



RADIANCEKIT

Manual do Usuário

Reconstrução 3D fotorrealista
via Gaussian Splatting

Versão 1.5.0 · macOS 26.0+ · Maio de 2026

BJOERN KINDLER · KINDLER-DEV.DE

Visão Geral

Introdução — O que você deve saber	3
O que é RadianceKit?	3
O que é Gaussian Splatting?	3
Capítulo 1 — Barra de menus	5
Menu File	6
Menu Mode	9
Menu Training	11
Menu Viewport	14
Menu Export	20
Menu Help	25
Aviso: Cmd-Z no menu Edit	29
Visão geral dos atalhos de teclado	30
Capítulo 2 — Inspetor (Expert View)	31
Seção Look (L1–L5)	34
Seção Predefinições (I1–I11)	37
Seção Configuração de treinamento (I12–I22)	43
Seção Enhancements (I26–I29, I42–I44)	49
Seção Métricas (I30–I38)	56
Seção Gráfico de Loss (I39–I41)	61
Quando recorrer ao Inspetor?	64
Capítulo 3 — Configurações	66
Aba General	67
Aba AI Helpers	73
Configurações Espelhadas do Inspetor	76
Quando o quê?	77
Capítulo 4 — Janelas auxiliares	78
User Guide (W1–W4)	79
Keyboard Shortcuts (W5–W6)	82
Manage Storage (W7–W12)	84
Pareto Dashboard (W13–W22)	87
Holdout Analysis (W23–W29)	93
BayesOpt Console (W30–W39)	98
Janela principal: curva de loss e Gaussian count (I39–I41, cross-reference)	105
Caixa de regras práticas	106
Capítulo 6 — Configuração de treinamento	107
Iteração (T1–T2)	109
Learning rates (T3–T10)	111

Densification — Classic (T11–T16)	116
Loss (T17–T20)	119
Progressão de SH degree (T21)	122
Performance (T22–T25)	123
Diagnóstico e preparação da nuvem de pontos (T26–T30)	125
Regularização (T31–T37)	127
Refinement (T38–T44)	131
Sky Dome (T45–T48)	134
Adam + LR Schedule (T49–T55)	136
Pós-processamento + Apple AI (T56–T60)	140
MCMC Densification (T61–T73)	142
Mip-Splatting (Q1.5) (T74–T76)	148
Adaptive Densification (Q5) (T77–T79)	150
Curriculum (Q6) (T80–T81)	152
Predefinições estáticas (TP1–TP9)	152
Método:	155
Qual campo para quê? (cheat sheet)	156
Campos perigosos	157
Capítulo 7 — Predefinições de qualidade embutidas	158
Qual predefinição, quando?	168
Comparação rápida	169
Predefinições próprias	171
Capítulo 8 — Formatos de exportação	172
Qual formato, quando?	186
Comparação rápida	187
Capítulo 9 — Backends de SfM	188
Qual backend, quando?	194
Comparação rápida	195
Capítulo 10 — Modo Iniciante	196
Z1 — Import (imagens & predefinição)	196
Z2 — Processamento (SfM + treinamento)	204
Z3 — Pré-visualização (girar o modelo 3D)	209
Z4 — Export (escolher formato & salvar)	212
Mudança para o Expert Mode	217
Perguntas frequentes	217

Como ler este manual

Cada entrada deste manual segue o mesmo esquema. Do lado esquerdo você encontra os caminhos de operação e os detalhes técnicos; à direita, em uma barra lateral em tom quente, está sempre a explicação em linguagem simples. Pequenos ícones no início de cada linha indicam, num relance, que tipo de informação vem a seguir.

OS QUATRO ÍCONES



Onde encontro isso? O caminho concreto de cliques pelo app — barra de menus, seção do inspetor ou passo do Modo Iniciante. Os atalhos de teclado correspondentes também aparecem aqui. O ícone é um pino de mapa e indica onde a funcionalidade está localizada na interface.



Detalhes. Valores padrão, faixas de valores e caminhos de código. Você encontra isso principalmente nas configurações de treinamento, que não são itens de menu, mas parâmetros numéricos. O ícone mostra uma pequena ficha de especificações.



Técnico. O que a funcionalidade faz internamente, quais parâmetros atuam, ao que ela reage e quais efeitos colaterais possui. Para leitores que querem entender o que acontece nos bastidores. O ícone é um bloco de controles deslizantes e representa simbolicamente os ajustes sob o capô.



Em palavras simples. A mensagem central em palavras claras — sem jargão, sem código. Leia esta seção primeiro se quer apenas saber rapidamente para que serve uma função e quando usá-la. O ícone é um balão de fala e significa “em resumo”. Esta coluna é sempre exibida em um tom de areia quente para que o olho a encontre imediatamente.

CORES DOS CAPÍTULOS

Cada capítulo tem sua própria cor de destaque, que você reconhece pela etiqueta de ID (por exemplo **M1**) à esquerda de cada título de entrada e pelos pequenos ícones à frente deles. Ao folhear, você vê imediatamente em qual capítulo está.

- | | | | | |
|------------------------|----------------------|------------------------|-----------------------------|----------------------|
| 1 Menus | 2 Inspetor | 3 Configurações | 4 Janelas auxiliares | 6 Treinamento |
| 7 Predefinições | 8 Exportações | 9 SfM | 10 Modo iniciante | |

DICAS DE NAVEGAÇÃO

Início rápido. Se você só se interessa pela operação, pule direto para o **Capítulo 10 — Modo Iniciante**. Essa é a variante guiada em quatro passos e não exige nenhum conhecimento prévio.

Aprofundamento. O **Capítulo 2 — Inspetor** e o **Capítulo 7 — Predefinições** explicam os controles e os perfis de qualidade pré-configurados disponíveis no Modo Especialista.

Consulta. O sumário e a busca de texto completo do PDF ajudam a encontrar uma funcionalidade específica. Você não precisa ler o manual do começo ao fim.

Introdução — O que você deve saber

O que é RadianceKit?

RadianceKit é um app nativo para macOS que transforma uma série de fotos comuns ou um vídeo em uma reconstrução 3D explorável. A entrada são, por exemplo, 50 a 500 imagens que você fez ao redor de um objeto, atravessando um ambiente ou percorrendo uma paisagem. A saída é uma chamada cena de Gaussian Splatting — um modelo 3D que você pode visualizar em tempo real no Mac a partir de qualquer perspectiva, que pode ser exportado e embutido em páginas da web e que, em seus principais aspectos, parece fotorrealista.

O app roda inteiramente local no seu Mac — nenhuma imagem é enviada para a nuvem, nenhum login é exigido, sem assinatura. Ele faz uso intensivo da GPU do seu Mac com Apple Silicon (série M): um treinamento completo pode levar de dois minutos a várias horas, dependendo da cena e da predefinição. Enquanto a computação acontece, você pode continuar trabalhando normalmente no Mac; o RadianceKit continua rodando em segundo plano e avisa quando o resultado estiver pronto.

Existem dois modos de operação: o *Modo Iniciante* (Simple Mode) guia você em quatro passos pelo fluxo Importar → Escolher Predefinição → Treinar → Exportar. O *Modo Especialista* (Expert Mode) abre um grande inspetor com todos os controles, uma janela de pré-visualização ao vivo e gráficos de diagnóstico. Você pode alternar entre os modos a qualquer momento; os dados da cena são preservados.

O que é Gaussian Splatting?

Gaussian Splatting (frequentemente abreviado como 3DGS ou simplesmente *Splatting*) é um método relativamente novo para representação 3D fotorrealista, apresentado em 2023 num artigo de Graz e do INRIA. A ideia: em vez de modelar uma cena como uma malha clássica de polígonos (triângulos) ou uma grade de voxels, ela é composta por milhões de pequenas e suaves nuvens 3D — cada nuvem individual é uma distribuição gaussiana 3D (daí o nome) com sua própria posição, tamanho, forma, cor e transparência. Essas nuvens são treinadas de modo que, em conjunto e de todos os ângulos de visão das suas fotos de entrada, resultem na imagem correta.

Na prática, isso significa que o Gaussian Splatting consegue representar reflexos, brilhos especulares, folhagens suaves, cabelos ou cortinas de um jeito que a modelagem 3D clássica não consegue — ou só consegue com esforço imenso. Em troca, o resultado não é um modelo 3D editável no sentido clássico — você não pode simplesmente mover uma parede ou reposicionar um vaso. É mais como uma *captura congelada* do espaço pela qual você pode se mover livremente. Para muitas aplicações — visualização arquitetônica, apresentação de produtos, tours virtuais, perícia, patrimônio cultural — essa é exatamente a força certa.

São necessários dois passos para transformar as imagens de entrada em uma cena 3D. Primeiro, o app calcula, por meio de um processo chamado *Structure-from-Motion (SfM)*, onde sua câmera estava posicionada em cada foto. Como subproduto, isso gera uma nuvem de pontos grosseira da cena. Depois começa o treinamento propriamente dito de Gaussian Splatting: partindo dessa nuvem grosseira, os milhões de nuvens 3D são gradualmente distribuídas, aumentadas, refinadas e reajustadas em posição e cor até produzirem a imagem adequada de todos os ângulos de entrada.

Você não precisa saber nada disso para usar o RadianceKit. O Modo Iniciante esconde esses passos por completo. Mas se você quer entender o que significam os números de diagnóstico no Modo Especialista (iteração, loss, Gaussians, SSIM ...) ou por que algumas cenas saem mais bonitas que outras, os capítulos seguintes do manual trazem as respostas.

CAPÍTULO

Capítulo 1 — Barra de menus

A barra de menus do RadianceKit organiza todas as funções que não estão diretamente na janela principal ou no Inspetor. Em primeiro lugar, são ações que atuam sobre toda a cena (abrir, salvar, novo projeto), controlam o treinamento (iniciar, pausar, continuar), operam o viewport (rotação automática, captura de tela, cor de fundo) e disparam exportações para vários formatos 3D e de mídia. Somam-se ainda pontos de salto para todas as janelas auxiliares (User Guide, Pareto Dashboard, Holdout Analysis, BayesOpt Console).

Os atalhos de teclado aparecem sempre à direita do item de menu. Convenções: `⌘` significa a tecla Command (Apple), `⇧` é Shift, `⌥` é Option (Alt) e `⌘` é Control. Exemplo: `⇧⌘T` significa Shift+Command+T. Todos os atalhos documentados aqui estão também listados, para consulta, em uma janela própria via `Help → Keyboard Shortcuts (⌘/)`.

As 42 entradas a seguir estão documentadas na ordem do inventário (M1–M42), agrupadas pelo menu de nível superior correspondente. Todas as entradas foram verificadas contra o estado atual do código em (linhas 175–477). Nenhuma entrada foi removida nem superada em relação ao inventário; uma nova entrada do menu Edit (Cmd-Z para „Remove Image“) é fornecida pelo framework NSUndoManager do sistema e por isso não aparece no código RadianceKitApp (ver aviso ao fim do capítulo).

Menu File

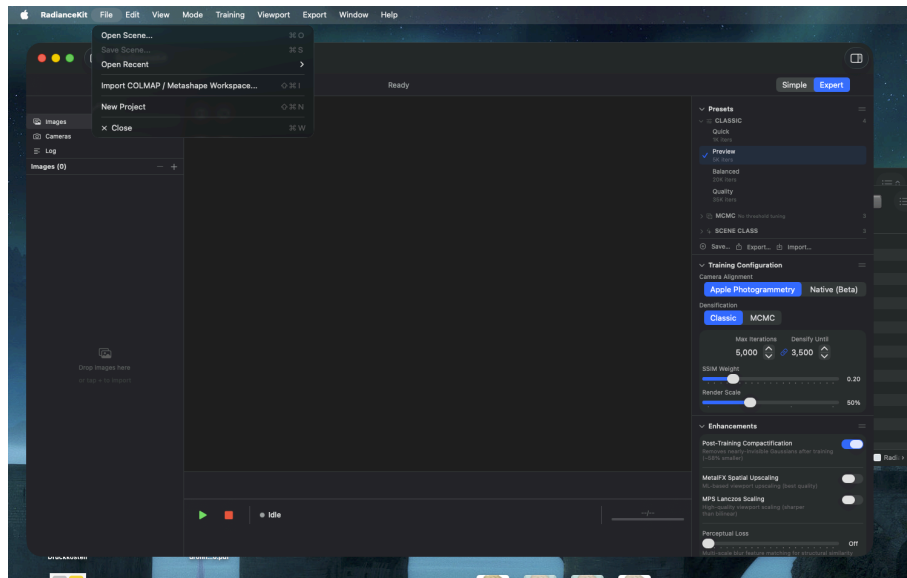


Figura 1: Menu File aberto — entradas M1 a M6

O menu File substitui o padrão „New Window“ da Apple por ações específicas do projeto. Inclui carregar/salvar cenas, uma lista dinâmica de recentes, o import de workspace e o reset duro para um estado vazio.

M1 File > Open Scene...



Barra de menus → File → Open Scene... (⌘O).



Abre um diálogo de arquivo para os formatos RadianceScene -bundle, `.ply`, `.splat` e `.spz`. Single selection, pode mostrar arquivos e diretórios (por causa do formato em bundle). Após a seleção bem-sucedida, o caminho é adicionado à lista de recentes e a cena é carregada de forma assíncrona — a anterior é substituída e o pipeline de treinamento é inicializado com o estado carregado. Arquivos PLY/SPZ/Splat são lidos pelos respectivos loaders de formato; o bundle `.radiancescene` é um diretório com manifest, snapshot de cloud e resultados de SfM.

EM PALAVRAS SIMPLES

É assim que você carrega no app uma cena já treinada. Funciona com o formato próprio do RadianceKit e com os formatos padrão PLY, SPLAT e SPZ que outros programas de splatting geram. Use isso, p. ex., quando treinou uma cena durante a noite e quer continuar ou exportar no dia seguinte. Ao abrir, o estado atual da janela principal é substituído — então salve antes, se a cena atual ainda for importante. O caminho vai automaticamente para „Open Recent“ (M3), para facilitar da próxima vez.

M2 File > Save Scene...

Barra de menus → File → Save Scene... (#S).



Abre um diálogo de salvar com o content type `RadianceScene -bundle` e nome de arquivo pré-preenchido `scene.radiancescene`. Grava um pacote de diretório com `manifest.json`, a nuvem de Gaussians serializada (snapshot PLY) e um dump do resultado de SfM, de modo que ao reabrir também o Continue Training funciona. A entrada fica desabilitada enquanto ainda não houver Gaussians. Não salva no caminho de logs de treinamento, mas onde o diálogo de salvar aponta — tipicamente em `~/Documents/`.

 EM PALAVRAS SIMPLES

Salva a cena atual como arquivo (mais precisamente: como pasta-pacote que parece um arquivo). Só depois disso você pode abrir essa cena mais tarde via „Open Scene...” (M1). No pacote vão tanto a nuvem Gaussian quanto o resultado de SfM, de modo que você pode usar Continue Training (M12–M14) mais tarde. Enquanto não houver treinamento concluído, a entrada fica cinza. O nome padrão é `scene.radiancescene` — mas você pode dar um nome próprio no diálogo de salvar.

M3 File > Open Recent > [nomes de cenas]

Barra de menus → File → Open Recent → (lista).



Submenu dinâmico gerado a partir de uma lista dos caminhos abertos mais recentemente (gravada nas configurações). Cada item recebe o nome do arquivo e é carregado ao clicar. Quando a lista está vazia, aparece em seu lugar o rótulo desabilitado „No Recent Scenes”. Como é típico na Apple, a lista mantém as N cenas mais recentes — o limite é aplicado na gravação nas configurações, não no construtor do menu.

 EM PALAVRAS SIMPLES

Aqui você vê as cenas abertas mais recentemente e pode retornar com um clique, sem passar pelo diálogo de arquivo. Se você acabou de começar, a lista está vazia e aparece cinza no menu. Toda cena aberta pelo „Open Scene...” (M1) entra automaticamente nessa lista. Se em algum momento ela ficar cheia demais ou você quiser limpar por privacidade, use „Clear Recent” (M4).

M4 File > Open Recent > Clear Recent

ONDE

Barra de menus → File → Open Recent → Clear Recent.



TÉCNICO

Esvazia a lista de recentes nas configurações. Atua imediatamente, sem diálogo de confirmação. A entrada só aparece no submenu quando há de fato itens na lista de recentes (fica abaixo de um divisor depois dos caminhos).

EM PALAVRAS SIMPLES

Apaga a lista de cenas abertas recentemente. Prático quando você brincou com um dataset de teste e não quer mais ver os caminhos. Os arquivos das cenas em si não são apagados — só a referência no menu. A ação atua imediatamente, sem perguntar; depois aparece „No Recent Scenes“ no submenu. A entrada só existe quando há cenas na lista — com lista vazia, fica invisível.

M5 File > Import COLMAP / Metashape Workspace...

ONDE

Barra de menus → File → Import COLMAP / Metashape Workspace... (⇧⌘I).



TÉCNICO

Abre um seletor de pasta. Espera uma pasta com o layout de workspace COLMAP (p. ex. `sparse/0/cameras.{bin,txt}` mais `images/`). Após a seleção é feita uma pré-verificação do workspace — ela detecta os três layouts (`sparse/0/`, `sparse/`, `raiz`) e se a reconstruction está em binário (`cameras.bin`) ou em texto ETH3D (`cameras.txt`). Em caso de sucesso o workspace é importado; caso contrário aparece apenas um aviso no log do app. Ver também Capítulo 9 „Backends de SfM“, Q6 para a lógica completa do pipeline.

EM PALAVRAS SIMPLES

Se você usa Metashape, COLMAP, RealityCapture ou software semelhante para a reconstrução de câmeras e tem um export, carrega a pasta aqui. O RadianceKit pula então o estágio SfM e começa direto o treinamento — isso poupa horas em cenas grandes. Arrastar e soltar na janela principal também funciona. Espera-se uma pasta com layout COLMAP (ou seja, `sparse/0/` com `cameras.*` mais a pasta `images/`). Mais sobre os layouts e workflows suportados está no Capítulo 9 „Backends de SfM“.

M6 File > New Project



ONDE

Barra de menus → File → New Project (⇧⌘N).



TÉCNICO

Verifica se há trabalho não salvo. Em caso positivo, aparece um diálogo de confirmação antes que qualquer coisa seja perdida. Quando não há nada a salvar, o reset roda direto — ele esvazia as imagens importadas, o resultado de SfM, a nuvem Gaussian, o estado de treinamento e todos os indicadores de UI dependentes. Atenção: uma biblioteca de predefinições criada pelo usuário é preservada, porque ela fica nas configurações do app e não no estado de projeto.

EM PALAVRAS SIMPLES

Reseta tudo para um começo vazio — como se você tivesse acabado de abrir o app. Se há trabalho não salvo, o app pergunta antes. Use isso quando quiser começar com uma cena completamente diferente. Imagens importadas, resultado de SfM, nuvem Gaussian e estado de treinamento são totalmente esvaziados. Suas predefinições próprias, porém, ficam preservadas, porque estão nas configurações do app e não pertencem à cena.

Menu Mode

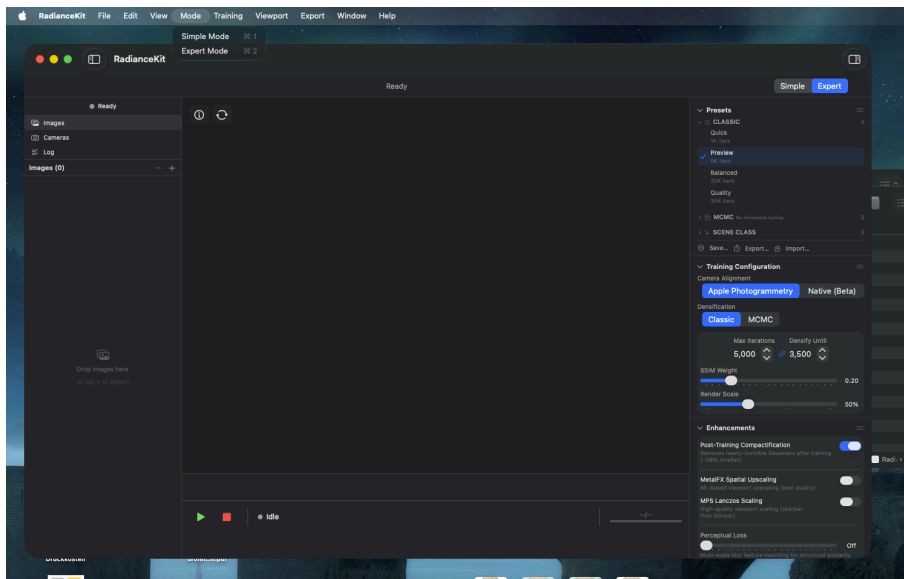


Figura 2: Menu Mode com chaves Simple e Expert Mode

Duas chaves simples entre o Simple Mode guiado (em assistente, 4 passos) e o Expert Mode completo (layout clássico de Inspetor com todos os controles).

M7 Mode > Simple Mode

Barra de menus → Mode → Simple Mode (⌘1).



Alterna o estado do app para Simple Mode. A área principal do app mostra então o fluxo guiado em vez do layout de Expert. O estado de modo é gravado nas configurações (ver S1 „Default Mode“ no Capítulo 3 Configurações).

EM PALAVRAS SIMPLES

Alterna para a variante passo a passo, em que o app conduz você por importar, processar, pré-visualizar e exportar. Recomendado se você está começando ou precisa de um resultado rápido. A maioria dos controles de detalhe fica oculta — você trabalha com predefinições sensatas. Se quiser depois aprofundar, basta mudar para o Expert Mode (M8). Qual modo está ativo ao abrir o app você define nas configurações (Capítulo 3, S1).

M8 Mode > Expert Mode

Barra de menus → Mode → Expert Mode (⌘2).



Alterna o estado do app para Expert Mode. Aparece então o layout completo de Inspetor com todas as seções (Presets, TrainingConfig, Enhancements, Metrics, LossChart, ProjectNavigator). No Expert Mode todos os parâmetros de treinamento, seletor COLMAP, toggles de mid-compact e diagnostics ficam acessíveis. O live preview também funciona somente neste modo.

EM PALAVRAS SIMPLES

Alterna para a visão completa com todos os controles. Aqui você vê gráficos de loss em tempo real, pode ajustar fino todos os parâmetros e gerenciar várias configurações em paralelo via predefinições. Recomendado se quer entender o que o treinamento faz internamente ou experimentar de forma direcionada. O live preview, o seletor COLMAP e os diagnostics também só são acessíveis aqui. Se se sentir sobrecarregado, volte ao Simple Mode via M7 — sua cena fica preservada.

Menu Training

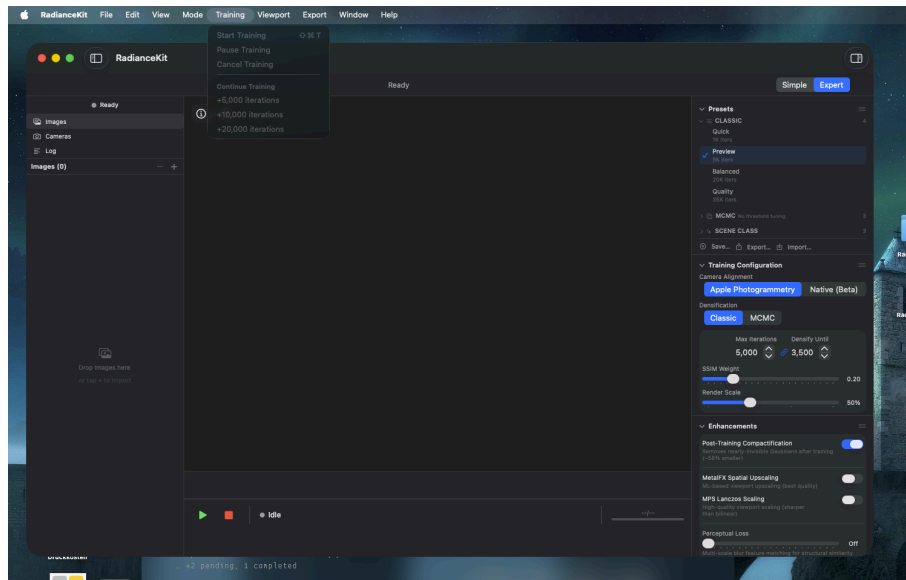


Figura 3: Menu Training com submenu Continue — entradas M9 a M14

Quatro ações em torno da corrida de treinamento: iniciar, pausar, cancelar e estender por um número de iterações pré-definido. Todas as três entradas Continue são gatedas por IAP (na versão Free Trial não são clicáveis).

M9 Training > Start Training

ONDE

Barra de menus → Training → Start Training (⇧⌘T).

TÉCNICO

Inicia o pipeline de treinamento de forma assíncrona. Requisitos: existe resultado de SfM e nenhum outro pipeline está rodando. As duas condições bloqueiam a entrada se não forem cumpridas. No início são lidos os valores atuais de configuração, é criado um novo log JSONL em `~/Documents/RadianceKit/Logs/training_YYYY-MM-DD_HHmss.jsonl` e, conforme a escolha de estratégia, é seguido o caminho clássico ou o caminho MCMC. O estado de treinamento passa de „idle“ a „training“.

EM PALAVRAS SIMPLES

Aperta o botão verde grande — assim que você importou fotos e a reconstrução de câmeras terminou, começa aqui o treinamento Gaussian Splatting propriamente dito. Deixe o app rodar; conforme a predefinição, entre 1 minuto (Quick) e várias horas (MCMC Quality). A entrada fica cinza enquanto não há resultado de SfM ou enquanto outro pipeline está rodando. Cada corrida grava paralelamente um log em `~/Documents/RadianceKit/Logs/`, que você pode analisar depois pelo Pareto Dashboard (M40).

M10 Training > Pause Training

Barra de menus → Training → Pause Training.



Pausa o treinamento em execução. Só é liberado quando o estado de treinamento é „training“. Pausar interrompe o loop de iterações no próximo ponto seguro de sync, mantém todo o estado da GPU (buffers de Gaussian, momentos do otimizador, posição do scheduler) e passa para „paused“. O resume acontece com novo clique (o título da entrada é estático — o app alterna entre pausar/continuar na própria lógica). Treinamentos pausados não sobrevivem a um quit do app; nesse caso, salve a cena e estenda mais tarde via Continue Training (M12–M14).

EM PALAVRAS SIMPLES

Pausa o treinamento brevemente, sem perder o progresso. Prático quando você precisa do computador para algo mais urgente. Clicar de novo continua. Não funciona entre reinícios do app — se quiser de fato continuar depois, cancele o treinamento com Cancel (M11), salve a cena com Save Scene (M2) e use depois Continue Training (M12–M14). Durante a pausa, a GPU descansa por completo; a memória continua ocupada.

M11 Training > Cancel Training

Barra de menus → Training → Cancel Training.



Cancela o treinamento em execução. Ativo quando o estado de treinamento não é „idle“. Seta o flag de cancel na engine de treinamento, o que encerra o loop de iterações no próximo ponto de sync de forma limpa, grava o summary final no log JSONL e retorna o estado para „idle“. A nuvem treinada até então permanece (pode ser salva ou exportada), mas é marcada como „cancelled“.

EM PALAVRAS SIMPLES

Cancela o treinamento em execução definitivamente. O estado anterior permanece — então, se você já tem um resultado apresentável depois de alguns milhares de iterações, pode exportar mesmo assim. Se for só interromper um pouco, use Pause (M10). No log o run é marcado como „cancelled“, mas o valor final de loss é gravado mesmo assim. Uma cena cancelada também pode ser estendida depois via Continue Training (M12–M14), desde que o app não tenha sido encerrado no meio.

M12 Training > Continue Training > +5 000 iterations ONDE

Barra de menus → Training → Continue Training → +5,000 iterations.

 TÉCNICO

Continua o treinamento por 5 000 iterações. Ativo quando há treinamento concluído continuável e a versão completa está desbloqueada. A continuabilidade vale quando existe um treinamento concluído e o estado completo do otimizador ainda está em memória. No continue os momentos Adam e o LR scheduler são prosseguidos, de modo que a continuação se comporta como uma corrida contínua de 25K/45K/60K em vez de um restart. O log JSONL recebe uma nova entrada de config com o setup incremental. Só disponível na versão completa.

 EM PALAVRAS SIMPLES

Anexa mais 5 000 passos de treinamento. Use quando o resultado depois da primeira corrida está perto, mas ainda não totalmente nítido. Funciona só na versão completa paga. Diferente de um run completamente novo, o estado do otimizador é preservado, de modo que a continuação se sente como uma corrida contínua. Se precisar de mais de 5 000 passos, vá direto para M13 (+10 000) ou M14 (+20 000).

M13 Training > Continue Training > +10 000 iterations ONDE

Barra de menus → Training → Continue Training → +10,000 iterations.

