとして機能します。トレーニング開始後、トレーニング Inspector はこのトレーニングの実際のライブ値を表示します。間隔 50 では視覚的印象は点群のスムーズな「成長」、1000 ではぎこちなく感じます。

簡単に言うと

トレーニングが実行されている間、3D ビューの更新頻度を選択できます。「Off」とは、トレーニング中に更新しないことを意味します (最速)。「Every 50 Iterations」は、シーンの形成をほぼリアルタイムで表示します (少し遅い)。小さなトレーニングでのんびり見るには、「Every 250」が良い妥協点です。

S6 Throttle Delay

場所

Settings → General → Training → Throttle
ピッカー。バインド:

AppState.trainingConfig.throttleDelayMs。デフォルト: 0 (Off)。

技術詳細

トレーニング反復間にミリ秒単位の人工的な遅延を挿入します。4つの離散値: 0 (「Off」)、2 (「Light」)、5 (「Moderate」)、10 (「Eco」)。意味: 長いトレーニング (数時間) では、GPU が 100% 使用されない限り、システム UI が顕著に遅くなります (マウスカーソルが引っかかり、他のアプリが鈍くなる)。Throttle 遅延は、他のタスクが実行できる休憩を GPU に与えます。パフォーマンス コストは大きい: 5 ms Throttle では、典型的な 40K トレーニングは Throttle なしと比較して約 50~80% 長くかかります。パフォーマンスモード「Eco」(10 ms) では、反復あたりの遅延は反復自体より長くなります — 2~3 倍遅い係数。Throttle がアクティブな場合、ピッカーの下にヒントが表示されます: 「Throttle is on. Training will be slower than usual.」アプリ自体は顕著に反応がよくなりません — 他のアプリだけが恩恵を受けます。

簡単に言うと

長時間のトレーニング中に Mac が熱くなりすぎたり、他のプログラムが重くなりすぎたりする場合、ここでブレーキを入れてください。「Off」は GPU に全力を与えます (最速)。「Light」は各ステップ間に小さな休憩を入れます (少し遅いが、システムの反応が良い)。「Eco」は最強のブレーキです — 熱くなりすぎるとはいけない MacBook での夜間トレーニングに適しています。

AI Helpers タブ

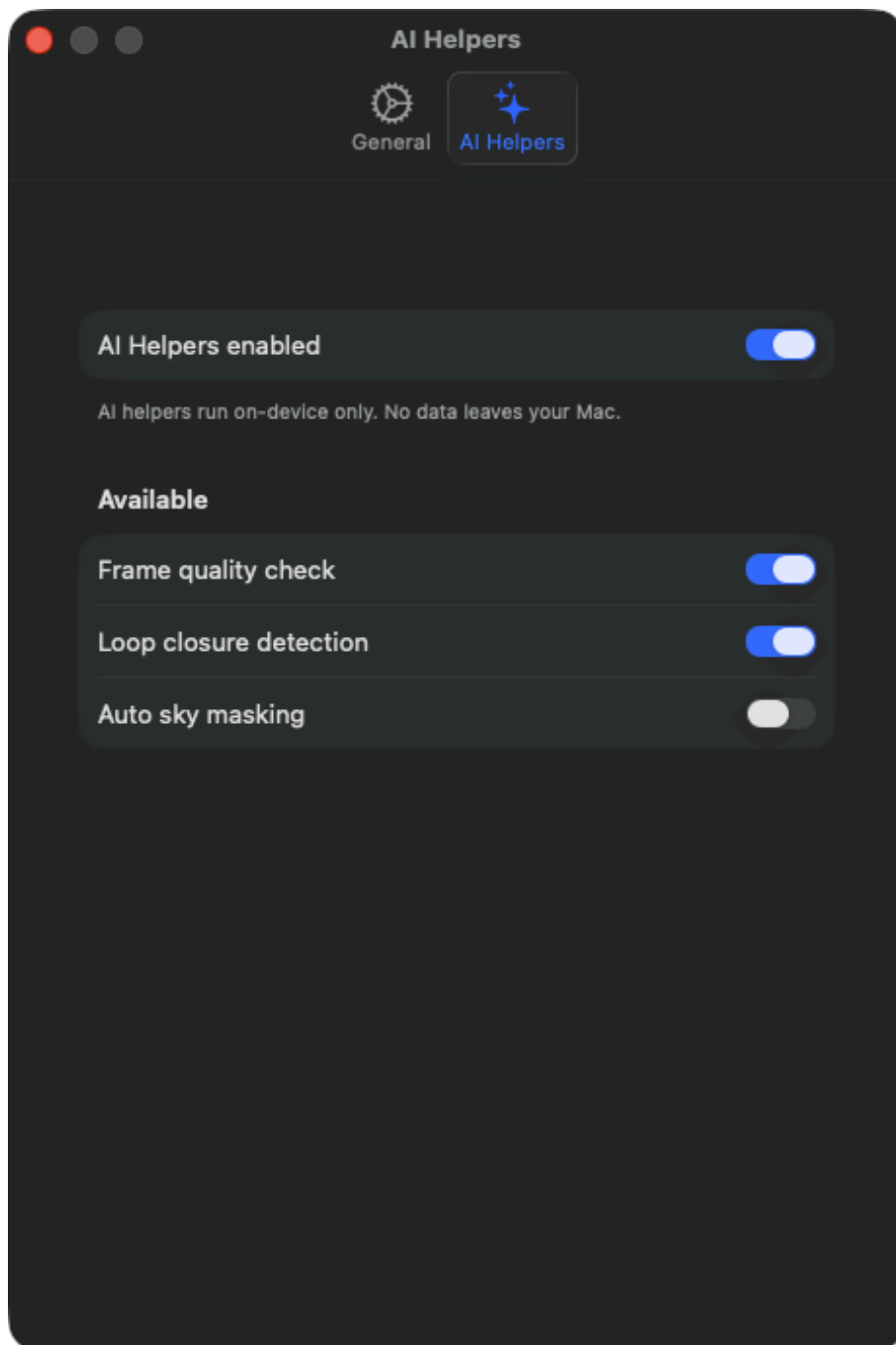


図 16: 設定 → AI Helpers タブ、マスタースイッチとサブトグル

S11 AI Helpers enabled (Master) 場所

Settings → AI Helpers → 最初のセクション → トグル「AI Helpers enabled」。バインド済み。デフォルト: `true`。

 技術詳細

パイプライン内のすべての AI Helpers 機能のマスタースイッチ。オフの場合、インポートと SfM パイプラインはすべての ML ベースの前処理ステージを完全にスキップします — Apple Vision 呼び出しなし、CoreML モデルロードなし、NPU 起動なし。オンの場合、個々のサブトグル (S12~S13) が参照されます。値は再起動を超えて記憶されます。以下のステージに影響します: (a) SfM 前のフレーム 品質事前チェック (S12)、(b) Loop Closure 検出 (S13)。重要: オフの場合、2つのサブトグルは無効化され、視覚的にグレーアウトされます。フッターヒントは、すべての AI Helpers が厳密にオンデバイスで実行されることを強調しています — 画像アップロードなし、クラウド処理なし。データ保護保証は、Apple Vision フレームワーク (Neural Engine 上で ローカル) と、アプリバンドルに直接配置された CoreML モデルのみを使用することで実現されています。

 簡単に言うと

内部で AI / 機械学習を使用するすべての機能のメインスイッチです。標準は「オン」です。なぜなら、ヘルパーは画像が Mac を離れることなく多くの時間を節約するからです。完全にオフにしたい場合 (例えば電力を節約したい、または Mac に NPU がない)、ここでオフにします — そうすると、下の2つのサブオプションは自動的にグレーになり、何もしません。

S12 Frame quality check 場所

Settings → AI Helpers → Available セクション → トグル「Frame quality check」。バインド済み。デフォルト: true。

 技術詳細

Frame Quality Screener (Phase 3.11) を有効化します。これは SfM 呼び出し前に各インポートされたフレームを分析します。フレームごとのパイプラインステップ: (a) Apple Vision からのラプラシアン分散フィルター (ぼけ検出 — 閾値 ~150)、(b) ヒストグラムベースの過/低露出チェック (閾値: 0 または 255 で 5% を超えるピクセル)、(c) 空白フレーム検出 (すべてのピクセルにわたって標準偏差 < 5)。3 つすべてのチェックに合格した フレームは直接通過します。少なくとも 1 つのチェックに失敗したフレームは、各問題のあるフレームをサムネイルと理由とともにリストし、削除するかどうかを尋ねる モーダル確認ダイアログをトリガーします。重要: 自動削除なし — ダイアログは 常に必要で、ユーザーが最終決定を保持します。パフォーマンス: M3 Ultra で フレームあたり ~50 ms、並列実行。オフの場合、すべてのフレームはチェックなしで SfM に渡されます。マスター (S11) が無効化されている場合、このトグルは視覚的に グレーアウトされ、効果がありません。メモリに従う出荷ステータス: SHIPPED 2026-05-23。

 簡単に言うと

実際のトレーニング前に、アプリは各写真を確認します: ぶれていますか? 完全に暗いか白いですか? 空ですか? もしそうなら、画像を捨てるかどうかを尋ねます — 自動的に削除することはありません。1 つの完全にぶれた画像が時々トレーニング全体を台無しにすることがあるため、これにより後で多くの時間を節約します。標準は「オン」です。なぜなら、コストはほぼゼロで利益が大きいからです。

S13 Loop closure detection

場所

Settings → AI Helpers → Available セクション → トグル「Loop closure detection」。バインド済み。デフォルト: true。

技術詳細

Apple Vision Feature Print ベースの Loop Closure 検出を有効化します。各インポートされたフレームについて、画像コンテンツの神経埋め込みを表す ~768 次元の特徴量ベクトルが計算されます。次に、すべての Feature Print がペアごとに Cosine Similarity 経由で比較されます。Similarity > 0.85 でフレーム インデックスの距離 > 50 (つまり非隣接フレーム) のペアが「Loop Closure 候補」として識別され、プロジェクトフォルダ内のサイドカー JSONL ファイルに書き込まれます。情報提供のみ — インポートされた画像シーケンスは変更されません。意味: SfM ソルバー (特に COLMAP) に、これらのフレームが 3D 空間で一緒にクラスタされることを示すヒントを与えます。ネイティブ SfM では、サイドカー情報は現在 文書化目的のみです。COLMAP はカスタム matches-file を介してヒントを内部的に 使用します (手動統合可能、自動的にリンクされません)。パフォーマンス: M3 Ultra でフレームあたり ~200 ms、並列実行。オフの場合、Feature Print は 生成されません。マスター (S11) が無効化されている場合は視覚的にグレーアウト。

簡単に言うと

オブジェクトの周りを写真撮影して最後に出発点に戻る場合、それを知ることはコンピューターにとって非常に役立ちます。このオプションは、どの写真が「ほぼ同じ場所から」撮影されたかを自動的に検出し、それを小さなヘルパーファイルに書き込みます。SfM ツール (特に COLMAP) はこの情報を使用して、よりクリーンな 3D 再構築を提供できます。標準は「オン」です。なぜなら、操作なしで実行され、画像を変更しないからです。

Inspector ミラー設定

インベントリテーブルの他の設定エントリ (S17~S33) は、エキスパート Inspector からの ミラーリングであり、第 2 章 (Inspector コントロール I12~I29) に記載されています。これらは設定ウィンドウに物理的に表示されません。インベントリでは、永続化される `TrainingConfig` プロパティを介して実行されるため、形式的に設定としての性質を持つという理由でのみ列挙されています。詳細な説明はそこをご覧ください。

いつ何を？

| 設定 | スコープ | 永続性 |
|-------------------------------|-----------------------|--------|
| S1 Default Mode | アプリ全体 | アプリ再起動 |
| S2 Language | アプリ全体 | アプリ再起動 |
| S3 Viewport Background | アプリ全体 (デフォルト) + ランタイム | アプリ再起動 |
| S4 Auto-Rotate After Training | アプリ全体 | アプリ再起動 |
| S5 Live Preview Interval | 新規トレーニングの デフォルト | アプリ再起動 |
| S6 Throttle Delay | 新規トレーニングのデフォルト | アプリ再起動 |
| S11 AI Helpers Master | アプリ全体 | アプリ再起動 |
| S12 Frame quality check | アプリ全体 | アプリ再起動 |
| S13 Loop closure detection | アプリ全体 | アプリ再起動 |

アプリ全体 = すべてのプロジェクトに作用。新規トレーニングのデフォルト = 次に作成されるトレーニングのみに作用、実行中のセッションは変更されません。現在のトレーニング = 実行中のトレーニング構成に即座に作用しますが、明示的な再インポートなしには永続化されません。

章

第 4 章 — 補助ウィンドウ

メインウィンドウ (3D Viewport と Inspector) に加えて、RadianceKit は Help メニューから開く 7 つの追加ウィンドウを管理します。上から下の順番: User Guide (⌘?), Keyboard Shortcuts (⌘/), Open Training Logs... (アプリウィンドウではなく Finder を開くため、ここでは扱いません)、Manage Storage...、Pareto Dashboard... (⇧⌘D)、Holdout Analysis... (⇧⌘H)、BayesOpt Console... (⇧⌘B)。そのうち 3 つ — Dashboard、Holdout、BayesOpt — は独立した分析ツールです。それぞれ独自の ViewModel スタックを持ち、ディスク上の JSON ファイルを読み書きし、各ウィンドウをアプリ起動時に特定のファイルを表示するように指定する CLI 引数があります (`--dashboard-dir`、`--holdout-file`、`--bayesopt-autorun`)。

4 つのシンプルなウィンドウ (User Guide、Keyboard Shortcuts、Manage Storage、加えてサブメニュー項目 Open Training Logs / Open Exports Folder) は、コントロールごとに短いエントリを得ます。3 つの分析ウィンドウはより詳細に記載されています — それぞれ、ウィンドウで何を見るか、いつ開くべきか、表示される画像をどのように解釈するかを説明する導入付きです。

章末には、メインウィンドウの Inspector へのクロスリファレンスセクションがあります: 実行中のトレーニング中にライブ Loss チャートと Gaussian Count 表示で意味を持って読み取れるもの。

User Guide (W1~W4)

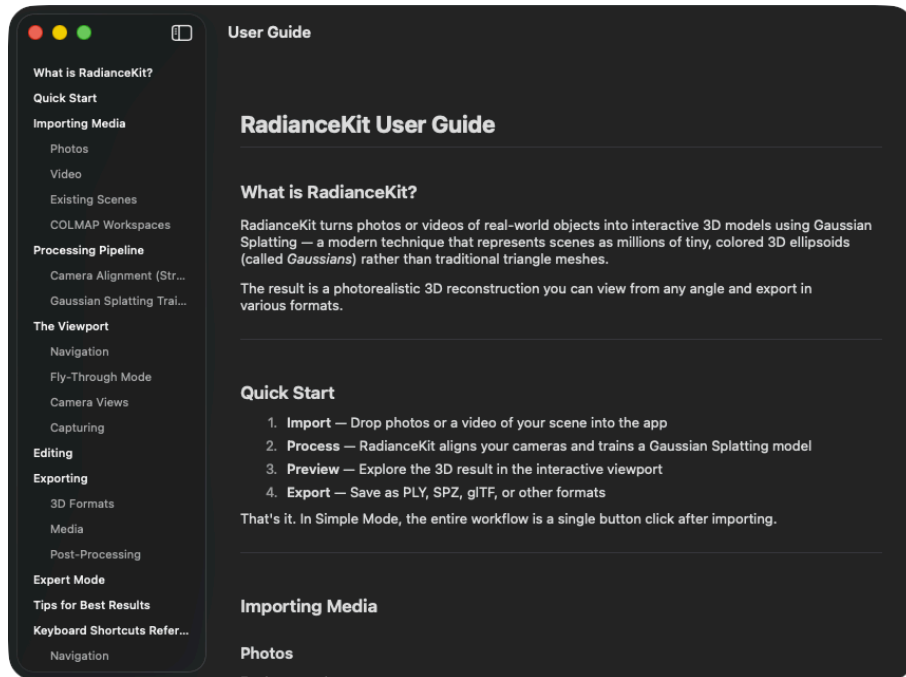


図 17: 左にサイドバー、右にレンダリングされた Markdown コンテンツ付き User Guide ウィンドウ

何か: アプリに同梱されている `guide_<言語>.md` をレンダリングする 組み込みヘルプウィンドウ。言語は設定 (タブ General → Language) から、またはそこに「System」がある場合は macOS 言語設定から導出されます。レイアウトは 古典的: 左にすべての見出しがあるサイドバー、右に本文。

いつ開くか 単一の点について素早いリマインダが必要な場合 — キーワード代替として。詳細なリファレンスはこのマニュアルです。組み込みヘルプ ウィンドウは、コマンドラインの `--help` に近いものです。各アプリリリースで 更新されますが、内容的にはより表面的に保たれています。

W1 NavigationSplitView (サイドバー + 詳細)



場所

Help → User Guide (⌘?)。



技術詳細

コンテンツツリー用の狭いサイドバー (最小 180 pt 幅) と、実際の Markdown コンテンツ用のスクロール可能な詳細領域を持つ 2 列レイアウト。ウィンドウは 700 × 500 pt の最小サイズです。最初に開くとき、ウィンドウは適切な `guide_<lang>.md` をアプリバンドルから読み込み (フォールバック `guide_en.md`)、ブロックレコード (H1~H4 見出し、段落、リスト、テーブル、区切り線) に解析し、サイドバー用に見出し構造を別に抽出します。インラインフォーマット (Bold、Italic、Code-Span) は組み込み Markdown エンジンでレンダリングされます。言語はアプリ設定から読み取られ、これらのバリエーションが zh または pt と異なるため、中国語 (zh-Hans) とブラジルポルトガル語 (pt-BR) は完全なロケールタグとして 保持される特別なケースです。

簡単に言うと

組み込みヘルプテキスト。左にトピックリスト、右にコンテンツ。言語はシステム設定に従って自動的に調整されます。オフラインで機能しますが、意図的にショーバージョンのみです — 完全なリファレンスはこのマニュアルです。

W2 List (見出しサイドバー)



場所

User Guide ウィンドウの左カラム。



技術詳細

現在の Markdown ドキュメントのすべての H2 と H3 見出しのリスト。H2 エントリはインデントなしで Medium フォントウェイト、H3 エントリは左に 16 pt のインデントと縮小された前景スタイルで表示されます。H4 以下は無視されます。深度がさらに深くなるとサイドバーが見にくくなるためです。アンカー ID は見出しテキストから slugification 経由で生成されます (小文字 + スペースをダッシュに + 文字/数字/ダッシュに対するフィルタリング — GitHub が Markdown アンカーで使用するのと同じアルゴリズム。そのため、ドキュメントへの外部 URL も同じアンカーに到達する可能性があります)。リストはネイティブ macOS スタイルを使用します。

簡単に言うと

左側のナビゲーションバー。エントリをクリックすると、セクションにジャンプします。

W3 Button (見出し → アンカージャンプ)



サイドバーの行ごとに 1 つのボタン。



各サイドバーエントリはボタンで、現在のアンカーを設定しますが、視覚的には リストエントリのように見えます。次に Observer 変数が、対応するアンカーへのスクロールジャンプを 0.3 秒のソフトアニメーションでトリガーします。ジャンプ後、同じアンカーへの次のクリックが再び発火するように、アンカー値がリセットされます (そうでなければ、値が変わっていないため、Observer は再度トリガーされません)。

簡単に言うと

クリックすると、右側のテキストの適切な場所に移動します。

W4 ScrollView (詳細コンテンツ)



右カラム。



スクロール可能で垂直に積み重なるコンテンツ領域で、長いガイドが簡単に 200 以上の Markdown ブロックを持つ可能性があるため、Lazy レンダリングが使用されます — 非 Lazy バリエントはすべてを同時にインスタンス化します。各ブロックは独自の ID を取得します。見出しアンカー (ジャンプ可能な H1~H3 用) またはインデックス プレースホルダーのいずれか。最大幅は 720 pt、パディングは水平 32、垂直 24 で、長い行が読みやすいレイアウトを保ちます。テーブルは水平スタックと区切り線でセル単位でレンダリングされます。インラインコードは組み込み Markdown エンジンによって。実際のコードブロックは現在段落として扱われます — Help ウィンドウの 既知の制限。

簡単に言うと

実際のヘルプテキスト。スクロール可能、読みやすい幅、明確なタイポグラフィ。

Keyboard Shortcuts (W5～W6)

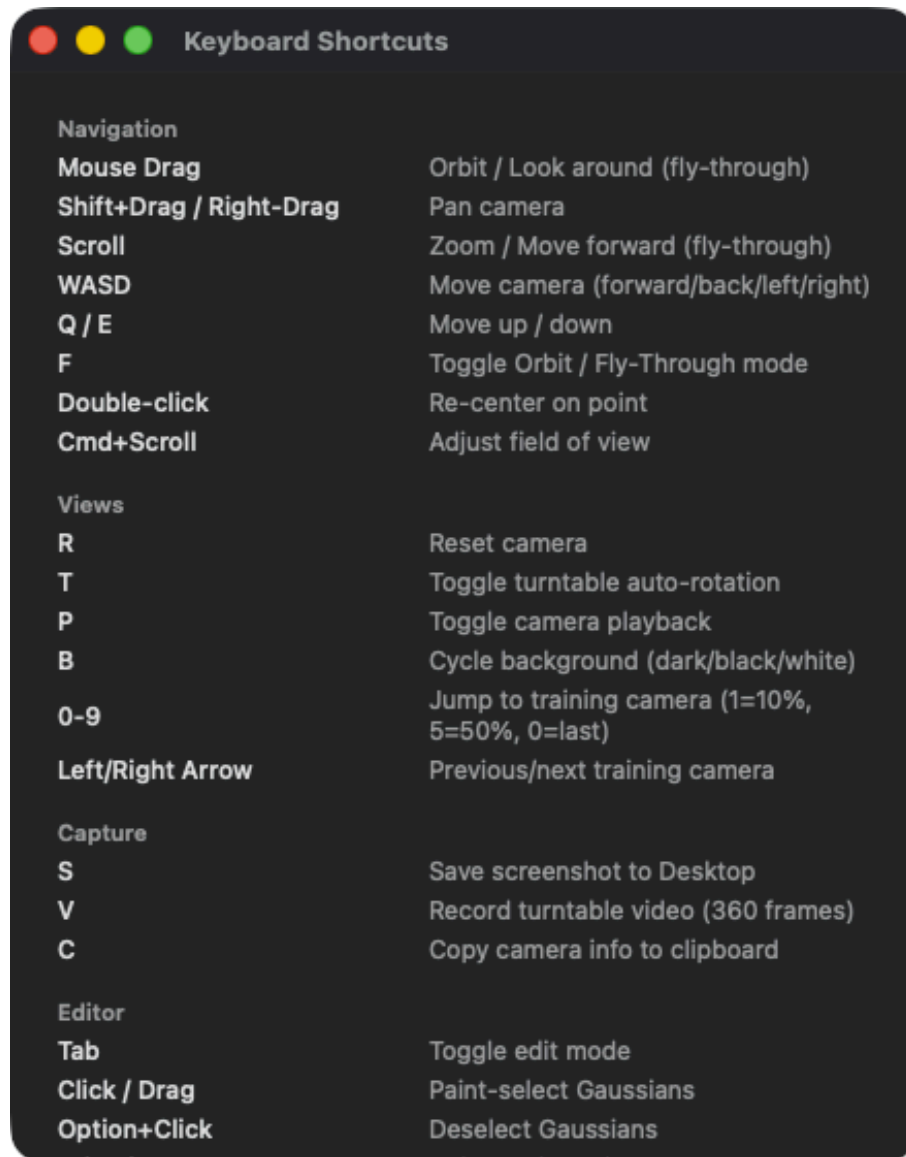


図 18: Keyboard Shortcuts ウィンドウ — Navigation/Views/Capture/Editor/Training の 5 グループ、左にホットキーカラム、右に説明

画像で見えるもの 5つのセクションでの静的リファレンスリスト。**Navigation:** Mouse Drag (Orbit/Fly)、Shift+Drag/Right-Drag (Pan)、Scroll (Zoom)、WASD (Fly-Through 移動)、Q/E (Up/Down)、F (Orbit/Fly 切り替え)、ダブルクリック (再センター)、Cmd+Scroll (FoV 調整)。**Views:** R (Reset Camera)、T (Auto-Rotation)、P (Camera Playback)、B (Background-Cycle)、0～9 (Training-Cam 1=10%/5=50%/0=last にジャンプ)、左/右矢印 (Prev/Next Cam)。**Capture:** S (Desktop に Screenshot)、V (Turntable Video)、C (Copy Camera Info)。**Editor:** Tab (Edit モード)、Click/Drag (Paint-Select)、Option+Click (Deselect)、X / Delete (選択を削除)、Cmd-Z (最後の削除を取り消し)、[/] (ブラシサイズの縮小/拡大)、Esc (選択解除)。**Training:** Start、Pause/Resume、Cancel、Continue +5K/+10K/+20K M9～M14 のメニューショートカット経由。

何か: すべてのキーボードショートカットのシンプルな静的概要 – Navigation、Views、Capture、Editor、Training。コンテンツはハードコードされており、Markdown ロードはありません。

いつ開くか Viewport で何かを行う最も速い方法を探している場合。WASD Fly-Through、カメラリセット用 R、背景循環用 B – すべてここにあります。

W5 ScrollView (コンテンツ領域)



場所

Help → Keyboard Shortcuts (⌘/).



技術詳細

垂直リストが含まれるシンプルなスクロール領域。20 のパディングが全周囲、サイドバーナビゲーションツリーなし (リストは十分短い)。コンテンツは 5 つのセクション (Navigation、Views、Capture、Editor、Training) にグループ化されています。キーの組み合わせごとに、両カラムに翻訳可能なテキストを持つ 1 行。左カラム (キーコード) は 180 pt 幅に固定されているため、右側の説明は垂直に整列したまま。スクロール以外の操作はありません – 行をクリックしても何もトリガーされません。ショートカットは、メニューと Viewport の実際のキーボード修飾子です。

簡単に言うと

すべてのショートカットキーのテーブル。素早く参照するための静的な チートシート。

W6 VStack (ショートカットセクション)



場所

ScrollView 内。



技術詳細

16 pt の間隔で左揃えにスタックされたセクション。5 つのセクション内に それぞれ見出し + 行のシーケンス。見出しはセカンダリサブヘッドラインスタイルを使用 します – セクションはナビゲート可能である必要がないため、Title フォーマットは 意図的に使用 しません。コンポーネントがすべての macOS バージョンで変更なしで実行され、ファイルが読みやすいままになるように、コンテンツは意図的にフラット (Disclosure、Search、Filter なし) です。

簡単に言うと

機能 (Navigation, Views, Editor など) ごとのキーのグループ化。

Manage Storage (W7~W12)

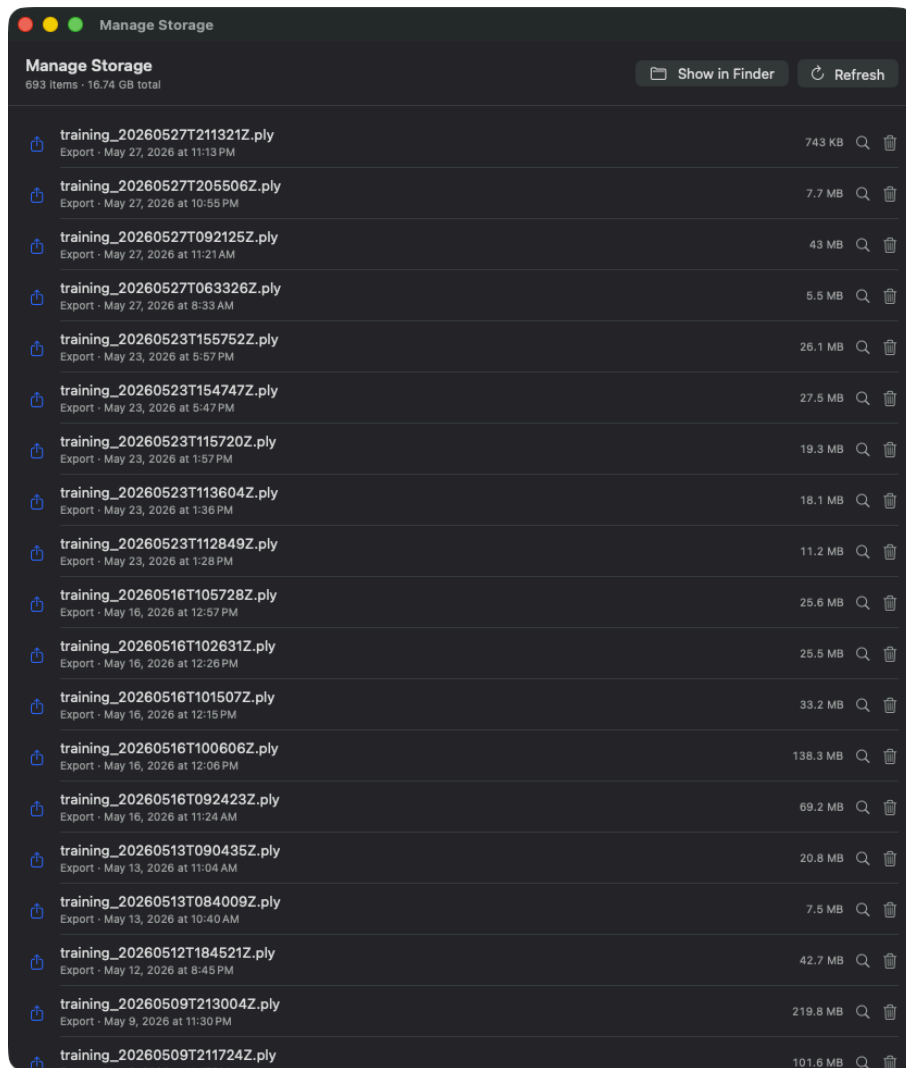


図 19: Manage Storage ウィンドウ — ヘッダーに「693 items · 16.74 GB total」、日付でソートされたエクスポート PLY ファイルのテーブル、各々形式ビルド + ファイル名 + サイズ + 日付

画像で見えるもの RadianceKit が管理するすべてのファイルのテーブルビュー。ヘッダーは 693 アイテム、16.74 GB の合計サイズをカウントします。上部のツールバー: 「Show in Finder」 + 「Refresh」。各行: PLY アイコン、ファイル名 (例えば training_20260527T211321Z.ply)、エクスポート日、サイズ (7 KB から 218 MB まで変動)、虫眼鏡アイコン (Reveal)、ゴミ箱アイコン (Move to Trash)。ファイルは日付でソートされ、最新のものがあります。このデモ撮影では、`--benchmark` で多く作業したため、PLY エクスポートが優勢です。

何か: RadianceKit が `~/Documents/RadianceKit/` 以下に配置するすべてのディスク使用量概要 — ログ、エクスポート、シーン、キャプチャバンドル (iOS コンパニオンから)、インポート (入力画像のステージングコピー)。エン트리ごとにバイト単位のサイズと 2 つのボタン: 「Finder で表示」と「ゴミ箱に移動」。自動クリーンアップは NOT — アプリ自体は何も削除しません。エン트리ごとに決定します。

いつ開くか ディスクがいっぱいになったとき。特にログが蓄積されます (トレーニング試行ごとに 1 つの JSONL、加えて `_qualityMetrics.json`)。エクスポートも当然 (PLY 100% 生データ、エクスポートごとに 1 つ)。クラッシュ後、インポートステージングディレクトリがまだ入力画像の古いコピーを保持している場合にも便利です (`dev_v549f-needle-reduction.md` の「Disk-pressure incident」参照)。

W7 ボタン「Show in Finder」



場所

Storage ブラウザウィンドウの右上のヘッダー。



技術詳細

Finder で RadianceKit ディレクトリ全体 (`~/Documents/RadianceKit/`) を開き、フォルダ構造を直接見て、Finder 自体で操作することもできます。アクションは新しい Finder ウィンドウを開き、アプリ Sandbox コンテナに切り替えません — `~/Documents/RadianceKit/` はアプリに通常アクセス可能な Documents ドメインで、Sandbox コンテナパスではありません。

簡単に言うと

Finder でディレクトリを開き、ファイルを自分で操作できるようにします。

W8 ボタン「Refresh」



場所

ヘッダー、Finder ボタンの隣。



技術詳細

大きなディレクトリツリーのスキャンが UI をブロックしないように、ユーザー開始の非同期タスクで実行されるバックグラウンドスキャンをトリガーします。実際のウォークは、各既知のサブフォルダ (Logs、Exports、Scenes、Captures、Imports) を通過し、直接の子ごとに Storage エントリを生成します。エントリごとに、再帰サイズが決定されます — 論理ファイルサイズへのフォールバックとともに、実際のディスク使用量 (APFS ハードリンク共有を含む) を優先します。

簡単に言うと

Finder で間あなたに何か削除または追加した場合、リストを再読み込みします。

W9 List (Storage エントリ)

ヘッダー下のメインコンテンツ。



行ごとに次のレイアウトを持つリスト: カテゴリ固有の SF Symbol アイコン (Logs 用ドキュメント、Exports 用アップロード矢印、Scenes 用キューブ、Imports 用トレイ)、名前 + サブタイトル (Kind ラベル + フォーマットされた変更日)、右側のバイトカウンター (右揃え、monospace)、Reveal ボタン (虫眼鏡シンボル)、Trash ボタン (ゴミ箱)。ソート: 主に種別 (最初に Scenes、次に Exports、Logs、Captures、Imports、Other)、次に変更日降順 (最新が上)。スキャンが実行中の場合、その場所には代わりに「Scanning...」進捗が表示されます。何も見つからない場合、Tray アイコンを持つ空状態表示。

簡単に言うと

タイプと最新性でソートされたすべての RadianceKit データのリスト。エントリごとにサイズを確認し、直接削除できます。

W10 行ボタン「Reveal in Finder」

行ごと、右側の虫眼鏡シンボル。



Finder を開き、特定のアイテム (ファイルまたはフォルダ) を選択します。W7 との違い: W7 はルートディレクトリを開きます。W10 はこの1つのエントリを正確にマークします。実用的なワークフロー: 大きなエントリを識別し、虫眼鏡をクリックし、例えば外部ボリュームにコピーします。

簡単に言うと

素早く見つけられるように、Finder で直接このエントリにジャンプします。

W11 行ボタン「Move to Trash」



行ごと、虫眼鏡の隣の右側のゴミ箱シンボル。



確認ダイアログボックス (W12) をトリガーします。確認後にのみ、macOS 標準の「ゴミ箱に移動」操作が実行されます (つまり、可逆的、直接削除ではありません)。ゴミ箱への移動が成功した後、エントリはリストから削除され、合計バイトカウンターが更新されます。エラーが発生した場合、モーダルエラーダイアログが表示されます。

簡単に言うと

エントリをゴミ箱に移動します。
ダイアログが事前に尋ねます。

W12 ConfirmationDialog (削除確認)



W11 によってトリガーされ、macOS シートとして表示されます。



動的タイトル「Delete <name>?」と、エントリがゴミ箱に着地し、そこから復元可能であることを明示的に指摘するメッセージ行を持つ標準確認ダイアログ (ゴミ箱が空にされるまで)。2つのボタン: 破壊的アクションとしての「Move to Trash」 (赤で表示) と自動 Esc バインディングを持つ「Cancel」。可逆的削除に対する macOS 標準であるため、ダイアログはアプリ全体ではなくこのウィンドウのみをブロックする意味で非モーダルです。

簡単に言うと

削除前の安全質問。「Move to Trash」はゴミ箱が空にされない限り可逆的です。

Pareto Dashboard (W13~W22)

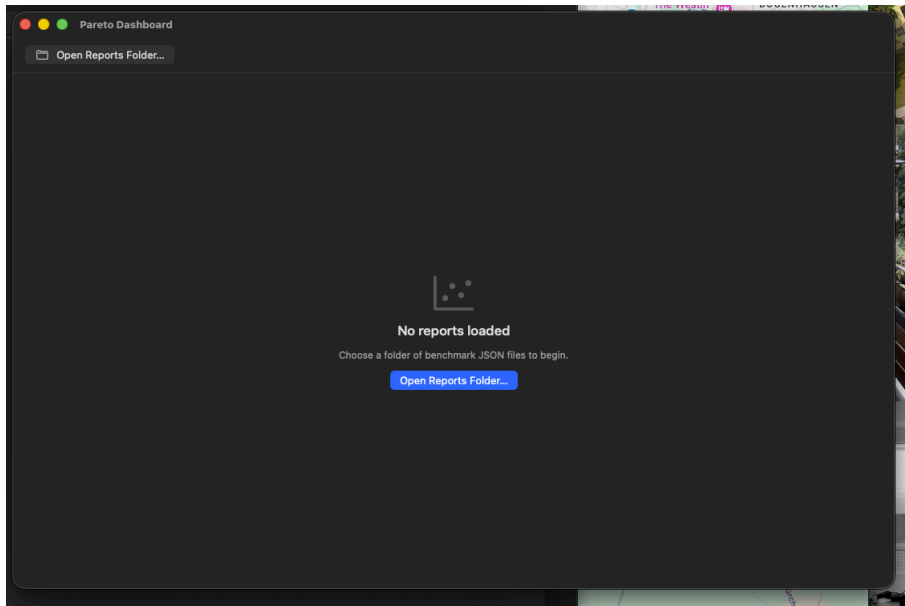


図 20: Pareto Dashboard — レポートインポート前の空状態

空状態 (最初に開いた後) — 「Open Reports Folder...」 コールトゥアクション付き 空状態。トレーニングレポートが読み込まれるとすぐにデータポイントが表示されます。次のショット参照。

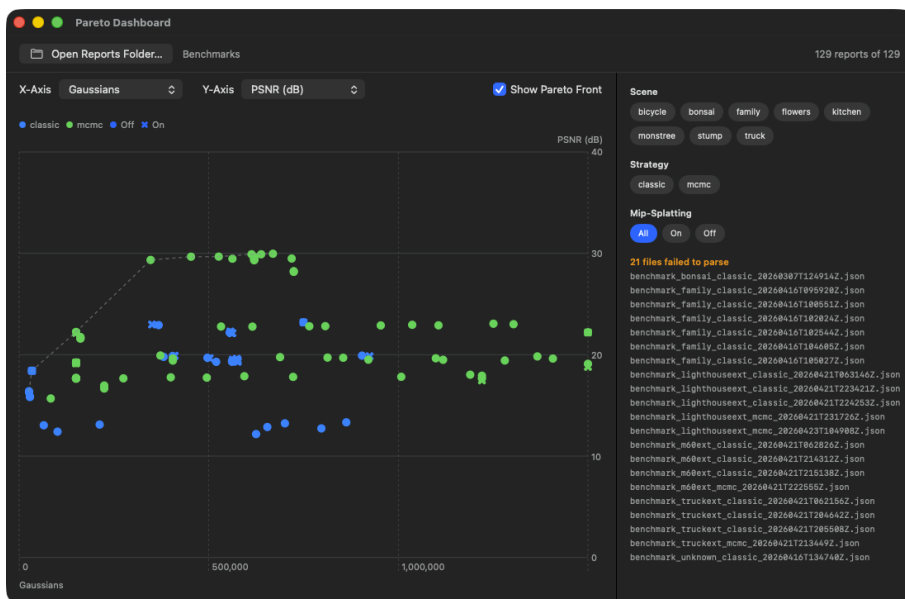


図 21: 129 ベンチマークレポートがロードされた Pareto Dashboard — Pareto Front 付き Gaussian 対 PSNR、Scene/Strategy/Mip フィルター

画像で見えるもの ヘッダーツールバーは「129 reports of 129」を表示します (選択されたフォルダ内のすべてのレポートが正常に解析されました — 21 の追加ファイルは古いフォーマットのため解析できませんでした。右側のヒントリスト参照)。軸: X 軸ピッカーは Gaussians、Y 軸ピッカーは PSNR (dB) 上。Scatter Plot: 緑のポイント = Classic 戦略、青のポイント = MCMC。破線 Pareto Front 線は、最高達成された PSNR 値に沿って実行さ

れ、約 500K Gaussian から PSNR \approx 30 dB でプラトーになります。右側のフィルターチップ: 7 シーン (bicycle、bonsai、family、flowers、kitchen、stump、truck)、2 戦略 (classic、mcmc)、3 Mip-Splatting オプション (All、On、Off)。現在、すべてのフィルターが開いているため、密なポイントクラスター。

何か: マルチ実行比較ツール。過去に複数のシーンまたは同じシーンを異なる プリセットでトレーニングしました – これらの各トレーニング実行は (`--benchmark` を渡したか、ベンチマーク機能経由で呼び出した場合) Final PSNR、SSIM、LPIPS、Gaussian カウント、Wallclock 時間を含む JSON レポートファイルを生成します。Dashboard は そのようなレポートのフォルダ全体を同時に読み込み、選択可能な軸を持つ 2D Scatter としてプロットします。さらに、Pareto Front (支配されない点のセット) が破線として 描画されます。

いつ開くか 少なくとも 3 つまたは 4 つのトレーニングレポートを作成した後。少ないポイントでは、Frontier ラインは意味がありません。典型的なユースケース: 屋外シーンを再構築しようとし、順次 P3 Balanced (Classic)、P4 Quality (Classic)、P7 MCMC Quality、P9 Outdoor (tuned) を試しました – 今、どの構成が秒あたりの 最良のトレーニング時間 PSNR を提供するか、または特定の PSNR に必要な最小の Gaussian を提供するかを知りたいです。