 TÉCNICO

Idêntico ao M12, mas com 10 000 iterações adicionais. Mesmas pré-condições, mesmo caminho de LR scheduler. Recomendado quando o treinamento inicial foi feito com uma predefinição mid-tier e você quer ver um ganho significativo de qualidade sem reiniciar totalmente o run.

 EM PALAVRAS SIMPLES

Estende o treinamento por 10 000 passos — o valor intermediário dos três disponíveis. Boa escolha quando a primeira corrida foi ok mas você quer ficar claramente melhor. Como M12 e M14, a curva de taxa de aprendizado é continuada sem emendas, em vez de reiniciar. Disponível só na versão completa.

M14 Training > Continue Training > +20 000 iterations

ONDE

Barra de menus → Training → Continue Training → +20,000 iterations.



TÉCNICO

Idêntico ao M12/M13, mas com 20 000 iterações adicionais. O maior salto Continue pré-definido. Em treinamentos MCMC isso costuma ser o que faz a diferença entre „passa“ e „padrão de benchmark“; em Classic, a partir de 35–40K, na experiência rende pouco.

EM PALAVRAS SIMPLES

Anexa 20 000 passos de treinamento, o valor máximo. Use quando realmente quer extrair a última gota de qualidade. Em Classic, após 40 000 passos, em geral não rende muito — em MCMC, ao contrário, costuma valer a pena, porque ali a convergência demora mais. Conte com tempo a mais conforme a cena. Como M12 e M13, essa entrada também só está na versão completa.

Menu Viewport

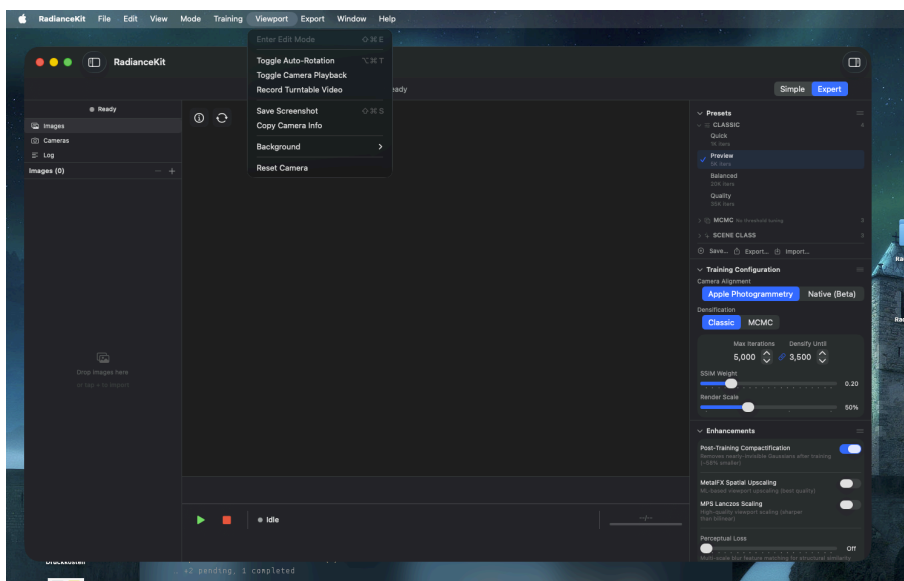


Figura 4: Menu Viewport com Edit Mode, controle de câmera e submenu de fundo

Controla o viewport 3D: Edit Mode para seleção e limpeza de Gaussians, controle de câmera (rotação automática, playback, recording), captura de tela, cor de fundo e reset.

M15 Viewport > Enter/Exit Edit Mode

Barra de menus → Viewport → Enter Edit Mode (ou „Exit Edit Mode“, conforme o estado). ⌘⌘E.



O título da entrada é dinâmico e mostra, conforme o estado, „Exit Edit Mode“ ou „Enter Edit Mode“. Ao pressionar, o Edit Mode é alternado no renderer do viewport. Ao sair do Edit Mode, a seleção atual é resetada. O Edit Mode ativa a seleção por clique de Gaussians, a seleção por caixa e a remoção dos Gaussians marcados (ver área de Editor da UI). Desabilitado enquanto não há renderer do viewport conectado.

EM PALAVRAS SIMPLES

Alterna entre a visão 3D normal e um modo de edição em que você pode marcar e remover Gaussians individuais (p. ex. floaters ou outliers no fundo). Ao sair, a seleção é resetada automaticamente. A entrada fica cinza enquanto não há cena visível no viewport. O rótulo alterna entre „Enter Edit Mode“ e „Exit Edit Mode“ — assim você sempre sabe em que modo está.

M16 Viewport > Toggle Auto-Rotation

Barra de menus → Viewport → Toggle Auto-Rotation (⌘⌘T).



Liga ou desliga a rotação contínua da câmera do viewport em torno de um eixo vertical pelo centro da cena. Eixo e velocidade vêm da configuração de controle de câmera. A rotação automática é um efeito puro de viewport e não influencia treinamento nem recording — mas, se você usa em paralelo o recorder de vídeo turntable (M18), a rotação automática entrega exatamente o caminho que o recorder captura.

EM PALAVRAS SIMPLES

Gira a câmera continuamente devagar em torno da sua cena, para você ver de todos os lados sem arrastar com o mouse. Clicar de novo para. Prático para avaliar cenas já treinadas ou como animação de fundo numa demo ao vivo. Se você grava um vídeo em paralelo (M18), a rotação automática fornece exatamente o movimento que o recorder captura.

M17 Viewport > Toggle Camera Playback

Barra de menus → Viewport → Toggle Camera Playback.



Liga/desliga o playback de caminho de câmera. Quando existe um caminho gravado (p. ex. de uma gravação anterior ou porque um `transforms.json` foi carregado), o caminho roda — a câmera do viewport não se move mais por entradas de mouse/trackpad, mas reproduz a trajetória frame a frame. Novo clique pausa o playback.

EM PALAVRAS SIMPLES

Reproduz uma trajetória de câmera gravada ou importada anteriormente. Assim você refaz o caminho original usado na captura da cena ou verifica um movimento orbital planejado antes do export em vídeo. Enquanto o playback roda, entradas de mouse e trackpad ficam desativadas — a câmera segue rigorosamente o caminho. Clicar de novo pausa. Se você não carregou nem gravou um caminho, nada acontece.

M18 Viewport > Record Turntable Video

Barra de menus → Viewport → Record Turntable Video.



Liga/desliga a gravação do viewport. No primeiro clique inicia uma captura de frames para um caminho temporário; no segundo clique a gravação termina, é codificada e gravada num caminho MP4 (caminho pedido por um diálogo de salvar). Diferente de Export → Media → Orbit Video (M31), que gera um caminho 360° fixo com duração ajustável, o recorder turntable grava *ao vivo* o que você vê no viewport — também é possível gravar uma trajetória manual.

EM PALAVRAS SIMPLES

Grava um vídeo direto no viewport. Não importa se a câmera gira automaticamente ou se você a move com o mouse — tudo que você vê vai para um arquivo MP4. Diferente do export „Orbit Video“ (M31), aqui você define a trajetória. O primeiro clique inicia a gravação, o segundo finaliza e pergunta onde salvar. Prático quando você quer mostrar, p. ex., um certo panorâmico de detalhe que o movimento orbital fixo não permitiria.

M19 Viewport > Save Screenshot

Barra de menus → Viewport → Save Screenshot (⇧⌘S).



Captura um único frame do viewport em resolução plena de render (ou seja, não o layout de pixels da janela, mas o conteúdo completo do alvo de render) como arquivo PNG. O caminho é pedido via diálogo de salvar. A cor de fundo (M21–M23) é embutida. Configurações de upscaling MetalFX/MPS dos Enhancements (ver I27/I28) atuam quando ativas — a captura mostra o output upscaled.

EM PALAVRAS SIMPLES

Salva um snapshot da sua visão 3D atual como imagem PNG. Prático para material de marketing ou uma comparação rápida. Atenção: o fundo faz parte da imagem — se você precisa de transparência, prefira exportar um arquivo de cena. A resolução corresponde ao alvo interno de render, não ao tamanho da janela — a imagem geralmente é mais nítida do que aparenta na janela. Eventuais configurações de upscaling (Inspetor → Enhancements) também entram.

M20 Viewport > Copy Camera Info

Barra de menus → Viewport → Copy Camera Info.



Lê a pose atual da câmera do viewport (posição, ponto de look-at, vetor up) e os valores de FOV do controle de câmera e os grava como texto multilinha na área de transferência. O formato é legível por humanos (label = value por linha), não JSON. Útil para reproduzir uma visão específica em depuração ou compartilhar com o suporte.

EM PALAVRAS SIMPLES

Copia a posição e a direção atuais da câmera como texto para a área de transferência. Se você quer, p. ex., mostrar a um colega de onde uma parte da cena parece estranha, basta colar o texto num e-mail ou janela de chat. O formato é legível (uma linha por valor), sem JSON. Pensado principalmente para bug reports ou suporte.

M21 Viewport > Background > Dark Gray

Barra de menus → Viewport → Background → Dark Gray.



Define a cor de fundo do viewport como cinza escuro (RGB 0.1/0.1/0.1). O renderer usa essa cor como fundo, sobre a qual os Gaussians são compostos. A cor padrão no início do app é controlada pela opção S3 „Default Viewport Background“ em Configurações.

EM PALAVRAS SIMPLES

Pinta o fundo do viewport 3D de cinza escuro. Escolha padrão para a maioria das cenas — oferece bom contraste com Gaussians claros e escuros sem que o olho se prenda a uma superfície totalmente preta ou branca. A cor também vai para os screenshots (M19) e orbit vídeos (M31). Se cinza escuro lhe parecer sem graça, experimente Black (M22) ou White (M23). Qual cor está ativa ao abrir o app você define em Configurações (S3).

M22 Viewport > Background > Black

Barra de menus → Viewport → Background → Black.



Define a cor de fundo do viewport como preto puro (RGB 0/0/0). Ajuda se a cena tem muitos floaters claros e você quer identificá-los ou para material de marketing com look dark.

EM PALAVRAS SIMPLES

Fundo preto. Bom para cenas muito claras ou quando você quer entrar no Edit Mode procurando Gaussians claros pequenos (floaters) que se perdem no cinza. Também ideal para material de marketing com visual escuro e dramático. A cor vai para screenshots e orbit vídeos — se precisa de transparência para um composite posterior, preto é a pior escolha. Para floaters escuros, vá para o lado oposto, White (M23).

M23 Viewport > Background > White

Barra de menus → Viewport → Background → White.



Define a cor de fundo do viewport como branco puro (RGB 1/1/1). Útil quando a cena é predominantemente escura e você quer ver floaters escuros (ruído típico de fundo em externos).

 EM PALAVRAS SIMPLES

Fundo branco. Prático quando o motivo aparece melhor claro-sobre-escuro ou para encontrar outliers escuros que você depois remove no Edit Mode (M15). Em cenas externas, branco costuma ser mais útil que preto, porque os floaters externos típicos tendem a ser escuros. Como nas outras opções de fundo, a cor vai para screenshots e vídeos.

M24 Viewport > Reset Camera

Barra de menus → Viewport → Reset Camera.



Reseta a câmera do viewport, sai da visão Training Camera e para a rotação automática. Assim a câmera volta para a posição inicial (tipicamente: diante da cena, olhando ligeiramente de cima), a rotação automática fica desligada e, se o renderer estava mostrando a Training Camera (uma das poses do SfM), volta para a Free Camera.

 EM PALAVRAS SIMPLES

Devolve a câmera do viewport à posição inicial. Se você se perdeu girando ou empurrou a cena para fora da imagem — um clique aqui e você vê de novo o que deveria ver. Desliga ao mesmo tempo a rotação automática, se estiver rodando, e sai de uma Training Camera congelada para a visão livre. Assim você obtém em qualquer caso um restart limpo da visão.

Menu Export

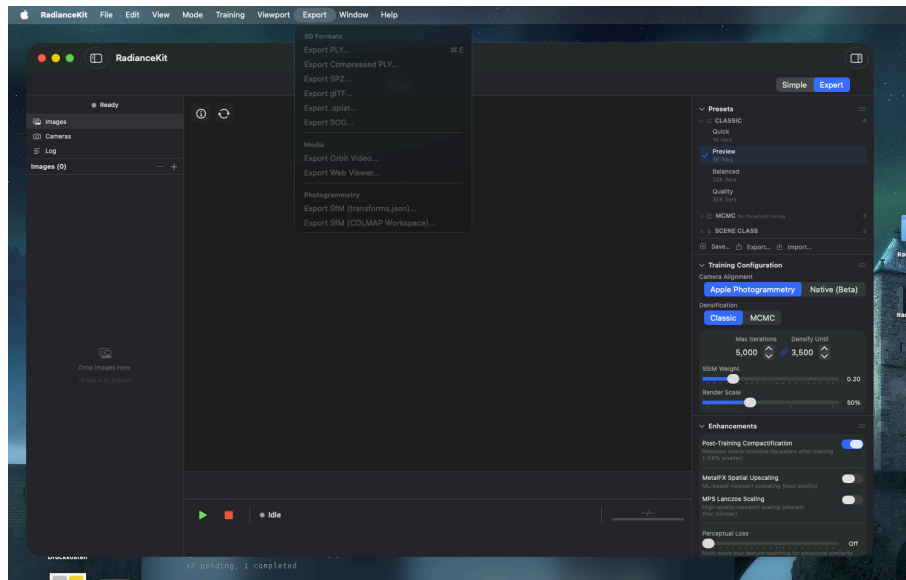


Figura 5: Menu Export com três grupos de submenu — 3D Formats, Media e Photogrammetry

Oito destinos de exportação mais dois exports de fotogrametria, agrupados em três seções (3D Formats, Media, Photogrammetry). Os primeiros seis são construídos por uma rotina helper comum, que abre um diálogo de salvar e registra a exportação no catálogo de formatos. As entradas Photogrammetry têm lógica individual. Todas as exportações Photogrammetry e algumas 3D só estão disponíveis na versão completa.

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



Barra de menus → Export → 3D Formats → PLY (⌘E).



Abre um diálogo de salvar com nome padrão `gaussians.ply`. Em OK, a nuvem Gaussian atual é gravada no formato PLY ASCII/Binary padronizado — compatível com SuperSplat, PolyCam, PlayCanvas e todos os viewers 3DGS comuns. Coeficientes SH completos, precisão completa (Float32 por campo). Tamanho de arquivo frequentemente várias centenas de MB com $\geq 500K$ Gaussians.

EM PALAVRAS SIMPLES

Salva sua cena 3D como arquivo PLY padrão. É o formato mais universal — quase todo software lê, do SuperSplat ao PolyCam e PlayCanvas. Mas os arquivos ficam grandes, muitas vezes centenas de megabytes. Use PLY quando quiser continuar em qualidade total ou arquivar. Se quer compartilhar a cena via web, veja antes SPZ (M27) ou Compressed PLY (M26) — são bem menores.

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

Barra de menus → Export → 3D Formats → Compressed PLY.

 TÉCNICO

Grava a nuvem Gaussian no formato Compressed PLY com quantização customizada de posição, escala, rotação e campos SH. Arquivos 5–10× menores que o PLY não comprimido (M25) com perdas visuais mínimas. Compatível com SuperSplat (que lê o padrão Compressed PLY) e PlayCanvas. Nome padrão `gaussians_compressed.ply`.

 EM PALAVRAS SIMPLES

Como o PLY normal, mas 5–10× menor. A qualidade fica quase igual. Use para compartilhar a cena online ou por e-mail. Funciona direto com SuperSplat e PlayCanvas. Mas se o sistema de destino precisa de arquivos ainda menores (mobile, demos de browser), use SPZ (M27) — mais comprimido. Para qualidade plena de edição, fique no PLY não comprimido (M25).

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

Barra de menus → Export → 3D Formats → SPZ.

 TÉCNICO

Grava a nuvem Gaussian em formato SPZ — o formato comprimido de splat publicado pela Niantic, com quantização agressiva (~90 % menor que PLY não comprimido). Otimizado sobretudo para viewers web e apps mobile. Compatível com Niantic Splatt3R, `gsplat.js` e o viewer Niantic em browser.

 EM PALAVRAS SIMPLES

Um dos menores formatos. Cerca de 10× menor que um PLY normal. Use sobretudo se quer mostrar a cena num browser ou ver num app de celular. Para qualidade máxima, PLY é a melhor escolha. SPZ é desenvolvido pela Niantic e funciona direto com `gsplat.js`, Splatt3R e o viewer web da Niantic. Devido à compressão forte, você não consegue retomar facilmente o treinamento de arquivos SPZ — para edição, use PLY.

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

Barra de menus → Export → 3D Formats → glTF.



Grava um arquivo `.glb` (binary glTF) com a extensão `KHR_gaussian_splatting`. Conforme o padrão, adequado a pipelines que usam engines glTF como Babylon.js ou Three.js e implementam a extensão `KHR_gaussian_splatting`.

 EM PALAVRAS SIMPLES

Salva a cena em formato glTF, que muitos programas 3D e engines web entendem — desde que suportem a extensão Gaussian Splatting. Se você tem um pipeline 3D específico (p. ex. Three.js ou Babylon.js) que entende, esse é o formato. O arquivo sai como `.glb` binário — um único pacote com tudo. Para workflows clássicos de splatting, em geral PLY ou SPZ são melhor, porque mais ferramentas leem direto.

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

Barra de menus → Export → 3D Formats → .splat.



Grava o formato `.splat` do antimatter15 — fixo de 32 bytes por Gaussian (posição como 3× Float32, escala como 3× Float32, rotação como 4× Uint8 quaternion normalizado, RGB+opacidade como 4× Uint8). Sem coeficientes SH além do DC. Menor arquivo com compatibilidade direta de browser. Para `gsplat.js` e o viewer demo online do antimatter15.

 EM PALAVRAS SIMPLES

O formato de viewer web mais simples. Pequeno e imediatamente exibível em qualquer browser. Mas perde detalhes de iluminação (coeficientes SH mais altos vão embora — o splat parece o mesmo de qualquer ângulo, em vez de reagir à luz). Bom para performance máxima na web, ruim para foto-realismo — aí SPZ ou PLY. Funciona com o viewer online do antimatter15 e `gsplat.js`. Cada Gaussian ocupa fixos 32 bytes, o que torna o formato simples e compatível — mas ao custo da profundidade de detalhe.

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

Barra de menus → Export → 3D Formats → SOG.



Grava a nuvem Gaussian em formato SOG. SOG („Self-Organizing Gaussian“) é o formato PlayCanvas com layout de texture atlas e compressão WebP dos dados quantizados. Escala com razão de tamanho 15–20× melhor que PLY. A exportação chama internamente `cwebp` como ferramenta externa — por isso, na variante sandbox (App Store) pode ficar limitada.

 EM PALAVRAS SIMPLES

Formato muito pequeno para workflows PlayCanvas. Cerca de 15–20× menor que PLY, porque os dados são empacotados num layout de texture atlas e comprimidos em WebP. Se você não trabalha com PlayCanvas, SPZ ou Compressed PLY costuma ser melhor. A exportação chama internamente `cwebp` como ferramenta externa — na versão da App Store (sandbox) esse passo pode estar limitado.

M31 Export > Media > Export Orbit Video...

Barra de menus → Export → Media → Orbit Video.



Renderiza uma órbita 360° em torno do centro da cena e codifica como MP4 (H.264) ou MOV (HEVC, conforme o padrão do sistema). Ao contrário de M18 (gravação ao vivo), o caminho aqui é fixo — a duração é escolhida nas configurações ou no passo de export do Modo Iniciante.

 EM PALAVRAS SIMPLES

Gera automaticamente um vídeo com rotação em torno da sua cena. Sem precisar movimentar manualmente. Bom para redes sociais ou demo rápida. Se quer controlar a câmera você mesmo, use Record Turntable Video (M18). O caminho é fixo: uma órbita 360° em torno do centro da cena; a duração você escolhe nas configurações ou no passo de export do Modo Iniciante. O vídeo sai conforme o sistema como MP4 H.264 ou MOV HEVC.

M32 Export > Media > Export Web Viewer...

Barra de menus → Export → Media → Web Viewer.



Empacota um viewer HTML standalone (baseado em gsplat.js) mais os dados Gaussian codificados em base64 num único arquivo `.html`. Esse arquivo roda offline em qualquer browser moderno — sem dependências de servidor, sem URLs externas. Tamanho aproximadamente 1.3× a variante SPZ (devido ao overhead de base64).

 EM PALAVRAS SIMPLES

Salva a cena como página web autônoma. Duplo clique no arquivo HTML → o browser abre → cena 3D interativa pronta. Funciona sem internet, pode ser enviada por e-mail; é a forma mais simples de compartilhar o resultado com amigos ou clientes. O arquivo contém todo o viewer gsplat.js e os dados Gaussian num único documento — nada é baixado da web. O arquivo é cerca de um terço maior que um export SPZ, em troca quem recebe não precisa de software extra.

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

Barra de menus → Export → Photogrammetry → Export SfM (transforms.json).



Caminho próprio de exportação (não passa pela rotina helper comum), porque não exporta a nuvem Gaussian, mas o resultado de SfM. Abre um diálogo de salvar com `transforms.json` como nome padrão e content type `json`. Em OK é gravado um `transforms.json` compatível com nerfstudio, com intrínsecas, poses (como matriz 4×4 na convenção NeRF) e caminhos de frame. O texto de ajuda na UI alerta que as imagens de treinamento devem ser copiadas para uma pasta sibling `images/`. Ativo só quando há um resultado de SfM e a versão completa está desbloqueada.

 EM PALAVRAS SIMPLES

Se quer reutilizar o resultado de SfM em outro software como nerfstudio, Brush, gsplat ou OpenSplat, é aqui que você exporta as posições de câmera. Coloque suas imagens de treinamento adicionalmente numa pasta `images/` ao lado do arquivo `transforms.json` — caso contrário o programa-alvo não consegue relacionar as imagens. A entrada fica cinza enquanto não existir resultado de SfM, e travada na versão Free Trial. Para o workflow COLMAP, use M34.

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



ONDE

Barra de menus → Export → Photogrammetry → Export SfM (COLMAP Workspace).



TÉCNICO

Abre um diálogo de salvar com nome padrão `colmap-workspace` (sem extensão, porque é uma pasta). Grava um workspace COLMAP padrão com `sparse/0/cameras.bin`, `images.bin`, `points3D.bin`. Permite abrir, em outras ferramentas como Postshot, Nerfstudio ou Meshroom, uma reconstrução SfM calculada ou importada no RadianceKit, ou, num re-run A/B, recarregar como entrada já calculada no próprio RadianceKit (via M5) — poupa tempo de cálculo. Ativo só quando há um resultado de SfM e a versão completa está desbloqueada.

EM PALAVRAS SIMPLES

Como M33, mas em formato COLMAP em vez de nerfstudio. Se você usa Postshot, Meshroom, Nerfstudio ou outra ferramenta com workflow COLMAP, é o seu export. Efeito colateral útil: você pode recarregar essa pasta depois via M5 no RadianceKit e poupar o tempo de cálculo do SfM na próxima corrida — em cenas grandes, horas economizadas. Como M33, só disponível quando há resultado de SfM, e travado na versão Free Trial.

Menu Help

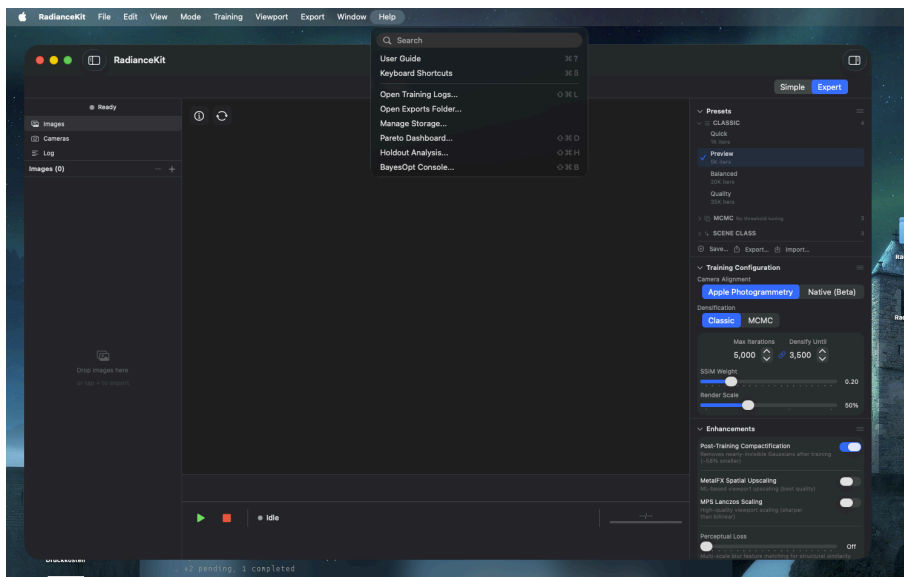


Figura 6: Menu Help com entradas de documentação, pastas e análise

Sete entradas: duas janelas de documentação (User Guide, Keyboard Shortcuts), três atalhos para pastas (Training Logs, Exports, Storage) e três janelas de análise (Pareto Dashboard, Holdout Analysis, BayesOpt Console). Como é típico na Apple, o menu Help fica na ponta direita. O menu Help padrão é totalmente substituído pela variante própria do RadianceKit.

M35 Help > User Guide

Barra de menus → Help → User Guide (⌘?).



Abre a janela User Guide. Mostra uma navegação com sidebar de tópicos e área de detalhe com scroll, em tamanho padrão 860×640. Os conteúdos estão estáticos (não vêm de markdown parseado).

EM PALAVRAS SIMPLES

Abre o guia interno do app. Se você não quiser consultar tudo neste manual, encontra ali os passos mais importantes direto no programa. O guia é uma janela própria com sidebar de tópicos — você pode pular direto para temas específicos. Os conteúdos são mais curtos que este manual e focam nos workflows mais comuns.

M36 Help > Keyboard Shortcuts

Barra de menus → Help → Keyboard Shortcuts (⌘/).



Abre a janela Keyboard Shortcuts — um layout simples com scroll e todos os atalhos do app, agrupados por menu de nível superior. Tamanho padrão 440×560. Conteúdos também estáticos.

EM PALAVRAS SIMPLES

Abre uma janela com a lista completa de atalhos de teclado. Se você não lembra, p. ex., a tecla para iniciar o treinamento, olhe aqui. Um resumo também aparece no fim deste capítulo. A lista é agrupada por menu de nível superior, para você ir rapidamente à área certa. Útil ao trocar do estilo mouse para o estilo teclado.

M37 Help > Open Training Logs...

Barra de menus → Help → Open Training Logs... (⇧⌘L).



Calcula a pasta de logs como ~/Documents/RadianceKit/Logs, cria se necessário e abre no Finder. Cada corrida de treinamento grava ali seu próprio JSONL `training_YYYY-MM-DD_HHmms.jsonl`.

EM PALAVRAS SIMPLES

Abre no Finder a pasta com todos os logs de treinamento até agora. Se algo deu errado ou você quer ver exatamente quando o treinamento convergiu para qual valor, vai encontrar aqui em arquivos JSONL. Por corrida de treinamento existe um arquivo com timestamp — pode importar em outras ferramentas ou enviar por e-mail ao suporte. Para uma avaliação gráfica, o Pareto Dashboard (M40) é a porta de entrada melhor.

M38 Help > Open Exports Folder...

Barra de menus → Help → Open Exports Folder...



Análogo a M37, mas com ~/Documents/RadianceKit/Exports. Criado na primeira corrida de auto test ou no primeiro clique; depois disso, os caminhos padrão de todas as exportações de auto test vão para ali (p. ex. autotest_<timestamp>.ply). Exportações escolhidas manualmente pelo diálogo de salvar NÃO vão necessariamente para esta pasta, mas para onde o usuário salvar — por isso essa pasta interessa sobretudo a auto tests.

 EM PALAVRAS SIMPLES

Abre a pasta em que o app grava suas próprias exportações (sobretudo runs de auto test). Se você salvou um export manual em outro lugar pelo diálogo de salvar, ele está lá, não aqui. Útil para limpar ou ver quanto espaço exports de teste antigos ocupam. Para um overview completo com logs e bundles de cena, use Manage Storage (M39).

M39 Help > Manage Storage...

Barra de menus → Help → Manage Storage...



Abre o Storage Browser (ver Capítulo 4 Janelas auxiliares, IDs W7–W12). Lista todas as cenas persistentes, logs de treinamento, exports e caches em ~/Documents/RadianceKit/, com tamanho; permite Reveal in Finder e Move to Trash por entrada.

 EM PALAVRAS SIMPLES

Abre um browser em janela que mostra quanto espaço o RadianceKit ocupa no seu disco — por cena, log e export. Você pode apagar coisas individualmente, sem ir ao Finder. Útil depois de uso prolongado, quando o disco enche — logs antigos e exports de auto test podem somar vários gigabytes. Pelo Reveal in Finder você chega à visão clássica a qualquer momento.