どのように解釈 両軸は自由に選択可能 (X 軸: `psnr`、`ssim`、`lpips` など。Y 軸も同じ)。ParetoFront2D.indices の Pareto Front ロジックは、メトリックごとに「小さい = 良い」(例えば LPIPS、Loss、Time) または「大きい = 良い」(PSNR、SSIM) かを知っています – したがって、線は軸の選択に応じて左下から右上、または左上から右下に実行され、常に最高達成された組み合わせに沿って。点は、他の点が両方の次元で少なくとも同じくらい良くない場合、Pareto 最適です (つまり、他の点が支配しません)。Pareto 最適点は線上にあり、他の点は線の右/上 (軸の方向に応じて) にあります。線上の点は「最高のプリセット」の本当の候補です。線から遠い点は無駄なトレーニング時間です。

フィルターチップ 選択を特定のシーン (例えば屋外実行のみを比較したい場合)、特定の戦略 (Classic または MCMC)、または Mip-Splatting オン/オフ (Phase Q1.5 以降に関連、ここで Mip は opt-in Advanced フラグとして残ります) に制限できます。

ワークフロー例 「truck」シーン用の 3 つのレポートが `~/Documents/RadianceKit/Reports/` の下にあります: Run A (P4 Quality、40K 反復、524K Gs、105 秒、PSNR 23.4)、Run B (P7 MCMC、200K 反復、150K Gs、693 秒、PSNR 24.6)、Run C (P9 Outdoor、100K 反復、1.25M Gs、312 秒、PSNR 25.8)。X 軸を trainingTime に、Y 軸を PSNR に設定します。Run B は右上に、Run C はさらに右上に、Run A は左下にあります。Pareto Front は A と C を接続します – 両方とも 支配されていません。Run B は「失われ」ています (C は Time と PSNR の両方でより良い)。洞察: 「truck」では MCMC デフォルトは価値がありません。高速 + OK (A) または長く + 非常に良い (C) のいずれか。C からの構成を独自プリセットとして保存 (Inspector \rightarrow I1 Save Preset)。

次のアクション: 最良の構成をプリセットとして保存します。具体的に: Pareto 点を見て (Hover でツールチップに PSNR/SSIM/LPIPS/Gs/Time が表示)、Time vs Quality トレードオフが最も適しているものを決定し、対応するレポートを開き (ファイル名に Run タイムスタンプが含まれます)、そのトレーニング構成を新しい実行にコピーするか、次のトレーニングセッション後に Inspector 経由でプリセットとして保存します。

W13 ボタン「Open Reports Folder...」

場所

左上のツールバー。



技術詳細

「Select a folder containing benchmark .json reports」プロンプト付き フォルダ選択ダイアログを開きます。確認後、フォルダ内のすべての .json ファイルを 順次解析するバックグラウンドタスクが実行されます。欠陥のあるレポート (破損 JSON、誤ったスキーマ) が収集され、サイドバーの下に「N file failed to parse」として表示されます — クラッシュなし。最初のロードが実行中の場合に 2 回目のクリックが発生すると、以前のタスクが中止され、2 つの結果が同時に状態に書き込まれないようにします。

CLI 経由でも: `--dashboard-dir /path/to/reports` はアプリ起動時にフォルダをロードします。

簡単に言うと

ベンチマークレポートが配置されているフォルダを選択します。標準パスは `~/Documents/RadianceKit/Reports/`。次に、すべての JSON を一度に読み込みます。

W14 ピッカー「X-Axis」

場所

チャートの上、左。



技術詳細

Dashboard モジュールのすべての利用可能なメトリック軸 (PSNR、SSIM、LPIPS、Gaussian カウント、トレーニング時間など) を持つメニューピッカー。デフォルトは Gaussian カウントです。切り替え時、Hover ポイントがリセットされます。以前にハイライトされた位置が、軸変更後の古い軸座標系で意味をなさないためです。ピッカーはコンテンツ幅に制限され、幅全体に引き伸ばされないようにします。

簡単に言うと

水平軸にどのメトリックがあるべきか。通常「トレーニング時間」または「Gaussian カウント」、それらが比較したい「コスト」だからです。

W15 ピッカー「Y-Axis」

場所

チャートの上、X 軸の隣。



技術詳細

W14 と同じですが、デフォルトは PSNR です。軸選択は独立して保存されるため、ユーザーはナンセンスな組み合わせ (X=PSNR、Y=PSNR – すべての点を対角線に配置します) を選択することもできます。しかし、そのような組み合わせはキャッチされません。意図的な決定です。「SSIM vs PSNR」の比較は、メトリックがどれだけ一貫して動作するかを見るために十分興味深いからです。

簡単に言うと

垂直軸に何があるか。通常、品質尺度として「PSNR」または「SSIM」。

W16 トグル「Show Pareto Front」

場所

軸ピッカーの右隣。



技術詳細

標準 macOS トグル。アクティブな場合、Pareto チャートでは点群に加えて、計算された 2D Pareto Front の線が描画されます。スタイル: 破線 (ダッシュパターン 4-4)、グレー半透明、線太さ 1.5 pt。Pareto 計算はメインスレッドで実行されます – 典型的なレポート数 ($\leq \sim 50$) では問題なく高速です。トグルがオフの場合、線は省略され、生の点のみが表示されます。

簡単に言うと

「これまで最良の」点を通る線を表示します。線が邪魔な場合 (例えば、個々のトレードのみを比較したい場合)、それをオフにしてください。

W17 チップ「Scene」フィルター

Dashboard ウィンドウの右サイドバー。



ロードされたレポートに表示される各シーン用のフィルターチップ。幅が使い果たされるとすぐに、チップを複数行に自動的に再パッキングする独自の Flow Layout。アクティブなチップはアクセント背景を取得し、非アクティブなチップは中立的な標準 マテリアル背景を取得します。複数選択が可能 (Set セマンティクス)。チップが選択されていない場合、すべてのシーンが「通過」と見なされます — つまり、Set ロジックは「空の選択 = すべて」であり、「空の選択 = 何もなし」ではありません。

簡単に言うと

シーン名をクリックすると、点がこのシーンだけにフィルタリングされます。複数選択可能。空 = すべてのシーン。

W18 チップ「Strategy」フィルター

サイドバーの Scene フィルターの下。



W17 と同じですが、トレーニング戦略用 — 通常、ベンチマークレポート JSON の Strategy フィールドから導出された 2 つの値「classic」と「mcmc」。両方の戦略のレポートを混在させ、1 種類のみを表示したい場合 (例えば、Classic をすでに除外しているため「MCMC 実行のみを表示」) に役立ちます。

簡単に言うと

Classic または MCMC でフィルター。デフォルトでは両方がアクティブです。

W19 チップ「Mip-Splatting」フィルター

場所

サイドバーの Strategy フィルターの下。



技術詳細

3 値フィルター (W17/W18 のような Set ではなく): 「All」 / 「On」 / 「Off」。背景: Mip-Splatting は Phase Q1.5 で実験的マルチスケール改善として評価され、最終評決は「全体的に良い勝利はない。opt-in フラグとして保持」でした。Mip on/off 比較を行う場合、しばしば非常に明確に区別したいです。したがって、状態「すべて通す」、「Mip オンのみ」、「Mip オフのみ」を持つ専用の三値フィルター。サイドバーセクションは、データセットに少なくとも 1 つの Mip レポートと少なくとも 1 つの非 Mip レポートがある場合にのみ表示されます (そうでなければ、フィルタリングは意味がありません)。

簡単に言うと

Mip-Splatting オン/オフを比較したい場合、ここに 3 分割フィルター。そうでなければ無視。

W20 ChipButton (フィルタートグル、all/on/off)

場所

ヘルパーコンポーネント、W17/W18/W19 で使用。



技術詳細

ミニマリストのボタンラッパー。コンテンツ: Caption フォントグレードのラベルテキストとパディング水平 10 / 垂直 5。条件付き背景: アクティブな場合 → 白いテキストでアプリのアクセント色。それ以外、黒いテキストで中立的な標準 マテリアル背景。形状はカプセル (薬丸状)。Capsule マテリアルがシステム境界線でオーバーレイされないように、Plain ボタンスタイル。

簡単に言うと

丸いフィルターボタン自体。視覚的に iOS タグのように。

W21 チャート (Pareto Scatter)



場所

Dashboard の中央領域。



技術詳細

2つのレイヤーを持つ Swift Charts ダイアグラム: 1. レポートあたり1つのポイント — 選択された X と Y メトリックからの位置、戦略による色、Mip ステータスによる Symbol。Symbol サイズ通常 80、ハイライトされた 200 (ID が現在 Hover された レポートに対応する場合)。2. Pareto Front 用の線、トグルがオンの場合のみ。

チャートオーバーレイ: 透明な長方形がマウス移動を登録します。フレームごとに、プロットフレーム内の最も近いユークリッドポイント位置が決定され、距離が 24 px 未満の場合は Hover されたレポートが更新されます (そうでなければリセット)。これで、クリックせずにツールチップを取得できます — Hover で十分です。

簡単に言うと

実際の散布図。各点はトレーニング実行です。詳細ツールチップ用に Hover。

W22 ツールチップ (Hover 詳細)



場所

チャートの下、Hover 時に表示。



技術詳細

水平スタック: シーン名 (Headline)、Strategy タグ (Caption)、区切り線、次に PSNR/SSIM/LPIPS/Gs/Time メトリックがそれぞれ小さな垂直グループ (ラベル + monospace 値) に。Mip が有効化されていた場合、アクセント色の追加の「Mip」Capsule タグも。背景は半透明 Blur、8 pt 半径の角丸長方形。マウスが実際にポイント上にある場合にのみ表示。離れると自動的に消えます。

簡単に言うと

マウスをポイント上に置くと、下に詳細カードが表示されます。すべての品質メトリックと実行構成を一度に表示。

Holdout Analysis (W23~W29)

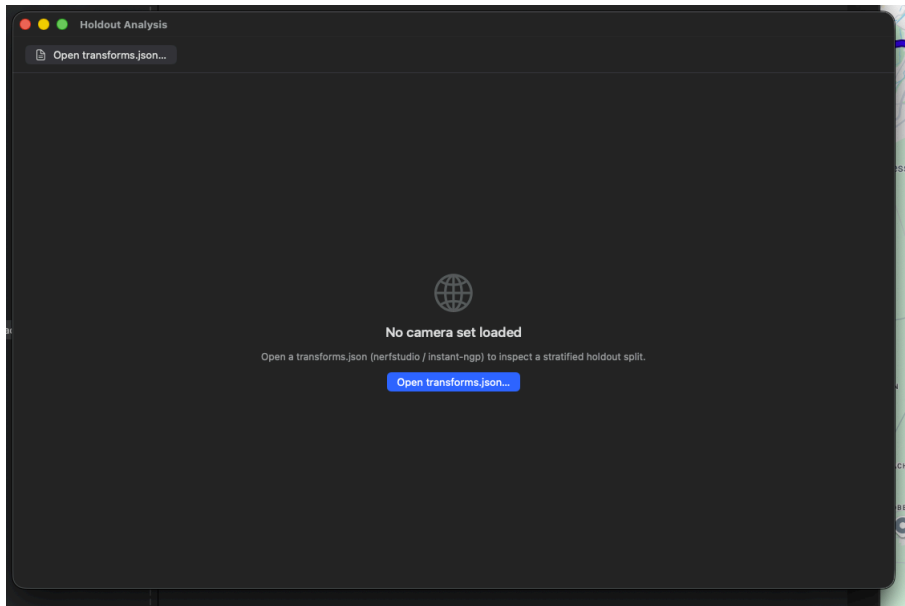


図 22: Holdout Analysis — transforms.json をロードする前の空状態

「Open transforms.json...」 コールトウアクション付き空状態と空状態。NeRF Studio と Instant-NGP 形式を受け入れます。

空状態 (最初に開いた後) — カメラマーカーは transforms.json がロードされるとすぐに表示されます。次のショット参照。

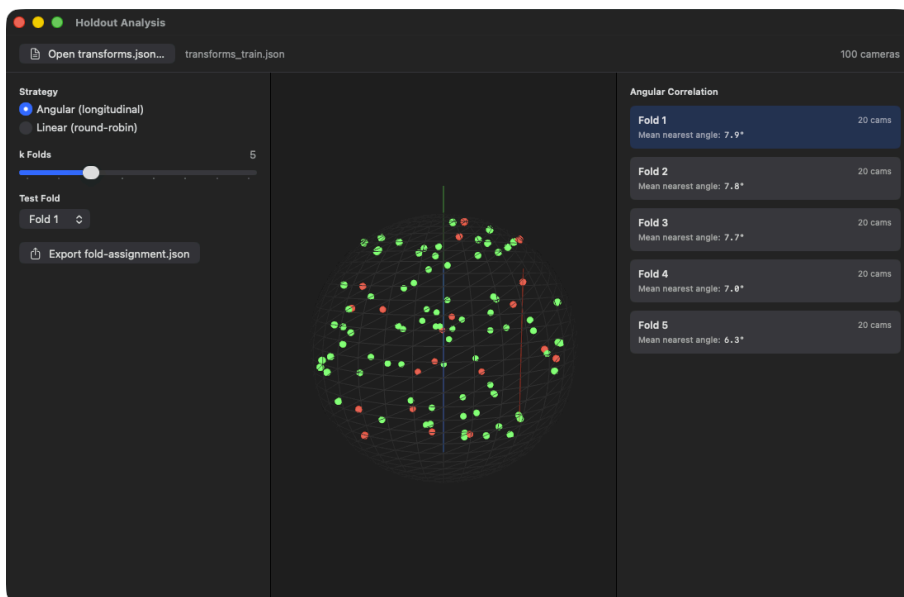


図 23: 100 NeRF-Blender Mic カメラ、5 fold × 20 カメラ、Angular 戦略 アクティブの Holdout Globe

画像で見えるもの ヘッダーはロードされたファイル (transforms_train.json) と Cam カウント (「100 cameras」) を表示します。左サイドバー: 2つのオプションを持つ Strategy ピッカー — アクティブな Angular (longitudinal) (Fold を球体上の経度/緯度セクターに合わせます。各 Test fold が幾何学的に密になるように) vs Linear (round-robin) (順序ベース、

すべての k 番目のフレームを Test セットとして)。k-Folds スライダーは 5、Test Fold ピッカーは Fold 1。Export ボタンは Nerfstudio/Instant-NGP 用の `fold-assignment.json` を生成します。中央パネル: 全 100 カメラの 3D グローブ 投影 – 緑のポイント = Train、赤のポイント = 現在の Test fold (20 カメラの Fold 1)。右サイドバー (Angular Correlation): Fold あたり 20 Cams + Mean Nearest Angle (Fold 1: 7.9°、Fold 2: 7.8°、Fold 3: 7.7°、Fold 4: 7.0°、Fold 5: 6.3°) – 小さい値は、この Fold 内のカメラが互いに密接に位置していることを意味します。つまり、Holdout 分割は空間的にコヒーレントです。

何か: クロス検証ロジックを備えたカメラ配置の 3D ビジュアライザー。 `transforms.json` (カメラポーズ用の Nerfstudio / Instant-NGP の標準形式) をロードし、アプリはすべてのカメラを読み取り、視線方向を単位球に投影し、仮想 グローブ上の小さな球マーカーとして表示します。次に、カメラを k Fold (選択された 戦略: angular または linear) に分割し、トレーニング部分を緑、テスト部分 (Holdout) を 赤でマークし、Fold ごとに Angular Correlation スコアを計算します。これは、Test fold が視線空間でトレーニング fold からどれだけ離れているかを示します。

いつ開くか Holdout 評価を行いたいとき – つまり: モデルは未見の 視点にどれだけ良く一般化するか? トレーニングの標準は「8 番目ごとのビューを Holdout として」(Mip-NeRF360 規約) ですが、これは非常に線形な分割です。例えば 画像が時間的にクラスター化されている場合 (まずオブジェクトの一方の側、次に もう一方)、「8 番目ごと」は代表的ではありません – ランダムなシーケンス位置が Test に着地しますが、すべての隣接が Training にあります。それは簡単すぎます。「angular」では代わりに視線空間にわたって層別化されます: 各 Fold には Orbit の 全ての領域からのカメラが含まれるため、Test は本当に一般化ギャップをテストします。

どのように解釈 Angular vs Linear: - Angular (標準): カメラを Longitudinal 角度 (Y 軸周りの ϕ 座標) で k 個の等しいセクターに分割します。Fold 0 は $\phi \in [0^\circ, 360/k^\circ)$ のカメラ、Fold 1 は次のなど。利点: 各 Fold が Orbit の一部をカバーします。Test fold は空間的にコンパクトですが、ワールドデータセット全体に広く分布しています。古典的なオービット撮影に良い。- Linear (Round-Robin): Fold インデックス = $(\text{image_index} \bmod k)$ 。これはシンプルな「 k 番目ごと」分割。画像の順序に空間バイアスがない (例えばランダムにソートされたドローン撮影) 場合に機能します。画像が時間的に クラスター化される場合、うまく機能しません。

3D グローブで、すぐに見えます: 緑のポイント (Training) と赤のポイント (Test)。赤のポイントがすべて角にクラスター化されている場合、Holdout は悪いです (良い 一般化テストではありません)。緑の間に均等に位置している場合、それは良いです。Fold ごとの Angular Correlation スコア (右サイドバー、度数) はさらに、より小さい 値 = Test がトレーニングに近い (各 Test カメラに近いトレーニングカメラがある、簡単なテスト) を伝えます。より大きい値 = Test がトレーニングから離れている (より難しい一般化)。

ワークフロー例 Truck シーンを 251 枚の画像で撮影し、メニュー項目 M33 (Export SfM `transforms.json`) 経由で nerfstudio ファイルをエクスポートします。Holdout ウィンドウ (⇧H) を開き、「Open transforms.json...」経由で JSON をロードし、グローブを見ます。 $k=5$ (デフォルト) は 5 Fold を提供します。「Fold 3」をクリック – 赤のマーカーがある程度均一かを確認します。はいの場合: 「Export fold-assignment.json」で、エクスポートされたファイルを Reports フォルダに配置し、 `--benchmark` (または対応する Inspector 設

定) を使用した次のトレーニング実行で、標準の「8 番目ごと」の代わりに正確にこの Fold 分割が Test Holdout として 使用されます。

W23 ボタン「Open transforms.json...」

 場所

左上のツールバー。

 技術詳細

JSON ファイルに制限されたファイル選択ダイアログを開きます。確認後、Holdout モジュールはファイルをロードします。ローダーは、nerfstudio 形式 (カメラ内部パラメータと画像パスと変換マトリクスを持つフレームのリスト) と instant-ngp 形式 (同じ構造) の両方を解析します。フレームごとに、視線方向が変換マトリクスから抽出され (カメラローカル基底の z 軸)、保存されます。解析が失敗した場合、Status 領域にエラーメッセージが表示されます。

CLI 経由でも: `--holdout-file /path/to/transforms.json` は、ロード済みのファイルで直接ウィンドウを開始します。

簡単に言うと

カメラポーズ JSON をロードします。標準は Nerfstudio と Instant-NGP エクスポートです。RadianceKit 自体は、メニュー → Export → SfM 経由で transforms.json をエクスポートできます。

W24 ピッカー「Strategy」 (angular/linear)

 場所

左サイドバー、上部。

 技術詳細

2 つのオプションを持つラジオピッカー: Angular と Linear。戦略の切り替えは、fold の自動再計算をトリガーします。視線方向は球面上の 3D 単位ベクトルのリストです。Angular 戦略はそれらを Longitudinal 角度 ϕ に投影してソートし、Linear 戦略はフレームインデックスにわたって単にモジュール分割を行います。

簡単に言うと

均等な Orbit 撮影には Angular (標準、安全)、画像が空間的にクラスタ化されない場合にのみ Linear。

W25 スライダー「**k Folds**」

場所

左サイドバー、中央。



技術詳細

3 から 10 までのスライダー、ステップ幅 1。変更時、Fold 計算が自動的に再トリガーされ、Folds リスト、Training/Test インデックス、Fold ごとのスコアがすぐに再計算されます。選択された値は、ラベルの右側に monospace 数字テキストとして表示されます。

経験則: $k=5$ が標準 (Fold ごとに 20% のテストを提供、これがクロス検証の標準) です。非常に多くのデータがあり、統計的有意性のためにより多くの Fold が必要な場合は $k=10$ 。データが少ない場合は $k=3$ 。

簡単に言うと

いくつの Fold を分割に。5 が標準で、ほぼ常に適合します。

W26 ピッカー「**Test Fold**」

場所

左サイドバー、k スライダーの下。



技術詳細

メニューピッカー。オプションは動的に $0..<k$ 、ラベル「Fold 1」から「Fold N」(つまり UI で 1-indexed、内部では 0-indexed)。以前に選択されたインデックスが k 以上の場合 (例えば k を 10 から 5 に減らしたため)、自動的に 0 にリセットされます。選択された Test fold はグローブで赤で表示され、他のすべては緑です。

簡単に言うと

どの Fold が現在 Test fold か。クリックして、グローブで各 Fold がどのように見えるかを確認できます。

W27 ボタン「Export fold-assignment.json」

場所

左サイドバー、下部。



技術詳細

デフォルトファイル名 `fold-assignment.json` を持つ保存ダイアログを開きます。確認後、Holdout モジュールは現在の分割を JSON スキーマ (フレームごとの Fold 割り当て + Strategy メタブロック) にエンコードします。このファイルは、`--benchmark` を使用した次のトレーニングに渡すことができ、同じ Holdout が最終メトリック評価に使用されます。書き込みエラーはエラーテキストとして表示されます。成功は緑のテキストで「Saved to (filename)」として。

簡単に言うと

現在の Train/Test 分割を JSON として保存します。このファイルをトレーニング時に直接渡すと、同じ Test セットが再度使用されません。

W28 SCNView (3D Camera Globe)

場所

Holdout ウィンドウの中央パネル。



技術詳細

SceneKit グローブビュー。シーンは: ワイヤフレーム球 (半径 1.0、36 セグメント、暗いグレー)、3 つの色付き軸スタブ (X/Y/Z 用の赤/緑/青、それぞれ 1.2 長)、対応する視線方向位置の単位球上のカメラごとの小さなマーカー球 (半径 0.03) (ワイヤフレーム球内に消えないように、わずかに外側) で構成されます。マーカーは Fold 変更ごとに再構築されません — 再構築は、フレームリストが変更されたとき (つまり、新しい JSON がロードされたとき) にのみ必要です。代わりに、更新ごとにマテリアル色のインプレース更新が実行されます: Test インデックスには赤、Training には緑、どちらでもない場合は明るいグレー。これにより、 $N > 1000$ カメラでもスライダーティックがパフォーマンスを維持します。

カメラコントロールが有効化されています — マウスでグローブを回転、ズーム、パン できます。照明により、マーカーが平らに見えないようにします。背景は暗いグレーです。

簡単に言うと

カメラ位置を持つ 3D グローブ。緑 = Training, 赤 = Test, 明るいグレー = 未割り当て (発生しません、すべてのカメラがどこかに属します)。マウスで、グローブを回転およびズームできます。

W29 FoldCard (タップで Fold を選択)



場所

右サイドバー、「Angular Correlation」セクション。



技術詳細

Fold ごとに 1 つのカードビュー — 6 pt 半径の角丸長方形、パディング 10、2 行の垂直レイアウト (上「Fold N」+ カメラ数、下「Mean nearest angle:」+ 度数値)。背景色は条件付き: アクティブな Fold = アクセント色半透明、非アクティブ = 中立的な標準マテリアル。タップで Fold を選択し、グローブはライブで再着色します。

「Mean nearest angle」スコアは、Test カメラごとの最も近いトレーニングカメラまでの平均最小角度です (内部的にラジアンで計算され、UI では度数で表示)。

簡単に言うと

Fold ごとに、右側にカメラ数と次のトレーニングカメラまでの平均距離を持つ小さなカード。それをクリックすると、この Fold が Test として選択されます。

BayesOpt Console (W30~W39)

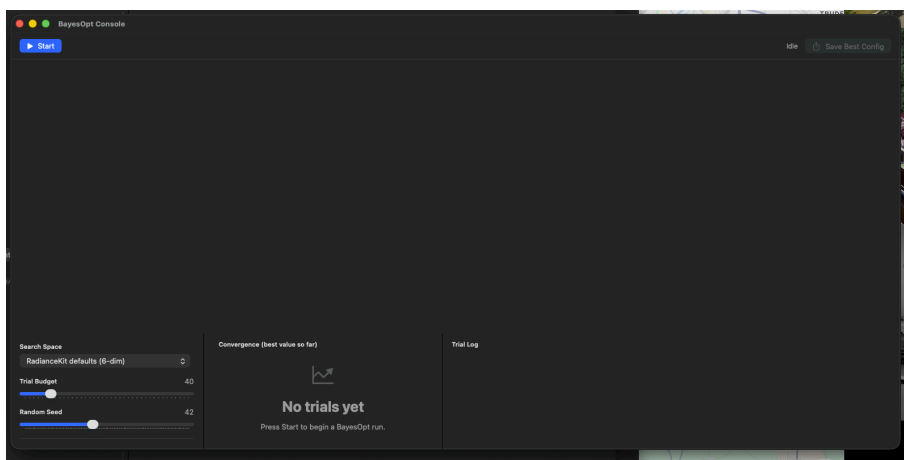


図 24: BayesOpt コンソール — 試行開始前の空状態

Search-Space ピッカー (RadianceKit defaults (6-dim))、Trial-Budget スライダー (デフォルト 40)、Random Seed (42)、Convergence チャート、Trial Log、Search-Space パラメータリスト用の 3 つの空パネルを持つ空状態。

空状態 (最初に開いた後) — Convergence チャートと Trial テーブルは、実行が開始されるとすぐに埋まります。次のショット参照。



図 25: 40 試行後の BayesOpt コンソール — Convergence チャートは試行 15 まで急上昇、Best Value 0.9943、init/bo/restart タグ付き Trial Log

画像で見えるもの 右上のステータス「Finished — best 0.9943 after 40 trials」。左サイドバー: Search-Space ピッカーは RadianceKit defaults (6-dim)、Trial Budget 40、Random Seed 42。Parameter リストは、値範囲を持つチューニングする 6 つのハイパーパラメータを表示します: 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]。中央: Convergence チャート (X = Trial インデックス 1~40、Y = Objective Value 0~1) — グレーポイント = 初期サンプル (LHS)、青ポイント = BayesOpt 取得、オレンジポイント = Restart 試行 (#22 と #31)。Best Value 線は試行 ~7 まで急上昇し、その後試行 15 までわずかな改善のみ、それからフラットなプラトーが 0.99+。右サイドバー: スコア + タグ (init/bo/restart) 付き Trial Log #1~#34。右上の Save Best Config ボタンは bayesopt-best.json を書き込みます。

何か: ハイパーパラメータ検索用のベイズ最適化コンソール。Bayes-Opt は、できるだけ少ない実験で未知の関数の最適点を見つけようとする自動プロセスです — 通常: 「mcmcMaxGaussians、capMultiplier、ssimWeight、gradThreshold のどの組み合わせが私のシーンクラスに最良の PSNR を提供するか?」 $6^4 = 1296$ 試行のグリッドの代わりに、Bayes-Opt は約 40~100 の情報を持つ試行を試し、それで最適に近づきます。

重要: アプリで提供される現在のバージョンは、実際のトレーニング実行に対して最適化を実行しません (数日かかる)。代わりに合成デモオブジェクトに対して実行します — Hill-Climbing 特性と軽いノイズを持つマルチモーダルランドスケープ。これは意図的: ウィンドウは Optimizer の動作 (収束進行、サンプルポイント、Best So Far) を示し、Search Space 定義を理解させる必要があります。実際のトレーニング駆動の BayesOpt 実行 (Phase Q7 で Scene-Class プリセット用に実行されたように) には、別のオフライン CLI ワークフローが使用されます。ウィンドウはライブ UI バリエーションです。

いつ開くか 3 つのユースケース: 1. BayesOpt がどのように動作するかを理解したい — そうしたらデモ実行を開始し、Convergence チャートを観察します。2. 組み込みの 10 プリセットが完全に適合しない新しいシーンクラス (「アクアリウム」や「アンティーク家具」など) を計画しています。検索空間を心理的に定義し、ここで「Bowl demo」や「Densify」プリセットでテストし、Best Config を JSON としてエクスポートし、実際のトレーニング実行の出発点として使用します。3. RKBayesOpt パッケージで定義

されたデフォルト Search Space (Mip サブセット、RadianceKit デフォルト) を検査したい – これらは左サイドバーの Parameter パネルにリストされます。

どのように解釈 - **Convergence チャート** (中央列): Y = これまで 最高達成された Objective Function 値。X = Trial インデックス。最初は急上昇 (BayesOpt は初期サンプルをランダムに試し、一部は幸運)、次第に平坦になります。最適に近い領域が使い果たされたためです。線が 20+ 試行にわたって平坦なままの場合、実行を停止できます – さらなる試行は何ももたらしません。チャート内の個々の点は 個々の Trial 値 (つまり、「これまで最良」ではない) で、フェーズによって色付け: グレー = 初期サンプル、青 = bayesopt 取得、オレンジ = restart。- **Trial テーブル** (右列): #1、#2、#3、... それぞれ値と Phase タグ。これまで最良の Trial は黄色の星でマークされます。テーブルから Best Trial を識別し、そのパラメータ 値を後でエクスポート時に見られます。- **Search Space インспекター** (左サイドバー): 選択されたプリセットのすべてのパラメータ名とその検索範囲 [lo, hi] を表示します。「RadianceKit defaults (6-dim)」プリセットの場合、例えば「densifyGradThreshold [5e-7、5e-6]」 – つまりこれら 2 つの値の間の log-uniform が表示されます。

ワークフロー例 プリセット「RadianceKit defaults (6-dim)」、Trial Budget 40、Seed 42 を選択。「Start」をクリック。観察: 最初の 8 試行はグレー (初期サンプル、LHS Latin-Hypercube)、続く青 (BayesOpt 取得)。Convergence チャートは試行 ~15 まで急で、その後平坦になります。試行 ~30~40 で Best Value が安定します。「Save Best Config」をクリック – プリセット名、Trial インデックス、値、デコードされたパラメータ 値を含む bayesopt-best.json が保存されます。この JSON を、手動でプリセット定義に引き継ぎます。

W30 ボタン「Start」



場所

Idle/Finished 状態の左ツールバー。



技術詳細

Trial リストをリセットし、Running 状態に切り替え、新しい Run ID を生成 (複数 Start クリックでの Stale 検出用) し、新しい Pause ゲートを生成します。次に、Optimizer を非同期ストリームとして実行するバックグラウンドタスクが開始されます。初期サンプルサイズは $\min(8, \text{budget} / 4 + 1)$ から導出されます – つまり、Budget ≥ 28 で通常 8 Latin-Hypercube サンプル、小さい Budget でより少ない。Trial 更新は 増分的に受信され、リストに追加されます。Stale Run 保護: 2 番目の Start クリックがその間に Run ID を新しく設定する場合、古い実行からの更新は破棄されます。

目立つボタンの外観用の Primary Action スタイル。

簡単に言うと

現在の検索空間、Budget,Seed で新しい最適化実行を開始します。

W31 ボタン「Pause」

場所

Running 状態の左ツールバー。



技術詳細

Pause ゲートをアクティブに設定し、Paused 状態に切り替えます。実際の効果: ランナーは次の Objective 関数を評価する前に 50 ms ポーリンググループで待機します。これは、現在実行中の Trial が完了することを意味します (合成でマイクロ秒のみ続く) が、それ以上の Trial は開始されません。Resume が実行されるとすぐに、停止した場所から 続行されます。

簡単に言うと

実行を停止します。現在の計算はまだ完了し、その後一時停止します。

W32 ボタン「Stop」

場所

Running と Paused 状態の左ツールバー。



技術詳細

ランナータスクを中止し、参照をクリアし、Pause ゲートを解除 (まだ paused だった場合) し、Finished 状態 (Trial が存在する場合) または Idle 状態 (存在しない場合) に切り替えます。すでに計算された Trial はリストで表示されたままです — Stop は削除しません。破壊的なボタンの役割は、ボタンを赤で示します。実行を中止するためです。

簡単に言うと

実行を最終的に中止します。Trial は表示されたままで、Best Config をそれでもエクスポートできます。

W33 ボタン「Resume」

場所

Paused 状態の左ツールバー。



技術詳細

Pause ゲートを解除し、Running 状態に戻します。ランナータスクはすでに 実行中 (ポーリンググループで待機中)。ループが Pause が解除されたことに気付くとすぐに、 継続して次の Trial を開始します。

簡単に言うと

一時停止した実行を続行します。

W34 ボタン「Save Best Config」



場所

右ツールバー、常に表示 (ただし `bestTrial` が存在しない場合は無効)。



技術詳細

JSON に制限された、デフォルトファイル名 `bayesopt-best.json` を持つ 保存ダイアログを開きます。確認後、Payload Dictionary が構築されます: プリセット名、Trial インデックス、値 (Objective スコア)、Parameter (デコードされたパラメータ名 → 値の Dictionary)。デコーディングは、正規化された検索空間座標を $[0,1]^d$ で元の 値範囲に戻し投影します (それに応じて `log-uniform/linear/integer` スケール付き)。JSON 出力は `pretty-print` されており、キーがソートされています。書き込みエラー時 (現在のデモバージョン) は静かに無視されます – これがデモパスであるため、エラー UI はありません。

Trial が実行されていない限り、ボタンはグレーのままです。

簡単に言うと

これまで最良の Trial のパラメータ値を JSON として保存します。次にこれらの値をプリセット構成に手動で引き継ぎます。

W35 ピッカー「Search Space」プリセット



場所

左サイドバー、上部。



技術詳細

4 つのプリセットオプションを持つメニューピッカー:

- 「RadianceKit defaults (6-dim)」 – すべての Q7 ハイパーパラメータを持つ完全な標準検索空間。
- 「Mip subset (2-dim)」 – `mipSmoothing3DScale` [0.05、0.5] `log-uniform` と `mipFilter2DVariance` [0.1、0.6] `linear` のみ。シーンクラス用に Mip-Splatting をチューニングしたい場合に便利。
- 「densify-until + ssim-weight + grad-thresh」 – 3 つの Densify 関連パラメータ (`densifyGradThreshold` `log-uniform`、`ssimWeight` `linear`、`densifyUntilIter` `integer`)。
- 「Bowl demo (1-dim)」 – 「BayesOpt はこう動作する」 デモ用の教育的シングルパラメータ検索空間。

実行がアクティブな間、検索空間は切り替えられません (Optimizer を混乱させます)。

簡単に言うと

BayesOpt がどのハイパーパラメータ検索空間を検索するか。標準は「RadianceKit defaults」。Mip 特定のチューニング試行には「Mip subset」。BayesOpt の動作を理解するには「Bowl demo」。

W36 スライダー「Trial Budget」

場所

左サイドバー、Search Space ピッカーの下。



技術詳細

10 から 200 までのスライダー、ステップ幅 5。デフォルト 40。それが意味するもの: BayesOpt は最大 N Trial を行うことができます。そのうちの最初のいくつかは 初期サンプル (Latin-Hypercube) で、残りは実際の BayesOpt Trial です。実践用の 経験則: d 次元を持つ検索空間は、良い最適化のために約 10d から 20d Trial を 必要とします。6-dim Defaults では 60~120、2-dim Mip サブセットでは 20~40、1-dim Bowl デモでは 10~20。

実行中、スライダーは無効化されています。

簡単に言うと

最大いくつの最適化試行。試行が多いほど = より良いソリューションだが時間がかかる。40 はデモオブジェクト用の良いデフォルト。

W37 スライダー「Random Seed」

場所

左サイドバー、Budget スライダーの下。



技術詳細

1 から 100 までのスライダー、ステップ幅 1。デフォルト 42。Seed は、初期 Latin-Hypercube サンプルとデモオブジェクトの Noise コンポーネントの両方に 渡されます。再現性: 同じ Seed + 同じ検索空間 + 同じ Budget は、まったく同じ Trial シーケンスを提供します。「同僚全員がデモを再構築すると同じ実行を取得する?」に 役立ちます。実行中は無効化。

簡単に言うと

乱数生成器を制御します。同じ Seed = 同じ実行 — 再現用。

W38 チャート (Convergence)

場所

ウィンドウの中央列。



技術詳細

2つのレイヤーを持つ Swift Charts ダイアグラム: 1. Trial ごとの「best-value-so-far」用の線 — アクセント色の単調増加または一定の曲線。2. Trial ごとの個別の Objective 値を持つポイント、フェーズによって色付け。Symbol サイズ 40。3つのフェーズラベル: 「init」(グレー)、「bo」(青)、「restart」(オレンジ)。

小さな凡例が左上に Phase の色を表示します。Trial リストが空の場合 (最初の Start 前)、チャートアイコンとヒント「Press Start to begin a BayesOpt run.」を持つ空状態表示が代わりに表示されます。

簡単に言うと

進捗チャート。実線は「これまで見つかった最良のソリューション」、ポイントは個々の試行。線が長い間平坦なままの場合、BayesOpt は最適を見つけました。

W39 テーブル (Trial Log)

場所

ウィンドウの右列。



技術詳細

Lazy にスタックされた Trial 行を持つスクロール領域。行ごとに水平スタック: Trial 番号 (3 桁 monospace、左)、値 (monospace、右揃え、70 pt 幅)、Phase タグ (Capsule、Phase 色で 25% 透明度で塗りつぶし)、この Trial が現在最良の場合は オプションで黄色の星。Auto-Scroll メカニズムは、新しい Trial が追加されるとすぐに、自動的に最後にジャンプします — 自分でスクロールすることなく、ライブ進行を画面下部で読み取れます。

簡単に言うと

すべての試行のテーブル。値、フェーズ、最良の星。新しい Trial は下に表示され、自動的にスクロールします。

メインウィンドウ: Loss 進行と Gaussian Count (I39~I41、クロスリファレンス)

メインウィンドウの 3つの Inspector 表示には独自の説明が必要です。実行中のトレーニング中に常に表示され、進行が健全に見えるかの重要な経験則があるためです。表示は Inspector の「Loss Chart」セクション (第 2 章 — Inspector 参照) にあり、上の補助ウィンドウからの Holdout 分析を補完します。

Loss 曲線はいつ健全か? 健全な Loss 曲線は 3 つのフェーズを示します: (1) **Warmup** — 最初の 200~500 反復で、Loss は高 (シーンに応じて L1+SSIM 結合で通常 0.15~0.25) から約半分に急落します。Loss がこのフェーズで落ちない場合、通常入力の間違っています (画像が破損、SfM ポーズが悪い、初期 Gaussian の数が少なすぎる)。 (2) **Densification** — ~500 から densifyUntilIteration (古典的に 15K、MCMC で 20K または 25K まで) の間で Loss はさらに落ちます。Densify 操作が新しい Gaussian を挿入し、Optimizer がそれを利用すると、しばしば小さな下方ジャンプが発生します。Gaussian カウントはこのフェーズで上昇します。 (3) **Refinement** — その後、Loss はより平坦な Tail に入ります。典型的な終値: P4 Quality の Tanks-&-Temples Truck は $L1 \approx 0.023$ 、Full Classic V546 の Horse は $L1 \approx 0.0230$ 、Outdoor Mip-NeRF360 シーンはしばしばより悪い (0.04~0.07)。

プラトーは何を意味するか? プラトー (Loss 曲線が数千の反復にわたって水平に走る) は 2 つの解釈を持ちます: (a) モデルが収束しました、それ以上のトレーニングは 何ももたらしません — それが良いケースです。 (b) モデルがスタックしています (ローカル最小値、悪い勾配情報、バッファ制限のキャップ) — 悪いケース。両方ともチャートで同じに見えます。区別: Gaussian カウントを見ます。それも平坦で MCMC キャップに近い場合 (例えば `.fullMCMC` で 150K のうち 150K)、限界にいます — キャップを上げるかプラトーを受け入れます。Gaussian カウントがまだ成長しているが Loss が落ちない場合、スタックしています。

中止 vs 続行のタイミングは? 経験則: Min-Loss の改善なしで 10K 反復 → 中止、さらなる反復は無駄です。その前: 境界改善が見える場合、`Cmd+T` (Training メニュー → Continue Training → +5K 反復) 経由で延長を追加できます。注意: MCMC ではプラトーはしばしば本物です — キャップは自然な制限です。

Gaussian カウントプラトーは「完了」信号ではありません。 MCMC がキャップに達したか、Classic Densification が使い果たされたことを意味するだけです。実際の「完了」質問は Holdout 分析でのみ尋ねられます — Holdout ウィンドウ (W23~W29) や `--benchmark` フラグで評価された独立したテストセットでの PSNR/SSIM/LPIPS。

PSNR/Holdout は真実、Loss はプロキシのみ。 Loss は相対メトリックです: モデルがトレーニングビューに適応する間に落ちます。しかし、低い Loss は自動的に良いモデルを意味しません — モデルがトレーニング画像を暗記した (オーバーフィッティング) 場合、Loss は小さいですが、未見ビュー (Holdout) での PSNR は悪いでしょう。したがって: 最終品質評価では常に Holdout メトリックを見て、End-Loss だけを見ません。

経験則ボックス

- User Guide と Keyboard Shortcuts は静的ヘルプ — キーワード質問には速い、深さにはここにあるマニュアルを使用してください。
- ディスクが 10% 未満の空き容量に落ちるとすぐに Manage Storage を開きます。ログとインポートステージングが通常の罪人です。
- Pareto Dashboard は、少なくとも 3~4 のトレーニングレポートを取得した後にのみ意味があります。X 軸 = コスト (Time / Gs)、Y 軸 = 品質 (PSNR / SSIM)。Pareto Front は効率的な組み合わせを示します。

- 他者と PSNR ベンチマークを公開する前に Holdout Analysis を使用してください – テストセットが本当に代表的であることを保証します。
- BayesOpt Console は主に Search Space 定義の学習および検査ツールです。実際のトレーニング駆動のハイパーパラメータチューニングには、オフライン CLI ワークフローを使用してください。
- Loss プラトーと Gaussian カウントプラトーは別々に解釈する必要があります。Cap 制限は「完了」信号ではありません。実際の品質は Holdout PSNR のみが測定します。
- Min-Loss の改善なしで 10K 反復 → トレーニングを停止します。

章

第 6 章 — トレーニング構成

```
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,
    "postCompaction": true,
    "perceptualLoss": 0.0,
    "metalFXUpscaling": false,
    "mpsLanczosScaling": false,
    "skyMasking": false,
    "midTrainingFloaterCleanup": true,
    "scaleRegularization": false
  }
}
```

図 26: JSON としてエクスポートされ `TextEdit` で表示された `Preview` プリセット — フィールド `id/name/category/version/createdAt/description`、すべての関連パラメータを持つ `trainingConfig` (`maxIterations` 5000、`densifyUntilIteration` 3500、`ssimWeight` 0.20、`renderScale` 0.50、`strategy` `classic`、`cameraAlignment` `applePhotogrammetry`、`densifyGradThreshold` `2.0e-06`、`opacityResetInterval` 3000、`minOpacity` 0.005、6 つの `Boolean` トグル)

画像で見えるもの 典型的なプリセット JSON エクスポート。トップレベルフィールド: `id` (UUID)、`name`、(`classic` | `mcmc` | `sceneClass` | `custom`)、(スキーマバージョン)、(タイムスタンプ)、(フリーテキスト)。ネストされたオブジェクトには、再現性に重要なパラメータが含まれます — インポート時、ブロック全体が `TrainingConfig` 構造体にデシリアライズされ、アプリバージョンからのデフォルトが JSON に欠けているフィールド (例えばアプリ更新後) を埋めます。プリセットを別の Mac に渡す場合、単にこの JSON ファイルを送ります。

`TrainingConfig` 構造体は `RadianceKit` の各トレーニング実行の核です。トレーニングに影響するすべてのパラメータを集めます — 最大反復数から 8 つの学習率、MCMC、Mip-Splatting、Curriculum、scene-aware Cap ロジックの専用フィールドまで。サイドバー

のトレーニング構成セクション (エキスパートビュー) で編集し、プリセットとして保存するか、JSON エクスポートとして別の Mac に渡します。トレーニング時、このオブジェクトが正確に固定され、GPU バックエンドに渡されます。

この章は、パワーユーザーとスクリプト作成者向けのリファレンス資料です。すべての 81 のパブリックフィールド、9 つの静的プリセット、1 つのパブリックメソッドをリストします。ソースファイルは TrainingConfig.swift — 疑問があれば、そこに保存されている Doc コメントと Initializer デフォルトが Source-of-Truth として有効です。

注意 · UI VS. PRESET/CLI

81 フィールドのうち、Inspector (Sandbox App Store ビルド) に直接スライダー、トグル、ピッカーを持つのは 12 個だけです: **T1**、**T2**、**T17**、**T20**、**T22**、**T38**、**T56**～**T58**、**T60**、**T61**、**T73**。残りの 69 フィールドは選択された **プリセット** (第 7 章) で設定され、**CLI フラグ** (第 5 章参照) 経由でのみ直接上書き できます。この分離は意図的 です: デフォルトは安定し、本番テスト済みで、パワーユーザーには依然として脱出口があります。フィールドが特に興味深い場合は: まず第 2 章 (Inspector) と第 5 章 (CLI) で、JSON 操作なしで到達できるかを確認してください。

目次:

1. 反復 (T1～T2)
2. Learning Rate (T3～T10)
3. Densification — Classic (T11～T16)
4. Loss (T17～T20)
5. SH 次数進行 (T21)
6. パフォーマンス (T22～T25)
7. 診断と点群準備 (T26～T30)
8. 正則化 (T31～T37)
9. Refinement (T38～T44)
10. Sky-Dome (T45～T48)
11. Adam + LR スケジュール (T49～T55)
12. ポスト処理 + Apple AI (T56～T60)
13. MCMC Densification (T61～T73)
14. Mip-Splatting (Q1.5) (T74～T76)
15. 適応 Densification (Q5) (T77～T79)
16. Curriculum (Q6) (T80～T81)
17. 静的プリセット (TP1～TP9)
18. メソッド:
19. どのフィールドが何のためか? (チートシート)
20. 危険なフィールド

反復 (T1~T2)

T1 maxIterations

詳細

デフォルト: 30 000 (Initializer)、35 000 (`.full`)、200 000 (`.fullMCMC`) **範囲:** 1 000 – 500 000 (UI スライダー)、ロジック内に ハード上限なし **定義場所:**

技術詳細

バックエンドが実行するトレーニング反復の総数。1 反復は、単一のトレーニングカメラの順方向レンダリング、すべての Loss コンポーネント (L1 + SSIM + オプションの正則化 + Sky マスク) にわたる逆方向パス、Adam Optimizer ステップを意味します。この数値は他のスケジュールに直接影響します: Position Learning Rate は Cosine Annealing 曲線に従い 0 から T1 自体、または T49 positionLRScheduleEndIteration まで。Densification は T2 densifyUntilIteration で停止。MCMC Noise Decay は T69 mcmcNoiseDecayEnd で終了。SH Degree アップグレードは T21 で定義された 3 つのマークで発生します。クラシック Densification では、経験的に決定されたスイートスポットは 20 000~35 000 反復 (Sessions 1~32、V546 テスト)、MCMC では 60 000~200 000 (V534)。プリセットに保存された値を大幅に超える増加は、めったに追加の品質をもたらしません — Adam Momentum が飽和し、LR Decay の終了がないと Loss が停滞します。逆に、~5 000 を下回ると、不完全に収束したジオメトリになります (Density Control が クローン/分割の時間が足りない)。

簡単に言うと

アプリの計算時間。反復が多いほど = より良い結果ですが、ある時点からそれ以上の改善は感じられず、はるかに長くなります。プリセットは考えずに良い値を持つように選択されています: Quick 1 000, Preview 5 000, Balanced 20 000, Quality 35 000, MCMC Quality 200 000。自分で調整する場合: MCMC では高くしても構いません (100 000~200 000), Classic では 40 000 を超えないでください — それ以上は何ももたらしません。

T2 densifyUntilIteration 詳細

デフォルト: 15 000 (Initializer)、5 000 (`.full`)、160 000 (`.fullMCMC`) **範囲:** 0 – **定義場所:**

 技術詳細

Densification が終了する反復。ここまで Gaussian は T11~T16 (Classic) または T67~T70 (MCMC) でパラメータ化されたルールでクローン、分割、Prune され、その後 Gaussian 数は一定のまま、位置、回転、スケール、Opacity、SH 係数のみが最適化されます (Refinement フェーズ)。3DGS オリジナル論文では値は T1 の 50% ですが、RadianceKit `.full` プリセットでは ~14% (35 000 のうち 5 000) — V310/V338 実験の結果、5 000 反復後のさらなる Densification は結果を悪化させる (より多くの Floater、より多くのメモリ要件、品質向上なし) と示されました。MCMC は対照的に Relocation を T1 の 80% (V504b) まで実行します。MCMC が有害な Floater を生成しないためです。T2 を小さすぎる (< 1 000) と Gaussian が少なすぎ、Classic で大きすぎる (> T1 の 50%) と Overgrowth と RGB 飽和外れ値になります (Outdoor 過剰トレーニング所見参照)。

 簡単に言うと

アプリがいつまで新しい Gaussian を作成できるか。その後は、すでにあるものを洗練するだけです。35 000 反復のクラシックトレーニングでは、ここで 5 000 が正しい値です — それ以上はシーンをよりぼかします。MCMC では総反復の 80% (つまり 200 000 実行で 160 000)。Quality プリセットを変更する場合、このフィールドはそのままにしておく方が良いです。

Learning Rate (T3~T10)

T3 positionLearningRate

📖 詳細

デフォルト: 0.00016 **範囲:** 1e-7 – 1e-3 (推奨) **定義場所:**

🔧 技術詳細

トレーニング開始時 (反復 0) の各 Gaussian の XYZ 位置用の Adam 学習率。Cosine Annealing 曲線に従い、トレーニング中に T4 positionLearningRateFinal に低下します。デフォルト 0.00016 は 3DGS オリジナル論文 (Kerbl et al. 2023) からのものです。RadianceKit では、画像解像度を増加させても、ピクセル空間ではなく世界座標系で Gaussian が移動するため、スケールアップしません。著しい増加 (> 0.0005) は Gaussian が長距離をジャンプし、Loss が不安定になります。それを大幅に下回る値 (< 0.00005) は、誤って初期化された点群が決して位置を見つけないことを意味します。V414 は初期値の倍化をテスト → 16.8% 悪い L1 Loss。V544a チューニングは Paper デフォルトを最適として確認しました。注意: `.fullMCMC` では、この値を意図的にデフォルトのままにします — MCMC は Relocation ロジックのために一定の学習率を必要とするため、ここでのチューニングは何ももたらしません。

💬 簡単に言うと

Splat 点が空間でどれだけ速く移動できるか。標準値は非常によく調整されており、基本的に変更は不要です。画像内で Splat が「揺れ動く」、または何も動かないため全体の角が欠けている場合のみ、学習率が調整すべき点になります — しかし、その時には通常、他の何か (カメラポーズ、初期点群) がすでに間違っています。

T4 positionLearningRateFinal 詳細

デフォルト: 0.0000016 (Initializer + Paper)、
0.000016 (`.full`、`.fullMCMC` — 10× 高い) **範囲:**
0 – 定義場所:

 技術詳細

Position LR Cosine Annealing 曲線の終値。

`T1 maxIterations` で、または 設定されている場合は `T49 positionLRScheduleEndIteration` で達成されます。RadianceKit `.full` プリセットは 0.000016 — Paper デフォルト 0.0000016 の 10× を使用します。V420 実験は、Final 値の 0.5× (0.000008) が Loss を 6.4% 悪化させると示しました。V414 は、初期値の 2× が 16.8% 悪化させると示しました。高い Final 値はトレードオフではなく意図的な選択: 強すぎる Decay では、Refinement フェーズ中に新しく追加された Densification 候補に Gaussian が適応する能力を失います。V431/V433 拡張により、スケジュールフェーズを短縮 ($T49 < T1$) し、`T4` がトレーニング終了前に達成され、トレーニングの残りが一定の Mini-LR で実行されます — 典型的な構成: `T49 = 20 000`、`T1 = 35 000`、つまり Refinement は 15 000 反復にわたって 0.000016 で実行。

 簡単に言うと

トレーニング終了時に Position 学習率がどれだけ遅くなるか。意図的にオリジナル論文より積極的でない設定にしました — Splat は最後まで少し揺れることができ、それがより鮮明にします。これを調整する場合: 高い = 終了時の Splat がより不安定、低い = 新しい Splat が現れたときに Splat が適応できない。

T5 shDCLearningRate 詳細

デフォルト: 0.0025 (Initializer + Paper)、0.005
(`.full` とすべての MCMC プリセット - 2x) **範囲:**
0.0001 - 0.05 **定義場所:**

 技術詳細

球面調和色の DC 部分 (degree 0、つまり一定アルベド) 用の Adam 学習率。SH-DC は Gaussian の方向に依存しない基本トーンに対応します。いわば「基本色」です。V176 と V188 実験は Paper デフォルトの 2x が最適であると見つけました - より速い色収束、特に短いトレーニング (、5 000 反復) では SH-DC が形にならないからです。幾何学的 LR とは異なり、SH-DC には Decay がありません。学習率はすべての反復で一定のまま (またはオプションの `T51` 拡張フェーズ Decay のみに従う)。V416 は 0.01 への 4 倍をテスト → `beta2=0.99`-Adam で 6.4% 悪い Loss。

 簡単に言うと

各 Splat の基本色がどれだけ速く適応するか。値はほとんど自分で変更しません - プリセットは正しい値を持ちます。高くすればより速いですが、不安定な色になる可能性があります。

T6 shRestLearningRate 詳細

デフォルト: 0.000125 (Initializer + Paper)、0.00025
(`.full` と MCMC - 2x) **範囲:** 0.000001 - 0.005 **定義場所:**

 技術詳細

高次 SH 係数 (Degree 1、2、3 - つまり、ハイライト、反射、ソフトシェーディングを引き起こす view 方向依存色部分) 用の Adam 学習率。Paper 規約に従い `T5` より 20x 小さい。これらの係数は数で 2 次的に成長し (Degree 1 で 3、Degree 2 で 5、Degree 3 で 7 → 合計 Gaussian あたり 15 Float)、小さい学習率なしでは画像を過飽和させます。`T21 shDegreeUpgradeIterations` の最初のマークまで Degree 0 のみがアクティブ (つまり `T5` のみ)、その後 1、後で 2、最後に 3。ここでの低い値は、拡散照明の多いシーンで特に重要です。非常に光沢のある表面 (車のペイント、水) では、調整は価値がありません - SH 表現自体が制限されています。

 簡単に言うと

方向依存の色効果 (反射、光沢) がどれだけ速く学習するか。標準では非常に小さい。そうでなければすべてが光ります。値はそのままにしておいた方が良いでしょう - ハイライトをより良く取得したい人は、この LR よりも MCMC と長いトレーニング時間によってより良くサービスされます。

T7 opacityLearningRate

📖 詳細

デフォルト: 0.05 (Initializer + Paper)、0.1 (.full、MCMC — 2x) **範囲:** 0.001 – 1.0 **定義場所:**

🔧 技術詳細

各 Gaussian の logit Opacity 用の Adam 学習率。アプリは Opacity を 制約のない Float 値として保存し、Sigmoid で $[0, 1]$ に変換します。LR は logit 空間で動作します。Paper デフォルト 0.05 は V50 テスト (Best Single-Run L1 0.1664) 後に 復元されました。V71 は V67 の 0.025 を元に戻しました。V188 の 0.1 への倍化は Pruning をより効率的にします — 死んだ Gaussian は `T14 pruneOpacityThreshold` の下により速く落ちます。V418 は: 0.05 with $\beta_2=0.99$ -Adam は 0.1 より 7.1% 悪い — Adam 構成との相互作用は自明ではありません。低い値 (< 0.01) は「死んだ」Gaussian が 永遠に存在しメモリを消費することを意味します。高すぎる値 (> 0.5) は Opacity 爆発を引き起こす可能性があるため、logit 値が Optimizer で $[-15, 3]$ にクランプされます (CLAUDE.md の「Opacity Explosion Prevention」注参照)。

💬 簡単に言うと

Splat がどれだけ速く透明または不透明になるか。クリーンアップに重要 — 何も寄与しない Splat はベールが形成されないようにすぐに消える必要があります。標準値は適切で、プロのみが変更します。

T8 opacityLearningRateFinal

📖 詳細

デフォルト: 0.0 (= 「Decay なし」) **範囲:** 0 または 0.001 – **定義場所:**

🔧 技術詳細

Opacity LR 用のオプションの Cosine Decay 終値 (V427)。0.0 の場合、Decay は 無効化され、Opacity LR は全トレーニングにわたって T7 で一定のままです。V427 は Decay 0.1 → 0.01 をテスト — 結果 11.5% 悪い Loss。元に戻され、デフォルトは「オフ」。フィールドの仮説: Refinement フェーズで、一定の Opacity LR は振動を引き起こす可能性があり、すでに正しい透明度に達した Splat がランダムな勾配変動で再び移動される。経験的に確認されません — Logit クランプロジックがすでにそれをキャプチャします。フィールドは将来の実験のために使用可能なままで、非常に長い MCMC 実行 ($> 500K$ 反復) もこれから利益を得られる可能性があります。

💬 簡単に言うと

Opacity 学習率を終了に向けて小さくするかどうか。標準: いいえ。試しましたが悪かったので、無効のままにします。0 のままにしてください。

T9 scaleLearningRate 詳細

デフォルト: 0.005 (Initializer + Paper)、0.01
(.full、MCMC — 2x) **範囲:** 0.0001 – 0.1 **定義場所:**

 技術詳細

対数空間 (RadianceKit は $\log(\text{scale})$ を保存し、スケールを正に保ちます) で各 Gaussian の 3 つのスケールコンポーネント用の Adam 学習率。Paper デフォルト 0.005、RadianceKit では 0.01 に倍化されており、最適化された学習率構成でのより良いスケール収束のため。V423 実験: $\beta_2=0.99$ -Adam で 0.005 \rightarrow 18.7% 悪い Loss と目に見えて少なすぎる Gaussian (スケール更新が遅すぎたため Density Control がクローンできなかった)。スケールは各 Gaussian の範囲を制御します — 学習が速すぎると「ニードル」 Gaussian (非常に長い細い Splat、T34 scaleRatioPruneThreshold 参照) になり、学習が遅すぎると Splat がコンパクトすぎるままで Density Control が頻りに分割しなければなりません。

 簡単に言うと

Splat の形がどれだけ速く適応するか。標準は良好。高くすると、「ニードル」 Splat — 画像を浮遊させる極端に長い細いしずく — が得られる可能性があります。

T10 rotationLearningRate 詳細

デフォルト: 0.001 (Initializer + Paper)、0.002
(.full、MCMC — 2x) **範囲:** 0.0001 – 0.05 **定義場所:**

 技術詳細

各 Gaussian の 4 つのクォータニオンコンポーネント用の Adam 学習率。クォータニオンは各 Optimizer ステップで Adam 更新後に再正規化されます (L_2 ノルム = 1) — そうでなければ共分散マトリクスが退化します。RadianceKit は Quality プリセットで Paper デフォルトを倍化します。回転はスケール/位置と比較してより小さな絶対勾配 マグニチュードを持ち (単位球面で各ステップは短いま)、2x なしでは回転は 35 000 反復ウィンドウで著しく収束不足になります。V188 文書化。NeRF-Blender シーン (Lego、Chair) では、回転は特に影響します — オブジェクトのエッジは 5 000 ~ 10 000 反復後に初めて正しく整列します。

 簡単に言うと

Splat がどれだけ速く回転を学習するか — つまり、オブジェクトの表面で正しい方向に来るか。標準は適切です。言い換えると、Splat が表面に密着するのではなく歪んだブロックのように見える場合、トレーニング時間が短すぎるのであって、この学習率が低すぎるわけではありません。

Densification — Classic (T11~T16)

T11 densifyGradThreshold

📖 詳細

デフォルト: 0.000002 (Initializer、0.5× 解像度用にキャリブレート)、0.0000011 (.full、1.0× 用にキャリブレート)、0.000004 (.quickTest、0.25× 用にキャリブレート)、2e-7 (.fullClassicPaper)
範囲: 1e-8 – 1e-3 (解像度依存) **定義場所:**

🔧 技術詳細

画面空間投影された勾配 `dMean2D` の L2 ノルムの閾値で、これを超えると Gaussian がクローンまたは分割の対象としてマークされます。絶対値はトレーニング解像度に直接依存します — `dMean2D` は約 1/解像度² のようにスケールします (より多くのピクセル = より小さいピクセルごとの勾配)。したがって、各 T22 `trainingRenderScale` レベルにはキャリブレートされた閾値が必要です: 0.25x → 4e-6、0.5x → 2e-6、1.0x → 5e-8 ... 1.1e-6 (.full)。Paper デフォルト 0.0002 は NDC 正規化されており、RadianceKit の世界空間パイプラインで直接比較可能ではありません。V440 で追加された T52 `adaptiveDensifyThreshold` フラグでランタイムで現在の勾配分布の p98 から値を計算できます — しかし V440 はそれを実際のシーンでテストして 63 K Gaussian を生成しました (壊滅的な Pruning 損失)。フラグはオフのままです。Q5 (T77~T79) は rolling median 経由の代替適応ロジックを提供します。このフィールドは **無害ではありません** — 半分にすると 2~4x の Gaussian (メモリ圧、OOM リスク)、2 倍にするとシーンが過小 Densify される可能性があります。

💬 簡単に言うと

Splat が表示不足で複製されるべきかを決定する際のアプリの感度。低い値 = より敏感 = より多くの Splat。高い = より少ない Splat。これは最も危険な値の 1 つです: 低すぎると Mac が数百万の Splat でメモリがいっぱいになり、おそらくクラッシュします。このフィールドはそのままにしておくか、10% のステップでのみ変更してください。

T12 densifyFromIteration

詳細

デフォルト: 500 **範囲:** 100 – 5 000 **定義場所:**

技術詳細

Densification がアクティブになる最初の反復。それ以前は、新しい Gaussian が作成されることなく、初期 SfM 点群での「裸の」学習のみが行われます。デフォルト 500 は 3DGS 論文からのもので、初期化に安定する時間を与えます — 反復 0 から Densify されると、誤って配置された SfM 点が、正しい位置を見つける前に何度もクローンされます。V349 は 1000 をテスト → わずかに悪い Loss。デフォルトが最適です。

簡単に言うと

アプリが Splat のクローンを初めて開始するとき。それ以前は、既存の点のみを学習します。500 が標準値です — アプリが繰り返す前に最初に方向付けるのに十分な時間を与えます。

T13 densifyInterval

詳細

デフォルト: 100 (Initializer、MCMC)、200 (.full)**範囲:** 50 – 1 000 **定義場所:**

技術詳細

2 つの Densification ステップ間にある反復の数。Paper デフォルト 100 — 100 反復ごとに densify 候補のリストが評価され、クローン/分割され、同時に prune 候補のリスト ($\text{sigmoid}(\text{opacity}) < \text{T14 pruneOpacityThreshold}$) が削除されます。V112 テストは 200 を .full に最適と発見 — 再編成パスが少ないため GPU が軽減され、クローンアクション後に各 Gaussian に安定する時間がより長く与えられます。V417 は $\text{beta2}=0.99$ で 100 をテスト → 5.8% 悪い (957 K Gaussian、過剰 Densification)。MCMC では同じフィールドが Relocation 間隔として解釈されます。MCMC 固有のロジックは T67 `mcmcRelocationInterval` を参照してください。

簡単に言うと

アプリがどれくらいの頻度で新しい Splat を探すか。100 = しばしば、200 = 中程度。高いほど: 各 Splat が再度複製される前に、配置する時間が長くなります。それは良いことです。50 に下げると、GPU が顕著に良くなることなく永続的に占有される可能性があります。

T14 pruneOpacityThreshold 詳細

デフォルト: 0.005 (Initializer、Paper、MCMC)、0.001 (`.full`) **範囲:** 0.0001 – 0.1 **定義場所:**

 技術詳細

次の Densification ステップで Gaussian が削除される Sigmoid Opacity 閾値。T7 `opacityLearningRate` と Optimizer の Logit クランプロジックと連携して動作します。V393 は `.full` でデフォルトを 0.005 から 0.001 に下げました — 結果: エキゾチックな視点でのみ役割を果たす Splat はより長く残り、SH デイテールに 寄与します。V394 は 0.0001 をテスト → わずかに悪い (Prune が少なすぎ、メモリが無駄)。重要: Density Control は他の対策ですでにバッファ容量がいっぱいでも、常に Prune する必要があります (CLAUDE.md の「Density Control Must Always Prune」参照) — そうでなければ死んだ Gaussian が蓄積し、カウントが凍結します。

 簡単に言うと

Splat が削除されるのに十分「透明」と見なされるとき。0.005 が Paper 標準で、Quality では 0.001 — つまり、Splat により長いチャンスを与えます。これによりソフトライトと弱いシャドウがより良く表現可能になります。より高く (0.01 以上) 設定すると、Splat 数が急速に減少 — メモリ不足の場合に意味があるかもしれませんが、デイテールが犠牲になります。

T15 opacityResetInterval 詳細

デフォルト: 3 000 (Initializer + Paper)、100 000 (`.full` = 事実上無効化)、200 000 (`.fullMCMC` = 無効化) **範囲:** 1 000 – 100 000+ **定義場所:**

 技術詳細

反復ごとに、すべての Gaussian の Opacity が低い値 (~0.01) にリセットされます — 「凍結」 Splat を再評価するための 3DGS 論文の措置。V194 は RadianceKit の Warmup + Stochastic Training Setup + 2× 学習率では、Opacity Reset が 5.5% の品質をコストし、Logit クランプが Reset 機能をすでにカバーしていることを示しました。したがって `.full` では事実上無効化 (100 000 > 35 000 = 決してトリガーされない)。V421 は `beta2=0.99` で 3 000 ごとの Reset をテスト → 4.9% 悪い。元に戻されました。`.fullClassicPaper` (Q1.5-A、Paper 忠実テスト) では、意図的に再び 3 000 に設定されています — これは Paper マグニチュード Gaussian バジレットに到達するためのレバーの 1 つでした。

 簡単に言うと

反復ごとに、アプリがすべての Splat の可視性を「ほぼ見えない」にリセット — Opacity のリセットボタンの一種。私たちでは無効化 (値が非常に高いため発生しない)。他のメカニズムがこれを不必要にするためです。Paper 忠実な実験でのみ有効にしてください。

T16 maxScreenSize 詳細

デフォルト: 0.0 (= 無効化) **範囲:** 0 (オフ) または > 0
定義場所:

 技術詳細

Gaussian が強制的に分割される前に達成できる最大画面空間サイズ (投影ピクセル単位)。値は 0 に設定 (V48 はテストして元に戻しました) — RadianceKit の Density Control は代わりに `dMean2D` ロジックから世界空間スケール 閾値を使用します。Mip-Splatting (T74~T76) やシーン固有の Splatting 戦略を持つ 将来の実験がこれから利益を得る可能性があるため、フィールドカタログに残ります。有効化 (値 > 0、例えば 20) は、画面上で非常に大きくなった Splat を強制的に分割します — 単一の巨大な Splat がディテールを提供しない大きな滑らかな壁面に関連します。

 簡単に言うと

単一の Splat が画面でどれだけ大きくなれるかの制限。私たちではオフ。有効にすると、巨大なフラットな Splat (例えば壁) を強制的に複数の小さなものに分解します。実験する場合を除いてオフのままにしてください。

Loss (T17~T20)

T17 ssimWeight

詳細

デフォルト: 0.2 (Initializer + Paper + `.full`), 0.05 (すべての MCMC プリセット) **範囲:** 0.0 – 1.0 **定義場所:**

技術詳細

$loss = (1 - \lambda) * L1 + \lambda * D\text{-SSIM}$ の結合 Loss
関数の D-SSIM 部分の重み、ここで $\lambda = T17$ 。3DGS Paper デフォルト 0.2 は Classic Densification に最適 – V383 は 0.3 をテスト → 28.9% 悪い、V373b は 0.2 を sweet spot として確認しました。MCMC については V521b/V534 で独立して: 0.05 が最適と判明、MCMC は確率的探索を通じてより強力な L1 信号部分を必要とするため – 高い SSIM 重みは Relocation 決定を希釈します。SSIM は L1 より計算がはるかに高価です (画像全体のローカル 11x11 ウィンドウ)。RadianceKit は MPS 加速実装を使用し、1080p 画像あたり 1 ms 未満です。Q7 BayesOpt スイープは、0.05 (`.outdoorPreset : 0.082`) から 0.171 (`.indoorPreset`) の間でシーン固有の最適化を見つけました。

簡単に言うと

アプリが「各ピクセルが正しい」に加えて「構造が似ている」をどれだけ重要視するか。0.2 が標準で、良い画像を提供します。低い = ピクセル正確だが、より柔らかい遷移になる可能性があります。高い = より構造的に類似していますが、ディテールが柔らかくなります。プリセットに決定させてください。

T18 ssimWeightRefinement 詳細

デフォルト: 0.0 (= 「切り替えなし、ssimWeight を保持」) **範囲:** 0 または 0 – 1.0 **定義場所:**

 技術詳細

T2 densifyUntilIteration 後の Refinement フェーズ用のオプション SSIM 値。V428 はテスト 0.2 → Refinement で 0.3 → 16% 悪い Loss (L1 と SSIM の両方が悪化)。元に戻されたためデフォルト 0.0。フィールドの仮説は、Densification 後 — 新しい Gaussian がもう作成されない場合 — より強い SSIM 部分が構造的シャープネスを最大化するでしょう。経験的に間違い: SSIM 重みを増やすことは間接的に L1 重みを下げることの意味し、L1 は最終 Refinement フェーズで著しくより意味のある信号です。フィールドは知覚 Loss (T60) やエッジ Loss (T19) を使用した将来の実験で利用可能なまま、そこで Refinement 固有の Loss 構成が意味を持つ可能性があります。

 簡単に言うと

トレーニングの第 2 フェーズ (Splat 複製後の Refinement) の特別な設定。0.0: 前と同じ SSIM 重み付け。経験的に調整は何ももたらしません、したがってオフ。

T19 edgeLossWeight 詳細

デフォルト: 0.0 (= 無効化) **範囲:** 0 または 0.001 – 1.0 **定義場所:**

 技術詳細

V437 実験的 Loss: 画像エッジを直接比較する Sobel 勾配ドメイン L1 Loss の重み (Ground-Truth Sobel vs Render Sobel) を L1+SSIM に加えて。仮説: エッジ情報は 画像品質の知覚的礎石で、明示的な項は Gaussian にエッジをより良くヒットさせるよう 奨励すべきです。テスト結果: 重み 0.1 → 11% 悪い Loss、0.01 → 品質ニュートラルだが 10% 遅い。Sobel パスは Ground Truth と Render での追加 MPS 順方向をコストします。したがって永続的に無効化。将来のユーザー: 硬い人工エッジを持つシーン (建築、家具、レンダリング) が利益を得る可能性 — Q7 Scene-Class プリセットはそれをピックアップしませんでした、代わりに SSIM 重みをスケールリングしました。

 簡単に言うと

エッジを特に重要視する実験的追加。経験的に何ももたらしません。オフのままです。

T20 skyMaskingEnabled **詳細**

デフォルト: false (Initializer とすべてのプリセット) **範**

囲: boolean **定義場所:**

 **技術詳細**

Sky マスキングをオンにしま

す。各画像で Apple Vision Framework

(VNGenerateForegroundInstanceMaskRequest) 経由で Sky 領域がマスクアウトされ、この領域の Loss がゼロに設定されます。意味: 屋外シーンは、青/灰色/白の Sky ピクセルがアプリにそこに Gaussian を配置させることでしばしば苦しみます — 「Floater」として認識されます。Sky マスクなしでは、この領域の Loss はゼロになりません。空が画像内でわずかに変化し、アプリは Splat でそれを再構築しようと永遠に試みるためです。Vision マスクはトレーニング前にカメラごとに1回計算され、RAM に保持されます。通常は T45 skyDomeEnabled と一緒に有効化されます (Settings View の UI ロジック)。屋内シーンや合成レンダリングでは無効のままにします — マスクは天井や壁を誤って「Sky」として認識します。

 **簡単に言うと**

屋外撮影用の特別モードをオンにします: トレーニング中に空が無視され、Splat で再構築されません。すべての屋外シーンに推奨。屋内や Blender からの 3D レンダリングではオフのままにします。

SH 次数進行 (T21)

T21 shDegreeUpgradelterations

詳細

デフォルト: [1_000, 2_000, 3_000] (Initializer)、
[2_000, 5_000, 8_000] (.full、MCMC)、
[1_000, 2_000] (.preview — Degree 3 スキップ)
範囲: [Int]、各値が [0, maxIterations] に、単
調増加 **定義場所:**

技術詳細

アクティブな SH Degree が 0→1、1→2、2→3 に上昇する反復。最初のマークの前は DC コンポーネントのみアクティブ (つまり T5 shDCLearningRate)、最初のマーク後は DC + 3 つの Degree-1 係数、2 番目のマーク後は + 5 つの Degree-2 係数、3 番目のマーク後はすべての 15 係数。Gaussian あたりのメモリ需要は段階的に成長 — 4 Float → 16 Float → 36 Float → 64 Float。Quality プリセットは、ジオメトリが最初に安定してから、より高い周波数で色の詳細が来るように、Initializer デフォルトに対してアップステップを遅延させます (V228)。V384 は .full で [1K, 2K, 3K] をテスト → 9.3% 悪い — 遅延を確認。.preview は Degree 2 でキャップします、Degree 3 が 5 000 反復で収束せず Optimizer 容量を消費するためです。Q6 (T80~T81) は、このリストを動的に上書きする代替 Curriculum ロジックを提供します。

簡単に言うと

トレーニングのどの時点でアプリが、色が異なる視点から異なって見えること (ハイライト、反射) を学ぶか。後半に — まず形が正しく、次に色。プリセットの値はうまく機能するように設定されています。正確に理由を知らない限り、変更しないでください。

パフォーマンス (T22~T25)

T22 trainingRenderScale

📖 詳細

デフォルト: 1.0 (Initializer、`.full`、MCMC、Scene-Class)、0.5 (`.preview`)、0.25 (`.quickTest`) **範囲:** 0.05 – 2.0 (典型 0.25、0.5、1.0)
定義場所:

🔧 技術詳細

トレーニング画像のオリジナル解像度に対するトレーニング時のレンダリング 解像度。0.5 では各画像が 50% 幅 × 50% 高さ (つまりピクセルの 25%) に縮小され、Gaussian レンダリングはこの小さい解像度で行われます。メモリと計算量の両方を 2 次 減少させます。重要: T11 `densifyGradThreshold` は選択された解像度に一致する必要があります — 勾配マグニチュードは $1/\text{解像度}^2$ でスケールするため、`.quickTest` (0.25x) は `.full` (1.0x、 $1.1e-6$) よりはるかに高い閾値 ($4e-6$) を持ちます。RadianceKit は非常に大きな画像で警告し、自動的に調整します — 3 MP ターゲット 解像度。極端な 4K 入力画像では、0.5 または 0.25 でも意味があります。そうでなければどの Mac も CPU Compaction でのみ実行されます。

💬 簡単に言うと

トレーニング時の画像の大きさ。1.0 = オリジナル、0.5 = 半分の大きさ。半分のサイズ = 4 倍速いですが、最細部のディテールが欠落します。プリセットは正しい値を選択します。非常に大きな入力画像 (12 メガピクセル以上) では、アプリが自動的にダウンスケールします。

T23 resolutionWarmupScale 詳細

デフォルト: 0.0 (= 無効化) **範囲:** 0 または 0.1 – **定義場所:**

 技術詳細

V133 最適化: Densification フェーズ (Iter 0 から T2 まで) を Refinement フェーズより低い解像度でトレーニング。V308 は `.full` で再びオフにしました – T22 = 1.0 と Cosine Annealing で時間勝利が最小であり、品質がわずかに苦しんだため。4K 入力と長いトレーニング実行で再び意味を持つ可能性があるため、フィールド カタログに残ります – Q6 Curriculum (T80) は同様のロジックを採用しましたが、そこでは LR スケジュールに結合されています。アクティブで T80 curriculumResolutionRamp も true の場合、Q6 が勝ち、この値を上書きします。

 簡単に言うと

特別機能: 最初のトレーニング半分でより小さい画像で、2 番目で大きい画像で学習。時間を節約。オフ、新しい Q6 バリエーションがこれをより良く解決するためです。

T24 tileSize 詳細

デフォルト: 16 **範囲:** 8、16、32 **定義場所:**

 技術詳細

ピクセル単位のラスタライゼーションタイルのサイズ。Gaussian Splatting レンダリングはタイルベース: 画像は 16x16 ピクセルのタイルに分割され、各タイルは関連する Gaussian を収集し、深度でソートし、ブレンドします。16 は実質的にすべての 3DGS 実装で使用される標準で、RadianceKit Metal カーネルにハードコードされています。この値の変更はシェーダーの再コンパイルを必要とし、現在の状態では有効ではありません。将来のエンジンバージョンがタイルサイズを動的にサポートする場合のフィールドとして残ります。

 簡単に言うと

内部レンダリングパラメータ。標準 16,変更しないでください。

T25 throttleDelayMs 詳細

デフォルト: 0 (Initializer、`.full`、MCMC、Scene-Class)、0 (`.preview`) **範囲:** 0 – 100 **定義場所:**

 技術詳細

ミリ秒単位のトレーニング反復間の人工遅延。0 = フル速度 (標準)。高い値は GPU/CPU が定期的に休息を得ることでトレーニング中の Mac を「より使いやすく」します — 他のアプリの使用感が上がりますが、トレーニング時間は遅延と線形に。典型値: 1~2 ms (「軽い」スロットリング、+5% トレーニング時間、Mac がより反動的に感じる)、5 ms (「中程度」、+15% トレーニング時間)、10+ ms (「Eco」、潜在的に倍のトレーニング時間)。Inspector の「Performance」で提供されますが、標準ビューにはありません — Expert View から削除することを提案する `dev_ux-backlog.md` バックログ参照。誤解すると、トレーニング時間が劇的に延長されるためです。

 簡単に言うと

トレーニングステップ間のアプリの一時停止のミリ秒数。0 = 一時停止なし、できるだけ速く。高い値は、トレーニング中の Mac をより使いやすくしますが、トレーニング時間もより長くなります。M3 Ultra または Mac Studio では 0 のままにできます。MacBook Air では 2 または 5 が良い値でしょう。

診断と点群準備 (T26~T30)

T26 depthDistortionWeight

詳細

デフォルト: 0.0 (= 無効化) **範囲:** 0 または 0.0001 – 0.05 **定義場所:**

技術詳細

V366 実験: 深度歪み正則化 Loss の重み。レンダリング光線に沿って深く ステージされていますが概念的に同じ表面に属する Gaussian にペナルティを与えます — 集中した深度分布を奨励し、Floater を減らします。テスト: 0.01 → 4.5% 悪い、0.001 → 8.1% 悪い。理論的な利点 — マルチビュー一貫性の改善 — は L1 Loss に反映されません。仮説は暗黙的に SfM ジオメトリが正しく、Gaussian は「スタック」するだけでよいと仮定するためです。実際には、SfM 点群は通常最も弱いコンポーネントで、スタッキングではありません。特にクリーンなポーズを持つマルチビューデータセット (合成、Mip-NeRF 360 with Ground Truth) のために使用可能なままです。

簡単に言うと

同じ場所で連続した複数の Splat を避けるための実験機能。テストが何ももたらさなかったため、有効化されません。

T27 singleViewOverfit

詳細

デフォルト: false **範囲:** boolean **定義場所:**

技術詳細

診断フラグ: true の場合、Camera プールからランダムに選択する代わりに、各トレーニング反復で Camera Index 0 が強制的に使用されます。意味: モデルが単一のビューさえオーバーフィットできない場合 (つまり、View 0 の Loss が 10 000 反復後もゼロに近づかない)、順方向/逆方向パスに根本的なバグがあります。このスイッチは Metal シェーダーと微分可能ラスタライザーカーネルの開発中に集中的に使用されました — V42~V47 フェーズ。今日では、誰かがバックエンドコードを変更してリグレッションテストを実行したい場合のサニティチェックとしてのみ利用可能。CLI 経由で `--single-view`。

簡単に言うと

開発者向けのテストモード。アプリが1つの画像から学習できるかどうかを確認するために使用できます。通常のユーザーには無関係、常にオフのままにしてください。

T28 maxCameras 詳細

デフォルト: 0 (= 「すべてのカメラを使用」) **範囲:** 0
または 1 - N **定義場所:**

 技術詳細

V43 からの診断制限: 最初の N カメラのみでトレーニングし、それ以上は無視します。元の意味: 多すぎるカメラが勾配競合を生成する (同じ Gaussian に対する矛盾する Loss 信号が多すぎる) という仮説をテスト。テスト結果: 人工的な制限による体系的な利点なし — より多くのフレームは事実上常により多くの品質をもたらします。ターゲットを絞った実験のために CLI フラグ (`--max-cameras N`) として残ります。例えば「1500 画像ドローン飛行の最初の 100 画像でトレーニングは機能するか?」UI に公開されていません。

 簡単に言うと

開発者用診断フィールド — 最初の N 画像のみを使用し、残りを無視。通常のユーザーには不要、値 0 = すべての画像。より多くの画像 = より良い結果 (`feedback_more-frames-better.md` 参照)。

T29 maxInitialPoints 詳細

デフォルト: 0 (= 「すべての SfM 点を使用」) **範囲:** 0
または 1 000 - 200 000+ **定義場所:**

 技術詳細

V54 セーフガード: トレーニングを開始する初期 SfM 点の数を制限します。密な COLMAP 再構築は > 60 000 点を生成でき、大きな初期スケールではピクセルオーバーラップあたり 200~300 Gaussian になります — これは「霧の場」を作り、トレーニングが収束しません。~16 000 点へのサブサンプリング (トレーニングエンジンの Hard-cap ロジック) は、初期密度をリファレンス 3DGS が使用するレベルに引き上げ、オーバーラップを劇的に減らします。非常に密な SfM では自動的に設定されます。CLI 経由で