M40 Help > Pareto Dashboard...

Barra de menus → Help → Pareto Dashboard...
(⇧⌘D).



Abre o Pareto Dashboard (ver Capítulo 4, IDs W13–W22). O dashboard carrega todos os logs JSONL de treinamento em `~/Documents/RadianceKit/Logs/`, agrupa por cena e predefinição e desenha um scatter plot de Pareto (padrão: loss vs Gaussians, opcional loss vs wallclock ou PSNR vs iterações).

EM PALAVRAS SIMPLES

Abre uma visão geral de todos os treinamentos até agora como gráfico. Você vê na hora qual run deu a melhor relação entre qualidade e tamanho. Útil para comparar várias predefinições. Por padrão o gráfico mostra loss contra número de Gaussians — pode trocar também para tempo wallclock ou PSNR. Os dados vêm dos logs JSONL de treinamento (M37); quanto mais runs você tem, mais expressiva a análise.

M41 Help > Holdout Analysis...

Barra de menus → Help → Holdout Analysis...
(⇧⌘H).



Abre a janela Holdout Analysis (ver Capítulo 4, IDs W23–W29). Carrega um `transforms.json`, desenha as câmeras como um globo 3D e permite divisões em folds train/test (angulares ou lineares, 2–8 folds). A saída é um `fold-assignment.json` que o treinamento pode usar nas configurações como conjunto de teste.

EM PALAVRAS SIMPLES

Ajuda a dividir suas capturas de câmera em conjuntos de treinamento e teste — para você medir objetivamente quão boa é a cena (em imagens que o treinamento não viu). Mais uma ferramenta de pesquisa e benchmark. As câmeras aparecem como globo 3D; você pode escolher entre 2 e 8 folds, uniformemente por ângulo ou linearmente pela ordem. O resultado é um pequeno JSON que o treinamento usa como conjunto de teste.

M42 Help > BayesOpt Console... **ONDE**

Barra de menus → Help → BayesOpt Console... (⇧⌘B).

 **TÉCNICO**

Abre o BayesOpt Console (ver Capítulo 4, IDs W30–W39). Carrega espaços de busca pré-definidos (p. ex. „MCMC scale-reg + opacity-reg + ssim“), roda trials de Bayesian Optimization de forma assíncrona e mostra a curva de convergência e o log de trials ao vivo.

 **EM PALAVRAS SIMPLES**

Um console auto-tuner embutido. Em vez de testar manualmente vários parâmetros, o app faz isso sozinho durante a noite e ao fim sugere os melhores valores para sua cena. Ferramenta muito avançada — na maior parte dos workflows basta uma boa predefinição (ver Capítulo 7). Você escolhe um espaço de busca pré-definido (p. ex. „MCMC scale-reg + opacity-reg + ssim“) e vê ao vivo a curva de convergência e o log de trials. Conforme o setup, planeje várias horas a dias.

Aviso: Cmd-Z no menu Edit

Desde maio de 2026, o Project Navigator no Expert Mode suporta a remoção de imagens importadas pelo botão de menos ou pela tecla Backspace, e o desfazer via `Cmd-Z`. Essa ação de Cmd-Z aparece no menu Edit do macOS (fornecido pelo SwiftUI) como „Undo Remove Image“, enquanto houver uma imagem removida que possa ser restaurada. É registrada via o sistema padrão, não em ; por isso não existe entrada própria de M-ID no inventário.

Visão geral dos atalhos de teclado

Item de menu	Atalho
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

Menu Edit (fornecido pelo sistema, no Expert Mode com seleção ativa no Project Navigator):

Ação	Atalho
Undo Remove Image	⌘Z
Remove Selected Image	Backspace / Delete

CAPÍTULO

Capítulo 2 — Inspetor (Expert View)

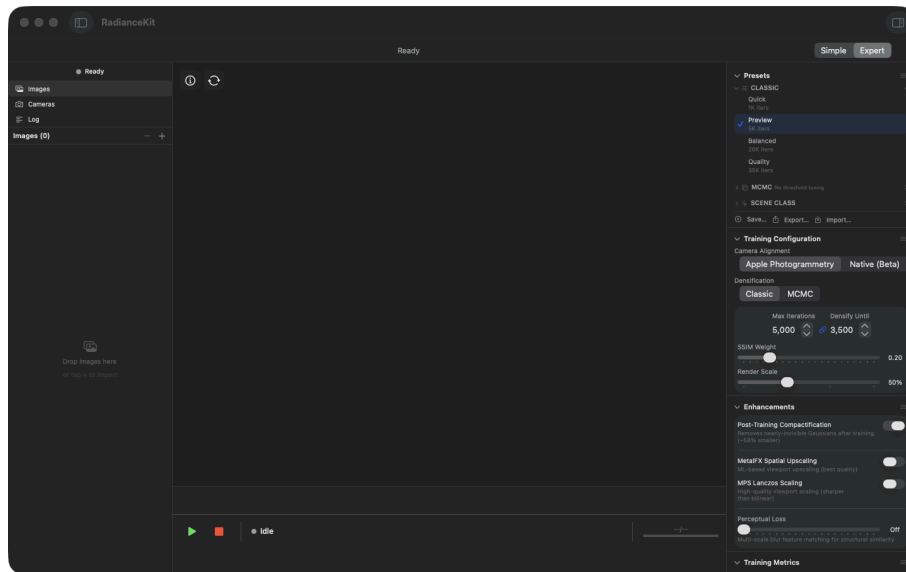


Figura 7: Expert Mode vazio — Project Navigator à esquerda (Images 0, Cameras, Log), viewport vazio no centro, Inspetor à direita com seções Presets/Training Configuration/Enhancements/Training Metrics

Inspetor vazio antes do import: sidebar esquerda mostra contador de Images 0 e a dica „Drop images here / or tap + to import“. O Inspetor à direita está totalmente funcional, mas as predefinições são apenas informativas (sem treinamento ativo). A predefinição padrão „Preview“ (5K iters) está marcada. Camera Alignment em Apple Photogrammetry, Densification Classic, SSIM Weight 0.20, Render Scale 50 %. Empty states em Training Metrics („Start training to see live metrics“) e em Loss History („Loss curve will appear during training“).

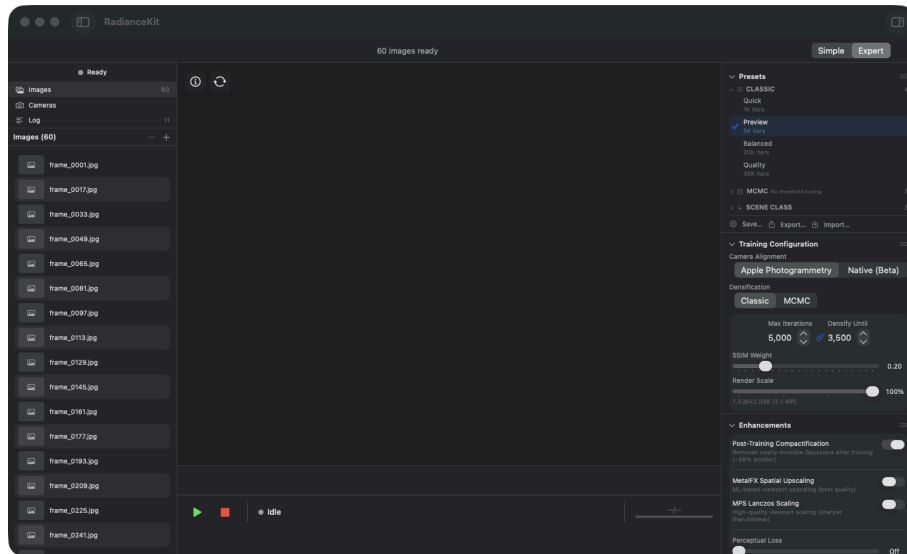


Figura 8: Inspetor com 60 imagens carregadas — sidebar de imagens mostra os primeiros nomes `frame_0001.jpg` ff, cabeçalho „60 images ready“

Inspetor após o import: status no cabeçalho „60 images ready“. A sidebar de imagens lista todos os 60 frames importados (`frame_0001.jpg` até `frame_0945.jpg` , cada 16° frame do dataset de buquê de 960 câmeras como subconjunto para iterações rápidas). A lógica de auto render scale verifica a resolução da imagem ($1536 \times 2048 = 3.1$ MP) e ajusta o Render Scale. O botão Play (verde, embaixo à esquerda) está ativo e inicia o treinamento com a predefinição ativa.

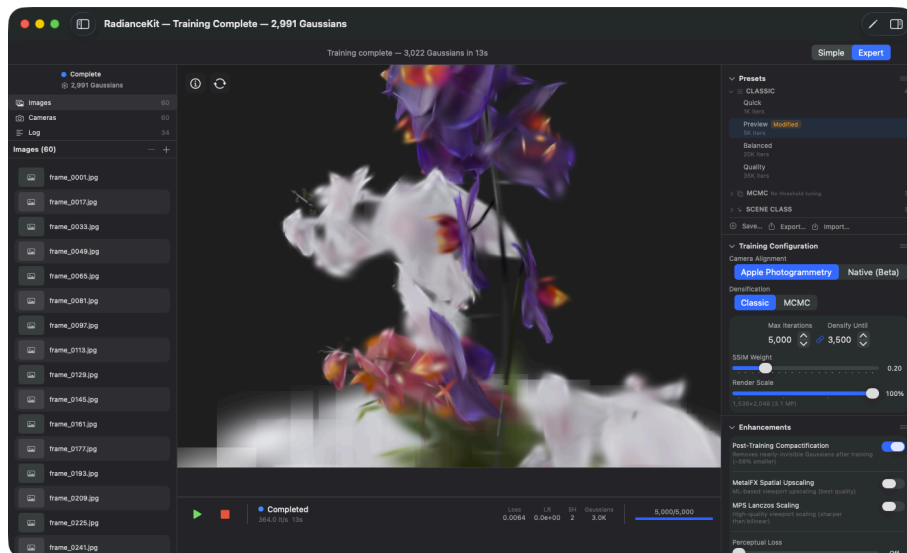


Figura 9: Inspetor durante o treinamento — viewport ao vivo mostra a reconstrução do buquê, barra de métricas embaixo (Loss / LR / Gaussian Count / Iterações), card de predefinição „Preview“ com badge „Modified“ se parâmetros foram ajustados

Inspetor durante o treinamento: a barra de título mostra o progresso global „RadianceKit — Training NN %“. O viewport renderiza em tempo real a reconstrução Gaussian em andamento (atualizada a cada 50 iterações — o intervalo do live preview é ajustável em Configurações → General → Training → Live Preview). Barra de métricas embaixo do viewport: loss atual, learning rate, número de Gaussians e contador de iterações (p. ex.

1.600/5.000 com a predefinição Preview). O card de predefinição „Preview” no Inspetor recebe um badge „Modified” assim que qualquer parâmetro se desvia do padrão built-in. A sidebar „Log” coleta eventos das fases de SfM e treinamento.

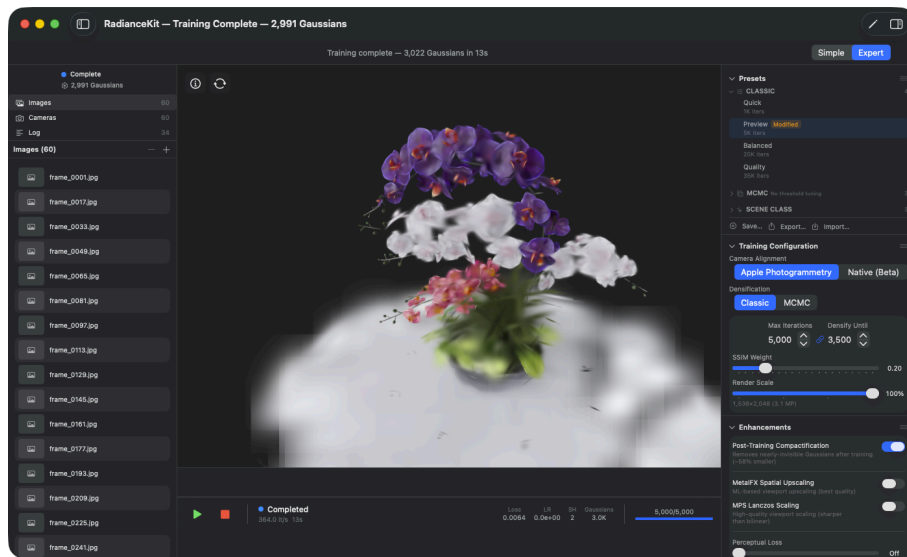


Figura 10: Inspetor após conclusão do treinamento — viewport mostra reconstrução final do buquê (2.991 Gaussians após 5K iterações em 13s), barra de título „Training Complete — 2,991 Gaussians”

Inspetor após o treinamento: a barra de título mostra o número final de Gaussians (aqui 2 991 — muito compacto porque a cena sintética de buquê em fundo claro tem geometria simples). O viewport mostra a nuvem de pontos final — navegação orbital por drag está ativa (rotaciona em torno do centro da cena). A seção Training Metrics agora está preenchida com valores finais; o gráfico Loss History mostra a curva das 5 000 iterações. A seção Export embaixo agora está ativa (todos os botões de formato habilitados).

O Inspetor é a barra lateral direita no Expert Mode (⌘2). Ele agrupa todos os parâmetros relevantes ao treinamento em sete seções colapsáveis. A ordem padrão de cima para baixo no primeiro início é: Look, Predefinições, Configuração de treinamento, Métricas, Gráfico de Loss, Enhancements e Export. A seção „Look” (ajustes de imagem pós-treinamento) é a renomeação real na UI da antiga seção „Finishing” — seu `rawValue` interno de enum permanece „Finishing” por motivos de persistência, mas o cabeçalho exibido diz „Look”. Cada seção pode ser fechada por clique no cabeçalho, a ordem pode ser reorganizada por arrastar/soltar (`InspectorView.swift:81–97`). **No primeiro início todas as sete seções estão colapsadas** (`InspectorCollapsibleSections` assume como padrão `Set(InspectorSection.allCases)`); o app salva as preferências de colapso e ordem entre inícios.

Vários controles do Inspetor aparecem em forma quase idêntica também nas Configurações (Capítulo 3) — tipicamente backend SfM, sky masking e padrões semelhantes. A separação é proposital: as Configurações fornecem o template global do app para novos projetos; o Inspetor sobrescreve esses valores para o projeto aberto no momento. Quem entende a lógica de um lado consegue usar o outro às cegas.

A coluna esquerda no Expert Mode — o Project Navigator — não faz parte do Inspetor, mas é seu vizinho direto. Lá você pode selecionar imagens importadas por clique, ver

em quick look pela barra de espaço e remover pelo botão de menos ou pela tecla Delete (com Cmd-Z para desfazer). O Inspetor segue a seleção atual da sidebar com detalhes contextuais, mas as sete seções principais ficam sempre disponíveis.

Seção Look (L1–L5)

A seção Look (`rawValue` interno ainda „Finishing“) é a seção superior do Inspetor e reúne os ajustes de imagem **pós-treinamento** em um só lugar. Todos os controles trabalham de forma **não destrutiva**: cada slider reaplica o `FinishingPass` sobre um snapshot pristine inalterado (cor DC, opacity e escala originais) — o ajuste é portanto **idempotente**, não cumulativo. O resultado aparece **ao vivo no viewport** (WYSIWYG, exatamente como o export posterior) e é **gravado em cada export**. A seção só fica disponível **após a conclusão de um treinamento** (antes disso aparece „Available after a training run completes.“); seus valores são **resetados a cada novo treinamento**. Enquanto um export está rodando, todos os controles ficam **bloqueados** — um aviso de lock „Locked while exporting — the file uses the current settings.“ aparece e a GroupBox fica desabilitada.

L1 Slider Saturation



Inspetor → seção Look → GroupBox → Saturation.



Slider 0.5–1.2, exibição com duas casas (p. ex. „1.00“). Escala a croma DC-SH de cada Splat em torno do valor de luminância: 1.0 = inalterado, < 1.0 = dessaturado (cor puxada para tons de cinza), > 1.0 = mais vívido. Matematicamente a cor DC é recalculada a partir do snapshot pristine (`desaturateDC`), de modo que arrastar repetidamente não se acumula. Foi validado em material de drone DJI (viaduto de Pensford), que tende a saturar demais — o padrão para drone fica em 0.82. Atua só sobre a base de cor (grau SH 0), os coeficientes SH superiores ficam intocados.

EM PALAVRAS SIMPLES

Quão vívidas são as cores do Splat finalizado. 1.00 deixa tudo como treinado, valores abaixo puxam a cor para o cinza — ótimo para material de drone ou vídeo, que costuma sair supersaturado. Valores acima de 1.0 deixam tudo mais vívido. Você pode arrastar de um lado para o outro à vontade sem que nada se „acumule“, porque o app sempre recalcula a partir do estado original inalterado. Visível ao vivo no viewport e idêntico no export.

L2 Slider Splat length



ONDE

Inspetor → seção Look → GroupBox → Splat length.



TÉCNICO

Slider 0.3–1.0, exibição com duas casas. Puxa os três eixos de escala de cada Gaussian no espaço logarítmico em direção à sua média (`shortenScale` , fator `alpha`): 1.0 = inalterado, valores menores deixam Splats „agulha“ alongados mais arredondados, 0 seriam esferas puras. Ataca Splats em forma de agulha e superesticados sem mudar o tamanho geral, reduzindo assim os artefatos típicos de „confete“. Aplicado a partir do snapshot pristine (escala log original), portanto idempotente. Comuta com Splat size (L3), porque ambos trabalham no espaço logarítmico.

EM PALAVRAS SIMPLES

Deixa Splats compridos e estilizados mais arredondados. 1.00 deixa a forma como treinada, valores menores comprimem as „agulhas“ alongadas em bordões mais arredondados — isso acalma reconstruções granuladas, atormentadas por artefatos de confete. O tamanho geral fica igual, é só uma questão de alongamento. Pode combinar com Splat size (L3) sem risco.

L3 Slider Splat size



ONDE

Inspetor → seção Look → GroupBox → Splat size.



TÉCNICO

Slider 0.5–2.0, exibição com duas casas. Escala cada Gaussian uniformemente nos **três** eixos (`sizeScale`): 1.0 = inalterado, < 1.0 = menor/mais denso/mais nítido, > 1.0 = maior/mais „fofo“ (preenche lacunas entre os Splats). Como as escalas estão no espaço logarítmico, a multiplicação é realizada como um offset aditivo $\log(\text{factor})$ — isso comuta com Splat length (L2), porque um offset constante deixa o desvio em relação à média intocado. A partir do snapshot pristine, portanto idempotente. Novo nesta versão.

EM PALAVRAS SIMPLES

Escala todos os Splats uniformemente maiores ou menores. 1.00 é o estado treinado, valores abaixo deixam a nuvem de pontos mais apertada e nítida, valores acima cobrem lacunas entre os Splats (fica mais suave/„fofo“). Prático para fechar opticamente uma reconstrução esburacada ou, ao contrário, revelar mais detalhe. Se dá bem com Splat length (L2) sem problemas — os dois controles não interferem entre si.

L4 Fade far region (com sub-sliders)

ONDE

Inspetor → seção Look → GroupBox → Toggle „Fade far region“ mais os sub-sliders „Fade start xradius“ e „Fade floor“.

TÉCNICO

Toggle que ativa uma queda radial de opacity com a distância em relação ao centroide das câmeras — os „far-confetti“ fracamente observados ao fundo são desvanecidos. **Só para capturas em órbita:** o toggle fica desabilitado quando `finishingContext.fadeEligible` é false (voos lineares, câmeras insuficientes ou degeneradas); então, em vez dos sub-sliders, aparece o aviso „Far-fade applies only to orbit captures (not this scene).“ A elegibilidade é determinada pela cobertura de azimute das posições de câmera (uma órbita circunda o centroide e preenche muitos setores da bússola, um voo linear apenas ~2). Dois sub-sliders controlam a geometria: **Fade start xradius** (1.0–3.0) define o raio interno como múltiplo do raio da órbita, dentro do qual vale opacity plena; **Fade floor** (0.0–1.0) é o fator de opacity bem além do raio de fade. Importante: o fade **pula a região do sky-dome** (os Gaussians congelados dos índices [0, frozenCount]), para que a cúpula de fundo intencional não seja escurecida junto.

EM PALAVRAS SIMPLES

Desvanece os restos esponjosos na borda externa da cena — justamente os grumos de „far-confetti“ que flutuam bem ao fundo em capturas em volta. Funciona só com capturas reais de órbita/volta; em voos retos de drone ou com câmeras de menos, o botão fica cinza e um aviso explica por quê. Quando ativo, somam-se dois ajustes finos: „Fade start xradius“ define a partir de que distância (como múltiplo do raio da volta) o desvanecimento começa, „Fade floor“ o quanto os Splats distantes ainda permanecem visíveis no fim (0 = some de vez, 1 = inalterado). Um sky-dome reconstruído de propósito (I44) nunca é tocado nesse processo — o céu permanece preservado.

L5 Botão Reset finishing

ONDE

Inspetor → seção Look → GroupBox → „Reset finishing“ (embaixo, botão pequeno).

TÉCNICO

Reseta todos os ajustes de Look para os padrões (`FinishingPass.Settings() = Saturation 1.0, Fade desligado, Splat length 1.0, Splat size 1.0`) e dispara imediatamente um novo finishing, de modo que o viewport volta ao estado treinado inalterado. `controlSize(.small)`. Como toda a pilha de Look calcula de forma idempotente a partir do snapshot `pristine`, „voltar ao padrão“ é exatamente a saída de treinamento original — sem perda de qualidade por vai e vem repetido. Como todos os controles da seção, bloqueado durante um export em andamento.

EM PALAVRAS SIMPLES

Devolve com um clique todos os controles de Look ao padrão (Saturation 1.00, Fade desligado, ambos os sliders de Splat em 1.00) — depois o viewport mostra de novo exatamente o resultado recém treinado. Prático quando você se empolgou e quer recomeçar do zero com tudo limpo. Como o app sempre calcula a partir do estado original, não há perda de qualidade nisso. Enquanto um export está rodando, o botão (como os sliders) fica bloqueado.

Seção Predefinições (I1–I11)

A seção de predefinições é a forma mais rápida de aplicar uma configuração testada. Predefinições built-in (Capture Class, Classic, MCMC, Hybrid) entregam pontos de partida reproduzíveis a partir de mais de 560 experimentos documentados; predefinições próprias podem ser salvas, exportadas, importadas e compartilhadas. A lista é agrupada por categorias (Capture Class, Classic, MCMC, Hybrid, Custom) e mais de uma categoria pode ficar aberta ao mesmo tempo. Pelo mecanismo de menu de contexto (clique direito numa linha), ficam acessíveis Export, Duplicar e — em predefinições próprias — Apagar.

I1 Botão Save...



Inspetor → seção de predefinições → botão Save... (barra de ações embaixo).



Abre um popover com campo de texto e botões Save/Cancel. O estado atual da TrainingConfig é persistido como nova predefinição do usuário (codificada em JSON, salva entre apps). A ação de salvar copia todos os 81 parâmetros de treinamento mais a estratégia de densification atual. A predefinição vai automaticamente para a categoria Custom, independentemente de qual built-in serviu de base. Nomes vazios e só com whitespace são descartados. Nomes já existentes não são rejeitados — cada predefinição tem ID interno próprio; nomes duplicados são tecnicamente permitidos, mas confundem na prática.

EM PALAVRAS SIMPLES

Salva sua configuração atual como predefinição reutilizável. Aperte o botão, dê um nome no popover e clique Save — todos os 81 parâmetros, incluindo a estratégia de densification, ficam sob o nome escolhido na categoria Custom. Você precisa disso quando se deu trabalho e não quer mexer tudo de novo no próximo projeto. Particularmente útil para setups recorrentes como „Drone 4K“ ou „Indoor rápido“. Nomes duplicados são tecnicamente permitidos mas confusos — escolha algo descritivo.

I2 Campo Preset Name



Popover Save → campo de texto „Preset Name“.



Campo de texto simples com moldura arredondada, forma larga. O valor é assumido como nome de predefinição ao clicar em Save. Sem limite de tamanho na UI, mas o nome salvo precisa ser codificável em JSON e exibível nas listas da UI — emojis e acentos funcionam. O conteúdo é resetado para string vazia ao abrir o popover. O botão Save fica desabilitado enquanto o campo após trim estiver vazio. Não há auto-suggest nem pré-preenchimento com o nome da predefinição ativa.

EM PALAVRAS SIMPLES

Aqui você digita o nome da sua predefinição. Escolha algo descritivo como „Drone 4K 30fps“ ou „Indoor rápido“ — ajuda a encontrar depois na categoria Custom. Emojis e acentos permitidos; sem limite duro de tamanho. Enquanto o campo estiver vazio ou só com espaços, Save fica cinza. Ao reabrir o popover, o campo volta a vazio — não há pré-preenchimento com o nome da predefinição ativa.

I3 Botão Cancel (diálogo Save)



Popover Save → botão Cancel (à esquerda).



Fecha o popover sem salvar. Descarta o conteúdo do campo — na próxima abertura, ele é resetado para vazio pela lógica do botão Save... (I1). Estilo de botão padrão, sem diálogos de confirmação, sem hotkeys. A TrainingConfig atual permanece inalterada, pois o caminho de save não foi executado.

EM PALAVRAS SIMPLES

Fecha o popover Save sem salvar nada. Se mudou de ideia, errou ou abriu o diálogo sem querer — clique Cancel. A configuração atual fica intocada, porque nada foi gravado. Ao reabrir, o nome volta vazio. Sem confirmação, sem hotkey — clique e pronto.

I4 Botão Save (diálogo Save)



Popover Save → botão Save (à direita, estilo proeminente).



Dispara a persistência real. Faz nova validação de nome não vazio (check defensivo) e grava a TrainingConfig atual como JSON no armazenamento do app. Depois fecha o popover. Destacado em azul, fica cinza enquanto o campo está vazio. Se o save falhar (p. ex. armazenamento cheio — muito improvável), por ora não há diálogo de erro visível; a predefinição simplesmente não apareceria no próximo início do app.

EM PALAVRAS SIMPLES

Ao clicar Save você confirma o nome e grava seu setup atual como nova predefinição. O popover fecha, a predefinição aparece imediatamente na categoria Custom da lista e pode ser ativada por clique. O botão tem destaque azul (`borderedProminent`) e fica cinza enquanto o campo está vazio. Se o save falhar (p. ex. UserDefaults cheio), não há diálogo de erro visível — a predefinição simplesmente não estaria lá no próximo início.

I5 Botão Export...



Inspetor → seção de predefinições → barra de ações → botão Export...



Exporta a predefinição atualmente selecionada como arquivo `.radiancepreset` (internamente JSON). Desabilitado se nenhuma predefinição estiver selecionada. Ao clicar, o app abre um diálogo de salvar com nome pré-definido (nome da predefinição + extensão `.radiancepreset`). O formato salvo contém a TrainingConfig completa mais metadados (nome, categoria, ID, flag built-in). Duplo clique no Finder abre o app — mas **não** dispara o import automaticamente; o usuário precisa usar o botão Import (I6).

EM PALAVRAS SIMPLES

Selecione uma predefinição na lista e clique Export — pode salvá-la como arquivo `.radiancepreset` e p. ex. mandar para um colega ou levar para outro Mac. O destinatário carrega lá pelo botão Import... (I6). Funciona igualmente bem para built-ins e suas próprias predefinições Custom. O botão fica cinza enquanto nada está clicado na lista. Dica: pelo menu de contexto (I8) é ainda mais rápido — não precisa selecionar primeiro.

I6 Botão Import...



Inspetor → seção de predefinições → barra de ações → botão Import...



Abre um diálogo de arquivo que só permite `.radiancepreset` (multi-seleção desabilitada). Ao escolher, o JSON é carregado, validado e inserido na categoria Custom — com nova ID interna para não colidir com built-ins. O import sempre seta a categoria como Custom, mesmo que a predefinição original tenha sido um built-in. Arquivos corrompidos ou incompatíveis com versão de schema antiga são silenciosamente rejeitados, sem diálogo de erro (o log do console traz a razão).

EM PALAVRAS SIMPLES

Carrega um `.radiancepreset` do disco. Útil quando alguém envia um setup testado ou você quer manter suas predefinições favoritas sincronizadas entre Macs. Importados sempre vão para Custom — mesmo se vieram de built-ins. Arquivos corrompidos ou antigos são ignorados em silêncio; o log do console diz o motivo. Multi-seleção desabilitada, só um arquivo por clique.

I7 Linha de predefinição (ativação por clique)

ONDE

Inspetor → seção de predefinições → toda linha de predefinição em toda categoria.

TÉCNICO

Clicar numa linha substitui todos os campos da TrainingConfig pelos valores da predefinição, memoriza o ID da ativa e zera o status Modified. O check de ativo antes da linha só aparece quando a predefinição está selecionada E não modificada. Assim que um valor é alterado na TrainingConfig (slider, stepper, toggle nas outras seções do Inspetor), aparece um badge laranja „Modified“ depois do nome. Predefinições built-in não podem ser sobrescritas — em caso de modificação, é preciso criar uma cópia via Save... (I1).

EM PALAVRAS SIMPLES

Clicar numa linha ativa a predefinição e aplica todos os seus valores às configurações atuais. O check antes do nome mostra qual está ativa. Assim que você mexe num slider, stepper ou toggle nas outras seções, aparece o badge laranja „Modified“ depois do nome — porque o setup agora difere da predefinição. Built-ins não podem ser sobrescritas; se quiser manter mudanças, crie uma cópia via Save... (I1) ou duplique (I9).

I8 Menu de contexto „Export...“

ONDE

Clique direito em qualquer linha → primeiro item „Export...“.

TÉCNICO

Mesma funcionalidade do I5 (botão Export...), mas mais cômoda — não precisa pré-selecionar. Exporta direto a predefinição da linha clicada. Funciona igualmente para todas as categorias (built-in ou Custom), sem restrição. A exportação contém o flag built-in e a categoria original; no re-import, a categoria vai para Custom como descrito em I6.

EM PALAVRAS SIMPLES

Caminho rápido para exportar — clique direito na predefinição desejada e „Export...“. Poupa a etapa de pré-selecionar e depois clicar no botão Export.... Funciona igualmente em todas as categorias, inclusive built-ins. O .radiancepreset gerado é idêntico ao de I5; no re-import, ele vai para Custom.

I9 Menu de contexto „Duplicate“



Clique direito em qualquer linha → segundo item „Duplicate“.



Clona a predefinição para a categoria Custom. Gera nova ID interna, acrescenta „ Copy“ ao nome e salva a cópia. Funciona também para built-ins — o clone fica editável. O original permanece intocado. A TrainingConfig é copiada valor por valor (JSON roundtrip), sem ligações de referência entre original e cópia.

EM PALAVRAS SIMPLES

Cria uma cópia editável de uma predefinição na categoria Custom. Útil quando você quer, p. ex., usar o built-in „Quality“ como base e só deslocar o slider SSIM um pouco. Fluxo: duplicar, renomear (menu de contexto ou novo Save...), ajustar, pronto. O original fica intocado — você pode voltar a qualquer hora. Funciona também para built-ins, único caminho de usar seus valores como base e ao mesmo tempo deixá-los editáveis.

I10 Menu de contexto „Delete“



Clique direito em linhas próprias → último item „Delete“ (vermelho, destrutivo).



Só visível em predefinições Custom. Built-ins não são apagáveis. Marcado como destrutivo, aparece em vermelho no menu de contexto e fica após um divisor para evitar cliques acidentais. **Não há** diálogo de confirmação — um clique apaga imediatamente. Apagada não é restaurável (Cmd-Z não funciona aqui — undo existe no build atual só para a lista de imagens, não para operações de predefinição). Se a apagada estava ativa, a TrainingConfig atual permanece; só a seleção é zerada.

EM PALAVRAS SIMPLES

Apaga predefinições próprias. Em built-ins (Quick, Preview, Balanced, Quality, Ultra Detail, Drone / Aerial, 360° Walkaround, Photo / Object etc.), „Delete“ nem aparece — não tem como apagar por engano. Atenção: sem confirmação e sem undo, um clique e some. Se não tem certeza, faça antes um Export... (I5/I8) para o disco — pode importar depois. Se a apagada estava ativa, sua TrainingConfig fica intocada; só o check some.

I11 Cabeçalho de categoria (abrir/fechar)

ONDE

Inspetor → seção de predefinições → cada cabeçalho de categoria (Capture Class, Classic, MCMC, Hybrid, Custom).

TÉCNICO

Estado de colapso por categoria com padrões diferentes: o grupo curado Capture Class começa **aberto**, Classic, MCMC, Hybrid e Custom começam **fechados**. O estado não é persistido — ao reiniciar o app, todas voltam ao padrão. A setinha gira com animação. O número à direita no cabeçalho mostra quantas predefinições há na categoria. A área de clique cobre todo o cabeçalho.

EM PALAVRAS SIMPLES

Abrir e fechar categorias para deixar a lista limpa. No início do app, Capture Class está aberta; Classic, MCMC, Hybrid e Custom estão fechadas. Clique no cabeçalho (área toda clicável) e a lista abre/fecha com uma curta animação. O número à direita mostra quantas predefinições há na categoria. Após reiniciar, o estado padrão volta — o app deliberadamente não salva esse colapso.

Seção Configuração de treinamento (I12–I22)

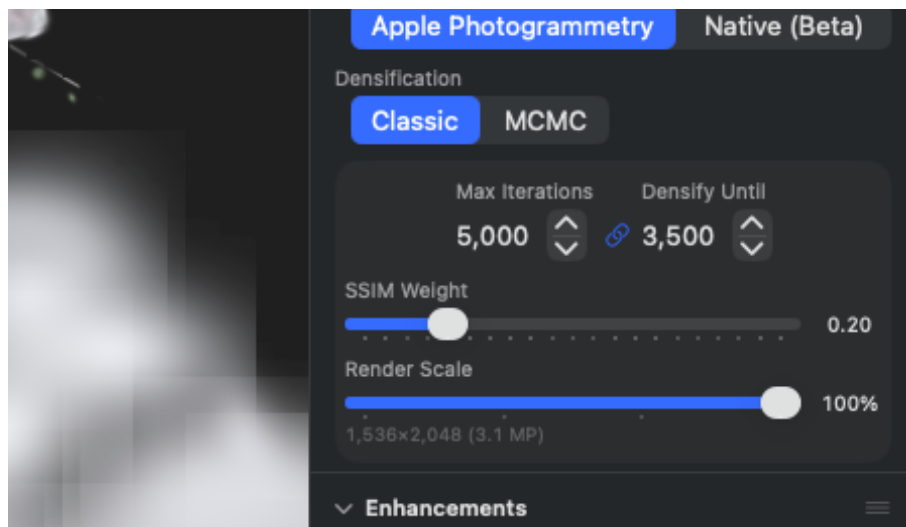


Figura 11: Recorte só da seção Configuração de treinamento — Camera Alignment (Apple Photogrammetry ativo, Native (Beta) inativo), Densification (Classic ativo), Max Iterations 5,000 / Densify Until 3,500 com ícone de link, slider SSIM Weight 0.20, slider Render Scale em 100 % (1,536×2,048 = 3.1 MP)

Aqui ficam as alavancas centrais: qual backend SfM calcula, como a densification trabalha, quantas iterações, qual o peso do SSIM. Na estratégia MCMC surgem dois toggles adicionais („MCMC Quality“ e „Auto-scale by scene“) que ficam ocultos no modo Classic. No backend Native SfM aparece o campo FOV Override, necessário só para frames de vídeo sem distância focal no EXIF.

I12 Seletor Camera Alignment



Inspetor → Configuração de treinamento → Camera Alignment (seletor segmentado em cima).



Seletor segmentado com duas opções: Apple Photogrammetry e Native (Beta). A escolha define o backend SfM usado na próxima reconstrução de câmeras. Influencia ainda quais outros elementos do Inspetor ficam visíveis: Native exibe o FOV Override (I13), necessário só para frames de vídeo sem EXIF. Observação: para capturas externas muito grandes, você pode carregar o resultado de uma ferramenta externa (Metashape ou COLMAP) via import de workspace — ver Capítulo 1 (M5) e Capítulo 9 (Q3, Q6).

EM PALAVRAS SIMPLES

Aqui você escolhe como as posições de câmera são reconstruídas — a chave mais importante para a qualidade final. Apple Photogrammetry é o padrão rápido e basta para a maioria dos scans de objeto. Native (Beta) é o desenvolvimento próprio compatível com a App Store, bom para orbits e cenas turntable, e precisa de FOV Override (I13) em frames sem EXIF. Em sets externos muito grandes, você pode calcular as câmeras no Metashape ou COLMAP e carregar via import de workspace. Detalhes e recomendações por tipo de cena no Capítulo 9.

I13 Campo FOV Override (Native SfM)



Inspetor → Configuração de treinamento → FOV Override (só visível com Camera Alignment = Native).



Campo de texto numérico (faixa 0–170°), padrão 0 = determinação automática por EXIF ou heurística. Entrada manual é necessária quando as imagens vêm de vídeo sem metadados de distância focal. Valores típicos: iPhone Wide $\approx 73^\circ$, DJI Mavic Wide-Crop $\approx 70^\circ$, drone com sensor full frame $\approx 84^\circ$. O valor é clampeado para [0, 170] — valores fora são reduzidos. Atua só no pipeline SfM nativo (Q4/Q5); Apple Photogrammetry ignora este valor.

EM PALAVRAS SIMPLES

Se suas imagens não têm EXIF (típico em frames extraídos de vídeo), preencha aqui o campo de visão horizontal da câmera em graus. Regra prática: iPhone Wide $\approx 73^\circ$, DJI Mavic Wide-Crop $\approx 70^\circ$, drone com sensor full frame $\approx 84^\circ$. O deixa o app adivinhar — costuma funcionar, mas pode falhar em lentes raras. Valores acima de 170° são reduzidos automaticamente. O campo só aparece e só atua quando você escolheu Native em Camera Alignment (I12) — Apple Photogrammetry ignora.

I15 Seletor Densification

Inspetor → Configuração de treinamento → Densification (seletor segmentado, sempre visível).



Alterna entre as duas estratégias de densification: Classic (procedimento 3DGS original com clone/split/prune e threshold de gradiente) e MCMC (Stochastic Gradient Langevin Dynamics com relocação, NeurIPS 2024). Ao mudar de Classic para MCMC, o app reseta automaticamente os campos específicos de MCMC para valores padrão testados (pesos de regularização = 0, multiplicador de cap MCMC 3.0, schedule de sample/noise). Sem essa inicialização automática, sessões com predefinições antigas sofriam do bug de MCMC collapse de v1.4.4 (460K→5 Gaussians, watchdog kill). A escolha do seletor define também quais elementos do Inspetor ficam visíveis — em MCMC aparecem I16/I17. Efeito detalhado de cada campo no Capítulo 6, T11–T16 (Classic) e T61–T73 (MCMC).

EM PALAVRAS SIMPLES

A escolha estratégica central para o crescimento do número de Gaussians. Classic é bem ajustado por 459 experimentos, dá resultados rápidos e de alta qualidade, sem precisar conhecer campos MCMC. MCMC é a abordagem mais nova (NeurIPS 2024), mais reprodutível e dispensa ajuste manual de threshold — em compensação roda cerca de 6x mais com qualidade similar. Ao trocar para MCMC, o app define padrões seguros automaticamente para evitar o collapse da v1.4.4. Detalhes dos campos de estratégia no Capítulo 6 (T11–T16 Classic, T61–T73 MCMC).

I16 Toggle MCMC Quality

Inspetor → Configuração de treinamento → MCMC Quality (só com Densification = MCMC).



Alterna a gradient accumulation entre 2 passos (ativo) e 1 passo (inativo). Acumula gradientes de duas vistas de câmera consecutivas antes do passo do otimizador. Empiricamente (Sessão 33, V544a) reduz o L1 final em cerca de 6% (0.0246 com Quality vs 0.0261 sem, em média de 3 trials em Horse Full MCMC). Preço: tempo de treinamento dobra. Em treinamentos muito longos (200K iterações), gera 10+ minutos extras de espera — vale apenas quando os últimos pontos percentuais de qualidade são realmente necessários. Atua apenas no treinamento, não no formato de export nem no viewport.

EM PALAVRAS SIMPLES

Modo Quality para MCMC com gradient accumulation por duas vistas. Deixa o resultado cerca de 6% melhor empiricamente (L1 0.0246 em vez de 0.0261 no teste Horse), mas custa o dobro de tempo. Se você já roda um MCMC de 200K (facilmente 2 horas), vai juntar quase uma hora a mais. Vale para renders finais de showcase ou no fim de uma sessão de quality sweep, no workflow diário nem tanto. Só aparece quando Densification está em MCMC (I15).

I17 Toggle Auto-scale by scene

ONDE

Inspetor → Configuração de treinamento → Auto-scale by scene (só com MCMC).



TÉCNICO

Quando ativo, escala o limite efetivo de Max Gaussians por SfM-Init-Point-Count × MCMC Cap Multiplier (padrão 3.0). Exemplo: SfM entrega 250K pontos iniciais, base cap = 150K, multiplier 3.0 → limite efetivo = $\max(150K, 750K) = 750K$. Quando desativado, vale somente a base. Introduzido na v1.4.5 porque grandes capturas externas com mais de 1000 frames e densidade proporcional de pontos SfM mato a densification com a cap fixa de 150K — pontos supérfluos ficavam, novos não podiam surgir. Padrão OFF em Custom, ON em built-ins MCMC. Atua apenas em tempo de treinamento, não no export.

EM PALAVRAS SIMPLES

Deixa o número máximo de Gaussians crescer com o tamanho da cena (mais precisamente: com o número de pontos iniciais do SfM). Em cenas pequenas, mal nota diferença; em cenas externas grandes costuma ser decisivo para a qualidade — senão o treinamento „sufoca“ porque o limite padrão de 150K é baixo demais. Introduzido especificamente na v1.4.5 depois que sets externos muito grandes (1000+ frames) bateram visivelmente na cap. Já vem ligado nos built-ins MCMC; em predefinições próprias, padrão desligado.

I18 Stepper Max Iterations

ONDE

Inspetor → Configuração de treinamento → Group-Box → Max Iterations.



TÉCNICO

Stepper com faixa 1 000–100 000, passo 1 000. Define o número total de iterações do otimizador. Correlaciona linearmente com o tempo de treinamento (metade \approx 50 % do tempo). Sweet spots empíricos: 20K (Classic Balanced, $L1 \approx 0.028$), 40K (Classic Quality, $L1 \approx 0.023$), 200K (MCMC Full, $L1 \approx 0.0246$). Acima de 40K no Classic quase não há ganho médio — diminishing returns. Ao alterar, se o link (I19) está ativo, Densify Until é proporcionalmente arrastado (razão padrão 0.5, i.e. Densify-Until = $\text{Max}/2$).

EM PALAVRAS SIMPLES

Quantos passos de treinamento serão feitos — mais é melhor, mas custa linearmente mais tempo. Regra prática: 20 000 para boa qualidade, 40 000 para o ótimo em Classic (acima quase não ganha mais). MCMC precisa de muito mais; 200 000 é padrão. Dobrar iterações dobra grosso modo o tempo. Com link ativo (I19), Densify Until é arrastado junto — praticamente sempre o que você quer.

I19 Botão Link/Unlink (Densify ↔ Iterations)

ONDE

Inspetor → Configuração de treinamento → Group-Box → pequeno botão de link entre Max Iterations e Densify Until.



TÉCNICO

Botão toggle que congela a razão de Densify Until para Max Iterations. Em ativo (ícone de link destacado), a cada mudança em Max Iterations, Densify Until é puxado proporcionalmente. Em unlink (ícone link-plus), os valores ficam independentes. Padrão é linked, porque reflete a correlação típica — dobrar iterações em geral exige densification proporcionalmente mais longa. A razão é calculada a partir do valor atual no instante de ativar o link; razão típica é 0.5 (Densify Until = metade do número de iters).

EM PALAVRAS SIMPLES

Pequeno botão entre Max Iterations e Densify Until. Quando ativo (ícone destacado), os dois andam juntos — dobrar Max Iterations dobra Densify Until na mesma razão. Quando não (ícone `link.badge.plus`), você define cada um. Padrão é linked, porque reflete a correlação típica — treinamento mais longo costuma querer densification mais longa também. Em 99% dos casos, deixe travado.

I20 Stepper Densify Until

ONDE

Inspetor → Configuração de treinamento → Group-Box → Densify Until.



TÉCNICO

Stepper com faixa 500–50 000, passo 500. Define o índice de iteração a partir do qual nenhum novo Gaussian é adicionado por clone/split (Classic) ou relocação (MCMC). Depois disso, só posição e cor são refinadas. Valores mais altos = mais Gaussians = arquivo maior, tempo por iteração maior (+30-60% de GPU time por passo). Valores típicos: 15K (para 30K Max-Iter), 20K (para 40K), 100K (para 200K MCMC). Com link ativo (I19), escala junto. Atua diferente em Classic vs MCMC: Classic para o crescimento totalmente; MCMC para a lógica de relocação, mas sample/noise adaption continuam.

EM PALAVRAS SIMPLES

Até qual iteração novos Gaussians podem ser acrescentados — em Classic via clone/split, em MCMC via relocação. Depois disso, só refinamento de cor e forma dos pontos existentes. Mais alto = mais detalhe, mas também arquivo maior e +30-60% de GPU por passo. Valores típicos: 15K (para 30K Max Iter), 20K (para 40K), 100K (para 200K MCMC). Normalmente fica preso a Max Iterations via link (I19) — raramente faz sentido soltar manualmente.

I21 Slider SSIM Weight



Inspetor → Configuração de treinamento → Group-Box → SSIM Weight.



Slider 0.0–1.0 em passos de 0.05, exibição „0.20“. Mistura L1 loss (0.0) e SSIM loss (1.0). L1 puxa brilho por pixel, SSIM puxa similaridade estrutural (bordas, estatísticas locais). Padrão 0.2 vem do paper original 3DGS (Kerbl 2023) e foi confirmado em muitas sessões como compromisso robusto. Valores mais altos (0.5+) priorizam preservação de detalhe, mas podem ignorar erros de brilho locais. Mais baixos (< 0.1) levam à perda de detalhe em bordas nítidas. O cálculo de SSIM roda no shader com janela Gaussian 11×11. Performance: em 0.0 (só L1) o treinamento é cerca de 8–12 % mais rápido, pois o cálculo SSIM é pulado no shader.

EM PALAVRAS SIMPLES

Quão fortemente a similaridade estrutural (bordas, padrões locais) é ponderada em relação à comparação pura de brilho. 0.2 é o padrão do paper original 3DGS e basta para quase todas as cenas. Mais alto (0.5+) em estruturas finas como cabelos, pelagem ou vegetação — ali peso estrutural ajuda. Mais baixo (0.0) acelera o treinamento 8–12% pulando o SSIM no shader, mas custa detalhe em bordas nítidas. Sem bom motivo, deixe 0.2.

I22 Slider Render Scale



Inspetor → Configuração de treinamento → Group-Box → Render Scale.



Slider 0.25–1.0 em passos de 0.25, exibição „100%“. Escala a resolução de renderização do treinamento relativa ao tamanho da imagem fonte. Maior alavanca sobre performance: 50 % reduz GPU time em 75 % (4x menos pixels), 25 % em 94 %. O gradient threshold escala junto automaticamente. Sob o slider aparece um indicador ao vivo de resolução em MP (p. ex. „2304×1296 (3.0 MP)“). Se o valor atual diverge do recomendado, aparece em laranja „— recommended: 50%“. A recomendação visa 3 MP efetivos — a região mais eficiente em GPUs Apple Silicon. Imagens fonte em 4K recebem 25 % recomendado; FullHD recebe 100 %. Mudar também dispara realocação de buffers.

EM PALAVRAS SIMPLES

Com qual resolução o treinamento renderiza — uma das maiores alavancas de performance. Cheio (100 %) dá melhor qualidade, mas custa muito GPU em imagens grandes. Metade (50 %) poupa 75 % do GPU por calcular 4x menos pixels — perfeito para fontes em 4K. Sob o slider você vê a resolução efetiva em MP; o app mira 3 MP, que é o mais eficiente em Apple Silicon. Se o seu valor diverge, aparece um aviso laranja „recommended“ — vale seguir na maioria das vezes.

Seção Enhancements (I26–I29, I42–I44)

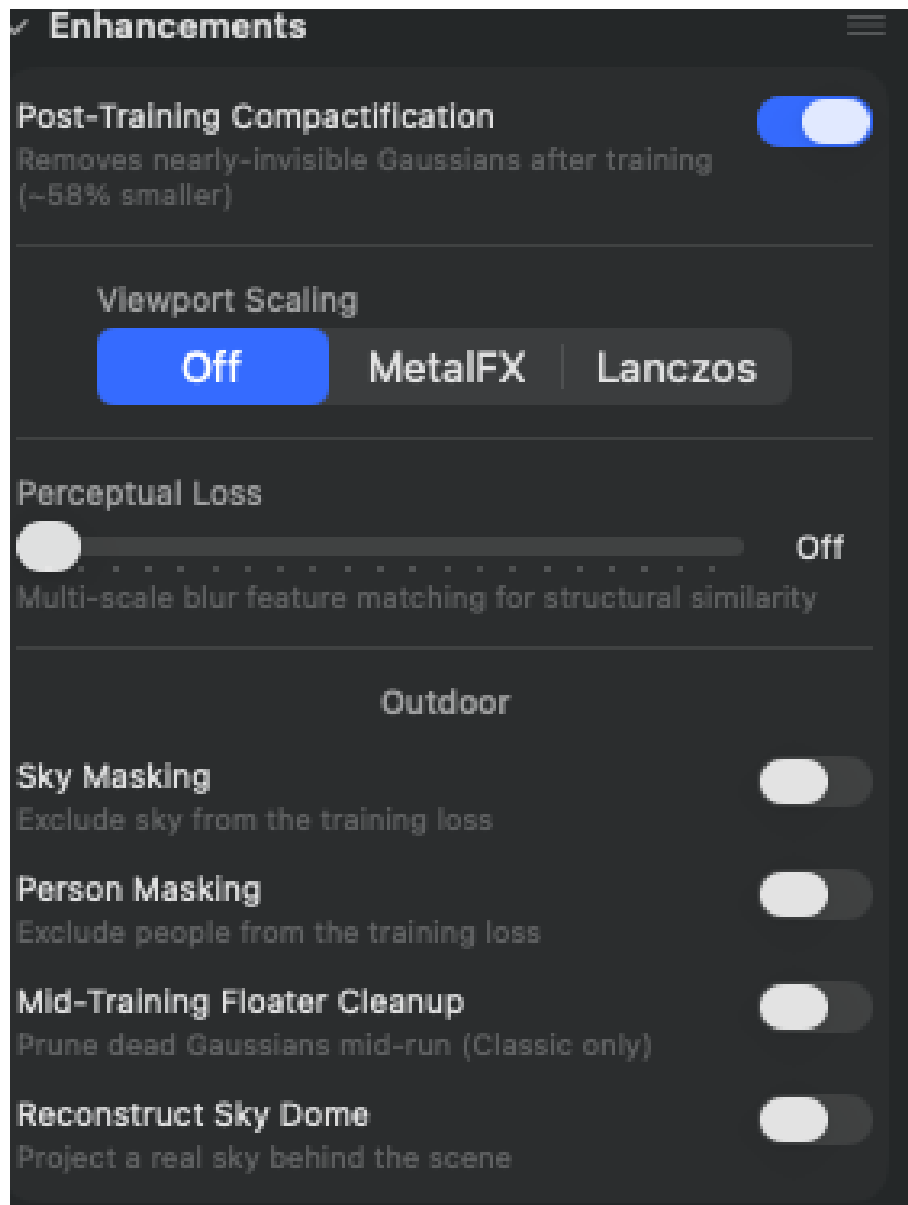


Figura 12: Recorte só da seção Enhancements — três linhas: Post-Training Compactification (toggle ligado), Viewport Scaling (seletor segmentado Off/MetalFX/Lanczos), Perceptual Loss (slider em „Off“). Cada linha com subtítulo explicando a função

A seção Enhancements agrupa três features que melhoram a qualidade de imagem sem mudar o loop central de treinamento. As duas primeiras (I26–I27) são **pós-treinamento** ou **estágios de viewport**: Compactification limpa após o treinamento; o Viewport Scaling é um puro renderer de viewport, sem efeito sobre o treinamento em andamento. A Perceptual Loss (I29), apesar de estar nesta seção, é parte do treinamento — é ativada durante o treinamento como termo de loss adicional; por isso o divisor a separa dos controles de viewport. A partir da v1.6 a seção também tem um grupo Outdoor (I42–I44: Sky Masking, Mid-Training Floater Cleanup, Reconstruct Sky Dome) — opções de treinamento contra floaters de céu que antes ficavam na janela de Configurações e agora moram aqui por projeto.

I26 Toggle Post-Training Compactification **ONDE**

Inspetor → Enhancements → Post-Training Compactification.

 **TÉCNICO**

Ativa o pós-processamento V443: após o fim das iterações, Gaussians com opacidade abaixo de 0.01 (1 % de visibilidade) são removidos. Empiricamente reduz o tamanho do arquivo em 55–58 % sem perda visual — porque esses Gaussians visualmente não contribuía. Roda como pass GPU de compactificação; duração de frações de segundo a poucos segundos, conforme a contagem. Não afeta performance de treinamento. Com este toggle desligado, Gaussians invisíveis são exportados também — relevante apenas se você pretende usar o formato para outro estágio de treinamento (Continue Training); senão, desperdício de espaço.

 **EM PALAVRAS SIMPLES**

Limpa após o treinamento Gaussians que você não vê de qualquer forma (opacidade abaixo de 1%). Reduz os arquivos de export à metade aproximadamente (55–58 % de redução) sem perda visual. Roda como pass GPU curto após a última iteração, durando de frações a poucos segundos. Praticamente sempre deve ficar ligado — só desligue se quiser depois Continue Training e precisar manter os invisíveis. Nos workflows normais de export, deixe ligado.

I27 Seletor Viewport Scaling ONDE

Inspetor → Enhancements → Viewport Scaling (seletor segmentado com três opções: Off, MetalFX, Lanczos).

 TÉCNICO

Um único seletor segmentado que escolhe o upscaler do viewport — as três opções são **mutuamente exclusivas**. Se a resolução de treinamento (via I22 Render Scale) é menor que o tamanho do viewport, o modo escolhido upscala o frame renderizado para o tamanho de exibição. **Off** = simples esticamento bilinear. **MetalFX** = o Spatial Upscaler baseado em ML da Apple, a opção mais nítida (o modelo ML é otimizado para bordas nítidas), overhead de 1–2 ms por frame em GPUs M3. **Lanczos** = os Metal Performance Shaders da Apple com resampling sinc de 8 taps, clássico sem ML, overhead mínimo (<0.5 ms), qualidade abaixo do MetalFX, mas sem o „suavizar“ típico de ML em estruturas finas de linha. O pipeline do renderer é reconfigurado ao vivo ao alternar — visível imediatamente, sem reiniciar. **Histórico:** antes eram dois toggles separados (MetalFX + Lanczos) que podiam estar ligados ao mesmo tempo — um estado contraditório em que MetalFX silenciosamente passava por cima do Lanczos. O seletor remove esse estado; um eventual estado „ambos ligados“ herdado de sessões antigas se autocura para MetalFX na próxima troca. Atua **só** no live viewport, não em exports renderizados (orbit video, screenshots) — esses vão em resolução de fonte plena.

 EM PALAVRAS SIMPLES

Aprimora o live viewport — particularmente útil quando você trabalha com resolução reduzida de treinamento (Render Scale 50 %, veja I22). Três níveis, dos quais só um fica ativo: „Off“ simplesmente estica os pixels, „MetalFX“ usa o Machine Learning da Apple para as bordas mais nítidas (praticamente sempre a melhor escolha), „Lanczos“ é o filtro clássico sem ML — use esse como fallback caso o MetalFX suavize linhas ou mostre artefatos em alguma cena. Atua ao vivo, sem reiniciar. Funciona só no live viewport, não em orbit videos ou screenshots exportados — esses vão em resolução de fonte plena. Diferente de antes, você não consegue mais escolher dois modos ao mesmo tempo por engano.

I29 Slider Perceptual Loss ONDE

Inspetor → Enhancements → Perceptual Loss.

 TÉCNICO

Slider 0.0–0.2 em passos de 0.01; exibe em 0.0 como „Off”, senão como „0.05” etc. Ativa um termo de loss adicional que compara blur Gaussian multi-escala do render com a imagem ground truth (3 escalas de blur). Captura diferenças estruturais que L1+SSIM sozinhos não pegam. Implementação V460. Empiricamente, 0.05–0.1 melhora o L1 em alguns por cento, mas custa 5 % de tempo de treinamento (forward pass extra pelos kernels). Acima de 0.15, o treinamento fica instável e o L1 piora (o termo domina a otimização). Atua **durante** o treinamento, não no pós-processamento — apesar da posição em „Enhancements”, não é um realce posterior.