`--max-points N`。

 簡単に言うと

カメラ再構築から使用される開始点の数。非常に密な再構築 (60 000 を超える) では、アプリは自動的に 16 000 に制限します — そうでなければ最初に霧が多すぎます。これを設定する必要はありません。アプリが処理します。

T30 cameraClusterOutlierMultiplier **詳細**

デフォルト: 10.0 (すべてのプリセット — 決して上書きされない) **範囲:** 1.0 – 100.0 **定義場所:**

 **技術詳細**

Phase 3.10 A.1 で導入された Camera Cluster Outlier Filter の Multiplier。トレーニング前に、トレーニングエンジンはすべてのカメラ位置の重心とカメラから重心までの最大距離を計算します。重心からの距離が $\text{multiplier} \times \text{maxCameraDistance}$ を超える SfM 点は Outlier として破棄されます。デフォルト 10x は Phase 3.10 前の動作を維持します。微妙なバグ: タイトな SfM (カメラがより近い) → 小さい → 小さい閾値 → より多くの点が Outlier として破棄される。緩い SfM → より大きい閾値 → より少ない点が破棄される。これは Phase 3.9 Funnel-vs-Training 反相関の原因の1つです: より良い SfM はより多くの初期点が殺されるため、ダウンストリームでより悪いトレーニングにつながる可能性があります。フィールドは A.3 スイープ用の CLI オーバーライド (`--camera-cluster-outlier-multiplier`) として配置されています。UI に公開されていません。5 未満の値は通常制限的すぎ、20 を超える値は効果がありません。

 **簡単に言うと**

カメラ雲から遠く離れた点を破棄する特別なフィルター。10 = アプリは寛大で、ほぼすべてを保持します。遠方の点 (遠くの山) が浮遊する小さな塊のように画像で見える場合、増加することは意味があります。緊急時のみ低く設定 — 遠くのディテールを失います。

正則化 (T31~T37)

T31 coarseToFineBlurRadius

詳細

デフォルト: 0 (= 無効化) **範囲:** 0 または 1 - 10 **定義場所:**

技術詳細

V369 実験: Densification フェーズの開始時に Ground Truth 画像に適用され、Densification (T2) の終わりまで線形に 0 に減少する Box Blur 半径。仮説: Coarse-to-Fine トレーニング — 最初に粗い構造を学び、次にディテール — はより安定したジオメトリを提供するべきです。テスト: $r=3 \rightarrow 9.6\%$ 悪い, $r=1 \rightarrow 5.1\%$ 悪い。失敗の理由: Densification は画像ドメイン勾配に基づいて決定し、Blur は「ここでクローンする必要がある」に重要な信号を正確に減らします。他の Density Control スキームを使用した将来のテスト用にフィールドカタログに残ります。

簡単に言うと

実験的「最初に粗く、次にディテール」モード。何ももたらしませんでした、オフのままです。

T32 scaleRegWeight

詳細

デフォルト: 0.0 (= 無効化) **範囲:** 0 または 0.0001 - 0.05 **定義場所:**

技術詳細

V370 実験: 世界空間スケールでの L1 正則化。大きくなりすぎる Gaussian にペナルティを与え — 1 つの Gaussian で壁全体をカバーする「メガ Splat」を防ぎます。テスト: $0.01 \rightarrow 200\%$ 悪い Loss (2 M Gaussian、全爆発)、 $0.001 \rightarrow 214\%$ 悪い。理由: スケール正則化は Density Control と競合します — より小さいスケールは、より多くの Gaussian が必要になることを意味し、Density Control はより頻繁に分割し、それがさらに多くの勾配作業を意味します。無効化、ただし Mip-Splatting 実験 (T74) のために文書化されています: この文脈では、スケール下限が意味を持つ可能性があります。

簡単に言うと

Splat を小さく保つことを強制する正則化。テストで Splat 爆発 (数百万の Splat) を引き起こしました。有効にしないでください。

T33 anisotropyRegWeight 詳細

デフォルト: 0.0 (= 無効化) **範囲:** 0 または 0.0001 – 0.05 **定義場所:**

 技術詳細

V445 実験: $\max(\text{scale})/\min(\text{scale})$ 比率へのペナルティ、Floater として認識される極端に細長い「ニードル」 Gaussian を防ぐべきです。テスト: 0.01 → 69% 悪い、0.001 → 15% 悪い。理由: 正則化は Splat を「丸い」形に強制しますが、平らな表面 (壁、テーブル、床) ではまさに間違っています — ここではフラットで広い Gaussian が球形よりも効率的です。無効化。V549f は T34 `scaleRatioPruneThreshold` で代替のターゲットを絞ったアプローチを提供しましたが、それも元に戻されました。

 簡単に言うと

長すぎる細い Splat にペナルティを与える正則化。意味があるように聞こえますが、テストでは悪かったです。オフ。

T34 scaleRatioPruneThreshold 詳細

デフォルト: 0.0 (= 無効化) **範囲:** 0 または 5.0 – 100.0 (典型 10.0 – 30.0) **定義場所:**

 技術詳細

実験的ポストトレーニング Pruning、 $\max(\text{scale})/\min(\text{scale})$ 比率がここで設定された線形閾値を超える各 Gaussian を削除します。正則化だけでは排除できない極端に細長い「ニードル/ディスク」Floater をターゲットにします。テストでは Pruning は期待通り Floater を削除しましたが、同時に壁や床の意味のあるフラット Splat も削除しました — 画像はより穴があきました。したがってデフォルトでオフ、CLI フラグ (`--scale-ratio-prune N`) は対象を絞った実験のために利用可能です。それでもテストしたい場合の推奨値: 30 (非常に保守的、極端な外れ値のみを削除)、10 (積極的、ディテールをコスト)。

 簡単に言うと

トレーニング後に非常に細長い Splat をフィルタリングする試み。テストで ネットネガティブでした — Floater は除去されましたが、ディテールも除去されました。オフ。

T35 opacityRegWeight 詳細

デフォルト: 0.0 (= 無効化) **範囲:** 0 または 0.0001 – 0.05 **定義場所:**

 技術詳細

V446 実験: Opacity を 0 または 1 に引き寄せる Binary-Cross-Entropy ペナルティ (つまり「半透明」から離れる)。仮説: より鋭い Opacity 分布は画像の明瞭さを改善するべきです。T33 と組み合わせたテスト → 正則化が品質をコスト、両方無効化。無効化。注意: 1.4.3 ベータで、このフィールドが Default 値の変更 (Initializer = 0.01) で正確にあるバグが現れ、Gaussian カウントの Mass-Extinction (1 反復で 460 K → 5) になりました。1.4.4 以降、デフォルトとして 0.0 に固定。

 簡単に言うと

Splat を完全に透明または完全に固体にする正則化。何ももらしません。危険になる可能性もあります (1.4.3 バグ Mass-Extinction)。0 のままにしてください。

T36 opacityDecayFactor 詳細

デフォルト: 0.0 (Initializer = 無効化)、0.9995 (.full、.classicBalanced — HTGS 標準) **範囲:** 0 (オフ) または 0.95 – 1.0 **定義場所:**

 技術詳細

HTGS スキームの V546 実装 (Hierarchical Time-Gating, Eurographics 2025): すべての T37 opacityDecayInterval 反復で、各 Gaussian の Sigmoid Opacity にこの係数が掛けられます。100 回適用で 0.9995^x は Densification フェーズあたり ~95% 残存をもたらします — すべての Opacity に対するわずかだが持続的な下方圧力で、弱く寄与する Gaussian を確実に T14 pruneOpacityThreshold の下に下げます。結果: Decay なしの V438 と比較して Horse Full (3 試行平均 V546) で 14% 良い L1 Loss。Densification フェーズ中のみアクティブ (T2 まで)、その後トレーニングは Decay なしで続行され、Refinement で確立された Opacity が安定したままになります。MCMC では使用されません (MCMC は T67 mcmcRelocationInterval + T68 mcmcDeadOpacityThreshold 経由で独自のメカニズムを持ちます)。

 簡単に言うと

トレーニング時間にわたるすべての Splat の「ソフトフェード」。非アクティブになった Splat をより速く透明にし、クリーンアップで削除されるようにします。これは V546 更新の最も重要な品質レバーでした: 14% 良い。Quality プリセットに組み込まれています。正確に調整されているため、自分で調整することはお勧めしません。

T37 opacityDecayInterval 詳細

デフォルト: 50 **範囲:** 10 – 500 **定義場所:**

 技術詳細

T36 opacityDecayFactor が適用される反復間隔。
HTGS Paper デフォルト 50、`.full` で維持。長い間隔 (>200) は効果を部分的に取り消します。2つの適用の間で十分な勾配更新が発生し、Opacity が再び上昇するためです。より短い間隔 (<20) は Decay を積極的すぎにします。Densification フェーズでのみアクティブ。

 簡単に言うと

「フェード」が適用される頻度。
50 = 50 反復ごとに少しフェードステップ。適切です。

Refinement (T38~T44)

T38 gradientAccumulationSteps 詳細

デフォルト: 1 (= 「Adam ステップごとに 1 ビュー」)
範囲: 1 – 8 **定義場所:**

 技術詳細

V424 機能: Adam 更新が実行される前に勾配が蓄積されるビューの数。> 1 の場合、アプリは別の「unfused」逆方向プロジェクトパスで実行され、勾配を別のバッファに集計します。最終適用は 1/N でスケールしてマグニチュードを一定に保ちます。V424 は 2 ビューをテスト → 品質ニュートラルだが 10% 遅い (unfused パスが fused パスより高価なため)。`.full` では元に戻されましたが、MCMC では意図的に使用 — `.fullMCMC` は で実行されますが、V544a テストは Classic への品質ギャップが (11% の代わりに) 5% に縮小すると示しました。Initializer デフォルト 1、現在のプリセットで 1、CLI フラグ (`--accum-steps N`) として残ります。

 簡単に言うと

アプリが Splat を調整する前に見る画像の数。1 = 各画像個別。高い = 複数の画像を同時に見て、平均を適用。標準ケースでは何ももたらしません。MCMC では 2 が少し役立つ可能性があります。

T39 testViewIndices 詳細

デフォルト: `[]` (= 空、すべてのビューがトレーニングに使用) **範囲:** `Set<Int>`、Camera インデックスの任意のサブセット **定義場所:**

 技術詳細

V546 機能: トレーニングに使用されず、PSNR/SSIM/LPIPS 評価のための Holdout として節約される Camera インデックスのセット。 `--benchmark` CLI フラグがアクティブな場合に自動的に設定されます: その場合、インデックス 0 から始まる 8 番目ごとのビュー (LLFF 標準、Mip-NeRF 360 および 3DGS 論文規約と同一)。Benchmark なしでは空 — トレーニングはすべてのビューを使用します。 **注意:** インデックスを理解せずにこのフィールドを手動で設定すると、Benchmark が使用不可になる可能性があります (例えば N より上のすべてのインデックスを設定するが、N-50 ビューしかない場合 → Holdout なし → 評価なし)。独自のプリセットエクスポートでは、testViewIndices はシーン依存で、異なるデータセット間で意味のない値を残すため、永続化されません。

 簡単に言うと

後の品質測定のためにトレーニング時に「保留」される画像。これを自分で設定しません。`--benchmark` フラグが自動的に行います (8 番目ごとの画像がテスト)。独自のインデックスを設定する場合: 危険、Benchmark を歪める可能性があります。

T40 refinementPruneInterval 詳細

デフォルト: 0 (= 無効化) **範囲:** 0 または 100 – 5 000 **定義場所:**

 技術詳細

V425 機能: Refinement フェーズ中 (T2 後) の N 反復ごとに、 $\text{sigmoid}(\text{opacity}) < T41$ refinementPruneOpacityThreshold の Gaussian を削除する追加の Prune パスが実行されます。意味: Densification 中は定期的な Density Control コールがあり、その後はありません — Opacity がさらに低下する Gaussian はバッファに残ります。V425 はテストして元に戻しました: 追加の Pruning は V426 (Two-Phase Densification、これも 0 Gaussian カスケード失敗で中止) と関連しました。無効化。実験用 CLI フラグ利用可能。有効化された場合、1 000 または 2 000 が意味のある値です。

 簡単に言うと

Refinement フェーズ中の追加のクリーンアップ。何ももたらしません、オフのままです。

T41 refinementPruneOpacityThreshold 詳細

デフォルト: 0.0 (= 「 T14 を使用」) **範囲:** 0 または 0.001 – 0.1 **定義場所:**

 技術詳細

V425b: Refinement Pruning 用の別の Opacity 閾値。Densification 後、ほとんどの Gaussian は著しく高い Opacity (> 0.001) に達しているため、標準 T14 pruneOpacityThreshold は緩すぎます。T40 がアクティブな場合、このフィールドが独自の閾値を決定します。0.0 では T14 が引き続き使用されます。T40 > 0 の場合のみ 関連します。

 簡単に言うと

追加の Refinement クリーンアップの閾値 (T40 参照)。両フィールド非アクティブ、したがって無関係。

T42 midTrainingCompactificationIterations 詳細

デフォルト: [] (= 無効化) **範囲:** [Int]、(densifyUntilIteration, maxIterations) の値 **定義場所:**

 技術詳細

V549 機能: Refinement フェーズ中の明示的な反復ポイントで、Compactification パスが実行されます (sigmoid(opacity) < 0.01 + 外れ値スケール Gaussian を削除、T56 postTrainingCompactification と同じロジック)。意味: 長い Refinement フェーズは Confetti/Floater 蓄積を示す可能性があり、その SH はビュー固有のアーティファクトにオーバーフィットします。有効化された場合の典型的な構成: [10000, 20000, 30000] for 40K Classic。しかし: V549 A/B テストは Family データセットで、すべての構成でより悪い L1 を示しました: [10K, 20K, 30K]@0.01 → -48% カウントだが +36% L1。 [20K, 30K]@0.005 → -44% カウントだが +45% L1。 [20K, 30K]@0.001 → -17% カウントだが +87% L1。したがって無効化。CLI フラグ --mid-compact "10000,20000" が利用可能、視覚的 Floater トレードオフ (Viewport で Confetti が少ない) を Loss リグレーションよりも優先する場合に。

 簡単に言うと

トレーニング中の中間クリーンアップアクション。テストでは、クリーンアップが最終結果を悪化させました (Floater は少ないが、デテールも少ない)。オフ、Floater が少しぼやけた画像よりも気になる場合は CLI 経由で有効にできます。

T43 frustumCullEnabled 詳細**デフォルト:** false **範囲:** boolean **定義場所:** 技術詳細

V549b 機能: トレーニング後、すべてのトレーニングカメラ Frusta の Union の 外側にあるすべての Gaussian が削除されます。そのような Gaussian は Loss 信号によって決して制約されず、常に Floater です。Novel View がカメラパスの後ろまたは横にあるシーン (例えば線形ドローン飛行の裏側) に特に効果的です — その Floater はトレーニングフェーズで決して見えませんが、後で 3D ビューアで動くときに非常に見えます。V549b A/B はドローン飛行で肯定的な結果、したがって Opt-In として利用可能。デフォルト false。完全なオービットカバレッジを持つ Object キャプチャでは Frustum Union がシーン全体を含み、機能が何も削除しないため — Settings の「Floater Reduction」で提供され、Q9 Outdoor プリセットで T44 frustumCullExpansion 経由で 暗黙的にテストされます (Q7 BayesOpt はそれをアクティブにしませんでした。Outdoor Sky-Dome が同じ問題をより良く解決するため)。

 簡単に言うと

ドローン飛行や線形撮影用の特別フィルター: トレーニング後、どのカメラにも「見られていない」Splat が削除されます。Settings でオプションでオン可能。単純なオブジェクト撮影では不要。

T44 frustumCullExpansion 詳細**デフォルト:** 1.1 **範囲:** 1.0 – 2.0 **定義場所:** 技術詳細

T43 frustumCullEnabled の NDC マージン。1.0 は画像境界で正確に切断し、ぼやけた Splat を境界で切り詰めすぎます。1.1 = 正確なカメラフレーミングを超えた 10% パディング — わずかにオフセットされた Novel View で見える可能性のある境界 ピクセルに対する許容範囲を与えます。> 1.2 の値は Cull を実質的に無効にします。拡張された Frustum がはるかに多くの空間をカバーするためです。

 簡単に言うと

上記のフィルターがどれだけ厳密にトリミングするか。1.1 = 画像境界へのわずかな安全マージン。値はそのままだ。

Sky-Dome (T45~T48)

T45 skyDomeEnabled

詳細

デフォルト: false (Initializer + P9 Outdoor を除くすべてのプリセット) **範囲:** boolean **定義場所:**

技術詳細

V549e 機能: トレーニング開始前、球状点群が生成され (Fibonacci sphere、T46 Sample 点)、シーン中心の周りの T47 skyDomeRadiusMultiplier × scene_extent の半径に配置され、すべてのトレーニングカメラの Sky マスクされたピクセルからの色 (T20 skyMaskingEnabled 参照) で初期化されます。これらの Sky-Dome Gaussian は Gaussian バツファの先頭に挿入され、トレーニング中に「凍結」されます (Position/ Scale/Rotation 勾配 = 0、SH と Opacity のみ最適化可能)。効果: 遠くの黒い「Confetti」領域の代わりに、Novel View で実際の空が表示されます。V549e MVP はドローンと風景 シーンで非常にうまく機能します。P9 Outdoor プリセットでデフォルトオン。室内シーンでは オフのまま — 球は部屋の外に意味なくぶら下がります。

簡単に言うと

シーン周りの人工「Sky Dome」をオンにします。屋外撮影をはるかに美しくします: 画像の境界の黒い小さな塊の代わりに、アプリは実際の空を表示します。ドローン飛行と風景には必須、室内では意味がありません。

T46 skyDomeSampleCount

詳細

デフォルト: 5 000 **範囲:** 1 000 – 50 000 (典型 2 000 – 10 000) **定義場所:**

技術詳細

Sky-Dome 球面上の Fibonacci Sphere Sample 点の数。高い値 → より密な Sky-Dome (大きな解像度と多くの可視天空でより良い) ですが、より多くのメモリ需要。5 000 は 4K レンダリング用のスイートスポット。低い解像度では 2 000~3 000 で十分。点は各トレーニングカメラ View ベクトルへの Cosine Distance によって、対応する Sky マスクされたピクセルで初期化されます — View Cone がカメラを見ない Sample 点は低い Opacity 初期値で背景に残ります、しかしトレーニングで変更されません (凍結)。

簡単に言うと

人工的な空がどれだけ密か。5 000 点で通常十分です。より多い = 遠くからのより良い遷移、ただし少しメモリをコスト。

T47 skyDomeRadiusMultiplier **詳細**

デフォルト: 30.0 (Initializer + ほとんどのプリセット)、59.0 (P9 Outdoor、Q7 BayesOpt 最適化) **範囲:** 5.0 – 200.0 **定義場所:**

 **技術詳細**

シーン範囲 (= カメラ位置間の平均距離) に対する Sky-Dome 球の半径。30 = 球は カメラ雲の直径の 30 倍を持ちます。小さすぎる (< 5) → Sky-Dome はシーン自体と干渉します (例えば Sky-Dome Splat が前景に着地)。大きすぎる (> 100) → Sky-Dome 位置で float32 精度損失、遠くでレンダリンググリッチを引き起こします。Q7 BayesOpt on Bicycle (Mip-NeRF 360) は屋外用のシーン固有最適化として 59.0 を見つけました — これは標準 30.0 が深い風景に小さすぎ、Sky-Dome ピクセルが画像境界領域で「壁」として見えるようにレンダリングされることを示唆しています。

 **簡単に言うと**

人工 Sky Dome がどれだけ遠くにあるべきか。30 = かなり遠い。大きな風景では 50~60 がより良い (Outdoor プリセットは自動的にそれを行います)。小さすぎることは、レンズの直前に塊があるようなものです。

T48 frozenGaussianCount **詳細**

デフォルト: 0 (= 凍結された Gaussian なし) **範囲:** 0 または 1 – T46 **定義場所:**

 **技術詳細**

Optimizer で Position/Scale/Rotation 勾配がゼロに設定される、バッファの先頭にある Gaussian の数 — 全トレーニングにわたって空間的に固定されたままです。Density Control はそれらをクローン、分割、または Prune してはいけません。Sky-Dome 注入用 (T45 参照): Sky-Dome がオンの場合、このフィールドは自動的に T46 skyDomeSampleCount に設定されます。手動設定は可能 (例えば LiDAR スキャンから事前配置された点群を凍結) ですが、UI から直接アクセスできません。重要: バッファの最初の N Gaussian は常に frozen — バッファ内の順序が決定し、明示的なインデックスではありません。

 **簡単に言うと**

開始時に固定で動けない Splat の数。Sky-Dome がオンのとき、自動的に Sky-Dome カウントに設定されます。自分で調整する必要はありません。

Adam + LR スケジュール (T49~T55)

T49 adamResetIteration

📖 詳細

デフォルト: 0 (= 無効化) **範囲:** 0 または 100 – **定義場所:**

🔧 技術詳細

V430 機能: Adam Optimizer Momentum アキュムレータ (m1、m2) がゼロにリセットされる反復。その後のバイアス修正は `iter` の代わりに (`iter - adamResetIteration`) で実行されます。V430 は 5 000 (Densification 終了後) でリセットをテスト → 12.8% 悪い Loss。理由: Densification 中に蓄積された Adam Momentum は典型的な勾配マグニチュードに関する情報を運び、Refinement フェーズを加速します。それを捨てることは Refinement の最初の ~500 反復の収束をコストします。無効化。研究実験用 CLI フラグとして残ります。

💬 簡単に言うと

内部 Adam Optimizer 「メモリ」のリセットボタン。テストで害を及ぼしました、オフのままです。

T50 positionLRScheduleEndIteration

📖 詳細

デフォルト: 0 (Initializer = 「maxIterations を使用」)、20 000 (`.full` — Cosine が `maxIter=35K` にもかかわらず 20K で終了)、30 000 (`.fullClassicPaper`) **範囲:** 0 または 1 000 – **定義場所:**

🔧 技術詳細

V431 機能: Position LR の Cosine Annealing 曲線が最小に達する反復。0 の場合、`T1 maxIterations` と同じです。> 0 の場合、スケジュールはこの値まで実行され、その後 `T4 positionLearningRateFinal` で一定のままです。これにより、最小だが一定の学習率を持つ「拡張された Refinement フェーズ」が可能になり、新しい Decay なしで位置をゆっくり洗練します。`.full` はこれを行います (Schedule End at 20K、Training runs until 35K)、V434c/V434d は確認しました: 15K と 25K の両方が約等しく、20K がわずかに最適。`T51` と組み合わせて使用され、拡張フェーズの非位置 LR も修正します。

💬 簡単に言うと

アプリが Position 学習率の低下を止めるとき。最大反復より低い場合、その後は一定の Mini レートで実行されます — 非常に遅いが非常に安定して洗練します。Quality プリセットに組み込まれているため、調整する必要はありません。

T51 extendedPhaseLRDecay 詳細

デフォルト: 0.0 (= 無効化、一定の LR) **範囲:** 0 または 0.01 – 1.0 **定義場所:**

 技術詳細

V433 機能: 「拡張フェーズ」での非 Position LR (スケール、回転、Opacity、SH) の最小乗数 — つまり: T50 に達した後、Position LR がすでに T4 の場合。0.1 の場合、スケール/回転/Opacity/SH は、それぞれ 1.0 (= 標準 LR) からその標準の 0.1x に Cosine Decay されます。0.0 (デフォルト) では、それらは一定のままです。V457 は完全な Decay (0.0 = ゼロへの Decay) を no-Decay に対してテストし、見つけました: 平均 0.0400 (2 実行) = V438 と同じ Loss、Decay なし。Decay でよりクリーンな動作ですが、測定可能により良くありません。したがって無効化。CLI に `--nonpos-lr-scale F` として残ります。

 簡単に言うと

後半の Refinement フェーズでも色と形の学習率を小さくする。トレーニングを「より安定」にしますが、経験的により良くありません。オフ。

T52 adaptiveDensifyThreshold 詳細

デフォルト: false **範囲:** boolean **定義場所:**

 技術詳細

V440 実験: true の場合、アプリは各 Densification ステップで現在の勾配分布の p98 を計算し、それを動的閾値として使用します (T11 の設定値の少なくとも 0.5x にクランプ、あまり外れないように)。仮説: 現在のシーンフェーズへの自動調整は、Density Control をより堅牢にします — 例えば最初は厳密な Pruning、後で緩い、または逆。V440 はテストして元に戻しました: 63 K Gaussian (Mass-Pruning) への壊滅的なドロップ、最初の反復で p98 が非常に高く、その後ほとんど何も閾値を超えないためです。固定閾値はすでによくキャリブレートされており、動的調整は害よりも害が多いです。Q5 (T77) は問題を回避する rolling median 経由の代替適応ロジックを提供します。

 簡単に言うと

Densify 閾値の適応バージョン。テストで壊滅的でした (Splat 数が 63K にクラッシュ)。オフ。Q5 はそのより良いバリエーションを持っています。

T53 mergeAfterDensification 詳細

デフォルト: false (Initializer)、true (`.full`、`.classicBalanced`、`.fullClassicPaper`) **範囲:** boolean **定義場所:**

 技術詳細

V438 機能: Densification フェーズの終わり (Iter `T2`) で、類似の スケールと色を持つ近接 Gaussian を結合する 1 回限りの Merge パスが実行されます。Gaussian 数を視覚的品質損失なしで通常 5~15% 削減します。意味: 集中的なクローニング後、何も新しいものを寄与しない準同一の Gaussian のクラスタが形成されます — マージは他の領域のための Optimizer 容量を解放します。Classic Quality プリセットで標準。MCMC では使用されません。MCMC は Relocation ロジックを通じてそのようなクラスタを最初から形成させないためです。

 簡単に言うと

Splat 複製フェーズの終わりに、ほぼ同一のクローンを結合します。視覚的効果なしでデータ量を削減します。Quality プリセットでデフォルトオン。

T54 densifyPhase2FromIteration 詳細

デフォルト: 0 (= 無効化) **範囲:** 0 または `T2` - `T1` **定義場所:**

 技術詳細

V426 実験: Refinement 休止後にこの反復から開始し、`T55` まで実行する 2 番目の Densification フェーズを有効にします。仮説: Refinement フェーズ後、勾配 アキュムレータはより安定したマグニチュードを持ち、どの領域がまだ追加の Gaussian を必要とするかをより正確に伝えることができます。V426 はテストして元に戻しました: Two-Phase Densification は 0 Gaussian カスケード失敗に陥りました (V425 Refinement Pruning と組み合わせるとバッファを破壊)。無効化。実験用 CLI フラグ利用可能。

 簡単に言うと


休止後の 2 回目の複製ラウンド。テストで Splat 在庫を破壊しました。オフ。

T55 densifyPhase2Untillteration 詳細

デフォルト: 0 範囲: 0 または T54 - T1 定義場所:

 技術詳細

V426 Two-Phase Densification の終了。T54 > 0 の場合のみ関連。両フィールド一緒に無効化。

 簡単に言うと

2 回目の複製ラウンドの終了 (T54 参照)。両方オフ。

ポスト処理 + Apple AI (T56~T60)

T56 postTrainingCompactification 詳細

デフォルト: true (すべての Production プリセット)、false (.quickTest 、 .preview) 範囲: boolean 定義場所:

 技術詳細

V443 機能: トレーニング終了後、sigmoid(opacity) < 0.01 の Gaussian がハード除去されます (実質的に画像にもう寄与しません)。Gaussian カウントを通常 58%、エクスポートファイルサイズを 55% 視覚的品質損失なしで削減します。Production プリセットでデフォルトアクティブ — 最終結果はできるだけコンパクトに配信できるべきです。 .quickTest ではオフ、診断実行はとにかくエクスポートされないためです。T42 midTrainingCompactificationIterations (V549) とは異なり、Compactification は終了時のみ行われます — Refinement はそれまですべての Gaussian を使用できます。

 簡単に言うと

トレーニング後のクリーンアップ: ほぼ見えない Splat が削除されます。エクスポートファイルを品質損失なしで約半分のサイズにします。必須機能、診断実行でのみオフにします。

T57 metalFXUpscaling 詳細

デフォルト: false **範囲:** boolean **定義場所:**

 技術詳細

V444 機能: 3D ビューア出力のバイリニア補間の代わりに Apple の MetalFX Spatial Upscaler を有効化します。トレーニング解像度 < Viewport サイズ (例えば 0.5x でトレーニング、フル解像度で Viewport 表示) の場合、MetalFX は著しく鮮明な画像を提供できます。Viewport でライブに変更され、再トレーニング不要。T58 mpsLanczosScaling と相互排他 — MetalFX が優先。推奨: 期待されるディテールと比較して Viewer の画像が「ぼやけて」見える場合は、オンにします。

 簡単に言うと

3D Viewer での Apple-ML ベースの画像シャープニング。より低い解像度でトレーニングして結果をフルスクリーンで表示する場合に役立ちます。ライブトグル、お試しください。

T58 mpsLanczosScaling 詳細

デフォルト: false **範囲:** boolean **定義場所:**

 技術詳細

V444 機能: Viewport スケーリング用のバイリニア補間の代わりに MPSImageLanczosScale。Lanczos は古典的な Sinc ベースのリサンプリング手順で、バイリニアより著しく鮮明な結果を最小オーバーヘッドで提供します。ライブトグル。両方がオンの場合、T57 で上書きされます。

 簡単に言うと

3D Viewer 用の古典的なシャープニング手順 (Lanczos)。MetalFX (T57) は ML ベースで通常より良いですが、Lanczos はあまり積極的でない代替手段です。

T59 livePreviewInterval 詳細

デフォルト: 50 (Initializer とほとんどのプリセット) **範囲:** 0 (オフ) または 10 – 5 000 **定義場所:**

 技術詳細

トレーニング中に 3D Viewer が現在の Gaussian で更新される頻度。50 = 50 反復ごとに Viewer の新しいレンダリング — 進行を観察するのに十分良く、トレーニングを遅くしません。0 = Viewer はまったく更新されません (バックグラウンドトレーニング、最大速度)。典型的な調整: `.quickTest` で 10 まで下げる (各ステップを見たい)、長い MCMC 実行で 500~2000 まで上げる (合計の更新オーバーヘッドが顕著)。

 簡単に言うと

トレーニング中に 3D プレビューがどれくらいの頻度で更新されるか。50 = 50 反復ごと。高いほど = 頻度が少ない = やや速いですが、進行を見る頻度が少なくなります。0 = プレビューなし (最大速度)。

T60 perceptualLossWeight 詳細

デフォルト: 0.0 (= 無効化) **範囲:** 0 または 0.001 – 0.5 **定義場所:**

 技術詳細

V444 将来機能: MPSGraph 経由の知覚 Loss 項の重み (VGG ライクな小さなネットワーク)。L1+SSIM より高いセマンティックレベルで構造的およびテクスチャ類似性をキャプチャします — 通常、研究パイプラインで、「ピクセルパーフェクト」が「リアルに見える」より重要でない場合。実装はまだ保留中 (コードスタブは存在しますが、順方向パスは実装されていません)。デフォルト 0.0。将来のアクティベーション用に フィールドカタログに残ります。CLI フラグ `--percep-weight F` 予約。

 簡単に言うと

「ピクセル正確」の代わりに「自然に見える」を目指す AI 支援機能の計画。まだ完全に実装されていません。

MCMC Densification (T61~T73)

T61 densificationStrategy

📖 詳細

デフォルト: `.classic` (Initializer + Classic プリセット)、`.mcmc` (すべての MCMC プリセット + Scene-Class) **範囲:** `.classic` または `.mcmc` **定義場所:**

🔧 技術詳細

Classic Densification (Clone/Split/Prune、Kerbl et al. 2023) と MCMC Densification (Stochastic Gradient Langevin Dynamics with Relocation、Kheradmand et al. NeurIPS 2024) の間で選択します。

`.classic` では T11~T16 が評価され、`.mcmc` では T62~T73 が評価されます。切り替え時の注意: Classic デフォルトと MCMC デフォルトは完全に異なってキャリブレートされています — Expert View でピッカーをフリップして、適切なプリセットをロードしない場合、1.4.3 バグスタイルの Mass-Extinction を冒険します (1 反復で 460 K → 5、MCMC OpacityReg 0.01 が Classic Opacity を殺すため)。したがって、MCMC Init デフォルトは意図的に「ソフト化」されています (すべての Reg 値 0.0)。

💬 簡単に言うと

Splat を増やすのにどのアルゴリズムが使用されるか。Classic = 元の方法 (速い、多くの Splat)。MCMC = 新しい方法 (遅い、はるかに少ない Splat, その代わりよりコンパクト)。プリセットは正しいものを選択します。適切なプリセット (P5~P7 または P8~P10) もロードする場合のみ、自分で切り替えてください。

T62 mcmcMaxGaussians

📖 詳細

デフォルト: 150 000 (Initializer + `.fullMCMC` + `.mcmcBalanced`), 100 000 (`.mcmcPreview`), 1 500 000 (`.fullMCMCMip` — Mip-Splatting バリエーション with 10× バジレット), 1.19 M (`.renderPreset`), 1.25 M (`.outdoorPreset`), 670 K (`.indoorPreset`) **範囲:** 0 (= 「バッファ容量を使用」) または 10 000 – 5 000 000 **定義場所:**

🔧 技術詳細

MCMC 戦略での Gaussian 数のハード上限。数はキャップに達するまで Relocation ステップごとに `T70 mcmcGrowthRate` (通常 5%) で徐々に増加します。V473/V531 は 150 K をスイートスポットと見つけました — 200 K を超えると Splat 品質が薄まり (小さく冗長な Gaussian が多すぎる)、100 K 未満ではシーンが過小 Densify されたまま。非常に大きなシーン (例えば 1 545 枚のドローン飛行で 158 K SfM-init) では 150 K が低すぎる — したがって 1.4.5 拡張 `T72 mcmcCapMultiplier` + `T73 mcmcAutoScaleByScene`。Q7 BayesOpt はシーン固有最適化を 670 K (Indoor) から 1.25 M (Outdoor) の間で見つけました。値 0 では、エンジンは完全なバッファ容量をキャップとして使用します。

💬 簡単に言うと

MCMC での最大 Splat 数。150 000 が標準で、ほとんどのシーンに十分です。Outdoor と Render プリセット (P8,P9) はより詳細なシーンのために 1 百万以上になります。高く設定するとディテールがもたらされますが、メモリをコスト。下に設定することはむしろ緊急ブレーキです。

T63 mcmcNoiseScale 詳細

デフォルト: 0.00005 (5e-5 = Paper デフォルト) **範囲:** 1e-6 – 1e-3 **定義場所:**

 技術詳細

各 MCMC 反復で各 Gaussian の位置に追加されるガウスノイズの乗数 (SGLD ロジック)。高い = より多くの探索 (Gaussian がより放浪し、潜在的により良い場所を見つける)、低い = より多くの活用 (Gaussian がすでに良い場所に留まる)。V467 と V536 は 5e-5 を最適と確認 — 1e-5/2e-5 探索が少なすぎ、1e-4 多すぎ (Splat が流れる)。トレーニング時間にわたって T69 mcmcNoiseDecayEnd まで Cosine Decay されます — Decay 範囲の終わりでノイズは事実上 0 で、Gaussian は収束します。

 簡単に言うと

アプリが Splat に許可するランダムな「揺れ」の量、Splat 自身が最適な場所を見つけられるように。標準値は最適にテストされています。これを上げると、Splat が落ち着かなくなります。

T64 mcmcOpacityRegWeight 詳細

デフォルト: 0.0 (= RadianceKit デフォルトで無効化、Paper: 0.01) **範囲:** 0 または 0.001 – 0.05 **定義場所:**

 技術詳細

Opacity 用の MCMC 固有 L1 ペナルティ。Paper デフォルト 0.01 (未使用の Gaussian をゼロに押し、Relocation 用に利用可能にします)。しかし V464b は示しました: Reg なしでは RadianceKit で測定可能により良い (Session 28 確認)。理由: T68 mcmcDeadOpacityThreshold で定義された Pruning 基準は単独で十分 — 追加の L1 ペナルティは、貴重で低 Opacity の Gaussian も死亡させます。したがってデフォルト 0。注意: 1.4.3 ベータビルドでは Initializer デフォルトが誤って 0.01 で、Mass-Extinction バグが発生しました (T61 説明参照)。1.4.4 以降 0.0 に固定。

 簡単に言うと

MCMC 特別正則化。他の MCMC メカニズム (T68 の閾値) がすでにカバーするため、オフ。0 のままに。

T65 mcmcScaleRegWeight 詳細

デフォルト: 0.0 (= 無効化、Paper: 0.01) **範囲:** 0 または 0.001 – 0.05 **定義場所:**

 技術詳細

スケール固有値用の MCMC 固有 L1 ペナルティ。
Paper デフォルト 0.01。V464b: Reg なしでより良い、
T64 と同じ理由。すべての RadianceKit MCMC プリ
セットで無効化。T64 と同様の注意: 1.4.3 バグ。

 簡単に言うと

T64 のように、しかし Splat サイ
ズ用。オフ。

T66 mcmcRelocationInterval 詳細

デフォルト: 100 (Initializer + すべての MCMC プ
リセット、Paper 標準)、155 (P9 Outdoor — Q7
BayesOpt 最適化) **範囲:** 50 – 500 **定義場所:**

 技術詳細

MCMC が死んだ Gaussian ($\text{sigmoid}(\text{opacity}) < T68$
 $\text{mcmcDeadOpacityThreshold}$) を 新しい位置に再配置す
る反復間隔。V537 は 50 (混乱しすぎ、Loss 変動)
と 200 (わずかに 悪い、MCMC が反応性を失う) をテ
スト。100 が最適。Q7 BayesOpt on Bicycle はシーン
固有最適化として 155 を見つけました — やや長い間
隔は、次の Reloc イベントが それらを圧力下に置く前
に、Adam に新しく配置された Gaussian を統合する
時間をより 多く与えます。

 簡単に言うと

MCMC が死んだ Splat をどこかに
移動する反復ごとに。100 が標
準。自分で調整する必要はありま
せん — Outdoor プリセットはす
でに最適値を持っています。

T67 mcmcWarmupIterations 詳細

デフォルト: 500 **範囲:** 100 – 5 000 **定義場所:**

 技術詳細

MCMC Relocation がまだ発生しない初期反復の数。この Warmup 後、Reloc ロジックが開始されます。意味: 最初の反復では、Opacity 値はまだ落ち着いていません — すぐに Reloc で開始されると、Gaussian は間違った場所に配置され、すぐに再び移動される必要があります、Adam Momentum を破壊します。Paper デフォルト 500。RadianceKit はこの値を引き継ぎます、V464b がそれが堅牢であることを示したためです。

 簡単に言うと

MCMC が Splat の再配置を開始する前に「到着」できる反復数。500 が標準で適切です。

T68 mcmcDeadOpacityThreshold 詳細

デフォルト: 0.005 (Initializer、Paper 標準)、0.01 (`.fullMCMC` とすべての MCMC プリセット — V535 最適化) **範囲:** 0.001 – 0.05 **定義場所:**

 技術詳細

Gaussian が「死んだ」とみなされ Relocation の対象となる sigmoid(Opacity) 閾値。V535 は 0.01 を最適と発見 (0.005 わずか、0.02 悪い)。高い = より積極的な Reloc (より多くの Gaussian が移動)、低い = より慎重。0.01 はおおよそ「0.5% 視覚的 可視性」に相当します。P10 Indoor は Q7 BayesOpt 経由で 0.0142 を最適として使用します。

 簡単に言うと

Splat が MCMC によってどこかに移動される「死亡」とみなされる透明度。0.01 は私たちのテストで最適です。自分で調整する必要はありません。

T69 mcmcNoiseDecayEnd 詳細

デフォルト: 0 (Initializer = 「Decay なし」)、160 000 (`.fullMCMC` = 200K の 80%)、96 000 (`.mcmcBalanced` = 120K の 80%)、40 000 (`.mcmcPreview`) **範囲:** 0 または 1 000 – **定義場所:**

 技術詳細

T63 `mcmcNoiseScale` ノイズが完全にゼロに減衰される反復 (Iter 0 から ここまでの Cosine Decay)。V497c/V502 は `maxIterations` の 80% を最適と発見 — MCMC に十分な探索時間を与え、最後の 20% をノイズなしでの収束に残します。0 = すべての反復にわたって一定のノイズ (まれに意味があり、MCMC は収束できません)。

 簡単に言うと

Splat のランダムな「揺れ」がいつ止まるか。MCMC プリセットでは総反復の 80% で — 最初に探索、次に収束。値はそのままに。

T70 mcmcGrowthRate 詳細

デフォルト: 0.05 (Paper 標準 = 5%) **範囲:** 0.01 – 0.2 **定義場所:**

 技術詳細

Relocation ステップごとの MCMC 人口ターゲットの成長率。ロジック: 各 Reloc イベントで、ターゲット人口サイズが T62 `mcmcMaxGaussians` (または T72/T73 経由でスケールされたバリエント) に達するまで $(1 + \text{growthRate})$ で増加されます。V512/V522 は 0.05 を最適と発見 — 高い値は速すぎる成長になります (Adam Momentum が統合する前に Gaussian が挿入される)、低い値は終了時にシーンが過小 Densify されたままになります。