 EM PALAVRAS SIMPLES

Um termo de loss adicional que checa similaridade estrutural em três níveis de blur. Ajuda especialmente em cenas com estruturas finas como cabelos, tecido ou vegetação, porque pega padrões que L1+SSIM não veem. Valores pequenos são mais seguros — 0.05 a 0.1 é o sweet spot; acima de 0.15 o treinamento fica instável e o loss piora. Em 0 (Off), a função está totalmente desligada e não custa nada; ativa consome 5 % de tempo de treinamento extra pelos kernels. Apesar de estar em „Enhancements”, atua durante o treinamento, não só no pós.

I42 Sky Masking ONDE

Inspetor → Enhancements (grupo Outdoor) → toggle „Sky Masking“. Bound: `AppState.trainingConfig.skyMaskingEnabled` (por projeto, `@DefaultFalse`). Default: `false` .

 TÉCNICO

Ativa a segmentação pré-treinamento de pixels de céu baseada em Apple Vision. Antes do início do treinamento, a região do céu é extraída para cada câmera de entrada via Apple Vision foreground mask (Sky = Background) e atribuída como máscara por pixel à respectiva câmera. Durante o treinamento, a contribuição de loss por pixel é multiplicada pelo complemento da máscara de céu — pixels de céu contribuem com 0 para o gradiente, de modo que Gaussians que projetam no céu não recebem sinais de otimização e portanto não ficam mais „densos“ ou „mais brilhantes“. Reduz significativamente os floaters (grumos escuros no céu) em cenas externas/de drone. Custa ~3% de regressão de L1 no treinamento Classic de 40K (ver [memory/dev_outdoor-floater-reduction.md](#)). Só faz sentido em cenas externas com céu claramente reconhecível; em cenas internas ou com fundo branco, a segmentação de céu identifica áreas erradas e bloqueia sinais de loss válidos. O valor agora é armazenado por projeto (não mais global no app) e acompanha a predefinição / o arquivo de cena.

 EM PALAVRAS SIMPLES

Em fotos externas com céu na imagem, costumam surgir grumos pretos ou coloridos no céu — os chamados „floaters“. Esta opção detecta automaticamente onde está o céu e diz ao treinamento: „Deixa o céu em paz.“ Funciona muito bem em voos de drone e cenas de paisagem. Para cenas internas ou fundos escuros, pode piorar a imagem — então só ligue quando houver céu de verdade na cena. Detalhes: [memory/dev_outdoor-floater-reduction.md](#).

I43 Mid-Training Floater Cleanup



Inspetor → Enhancements (grupo Outdoor) → toggle „Mid-Training Floater Cleanup“. Bound: `AppState.trainingConfig.floaterCleanupEnabled` (por projeto, `@DefaultFalse`). Default: `false` .



Liga, no treinamento Classic 40K (preset „P4 Quality“), dois passes adicionais de controle de densidade: na iteração 20,000 e na iteração 30,000. Ambos varrem todos os Gaussians por três critérios: (a) opacidade muito baixa (padrão 0.005), (b) tamanho ínfimo em screen-space, (c) sem contribuição de loss nas últimas 1000 iterações. Gaussians que satisfazem todas as três condições são removidos. Efeito: ~5–15% menos Gaussians ao fim do treinamento, visivelmente menos grumos escuros no céu em cenas de drone/externas. Custa ~1–3% de regressão de L1 em cenas internas em close-up, por isso não é ativado por padrão. As duas iterações de limpeza (20K, 30K) são fixas e atualmente não podem ser alteradas via UI; em treinamentos mais curtos (p. ex. P2 Preview 5K) o toggle não tem efeito, pois nunca alcança essas marcas de iteração. **Novo:** o toggle só fica utilizável quando o preset ativo usa o densificador **Classic** (`densificationStrategy == .classic`). Em MCMC ou Hybrid ele fica **disabled** e aparece um aviso inline, porque essas estratégias já lidam com Gaussians mortos por conta própria (MCMC via relocation, Hybrid via lógica combinada de reloc/noise) — ali os passes manuais de limpeza seriam ineficazes ou até contraproducentes. Referência de código: `RadianceKitApp.swift`, aba General. Detalhes: `memory/dev_outdoor-floater-reduction.md`.

EM PALAVRAS SIMPLES

Durante o treinamento, às vezes surgem Gaussians „mortos“ que não contribuem mais para a qualidade da imagem, mas ocupam memória. Esta opção limpa duas vezes ao longo de um treinamento longo (em 20K e 30K iterações) e remove esses cadáveres. Em cenas externas com céu, isso é especialmente útil porque os floaters se acumulam mais ali. Em treinamentos pequenos ou em closes de móveis, nem tanto. A chave só pode ser ligada quando seu preset usa o densificador Classic — em presets MCMC ou Hybrid ela fica acinzentada (com uma breve explicação), porque eles limpam seus pontos mortos sozinhos.

I44 Reconstruct Sky Dome **ONDE**

Inspetor → Enhancements (grupo Outdoor)
→ toggle „Reconstruct Sky Dome“. Bound:
AppState.trainingConfig.skyDomeEnabled (por pro-
jeto, @DefaultFalse). Default: false .

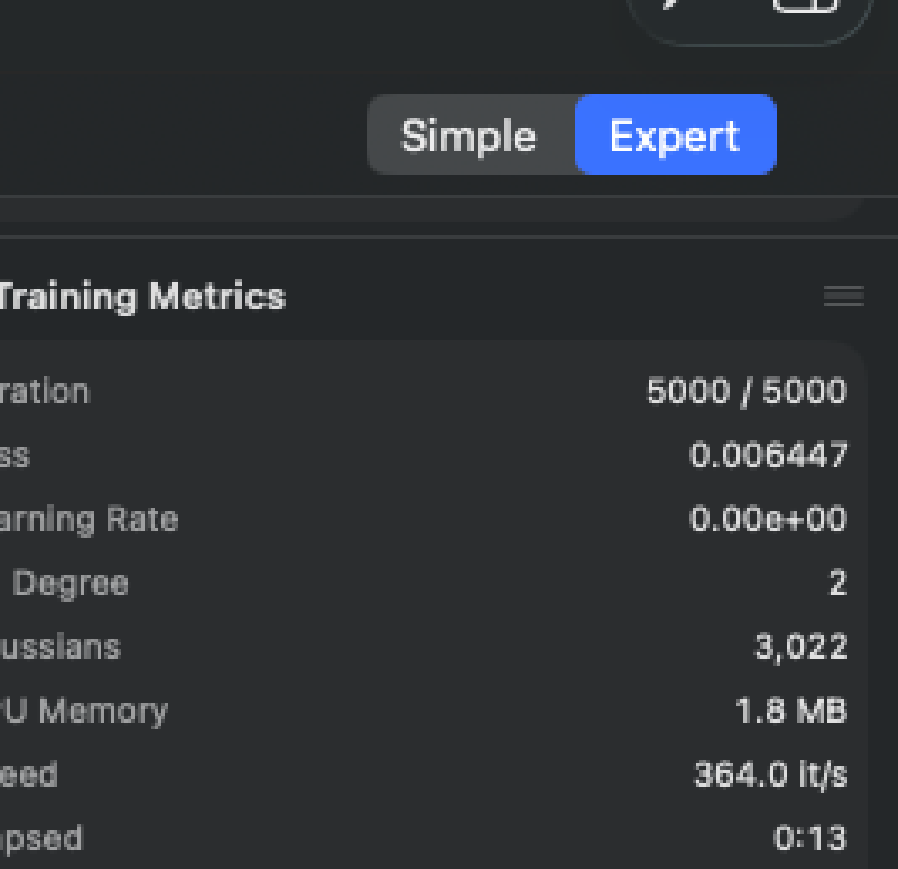
 **TÉCNICO**

Ativa a projeção de Sky Dome pré-treinamento (V549e MVP). Depois do SfM e antes do início do treinamento, a máscara de céu Apple Vision compartilhada com I42 é extraída da imagem para cada câmera de entrada, os pixels de céu são des-projetados com as intrínsecas da câmera para uma superfície esférica virtual (raio padrão 8x o raio da cena). Nessa esfera, ~5000 novos Gaussians são inicializados com médias de cor dos pixels de céu projetados, escala muito grande (1.0 em unidades de cena) e opacidade inicial 0.95. Esses 5000 Gaussians não são uma máscara de céu no sentido clássico — eles são treinados como todos os outros, mas a alta opacidade inicial os mantém numa casca fina. Resultado: em vistas novas 360° em cenas externas/de drone aparecem cor de céu real e estruturas de nuvens em vez de grumos escuros tipo confete. Só faz sentido em cenas externas com pelo menos 360° de cobertura de câmera; em capturas puras de objeto sem visão de céu não tem efeito. Status: experimental, a validação A/B mais ampla em outros conjuntos externos ainda está pendente.

 **EM PALAVRAS SIMPLES**

Em vez de o treinamento tentar „adivinhar“ o céu a partir dos poucos pixels visíveis (o que leva a floaters), o RadianceKit projeta os pixels de céu diretamente em uma esfera virtual ao redor da cena antes do treinamento começar. Quando você gira a cena pronta em 360°, vê céu de verdade em vez de grumos pretos. Funciona só em fotos externas em que há realmente céu na imagem. Em escaneamentos de sala ou estúdio, não traz nada.

Seção Métricas (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

Figura 13: Recorte só da seção Training Metrics após conclusão do treinamento no buquê (5K iterações, 2 991 Gaussians final) — tabela com métricas de treinamento (Iteration, Loss, SSIM Loss, Combined Loss, Gaussian Count, Learning Rate, Elapsed, ETA)

Enquanto um treinamento roda, a seção de métricas mostra nove valores ao vivo do loop. Antes de iniciar, a seção está vazia („Start training to see live metrics“). Todos os valores são atualizados a cada 30 iterações (frequência do stream). A seção é read-only — nenhum elemento é clicável ou alterável. Para análise mais profunda, recorra aos logs JSONL de treinamento em `~/Documents/RadianceKit/Logs/` (script `python3 scripts/analyze_logs.py best 5`).

I30 Iteration

ONDE

Inspetor → Métricas → Iteration. Read-only.

TÉCNICO

Exibição no formato „4523 / 40000” — iteração atual sobre o total planejado. Conta em sincronia com o loop, que empurra valores a cada 30 iters. A segunda corresponde ao valor de Max Iterations no instante de início; não muda mais, mesmo se o usuário alterar o stepper — a corrida em andamento usa snapshot próprio. Se Continue Training (+5K/+10K/+20K) for usado, o denominador aumenta.

EM PALAVRAS SIMPLES

Onde o treinamento está. „4523 / 40000” significa: 4523 de 40 000 passos passaram, 11 % feito. A esquerda cresce a cada segundo; se ficar parada minutos, o treinamento travou — normalmente throttling de GPU ou app concorrente. A da direita corresponde a Max Iterations (I18) no início e não muda mais, mesmo se você ajustar o stepper. Em Continue Training (+5K/+10K/+20K), ela cresce com os passos adicionais.

I31 Loss

ONDE

Inspetor → Métricas → Loss. Read-only.

TÉCNICO

Float com seis casas decimais (p. ex. „0.024385”). Mede o L1+SSIM combinado (mix por I21 SSIM Weight) mais opcionalmente Perceptual Loss (I29) e outros regularizadores. A escala não é absoluta, depende da cena — para comparações exige o mesmo dataset. Valores finais típicos em boas configurações:

- Classic Quality 40K iters: 0.022–0.025 (Horse, Truck, Garden)
- MCMC Full 200K iters: 0.024–0.028
- Outdoor drone 30K: 0.030–0.060 (pior por geometria)
- Indoor apartamentos: 0.018–0.025

Valores acima de 0.10 após 5K iters indicam problemas de SfM (poses ruins) — abortar e recalcular.

EM PALAVRAS SIMPLES

Quão longe a imagem renderizada ainda está do original — combinação de L1, SSIM e eventualmente Perceptual. Menor é melhor. Abaixo de 0.03 costuma ser bem bom, abaixo de 0.05 ok; cenas externas ficam por questão de geometria em 0.03–0.06. Acima de 0.10 após muitos milhares de iters é sinal de alerta — em geral a reconstrução de câmeras (SfM) não foi limpa. A escala não é absoluta, é dependente da cena; comparar só dentro do mesmo dataset. Se a número de repente sobe muito, provavelmente um evento de explosão de gradiente.

I32 Learning Rate



ONDE

Inspetor → Métricas → Learning Rate. Read-only.



TÉCNICO

Exibição em notação científica (p. ex. „1.60e-04”). Learning rate atual dos parâmetros de posição (3DGS tem seis LR independentes para posição, SH-DC, SH-Rest, opacidade, escala, rotação — a de posição é mostrada como representativa). Valor inicial padrão 1.6e-4, com decaimento exponencial até 1.6e-6 no fim. O decaimento é ajustável pelo campo LR Schedule na configuração de treinamento (campo T no Capítulo 6). Se o LR continuar incomumente alto (p. ex. 1e-3 ou mais depois de 10K iters), pode indicar config carregada errada.

EM PALAVRAS SIMPLES

Quão grandes são os passos de otimização agora — mais especificamente a learning rate das posições dos Gaussians. Começa em 1.60e-04 e decai exponencialmente para 1.60e-06 ao fim („1.60e-06” = 0.0000016). O decaimento roda automaticamente; você não ajusta aqui. Se depois de 10 000+ iters ainda for maior que 1e-3, provavelmente uma config errada foi carregada — abortar e escolher predefinição. Internamente, 3DGS tem seis LR independentes (Posição, SH-DC, SH-Rest, Opacidade, Escala, Rotação); aqui você só vê a de posição como representativa.

I33 SH Degree



ONDE

Inspetor → Métricas → SH Degree. Read-only.



TÉCNICO

Inteiro 0–3. Grau de spherical harmonics da representação de cor. Começa em 0 (só DC, ou seja, cor independente de direção — só uma constante RGB por Gaussian) e sobe ao longo do treinamento até 3. O schedule padrão sobe 1 a cada 1000/2000/3000 iters. SH-3 corresponde a 48 coeficientes de cor por Gaussian (3 canais × 16 funções base SH). Grau mais alto significa mais reflexão dependente de direção (superfícies brilhantes parecem corretamente diferentes em ângulos diferentes), mas também mais memória e treinamento mais lento.

EM PALAVRAS SIMPLES

Quão complexa é a representação de cor por Gaussian no momento. Começa em 0 (uma cor independente de direção por ponto) e sobe em degraus até 3 — tipicamente nas iters 1000, 2000 e 3000. O grau 3 dá 48 coeficientes de cor por Gaussian e permite reflexões dependentes de direção, ou seja, superfícies brilhantes parecem diferentes em ângulos diferentes. Não precisa mexer; o schedule é automático. Grau maior consome mais memória e desacelera um pouco o treinamento — preço por highlights realistas.

I34 Gaussians



ONDE

Inspetor → Métricas → Gaussians. Read-only.



TÉCNICO

Número atual de Gaussians no modelo, formatado com separador de locale (p. ex. „524.318”). Crescimento:

- Classic: começa nos pontos iniciais SfM (50K–300K), cresce

por clone/split até antes do Densify Until, depois estático até o fim (modulo pruning).

- MCMC: amostras adicionadas até o cap MCMC, depois só relocação.

Valores finais saudáveis:

- Classic Quality: 400K–700K (Horse 524K, Garden 800K)
- MCMC Full: exatamente no cap (padrão 150K, com auto-scale

multiplier × contagem SfM, dependendo da cena 500K–1.5M)

Em MCMC, queda abaixo de 60 % do cap → anomalia (indicador de collapse; aponta regularizadores agressivos demais).

EM PALAVRAS SIMPLES

Quantos pontos Gaussian o modelo tem agora. Cresce durante o treinamento até Densify Until (I20); depois fica quase constante. Mais pontos = mais detalhe, mas arquivo maior e renderização mais lenta no viewport. 500.000 é um valor médio típico para Classic Quality em cena média; MCMC Full chega a 500K–1.5M conforme Auto-scale (I17). Se em MCMC cair abaixo de 60 % do cap, é indicador de collapse — em geral regularizadores agressivos demais.

I35 GPU Memory



ONDE

Inspetor → Métricas → GPU Memory. Read-only.



TÉCNICO

Estimativa do consumo do buffer de Gaussians como contagem × 616 bytes (formatado em memory style). 616 bytes é o tamanho empírico de um Gaussian completo (posição, escala, rotação, opacidade, coeficientes SH grau 3, acumulador de gradiente). A exibição **não** inclui overhead do renderer (tile buffer, sort buffer, backward buffer) — o consumo real de GPU costuma estar 2–3× acima. Em 500K Gaussians: exibido 290 MB, real 700 MB. Em 1.5M: exibido 880 MB, real 2.5 GB. Em M3 Max com 64+ GB de unified memory, não crítico; em M3 Pro com 18 GB, já no limite.

EM PALAVRAS SIMPLES

Estimativa de quanto os Gaussians ocupam em GPU — cerca de 616 bytes por ponto. O consumo real fica 2–3× acima por causa de tile/sort/backward buffers do renderer. Em MacBook com 16–18 GB de unified memory, fique abaixo de 500K Gaussians; com M3 Max ou Studio (64+ GB), 1.5M ou mais tranquilo. Se o treinamento crashar do nada ou o sistema fizer swap, geralmente é aqui que bateu — baixe Render Scale (I22) ou reduza Densify Until (I20).

I36 Speed

ONDE

Inspetor → Métricas → Speed. Read-only.



TÉCNICO

Iterações por segundo com uma casa decimal („24.3 it/s“). Calculado pelo trainer como média móvel das últimas 100 iters. Valores típicos:

- Preset Quick (1K iters): 80–120 it/s (curto, sem steady state)
- Classic 20K @ 1.0 Render Scale (cena Truck, 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 (mais lento por

relocação)

- Com 1M+ Gaussians e resolução plena: <10 it/s

Velocidade caindo ao longo do treinamento é normal — mais Gaussians = mais compute por iter.

Quedas abruptas (p. ex. de 30 para 5 it/s) indicam thermal throttling de GPU ou apps concorrentes.

EM PALAVRAS SIMPLES

Quão rápido o treinamento roda, em iterações por segundo. Tipicamente 20–50 it/s; com Render Scale reduzido (50%) e cenas pequenas, até 80–120 it/s. Cai naturalmente ao longo do treinamento — mais Gaussians = mais trabalho por iter. Quedas abruptas (30 → 5 it/s) indicam thermal throttling de GPU ou apps concorrentes — tabs de browser com vídeo, backup do Time Machine, indexação do Photos. Manter o app em primeiro plano e fechar background ajuda. Com 1M+ Gaussians e resolução plena, abaixo de 10 it/s é normal.

I37 Elapsed

ONDE

Inspetor → Métricas → Elapsed. Read-only.



TÉCNICO

Tempo já decorrido como „4:23“ (m:ss) ou „1:23:45“ (h:mm:ss). Troca de formato a partir de 1 hora. Mede apenas o tempo de treinamento, não as fases anteriores (cálculo SfM, import de imagens). Em pause/resume o relógio continua — é wall clock, não CPU time.

EM PALAVRAS SIMPLES

Quanto tempo o treinamento já roda, em cronômetro puro (wall clock). Formato „m:ss“ até uma hora, depois „h:mm:ss“. Não é „tempo de CPU“, e sim „quanto já esperamos“ — pauses contam também. Mede só o treinamento, não SfM e import. Útil para comparar com a ETA (I38) — se Elapsed ultrapassa muito a ETA original, alguma coisa ficou mais lenta.

I38 ETA

ONDE

Inspetor → Métricas → ETA. Read-only.

TÉCNICO

Tempo estimado restante como „17:42” ou „1:12:35”.
Cálculo: $(\text{Max Iterations} - \text{iter atual}) / \text{iters por segundo}$. Mostra „-” quando a velocidade é zero (logo no início ou em pause). A estimativa **não** é ajustada para a desaceleração típica perto do fim — em MCMC e Classic com Densify Until grande, o treinamento tende a ficar mais lento porque mais Gaussians entram em cena. Real geralmente fica 10–20 % acima da ETA inicial.

EM PALAVRAS SIMPLES

Quanto provavelmente ainda vai esperar — calculado pelos iters restantes e pela Speed atual (I36). Estimativa grosseira: se o Mac desacelera (mais Gaussians na fase densify, thermal throttling, outros apps), pode demorar mais que o exibido. O app não embute a desaceleração típica do fim, então o fim real costuma ficar 10–20 % acima da ETA inicial. Adicione 15 % e dá certo na maioria das vezes. Mostra „-” quando a velocidade é zero (começo ou pause).

Seção Gráfico de Loss (I39–I41)

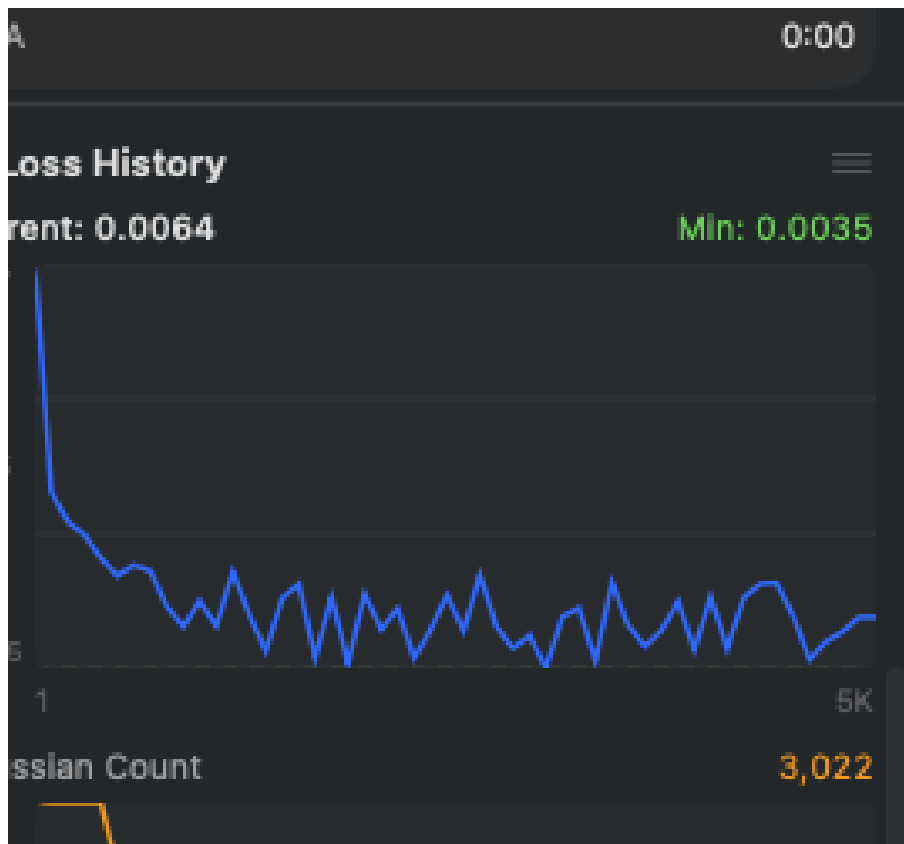


Figura 14: Recorte só da seção Loss History após conclusão — Current 0.0064, Min 0.0035 (verde), curva azul de 0.027 (iteração 1) a 0.0035 (iteração 5K) com curvatura característica perto da iter 200, abaixo o Gaussian Count chart em laranja

A seção do gráfico de loss visualiza o curso do treinamento ao longo do tempo. Consiste em dois charts: um grande, em cima, azul, com a curva de loss; e um menor, embaixo, laranja, com a contagem de Gaussians. Ambos são construídos ao vivo e permanecem até o próximo início. Antes do primeiro treinamento, a área está vazia („Loss curve will appear during training“). Os charts são desenhos SwiftUI Path puros (sem framework Swift Charts), para que rodem fluido também com 100K+ pontos.

I39 Current Loss (exibição)



ONDE

Inspetor → Gráfico de loss → label à esquerda „Current: 0.0287“. Read-only.



TÉCNICO

Float do último sample de loss, com quatro casas. Idêntico ao I31 (Loss em métricas), só com formato mais compacto. Fonte é a Loss History — uma lista que ganha entrada a cada 30 iters. Só valores finitos vão para a lista — NaN/Infinity (raro, em bugs de explosão de gradiente) são filtrados.

EM PALAVRAS SIMPLES

O valor atual de loss em forma mais curta que na seção de métricas (quatro casas). Conteúdo idêntico ao I31, mas a exibição fica colada ao chart e dá o número exato ao observar a curva. Atualizado, como todas as métricas ao vivo, a cada 30 iters. Valores NaN ou Infinity (raros em bugs de gradiente) são filtrados. Útil para não pular para outra seção ao olhar o gráfico.

I40 Min Loss (exibição)



ONDE

Inspetor → Gráfico de loss → label à direita „Min: 0.0245“ (verde). Read-only.



TÉCNICO

Mínimo de todos os loss vistos no run atual. É recomputado ao vivo da Loss History — sem persistência separada. Exibido em verde porque „Min“ = „best so far“. A linha tracejada verde na borda inferior do chart marca essa Y visualmente. Em sessões Continue Training, o rastreo de mínimo reinicia — a antiga history é substituída pela nova na UI (não apensada). Se a corrida atual rodar pior que a anterior, o Min exibido pode ser maior que o resultado final anterior.

EM PALAVRAS SIMPLES

O loss mais baixo que esse treinamento viu — em verde porque é „best so far“. A linha tracejada verde no fundo do chart marca essa posição. Se a curva atual está bem acima, com sorte ainda melhora; em geral o Min é o indicador do fim que interessa depois. Em Continue Training, o rastreo reinicia, pois a antiga history é substituída — o Min pode parecer pior que o resultado anterior.

I41 Gaussian Count Chart



ONDE

Inspetor → Gráfico de loss → segundo chart embaixo (laranja). Read-only.



TÉCNICO

Gráfico de linha do número de Gaussians ao longo das iters. Fonte: a Gaussian Count History (lista de pares (iter, count) preenchida pelo trainer a cada 30 iters). Escala Y dinâmica entre mínimo e máximo da history. Em Classic, a curva sobe firme até Densify Until; depois fica plana (com pequenas oscilações de pruning). Em MCMC: sobe íngreme até o cap, depois fica horizontal (relocação mantém constante). Se a curva **cai** apesar do treinamento ativo, a densification está prunindo agressivo demais — indício de padrões errados ou do bug conhecido MCMC collapse (tema do hotfix v1.4.4).

EM PALAVRAS SIMPLES

Como o número de Gaussians evolui ao longo do treinamento — o chart laranja menor sob a curva de loss. Em Classic, sobe firme até Densify Until (I20); depois fica plano com pequenas oscilações de pruning. Em MCMC, sobe íngreme até o cap e fica horizontal, porque a relocação mantém o número fixo. Se a curva, apesar do treinamento ativo, de repente cai, a densification está prunindo agressivo demais — sinal clássico do bug MCMC collapse da v1.4.4. Aí, atualize o app ou volte para Classic.

Como ler a curva de loss?

O loss chart é a ferramenta diagnóstica mais importante do Inspetor — nenhum outro indicador mostra tão diretamente se o treinamento avança útil ou está travado. A forma típica saudável é uma queda rápida nas primeiras 1000–3000 iters (de 0.15 para 0.05), seguida de queda lenta e uniforme até o fim (0.020–0.030). Em escala log, a curva parece uma diagonal suave.

O que significa um plateau no loss? Se a curva fica plana por vários milhares de iters, há duas leituras possíveis: (a) o treinamento „convergiu“ — o loss não pode mais cair significativamente, o modelo é o melhor possível com esses dados e configurações. Isso é desejado; é „pronto“. (b) o treinamento „travou“ — o loss poderia cair, mas a otimização estagna (mínimo local, learning rate pequeno, densification desligada). Para diferenciar: se o loss está em faixa boa típica (0.020–0.030 indoor/objeto, 0.040–0.060 outdoor) e a curva está plana há 5K+ iters, convergiu. Se o valor está bem acima de cenas comparáveis (p. ex. 0.08), travou.

Atenção: plateau de Gaussians ≠ plateau de loss. Um plateau na contagem de Gaussians **não** significa „treinamento pronto“. Significa só que a densification parou de adicionar pontos — ou porque foi atingido (Classic) ou porque o cap MCMC encheu. O treinamento continua refinando os existentes. O sinal real de „pronto“ você lê na curva de loss e na exibição de iteração (I30), não aqui.

Regra prática para abortar: se a curva de loss após 5000+ iters ainda fica acima de 0.08 e mal cai, é alta a chance de a reconstrução SfM estar errada. Aborte, consulte o Capítulo 9 se o backend escolhido combina com a cena, troque para COLMAP/Native se necessário e reinicie. Melhor investir 10 minutos em SfM melhor do que 2 horas de treinamento com alinhamento ruim.

Quando recorrer ao Inspetor?

Referência rápida: qual seção + quais controles para qual caso de uso típico?

Tarefa comum	Seção	IDs de controle
Dessaturar as cores do Splat finalizado	Look	L1 (Saturation)
Arredondar Splats agulha/confete	Look	L2 (Splat length)
Preencher uma nuvem esburacada / aumentar Splats	Look	L3 (Splat size)
Desvanecer „far-confetti“ distante em órbitas	Look	L4 (Fade far region)
Descartar ajustes de Look	Look	L5 (Reset finishing)
Carregar setup pronto	Predefinições	I7 (clicar linha)
Salvar próprio setup	Predefinições	I1 → I2 → I4
Compartilhar setup com colegas	Predefinições	I5 (Export) ou I6 (Import)
Trocar backend SfM (p. ex. Apple-PG instável)	Configuração de treinamento	I12 (ver Cap. 9)
Processar frames de vídeo sem EXIF	Configuração de treinamento	I13 (FOV Override)
Performance COLMAP: GLOMAP em vez de clássico	Configuração de treinamento	I14
Trocar de Classic para MCMC	Configuração de treinamento	I15
Estender treinamento	Configuração de treinamento	I18 (Max Iter) + I20 (Densify Until) — ligados via I19
Cortar GPU pela metade	Configuração de treinamento	I22 (Render Scale 50 %)
Qualidade +6 % (MCMC)	Configuração de treinamento	I16 (MCMC Quality)
Cena externa com muitos pontos SfM	Configuração de treinamento	I17 (Auto-scale by scene)
Configurar/trocar caminho COLMAP	Configuração de treinamento	I23 / I24 / I25
Tornar arquivos de export menores	Enhancements	I26 (sempre ligado)
Viewport mais nítido sem mais tempo de treinamento	Enhancements	I27 (Viewport Scaling → MetalFX)
MetalFX suaviza demais → alternativa	Enhancements	I27 (Viewport Scaling → Lanczos)
Último ganho de detalhe em	Enhancements	I29 (Perceptual Loss 0.05–0.1)
Desatenuar o fundo para melhor	Clustering	I30 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I31 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I32 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I33 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I34 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I35 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I36 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I37 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I38 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I39 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I40 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I41 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I42 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I43 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I44 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I45 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I46 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I47 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I48 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I49 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I50 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I51 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I52 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I53 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I54 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I55 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I56 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I57 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I58 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I59 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I60 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I61 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I62 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I63 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I64 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I65 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I66 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I67 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I68 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I69 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I70 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I71 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I72 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I73 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I74 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I75 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I76 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I77 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I78 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I79 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I80 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I81 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I82 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I83 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I84 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I85 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I86 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I87 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I88 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I89 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I90 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I91 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I92 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I93 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I94 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I95 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I96 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I97 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I98 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I99 (Background Splat)
Desatenuar o fundo para melhor	Clustering	I100 (Background Splat)

CAPÍTULO

Capítulo 3 — Configurações

A janela de Configurações se abre via `RadianceKit` → `Configurações...` ou pelo atalho padrão `⌘, .` Ela contém duas abas: **General** e **AI Helpers**. Diferente dos valores do Inspetor do Capítulo 2, as configurações desta janela atuam de forma **global no app** (em todos os projetos) — elas são persistidas em e sobrevivem a reinicializações do app. A aba `General` agrupa três seções de conteúdo: `Interface`, `Viewport` e `Treinamento`. (Os três toggles de `Outdoor Floater` — `Sky Masking`, `Mid-Training Floater Cleanup`, `Reconstruct Sky Dome` — que antes ficavam aqui foram movidos, a partir da v1.6, para a seção `Enhancements` do `Expert Inspector`, onde agora são armazenados por projeto; ver Capítulo 2, I42–I44.) A aba `AI Helpers` ativa os auxiliares de machine learning on-device (`Vision`, `CoreML`) para pré-processamento de SfM e treinamento.

Controles anteriores para ativar ou desativar coletivamente todos os `AI Helpers` não existem mais na versão atual — portanto não são documentados aqui. A antiga área „Coming Soon“ para auxiliares ainda não lançados também foi removida e não é referenciada aqui.

Aba General

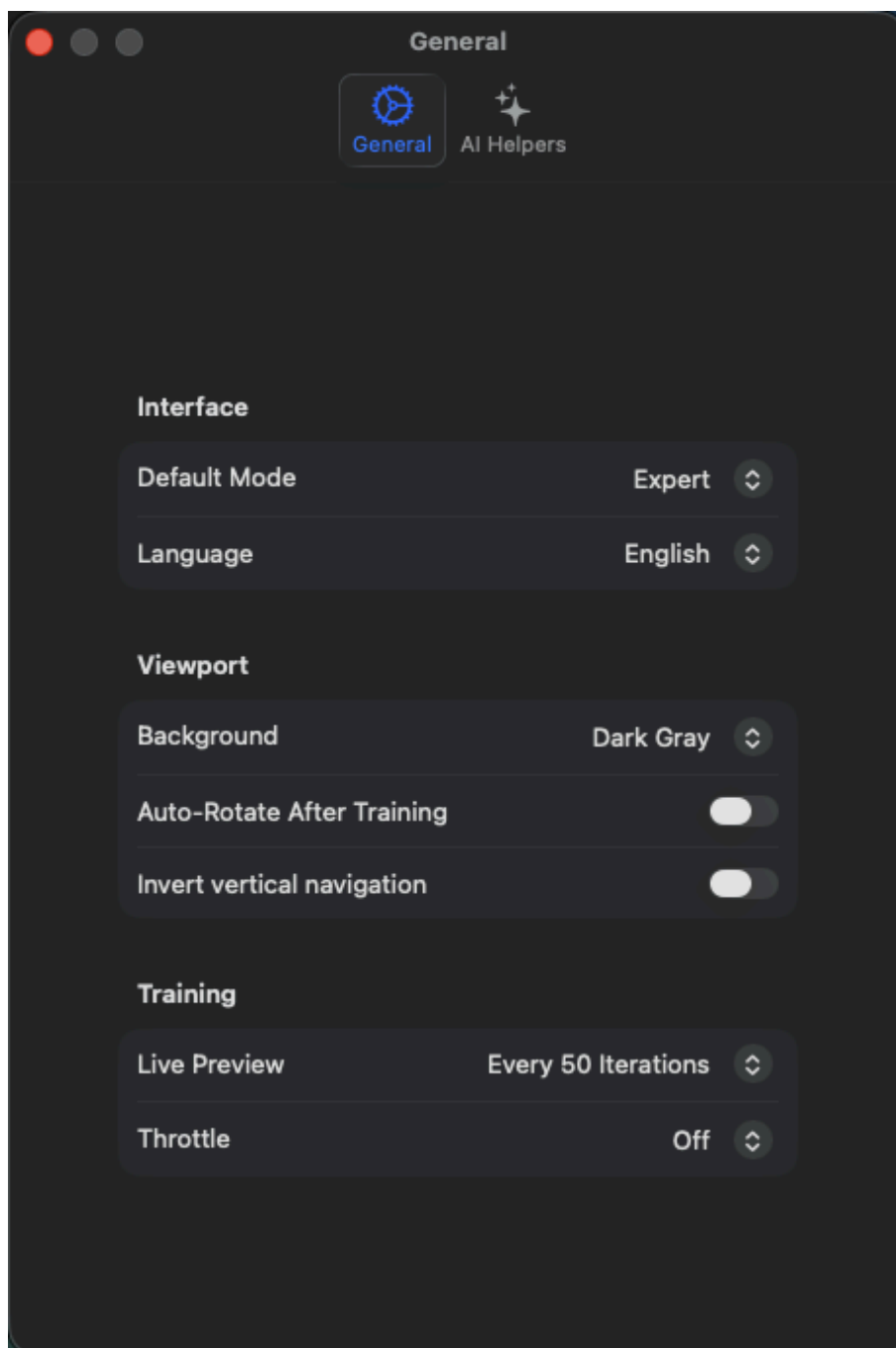


Figura 15: Configurações → aba General com Interface, Viewport, Treinamento e seção Experimental

S1 Default Mode

ONDE

Settings → General → Interface → seletor Default Mode. Bound: Default: `.simple`.

TÉCNICO

Controla em qual dos dois modos de interface o app abre após o próximo início. „Simple Mode” é o fluxo guiado em assistente de 4 passos (Importar → Processamento → Pré-visualização → Exportar, documentado no Capítulo 10 sob Z1–Z4), „Expert Mode” é o layout clássico de três painéis com Navigator, Viewport 3D e Expert Inspector do Capítulo 2. O valor é lembrado entre reinicializações. Efeito idêntico ao menu Mode → Simple Mode (⌘1) / Mode → Expert Mode (⌘2), apenas que o menu alterna a sessão em execução, enquanto este seletor define o padrão para sessões futuras. Ambos os modos acessam o mesmo estado de projeto — projetos, câmeras e configuração de treinamento são preservados ao trocar de modo. Botões da barra de ferramentas específicos do modo são renderizados imediatamente.

EM PALAVRAS SIMPLES

Aqui você escolhe com qual interface o RadianceKit inicia no próximo lançamento. „Simple Mode” é o modo iniciante: quatro passos claros, predefinições estipuladas, quase nenhuma opção. „Expert Mode” é o layout completo de caixa de ferramentas com todos os controles que você vê no Capítulo 2. Você pode alternar a qualquer momento pelo menu „Mode”, sem perder imagens nem progresso de treinamento.

S2 Language

ONDE

Settings → General → Interface → seletor Language. Bound: Default: `.system` (segue o idioma do macOS).

TÉCNICO

Seleciona o idioma de exibição de toda a interface do app, independentemente do idioma do sistema macOS. O RadianceKit é localizado em 17 idiomas (`de`, `en`, `pl`, `en-AU`, `ar-SA`, mais 12 outros). Em „System“, o app segue o idioma do macOS. Com uma escolha explícita, a configuração de idioma é lembrada entre reinicializações; o efeito completo geralmente exige um reinício do app, porque os bundles de localização são carregados apenas na inicialização. As 298 chaves de localização documentadas no projeto são todas consideradas, incluindo todos os textos em sub-views e tooltips de ajuda.

EM PALAVRAS SIMPLES

Se o seu Mac roda em inglês mas você prefere a interface do RadianceKit em português (ou vice-versa), você ajusta isso aqui. A maioria dos textos troca imediatamente. Alguns diálogos aparecem no novo idioma só depois de reiniciar o app.

S3 Viewport Background

ONDE

Settings → General → Viewport → seletor Background. Bound: Default: `.darkGray` (RGB 0.1, 0.1, 0.1).

TÉCNICO

Define a cor de fundo padrão do viewport 3D. Três opções: „Dark Gray“ (RGB 0.1, 0.1, 0.1 — padrão), „Black“ (0, 0, 0) e „White“ (1, 1, 1). A configuração persiste o padrão para novos projetos e sessões entre reinicializações e ao mesmo tempo atualiza o renderer Metal em execução imediatamente. Opções idênticas também estão no menu Viewport → Background (M21, M22, M23), mas o seletor das Configurações define o padrão, enquanto o menu alterna a exibição em execução. Importante para capturas de tela e vídeos demo: fundos brancos realçam mais floaters verdes/azuis, fundos escuros são melhores para gravações limpas de renderização.

EM PALAVRAS SIMPLES

A cor por trás dos seus modelos 3D na janela de pré-visualização. Cinza escuro é o padrão e combina com a maioria das cenas. Branco é bom para capturas de tela, preto fica mais elegante em gravações de render. Você pode trocar a cor a qualquer momento pelo menu „Viewport → Background“ para a cena em execução — esta configuração só define qual cor estará ativa de novo na próxima abertura.

S4 Auto-Rotate After Training

ONDE

Settings → General → Viewport → toggle „Auto-Rotate After Training“. Bound:.. Default: `false` .

TÉCNICO

Inicia uma rotação contínua tipo prato giratório da câmera do viewport ao redor do centroide da cena imediatamente após o fim do treinamento (taxa padrão ~ 0.3 rad/s). Praticamente útil para sessões demo, comparações A/B e para avaliar logo de uma visão 360° se „floaters“ se formaram na borda da cena. O efeito é visualmente idêntico ao menu Viewport → Toggle Auto-Rotation (M16, $\text{⌘} \backslash T$), exceto que este toggle dispara o comportamento automaticamente após o fim do treinamento, em vez de manualmente. Pode ser interrompido a qualquer momento pelo menu ou clicando no viewport (o que pausa a rotação). Não tem influência sobre a performance de treinamento — a rotação só roda depois que o treinamento termina.

EM PALAVRAS SIMPLES

Quando ativado, a cena 3D gira automaticamente assim que o treinamento termina — como um carrossel. Bom para quando você faz treinamento noturno e quer ver o resultado já em movimento de manhã, sem precisar clicar. Em sessões longas em que você só monitora o treinamento, deixe desligado.

S5 Live Preview Interval **ONDE**

Settings → General → Training
→ seletor Live Preview. Bound:
AppState.trainingConfig.livePreviewInterval.
Default: 0 (Off).

 **TÉCNICO**

Determina o intervalo de iterações em que o snapshot de treinamento em andamento é renderizado no viewport 3D. Quatro valores discretos: 0 („Off“), 50, 250, 1000 iterações. Com Live Preview ativo, o treinador copia o buffer de Gaussians da GPU para um buffer de render separado e dispara um redraw do viewport. Com „Off“, o viewport só é atualizado após a conclusão do treinamento. Custo de performance: a cada 50 iterações ~5–10% mais lento em M3 Ultra, a cada 250 iterações ~1–2% mais lento, a cada 1000 iterações imperceptível. Overhead de memória constante ~2 GB para o buffer de snapshot, independentemente do intervalo. O valor serve como padrão para novos treinamentos; após o início do treinamento, o Training Inspector mostra o valor real ao vivo daquele treinamento. No intervalo 50, a impressão visual é uma „cresca“ fluida da nuvem de pontos; em 1000 parece travada.

 **EM PALAVRAS SIMPLES**

Enquanto o treinamento roda, você pode escolher com que frequência a vista 3D é atualizada. „Off“ significa: sem atualização durante o treinamento (mais rápido). „Every 50 iterations“ mostra quase em tempo real como sua cena vai surgindo (um pouco mais lento). Para assistir tranquilamente em pequenos treinamentos, „Every 250“ é um bom meio-termo.

S6 Throttle Delay

ONDE

Settings → General → Training → seletor Throttle.
Bound: `AppState.trainingConfig.throttleDelayMs`.
Default: 0 (Off).

TÉCNICO

Insere um atraso artificial em milissegundos entre as iterações de treinamento. Quatro valores discretos: 0 („Off“), 2 („Light“), 5 („Moderate“), 10 („Eco“). Sentido: em treinamentos mais longos (várias horas), a GPU é normalmente usada a 100%, o que leva a uma UI do sistema visivelmente mais lenta (cursor travando, outros apps ficam pesados). O atraso de throttle dá à GPU pausas em que outras tarefas podem ser executadas. O custo de performance é considerável: com 5 ms de throttle, um treinamento típico de 40K dura cerca de 50–80% mais que sem throttle. No modo de performance „Eco“ (10 ms) o atraso por iteração é maior que a própria iteração — fator 2–3× mais lento. Com throttle ativo, aparece um aviso abaixo do seletor: „Throttle is on. Training will be slower than usual.“ O app em si não reage de forma perceptivelmente melhor — só outros apps se beneficiam.

EM PALAVRAS SIMPLES

Se o seu Mac esquenta demais durante um treinamento longo ou outros programas ficam muito lentos, ligue um freio aqui. „Off“ dá vazão total à GPU (mais rápido). „Light“ faz uma pequena pausa entre cada passo (um pouco mais lento, mas o sistema responde melhor). „Eco“ é o freio mais forte — bom para treinamentos noturnos em um MacBook que não devem ficar quentes demais.

Aba AI Helpers

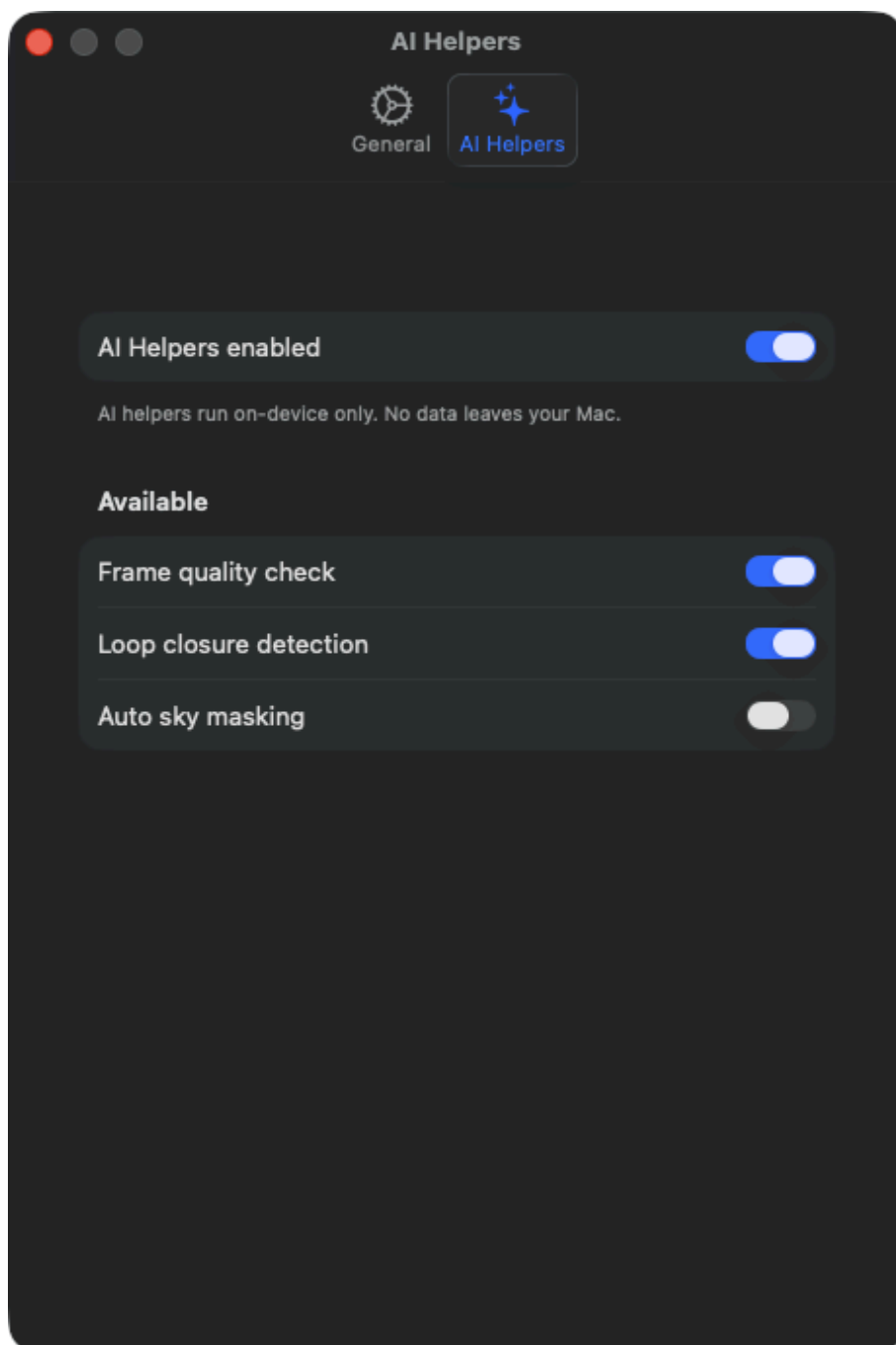


Figura 16: Configurações → aba AI Helpers com chave mestra e sub-toggles

S11 AI Helpers enabled (Master) **ONDE**

Settings → AI Helpers → primeira seção → toggle „AI Helpers enabled“. Bound: . Default: `true` .

 **TÉCNICO**

Chave mestra sobre todas as features de AI Helpers no pipeline. Quando desligada, o pipeline de import e SfM pula completamente todos os estágios de pré-processamento baseados em ML — sem chamada a Apple Vision, sem carregamento de modelo CoreML, sem acordar a NPU. Quando ligada, os sub-toggles individuais (S12–S13) são consultados. O valor é lembrado entre reinicializações. Afeta os seguintes estágios: (a) pre-check de qualidade de frame antes do SfM (S12), (b) detecção de loop closure (S13). Importante: quando desligada, os dois sub-toggles ficam desabilitados e visualmente acinzentados. O aviso no rodapé enfatiza que todos os AI Helpers rodam estritamente on-device — sem upload de imagem, sem processamento na nuvem. A garantia de privacidade vem do uso exclusivo do framework Apple Vision (local na Neural Engine) e de modelos CoreML que vêm diretamente no bundle do app.

 **EM PALAVRAS SIMPLES**

A chave principal para todas as funções que usam internamente IA/machine learning. O padrão é „ligado“, porque os auxiliares poupam muito tempo sem que suas imagens saiam do Mac. Se você quiser eles totalmente desligados (p. ex. para poupar energia ou porque seu Mac não tem NPU), desligue aqui — aí as duas sub-opções abaixo ficam automaticamente cinzas e não fazem nada.

S12 Frame quality check ONDE

Settings → AI Helpers → seção Available → toggle „Frame quality check“. Bound: . Default: `true` .

 TÉCNICO

Ativa o screener de qualidade de frame (Fase 3.11), que analisa cada frame importado antes da chamada de SfM. Passos do pipeline por frame: (a) filtro de variância laplaciana do Apple Vision (detecção de borrão — limiar ~150), (b) verificação de super/sub-exposição baseada em histograma (limiar: >5% pixels em 0 ou 255), (c) detecção de frame em branco (desvio padrão < 5 em todos os pixels). Frames que passam em todos os três checks seguem diretamente. Frames que falham em pelo menos um check disparam um diálogo modal de confirmação que lista cada frame problemático com thumbnail e justificativa e pergunta se ele deve ser removido. Importante: sem exclusão automática — o diálogo é sempre obrigatório, o usuário tem a última palavra. Performance: ~50 ms por frame em M3 Ultra, roda em paralelo. Quando desligado, todos os frames vão direto para o SfM sem checagem. Com a chave mestra (S11) desabilitada, este toggle fica acinzentado e sem efeito. Status entregue conforme memória: SHIPPED 2026-05-23.

 EM PALAVRAS SIMPLES

Antes do treinamento de fato, o app olha cada foto: está tremida? totalmente escura ou clara? em branco? Se sim, ele pergunta se você quer descartar a imagem — ele nunca remove nada automaticamente. Isso poupa muitas horas depois, porque uma única imagem totalmente tremida pode às vezes estragar todo o treinamento. O padrão é „ligado“, porque o custo é quase zero e o benefício é grande.

S13 Loop closure detection **ONDE**

Settings → AI Helpers → seção Available → toggle „Loop closure detection“. Bound: Default: `true`.

 **TÉCNICO**

Ativa a detecção de loop closure baseada em Apple Vision Feature Print. Para cada frame importado, é calculado um vetor de features de ~768 dimensões, que representa um embedding neural do conteúdo da imagem. Em seguida, todos os feature prints são comparados par a par via cosine similarity. Pares com similaridade > 0.85 e distância no índice de frames > 50 (ou seja, frames não adjacentes) são identificados como „candidatos a loop closure“ e gravados num arquivo JSONL sidecar na pasta do projeto. Apenas informativo — a sequência de imagens importada não é modificada. Sentido: dá ao solver de SfM (especialmente COLMAP) uma dica de que esses frames formam cluster no espaço 3D. Para SfM nativo, a informação sidecar é atualmente só documentativa; o COLMAP usa as dicas internamente via custom matches file (integração manual possível, não conectada automaticamente). Performance: ~200 ms por frame em M3 Ultra, roda em paralelo. Quando desligado, não há feature prints gerados. Com a chave mestra (S11) desabilitada, acinzentado.

 **EM PALAVRAS SIMPLES**

Quando você anda em volta de um objeto fotografando e termina perto do ponto de partida, ajuda muito o computador saber disso. Esta opção detecta automaticamente quais das suas fotos foram tiradas „quase do mesmo local“ e escreve isso num pequeno arquivo auxiliar. Ferramentas SfM (sobretudo COLMAP) podem usar essa informação para entregar uma reconstrução 3D mais limpa. O padrão é „ligado“ porque roda sem sua intervenção e não altera nada nas suas imagens.

Configurações Espelhadas do Inspetor

As demais entradas de configurações (S17–S33) da tabela do inventário são espelhamentos do Expert Inspector e estão documentadas no Capítulo 2 (Controles do Inspetor I12–I29). Elas não aparecem fisicamente na janela de Configurações, mas foram listadas no inventário apenas porque rodam via propriedades `TrainingConfig` que são persistidas via e portanto têm formalmente caráter de Configurações. Para explicações de conteúdo, veja lá.

Quando o quê?

Configuração	Escopo	Persistência
S1 Default Mode	Global no app	Reinício do app
S2 Language	Global no app	Reinício do app
S3 Viewport Background	Global no app (padrão) + Runtime	Reinício do app
S4 Auto-Rotate After Training	Global no app	Reinício do app
S5 Live Preview Interval	Padrão para novos treinamentos	Reinício do app
S6 Throttle Delay	Padrão para novos treinamentos	Reinício do app
S11 AI Helpers Master	Global no app	Reinício do app
S12 Frame quality check	Global no app	Reinício do app
S13 Loop closure detection	Global no app	Reinício do app

Global no app = atua em todos os projetos. Padrão para novos treinamentos: atua só no próximo treinamento criado; sessões em execução permanecem inalteradas. Treinamento atual = atua imediatamente na configuração de treinamento em execução, mas não persiste sem reimport explícito.

CAPÍTULO

Capítulo 4 — Janelas auxiliares

Além da janela principal (viewport 3D mais Inspetor), o RadianceKit gerencia sete outras janelas, todas abertas pelo menu Help. A lista de cima para baixo: User Guide (⌘?), Keyboard Shortcuts (⌘/), Open Training Logs... (não abre janela do app, mas o Finder; por isso não tratado aqui), Manage Storage..., Pareto Dashboard... (⇧⌘D), Holdout Analysis... (⇧⌘H), BayesOpt Console... (⇧⌘B). Três delas — Dashboard, Holdout, BayesOpt — são ferramentas de análise autônomas. Cada uma tem seu próprio stack de view model, lê ou grava arquivos JSON no disco, e cada uma tem um argumento de CLI para apontar a janela já no início do app para um arquivo específico (`--dashboard-dir` , `--holdout-file` , `--bayesopt-autorun`).

As quatro janelas simples (User Guide, Keyboard Shortcuts, Manage Storage, mais os submenus Open Training Logs / Open Exports Folder) ganham uma entrada curta por controle. As três janelas de análise estão documentadas com mais detalhe — cada uma com uma introdução que explica o que você vê, quando deve abrir e como interpretar.

No fim do capítulo há uma seção de cross-reference para o Inspetor da janela principal: o que você consegue ler no live loss chart e no indicador de Gaussian Count durante um treinamento em andamento.

User Guide (W1–W4)

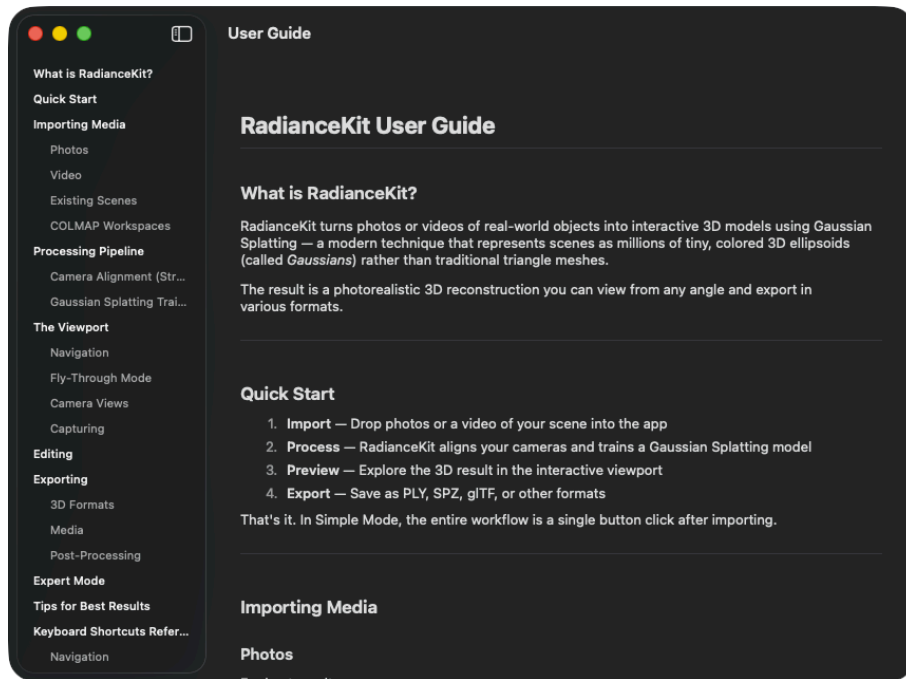


Figura 17: Janela do User Guide com sidebar à esquerda e conteúdo Markdown renderizado à direita

O que é: uma janela de ajuda embutida que renderiza o `guide_<idioma>.md` distribuído com o app. O idioma vem das Configurações (aba General → Language) ou, se estiver „System“, das preferências de idioma do macOS. Layout clássico: sidebar de títulos à esquerda, texto corrido à direita.