 簡単に言うと

MCMC で Splat 数がどれだけ速く成長するか。ステップあたり 5% が最適。値はそのままに。

T71 mcmcSigmoidK 詳細**デフォルト:** 100.0 **範囲:** 10.0 – 500.0 **定義場所:** 技術詳細

MCMC Noise 減衰用の Sigmoid Sharpness パラメータ。SGLD ステップでは、Gaussian あたりのノイズがで減衰されます — 高 Opacity Gaussian (Logit が正) は、低 Opacity よりも指数的に少ないノイズを受け取ります。K = 100 は鋭く、つまり「フルノイズ」から「ノイズなし」への遷移は Opacity 0.5 周辺で非常に速く発生します。V484~V487 は K = 100 を最適と発見 — 小さい値 (10~50) は高 Opacity Gaussian も揺らがせます (収束した Gaussian を破壊)、大きい (> 500) は遷移を人工的に硬くし、死んだ Gaussian がもう移動されません。

 簡単に言うと

MCMC が「移動するのに十分透明」と「固体、触らないで」をどれだけ鋭く区別するかを決定する特別パラメータ。標準値が最適です。調整しないでください。

T72 mcmcCapMultiplier 詳細**デフォルト:** 3.0 (Initializer + `.fullMCMC`)、2.0 (`.mcmcPreview`)、2.5 (`.mcmcBalanced`)、2.98 (P8 Render)、5.32 (P9 Outdoor)、1.76 (P10 Indoor) **範囲:** 0 (= 無効化) または 1.0 – 10.0 **定義場所:** 技術詳細

1.4.5 機能: シーン適応 Cap スケーリング。T73 `mcmcAutoScaleByScene` が true の場合、実効 Cap は (バッファ容量にクランプ) として計算されます。背景: 大きなシーン (例えば 1545 枚のドローン飛行 → 158 K SfM-init) では `T62 = 150 000` が低すぎる — Density Control はまったく成長できません。Multiplier 3.0 で、この例では Cap が 474 K にスケールされます (158 K × 3.0)。Q7 BayesOpt はシーン固有の最適化を見つけました: Outdoor は高 Multiplier (5.32 → ~830 K Cap at 156 K bicycle-init)、Indoor は 1.76 で満足 (壁がより早く飽和)。完全な Cap 解決はメソッド参照。

 簡単に言うと

Splat Cap をシーンサイズに自動的に適応させる Multiplier。大きなシーン = より多くの開始点 = 高い Cap。標準 3x はほとんどのシーンに適合します。Outdoor プリセットは 5x (大きな奥行き範囲)、Indoor は 1.76x (壁が制限) になります。

T73 mcmcAutoScaleByScene 詳細

デフォルト: true (Initializer + すべての MCMC プリセット) **範囲:** boolean **定義場所:**

 技術詳細

1.4.5 機能: scene-aware Cap ロジック (T72 + 参照) のマスタースイッチ。false の場合、T62 mcmcMaxGaussians のみが Cap として使用されます (1.4.4 動作に戻る)。デフォルトでオン、1.4.3 から大きなシーンでの Mass-Extinction 問題がそれなしで再発するためです。明示的にハード Cap を設定したい場合のみ手動で無効化 — 例えば最終サイズが計画可能な 150 K バリエントをトレーニングするため。

 簡単に言うと

Splat Cap をシーンサイズに自動的に適応させるオン。デフォルトでオン。特定の Splat 数を正確に持ちたい場合のみオフにします。

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

ステータス: Q1.5 は 2026-05-25 に、14 の自律反復 + Overnight 1.5M Confidence Check の後、「closed no-win」として破棄されました (max $\Delta@2\times = +0.27$ dB、オリジナルゲートは $0.5\times/2\times$ にわたる平均 $\geq +1.5$ dB を要求、11 ペアシーンの 0/11 で FAIL)。フィールドは研究実験用に **opt-in** のままです。すべての Production プリセットは持ちます。判定参照: docs/plans/2026-05-25-phase-q1.5-final-verdict.md。

T74 useMipSplatting 詳細

デフォルト: false (すべての Production プリセット)、true (`.fullMCMCMip` — 研究兄弟) **範囲:** boolean **定義場所:**

 技術詳細

Mip-Splatting を有効化 (Yu et al. CVPR 2024): 3D Smoothing Filter + 2D Filter + α 補償、Gaussian あたりの周波数を最も密なトレーニングカメラ サンプリングレートの Nyquist 制限に制限します。理論的な目標: トレーニング外 スケール (0.5x または 2x のトレーニング解像度) でのレンダリングでのエイリアシングの除去。Preprocess と Backward Projection シェーダーで有効化、Q1.5-D テストで機能的に正しいと検証されました。しかし: オリジナル受け入れゲート ($\Delta@1x \geq +0.3$ dB かつ $\text{avg}(\Delta@0.5x, \Delta@2x) \geq +1.5$ dB) は 11 ペアシーンのどれでも達成されませんでした。観測された最大値: family 750K classic $\Delta@2x = +0.270$ dB。屋外シーン (Truck、Flowers) は 1x と 0.5x で悪化さえ示しました。仮説: 3D Smoothing は high-Gs で MCMC Relocation と競合します。ワールドは正しい Mip-NeRF-360 方法論を持つ将来の Multi-Scale Re-Eval のために残ります (Benchmark パスの O3 バックログ参照)。

 簡単に言うと

2024 年の論文からのエイリアシングフィルター。理論的には素晴らしいが、実際には私たちのテストで何ももたらさず、時には害さえもたらしました。実験者向けに利用可能なままですが、お勧めしません。オフのままにしてください。

T75 mipSmoothing3DScale 詳細

デフォルト: 0.2 (Paper デフォルト) **範囲:** 0.05 – 1.0 **定義場所:**

 技術詳細

3D Smoothing スケールパラメータ (Yu et al. §3.3、Paper デフォルト 0.2)。大きい = Gaussian あたりの世界空間スムージングが多い (= より多くのアンチエイリアシング、しかしデフォルトスケールでより多くのブラー)、小さい = より鮮明だがエイリアシングに影響を受けやすい。T74 `useMipSplatting = true` の場合のみ参照されます。Q1.5 テストでさらに最適化されませんでした — A/B ゲートはすでに Paper デフォルト 0.2 で失敗、さらなるスイープは無駄でしょう。

 簡単に言うと

。Mip をオンにしていない場合、無関係。

T76 mipFilter2DVariance 詳細

デフォルト: 0.3 (= 正確に V242 レガシー動作) **範囲:** 0.1 – 1.0 **定義場所:**

 技術詳細

Σ_{2D} 対角に追加される 2D Mip Filter 分散 (分散直接、平方ではない)。0.3 は Mip-Splatting 前にカーネルにハードコードされていた正確な V242 レガシー値です。T74 useMipSplatting = false の場合、カーネルはこの値を完全に無視し、ハードコードされた 0.3 を書きます — ベースラインが回帰できないように (Codex Round 1 S3-1 保証)。の場合、ここで設定された値が使用されます。Mip スイープ用に フィールドカタログに残ります。

 簡単に言うと

別の Mip-Splatting パラメータ。
Mip オフ: 無関係。

適応 Densification (Q5) (T77~T79)**T77 adaptiveDensification** 詳細

デフォルト: false **範囲:** boolean **定義場所:**

 技術詳細

Q5 機能: 固定 T11 densifyGradThreshold への代替としての rolling-median トラッカー。true の場合、各 Densify ステップで現在の閾値が $\text{median}(\text{last } N \text{ avgGrad samples}) \times T79 \text{ adaptiveDensifyMultiplier}$ で上書きされます。N = T78 adaptiveWindow。V440 p98 (壊滅的な 63 K Pruning トラップ) より厳密、median + 2× は定常状態の勾配分布の約 p70~p80 に座ります。Q5 テスト: 単独で 3 シーンの 0 FAIL、しかし Q6 (T80/T81 参照) と一緒に 3 シーンの 1 PASS — Q5+Q6 バンドルは 2026-05-25 に opt-in としてパスし、CLI `--adaptive-densify` 経由で有効化可能です。Q6 はそこで品質向上の「キャリア」で、Q5 はむしろ安定性に貢献します。

 簡単に言うと

自己学習 Densify 閾値。固定の感度の代わりに、アプリがシーンに適応します。単独でテストしてもより良くありませんが、Q6 の Curriculum と一緒なら良い。両方一緒にオンにするか、両方オフにします。

T78 adaptiveWindow 詳細

デフォルト: 1 000 **範囲:** 100 – 10 000 **定義場所:**

 技術詳細

Densification イベント内の Rolling Median Window (反復ではない — 各 T13 densifyInterval ステップが 1 サンプルを提供します)。デフォルト 1 000 — これは、過去 100 000 のトレーニング反復が中央値に寄与することを意味し、つまり 通常ここまでのトレーニング履歴全体。早期フェーズ (T78 サンプル前): トラッカーは nil を返します → 固定閾値 T11 にフォールバック。アクティブな場合のみ関連。

 簡単に言うと

いくつかの古い Densify ステップが T77 用の中央値に流れ込むか。標準 1000 が良い。Q5 Adaptive オンの場合のみ関連。

T79 adaptiveDensifyMultiplier 詳細

デフォルト: 2.0 **範囲:** 1.0 – 4.0 **定義場所:**

 技術詳細

適応閾値用の Rolling Median の Multiplier。デフォルト 2.0 はおおよそ 典型的な勾配分布の p70~p80 に相当します。低い = より積極的な成長 (より多くのクローン)、高い = より厳密 (より少ないクローン)。Q5 テスト範囲 1.5~3.0 — 2.0 最高デフォルト。アクティブな場合のみ関連。

 簡単に言うと

T77/T78 用の係数。標準 2.0 = 典型的な中央値より厳密。調整しないでください。

Curriculum (Q6) (T80~T81)

T80 curriculumResolutionRamp

詳細

デフォルト: false **範囲:** boolean **定義場所:**

技術詳細

Q6 機能: トレーニング解像度は 0.5x で開始し、T50 positionLRScheduleEndIteration / 2 (または T50 が設定されていない場合は T1 maxIterations / 2) で T22 trainingRenderScale に切り替わります。Q1.5.1 で開発された `resize/restoreImageBuffers` インフラストラクチャを使用します。有効化された場合、T23 resolutionWarmupScale を上書きします。Q6 は Q5+Q6 バンドル (T77 参照) で「品質向上のキャリア」としてパス — 段階的な解像度増加は、アプリに低い解像度で粗いジオメトリを見つける時間を与え、その後細かい詳細作業に移行します。CLI 経由: `--curriculum-resolution`。

簡単に言うと

トレーニング解像度用の「最初に粗く、次に細かく」。最初の半分で半分の解像度、その後フル解像度。特定のシーンでは役立ち、他では役立たない — T81 と一緒にオンにすることをお勧めします。

T81 curriculumSHProgression

詳細

デフォルト: false **範囲:** boolean **定義場所:**

技術詳細

Q6 機能: T21 shDegreeUpgradeIterations を `[maxIter/4, maxIter/2, maxIter*3/4]` で上書き、SH アップグレードを front-load する代わりにトレーニング時間にわたって均等に分散します。仮説: 安定したジオメトリは カラーディテール爆発の前に確立され、View-Direction 依存の光沢効果がより正確に配置されます。Q5+Q6 一緒に 3 シーンの 1 PASS、Q6 が利益のキャリア (Q5 単独 FAIL)。CLI 経由: `--curriculum-sh`。

簡単に言うと

「最初に形、次に色」 — 光沢効果はトレーニング後半でのみリリースされるため、Splat は最初に位置とサイズを見つけます。T80 と一緒にオン可能。単独ではそれほどもたらしません。

静的プリセット (TP1~TP9)

ここでは Initializer デフォルトとの構造的な違いのみ。11 個の UI プリセット P1~P11 の完全なマーケティング説明は第 7 章にあります。

TP1 .preview

☰ 詳細

≥ 10 GB RAM のシステム用の診断/プレビュープリセット。Initializer に対する オーバーライド: - 30 000 → 5 000 - 15 000 → 3 500 (maxIter の 70%) - 1.6e-6 → 1.6e-5 (10× 高い、あまり積極的でない Decay) -、、、それぞれ 2× (V176) - 3 000 → 100 000 (事実上オフ、V172: Reset が短いトレーニングを破壊) - [1K, 2K, 3K] → [1K, 2K] (V182: Degree 3 が 2K Iter で収束しない) - 1.0 → 0.5

💬 簡単に言うと

新しくインポートされた画像シーケンスの初期評価 — 2~3 分の待ち時間、その後結果はバイナリ質問「Quality 実行は価値があるか?」に十分です。

TP2 .full

☰ 詳細

Production-Quality Classic。オーバーライド: - 30 000 → 35 000 (V550: 40K テスト Truck 過剰トレーニング Gs で +10.7%、L1 で -1.3%) - 15 000 → 5 000 (V310 スイートスポット、V338 7K 悪い) - すべての LR 2× (V188) - 1.6e-6 → 1.6e-5 (V45 10×) - 2e-6 → 1.1e-6 (V335) - 100 → 200 (V112) - 0.005 → 0.001 (V393) - 3 000 → 100 000 (V194 無効化、V421 確認) - [1K, 2K, 3K] → [2K, 5K, 8K] (V228 遅延) - 0.0 → 0.9995 (V546 HTGS、14% 改善) - 50 (変更なし、V546) - false → true (V438) - 0 → 20 000 (V431) - true (V443、すでに .full の Initializer デフォルト)

💬 簡単に言うと

< 500 画像での各標準写真撮影 (オブジェクト、小さな部屋、彫刻)。V546 で発表された V438 に対する 14% Loss 改善は、Horse Full で 3 試行平均 確認されました。

TP3 .fullClassicPaper

☰ 詳細

TP2 の Q1.5-A テスト兄弟、paper 忠実な Classic。TP2 に対するオーバーライド: - 35 000 → 30 000 (Paper 標準) - 5 000 → 15 000 (Paper: maxIter の 50%) - 1.6e-5 → 1.6e-6 (Paper デフォルト) -、、Paper デフォルトに戻す (0.05、0.005、0.001) - 1.1e-6 → 2e-7 (Bicycle で ~1-2M Gs 用にキャリブレート) - 200 → 100 (Paper) - 0.001 → 0.005 (Paper デフォルト) - 100 000 → 3 000 (Paper §5.2、危険 — V194 リグレーションをトリガーする可能性) - 0.9995 → 0.0 (Paper には Decay なし) - 20 000 → 30 000 (cosine は maxIter の 100% で実行)

💬 簡単に言うと

Q1.5 研究実験、Mip-Splatting テスト用に Paper マグニチュード Gaussian パッケージ (1~2 M) を必要とします。Q1.5 「closed no-win」評決後、プリセットは上級ユーザーがアクセス可能なままですが、Production 推奨ではありません。

TP4 `.fullMCMC`
 詳細

Production-Quality MCMC。Initializer に対するオーバーライド: - 30 000 → 200 000 (V534、MCMC は Classic より 5× 多くの Iter が必要) - 15 000 → 160 000 (V504b maxIter の 80%) - 1.6e-6 → 1.6e-5 - LR スケジュール TP2 と同じ (すべて 2×) - 0.2 → 0.05 (V521b/V534: MCMC はより強い L1 信号が必要) - [1K, 2K, 3K] → [2K, 5K, 8K] - `.classic` → `.mcmc` - 150 000 (Initializer ですでに、プリセットで確認) - 5e-5 (V467/V536 最適) - 0.005 → 0.01 (V535 最適) - 0 → 160 000 (maxIter の 80%、V497c/V502) - 3.0 (Initializer ですでに) - true (Initializer ですでに) - 3 000 → 200 000 (事実上オフ、MCMC は Reset の代わりに Reloc を使用)

 簡単に言うと

Web 配信、ディテール要求のあるオブジェクトキャプチャ、ドローン飛行 (P9 Outdoor がさらに良い場合でも)。同等の L1 で Classic より 71% 少ない Gaussian。

TP5 `.fullMCMCMip`
 詳細

TP4 の Q1.5-D テスト兄弟、Mip-Splatting + Paper マグニチュード MCMC バジレット付き。TP4 に対するオーバーライド: - mcmcMaxGaussians 150 000 → 1 500 000 (10×、Paper マグニチュード) - useMipSplatting false → true (Mip オン)

 簡単に言うと

他のすべてのフィールドは TP4 と同じ。2026-05-24 Bicycle で Q1.5 D-PASS (12-Iter Multi-Scale FAIL ストリークを破る)。Q1.5 最終判決 2026-05-25 でも closed-no-win — Mip-Splatting 利益は 11 ペアシーンにわたって再現できません。プリセットは opt-in のままです。

TP6 `.classicBalanced`
 詳細

Mid-Tier Classic。TP2 に対するオーバーライド: - 35 000 → 20 000 (V149: 20K = 30K で 33% 少ない時間) - 20 000 → 0 (Cosine が maxIter = 20K で実行、拡張フェーズなし)

 簡単に言うと

短い待ち時間の標準ケース。V149 をスイートスポットとして識別。

TP7 .mcmcPreview

詳細

MCMC 診断。TP4 に対するオーバーライド: - 200 000 → 60 000 (V494b) - 160 000 → 48 000 (80%) - 150 000 → 100 000 (V473b) - 160 000 → 40 000 (V494b) - 3.0 → 2.0 (1.4.5: Preview = より軽いスケリング)

簡単に言うと

TP4 または Scene-Class プリセットが価値があるかを判断するために、素早く MCMC 結果を見ます。

TP8 .mcmcBalanced

詳細

Mid-Tier MCMC。TP4 に対するオーバーライド: - 200 000 → 120 000 (V518) - 160 000 → 96 000 (80%) - 160 000 → 96 000 (80%) - 3.0 → 2.5 (Preview 2.0 と Full 3.0 の間)

簡単に言うと

完全な 200K 実行なしの MCMC。~120 K 反復は品質と待ち時間の良い妥協です。

TP9 .quickTest

詳細

純粋な機能テスト。Initializer に対するオーバーライド: - 30 000 → 1 000 - 15 000 → 500 - 2e-6 → 4e-6 (0.25x 解像度用にキャリブレート) - 100 → 50 - 3 000 → 100 000 (オフ、はるかに短すぎるため) - 1.0 → 0.25

簡単に言うと

サニティチェック「トレーニングはまったく意味を持って開始するか?」。M3 Ultra で < 30 秒の期間。間違いなくぼやけて見えます。

メソッド:

シグネチャ: `public func resolveMcmcMaxGaussians(initialPointCount: Int, bufferCapacity: Int) -> Int` **定義場所:**

技術詳細 「MCMC が最大どれだけの Gaussian を成長させられるか?」の質問への唯一の Source-of-Truth。3 つの入力から計算: 構成された T62 `mcmcMaxGaussians` (0 の場合 Mass-Extinction Floor 150 000 付き)、(SfM 初期点の数)、(事前割り当てされた Gaussian バッファサイズ)。ロジック:

1. `base = T62 > 0 ? T62: 150_000` (Mass-Extinction Floor は 1.4.3 Mass-Extinction 事故などの Initializer デフォルトバグを防ぎます)
2. `T73 mcmcAutoScaleByScene && initialPointCount > 0 && T72 mcmcCapMultiplier > 0` の場合: `scaled = max(base, ceil(initialPointCount × T72))` そうでなければ
3. `bufferCapacity > 0` の場合: `return min(scaled, bufferCapacity)`
4. そうでなければ `return scaled`

例: Bicycle (Mip-NeRF 360、194 写真フレーム) → SfM-init ~156 K 点、`T62 = 150 000`、`T72 = 5.32`、バッファ容量 8 M。Resolved Cap = $\min(8M, \max(150K, \text{ceil}(156K \times 5.32))) = \min(8M, 830K) = 830 K$ 。これが MCMC Relocation ロジックが従う実効成長 Cap です。

簡単に言うと MCMC での実際の最大 Splat 数を計算します。設定を取り、シーンが最初に必要なだけの点を持つかを見て、自動調整がオンの場合は `Multiplier` でスケールします。これにより、小さなシーンと巨大なシーンに同じ値を強制する代わりに、Cap がシーンに適応します。メソッドを自分で呼び出す必要はありません — トレーニングが内部で使用します。

どのフィールドが何のためか? (チートシート)

| 目的 | 調整するフィールド |
|--------------------------|---|
| 遠くでより多くのディテール | <code>T62 mcmcMaxGaussians</code> 高、 <code>T72 mcmcCapMultiplier</code> 5+ |
| 一般的により多くのディテール (Classic) | <code>T1 maxIterations</code> 高 ($\leq 40K$)、 <code>T2 densifyUntilIteration</code> <code>T1</code> の $\leq 14\%$ |
| ドローン飛行で Floater を削減 | <code>T43 frustumCullEnabled</code> オン、 <code>T20 skyMaskingEnabled</code> オン、 <code>T45 skyDomeEnabled</code> オン |
| 屋外シーンの美しい空 | <code>T45 skyDomeEnabled</code> オン、 <code>T47 skyDomeRadiusMultiplier</code> 30~60 |
| 小さいエクスポートファイル | 戦略 <code>.mcmc</code> (<code>T61</code>)、 <code>T56 postTrainingCompactification</code> オン、 <code>T62 mcmcMaxGaussians</code> $\leq 200K$ |
| より速いトレーニング | <code>T22 trainingRenderScale</code> 0.5、 <code>T1 maxIterations</code> 半分 — ただし両方ではない! |
| より良いハイライト | <code>T21 shDegreeUpgradeIterations</code> を <code>[2K, 5K, 8K]</code> (early-front-load なし)、MCMC + 200K iter |
| Mac を応答可能に保つ | <code>T25 throttleDelayMs</code> 5~10 (~15% のトレーニング時間をコスト) |
| より頻繁なライブプレビュー | <code>T59 livePreviewInterval</code> 10~20 に下げる |
| シャドウでより柔らかい遷移 | <code>T17 ssimWeight</code> 少し高く (0.15~0.25)、ただし 0.3 を超えない |
| 室内をコンパクトに保つ | <code>P10 Indoor</code> プリセット (、 <code>T72 = 1.76</code>) |

危険なフィールド

これらのフィールドは、誤って構成された場合、OOM、アプリクラッシュ、Gaussian の Mass-Extinction、または使用できないベンチマークデータにつながる可能性があります。慎重に扱ってください:

- T11 `densifyGradThreshold` — 半分にすると 2~4x の Gaussian を生成し、GPU メモリを急速に圧迫する可能性があります。また注意: T22 `trainingRenderScale` に一致する必要があります (1.0x → 1e-6、0.5x → 2e-6、0.25x → 4e-6)。
- T72 `mcmcCapMultiplier` — > 200 K の SfM-init 点と `Multiplier > 5` を持つ大きなシーンでは、数百万の Gaussian の Resolved Cap が形成されます。36 GB RAM Mac では OOM の可能性。Outdoor プリセット 5.32 は Mip-NeRF-360-Bicycle が 156 K init 点を持つためのみ機能 → 830 K Cap。
- T39 `testViewIndices` — 手動設定はベンチマークを使用不可にする可能性があります (すべてのインデックス > N → Holdout なし)。 `--benchmark` フラグに設定させてください。
- T64 `mcmcOpacityRegWeight` **および** T65 `mcmcScaleRegWeight` — 1.4.3 ベータでは 0.01 に設定され、Mass-Extinction (1 反復で 460 K → 5 Gaussian) につながりました。1.4.4 以降 0.0 に固定されていますが、手動で増加させると問題を再現できます。
- T15 `opacityResetInterval` — 100 000+ (事実上オフ) でなく、トレーニングが 10 000 反復より短い場合、Reset が収束を破壊します。 `.preview` は `maxIterations = 5 000` にもかかわらず、そのため 100 000 を持ちます。
- T54/T55 `densifyPhase2*` — Two-Phase Densification はテストで 0 Gaussian カスケードで中止されました。両方を 0 のままにします。
- T74 `useMipSplatting` — Q1.5 closed-no-win 2026-05-25、一部の屋外 シーンで PSNR を悪化させることさえあります。デフォルトオフ、研究のみの opt-in。

このリストにあるフィールドを変更したい場合は、事前に現在のプリセットのバックアップを作成 (JSON としてエクスポート) し、結果を再現可能に測定できるかを検討してください — そうしないと、後で改善または悪化を取得したかどうかわかりません。

章

第 7 章 — 内蔵品質プリセット

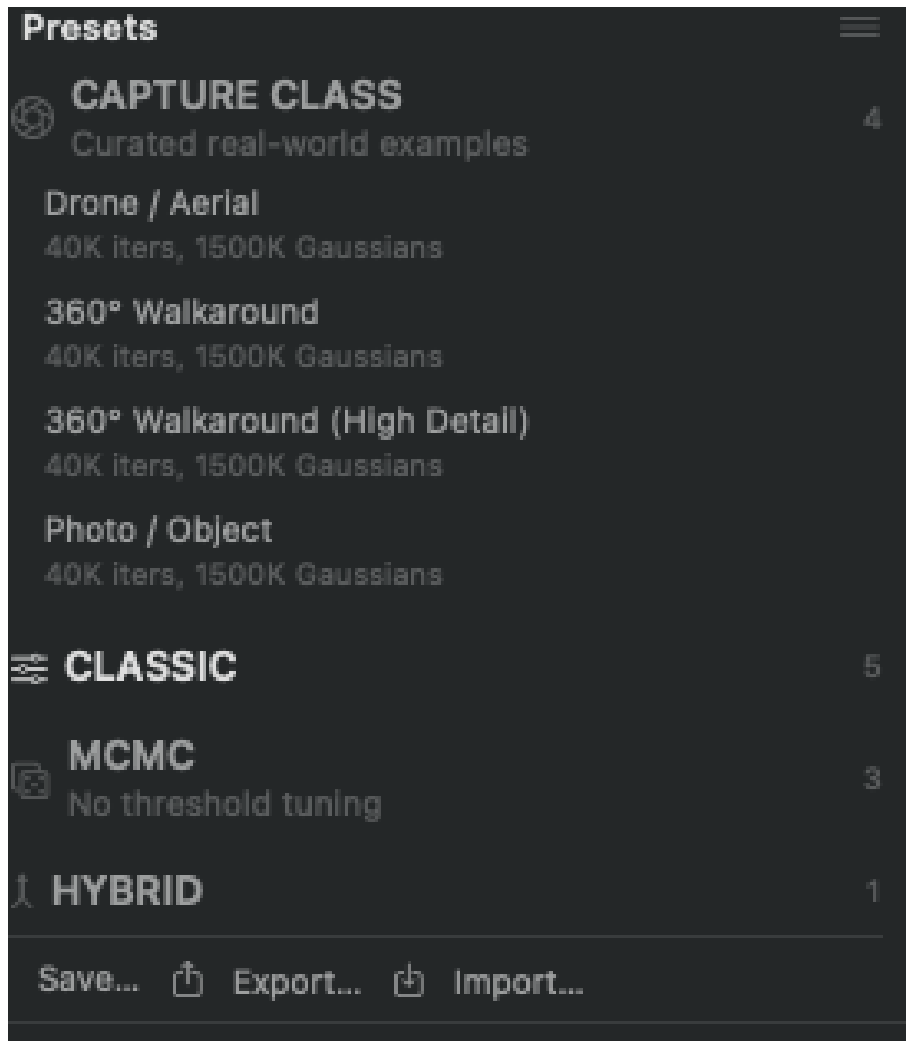


図 27: 4 つのグループがすべて展開されたプリセットセクション — CAPTURE CLASS (4 プリセット: Drone/Aerial、360° Walkaround、360° Walkaround (High Detail)、Photo/Object)、CLASSIC (5 プリセット: Quick/Preview/Balanced/Quality/Ultra Detail)、MCMC (3 プリセット、注記「閾値チューニング不要」)、HYBRID (1 プリセット: Balanced (Hybrid))

画像で見えるもの Inspector のプリセット セクションで、4 つのグループがすべて展開されています。CAPTURE CLASS には 4 つのキュレーションされた実世界プリセット (Drone / Aerial、360° Walkaround、360° Walkaround (High Detail)、Photo / Object) が並びます — これがプライマリ グループで、Simple モードで唯一表示されるものです。CLASSIC には Quick (1K 反復)、Preview (5K 反復、青いチェックマークでアクティブ選択)、Balanced (20K 反復)、Quality (35K 反復)、Ultra Detail (35K 反復) が並びます。MCMC には

副題「閾値チューニング不要」が付きます — MCMC では Densify-Until 閾値は不要です: Preview (60K 反復、100K ガウシアン)、Balanced (120K、150K)、Quality (200K、150K)。HYBRID には Balanced (Hybrid) (20K 反復、150K ガウシアン) があります。フッターのアクション行: Save...、Export...、Import...

プリセットとは、トレーニング用に準備された設定のことです。RadianceKit には 4 つのグループに分かれた 13 個の内蔵プリセットが付属しています — 4 つの **Capture-Class** プリセット (P9~P12) — 実際のコミュニティ素材で目視検証された、実世界の撮影タイプ (ドローン、360° ウォークアラウンド、写真オブジェクト) 向けのキュレーションされたレシピで、v1.6 以降のプライマリ軸です —、5 つの Classic プリセット (P1~P5: Quick/Preview/Balanced/Quality/Ultra Detail)、3 つの MCMC プリセット (P6~P8)、そして Classic 戦略と MCMC 戦略を組み合わせた 1 つのハイブリッドプリセット (P13) です。以前の「Scene-Class」プリセット (Render/3D、Outdoor、Indoor。Phase Q7 で Mip-NeRF-360 シーンと NeRF-Blender シーンに対して学術的にチューニング) は、v1.6 で可視グループとして撤回されました — 実素材で目視検証された Capture-Class が現在のプライマリ軸であり、Q7 チューニング済み構成は内部にのみ保持されます。これらはサイドバーの **プリセット** 領域、または Simple モードのインポート時に選択できます。+ ボタンで独自プリセットを並べて作成するダイアログが開きます — 13 個の内蔵プリセットは削除できませんが、複製は可能です。

エキスパートビューでは、プリセットは撮影タイプと戦略 (Capture Class / Classic / MCMC / Hybrid) ごとにグループ化されて表示されます。項目をクリックすると、保存されているトレーニング構成が現在の状態に書き込まれます。これはスナップショットではないので、その後スライダーを動かすと状態は変化しますが、プリセット自体は変更されません。カラフルなヒントが「modified」と表示されます。

どのプリセットがいつ正しいかは、主にシーンタイプとハードウェアに依存します。章末の 3 つの表形式の概要にまとめてあります。

| P1 — Quick



場所

Inspector → プリセットセクション → 「Classic」グループ → 「Quick」項目。UUID 接尾辞 ...001。



技術詳細

1 000 反復、クラシック (適応的) Densification 戦略、トレーニング 解像度スケーリング 0.25× (入力画像はトレーニング前に 25 % に縮小) を使用する 診断用プリセット。シーンを納品するためのものではなく、セットアップ (カメラポーズ、点群、画像シーケンス) が Loss 値に意味のある動きをもたらすかどうかを素早く確認するためのものです。M3 Ultra では 50~200 枚の画像で通常 30 秒未満。低解像度では実際の画像品質はわかりませんが、メモリ消費とレンダリング負荷は非常に低く保たれます。システムの RAM が 10 GB 未満の場合、初回起動時に自動的にデフォルトとして選択されます。

簡単に言うと

素早い動作テスト。画像を投入し、30 秒弱待ち、シーンの大まかな輪郭が表示されるかを確認します。ビューワーの画像がぼやけた塊のように見えたならそれで正常です。それに対して、暗い点だけが見える、または完全に歪んだ形が見える場合、おそらくカメラポーズが間違っています (第 9 章参照)。見せられる結果を得るには、その後最低でも P2 か P3 が必要です。

| P2 — Preview (Classic)



場所

Inspector → プリセットセクション → 「Classic」グループ → 「Preview」項目。UUID 接尾辞 ...002。



技術詳細

5 000 反復の Classic Densification、0.5× 解像度スケーリング、標準に対して 2 倍の学習率。Densification (クローン + 分割) は最初の 2 500 反復でアクティブで、その後は Pruning のみです。RAM ≥ 10 GB のシステム用デフォルト プリセット。M3 Ultra では 200 枚の画像シーンで通常 90 秒~3 分。形状とカメラ ポーズの使える印象を提供しますが、テクスチャは目に見えてぼやけます — 0.5× レンダリング解像度は、後で P3 や P4 で再トレーニングしても直接回避できません。なぜなら学習率が半解像度に合わせて調整されているからです。

簡単に言うと

「ちょっと見るだけ」の標準。新しい画像をインポートしてシーンがそもそも再構築可能かを見たい場合、これが正しいレベルです。約 2~3 分待ち、その後 3D ビューワーで回転させて、追加のトレーニング実行を投資する価値があるかを判断できます。プレビュー結果がすでに良く見えてから初めて、Balanced や Quality を行う価値があります。

| P3 — Balanced (Classic)



場所

Inspector → プリセットセクション → 「Classic」グループ → 「Balanced」項目。UUID 接尾辞 ...005。



技術詳細

フル画像解像度で 20 000 反復の Classic Densification。Densification は最初の 15 000 反復で実行され、Iter 3 000 から Densify 間隔 100 で実行されます。記録されたトレーニングセッションでの経験的「スイートスポット」: Horse Full と Truck でのクラシック Densification では、L1 Loss は Iter 18 000 ~ 22 000 で安定し、より長いトレーニングは Quality (P4) 未満で大きな改善をもたらしません。M3 Ultra では 200 枚の画像で通常 30 ~ 60 秒、1 000+ 枚の画像で 5 ~ 8 分。

簡単に言うと

「良い妥協点」。ほとんどのシーンはこれですでに良く見え、1 時間待つ必要はありません。最終結果をどこかに見せたい場合 (ソーシャルメディア、ウェブサイト、顧客デモ) には、しばしばこれで十分です。Splat モデルにズームインしたい、または表面テクスチャの詳細が必要な場合に初めて、P4 Quality または P7 MCMC へのジャンプが価値を持ちます。

| P4 — Quality (Classic)



場所

Inspector → プリセットセクション → 「Classic」グループ → 「Quality」項目。UUID 接尾辞 ...003。



技術詳細

V546 「Opacity Decay」(HTGS、Eurographics 2025) を使用した 35 000 反復の Classic Densification: 各 Densify サイクル後、すべての既存 Gaussian の Opacity に < 1.0 の係数が掛けられ、これにより非アクティブになった Gaussian が Pruning で確実に削除され、同じ反復数で従来の 35 000 反復より 14 % 良い L1 Loss を達成します。SSIM Loss が有効化されています (`ssimWeight=0.05`)。M3 Ultra では 200 枚の画像で通常 2 ~ 4 分。NeRF-Blender (Lego、Chair、Drums) で最終 $L1 \approx 0.023$ を提供 — 560+ の記録済み実験で 最高の Classic バリエーション。注意: 約 3 ~ 5 GB の GPU メモリが必要。8GB システムでは P3 が安全な選択です。

簡単に言うと

最高のクラシックバリエーション。鋭いテクスチャと細かい形状を提供します。特にオブジェクト撮影 (彫刻、椅子、花瓶) に向いています。一方、大きな屋外シーンや部屋では、Balanced との違いはほとんど感じられません — その場合は MCMC プリセット (P6 ~ P8) または Capture-Class プリセット (P9 ~ P12) への切り替えが、P3 から P4 へのジャンプより価値があります。Classic ファミリーの絶対的な最大値が欲しい場合は、P5 Ultra Detail を選びます。

| P5 — Ultra Detail (Classic)



場所

Inspector → プリセットセクション → 「Classic」グループ → 「Ultra Detail」項目。UUID 接尾辞 ...008。



技術詳細

約 35 000 反復の Classic Densification — 品質マトリクスの Held-out 実行 (2026-06-10) の勝者。テストした 3 つの Mip-NeRF-360 シーンすべてで、Ultra Detail は同等のウォールクロック時間で内蔵 MCMC 「Quality」プリセット (P8) を約 +0.94 dB PSNR 上回ります。これにより、Classic グループで最も強力な Quality プリセットであり、RadianceKit が提供する最もシャープな Classic バリエーションとなっています。M3 Ultra では P4 Quality と同じ時間枠 (200 枚の画像で 2~5 分) ですが、やや多くの GPU メモリが必要です。8GB システムでは P3 が安全な選択のままです。

簡単に言うと

最もシャープな Classic レベルであり、品質テストの Held-out 勝者です: 実シーンで MCMC 「Quality」バリエーションより約 1 デシベル良く — 同程度の待ち時間で。実績あるクラシック Densification で最大の細部忠実度が欲しく、十分な GPU メモリがあるなら、これが第一選択です。メモリが足りない、またはできるだけ小さいエクスポートファイルが必要な場合は、P4 Quality または MCMC プリセットに留まってください。

| P6 — Preview (MCMC)



場所

Inspector → プリセットセクション → 「MCMC」グループ → 「Preview」項目。UUID 接尾辞 ...006。



技術詳細

60 000 反復の MCMC Densification (3DGS-MCMC、NeurIPS 2024)、Gaussian 上限 100 000。MCMC はヒューリスティックなクローン/分割ロジックをマルコフ連鎖モンテカルロ再配置に置き換えます: 死んだ Gaussian は Sigmoid 重み付き サンプリング深度を介して再配置され、制御された再現可能な数の Gaussian が得られます。上限は最大数を 100K で強くキャップします — これによりメモリとレンダリング時間が節約されますが、細部が失われます。M3 Ultra では 200 枚の画像で通常 5~8 分。「MCMC 機能テスト」として適しています — Classic から MCMC への切り替えが意味があるかを判断するのに役立ちます。P7 や P8 に時間を投資する前に。

簡単に言うと

P2 Preview と同じですが、新しい MCMC 手法を使用しています。Classic バリエーションよりもしばしばよりコンパクトで均等に分布した Splat 雲を提供します。シーンの最初の評価には、5~8 分でほとんどの場合十分です。レビュー結果が気に入ったら、次のステップは P7 (Balanced) または直接 P8 (Quality MCMC) です。

■ P7 — Balanced (MCMC)

📍 場所

Inspector → プリセットセクション → 「MCMC」グループ → 「Balanced」項目。UUID 接尾辞 ...007。

🔧 技術詳細

120 000 反復の MCMC、Gaussian 上限 150 000。中間 MCMC レベル — P8 Quality とほぼ同じ最終 Gaussian 数ですが、反復回数は 60 % のみ。記録されたトレーニングセッションでは、Horse Full での L1 Loss は経験的に 0.026~0.028 で、P8 の 0.0246 と比較して約 7 % 高く、その代わり待ち時間は半分です。M3 Ultra では 200 枚の画像で通常 8~15 分。実効的な Gaussian 上限を入力 SfM 点群の点密度にスケールアップする手法を使用します (第 6 章 T75 参照)。

💬 簡単に言うと

P8 の長い完全実行なしで、適度な細部の MCMC。ほとんどのシーンではこれで十分です。特に MCMC 実行を昼休みの時間枠に収めたい場合に。メモリが不足している場合 (例えば 16 GB のみの M プロセッサ) は、ここに留まってください — P8 はより多くの GPU メモリが必要です。

■ P8 — Quality (MCMC)

📍 場所

Inspector → プリセットセクション → 「MCMC」グループ → 「Quality」項目。UUID 接尾辞 ...004。

🔧 技術詳細

200 000 反復の MCMC、Gaussian 上限 150 000、SSIM Loss 0.05、反復の 80 % にわたる MCMC ノイズ減衰。560+ 実験での最良単独実行 L1: Horse Full で 0.0238、3 試行平均で 0.0246 (同じシーンで P4 0.0230 に対して)。MCMC は 71 % 少ない Gaussian (150K 対 約 524K) を提供します — これは、結果をウェブで配信したい場合に決定的です。なぜなら、より小さな雲は大幅に小さいエクスポートファイルを生成するからです。M3 Ultra でのトレーニング時間は 200 枚の画像で通常 20~35 分。1 000+ 枚の画像セットでは 1~2 時間程度。最小の最終サイズで最大の画像品質が必要な場合の最良の選択。

💬 簡単に言うと

最高の MCMC パリアント。非常にクリーンでコンパクトな Splat 雲を提供します — 結果を後で Web 3D ビューワーに埋め込んだり、ファイルとして送信したい場合に理想的です (ファイルは同等の視覚品質で P4 Quality より小さい)。しかし、忍耐が必要です — 大きな撮影では 1 時間以上の待ち時間。「夜通しの実行」として計画してください。

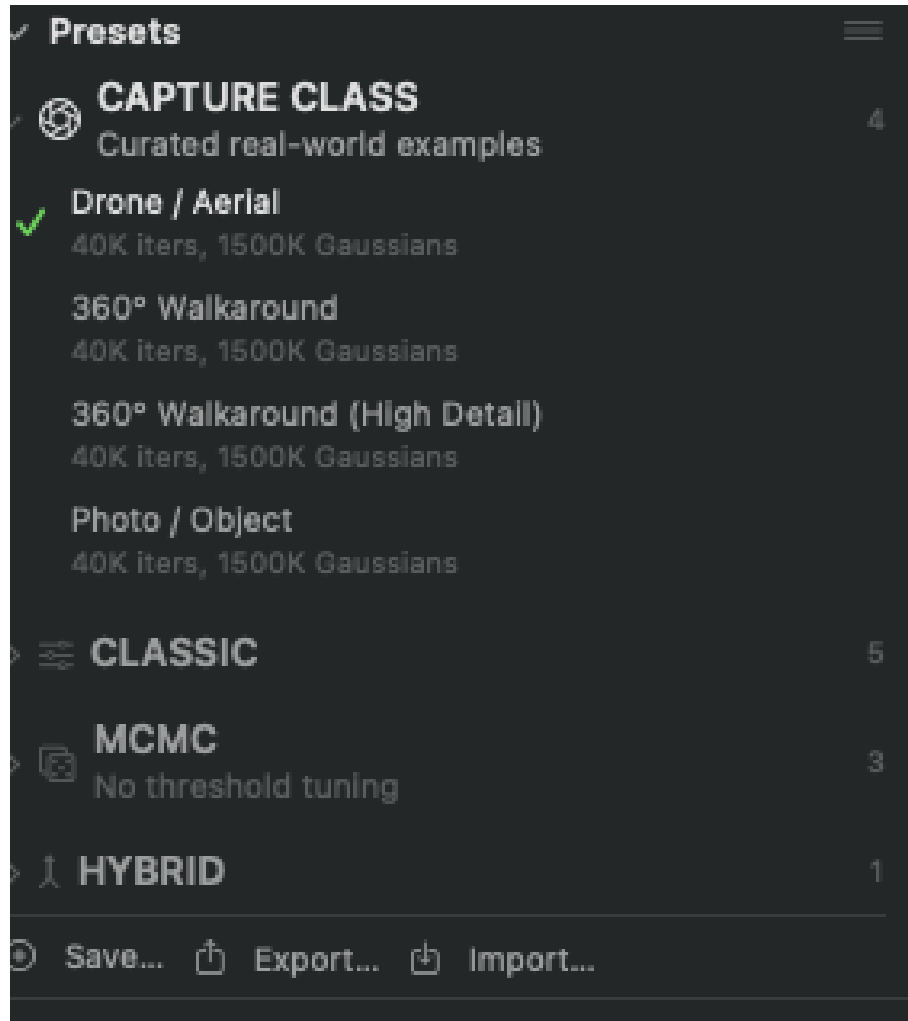


図 28: CAPTURE CLASS グループが展開され、4 つのキュレーションされた実世界 プリセットすべて — Drone / Aerial (MCMC、40K 反復、上限 1.5 M)、360° Walkaround (MCMC、40K、上限 1.5 M)、360° Walkaround (High Detail) (Hybrid、40K、上限 1.5 M、オプトイン)、Photo / Object (Hybrid、40K、上限 1.5 M)。このグループが一番上にあり、Simple モードで唯一表示されるものです。

画像で見えるもの CAPTURE CLASS グループが展開された Inspector — v1.6 以降のプライマリプリセットグループで、Simple モードで唯一表示されるものです。各項目は、特定の撮影タイプ (ドローン、360° ウォークアラウンド、写真オブジェクト) 向けに実際のコミュニティ素材で目視 検証されたレシピであり、学術的なテストセットに対して最適化された値ではありません。クリックで選択すると、保存されているトレーニング構成が現在の状態に書き込まれます。

| P9 — Drone / Aerial

場所

Inspector → プリセットセクション → 「Capture Class」グループ → 「Drone / Aerial」項目。UUID 接尾辞 ...010。

技術詳細

建物や風景の空撮・ドローンオービット用の Capture-Class プリセット。MCMC Densifier、40 000 反復、上限 150 万 Gaussian、SSIM Loss 0.5 に加えて Edge-Aware 項 0.1。決定的なのは、比率閾値 6 で重み 0.003 の異方性ペナルティです — ドローン素材が生成する典型的な針状アーティファクトに対する「スパゲッティキラー」です。Pensford 高架橋上の実際の DJI 4K ドローン飛行で検証済み (計量的だけでなく目視でも確認)。

簡単に言うと

空からの撮影用 — 建物の周りのドローン飛行、風景上、ファサードに沿って。強力な異方性ペナルティが、ドローン素材がよく生成する針状・スパゲッティ状のアーティファクトを取り除きます。素材が地上から撮影されている場合は、Photo / Object または Classic プリセットの方が適しています。

| P10 — 360° Walkaround

場所

Inspector → プリセットセクション → 「Capture Class」グループ → 「360° Walkaround」項目。UUID 接尾辞 ...011。

技術詳細

360° ウォークアラウンド動画用の Capture-Class プリセット。MCMC Densifier、40 000 反復、上限 150 万 Gaussian、SSIM Loss 0.5 に加えて Edge-Aware 項 0.1、穏やかな異方性ペナルティ (比率閾値 15 で重み 0.001)。人物マスクと空マスクが有効です。このプリセットは 360° の equirect 動画を想定しており、トレーニング開始前に内部で約 90° 幅の透視クロップに再投影されます。セルフィースティックでの 8K 360° ウォークアラウンド (Monument シーン、目視で確認) で検証済み。

簡単に言うと

360° ウォークアラウンド動画用 — 360° カメラやセルフィースティックを持って部屋の中やオブジェクトの周りを歩きます。RadianceKit は球面パノラマを自身で通常の視点に分解し、歩行者や空をマスクで除去します。同じ素材で最大のシャープさを得るには、追加で High-Detail パリアント (P11) をお試しください。

| P11 — 360° Walkaround (High Detail)



場所

Inspector → プリセットセクション → 「Capture Class」グループ → 「360° Walkaround (High Detail)」項目。UUID 接尾辞 ...013 (オプトイン)。



技術詳細

最大の細部を持つ 360° ウォークアラウンド動画用のオプトイン Capture-Class プリセット。Hybrid Densifier (クラシックな Abs 勾配の クローン/分割 + MCMC ノイズ + 再配置)、40 000 反復、上限 150 万 Gaussian、比率閾値 15 で異方性ペナルティ 0.0015、SSIM Loss 0.2、Edge-Aware 項 0 — ロックされた「r50」スクリーン分割レシピ。360° 素材では、標準 MCMC プリセット 「360° Walkaround」(P10) を PSNR、LPIPS、目視のコンフェッティで上回り、しかも Splat 数は約 3 分の 1 です。より多くのシーンで検証されるまで、標準 360 プリセットの隣に意図的にオプトインとして配置されています。

簡単に言うと

標準 360 プリセット (P10) のよりシャープな代替: より多くの細部、より少ないコンフェッティ、大幅に小さいファイル。置き換えるのではなく意図的に並べて配置されています — これまでに数えるほどのシーンで確認されました。360° ウォークアラウンドがきれいに撮影されているなら、まずこのプリセットを試して、結果を P10 と比較してください。

| P12 — Photo / Object



場所

Inspector → プリセットセクション → 「Capture Class」グループ → 「Photo / Object」項目。UUID 接尾辞 ...012。



技術詳細

シャープな単独写真 (動画ではない) からのオブジェクトオービット用の Capture-Class プリセット。Hybrid-t1 Densifier (再配置あり)、40 000 反復、上限 150 万 Gaussian、SSIM Loss 0.5 に加えて Edge-Aware 項 0.1、穏やかな 異方性ペナルティ (比率閾値 15 で重み 0.001)、50 反復ごとの Opacity Decay 0.9995、マスキング なし。骨格の 163 枚の高解像度 41MP 写真 (目視で確認) で検証済み。少数のビュー (約 600 まで) は Hybrid 崩壊閾値を下回ったままです。

簡単に言うと

シャープな単独写真からのオブジェクト撮影用 — 彫刻、模型、製品をカメラで一周し、動画ではなく写真を撮ります。シャープな写真は通常きれいな背景を持つため、マスキングなし。動画ソースには代わりに 360° または Drone プリセットを使ってください。

I P13 — バランス (ハイブリッド)

場所

Inspector → プリセットセクション → 「ハイブリッド」グループ → 「バランス (ハイブリッド)」項目。
UUID 接尾辞 ...009。

技術詳細

20 000 反復のハイブリッド Densification 戦略: 古典的な勾配駆動のクローン/分割が Loss の必要とする場所に容量を配置し、MCMC SGLD ノイズが探索を続け、死んだ Gaussian はプルーニングで失われる代わりに再配置されます。Opacity Decay (V546) が Opacity リセットを置き換え、異方性ペナルティ (重み 0.001、比率閾値 15) が針状の Splat を抑制します。Gaussian 上限はシーンに応じてスケールリングされます (ベース 150K、シーン認識 $\times 3.0$)。同じ予算の純粋な MCMC に対して 5 つのシーンで検証済み: 平均で +0.45 dB PSNR、Gaussian は 20~30 % 少なくなります (stonehenge +1.23、family +0.82、garden +0.47 dB)。M3 Ultra では 200 枚の画像で通常 5~10 分。

簡単に言うと

最終結果のための強力な第一選択: MCMC プリセットよりシャープな細部を同程度にコンパクトなファイルで、しかも P8 のトレーニング時間のほんの一部で提供します。品質実行に使える時間が 1 回分しかなく、どの Capture クラスも明確に当てはまらない場合は、ここから始めてください。素早いテストには引き続き Classic プリセットが優れており、シーンがこれらの撮影タイプのいずれかに明確に一致する場合は Capture-Class プリセット (P9~P12) が第一選択です。

どのプリセットをいつ？

| シナリオ | 初期テスト | メイン実行 |
|---------------------------|-----------------|---|
| 新規画像の動作テスト、< 30 秒 | P1 Quick | — |
| シャープな単独写真からのオブジェクトオービット | P2 Preview | P12 Photo / Object |
| 単独オブジェクトスキャン (動画)、< 500 枚 | P2 Preview | P4 Quality または P8 Quality MCMC |
| 360° ウォークアラウンド動画 | P6 Preview MCMC | P10 360° Walkaround (シャープ: P11 High Detail) |
| 空撮/ドローンオービット、風景 | P6 Preview MCMC | P9 Drone / Aerial |
| Web 配信 (小型、コンパクト) | P2 | P8 Quality MCMC (フル品質で最小ファイル) |
| 短時間でシャープな細部、コンパクトなエクスポート | P2 または P6 | P13 バランス (ハイブリッド) |
| 最大の細部忠実度、Classic 戦略 | P3 または P6 | P5 Ultra Detail |
| 印刷、マーケティング、フルディテール | P3 または P6 | P4 Quality (Classic) または P5 Ultra Detail |

クイック比較

| プリ セッ ト | 戦略 | 反 復 | 最大 Gs | レンダ リング Scale | 典型 時間 (200 枚、M3 Ultra) | Q-Sweep |
|--|---------|------------|----------|---------------------|------------------------|------------------------|
| P1 Quick | Classic | 1 000 | ∞ | 0.25x | ~30 秒 | — |
| P2 Preview | Classic | 5 000 | ∞ | 0.5x | 2–3 分 | — |
| P3 Balanced | Classic | 20 000 | ∞ | 1.0x | 30–60 秒 | — |
| P4 Quality | Classic | 35 000 | ∞ | 1.0x | 2–4 分 | V546 HTGS |
| P5 Ultra Detail | Classic | ~35 000 | ∞ | 1.0x | 2–5 分 | Matrix Δ+0.94 dB |
| P6 Preview MCMC | MCMC | 60 000 | 100 K | 1.0x | 5–8 分 | — |
| P7 Balanced MCMC | MCMC | 120 000 | 150 K | 1.0x | 8–15 分 | — |
| P8 Quality MCMC | MCMC | 200 000 | 150 K | 1.0x | 20–35 分 | V544a |
| P9 Drone / Aerial | MCMC | 40 000 | 1.5 M | 1.0x | 10–25 分 | 目 視 / Viadukt |
| P10 360° Walkaround | MCMC | 40 000 | 1.5 M | 1.0x | 10–25 分 | 目 視 / Monument |
| P11 360° Walkaround (High Detail) | Hybrid | 40 000 | 1.5 M | 1.0x | 10–25 分 | 目視 (オブ トイ ン) |
| P12 Photo / Object | Hybrid | 40 000 | 1.5 M | 1.0x | 10–25 分 | 目 視 / 骨格 |
| P13 パラ ンス (ハ イブ リッ ド) | Hybrid | 20 000 | 150 K | 1.0x | 5–10 分 | Matrix Δ+0.45 dB |

独自プリセット

プリセットセクション (第2章の I1) の **Save...** ボタンを介して、現在のトレーニング構成を独自プリセットとして保存します。独自プリセットは「Built-in」ではなく、名前変更、エクスポート (JSON として)、ドラッグ&ドロップで共有、複製、削除が可能です。13個の内蔵プリセット P1~P13 は削除ボタンの影響を受けません。

経験則: プリセットで変更したい要素を将来も使いたい場合 — Sky-Dome オン、特定のシーンクラス用の高い SSIM 重み、異なる反復数 — そのバリエーションを独自プリセットとして保存してください。そうすれば、次回の実行ですぐに、標準と異なる構成であることがわかります。

章

第8章 — エクスポート形式

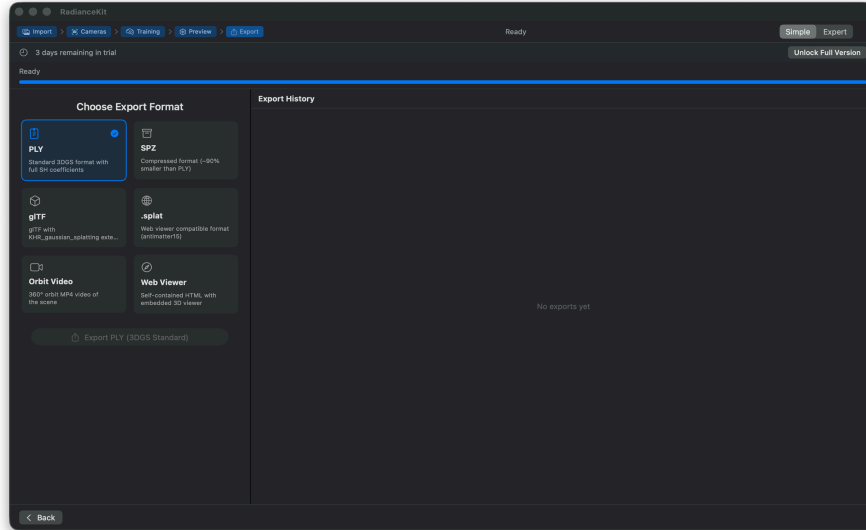


図 29: Simple モードでのエクスポート形式選択 — 6 つの形式カード

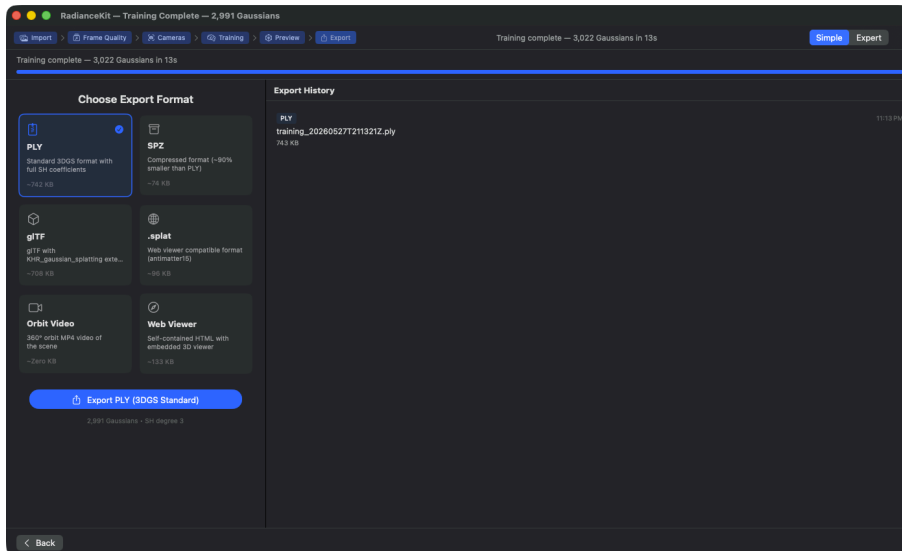


図 30: flowers ブーケで 5K 反復トレーニング後のライブエクスポート形式グリッド — 6 つすべてのカードに動的サイズ計算 (PLY 742 KB 選択、SPZ 74 KB、glTF 708 KB、.splat 96 KB、Orbit Video ~Zero KB、Web Viewer 133 KB)、右側のエクスポート履歴にすでに保存された PLY

画像で見えるもの (2 991 Gaussian、SH degree 3、Bjoern の合成 Blender ブーケを IP クリーンテストセットとして): 各形式カードの下のサイズ表示は、現在の Gaussian カウン

トと形式オーバーヘッドからライブで計算されます — ハードコードではありません。

2 991 Gaussian (SH degree 3) から、742 KB PLY、74 KB SPZ (量子化により約 10 倍小さい)、708 KB glTF (KHR_gaussian_splatting 拡張により、ほぼ PLY 相当)、96 KB .splat (圧縮された Gaussian あたり 24 バイト形式) が生成されます。Orbit Video は、サイズが MP4 エンコーディング後にのみ判明するため「~Zero KB」と表示されます。Web Viewer (133 KB) は、組み込み WebGL ビューアーと圧縮された Splat データを含む独立した HTML ファイルをバンドルします — ビューアーオーバーヘッドのため、純粋な .splat より大きくなります。右側のエクスポート履歴には、すでに完了した PLY エクスポート (「training_20260527T211321Z.ply、743 KB、23:13」) と形式ピル、Finder で開くアクションがリストされます。

完了したトレーニングは Gaussian Cloud を生成します — シーンを再構築する数十万から数百万の 3D ガウス分布のコレクションです。RadianceKit は、この Cloud をディスクに書き込む 10 通りの方法を知っています。そのうち 6 つは純粋な 3D データ形式 (PLY、Compressed PLY、SPZ、SOG、glTF、.splat)、1 つは Cloud を完成した HTML ビューアーとともにバンドル (Web Viewer)、1 つは Orbit カメラ走行から MP4 ファイルをレンダリング (Orbit Video)、2 つは Gaussian コンテンツをエクスポートせず、他のトレーニングパイプラインで再利用するための SfM 結果 (カメラポーズと粗い点群) のみをエクスポートします (transforms.json + COLMAP Workspace)。

どの形式がいつ正しいかは、目的によって異なります。品質損失なしのフルデータアーカイブには PLY を使用します。自分のサイトの Web ビューアーには、通常 .splat または組み込み Web Viewer で十分です。ファイルが最小である必要がある場合は、SPZ または SOG が価値があります。Nerfstudio、Postshot、または Brush で SfM 結果を再利用するには、transforms.json と COLMAP Workspace が正しい方法です。

すべてのエクスポート機能は「Export」メニューと、Simple モードの最後のウィザードステップにあります。ほとんどの形式は完全に Sandbox 準拠で、App Store バージョンで動作します。SOG のみ、App Store ビルドには必ずしも存在しない外部バイナリ (cwebp) が必要です — 詳細は E4 を参照してください。

E1 — PLY (.ply)

📍 場所

メニューバー → Export → 3D Formats → Export PLY... (⌘E)。Simple モード: ウィザードステップ Export → 形式カード「PLY」。**サイズ:** 通常 100 % (基準値)。**互換性:** SuperSplat、PolyCam、すべての 3DGS ビューアー。

🔧 技術詳細

PLY は 3D Gaussian Splatting の標準保存形式です。RadianceKit は標準化された 3DGS プロパティレイアウトを持つバイナリリトルエンディアン ファイルを書き込みます: Gaussian あたり 3 成分の位置、常にゼロに設定される 3 つの法線、基本 RGB 色用の 3 つの DC SH 係数 ($f_{dc_0..2}$)、続いて Kerbl 2023 論文で定義された転置チャンネルメジャー配置で最大 45 個の追加 SH 係数 ($f_{rest_0..44}$) (まず R チャンネル係数すべて、次に G、次に B)、続いて logit opacity (生の pre-sigmoid 値)、3 つの対数空間スケール、wxyz クォータニオン回転。最大エクスポート SH 次数は、ユーザー希望と実際に学習された次数の最小値にクランプされます。デフォルトは 3 (45 個の Rest 係数)。書き込み前に、非常に大きな Cloud でのオーバーフローを検出するため、ペイロードサイズが 64 ビット 整数で計算されます。ファイルはアトミックに書き込まれるため、大きな Cloud ではディスク容量を一時的に倍占有します。

💬 簡単に言うと

これは「オリジナルファイル」です。最大ファイル、最高互換性、損失なし。どの形式を選ぶべきか分からない場合は、PLY を選んでください— ほぼすべての 3DGS ツールで開けます。100 万 Gaussian の場合、SH 次数に応じて 200 から 800 MB になります。ファイルが大きすぎる場合は、E2 (圧縮 PLY) または E3 (SPZ) をご覧ください。

I E2 — Compressed PLY (.ply)

📍 場所

メニューバー → Export → 3D Formats → Export Compressed PLY...。Simple モード: 形式カード「Compressed PLY」。**サイズ:** PLY に対して約 10~20 % (5~10 倍の圧縮)。**互換性:** SuperSplat、PlayCanvas エンジン、ウェブベースのビューアー。

🔧 技術詳細

チャンクごとの量子化を持つ PLY 形式の PlayCanvas バリエーション。Gaussian は 256 個のチャンクにグループ化されます。チャンクごとに、位置、スケール、色の Min/Max 境界がヘッダーに個別に保存されます。個々の Gaussian はこれらの境界に対する値を参照し、それぞれ 32 ビットに圧縮されます: 位置とスケールは 11-10-11 ビットパッキング、回転は 2-10-10-10 ビット「Smallest-Three」クォータニオン、色は 8-8-8-8 RGBA。より高い SH 係数は、コンポーネントあたり 8 ビットのみで量子化されます (Gaussian あたり `shCoeffCount * 3 uchar`)。形式自体は依然として ASCII ヘッダー PLY であるため、PLY ツールで基本的に検証可能ですが、頂点プロパティは `uint` フィールドとして宣言されます。SH 次数は圧縮を最大化するためデフォルトで 0 (Rest 係数なし) です — より高い SH 次数を明示的に選択できます。

💬 簡単に言うと

スペース節約 PLY バリエーション。通常の PLY と同じエンジン互換性ですが、5~10 倍小さいです。SuperSplat と PlayCanvas はネイティブに読み込みます。Web デプロイには、通常の PLY よりほぼ常に優れています。量子化による品質損失は、シーンが極端に高周波の詳細を含まない限り、通常視覚的に認識できません。

E3 — SPZ (.spz)

📍 場所

メニューバー → Export → 3D Formats → Export SPZ...。Simple モード: 形式カード「SPZ」。**サイズ:** PLY に対して約 10 % (90 % 小さい)。**互換性:** Niantic Scaniverse、Niantic Spatial Fields、MetalSplatter。

🔧 技術詳細

Niantic の SPZ v2 形式。位置は 24 ビット固定小数点としてパックされます (約 0.25 mm の解像度)、スケールは対数空間で 8 ビット量子化、回転は 8 ビット Smallest-Three (v2 では xyz のみが保存され、w は Decoder でクォータニオン ノルムから導出される)、Opacity はシグモイド化された 8 ビット値。DC SH は SPZ 固有のパック式 ($dc_raw * 0.15 * 255 + 0.5 * 255$) で保存され、より高い SH バンドは係数あたり 5 ビット (Band 1) または 4 ビット (Band 2-3) で保存されます。全パックされたバイナリプロブは、標準 gzip (RFC 1952) で圧縮され、Magic Bytes `1f 8b` を持つ gzipped コンテナ形式が生成されます。Apple の組み込み zlib API が独自の Apple フレーミングを生成し、Spatial Fields や MetalSplatter の SPZ リーダーと互換性がないため、RadianceKit はシステム `gzip` を呼び出します。システム `gzip` は macOS Sandbox 内でも spawn 可能です。

💬 簡単に言うと

最小の標準ファイル。Niantic の Scaniverse を知っている場合 — これはそのアプリが使用する形式です。非常に小さく、モバイルアプリにとってロードフレンドリーです。Niantic 独自のクラウドビューアー (Spatial Fields) で直接使用可能です。同じデータの PLY より約 90 % 小さく、ほとんどのシーンで視覚的にほとんど区別できません。

E4 — SOG (.sog)

📍 場所

メニューバー → Export → 3D Formats → Export SOG...。Simple モード: 形式カード「SOG」。サイズ: PLY に対して約 5~6 % (15~20 倍圧縮 — 最小のオプション)。互換性: PlayCanvas エンジン、SuperSplat エディター。

🔧 技術詳細

「Spatially Ordered Gaussians」 — Cloud を複数の Lossless WebP 画像で GPU ready に保存する PlayCanvas 形式です。まず、すべての Gaussian が 3D Morton コード (30 ビット Z-Order、軸あたり 10 ビット) で空間的にソートされ、画像にレンダラーでの後のキャッシュ局所性が与えられます。次に、位置は対称対数変換 (より良いダイナミックレンジのため) で 16 ビット値に量子化され、2 つの RGBA 画像に分割されます (`means_l.webp` は下位 8 ビット、 `means_u.webp` は上位)。回転は、3×8 ビット + 2 ビットモードを持つ Smallest-Three として RGBA 画像にエンコードされます (モードはアルファに `252 + largest` として配置されます)。スケールと DC SH は、それぞれ 256 エントリのコードブック (パーセンタイルベースですべての値にわたって分散) で量子化され、インデックスは `scales.webp` と `sh0.webp` に配置されます。5 つの画像とコードブックと境界を含む `meta.json` は、ZIP ファイル (Sandbox がシステム `zip` をブロックするためカスタムエンコーダー) にパックされ、拡張子 `.sog` で保存されます。

Sandbox 注意: SOG は、外部バイナリを必要とする唯一の形式オプションです。WebP エンコーダステージは、`/usr/local/bin/cwebp` または `/opt/homebrew/bin/cwebp` から `cwebp` を呼び出します。cwebp バイナリが見つからない場合、コードは生の PNG エンコーディングにフォールバックします — しかし: **PNG フォールバックは SuperSplat で動作しません**。App Store バージョンでは、ビルドバリエーションに基づいて可用性を評価します。Developer バリエーションでは、cwebp を Homebrew 経由でインストールする必要があります (`brew install webp`)。

💬 簡単に言うと

SPZ よりかなり小さい、最小の 3DGS 形式。しかし: RadianceKit 自体がすべての画像形式を生成できないため、Mac 上に `cwebp` ツールが必要です。Homebrew で一度インストールすれば、すべて動作します。App Store バージョンでは完全に機能しない可能性があります — エクスポート時に WebP の代わりに PNG が出力される場合、ファイルを SuperSplat で直接開けません。Homebrew なしで作業する場合は、代わりに SPZ (E3) を使用してください。

E5 — glTF (.glb)

📍 場所

メニューバー → Export → 3D Formats → Export glTF...。Simple モード: 形式カード「glTF」。**サイズ:** PLY と同等。**互換性:** KHR_gaussian_splatting 拡張 (Khronos ドラフト標準) を持つ glTF ビューアー。

🔧 技術詳細

KHR_gaussian_splatting 拡張仕様に従って、自己完結型の .glb バイナリ ファイル (別の bin ファイル添付なし) を書き込みます。位置は通常の glTF POSITION 頂点データ (float3) として保存され、他のすべての属性 (回転は float4、スケールは float3、Opacity は float、SH 係数は float3 × shCoeffCount) は追加の 頂点属性に配置され、拡張経由で参照されます。重要: glTF は右手座標系 Y-up を使用し、COLMAP/3DGS は Y-down/Z-forward で動作します。そのため、エクスポーターは X 軸周りの 180 度回転を適用します — 位置は $(x, -y, -z)$ で書き換えられ、クォータニオンは $(w, x, -y, -z)$ に調整されます。これにより、glTF ビューアーで幾何学的に正しく、右手 (鏡像ではない) 表現が得られます。GLB 標準で要求される通り、JSON とバイナリチャンクは 4 バイトアラインメントにパディングされます。

💬 簡単に言うと

Khronos の 3D データ用公式標準形式で、Gaussian Splat 用の新しい拡張付きです。利点: glTF はすべての主要な 3D エンジン (Babylon.js, Three.js, Unity, Unreal) に普及しています。欠点: 2026 年時点で拡張はまだドラフト段階で、多くのビューアーがまだ対応していません。特に既存の glTF パイプラインに Splat データを統合する、またはすでに glTF 対応のビューアーを書く場合に有用です。

E6 — Splat (.splat)

📍 場所

メニューバー → Export → 3D Formats → Export .splat...。Simple モード: 形式カード「.splat」。 **サイズ:** Gaussian あたり正確に 32 バイト。 **互換性:** gsplat.js、ウェブベースのビューアー (antimatter15 リファレンス)、ほとんどのブラウザ 3DGS デモ。

🔧 技術詳細

antimatter15 `.splat` 形式 — Gaussian あたり 32 バイト、ヘッダーなし、間接化なし。エントリごとのレイアウト: 3 × float32 位置 (世界座標)、3 × float32 スケール (内部バッファの対数空間から exp 変換)、4 × uint8 RGBA 色 (DC SH 係数を `SH_C0 = 0.282...` でスケールし、[0,255] にクランプ)、4 × uint8 クォータニオン (`w,x,y,z`、正規化して `128 + 128*q` でバイト範囲にエンコード)。DC SH のみが保存されます — より高い SH バンドは破棄されます。これにより形式は非常にコンパクトになりますが、反射や鏡面ハイライトで発生するビュー依存の色変化が失われます。書き込み順序は Cloud のインデックス順序と正確に同じ (空間ソートなし) で、`gsplat.js` のような Web ビューアーはそれに基づいてレンダリングします。

💬 簡単に言うと

`gsplat.js` を使用して自分の Web ビューアーで Splat を表示したい場合の最適な形式。非常にコンパクト (Gaussian あたり 32 バイト) ですが、より高い SH 次数はありません — したがって、視点に応じた光沢のある反射や微妙な色の変化はありません。ほとんどの Web アプリケーションでは問題ありません。なぜなら、DC 色で十分であり、ビュー依存の欠如はほとんど目立たないからです。

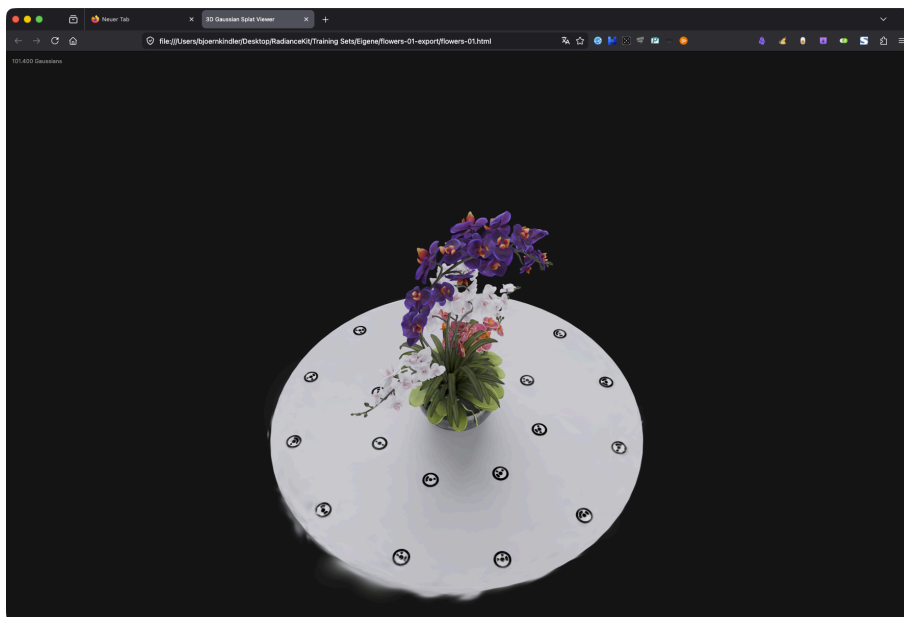


図 31: Firefox で開かれた Web Viewer — Bjoern のブーケ Splat が周囲のカメラマーカー球とともにレンダリングされ、上部にブラウザのタブバーが表示され、CDN/サーバーセットアップは不要。スタンドアロンの `flowers-01.html` が Finder からデフォルトブラウザでダブルクリックして直接開かれました — 組み込みの WebGL2 プログラムは、ネットワークやサーバーなしですぐに Gaussian Cloud をレンダリングします。ブーケの周りの黒いマーカーはトレーニングカメラで、オプションで表示できます。マウスドラッグで回転、スクロールでズーム。

E7 — Web Viewer (.html)

📍 場所

メニューバー → Export → Media → Export Web Viewer…。 Simple モード: 形式カード「Web Viewer」。 **サイズ:** base64 エンコードされた Splat データ (≈ 4/3 オーバーヘッド) + 約 5 KB HTML/JS シェル。 **互換性:** WebGL2 対応のすべての最新ブラウザ (すべてのデスクトップ、iOS 15+、Android 5+)。

🔧 技術詳細

Gaussian Cloud を完全にインラインで書かれた WebGL2 レンダラーと一緒に、単一の `.html` ファイルにバンドルします。CDN 依存性、WASM、2 番目のファイルはありません。Cloud は内部的に最初に `.splat` バイナリ (E6 と同じ 32 バイト ロジック) としてエンコードされ、次に base64 で埋め込まれ、ブラウザで `atob` でデコードされます。組み込みレンダラーは独自の WebGL2 ソート、マウスオービット制御、フレームごとの CPU ソートを実行します。すべての JS コード (シェーダー、数学、ループ) は出力 HTML で表示されます。ストレージとレンダラーの境界での軸規約は E5 と完全に同じです: 位置 $(x, -y, -z)$ 、クォータニオン $(w, x, -y, -z)$ 。オプションで、ブランディングオーバーレイを表示できます (Free Tier スイッチ)。すべてがインラインなので、ファイルは `file://` プロトコルから直接動作します — テストにローカル Web サーバーは不要です。

💬 簡単に言うと

誰かにメールで送ったり、ウェブサイト埋め込んだりできる単一の HTML ファイルです。Finder でダブルクリックすると、ブラウザがマウス回転でシーンを表示します。クラウドへのアップロード不要、2 番目のファイル不要、サーバー不要。顧客プレゼンテーション、ポートフォリオ、メール添付に最適です。欠点: base64 エンコーディングによりファイルは純粋な `.splat` より約 1/3 大きくなります — そのため、非常に大きなシーンでは、`.splat` ファイルを標準ビューアーと一緒に別々にホスティングする方が価値があります。

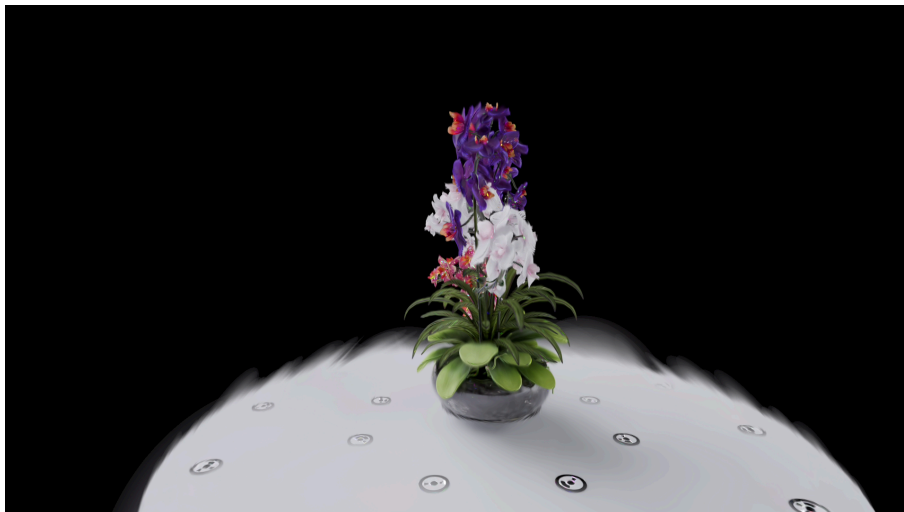


図 32: *flowers-01.mp4* から抽出された単一フレーム — Bjoern のブーケの プロファイルレンダリング、カメラマーカー付きの白いプラットフォームが見え、黒い背景 (デフォルトのビューポート背景、設定で変更可能)。カメラはパラメトリックパスでシーンを周回 (仰角 + 距離は固定、Yaw は回転)、通常 30 または 60 *fps* で 6~10 秒の長さ。フレーム解像度は *VideoPreset* 経由で 480p から 8K までスケーラブル。

E8 — Orbit Video (.mp4/.mov)

📍 場所

メニューバー → Viewport → Record Turntable Video
または メニューバー → Export → Media → Export
Orbit Video...。Simple モード: 形式カード「Orbit
Video」、Duration スライダー 3~30 秒。 **サイズ:** 長
さ、解像度、ビットレートに依存。 **互換性:** すべての
プラットフォーム (H.264 と HEVC は Apple 標準)。

🔧 技術詳細

パラメトリック Orbit カメラ走行に沿って Gaussian
Cloud をレンダリングし、 AVAssetWriter 経由で各フ
レームを MP4 または MOV ファイルにエンコードしま
す。 Orbit 構成は、回転数 (公転)、距離、仰角、FOV、
長さ、Ease-In/Out 係数を制御します。 Orbit ビデオ
エクスポートは RadianceKit 独自の ForwardPass を
完全な SH 評価とともに 通過します — アプリ内ビュー
ポートとピクセル単位で同一 (WYSIWYG) です。 フ
レームごとに、世界調整マトリクス (レンダラーで計算
され、内部座標を Y-up Orbit 世界に回転させる) が
カメラと乗算され、続いてカメラ規約の反転 (camFlip:
Orbit Y-up → COLMAP Y-down) が適用されます。 オ
フスクリーン レンダーターゲットは、IOSurface 経由
でエンコーダー用の CVPixelBuffer に引かれます。 エ
ンコーダーは H.264 と HEVC、構成可能なビットレ
ート、480p から 8K までの解像度をサポートします。
最初のフレームの前に、レンダラーは初期 Splat ソ
ートが完了するように 200 ms 待機します。 このエクス
ポートは GPU バウンドです — 8K と数百万 Gaussian
では、フレームあたりのレンダリング時間が数秒にな
り、6 秒のビデオで合計 10~30 分のレンダリング時
間になることもあります。

💬 簡単に言うと

シーンの回転を持つ完成した MP4
ファイル。ソーシャルメディア
、マーケティング、プレゼ
ンテーションに最適です。長さ
(3~30 秒)、回転方向、速度を 設定
できます。ファイルは YouTube,
Instagram, PowerPoint, その他の
場所に直接埋め込めます。アプ
リは各フレームを完全にレンダリ
ングする必要があるため、時々
遅くなります — 8K ビデオでは、
Gaussian の数に応じて 5~30 分
を予定してください。

| E9 — SfM Transforms (transforms.json)

📍 場所

メニューバー → Export → Photogrammetry → Export SfM (transforms.json)...。 **サイズ:** 通常 1~10 KB (ポーズと内部パラメータのみ、画像なし、Gaussian なし)。 **互換性:** nerfstudio、Brush、gsplat、OpenSplat、Meshroom、すべての最新フィードフォワード 3DGS トレーナー。

🔧 技術詳細

カメラポーズリストと共有内部パラメータを持つ nerfstudio `transforms.json` 形式を書き込みます。カメラごとに、ビューマトリクス (RadianceKit 内部: COLMAP 規約での World-to-Camera) が反転され、次にカメラ ローカル Y と Z 基底ベクトルが反転されて nerfstudio 規約 (OpenGL スタイル、カメラは `-Z` 方向を見る、`+Y` が上) に変換されます。最終 4×4 マトリクスは、各フレームの `transform_matrix` フィールドに `double` の行優先ネストされた配列として配置されます。内部パラメータはトップレベルに保存されます (焦点距離 `x/y`、主点 `x/y`、画像幅/高さ、`camera_model = "OPENCV"`、加えて歪み係数 `k1, k2, p1, p2`) — ただし、エクスポーターが複数の異なる内部パラメータセットを検出する場合は、フレームごとに書き込まれます。画像パスは JSON ファイルに対して相対的に `images/<filename>` として書き込まれます。ユーザーは、トレーニング写真を含む兄弟 `images/` フォルダを作成する必要があります。

💬 簡単に言うと

この JSON ファイルは、各写真について、カメラがどこにあって、どこを見ていたかを記述します。ファイル自体は小さく役に立たないものです — フォルダ内でオリジナル画像と一緒に使用されます。Nerfstudio, Brush, その他のトレーナーはまさにこの形式を読み込み、これを使って RadianceKit SfM 結果を別のツールに渡すことができ、そこでカメラ再構築を再計算する必要はありません。大きなシーンで時間を節約します。

E10 — COLMAP Workspace (sparse/0/)

📍 場所

メニューバー → Export → Photogrammetry → Export SfM (COLMAP Workspace)...。 **サイズ:** 3 つのバイナリファイル合計通常 4~8 MB — `points3D.bin` が支配的 (スパース Cloud の 3D 点ごとに 1 行)、`images.bin` と `cameras.bin` はそれぞれ 100 KB を大きく下回ります。 **互換性:** COLMAP 自体、Nerfstudio、Postshot、Meshroom、COLMAP sparse/ ディレクトリを期待するすべてのツール。