QUANDO ABRIR Quando você precisa de uma lembrança rápida sobre um ponto isolado — como substituto de palavra-chave. A referência completa é este manual; a janela embutida é o equivalente a um `--help` na linha de comando. É atualizada a cada release, mas mantida intencionalmente superficial.

W1 NavigationSplitView (sidebar + detalhe)



Help → User Guide (⌘?)..



Layout em duas colunas com sidebar estreita (mínimo 180 pt) para a árvore de conteúdo e área scrollável de detalhe para o markdown. A janela tem tamanho mínimo de 700×500 pt. Na primeira abertura, carrega o `guide_<lang>.md` do bundle (fallback `guide_en.md`), parseia em blocos (cabeçalhos H1–H4, parágrafos, listas, tabelas, separadores) e extrai separadamente a estrutura de headings para a sidebar. Formatação inline (bold, italic, code span) renderizada pelo engine markdown embutido. O idioma vem das configurações, com caso especial para Chinês (`zh-Hans`) e Português brasileiro (`pt-BR`) mantidos como locale tags completos, porque essas variantes diferem de zh ou pt respectivamente.

EM PALAVRAS SIMPLES

O texto de ajuda embutido, com a lista de tópicos à esquerda e o conteúdo à direita. O idioma se adapta automaticamente às preferências do sistema. Funciona offline mas é proposadamente apenas uma versão curta — a referência completa é este manual.

W2 List (sidebar de headings)



Coluna esquerda da janela User Guide..



Lista de todos os títulos H2 e H3 do documento. Entradas H2 sem recuo e em peso medium; H3 com 16 pt de recuo esquerdo e foreground reduzido. H4 ou mais profundos são ignorados para não confundir a sidebar. IDs de âncora geradas por slug do texto do heading (lowercase + espaços virando hifens + filtro a letras/números/hifens — mesmo algoritmo do GitHub para âncoras markdown, de modo que URLs externos podem cair no mesmo ponto). Lista usa estilo nativo do macOS.

EM PALAVRAS SIMPLES

A barra de navegação à esquerda. Toque numa entrada e você pula direto para a seção.

W3 Button (heading → salto para âncora)



Um botão por linha na sidebar..



Cada item da sidebar é um botão que define a âncora atual, mas tem aparência de item de lista. Uma variável observada dispara o scroll para a âncora correspondente com animação suave em 0,3 s. Após o salto, o valor da âncora é zerado, para que outro clique no mesmo dispare de novo (senão o observer não reagiria, pois o valor não mudaria).

EM PALAVRAS SIMPLES

O clique te leva ao trecho correspondente no texto à direita.

W4 ScrollView (conteúdo de detalhe)



Coluna direita..



Área scrollável com pilha vertical e render lazy, porque guias maiores podem ter mais de 200 blocos markdown — uma versão não-lazy instanciará tudo de uma vez. Cada bloco tem ID própria, seja a âncora do heading (para H1–H3 saltáveis) ou um placeholder de índice. Largura máxima 720 pt, padding 32 horizontal / 24 vertical, para linhas longas manterem layout legível. Tabelas são desenhadas célula a célula com stacks horizontais e divisores; código inline pelo engine markdown embutido. Blocos reais de código são tratados como parágrafo no momento — limitação conhecida da janela de ajuda.

EM PALAVRAS SIMPLES

O texto de ajuda em si. Scrollável, largura boa para leitura, tipografia clara.

Keyboard Shortcuts (W5–W6)

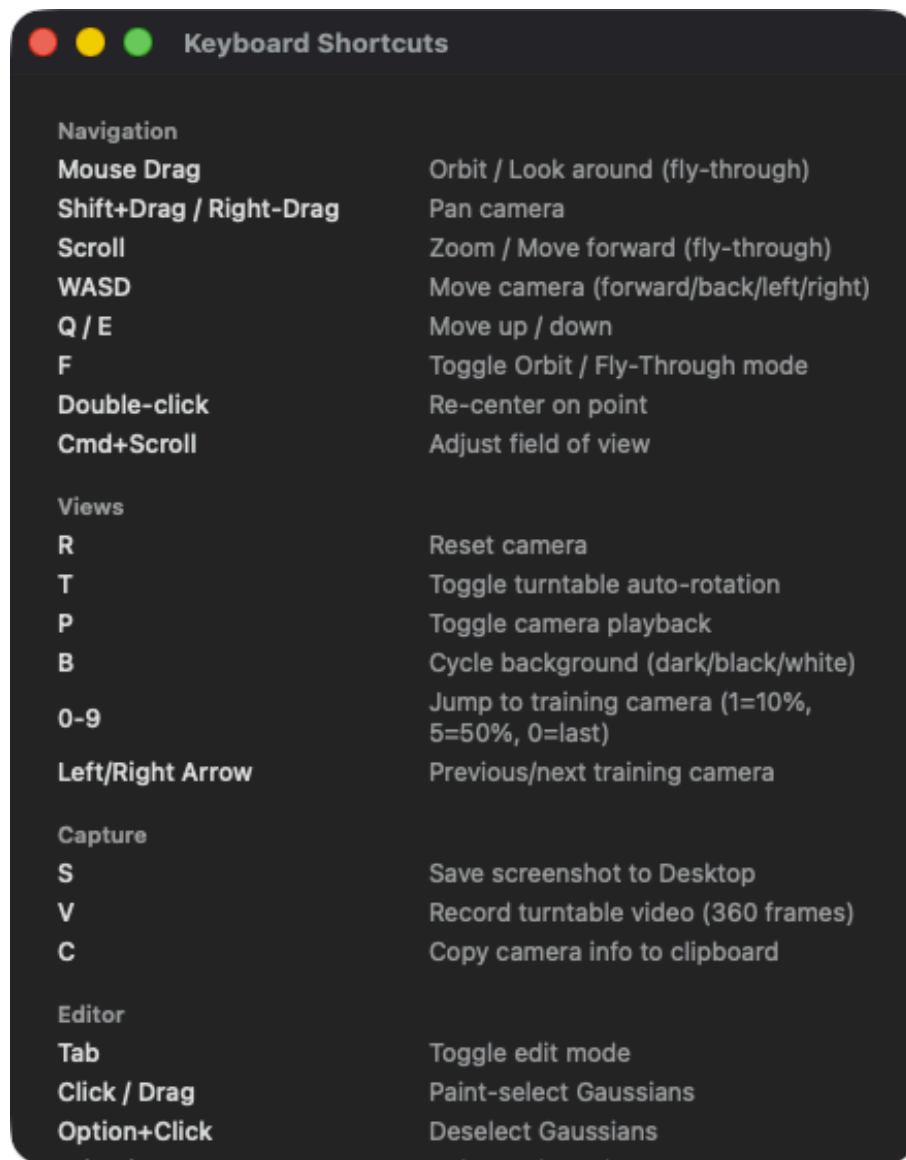


Figura 18: Janela Keyboard Shortcuts — cinco grupos Navigation/Views/Capture/Editor/Training com coluna de hotkey à esquerda e descrição à direita

O QUE APARECE NA IMAGEM Lista estática de referência em cinco seções. **Navigation:** arrastar mouse (orbit/fly), Shift+drag/right-drag (pan), scroll (zoom), WASD (movimentação fly-through), Q/E (up/down), F (toggle orbit/fly), duplo clique (recentralizar), Cmd+scroll (ajustar FoV). **Views:** R (reset câmera), T (auto rotação), P (camera playback), B (ciclar background), 0–9 (saltar para training cam 1=10%/5=50%/0=último), seta esquerda/direita (cam prev/next). **Capture:** S (screenshot para Desktop), V (vídeo turntable), C (copiar info de câmera). **Editor:** Tab (modo edit), clique/arrastar (paint select), Option+clique (deselect), X / Delete (remove seleção), Cmd-Z (desfazer última remoção), [/] (reduzir/aumentar pincel), Esc (limpar seleção). **Training:** start, pause/resume, cancel, continue +5K/+10K/+20K via atalhos de menu em M9–M14.

O que é: um overview estático simples dos atalhos — Navigation, Views, Capture, Editor, Training. Conteúdo hardcoded, sem markdown loading.

QUANDO ABRIR Quando você procura o caminho mais rápido para algo no viewport. Movimento WASD, R para resetar câmera, B para ciclar background — tudo está aqui.

W5 ScrollView (área de conteúdo)

 ONDE

Help → Keyboard Shortcuts (⌘/)..

 TÉCNICO

Área scrollável simples com lista vertical dentro. Padding 20 ao redor, sem árvore de navegação na sidebar (a lista é curta o suficiente). Conteúdo agrupado em cinco seções (Navigation, Views, Capture, Editor, Training). Por atalho, uma linha com texto traduzível nas duas colunas. Coluna esquerda (código de tecla) fixa em 180 pt para alinhar verticalmente. Sem interação além de scroll — clicar numa linha não dispara nada; os atalhos são modificadores de teclado reais no menu e no viewport.

 EM PALAVRAS SIMPLES

Tabela de todos os atalhos. Cheat sheet estático para consulta rápida.

W6 VStack (seções de atalhos)

 ONDE

Dentro do ScrollView..

 TÉCNICO

Seções empilhadas alinhadas à esquerda com 16 pt de espaçamento. Em cada uma das cinco, heading + sequência de linhas. Headings usam estilo subheadline secundário — deliberadamente sem formato título, porque as seções não precisam ser navegáveis. Conteúdo proposital plano (sem disclosure, busca, filtro), para o componente rodar inalterado em toda versão de macOS e o arquivo continuar legível.

 EM PALAVRAS SIMPLES

O agrupamento de teclas por função (navegação, views, editor e por aí vai).

Manage Storage (W7–W12)

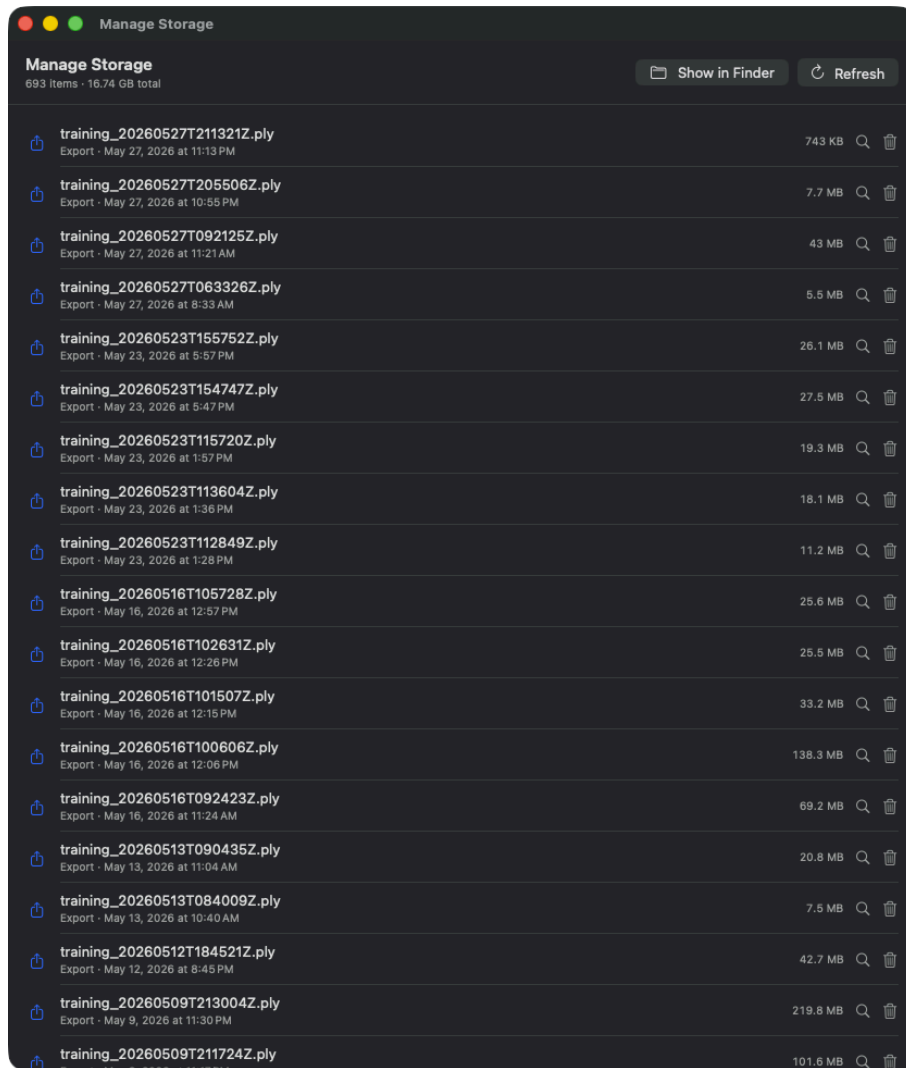


Figura 19: Janela Manage Storage — cabeçalho mostra „693 itens · 16.74 GB total“, tabela com exports PLY ordenada por data, cada um com pill de formato + nome + tamanho + data

O QUE APARECE NA IMAGEM Visão em tabela de todos os arquivos gerenciados pelo RadianceKit. Cabeçalho conta 693 itens, 16.74 GB. Toolbar acima: „Show in Finder“ + „Refresh“. Cada linha: ícone PLY, nome do arquivo (p. ex. training_20260527T211321Z.ply), data de export, tamanho (variando de 7 KB a 218 MB), ícone de lupa (Reveal) e ícone de lixeira (Move to Trash). Arquivos ordenados por data, mais novos em cima. Nesta captura demo, exports PLY dominam por causa do uso intensivo de `--benchmark`.

O que é: um overview de uso de disco para tudo que o RadianceKit grava em ~/Documents/RadianceKit/ — logs, exports, scenes, capture bundles (do companion iOS), imports (cópias staging das imagens de entrada). Por item um tamanho em bytes e dois botões: „Show in Finder“ e „Move to Trash“. NÃO é limpeza automática — o app não apaga nada por conta; você decide por item.

QUANDO ABRIR Quando o disco enche. Os logs são os maiores acumuladores (um JSONL por tentativa de treinamento, mais o `_qualityMetrics.json`); exports também (PLY 100 % cru, um por export). Também útil após um crash, quando o diretório de sta-

ging de imports ainda tem cópias antigas das imagens de entrada (veja „Disk-pressure incident“ em `dev_v549f-needle-reduction.md`).

W7 Botão „Show in Finder“

 ONDE

Cabeçalho acima à direita no Storage Browser..

 TÉCNICO

Abre todo o diretório do RadianceKit (`~/Documents/RadianceKit/`) no Finder, para você ver a estrutura de pastas e mexer com o próprio Finder. A ação abre uma nova janela do Finder e não muda para o container sandbox do app — `~/Documents/RadianceKit/` é a domain Documents acessível regularmente a apps, não um caminho de container sandboxed.

 EM PALAVRAS SIMPLES

Abre o diretório no Finder para você mexer nos arquivos por conta própria.

W8 Botão „Refresh“

 ONDE

Cabeçalho, ao lado do botão Finder..

 TÉCNICO

Dispara um scan em background numa task assíncrona iniciada pelo usuário, para que escanear árvores grandes não bloqueie a UI. O walk percorre cada subpasta conhecida (Logs, Exports, Scenes, Captures, Imports) e cria uma entrada de Storage por filho direto. Por entrada o tamanho recursivo é apurado — preferencialmente o uso real em disco (incluindo compartilhamento de hardlinks APFS) com fallback no tamanho lógico.

 EM PALAVRAS SIMPLES

Relê a lista, caso você tenha apagado ou adicionado algo no Finder no meio.

W9 List (entradas de Storage)

ONDE

Conteúdo principal abaixo do cabeçalho..



TÉCNICO

Lista com layout por linha: ícone SF Symbol específico da categoria (documento para Logs, seta de upload para Exports, cubo para Scenes, bandeja para Imports), nome + subtítulo (rótulo de tipo + data de modificação formatada), contador de bytes à direita (alinhado à direita, monospaced), botão Reveal (lupa), botão Trash (lixeira). Ordenação: primária por tipo (Scenes primeiro, depois Exports, Logs, Captures, Imports, Other), secundária por data de modificação descendente (mais nova em cima). Se o scan ainda roda, aparece um indicador „Scanning...“. Se nada foi encontrado, um empty state com ícone de bandeja.

EM PALAVRAS SIMPLES

Lista de todos os seus dados do RadianceKit, ordenada por tipo e atualidade. Por entrada você vê tamanho e pode apagar direto.

W10 Botão de linha „Reveal in Finder“

ONDE

Por linha, ícone de lupa à direita..



TÉCNICO

Abre o Finder e seleciona o item específico (arquivo ou pasta). Diferença para W7: W7 abre o diretório raiz; W10 marca exatamente essa entrada. Fluxo prático: identifique uma entrada grande, clique na lupa e copie para um volume externo, por exemplo.

EM PALAVRAS SIMPLES

Pula no Finder direto para esse item, para você achar rápido.

W11 Botão de linha „Move to Trash“

ONDE

Por linha, ícone de lixeira à direita da lupa..



TÉCNICO

Dispara o diálogo de confirmação (W12). Só após confirmar roda a operação padrão do macOS „mover para lixeira“ (reversível, sem apagar de fato). Após sucesso, a entrada some da lista e o contador de bytes total é atualizado. Em erros aparece um diálogo modal.

EM PALAVRAS SIMPLES

Move a entrada para a lixeira. O diálogo pergunta antes.

W12 ConfirmationDialog (confirmação de remoção)**ONDE**

Disparado pelo W11, exibido como sheet do macOS..

TÉCNICO

Diálogo de confirmação padrão com título dinâmico „Delete <name>?“ e mensagem explicitando que a entrada vai para a lixeira e pode ser restaurada de lá (até esvaziar). Dois botões: „Move to Trash“ como destrutivo (vermelho) e „Cancel“ com binding automático em Esc. Diálogo não modal no sentido de só bloquear essa janela, não todo o app — padrão macOS para remoções reversíveis.

EM PALAVRAS SIMPLES

Pergunta de segurança antes de apagar. „Move to Trash“ é reversível — enquanto a lixeira não for esvaziada.

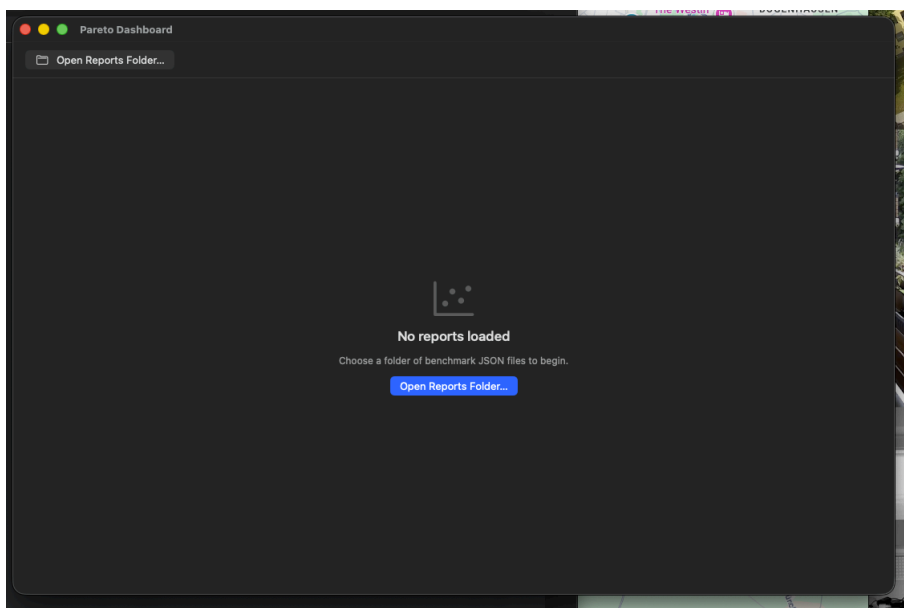
Pareto Dashboard (W13–W22)

Figura 20: Pareto Dashboard — estado vazio antes do import de reports

Estado vazio (na primeira abertura) — empty state com call to action „Open Reports Folder...“. Os pontos aparecem assim que training reports são carregados; ver próxima captura.

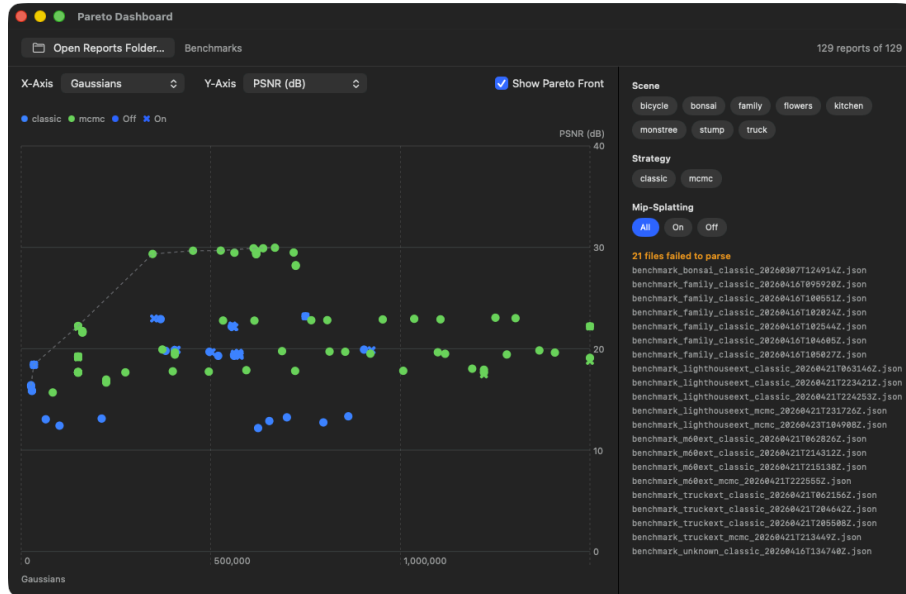


Figura 21: Pareto Dashboard com 129 reports de benchmark carregados — Gaussians vs PSNR com frente de Pareto, filtros Scene/Strategy/Mip

O QUE APARECE NA IMAGEM A toolbar mostra „129 reports of 129” (todos parseados com sucesso — 21 outros não puderam ser parseados por formato antigo; veja lista de hint à direita). Eixos: seletor X em `Gaussians`, Y em `PSNR (dB)`. Scatter plot: pontos verdes = estratégia Classic, azuis = MCMC. A linha tracejada da frente de Pareto segue os melhores valores de PSNR e platô em $\text{PSNR} \approx 30$ dB a partir de cerca de 500K Gaussians. Filtros à direita: 7 cenas (bicycle, bonsai, family, flowers, kitchen, stump, truck), 2 estratégias (classic, mcmc), 3 opções de Mip Splatting (All, On, Off). Nesta demo, todos os filtros abertos — daí o cluster denso.

O que é: uma ferramenta de comparação multi-run. Você treinou várias cenas, ou a mesma cena com várias predefinições, e cada corrida produz (quando você usa `--benchmark` ou a função de benchmark) um arquivo JSON de report com, entre outros, PSNR final, SSIM, LPIPS, Gaussian count e wallclock. O dashboard lê uma pasta inteira de reports e plota em scatter 2D com eixos selecionáveis. Além disso, a frente de Pareto (conjunto dos não dominados) aparece como linha tracejada.

QUANDO ABRIR Depois de pelo menos três ou quatro reports de treinamento. Com menos pontos, a linha de frente não é significativa. Caso típico: você tentou reconstruir uma cena externa rodando sequencialmente P3 Balanced (Classic), P4 Quality (Classic), P7 MCMC Quality e P9 Outdoor (tuned) — e agora quer saber qual config entrega o melhor PSNR por segundo de treinamento ou qual precisa de menos Gaussians para um dado PSNR.

COMO INTERPRETAR Os dois eixos são livres (X: `psnr`, `ssim`, `lpips`, ...; Y igual). A lógica `ParetoFront2D.indices` sabe por métrica se „menor é melhor” (p. ex. LPIPS, loss, time) ou „maior é melhor” (PSNR, SSIM) — a linha vai de baixo esquerda para cima direita, ou de cima esquerda para baixo direita, sempre na melhor combinação. Um ponto é Pareto-ótimo quando NENHUM outro é pelo menos tão bom em AMBAS dimensões (ninguém o domina). Os Pareto-ótimos ficam na linha; outros ficam à direita/acima dela.

Pontos NA linha são candidatos a „melhor preset“; pontos LONGE da linha são tempo de treinamento desperdiçado.

CHIPS DE FILTRO Você pode restringir a uma cena (p. ex. só Outdoor), a uma estratégia (Classic ou MCMC), ou Mip Splatting on/off (relevante pós Fase Q1.5, onde Mip ficou como flag opt-in avançado).

EXEMPLO DE FLUXO Você tem três reports da cena „truck“ em ~/Documents/RadianceKit/Reports/: Run A (P4 Quality, 40K iter, 524K Gs, 105 s, PSNR 23.4), Run B (P7 MCMC, 200K iter, 150K Gs, 693 s, PSNR 24.6), Run C (P9 Outdoor, 100K iter, 1.25M Gs, 312 s, PSNR 25.8). Eixo X em trainingTime, Y em PSNR. Run B fica acima à direita, Run C ainda mais acima à direita, Run A abaixo à esquerda. A frente de Pareto liga A e C — ambos não dominados. Run B fica „perdido“ (C melhor em tempo E PSNR). Conclusão: para „truck“, o default MCMC não vale; ou rápido+ok (A) ou longo+muito bom (C). Salve a config de C como predefinição própria (Inspector → I1 Save Preset).

Próxima ação: salve a melhor config como predefinição. Concretamente: olhe os pontos Pareto (hover mostra PSNR/SSIM/LPIPS/Gs/Time num tooltip), decida qual trade-off time-vs-quality combina, abra o report (nome contém timestamp), copie sua Training-Config num novo run ou salve como predefinição após a próxima sessão.

W13 Botão „Open Reports Folder..“



Toolbar em cima à esquerda..



Abre um seletor de pasta com instrução „Select a folder containing benchmark .json reports“. Após confirmar, uma task em background parseia sequencialmente todos os `.json`. Reports com falha (JSON quebrado, schema errado) são coletados e listados embaixo na sidebar como „N file failed to parse“ — sem crash. Se um segundo clique acontece durante um load anterior, o task anterior é cancelado, evitando dois resultados gravando no state.

Também via CLI: `--dashboard-dir /caminho/para/reports` carrega a pasta logo no início.

EM PALAVRAS SIMPLES

Escolhe a pasta com seus reports de benchmark. Caminho padrão é ~/Documents/RadianceKit/Reports/. Depois carrega todos os JSONs.

W14 Picker „X-Axis“

Acima do chart, à esquerda..



Picker de menu com todas as métricas disponíveis (PSNR, SSIM, LPIPS, contagem de Gaussians, tempo de treinamento etc.). Padrão é Gaussian count. Ao trocar, o ponto em hover é resetado, porque uma posição realçada não faz sentido em novo sistema de eixos. Limitado à largura do conteúdo, para não esticar por toda a largura.

EM PALAVRAS SIMPLES

Qual métrica fica no eixo horizontal. Normalmente „Training Time“ ou „Gaussian count“, porque esses são os „custos“ que você quer comparar.

W15 Picker „Y-Axis“

Acima do chart, ao lado do X-Axis..



Idêntico ao W14, mas o padrão é PSNR. A escolha é gravada independente, ou seja, o usuário pode pegar combinações sem sentido (X=PSNR, Y=PSNR — tudo na diagonal). Combinações assim não são impedidas; é decisão proposital, porque comparar „SSIM vs PSNR“ pode ser interessante para ver consistência entre métricas.

EM PALAVRAS SIMPLES

O que fica no eixo vertical. Normalmente „PSNR“ ou „SSIM“ como medida de qualidade.

W16 Toggle „Show Pareto Front“

À direita dos pickers de eixo..



Toggle padrão macOS. Quando ativo, além dos pontos é desenhada uma linha com a frente de Pareto 2D computada. Estilo: tracejado (4-4), cinza semi-transparente, 1.5 pt. Cálculo no main thread — para a quantidade típica de reports (≤ 50) é rápido. Quando desligado, a linha some.

EM PALAVRAS SIMPLES

Mostra a linha que liga os „melhores até agora“. Se atrapalhar (p. ex. só comparando trade-offs individuais), desligue.

W17 Chips de filtro „Scene“

ONDE

Sidebar à direita..