🔧 技術詳細

3 つのバイナリファイルを持つ標準 COLMAP sparse/0/ レイアウトを 書き込みます: `cameras.bin`、`images.bin`、`points3D.bin`。形式リファレンスは公式 COLMAP ドキュメントです。`cameras.bin` には、重複排除された内部パラメータ リストが含まれます (同じ内部パラメータと画像サイズを持つカメラは、単一の エントリにまとめられます)。使用されるカメラモデルは `OPENCV` (Model 4) で、`fx/fy/cx/cy` と 4 つの歪み係数 `k1/k2/p1/p2`。`images.bin` は、画像ごとに `wxyz` クォータニオンとしてのポーズ、続いて `translation`、カメラ ID、ファイル名を リストします。2D-3D 対応は保存されません。`points3D.bin` には、位置、色 (0-255 RGB)、再投影と Track 長のデフォルト値を持つ SfM 点群が含まれます。すべて リトルエンディアンで書き込まれます。RadianceKit への再インポートは File メニュー → 「Import COLMAP/Metashape Workspace...」 (SfM バックエンド章の Q3 参照) で機能します。

💬 簡単に言うと

公式 COLMAP 形式。Postshot, Nerfstudio, または他の COLMAP 対応 ソフトウェアでトレーニングを続行したい場合、これが方法です。3 つの小さなファイル プラスオリジナル画像、そしてターゲットプログラムは、COLMAP 自体がソース プログラムであったかのように受け入れられます。`transforms.json` 形式 (E9) より多くのプログラムがこれを理解しますが、テキストベースではなくバイナリのため、少し扱いにくいです。

どの形式をいつ？

| 目的 | 形式 |
|--|---|
| 自分のサイトの Web ビューアー | E7 Web Viewer (.html) |
| <code>gsplat.js</code> を使用した Web ビューアー | E6 Splat (.splat) |
| Postshot / Nerfstudio でのパイプライン再利用 | E9 transforms.json + E10 COLMAP Workspace |
| SuperSplat 編集 | E1 PLY または E2 Compressed PLY |
| Niantic Scaniverse / Spatial Fields | E3 SPZ |
| 最大圧縮 | E4 SOG (cwebp 必須) |
| マーケティング/ソーシャルビデオ | E8 Orbit Video |

クイック比較

| 形式 | 拡張子 | Sandbox | サイズ (1M Gauss) | 最適用途 |
|----------------------|-------------|-----------------|----------------|----------------------------|
| E1 PLY | .ply | はい | ~250 MB | アーカイブ、 最高 互換性 |
| E2 Compressed PLY | .ply | はい | ~40 MB | Web + SuperSplat |
| E3 SPZ | .spz | はい (gzip spawn) | ~40 MB | Niantic + モ バイル |
| E4 SOG | .sog | 条件付き (cwebp) | ~20 MB | 最大 圧縮 |
| E5 glTF | .glb | はい | ~250 MB | Khronos パ イプライン |
| E6 Splat | .splat | はい | ~32 MB | gsplat.js Web ビュー アー |
| E7 Web Viewer | .html | はい | ~45 MB | スタンドア ロン ブラウ ザファイル |
| E8 Orbit Video | .mp4 / .mov | はい | 可変 | ソーシャル/ マーケティ ング |
| E9 SfM Transforms | .json | はい | ~5 KB | ポーズ受け渡 し |
| E10 COLMAP Workspace | ディレクトリ | はい | ~4-8 MB | ポーズ受け渡 しバイナリ |

サイズ列は SH 次数 3 で 100 万 Gaussian の大まかな目安です。実際の値はシーンの 圧縮性によって異なります。SH 次数 0 では PLY/glTF が 4 倍縮小します。

章

第 9 章 — SfM バックエンド

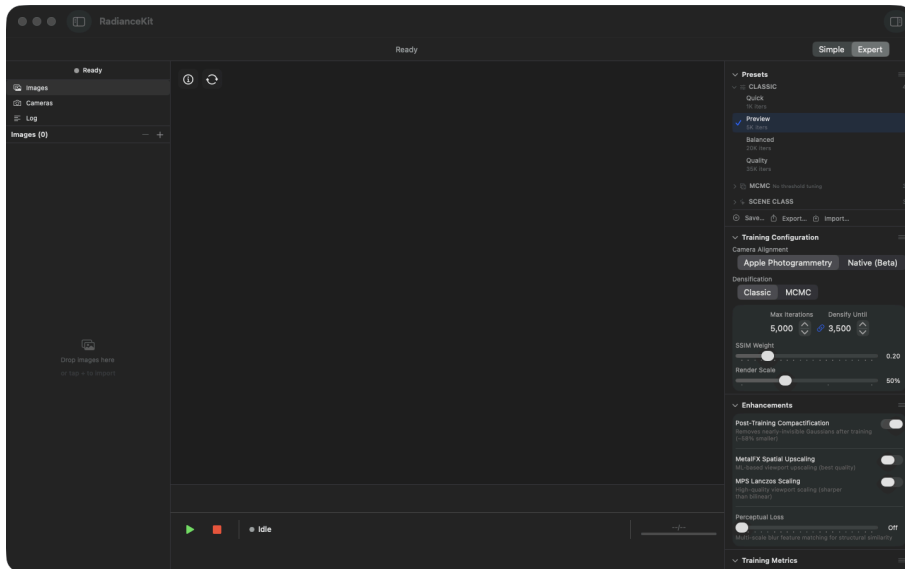


図 33: Inspector の Camera Alignment ピッカー付きエキスパートモード (Apple Photogrammetry / Native (Beta))

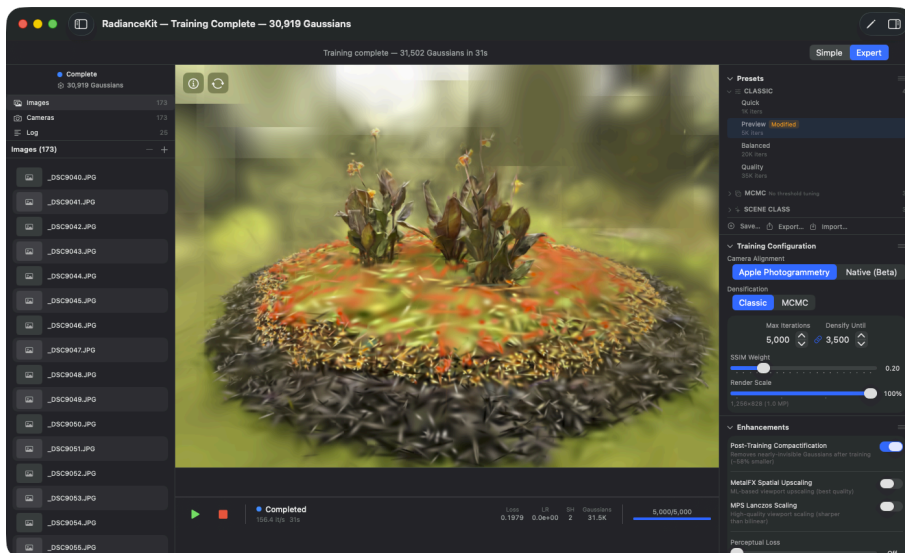


図 34: Native (Beta) がアクティブな Inspector — Camera Alignment ピッカーで 2 番目のオプションが選択され、他のトレーニング構成パラメータはすべて変更なし

画像で見えるもの Inspector の Camera Alignment ピッカーは、2 つのオプションを持つセグメンテッドコントロールです — Apple Photogrammetry (App Store ビルドのデフォルト、

完全に Sandbox 準拠) と Native (Beta) (RadianceKit 独自の FAST+BRIEF+GLOMAP パイプラインバックエンド、Phase 3.8/3.9 で開発、2026-05 時点)。Native (Beta) はオービット専用を検証され、 ≥ 1000 フレームで Apple Photogrammetry より高速ですが、Phase 3 §5 品質ゲート ($\text{finalLoss} \leq 0.0115$) はまだ満たしていません — そのため Beta タグが付いています。Metashape、COLMAP、または他のフォトグラメトリソフトウェアからの外部 SfM 結果は、追加で File メニュー (Q3 COLMAP テキスト形式、Q6 Workspace インポート) からインポートできます — ピッカーは切り替わりませんが、インポートされたポーズが SfM 結果を置き換えます。

SfM は **Structure from Motion** の略です。重複する写真群から、ソフトウェアは各画像について、共有の 3D 座標系内のカメラの位置と視線方向を再構築します。その過程で粗い 3D 点群が生成され、これが Gaussian Splatting のトレーニングを初期化します。SfM 結果は実際のトレーニングの入力であり、後の画像品質を 決定的に左右します。

RadianceKit は 5 つの SfM 経路を提供します: アプリに組み込まれた 2 つのバックエンド (Q1 Apple Photogrammetry、Q4/Q5 Native)、外部ツールからの 2 つのインポート経路 (Q3 COLMAP テキスト形式、Q6 バイナリ Workspace インポート)、および Q2 COLMAP バイナリ (App Store 外の Developer ビルドでのみ利用可能)。どれが正しいかは シーンタイプ (オブジェクトの周りのオービット、室内、ドローン飛行) と、外部ソフトウェアがすでに再構築を提供しているかどうかによって異なります。

Q1 — Apple Photogrammetry



場所

Expert View → Inspector → トレーニング構成
→ Camera Alignment ピッカー、項目「Apple Photogrammetry」。



技術詳細

もともと Object Capture 用に開発された Apple の組み込み フォトグラメトリフレームワークをラップします。Apple は内部的に独自パイプライン (手順は公開されていません) で特徴を抽出し、マルチビューマッピングで検証し、Apple Silicon Neural Engine + GPU 上でバンドル調整を解きます。バックエンドは完全に App Store 準拠 (外部バイナリなし、Sandbox=true、オンデバイス) ですが、カメラポーズと粗い点群のみを提供します — トラック長や再投影誤差などの診断メトリックはありません。Apple の推奨に従って数百枚の画像までスケールリングします。直線的なドローン飛行や大規模な屋外シーンで ~500 フレームを超えると、再現性のあるクラッシュや個々のカメラの静かな破棄が観察されています。

簡単に言うと

これが最もシンプルな方法です。画像を入れ、アプリが計算します。古典的なオブジェクトスキャンに非常に適しています — 家具や彫刻の周りを歩いて 50~200 枚の写真を撮る場合。風景上のドローン飛行や非常に多くの画像 (500 枚超) では、Apple の手法は不安定になりがちです。そのようなシーンでは Native バックエンド (Q4/Q5) をテストするか、カメラを Metashape で計算して Workspace インポート (Q6) 経由で読み込んでください。

パワーユーザー

Q2 COLMAP バイナリ — 外部 COLMAP プログラムをサブプロセスとして起動するため、App Store バージョン (Sandbox) では **利用できません**。App Store 外の Developer ビルドでのみ動作します。COLMAP が提供する品質を得るには、App Store バージョンでは Workspace インポート (Q3 または Q6) があります: SfM を外部で COLMAP や Metashape で計算し、結果を読み込んでください。

Q3 — COLMAP テキスト形式 (Metashape / ETH3D)

場所

メニュー 「File → Import COLMAP / Metashape Workspace...」 (Cmd+⇧+I) または `sparse/0/` `cameras.txt` を含むフォルダのドラッグ&ドロップ。

技術詳細

標準化された COLMAP テキストエクスポートを読み込みます — `sparse/0/` サブフォルダ内の 3 つのテキストファイル `cameras.txt`、`images.txt`、`points3D.txt` — そして内部の SfM 結果モデルに変換します。COLMAP バイナリ エクスポートと同じ形式定義で、バイナリの代わりに ASCII で記述されています。Agisoft Metashape、RealityCapture、PolyCam、ETH3D ベンチマークがまさにこのレイアウトで出力します。パーサーはバイナリパーサーとカメラモデル検出を共有します (すべての一般的なモデル: SIMPLE_PINHOLE、PINHOLE、OPENCV、OPENCV_FISHEYE、FULL_OPENCV)。コメント行と空行に対して堅牢。テストで ~1 400 カメラ (ETH3D Tunnel) まで問題なくスケーリングします。

簡単に言うと

すでに Metashape, RealityCapture, または他の商用写真 3D ソフトウェアで作業し、結果をエクスポートしている場合 — そのエクスポートをアプリ自体が再計算することなく RadianceKit に直接読み込みます。これで何時間もの待ち時間が節約できます。File メニュー経由でフォルダ全体を読み込むか、ウィンドウにドラッグしてください。

I Q4 — Native SfM (インクリメンタル)

場所

Expert View → Inspector → トレーニング構成 → Camera Alignment ピッカー、項目「Native (Beta)」。

インクリメンタルがこのバックエンドの デフォルトモード — Inspector には別のマッパーピッカーはありません。CLI 経由で モードを明示的に設定するには `--native-sfm` または `--sfm-mapper incremental` を使用します。

技術詳細

GPU 加速された SfM パイプライン全体の独自実装: FAST+BRIEF 特徴量、または CoreML 経由の SuperPoint+LightGlue (`--coreml-features` 付き)、続いて Hamming-KNN マッチング、RANSAC 基礎行列、トラック構築、初期ペア選択、2 視点ブートストラップ (F→E プラス DLT)、PnP 登録と マルチビュー三角測量を用いる greedy インクリメンタルマッパー、最後に Huber Loss と 解析的ヤコビアンを Cholesky Solve で用いる Schur 簡約 Levenberg-Marquardt によるバンドル調整。完全に App Store 準拠: 外部バイナリなし、Sandbox=true。Phase 3.10 で提供された R2 崩壊検出器付き: アプリが入力フレームの 60% 未満しか登録しない、またはカメラあたりの点数が 13 を下回ると、自動的にグローバルマッパー (Q5) にフォールバックします。経験的にオービット/ターンテーブルシーンでクリーン。より一般的な動き (ドローン飛行、複雑な形状の室内空間) では成功率は低くなります — ただし、検出器がこれらのケースをキャッチします。~200 カメラまで確実にスケールし、それ以上は実行時間が大幅に長くなります。

簡単に言うと

Apple の強み (App Store 互換、オービットで高速) に追加の診断値を加えたものです。Object Capture のように被写体の周りを歩く場合に特に適しています。より複雑な録画 (ドローン飛行やリビングルーム) では、RadianceKit は自動的にこれがうまくいかないことを検出し、グローバルプロセスに切り替えます。「Beta」とマークされているのはまだテスト中だからです — 標準推奨は依然として、シンプルなオブジェクトスキャンには Apple Photogrammetry、要求の厳しい屋外セットには Workspace インポート (Q3 または Q6) です。

Q5 — Native SfM (グローバル)

📍 場所

インクリメンタル Mapper (Q4) が崩壊検出器をトリガーした場合 (入力フレームの 60 % 未満が登録される、またはカメラあたりの点数が 13 を下回る)、自動的に呼び出されます。CLI `--sfm-mapper global` 経由でのみ手動で強制可能。Inspector では、グローバル手法は別のピッカーからアクセスできません — アプリが切り替えるタイミングを自分で決定します。

🔧 技術詳細

ネイティブパイプラインのグローバルバリエーション。まず Q4 と同じく 特徴抽出 + マッチング、続いてすべての検証済みペアの相対ポーズ推定、その後 回転平均化 (世界座標系内のすべてのカメラ回転を同期) と平行移動平均化 (大きなカメラ数での整数オーバーフローを避けるため、行列フリーのスパース 定式化に基づく LSQR)。原則として ~5 000 カメラまでスケールリングしますが、実際には数百カメラを超えると品質が低下します — K-1351 での Phase 3.8 §5 受け入れゲート測定では、目標の 0.0115 ではなく finalLoss 0.07 が得られました。「フォールバックティア」として扱われます: インクリメンタル Mapper が退化したときに登場しますが、それ自体は品質を再チェックされません。

💬 簡単に言うと

ネイティブエンジンのプラン B 経路。より高速なインクリメンタル経路が失敗した場合に自動的に呼び出されます。使える結果を提供しますが、非常に大きい、または難しいシーンでは、通常 Metashape や外部 COLMAP インストールから得られるものほど正確ではありません。Native が標準ワークフローになる場合、そのようなケースでは Workspace インポート (Q3 または Q6) 経由の迂回が価値があります。

Q6 — Metashape / COLMAP テキスト Workspace インポート (Phase Q7)

場所

File メニュー → 「Import COLMAP / Metashape Workspace...」 (Cmd+⇧+I)。sparse/0/cameras. {bin,txt} と images/ を含むフォルダのドラッグ&ドロップ。

技術詳細

ドラッグ&ドロップまたは Open パネルで選択されたフォルダが、3つのCOLMAP Workspace レイアウト (sparse/0/, sparse/, またはルート) のいずれかに一致するかどうか、および再構築がバイナリ (cameras.bin) かテキスト (cameras.txt) かを自動検出します。バイナリパスはCOLMAPバイナリパーサーを使用し、テキストパスはETH3Dローダーを使用します — どちらも同じSfM結果モデルを生成し、パイプラインの残り (画像をインポートし、MCMCトレーニングを開始) はソースに依存しません。画像はアプリ Sandbox ブックマークシステム経由でセキュリティスコープで開かれるため、インポートはApp Storeバージョンでも動作します。特に「Metashape エクスポートを再構築なしで再計算」のケースを意図しています。Fileメニュー項目で言及されている検出は、選択されたフォルダが認識可能なWorkspaceでない場合、アプリログで警告します。

簡単に言うと

特にMetashapeユーザー向けの機能。MetashapeまたはRealityCaptureのライセンスを持っていて、そこでカメラ再構築を行った場合、エクスポートフォルダをここに簡単にドラッグして、すぐにトレーニングを開始できます。RadianceKitがSfM自体を実行しないため、大きなシーンで数時間の計算時間を節約します。

どのバックエンドをいつ？

| シナリオ | 推奨バックエンド |
|-----------------------------------|---|
| オブジェクトスキャン、50-200 枚 | Q1 Apple Photogrammetry |
| 大規模屋外 / ドローン / >500 枚 | Q6 Workspace インポート (Metashape または COLMAP で計算してから読み込み) |
| Metashape/RealityCapture エクスポートあり | Q6 インポート (SfM 不要) |
| ETH3D / 学術 COLMAP テキストセット | Q3 COLMAP テキストインポート |
| 厳密な App Store 準拠 + オービットシーン | Q4 Native インクリメンタル |
| Q4 が失敗 | Q5 Native グローバル (自動) |
| ETH3D ベンチマークデータ | Q3 (autotest precomputed) |

クイック比較

| バックエンド | App Store | Sandbox | 外部 バイナリ | 最適用途 | 最大 ~Cams |
|---------------------|-----------------------|---------|---------------|---------------|-------------|
| Q1 Apple PG | ✓ | ✓ | — | Orbit-Object | ~300 |
| Q2 COLMAP Binary | ✗ (Developerビルドのみ) | — | colmap/glomap | Outdoor large | ~5 000 |
| Q3 COLMAP テキストインポート | ✓ | ✓ | — | Bench rigs | ~1 500 |
| Q4 Native インクリメンタル | ✓ | ✓ | — | Orbit-Object | ~200 |
| Q5 Native グローバル | ✓ | ✓ | — | Q4 フォールバック | ~1 351 |
| Q6 Workspace インポート | ✓ | ✓ | — | Metashape 再利用 | ソース 毎 |

章

第 10 章 — 初心者モード

初心者モード (英語 Simple Mode、Cmd+1) は、3D Gaussian Splatting シーンを初めて再構築する人向けのガイド付きワークフローです。Inspector フィールドでいっぱい のサイドバーを表示する代わりに、アプリは 4 つのステップを案内します: まず画像やビデオをインポートして品質プリセットを選択し、次に処理 (SfM + トレーニング) が実行され、完成したシーンを 3D プレビューで確認でき、最後に希望の形式にエクスポートします。ウィンドウ上部の細い進捗バーは、現在どのステップにいるかを常に示します。

すべての操作要素を同時に表示する Expert モード (Cmd+2) と比較して、初心者モードは未使用のオプションを非表示にし、画像が少なすぎる場合や悪い場合に検証警告を提供し、各ステップで現在の状態で意味のあるボタンのみを提供します。初心者モードと Expert モードの間でいつでも切り替え可能 (Cmd+1 / Cmd+2)、すべての状態 — インポートされた画像、選択されたプリセット、現在実行中のトレーニング、完成した点群 — は保持され、もう一方のモードですぐに利用できます。

Z1 — インポート (画像とプリセットの選択)

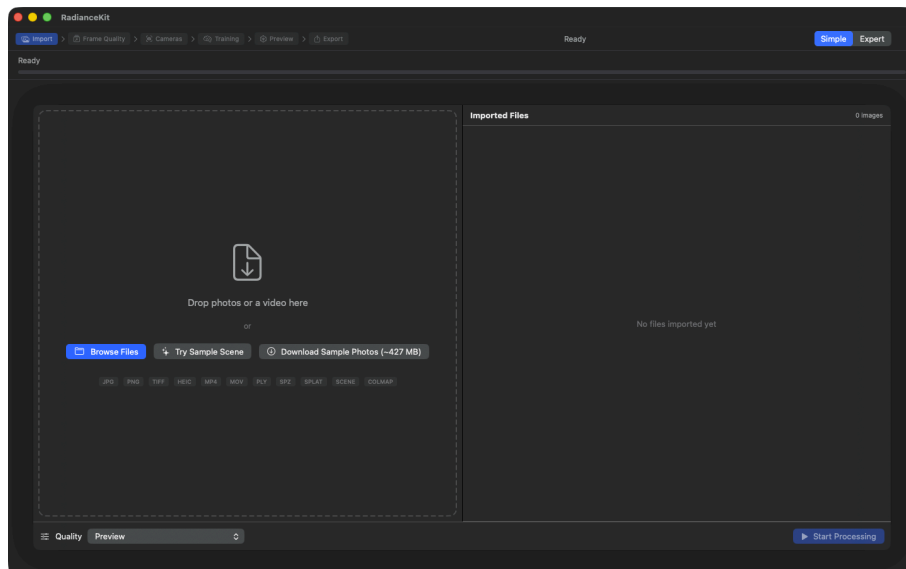


図 35: Simple モードステップ 1 — 画像インポート前の空のドロップゾーン、上部の Crumb Trail (Import → Frame Quality → Cameras → Training → Preview → Export)、Format Pills (JPG/PNG/TIFF/HEIC/MP4/MOV/PLY/SPZ/SPLAT/SCENE/COLMAP)

画像で見えるもの Crumb Trail (Import アクティブ) は 4 ステップワークフローを示します。3 つの CTA を持つ左のドロップゾーン: 「Browse Files」 (NSOpenPanel)、「Try Sample

Scene」(バンドルされたデモ)、「Download Sample Photos (~427 MB)」(Mip-NeRF360 flowers サブセット)。下の Format Pills は受け入れられるすべてのファイルタイプをリストします。右側には「Imported Files」、カウンター「0 images」、空状態「No files imported yet」。下部の品質ピッカー (デフォルト: Preview) と「Start Processing」(画像がない限り無効)。

最初のステップは、アプリに画像素材を提供することです。中央の大きな破線フィールドへのドラッグ&ドロップ、「Browse Files」ボタン、または同梱の Sample シーンをクリックすることで。右側には、解像度とファイルサイズを持つすべてのインポートされた画像のリストが表示されます。下のフローティングツールバーで品質プリセットを選択し、「Start Processing」でパイプラインを開始します。検証警告 (< 3 または < 10 画像で赤、10~19 でオレンジ) は、アプリが意味のある再構築を期待するかどうかを示します。

C-01 ProgressIndicator (ステップ表示)

場所

ワークフローの上、常に表示。

技術詳細

Stage 割り当て付きのパイプライン全体 (Frame Quality → SfM → Training) にわたる水平進捗バーを表示します: Frame Quality は 0~5% (Phase 3.11、非常に短い)、SfM はバーの 0~30%、Training 30~100% を占めます。横には Status Text と Phase 名付きパーセント表示 (「SfM 41%」、「Training 12 500/20 000」) があり、ユーザーが見かけ上の逆行「41% SfM → 25% Training」をエラーとして読まないように — バーはパイプライン全体の進捗を示し、サブステージではありません。ETA 計算は、十分なトレーニング速度が測定されるとすぐに開始されます (通常、最初の 100 反復後)。同じ表示が Expert モードでも Inspector の上に使用されます。

簡単に言うと

一番上の細いバーは、ワークフロー全体を通じたあなたの地図です。アプリが現在何をしているか (カメラ整列、トレーニング実行中、...) だけでなく、全体でどれだけ進んだかも伝えます。意図的に、カメラ計算がバーの最初の 1/3 を占め、実際のトレーニングが後ろの 2/3 を占めるように分割されています — そうでなければ、SfM 後に突然進捗がゼロに戻ったように見えます。リラックスして、バーを見るだけで大まかなステージを確認できます。横のテキストは SfM ステージ (例えば「SfM 41%」) かトレーニング (例えば「Training 12 500/20 000」) を伝え、数字が混乱を引き起こさないようにします。ETA が表示されない場合、トレーニングがまだ若すぎるだけです — アプリは十分な速度を測定してから推定します。

C-03 DropZoneView (ドラッグ&ドロップ領域)



場所

Import ステップの左側、シンボル付き大きな破線長方形。初心者モードではラベル「Drop photos or a video here」で表示されます。



技術詳細

シンボルが短く跳ね、Drag アイテムがフィールド上に浮かぶとすぐに背景を着色するドロップ領域。JPG、PNG、TIFF、HEIC、MP4、MOV、PLY、SPZ、.splat、.radiancecene バンドル、ディレクトリを受け入れます。タイプ別の Drop Routing: 画像は収集され、ソートされて渡される、ビデオは Frame Sampling パスをトリガー、Splat ファイルは直接プレビューを開く、Scene バンドルは読み込まれる。ディレクトリは列挙され、含まれるすべての画像がインポートされます。Sandbox 準拠 アクセス用の Security Scoped Bookmarks が正しく取得および解放されます。サポートされていない拡張子は 5 秒間警告バナーとして表示されます。

簡単に言うと

大きな破線フィールドが最初のステップのメイン操作です。写真やビデオ、またはフォルダ全体を単にドラッグします — アプリは知っているすべてを受け入れ、残りを無視します。フィールドが青くなり、シンボルが短く跳ねた場合、アプリは Drag を認識しました。離すと、すぐにインポートが開始されます: 画像は右側のリストに入り、ビデオは自動的に Frame Sampling ステップをトリガーし、すでにトレーニングされた .ply / .spz / .splat ファイルは直接プレビューを開きます。形式がまったく適合しない場合 (PDF や BMP など)、上端に短いヒントが表示されます — アプリは未知の素材を黙って飲み込みません。

C-05 Browse Files ボタン



場所

Drop ゾーン内、目立つボタン。



技術詳細

複数選択と、ファイルタイプ JPG、PNG、TIFF、MP4、MOV、フォルダ、アプリ独自の Scene 形式を持つ macOS ファイルダイアログを開くボタン。結果 URL は Security Scoped で、Drag and Drop と同じインポートパス経由で転送されます。ユーザーがフォルダを選択すると、再帰的に画像が列挙されます。

簡単に言うと

Drag and Drop が不便な場合は、このボタンをクリックして macOS ファイルダイアログで写真に移動するだけです。複数のファイルを同時に選択 (個々の画像で Cmd-Click) するか、フォルダ全体を選択できます — アプリはフォルダをサポートされるすべての画像タイプについて再帰的に検索します。撮影がサブフォルダにネストされている場合 (例えば「shoot-day1/」、「shoot-day2/」) に特に実用的です — メインフォルダのクリックで十分。機能的には、このボタンは Drag and Drop と同じことを行います。便利な方法を選択してください。

C-06 Try Sample Scene ボタン

場所

Drop ゾーン内、アプリバンドルに Sample Scene が含まれ、まだ画像/Splat が インポートされていない場合のみ表示。



技術詳細

次の場合にのみ表示されます: (a)

sample-scene.splat、.spz、または .ply がアプリバンドルに存在する、かつ (b) まだ画像/ビデオがインポートされておらず、点群もない場合。クリックすると、完成した点群がロードされ (最小の形式を優先 — .splat ~3 MB、.spz ~1.4 MB、フォールバック .ply)、400 ms 後に Flowers シーンのオリジナルメタデータからのハードコードされたカメラ値を、審美的に意味のある入力視点として設定します。

簡単に言うと

アプリを初めて起動して、最後に何が出てくるか単に見たい場合 — ここをクリック。これにより、すぐに回転およびエクスポートできる、完成したトレーニング済み花のシーンが開き、アプリは計算する必要がありません。カメラは審美的に意味のある入力視点に事前設定されているため、すぐにか何か美しいものが見えます。自分の撮影に取り組む前に、3D コントロールとエクスポートステップをリスクなしで試すのに最適です。自分の画像をインポートするとすぐに、ボタンは自動的に消えます — プロジェクトが完全に空である限りのみ表示されます。

C-07 Download Sample Photos ボタン

場所

Drop ゾーン内、「Try Sample Scene」の隣。同じ可視性条件。



技術詳細

約 427 MB の 960 のフル解像度フレームをロードしてアプリに入力するダウンロード (リポジトリ github.com/bkindler/radiancekit-sample-photos) をトリガーします。ダウンロード中、ボタンは無効化されます。進捗は上部の Progress Bar に「Downloading X%」として、独自のステージに表示されます。このステージは独自の 0~100% スケールを保持し、後の SfM ステージと重ならないためです。

簡単に言うと

Sample シーンとまったく同じですが、完成した結果の代わりにソース写真で。これにより、パイプライン全体を自分で 1 回実行し、SfM とトレーニングが Mac で実際にどれくらいかかるかを確認できます。ダウンロードは大きい (DVD の約半分 = 427 MB) ですが、1 回だけです — その後、写真はローカルで、異なるプリセットでパイプラインを何度でも再起動できます。ダウンロード中、上部の進捗バーは現在のダウンロード状態をパーセントで表示し、開始するタイミングを推定できます。ヒント: 高速 Wi-Fi または有線ネットワークを使用してください — 427 MB はそうでないと引き伸ばされます。

C-09 Quality Presets ピッカー

場所

Import オーバーレイのフローティング下部ツールバー、Start ボタンの左側。



技術詳細

ラベル「Quality」付きの操作要素が、利用可能なプリセットをカテゴリ別 (Classic / MCMC / Custom) にグループ化します。組み込みプリセットはカテゴリ別にグループ化されます。セクションヘッダーはハードコードされています。Custom プリセットは存在する場合のみ表示されます。ロック状態: Free リスト (Quick + Preview) にないプリセットは、ユーザーが購入していない場合、名前に「🔒」接尾辞が付きます。選択時、ピッカーは Preview に戻り、自動的に Purchase Sheet を開きます。選択時、プリセットが適用され、トレーニング構成全体が置き換えられます。

簡単に言うと

ここで、アプリがどれだけ正確に、どれだけ長く計算すべきかを選択します。「Quick」と「Preview」は購入なしで使用可能で、数分で最初の結果を提供します。一画像が意味があるかどうかをテストするのに理想的です。「Balanced」と「Quality」はフルバージョンが必要で、はるかに鮮明なモデルを提供しますが、分ではなく時間がかかります。MCMC は、より少ない Gauss Splat で十分な異なる戦略です。モデルを後でコンパクトにエクスポートしたり Web に置きたい場合に良い。Premium プリセットは名前の小さな鍵シンボルで識別できます。ライセンスなしでタップすると、ピッカーは Preview に戻り、Purchase Sheet が自動的に開きます。経験則: 常に Preview で開始し、結果を見て、それから長い実行が価値があるかどうかを決定します。

C-10 Start Processing ボタン

場所

Import オーバーレイのフローティング下部ツールバー、プリセットピッカーの右側。



技術詳細

画像もビデオもインポートされていない限り、グレーのままのボタン。クリックでパイプラインが開始され、Stage マシンが Frame Quality → SfM → Training の順序に切り替わります。ボタン自体には他のステータスはありません。実行中の処理は代わりに別の処理画面として表示されます。

簡単に言うと

「開始」ボタン。グレーのままの間は、入力画像またはビデオはまだありません。写真をドラッグすると、アクティブになり、それをクリックして SfM とトレーニングを順次開始します。それ以降、アプリが全プロセスを引き継ぎ、処理画面 (Z2) に自動的に着地します。何もクリックする必要はありません。トレーニング終了後、アプリはプレビュー (Z3) に再び切り替わります。考え直した場合、その後もいつでも Cancel で中止できます。

C-11 Video Sampling スライダー



場所

右の画像リスト、画像ではなくビデオがインポートされた場合のみ表示。



技術詳細

0.5 fps – 30 fps、0.5 ステップのスライダー。変更時、フレーム密度が更新され、密度とビデオ長から計算されたターゲットフレーム数 (最小 10) も追加で更新されます。リスト要素がスライダーからマウスイベントをブロックするため、スライダーは画像リストの外に配置されています。スライダーの下に、計算されたターゲットフレーム (「247 frames」) とビデオ長 (「1m23s video」) があります。ツールチップが警告します: 「密度を 2 倍にするとフレーム数が 2 倍になり、SfM 時間が ~100% 増加します。」

簡単に言うと

写真の代わりにビデオをインポートした場合、このスライダーは、アプリがビデオからどれだけの単一画像を抽出するかを決定します。より多くの画像 = より良い品質、しかし線形により多くの計算時間。30 秒の Orbit ビデオでは、5 fps (150 画像) が良い開始点です。1 分の撮影では、3 fps で十分なことが多い。スライダーの下、アプリは現在の設定でどれだけのフレームが出てくるかをライブで表示します — これにより、意味のある約 100~300 画像の範囲をヒットしているかどうかすぐに見えます。結果が悪い場合、スライダーを右に動かして再実行します。フレームレートを 2 倍にすると、SfM 期間もおおまかに 2 倍になります。

C-12 Clear All ボタン



場所

右の画像リスト、右下、画像がインポートされた場合のみ表示。



技術詳細

赤いボタン。クリックは、タイトル「Clear all imported files?」とメッセージ「N images will be removed.」で確認ダイアログを開きます。確認は、すべてのインポートされた画像/ビデオ、ステージングディレクトリ、点群、トレーニング ステータス、SfM 結果、すべてのキャッシュを消去します。ステージは Import に戻ります。Cancel ですべてが保持されます。ダイアログは非破壊的なデフォルトパスとして構成されています (破壊的ボタンは赤でマーク)。

簡単に言うと

完全に新規開始したい場合、ここをクリック。確認質問は、削除が、すでに計算されたカメラとトレーニング結果を含む現在のインポートをすべて破棄するため表示されます — 元に戻せません。選択した画像素材を完全に置き換えたい、または新しいプロジェクトを開始する前に古いプロジェクトを取り除きたい場合に意味があります。注意: 単一の画像を削除することは、右側のリストを介して (次の項目参照) 行います。このボタンを介してではありません。ディスク上のファイルは削除されません — アプリは参照を忘れるだけです。

C-13 File List ForEach (単一画像の削除) 場所

右の画像リスト、各エントリ。

 技術詳細

スワイプで削除付きのインポートされた画像のリスト。画像ごとに、アイコン、ファイル名、解像度 (「1920 × 1080」)、ファイルサイズ (フォーマット KB/MB) を持つ行。解像度はメタデータキャッシュから取得され、UI がブロックされないように、画像ヘッダーから非同期で埋められます。削除アクションは macOS 風のスワイプ削除 (行でのトラックパッド左スワイプ) と、選択された行での Delete キーボードを提供します。注意: 明示的なマイナスボタン、Backspace、Cmd-Z による Undo を持つ拡張画像削除パスは、Expert モードのみの Project Navigator で追加されました — 初心者モードではスワイプ削除のままです。

 簡単に言うと

右側のリストは、各インポートされた画像を解像度とファイルサイズとともに表示します — 高解像度と低解像度の素材を混ぜ合わせたかどうかを一目で確認するのに実用的です。単一の画像を削除するには、iOS Mail のようにトラックパッドで 2 本の指で左にスワイプするか、それを選択して Delete を押します。アプリはファイル自体を削除しません。それを現在のプロジェクトから単に削除するだけです。適切なマイナスボタンや Cmd-Z Undo が必要な場合、Expert モード (Cmd+2) に切り替えてください。そこには Project Navigator にあります。初心者モードでは、シンプルなスワイプパターンが意図的にそのまま残ります。

C-15 Validation Warnings (3 段階ティア) **場所**

画像リストの下、Clear All ボタンの上。

 **技術詳細**

インポートされた画像の数に基づく 3 つの連続した閾値 (画像が存在し、ビデオがない場合のみアクティブ): - < 3 画像: 赤バナー (red octagon)、テキスト「At least 3 images are required. Camera alignment cannot be computed from fewer images.」 - 3~9 画像: 赤バナー、テキスト「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 画像: オレンジバナー (warning triangle)、テキスト「Workable, but quality usually improves with 20+ images and good coverage around the scene.」

20 画像から、バナーは消えます。閾値はハードコードされており、経験的な 560+ トレーニング実験に基づいています。

 **簡単に言うと**

アプリは、どれだけ多くの画像をインポートしたかを見て、色付きの評価を提供します。赤は意味します: これは高い確率で何もならないだろう — SfM がカメラを計算できないか、トレーニングが少なすぎる素材にオーバーフィットします。オレンジは意味しません: 機能する可能性があります。が、アルゴリズムが画像間のオーバーラップをほとんど見つけないため、トップ品質を期待しないでください。バナーなしは意味しません: 良い前提条件、十分な素材があります。本当にクリーンなモデルが欲しい場合、被写体の周りで均等に分布した少なくとも 30~50 ショットを目指します — 屋外シーンや大きな部屋ではさらに多くてもかまいません。警告にもかかわらず開始できますが、SfM がコメントなしで中止したり、モデルが穴があいているように見えても驚かないでください。

C-16 COLMAP Workspace 検出 場所

フォルダのドロップ時 — 可視のボタンではなく、検出ロジック。

 技術詳細

ディレクトリのドロップ時、3つの正規 Workspace レイアウトのいずれかが含まれているかチェックされます: `sparse/0/cameras.bin`、`sparse/cameras.bin`、または直接 Root に `cameras.bin`。それが該当する場合、標準画像列挙が中止され、代わりに モーダルアラートが開かれ、ユーザーに既存の再構築を使用するか、画像を Apple Photogrammetry で新しく実行するかを尋ねます。テキスト形式 Workspace (`cameras.txt`) と ETH3D エクスポートでも同じパス。詳細は第 9 章 Backend Q6 参照。初心者モードでも Expert モードと同じように動作します。

 簡単に言うと

Metashape, RealityCapture, または COLMAP ですでに作業し、ここでカメラ計算を実行した場合、エクスポートフォルダをここに単純にドラッグできます。RadianceKit は内容から自動的にそれが COLMAP Workspace であることを認識し (`sparse/0/cameras.bin` などをチェックします)、完成した計算を引き継ぐか自分で再計算するかを尋ねます。引き継ぐと、SfM が完全にスキップされ、トレーニングがすぐに開始されるため、大きなシーンで数時間の待ち時間を節約します。テキスト形式 Workspace (`cameras.txt`) と ETH3D エクスポートも認識されます。この機能は初心者モードでも Expert モードと同じように利用可能です。詳細は第 9 章 Backend Q6 にあります。

次のステップへいつ?

(a) 少なくとも 1 つの画像またはビデオがインポートされ、(b) 検証バナーがオレンジまたは消えるとすぐに、Start Processing をクリックできます。赤バナーでもアプリは開始させますが、高い確率で処理をすぐに再び中止できます。推奨: 少なくとも 20 画像、鮮明、連続したショット間で明確なオーバーラップを持つ、被写体までほぼ同じ距離からのすべて。開始前に時間予算に合うプリセットを選択します — 30 画像と Quick プリセットでは数分以内に完了し、Quality では 1~2 時間かかります。

Z2 — 処理 (SfM + トレーニング)

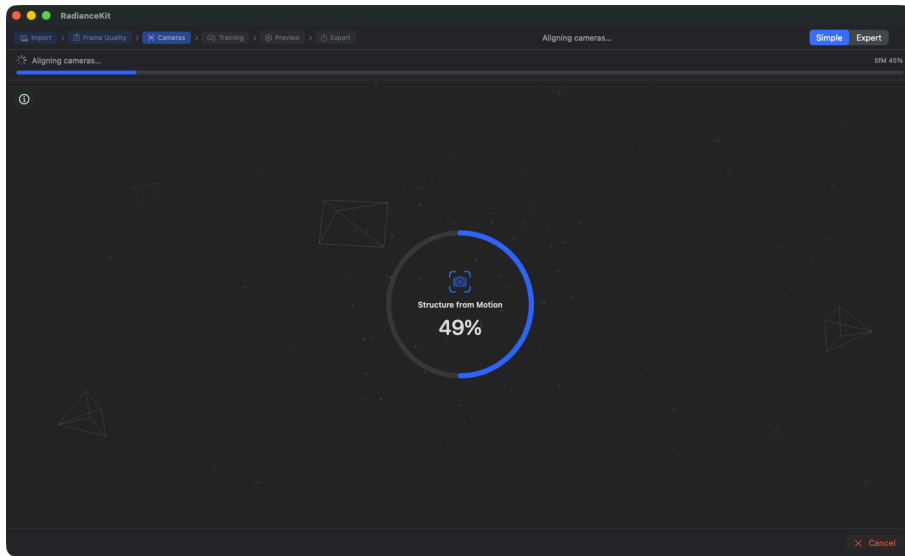


図 36: Z2 SfM フェーズ — Stage アイコン「Structure from Motion」、大きな円の中で 41%、上部ステータスバーは「SfM 25%」、右下に Cancel ボタン

SfM フェーズ (カメラが整列されています): 大きな進捗円は、サブステージ進捗 (ここでは実行中の Apple Photogrammetry セッションの 41%) を表示します。左上の Status Text「Aligning cameras...」。Crumb Trail は「Cameras」をアクティブステージとしてマークします。上部ステータスバーはパイプライン全体の進捗 (25%) を表示 — SfM はバーの最初の半分を占めます。背景のフローティングワイヤフレームカメラは、ポーズが推定されていることを示唆します。

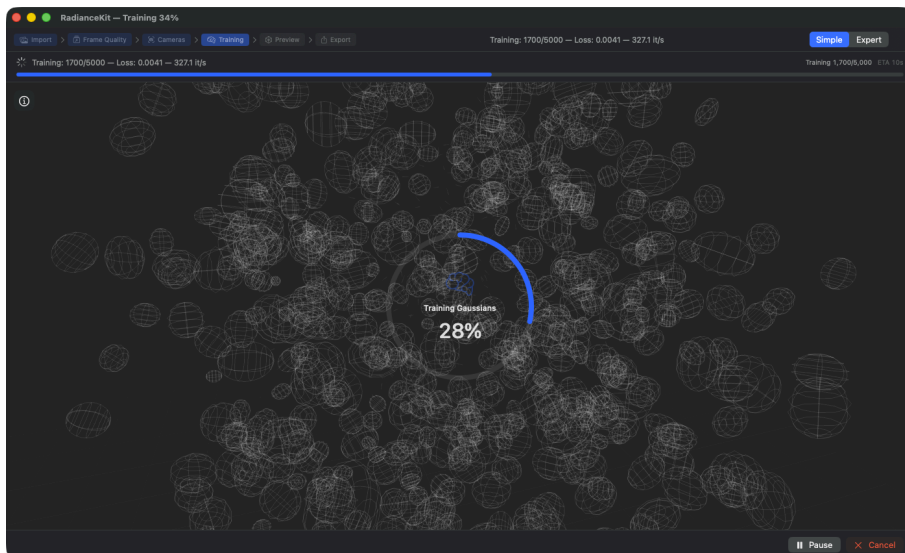


図 37: Z2 トレーニングフェーズ — Stage アイコン「Training Gaussians」で 6%、上部のライブメトリック (Training: 400/5000 — Loss: 0.1642 — 138.7 it/s)、ETA 33s、下部に Pause/Cancel

トレーニングフェーズ (Gaussian が最適化されています): サブステージアイコンが「Training Gaussians」に切り替わり、パーセントが選択されたプリセットから反復をカウントします (ここでは Preview プリセットで 400 / 5 000 = ステージの 8%)。ライブメト

リック行は、Loss 値 (0.1642)、秒あたり反復 (138.7 it/s)、ETA (33 秒) を表示します。パイプライン全体の進捗は、このフェーズ中に 50% から 100% に上昇します。SfM フェーズの Cancel のみとは異なり、Pause ボタンは後で Resume を可能にします。Cancel はトレーニング結果を破棄し、Z1 に戻ります。

パイプラインが実行されるとすぐに、アプリは Import オーバーレイを隠し、フルスクリーン処理画面を表示します。中央には、Stage アイコン、Status Text、パーセント数値を持つ大きな進捗円 (220 × 220 ピクセル) が走ります。背景では、控えめな Splat アニメーションが実行中の計算を象徴的に視覚化します。左上には、トレーニングと SfM からのライブメトリックを表示する情報パネルを表示できます。下には、Pause/Resume、Cancel、エラーの場合は Retry ボタンがあります。

C-18 SplatTrainingView (背景アニメーション)

場所

Progress 円の後ろのフルスクリーン背景、中止またはエラーで非表示。

技術詳細

パイプライン進捗 (0...1) に応じて増加する数の小さなアニメーション化された Splat 粒子をレンダリングする装飾アニメーション。ソースは、SfM フェーズを 0~0.2 に、Training を 0.2~1.0 に (Frame Quality を 0~0.05 に) マッピングする計算された進捗値です。これにより、トレーニング中に Splat が見えて「構築」されます。純粋に装飾的 — 表示は現在のトレーニングの実際の間接結果を示しません (それは Expert モードのライブプレビューでしょう)。Cancel や Failure では非表示になり、Status 円のみが見えます。

簡単に言うと

背景では、計算中に画面がそれほど空に見えないように、ダンスする点の小さなアニメーションが実行されます。これは実際の 3D モデルではありません — ステップ Z3 のトレーニング後にのみ見えます。ただし、アニメーションは同じトーンを持っているため、おおよその密度から、トレーニングがどれくらい進んでいるかを読み取れます。最初は少数の点のみが見え、終わりには背景がはるかに密に満たされます — 円のパーセント表示に加えての美しい視覚的インジケータ。アニメーションが気になる場合 (例えばバックグラウンドで並行して作業したいため)、Expert モードに切り替えるとそこでは消えます。

C-19 大きな Progress 円



場所

処理画面の中央、220 × 220 ピクセル。



技術詳細

2つの重ねられたリング: 外側にミュートされた Track リング、内側に Accent または Red Stroke (エラーで赤) を持つ塗りつぶされた Progress リング。円の中には、Stage アイコン (Training 用脳、SfM 用カメラ、Video Frame Extraction 用 フィルム、Frame Quality 用 Sparkles)、Stage タイトル、32 ポイント Rounded フォントでライブアニメーション化されたパーセント数値。アイコンは処理がアクティブな限り 柔らかく脈打ちます。表示は 30 Hz タイマーで現在の実際の進捗の方向に滑らかに補間 されます — Constant-Creep (0.0003/フレーム) + Proportional Anteil (ギャップの 4%)、次の予想されるマイルストーンの 80% に設定する Soft Ceiling (SfM ではハードコードされた マイルストーンテーブルから)。これにより、実際の SfM 更新が数秒ごとにしか入っても、進捗が流動的に感じられます。

簡単に言うと

中央の大きな円は、アプリが計算している間のメイン表示です。実際の計算更新が数秒ごとにしか入っても柔らかく満たされ — これにより、フリーズしたパーセントに何分間も凝視する代わりに、何かが起こっているように感じさせます。中央のシンボルは、現在フレームが抽出されているか (フィルムアイコン)、カメラが整列されているか (カメラアイコン)、または Gaussian がトレーニングされているか (脳アイコン) によって切り替わります。パーセント数値は現在のサブステップを指します — 全パイプラインは一番上の細かいバーで見えます。エラー時、リングは青ではなく赤になり、アイコンは脈打たなくなるため、何か問題が発生したことがすぐわかります。

C-22 Info ボタン (メトリック表示)



場所

処理画面の左上、32 × 32 ピクセル。



技術詳細

Material 背景を持つシンプルなボタン。Info パネルをオンまたはオフに切り替えます。アクティブ時、アイコンは Info Circle Outline と Info Circle Filled の間で切り替わります。柔らかいフェードインアニメーション。ツールチップは「Show detailed processing metrics」。

簡単に言うと

デフォルトでは、画面は意図的に整然としています — 大きな進捗円だけ、それ以上は最初は見えません。技術的に興味のあるユーザーとして、何が起きているか (どの反復、Loss の高さ、いくつの Gaussian) をより正確に知りたい場合は、左上の **i** シンボルをクリックします。下に小さなパネルが開き、すべてのライブ値が表示されます。再度クリックすると非表示になります。設定は永続的ではありません — 各新しいトレーニング実行で、パネルは最初再び非表示になり、これは初心者を驚かせないように意図的に選択されています。

C-23 Info Panel (ライブメトリック)



場所

処理画面の左下、`showProcessingInfo == true` の場合のみ表示。



技術詳細

Ultra-Thin Material 背景を持つ 2 カラムパネル。左カラム: ステージ固有の情報行 — SfM 用は Status Text とパーセント、Training 用は Iteration、結合 Loss、L1 Loss、D-SSIM Loss、Gaussian カウント (オレンジ色)、Speed (it/s)、経過時間、計算された ETA、SH Degree、学習率。右カラム: Status Text、Time Info 文字列、インライン Loss チャート (C-28 参照)、Discoverability Nudge (C-32 参照)。すべての値はトレーニングステータスから読み取られ、各トレーニング Tick で更新されます。

簡単に言うと

Info Panel は、Expert モードで Inspector サイドバーに永続的にあるであろうすべてのライブ値を表示します: 現在の反復、Loss 値 (小さい = 良い)、Gaussian の数、速度、推定残り時間、SH Degree、学習率。右側では、追加で小さな Loss 曲線が実行され、トレーニングが正しい方向に進んでいるかを一目で伝えます。トレーニングが鈍く感じる場合、ここを見ると役立ちます — もう落ちない Loss、もう下がらない ETA は問題を示唆します。Loss が爆発する (突然巨大になる) または NaN を表示する場合、トレーニングが不安定になり、Cancel + Retry または別のプリセットへの切り替えが意味があります。

C-25 Pause/Resume ボタン



場所

下部ナビゲーションバー、トレーニングステージ中のみ表示 (SfM 中は NOT)、処理が実行されている限り。



技術詳細

Bordered Button。ステータスに応じて Pause または Resume を呼び出します。ラベルは「Pause」(Pause アイコン付き) と「Resume」(Play アイコン) の間で切り替わります。SfM ステップ中、Apple Photogrammetry に Pause セマンティクスがないため、ボタンは表示されません。Pause 状態は、反復、Gaussian ステータス、Optimizer Momentum を完全に保持 — Resume は中断したところから続行します。

簡単に言うと

トレーニングが実行されている間、いつでも一時停止して後で続行できます。Mac で GPU を多く必要とする他のことを行いたい場合に意味があります — 例えばビデオ編集、ゲームテスト、別のアプリからのレンダリングエクスポート。Pause をクリックし、自分のことをして、Resume をクリックすると、トレーニングはまさにそこで続行されます。反復カウンター、Gaussian 数、Optimizer Momentum は完全に保持され、Pause 状態は品質を犠牲にしません。SfM フェーズ中、Pause は利用できません — Apple Photogrammetry は Pause 機能を知らないため、緊急時には Cancel で作業する必要があります。

C-26 Cancel ボタン 場所

下部ナビゲーションバー、処理が実行されている間 (SfM またはトレーニング) に表示。

 技術詳細

赤い Bordered Button。タイトル「Stop and discard progress?」、ボタン「Discard Progress」(破壊的) と「Keep Running」(Cancel) で確認ダイアログを開きます。確認で、Cancel フラグが設定され、トレーニングタスクが終了され、必要に応じて SfM サブプロセスが終了され、中止ステータスを持つ Summary 行が JSONL ログに書き込まれます。Pause とは異なり、トレーニングバッファとステータスは破棄されます。

 簡単に言うと

中止ボタン。Pause と異なり、これは最終的です—後で再起動する場合、処理は最初から実行され、すでにトレーニングされたすべての反復が失われます。プリセットを間違えた、トレーニングがあまりに遅い、またはアプリが明らかにごみの結果を生成して待ちたくない場合に意味があります。実際中止前に、誤って数時間の計算時間を失わないように、アプリは確認ダイアログで再度尋ねます。短期間だけ中断したい場合は、Pause を使用してください。

C-27 Retry ボタン 場所

下部ナビゲーションバー、パイプラインが失敗した場合に表示 (SfM ステータスが「SfM failed」で始まるか、トレーニングがエラー状態の場合)。

 技術詳細

Accent Button。パイプライン全体を再起動します。開始前に、まだインポートされた画像/ビデオが存在するかチェックされます。以前のエラーログは JSONL ディレクトリに残ります。新しい実行は、現在のタイムスタンプを持つ新しい Log ファイルを書き込みます。

 簡単に言うと

SfM またはトレーニングがエラーメッセージで中止された場合、ここで再試行できます。多くのステップ (RANSAC, Densification) にランダム部分があり、最初が失敗したところで 2 回目の試行が成功する可能性があるため、時々これが役立ちます。パイプライン全体が新しいフレッシュな JSONL ログファイルで最初から実行されます—SfM とトレーニング。2 回目の試行も失敗する場合、ほとんどの場合、入力画像が問題です (少なすぎる、オーバーラップが少なすぎる、ブレ、悪い光)。その場合 Back で戻り、素材を交換してください。ヒント: 並行してトレーニングログ (Help → Open Training Logs) を見てください。そこには、具体的にどこで詰まったかがより詳細に記載されています。

C-28 インライン Loss チャート



場所

Info Panel 内、右カラム、非空履歴を持つトレーニング中のみ表示。



技術詳細

コンパクトな描画領域 (40 ピクセル高)、Loss History をアクセント色の 1 ピクセル線として描画します。データは有限値にフィルタリングされます (不安定なトレーニング用 NaN 保護)。Min/Max は全履歴にわたって計算されます — チャートは 値範囲に自動ズームします。最後の Loss 値はチャートの右上にあります。履歴自体は 各トレーニング Tick (通常 100 反復ごと) でアプリ状態に構築されます。

簡単に言うと

トレーニングが「収束」(線が右に落ちる)しているか、詰まっているか爆発している(線が平らまたは上昇)かを一目で示す小さな Loss 曲線。健康なトレーニングでは、線は最初は急に落ち、その後平坦になります — これは予想される進行で、半減曲線に似ています。チャートは現在の値範囲に自動的にズームするため、トレーニング終了時の小さな改善も表示されたままです。線が突然上に飛んだり、フリーズしたりすると、何か問題が発生しているという良い兆候です — 素材に問題があるか、別のプリセットがより適しているかのどちらかです。チャートは、左上の *i* シンボルで表示する Info Panel にあります。

C-32 Discoverability Nudge (Expert モードヒント)



場所

Info Panel 内、右カラムの下部、トレーニング中かつ初心者モードでのみ表示。



技術詳細

目アイコンと Caption テキスト「Switch to Expert Mode (⌘2) for live splat preview」を持つ小さな行、控えめなトーンと 10 ポイントフォントで。インタラクティブ要素ではなく、単なるヒント。クリックに反応しません — ユーザーは実際に Cmd+2 を押すか、メニュー Mode → Expert Mode をクリックする必要があります。

簡単に言うと

Expert モードでトレーニング中に、3D モデルの現在の中間バージョンが Viewport でライブに見えることを示す控えめなヒント。初心者モードでは、インターフェイスを落ち着いた状態に保つために意図的に非表示にされています — しかし多くのユーザーはこの機能があることをまったく知らないため、ここで穏やかに指摘します。Cmd+2 を押すと、トレーニングはバックグラウンドで続行され、目の前でモデルが組み立てられるのを見ることができます。これは、数千の反復後にすでに結果が価値があるかを推定するか、中止して新規開始するための良いツールでもあります。Cmd+1 はいつでも初心者ビューに戻ります。

次のステップへいつ?

トレーニングが正常に完了するとすぐに、アプリは自動的に Z3 (プレビュー) に切り替わります — 何もクリックする必要はありません。下部ナビゲーションバーは Pause/Cancel から Back ボタン (Import に戻る) と Export ボタン (Export に進む) に切り替わります。エラーの場合 (赤いエラーメッセージ、Stage アイコンは X)、代わりに Retry が表示され、再度開始するか、Back で Import に戻って画像素材を変更するかを決定する必要があります。

Z3 — プレビュー (3D モデルを回転)

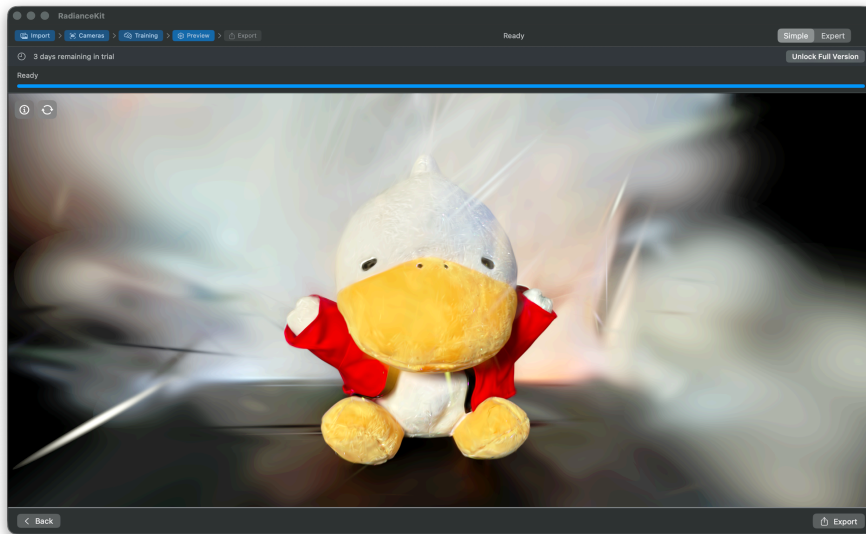


図 38: 3D Viewer 付き Simple モードプレビューステップ

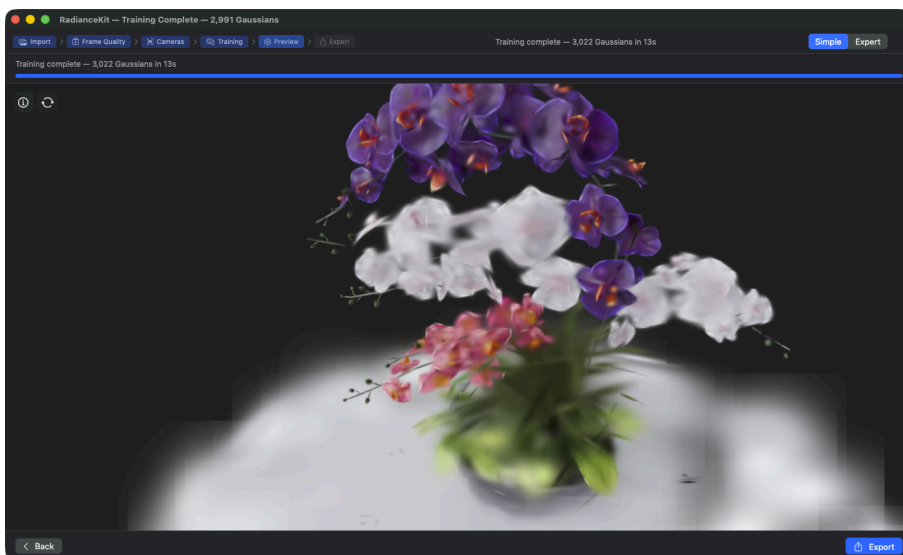


図 39: トレーニング完了後の Z3 プレビュー — Bjoern の Blender ブーケが再構築され、ヘッダーは「Training complete — 3,022 Gaussians in 13s」を表示、下部に Back と Export ボタン

画像で見えるもの Crumb Trail は「Preview」を アクティブステージとしてマークします。フルスクリーン 3D Viewport は完成した ブーケシーンをレンダリングします (Bjoern の合成 Blender テストセット、960 半球 カメラから 60 フレームサブセット)。ヘッダーステータスバー: 「Training complete — 3 022 Gaussians in 13 s」 — 最終 Gaussian 数とトレーニング時間を提供。Viewport の Drag はカメラを回転 (Yaw/Pitch)。Scroll Wheel は View Direction に沿ってズーム。「Back」ボタン (左下) は Resume または Re-Run のために Z2 に戻ります。「Export」ボタン (右下、Primary) は Z4 に進みます。

トレーニング完了後、アプリは自動的にプレビューに着地します。ここでは完成した Gaussian Splatting モデルをフルスクリーン Metal ビューで見て、マウスとトラックパッドで回転、ズーム、パンできます。Viewport の上面には、カメラコントロールと情報を持つ小さなオーバーレイがあります — Auto Rotation、トレーニング統計、Reset ボタン。次のステップ (Export) の前に、再構築がクリーンであることを確認するために、モデルをさまざまな角度から確認することをお勧めします。

C-36 SplatViewportView (3D メインビュー)

場所

プレビューステップのフルスクリーン背景。

技術詳細

完成した点群をレンダリングする Metal ベースの 3D Viewport。レンダラーは RadianceKit 独自の ForwardPass ラスタライザー — トレーニング中にすでに Splat をレンダリングしているのとまったく同じもの — であるため、真の WYSIWYG です (トレーニングされたものがそのまま表示され、エクスポートされます)。Order-Independent Transparency を持つタイルベースのレンダリングパイプライン。レンダラーが初期化できない場合 (例えば Metal がシステムで利用できない)、代わりに「Metal not available」テキスト付きの黒い背景が表示されます。ビューは Safe Area を無視するため、モデルはウィンドウ端に達します。

簡単に言うと

メイン Viewport。ここでは、写真から再構築された完成した 3D モデルを、GPU でリアルタイムにレンダリングして見ることができます。左マウスボタンでクリックしてドラッグして回転。スクロールホイールまたは 2 本指のトラックパッドジェスチャーでズーム。右マウスボタンまたは Cmd+ドラッグでパン。モデルは数万の半透明 3D 楕円体 (「Gaussian」) で構成され、シーンをフォトリアルに再構築します — 各々はトレーニングが学習した位置、方向、形状、色を持ちます。Mac が Metal をサポートしないまれな場合、代わりにヒントメッセージ付きの黒い背景が見えます — RadianceKit には Metal 対応 GPU が強制的に必要です。

C-37 CameraControlsOverlay (コントロールオーバーレイ)

場所

Viewport の上、フローティング。



技術詳細

Auto Rotation (Turntable)、Reset Camera、Background 選択 (Gray/Black/White)、Save Screenshot、Toggle Info Panel 用のボタンを持つコンパクト UI オーバーレイ。カメラパラメータ (距離、Azimuth、Elevation、Target、FOV) にバインドし、Auto Turntable を制御します。トレーニング中 (ユーザーが Expert モードで Viewport が実行されるのを見たい場合)、オーバーレイはさらにコンパクトなトレーニングステータス行を表示します。

簡単に言うと

モデル上の小さなフローティングバー。ここで **Auto Rotation** を開始 (モデルは自動的に回転、スクリーンショットと短いデモに適しています)、**Reset** でカメラを開始位置にリセット (迷子になった場合)、背景を変更 (中立用グレー、最大コントラスト用黒、明るいモデル用白)、**/Pictures** に保存されるスクリーンショットを直接撮影できます。モデル全体をエクスポートせずに、特定の詳細を特定の角度から示したい場合に実用的です。**Auto Rotation** は、モデルがすべての側面から同様に良く見えるか、撮影が欠けているために「汚い側」があるかを確認する良いテストでもあります。

C-38 Export ボタン (ナビゲーションバー)

場所

Z3 の下部ナビゲーションバー。



技術詳細

ラベル「Export」と Share アイコン付き Accent Button。クリックは Z4 への切り替えをトリガーします。事前に、親ビューはフルバージョンがアンロックされているかをチェックします — そうでない場合、Export ステージの代わりにロックビューが表示されず (U-06 参照)。

簡単に言うと

結果に満足したら、**Export** をクリックすると、形式を選択して保存する最後のステップに着地します。購入されたフルバージョンがない場合、代わりに **Unlock** ヒントと購入ボタンを持つ画面ロックに着地します — アプリはフルバージョンを押し付けたくありませんが、**Export** は **Premium** 機能の 1 つです。購入を完了すると、アプリはアンロック状態で直接続行され、お馴染みの **Export** ステージに着地します。考え直した場合、**Back** ボタン経由で再びプレビューに戻り、モデルを回転し続けることができます。

次のステップへいつ?

エクスポート前にモデルを一度完全に回転し、確認します: 入力画像でカバーされたすべての領域が存在するか? 浮遊する「Floater」(空中に浮かぶ自由な Gauss Splat 雲) はあるか? 背景/空はクリーンに見えるか、それともぼやけているか? 重大な問題は、再トレーニングでの

み修正できます — より多くの画像、別のプリセット、または Expert モードでの Floater Reduction 設定で。

Z4 — Export (形式選択 & 保存)

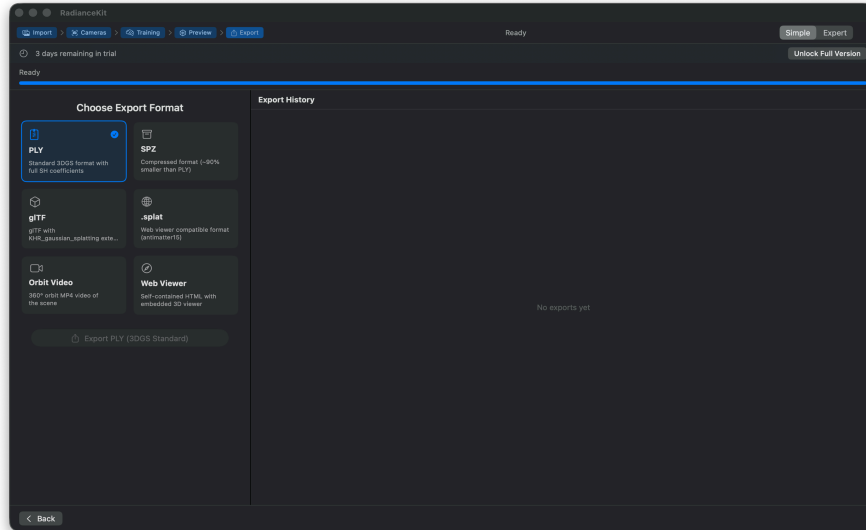


図 40: 形式カード付き Simple モード Export ステップ

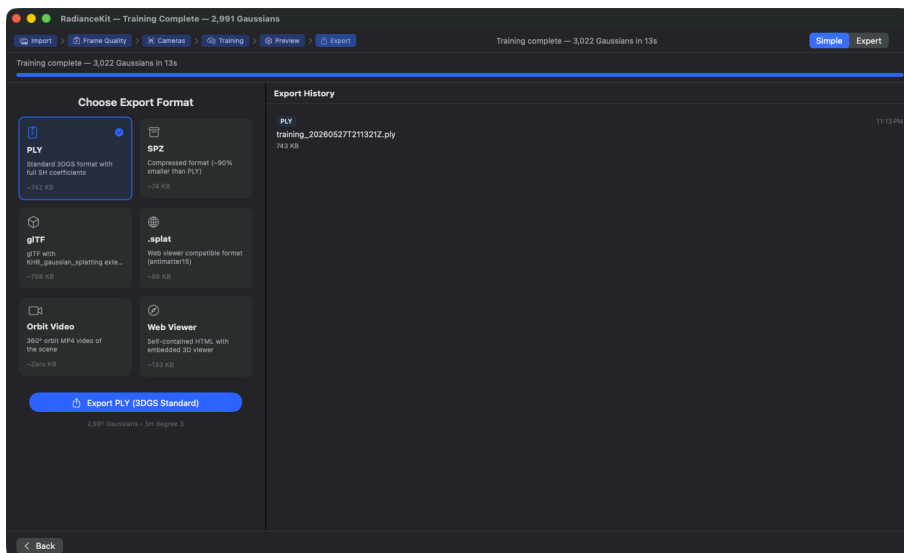


図 41: Z4 Export カード — 6 形式 (PLY 742 KB 選択、SPZ 74 KB、glTF 708 KB、.splat 96 KB、Orbit Video、Web Viewer 133 KB)、右側のすでにエクスポートされた PLY を持つ Export 履歴サイドバー

画像で見えるもの Crumb Trail は「Export」をアクティブステージとしてマークします。左のカードグリッド「Choose Export Format」と、6つのオプションすべて: PLY (標準 3DGS、742 KB、完全な SH 係数 — ここでは青いチェックマークで事前選択)、SPZ (圧縮 3DGS 形式、PLY より ~90% 小さい、74 KB)、glTF (KHR_gaussian_splating Extension 付き、708 KB)、.splat (antimatter15 経由の Web Viewer 互換、96 KB)、Orbit Video (シーンの

360° MP4、ライブ サイズ計算)、Web Viewer (組み込み 3D Viewer 付きの独立した HTML、133 KB)。サイズは現在の Gaussian カウントと形式オーバーヘッドからライブで計算されます。右の「Export History」は、完了したエクスポートを形式ピル、ファイル名、タイムスタンプとともに リストします — クリックで Finder で表示。左下の Primary CTA: 「Export PLY (3DGS Standard)」と Gaussian Subtitle 「2,991 Gaussians · SH degree 3」。最後のステップで、2 カラムカードグリッド経由で 6 のエクスポート形式 (PLY、SPZ、glTF、.splat、Orbit Video、Web Viewer) から選択し、Export をクリックして macOS ダイアログで保存場所を選択します。右側で、これまでのすべてのエクスポートの履歴が実行されます — カード選択で、各カードの下にすぐに推定ファイルサイズが表示されるため、例えば Web に行きたい場合は SPZ を優先 (小さい)、別のソフトウェア (SuperSplat、Postshot、Plugin 経由の Blender) にインポートしたい場合は PLY を優先 (大きく完全) します。

C-39 2 カラム形式グリッド



場所

Export ステップの左メインページ。



技術詳細

2 つの柔軟なカラムと 12 ポイント間隔のカードラスタ。初心者モードで提供される形式を反復処理 — 最も重要な 6 形式のみを含むフル形式リストのフィルタリングされたサブセット: PLY、SPZ、glTF、.splat、Orbit Video、Web Viewer。Compressed PLY と SOG は Expert モードでのみ提供されます。

簡単に言うと

初心者モードで関連する 6 形式を持つカードラスタ: PLY (他の 3D ツール用標準形式)、SPZ (Web 用圧縮バリエーション)、glTF (公式 Web3D 標準)、.splat (antimatter15 Web Viewer 用)、Orbit Video (展示用完成 MP4)、Web Viewer (組み込み 3D プレーヤー付き独立 HTML ファイル)。これにより、ユーザーの 90% をカバーします。あまり一般的でない形式の 1 つが必要な場合 (極端な圧縮用の Compressed PLY または SOG)、Expert モードに切り替えます。ここではすべての 8 形式が利用可能です。初心者が多様性に圧倒されないように、このコンパクトな選択は意図的です。

C-40 Format Card ボタン

場所

グリッド内の各カード。



技術詳細

カードレイアウト付きのシンプルなボタン: アイコン (例えば PLY 用 Document Zipper、SPZ 用 Archive Box、Orbit Video 用 Video アイコン) が上、Format Name が Headline、Description Caption (2 行切り詰め)、下に推定ファイルサイズ (形式、Gaussian カウント、SH Degree からライブ計算され、KB/MB としてフォーマット)。クリックで形式が選択されます。選択されたカードは Accent 背景、Accent Border、右上にチェックマークアイコンを取得します。ツールチップは Format Description です。

簡単に言うと

形式ごとに1つのカード。1つをクリックすると、Accent 色とチェックマークで強調表示され、下の Export ボタンがテキストを調整します (「Export PLY」、「Export SPZ」など)。各カードは、適切なシンボル、名前、2 行の短い説明、現在のトレーニング結果での推定ファイルサイズを表示します。サイズは賢明に選ぶのに役立ちます — 結果をメールで送りたい場合は、最小バリエーションを取ります (通常 SPZ または .splat)。別の 3D ソフトウェアで作業を続けたい場合は、最良の互換性を持つもの (通常 PLY) を取ります。カード上にホバーすると、ツールチップは形式間の違いが不明確な場合に、より詳細な説明を表示します。

C-41 Video Duration スライダー

場所

形式グリッドの下、ビデオ形式が選択されている場合のみ表示 (Orbit Video または Social Video)。



技術詳細

3~30 秒のスライダー、1 秒ステップ、アプリ状態のビデオ長にバインド。最大幅 300 ピクセル。ビデオ形式が選択されている場合のみ表示。非ビデオ形式では、スライダーはビューから完全に削除されます — デッドスペースなし。

簡単に言うと

Orbit ビデオをエクスポートとして選択すると、ここで長さを決定できます。3 秒 = 非常に速い回転、30 秒 = モデル周りの遅く落ち着いた回転。ソーシャルメディア リール (Instagram, TikTok) には、通常 6~10 秒が理想的です — モデルを表示するのに十分長く、視聴者が離れないように十分短い。プレゼンテーションやポートフォリオ ビデオには、15~20 秒が良いでしょう。スライダーはビデオ形式が選択されている場合のみ表示されます。PLY や SPZ のようなファイル形式では意味がなく、非表示になります。

C-42 Export ボタン 場所

形式グリッドの下 (およびビデオが選択されている場合は Duration スライダーの下)。

 技術詳細

大きな Accent Button。ラベル: 「Export {Format-Name}」、Share アイコン。クリックで、形式に適した拡張子と Default ファイル名 「scene.{ext}」を持つ macOS 保存ダイアログが開きます。確認で、エクスポートが選択された URL に書き込まれます。トレーニング結果がない、またはエクスポートがすでに実行中の場合は無効。

 簡単に言うと

クリック、macOS ダイアログで保存場所を選択、完了 — アプリは選択された形式のファイルを選択された場所へ書き込みます。Default 名は 「scene.{Endung}」 (例えば 「scene.ply」 または 「scene.spz」) で、保存前にダイアログで自由に変更できます。ボタンは、トレーニング結果がまだない (Export ステップに到達しないため、ここでは決して発生しません) または別のエクスポートがすでに実行中の限り、グレーです。エクスポートが実行されるとすぐに、その下に進捗表示が表示されます。アプリは操作可能なまま、次のエクスポートをすでに準備できます。

C-43 Export Progress Bar 場所

Export ボタンの下、エクスポートが実行されている間のみ表示。

 技術詳細

最大幅 300 ピクセルの進捗表示、下に Caption 「Exporting... N%」。値は 0 から 1 まで実行され、書き込み中に更新されます — PLY では 10 000 Gaussian のチャンクで、SPZ では量子化後に 1 回、Orbit Video ではフレーム間隔で。

 簡単に言うと

エクスポートが実行されている間、ここで進捗が細かいパーとパーセント表示として見えます。PLY はほとんどの場合数秒以内に完了します。ファイルが単にバイナリで書き出されるためです。SPZ は、データがその過程で量子化および圧縮されるため、やや長くかかります。Orbit Video は最も時間のかかるエクスポートです — ここでは各個別フレームが新しくレンダリングされます。解像度と長さに応じて、1 分以上かかることがあります。エクスポート中、アプリは操作可能なまま、次の形式をすでに準備したり、Viewport で操作を続けたりできます。

C-44 Export Error Display 場所

Progress Bar の下、最後のエクスポートでエラーが発生した場合のみ表示。

 技術詳細

Warning アイコンとエラーテキスト付き赤い行。赤の 8% 背景 Opacity、角丸。最大幅 400 ピクセル。一般的なエラー原因: SOG はシステム PATH に `cwebp` を期待 (App Store 準拠ではない)。フルディスクスペースでの書き込みエラー。許可された領域外の保存ターゲットでの Sandbox エラー。

 簡単に言うと

エクスポートが失敗した場合、ここに赤色で問題の短いクリアテキスト説明が表示されます。ほとんどの場合、原因は明らかです — ディスクにスペースがない、ターゲットフォルダの書き込み権限がない、または Sandbox 許可エリア外のターゲット。特に SOG 形式では、システムに `cwebp` がない場合があります。この場合、SOG は使用できず、SPZ に切り替える必要があります。エラーメッセージが不明確な場合、Log ディレクトリ (Help → Open Training Logs) を見てください。そこに何が間違ったかがより詳細に記載されています。疑わしい場合は、別の保存場所を選択するだけで役立ちます — 例えばデスクトップ。

C-46 Export 履歴リスト 場所

Export ステップの右側。

 技術詳細

Export 履歴のリスト (UserDefaults に JSON として永続的に保存、各成功したエクスポート後に保守)。各行は Format Badge (小さい、Accent 色)、Timestamp (HH:mm)、ファイル名 (1 行切り詰め)、フォーマットされたファイルサイズを表示します。行をクリックすると、選択されたファイルで Finder を開きます。Empty State: 「No exports yet」。

 簡単に言うと

これまでのエクスポートのリスト — 形式、時間、ファイル名、サイズ、時系列順。行をクリックすると、フォルダを自分でナビゲートすることなく、ファイルが Finder で強調表示されます。1 時間後に最後のエクスポートが再度必要で、どこに保存したかわからなくなった場合に実用的 — 履歴はそれを記憶します。まだ何もエクスポートしたことがない場合、ここに親しみやすいヒント「No exports yet」が表示されます。リストは UserDefaults に保存されるため、アプリの再起動を生き残ります。

C-48 History Context Menu (右クリック)

場所

History 行への右クリック。



技術詳細

各リストエントリのコンテキストメニュー、2つのアクション付き: 「Reveal in Finder」(シンプルクリックのように、選択されたファイルで Finder を開く)と「Copy Path」(完全なファイルパスをテキストとしてクリップボードに配置)。後者は、他のアプリへの Drag and Drop やコマンドラインへの転送に便利です。

簡単に言うと

History エントリへの右クリックは、2つのアクションを持つ小さなメニューを開きます。「Reveal in Finder」は通常のクリックと同じことを行います — 選択されたファイルで Finder を開き、すぐに表示できます。「Copy Path」は、Terminal コマンド、他のアプリ、メモなどに貼り付けられるように、完全なファイルパスをクリップボードに配置します。エクスポートを誰かに渡したい場合や、パス入力で動作する別のプログラムで開きたい場合に特に実用的。Mac 風の操作パターンに依存する小さいが役立つ詳細です。

ワークフローはいつ完了したか?

成功したエクスポート後、3D モデルをディスクにファイルとして持ち、履歴は新しいエントリを表示します。「Done」ボタンはありません — 再トレーニングなしで、さまざまな形式で任意の数のエクスポートを追加できます。プレビューに戻りたい場合 (例えば別のカメラパースペクティブを確認するため)、下部ナビゲーションバーの Back ボタンを使用します。完全に新しいシーンを開始したい場合は、Back 経由で Z1 まで行き、そこで Clear All を使用するか、File → New Project (Cmd+⇧+N) を使用します。

Expert モードへの切り替え

いつでも Cmd+2 を押すか、Mode → Expert Mode (M8) を選択します。すべての状態が保持されます: インポートされた画像、選択されたプリセット、実行中または完成したトレーニング、完成した点群、Export 履歴、現在のステージさえも。Expert モードでは、4 ステップステージの代わりに、すべての ~150 操作フィールドを持つフル Inspector サイドバーが表示されます。特に: Project Navigator (第 2 章参照) は拡張画像操作 (マイナスボタン、Backspace 削除、Cmd-Z Undo、Quick Look プレビュー)、トレーニング中の Viewport でのライブプレビュー、すべての Loss、MCMC、Densification、Mip-Splatting パラメータを提供します。Cmd+1 は初心者モードに戻します — それも 状態を失いません。

よくある質問

なぜ Start Processing ボタンがグレーのままですか?

まだ画像もビデオもインポートされていません。少なくとも 1 つのファイルを Drop ゾーンにドラッグするか、「Browse Files」を使用してください。右側の画像リストに少なくとも 1 つのエントリが含まれるとすぐに、ボタンがアクティブになります。(1~2 画像だけでは開始しますが、SfM はエラーで直接中止します — 赤い検証バナー参照。)

なぜ Export ボタンがロックされていますか？

初心者モードには 2 つの段階があります: (a) トレーニングパイプラインがまだ完了していない、または持っていない場合、ボタンは無効化されます — 最初に Z2 を完了する必要があります。(b) フルバージョンをまだ購入していない場合 (`PurchaseManager.hasAccess == false`)、Export ステージの代わりに鍵アイコンと「Unlock Full Version」ボタンを持つロックビューが表示され、Purchase Sheet を開きます。Quick と Preview プリセットは無料でトレーニングを許可しますが、Export は Premium です。

なぜプリセットを選択できないのですか？

選択できます — しかし、購入されたフルバージョンなしで Premium プリセット (Balanced、Quality、MCMC バリエーション) をタップすると、ピッカーが自動的に Preview に戻り、Purchase Sheet が開きます。Quick と Preview は無料で使用可能な唯一のプリセットです。

画像をドラッグしているのに、Drop ゾーンが空で破線グレーのままなのはなぜですか？

おそらく UTI タイプの不一致です。アプリは JPG、PNG、TIFF、HEIC、MP4、MOV、およびアプリ独自の Splat 形式を受け入れます。他の画像形式 (BMP、GIF、WebP、RAW 形式) は NOT 認識されます。画像タイプが含まれるべきだと確信している場合は、ファイル名拡張子を確認してください — アプリは主に拡張子で進み、ファイルコンテンツではありません。

30 画像しかないのに、なぜ SfM がこんなに長くかかるのですか？

Apple Photogrammetry は線形にスケールアップしません — 一部の画像構成 (複雑なテクスチャを持つ室内、ブレ、悪い光) では、画像数が示唆するよりはるかに長くかかります。SfM が 30 画像で 10+ 分後もまだスタックしている場合、中止して、より良い素材で再試行するか、Expert モードに切り替えて COLMAP/Native SfM (`Cmd+2` → Inspector → Camera Alignment) を試してください。

トレーニングログはどこにありますか？

Help → Open Training Logs (`Cmd+⇧+L`)。これは `~/Documents/RadianceKit/Logs/` を開きます。各トレーニングセッションは、ファイル名にタイムスタンプ付きの独自の JSONL ファイルを書き込みます — 最初の行は構成、その後 100 反復ごとに Progress 行、最後の行は Final Loss と Success Flag を持つ Summary です。



奥付

Set in SF Pro · Code in SF Mono · Typst 0.14 ·
2026 年 06 月 22 日

© 2026 Bjoern Kindler · Bischofshofener Str. 9, 82008 Unterhaching, ドイツ

Made with ❤️ in Unterhaching