TÉCNICO

Chips de filtro para cada cena nos reports carregados. Layout flow próprio, que quebra chips em várias linhas quando a largura acaba. Chips ativos têm cor de destaque; inativos, material padrão. Multi-seleção via set semantic; sem nada selecionado, todas as cenas passam — ou seja, „seleção vazia = tudo“, não „seleção vazia = nada“.

EM PALAVRAS SIMPLES

Clicar num nome de cena filtra pontos só daquela cena. Multi-seleção possível. Vazio = todas.

W18 Chips de filtro „Strategy“

ONDE

Abaixo do filtro Scene..



TÉCNICO

Exatamente como W17, mas para estratégias — tipicamente os dois valores „classic“ e „mcmc“, do campo strategy dos reports. Útil quando você tem reports das duas estratégias misturados e quer ver só uma (p. ex. „só MCMC porque Classic já tirei“).

EM PALAVRAS SIMPLES

Filtro por Classic ou MCMC. Padrão é com ambos ativos.

W19 Chips de filtro „Mip Splatting“

ONDE

Abaixo do filtro Strategy..



TÉCNICO

Filtro de três valores (em vez de set como em W17/W18): „All“ / „On“ / „Off“. Contexto: Mip Splatting foi avaliado como melhoria multi-escala experimental na Fase Q1.5 e o veredito final foi „sem ganho consistente; mantido como flag opt-in“. Quando você compara Mip on/off, quer separar nítido. Daí o filtro ternário dedicado com „passar tudo“, „só Mip on“, „só Mip off“. Seção só aparece quando há pelo menos um report Mip E um não Mip (senão filtrar não faz sentido).

EM PALAVRAS SIMPLES

Se quiser comparar Mip Splatting on/off, aqui o filtro tripartido. Se não, ignore.

W20 ChipButton (toggle de filtro, all/on/off)

Componente helper usado em W17/W18/W19..



Wrapper minimalista. Conteúdo: rótulo em caption + padding 10 horizontal/5 vertical. Background condicional: ativo → cor de destaque com texto branco; inativo → material neutro com texto preto. Forma capsule (pílula). Plain button style para o material capsule não ser sobreposto por borda do sistema.

EM PALAVRAS SIMPLES

Os botões redondos dos filtros. Visual como uma tag iOS.

W21 Chart (scatter Pareto)

Área central do dashboard..



Gráfico Swift Charts com dois layers: 1. um ponto por report — posição pelos X/Y escolhidos, cor por estratégia, símbolo por status de Mip. Tamanho normal 80, highlighted 200 (se a ID bate com o report em hover). 2. uma linha para a frente de Pareto, só com toggle on.

Overlay do chart: um retângulo transparente registra movimento de mouse; por frame, o ponto euclidiano mais próximo no plot frame é determinado e o report em hover atualizado se a distância < 24 px (senão reset). Tooltip sem clicar — hover basta.

EM PALAVRAS SIMPLES

O scatter de verdade. Cada ponto é um run de treinamento. Hover dá tooltip de detalhes.

W22 Tooltip (detalhe em hover)

 ONDE

Abaixo do chart, exibido em hover..

 TÉCNICO

Stack horizontal: nome da cena (headline), tag de estratégia (caption), divisor, depois métricas PSNR/SSIM/ LPIPS/Gs/Time em grupos verticais (label + valor monospaced). Se Mip estava ativo, tag „Mip“ capsule em cor de destaque. Background semi-transparente com blur, cantos arredondados 8 pt. Aparece só se o mouse está sobre um ponto; some ao sair.

EM PALAVRAS SIMPLES

O card de detalhes embaixo quando você passa o mouse sobre um ponto. Mostra todas as métricas de qualidade e a config do run de uma vez só.

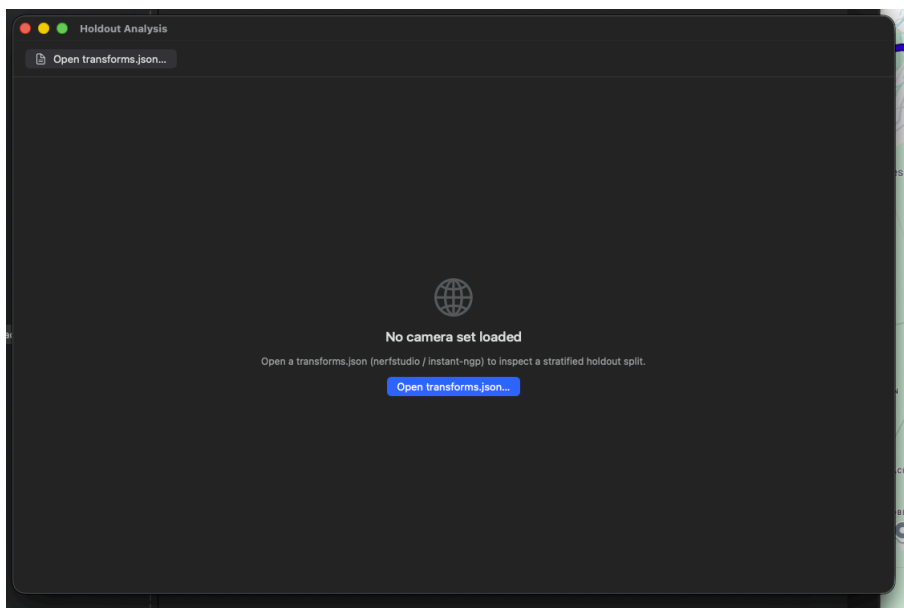
Holdout Analysis (W23–W29)

Figura 22: Holdout Analysis — estado vazio antes de carregar um transforms.json

Estado vazio com empty state e call to action „Open transforms.json...“. Aceita formatos NeRF Studio e Instant NGP.

Estado vazio (na primeira abertura) — os marcadores de câmera aparecem assim que uma `transforms.json` é carregada, veja próxima captura.

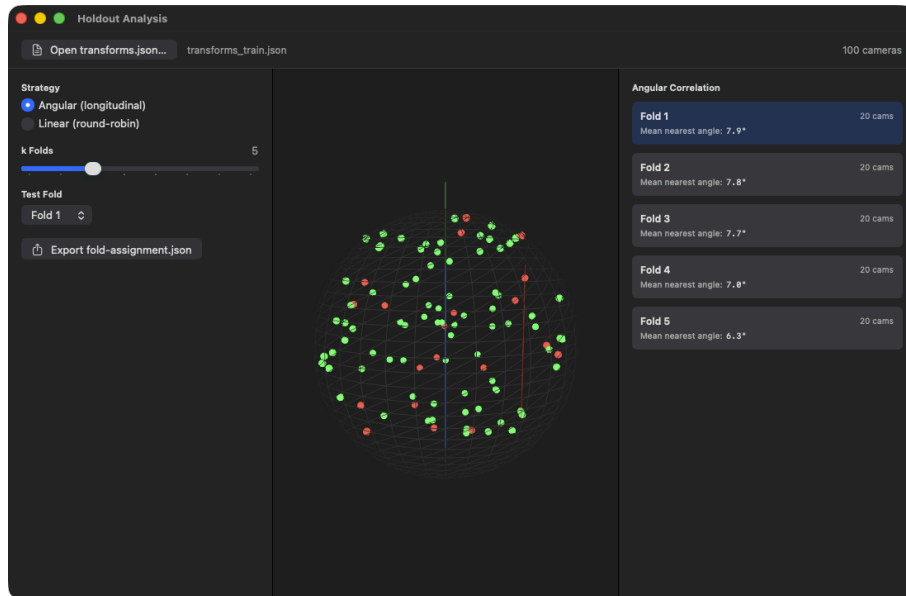


Figura 23: Globo Holdout com 100 câmeras NeRF Blender Mic, 5 folds de 20 câmeras, estratégia Angular ativa

O QUE APARECE NA IMAGEM Cabeçalho mostra arquivo carregado (`transforms_train.json`) e contagem de câmeras („100 cameras“). Sidebar à esquerda: picker Strategy com duas opções — Angular (longitudinal) ativo (alinha folds por setores long/lat na esfera, cada test fold fica geometricamente denso) vs Linear (round-robin) (baseado em ordem, todo k-ésimo frame como test set). Slider de k folds em 5, picker de test fold em fold 1. Botão Export gera um `fold-assignment.json` para Nerfstudio/Instant NGP. Pannel central: projeção em globo 3D das 100 câmeras — pontos verdes = train, vermelhos = test fold atual (fold 1 com 20 câmeras). Sidebar à direita (Angular Correlation): por fold 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°) — menor valor significa câmeras dentro do fold mais próximas entre si, ou seja, holdout split coerente espacialmente.

O que é: um visualizador 3D do arranjo de câmeras com lógica de cross validation. Você carrega um `transforms.json` (formato padrão Nerfstudio/Instant NGP para poses de câmera); o app lê todas as câmeras, projeta suas direções de vista numa esfera unitária e mostra como pequenos marcadores num globo virtual. Em seguida divide as câmeras em `k` folds (pela estratégia: angular ou linear), marca em verde o train e em vermelho o test (holdout), e calcula por fold um score de correlação angular dizendo quão longe o test está do train no espaço de ângulos.

QUANDO ABRIR Quando for fazer avaliação de holdout — ou seja: quão bem o modelo generaliza para ângulos não vistos? O padrão no treinamento é „toda 8ª view como holdout“ (convenção Mip-NeRF360), mas é uma divisão muito linear. Se suas imagens estão clusterizadas no tempo (primeiro um lado do objeto, depois o outro), „toda 8ª“ não é representativo — uma posição aleatória vai para test, mas todas as vizinhas estão em train, é fácil demais. Com „angular“ você estratifica pelo espaço de ângulos: cada fold contém câmeras de todas as regiões da órbita, e o test mede de fato lacunas de generalização.

COMO INTERPRETAR Angular vs Linear:

- Angular (padrão): divide as câmeras pelo longitude ϕ (coordenada em torno do eixo Y) em k setores iguais. Fold 0 são câmeras com $\phi \in [0^\circ, 360/k^\circ)$, Fold 1 as próximas, etc. Vantagem: cada fold cobre um sub-setor da órbita; o test fica espacialmente compacto mas amplo sobre o dataset mundial. Bom para capturas orbit clássicas.

- Linear (round robin): fold index = (image_index modulo k). É

o simples „every-k-th“. Funciona se a ordem das imagens NÃO tem viés espacial (p. ex. drone aleatório). Funciona mal se imagens clusterizam no tempo.

No globo 3D você vê na hora: pontos verdes (train) e vermelhos (test). Se os vermelhos estão todos num canto, o holdout está ruim (sem bom teste de generalização). Se ficam distribuídos entre os verdes, está bom. O score de correlação angular por fold (sidebar à direita, em graus) também diz: menor valor = test próximo do train (cada cam de test tem cam de train próxima, teste fácil); maior valor = test distante do train (generalização mais dura).

EXEMPLO DE FLUXO Você capturou sua cena Truck com 251 imagens, exporta via menu M33 (Export SfM transforms.json) um arquivo nerfstudio. Abra a janela Holdout (⇧⌘H), carregue o JSON via „Open transforms.json...“, veja o globo. $k=5$ (padrão) dá 5 folds. Clique em „Fold 3“ — veja se os vermelhos ficam mais ou menos uniformes. Se sim: „Export fold-assignment.json“, coloque na pasta Reports e no próximo treinamento com `--benchmark` (ou as configurações correspondentes do Inspetor), essa partição é usada como test holdout — em vez do padrão „toda 8ª“.

W23 Botão „Open transforms.json...“

ONDE

Toolbar acima à esquerda..

TÉCNICO

Abre um seletor de arquivo restrito a JSON. Após confirmar, o módulo holdout carrega. O loader parseia tanto o formato nerfstudio (intrínsecas + lista de frames com caminho e matriz transform) quanto instant ngp (mesma estrutura). Por frame extrai a direção de vista da matriz (eixo Z da base local da câmera) e salva. Se o parse falha, aparece uma mensagem de erro no status.

Também via CLI: `--holdout-file /caminho/para/transforms.json` inicia a janela já com arquivo carregado.

EM PALAVRAS SIMPLES

Carrega seu JSON de poses de câmera. Padrão são exports Nerfstudio e Instant NGP. O próprio RadianceKit exporta transforms.json via menu → Export → SfM.

W24 Picker „Strategy“ (angular/linear)

ONDE

Sidebar à esquerda, em cima..



TÉCNICO

Picker radio com duas opções: Angular e Linear. A troca dispara recálculo automático dos folds. As direções de vista são uma lista de vetores unitários 3D na esfera; a estratégia Angular projeta no longitude ϕ e ordena, a Linear faz divisão modulo pelo índice de frame.

EM PALAVRAS SIMPLES

Angular para capturas orbit uniformes (padrão, seguro); Linear só se suas imagens não clusterizam espacialmente.

W25 Slider „k Folds“

ONDE

Sidebar à esquerda, no meio..



TÉCNICO

Slider de 3 a 10, passo 1. Mudança dispara recálculo automático dos folds; a lista de folds, índices de train/test e o score por fold são recalculados em seguida. O valor é exibido como texto monospaced à direita do label.

Regra prática: k=5 padrão (20 % por fold, usual em cross validation). k=10 com muitos dados e mais folds para força estatística. k=3 com poucos dados.

EM PALAVRAS SIMPLES

Quantos folds na divisão. 5 é padrão e quase sempre cabe.

W26 Picker „Test Fold“

ONDE

Sidebar à esquerda, abaixo do slider k..



TÉCNICO

Picker de menu. Opções dinâmicas 0..<k, rótulo „Fold 1“ a „Fold N“ (1-indexed na UI, 0-indexed internamente). Se o índice escolhido antes $\geq k$ (p. ex. você reduziu k de 10 para 5), é resetado para 0. O test fold escolhido aparece em vermelho no globo; os outros em verde.

EM PALAVRAS SIMPLES

Qual fold é o test no momento. Você pode clicar por todos e ver cada um no globo.

W27 Botão „Export fold-assignment.json“

ONDE

Sidebar à esquerda, embaixo..



TÉCNICO

Abre um diálogo de salvar com nome padrão `fold-assignment.json`. Após confirmar, o módulo holdout codifica a partição atual num schema JSON (atribuição por frame + bloco meta de estratégia). O arquivo pode ser passado no próximo treinamento com `--benchmark`, usando o mesmo holdout para a avaliação de métricas final. Erros de gravação aparecem como texto de erro; sucesso em texto verde como „Saved to (filename)“.

EM PALAVRAS SIMPLES

Salva a divisão atual de train/test como JSON. Esse arquivo você pode passar direto no treinamento, e o mesmo test set é reutilizado.

W28 SCNView (globo de câmeras 3D)

ONDE

Painel central da janela Holdout..



TÉCNICO

Globo SceneKit. A cena consiste em: uma esfera wireframe (raio 1.0, 36 segmentos, cinza escuro), três pequenos cotos de eixo (vermelho/verde/azul para X/Y/Z, 1.2 de comprimento) e, por câmera, uma esfera marcadora (raio 0.03) na posição da direção de vista na esfera unitária (ligeiramente fora para não desaparecer DENTRO do wireframe). Os marcadores NÃO são reconstruídos a cada mudança de fold — rebuild só quando a lista de frames muda (novo JSON carregado). Por update há atualização in-place das cores de material: vermelho para test, verde para train, cinza claro para nenhum. Assim, os ticks do slider ficam performáticos mesmo com $N > 1000$ câmeras.

O controle de câmera está ativo — você rotaciona, dá zoom e faz pan com o mouse. Iluminação evita marcadores chatos. Fundo cinza escuro.

EM PALAVRAS SIMPLES

O globo 3D com as posições de câmera. Verde = Train, Vermelho = Test, cinza claro = não atribuído (não acontece; todas as câmeras pertencem a algum lado). Com o mouse você gira e dá zoom.

W29 FoldCard (toque para selecionar fold)

ONDE

Sidebar à direita, seção „Angular Correlation“..



TÉCNICO

Por fold um card view — retângulo arredondado raio 6 pt, padding 10, layout vertical em duas linhas (em cima „Fold N“ + número de câmeras; embaixo „Mean nearest angle:“ + valor em graus). Background condicional: fold ativo = cor de destaque semitransparente; inativos = material neutro. Tocar seleciona o fold; o globo recolora ao vivo.

O „Mean nearest angle“ é o menor ângulo médio por câmera de test para a câmera de train mais próxima (cálculo interno em radianos, exibido em graus).

EM PALAVRAS SIMPLES

Por fold um cardzinho à direita com número de câmeras e a distância média até a próxima câmera de train. Clicar nele torna esse fold o test.

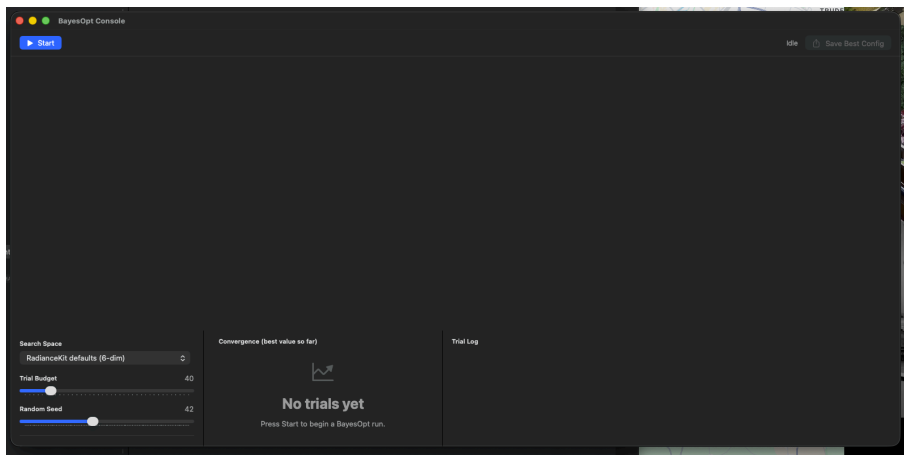
BayesOpt Console (W30–W39)

Figura 24: BayesOpt Console — estado vazio antes do início dos trials

Estado vazio com picker de search space (RadianceKit defaults (6-dim)), slider de trial budget (default 40), random seed (42) e três painéis vazios para convergence chart, trial log e lista de parâmetros do search space.

Estado vazio (na primeira abertura) — convergence chart e tabela de trials enchem assim que um run inicia; veja próxima captura.



Figura 25: BayesOpt Console após 40 trials — convergence chart sobre íngreme até trial 15, best value 0.9943, trial log com tags init/bo/restart

O QUE APARECE NA IMAGEM Status acima à direita „Finished — best 0.9943 after 40 trials“. Sidebar à esquerda: picker em RadianceKit defaults (6-dim), trial budget 40, random seed 42. Lista de parâmetros mostra os seis hiperparâmetros a ajustar e suas faixas: 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]. Centro: convergence chart (X = índice de trial 1–40, Y = objective value 0–1) — pontos cinzas = samples iniciais (LHS), azuis = acquisition BayesOpt, laranjas = trials restart (#22 e #31). Linha de best so far sobre íngreme até trial 7, depois melhoria marginal até trial 15; daí plateau em 0.99+. Sidebar à direita: trial log #1–#34 com score + tag (init/bo/restart). Save Best Config no topo direito grava `bayesopt-best.json`.

O que é: um console de otimização bayesiana para busca de hiperparâmetros. BayesOpt é um método automático que tenta, com o mínimo de experimentos possível, encontrar o ponto ótimo de uma função desconhecida — tipicamente: „qual combinação de mcmcMaxGaussians, capMultiplier, ssimWeight e gradThreshold dá o melhor PSNR para minha classe de cena?“ Em vez de um grid de $6^4 = 1296$ trials, BayesOpt tenta cerca de 40–100 trials informados e chega perto do ótimo.

Importante: a versão atual no app não executa a otimização contra runs reais de treinamento (levaria dias), mas contra uma demo objetiva sintética — uma paisagem multimodal com caráter de hill climbing e ruído leve. Foi proposital: a janela deve te mostrar o comportamento do otimizador (curva de convergência, samples, best so far) e te fazer entender as definições de search space. Para runs reais BayesOpt orientados por treinamento (como na Fase Q7 para os Scene Class presets), um workflow offline de CLI é usado separadamente; a janela é a variante UI live.

QUANDO ABRIR Três casos:

1. Você quer entender como BayesOpt trabalha — inicie um run demo e veja o convergence chart.
2. Você planeja uma nova classe de cena (p. ex. „aquários“ ou

„móveis antigos“) para a qual os 10 presets embutidos não encaixam. Defina mentalmente um search space, teste aqui com „Bowl demo“ ou o preset „Densify“, exporte a melhor config como JSON e use como ponto de partida para um run real.

3. Você quer inspecionar os search spaces default definidos no

package RKBayesOpt (Mip subset, RadianceKit defaults) — eles aparecem no painel de parâmetros da sidebar à esquerda.

COMO INTERPRETAR

- **Convergence chart** (coluna central): Y = melhor valor

objetivo até agora. X = índice de trial. No início sobe íngreme (BayesOpt sorteia samples iniciais; alguns dão sorte); depois fica mais plano, porque a vizinhança próxima do ótimo foi explorada. Se a linha fica plana por 20+ trials, pode parar — mais trials não trazem nada. Os pontos individuais são os valores de trial (não „best so far“), coloridos por fase: cinza = sample inicial, azul = acquisition, laranja = restart.

- **Tabela de trials** (coluna direita): #1, #2, #3, ...

com valor e tag de fase. O melhor trial tem estrela amarela. Da tabela você identifica o melhor e vê os valores de parâmetro no export.

- **Search space inspector** (sidebar à esquerda): mostra,

para o preset escolhido, todos os nomes de parâmetro e suas faixas `[lo, hi]`. Em „RadianceKit defaults (6-dim)“ você vê, p. ex., „densifyGradThreshold [5e-7, 5e-6]“ — log-uniform entre os dois.

EXEMPLO DE FLUXO Escolha o preset „RadianceKit defaults (6-dim)“, trial budget 40, seed 42. Clique „Start“. Observe: os primeiros 8 trials são cinzas (samples iniciais, LHS Latin Hypercube); os próximos azuis (acquisition). O convergence chart sobe íngreme até trial 15; depois achata. Em trial 30–40 o melhor estabiliza. Clique „Save Best Config“ — `bayesopt-best.json` é salvo com nome do preset, índice do trial, valor e parâmetros decodificados. Pode levar manualmente para sua definição de preset.

W30 Botão „Start“

Toolbar à esquerda, em idle/finished..



Reseta a lista de trials, vai para running, gera novo run ID (para detecção de stale em cliques múltiplos) e cria pause gate fresco. Em seguida inicia uma task em background que roda o otimizador como stream assíncrono. O tamanho de samples iniciais sai de $\min(8, \text{budget} / 4 + 1)$ — típico 8 LHS com $\text{budget} \geq 28$, menos com budget pequeno. Updates incrementais são anexados à lista. Stale run protection: se um segundo clique em Start define novo run ID no meio do caminho, updates do run anterior são descartados.

Primary action style para um botão proeminente.

 EM PALAVRAS SIMPLES

Inicia um run novo com o search space, budget e seed atuais.

W31 Botão „Pause“

Toolbar à esquerda, em running..



Ativa o pause gate e vai para paused. Efeito real: o runner espera num polling de 50 ms antes da próxima avaliação. O trial em curso termina (sintético, dura microssegundos), mas nenhum próximo é disparado. Em resume, continua de onde parou.

 EM PALAVRAS SIMPLES

Pausa o run. O cálculo atual termina, depois pausa.

W32 Botão „Stop“

Toolbar à esquerda, em running e paused..



Cancela a task do runner, zera a referência, solta o pause gate (se em paused) e vai para finished (se há trials) ou idle (se nenhum). Trials já calculados permanecem visíveis — Stop não apaga. Role de botão destrutivo (mostra em vermelho), porque cancela.

 EM PALAVRAS SIMPLES

Cancela o run em definitivo. Trials ficam visíveis; pode exportar a best config mesmo assim.

W33 Botão „Resume“ ONDE

Toolbar à esquerda, em paused..

 TÉCNICO

Solta o pause gate e volta para running. A task já está rodando (esperando no polling); assim que o loop percebe o release, segue e dispara o próximo trial.

 EM PALAVRAS SIMPLES

Continua um run pausado.

W34 Botão „Save Best Config“ ONDE

Toolbar à direita, sempre visível (mas desabilitado sem bestTrial)..

 TÉCNICO

Abre diálogo de salvar com nome padrão `bayesopt-best.json`, restrito a JSON. Após confirmar, monta um dicionário payload: nome do preset, índice do trial, valor (objective score), parâmetros (dict de nomes de parâmetro decodificados → valores). A decodificação projeta as coordenadas normalizadas em $[0,1]^d$ de volta na faixa original (com escalas log-uniform/linear/inteira). JSON pretty printed com chaves ordenadas. Em erro (na versão demo atual) ignora em silêncio — sem UI de erro, é um caminho demo.

O botão fica cinza enquanto nenhum trial rodou.

 EM PALAVRAS SIMPLES

Salva os parâmetros do melhor trial como JSON. Você pode levar esses valores manualmente para sua predefinição.

W35 Picker de preset „Search Space“

ONDE

Sidebar à esquerda, em cima..



TÉCNICO

Picker de menu com quatro opções:

- „RadianceKit defaults (6-dim)“ — search space completo

com hiperparâmetros Q7.

- „Mip subset (2-dim)“ — só `mipSmoothing3DScale` [0.05,

0.5] `log-uniform` e `mipFilter2DVariance` [0.1, 0.6] `linear`. Útil para tunar Mip Splatting numa classe de cena.

- „densify-until + ssim-weight + grad-thresh“ — três

parâmetros `densify-relevantes`

(`densifyGradThreshold` `log-uniform`, `ssimWeight` `linear`, `densifyUntilIter` inteiro).

- „Bowl demo (1-dim)“ — search space pedagógico de um

parâmetro para demos „assim funciona BayesOpt“.

Com um run ativo, o search space não pode ser trocado (confundiria o otimizador).

EM PALAVRAS SIMPLES

Qual hiperparâmetro-espaço BayesOpt investiga. Padrão é „RadianceKit defaults“. Para tuning focado em Mip, „Mip subset“. Para aprender como BayesOpt funciona, „Bowl demo“.

W36 Slider „Trial Budget“

ONDE

Sidebar à esquerda, abaixo do picker..



TÉCNICO

Slider de 10 a 200, passo 5. Padrão 40. Significa: BayesOpt pode fazer no máximo N trials. Destes, os primeiros são samples iniciais (Latin Hypercube); o resto são BayesOpt reais. Regras práticas: um search space com d dimensões precisa de cerca de $10d$ a $20d$ trials. Em 6-dim defaults, 60–120; em 2-dim Mip subset, 20–40; em 1-dim Bowl, 10–20.

Durante o run, slider desabilitado.

EM PALAVRAS SIMPLES

Quantos trials máximos. Mais = melhor solução, mas custa tempo. 40 é bom padrão para a demo.

W37 Slider „Random Seed“

ONDE

Sidebar à esquerda, abaixo do slider de budget..



TÉCNICO

Slider de 1 a 100, passo 1. Padrão 42. Seed é repassado tanto para os samples LHS iniciais quanto para o componente de ruído da demo objective. Reprodutibilidade: mesmo seed + mesmo search space + mesmo budget dá sequência de trials idêntica. Útil para „seus colegas pegam o mesmo run ao refazer a demo?“. Desabilitado durante o run.

EM PALAVRAS SIMPLES

Controla o gerador aleatório. Mesmo seed = mesmo run — para reproduzir.

W38 Chart (convergência)

ONDE

Coluna central da janela..



TÉCNICO

Gráfico Swift Charts com dois layers:

1. uma linha para „best-value-so-far“ por trial — curva

monotonicamente crescente ou constante na cor de destaque.

2. um ponto por trial com o valor objetivo individual, colorido por fase. Tamanho de símbolo 40. Três rótulos de fase: „init“ (cinza), „bo“ (azul), „restart“ (laranja).

Pequena legenda mostra as cores em cima à esquerda. Se a lista de trials está vazia (antes do primeiro start), aparece um empty state com ícone de chart e a dica „Press Start to begin a BayesOpt run.“.

EM PALAVRAS SIMPLES

O chart de evolução. A linha contínua é „melhor solução achada“; os pontos são os trials individuais. Se a linha fica plana por muito tempo, BayesOpt encontrou o ótimo.

W39 Table (trial log)

ONDE

Coluna direita da janela..



TÉCNICO

Área scrollável com linhas de trial empilhadas lazy. Por linha um stack horizontal: número do trial (3 dígitos monospaced, à esquerda), valor (monospaced, alinhado à direita, 70 pt), tag de fase (capsule, cor da fase em 25 % opacity); opcionalmente uma estrela amarela se for o melhor. Auto scroll automático: o painel salta para o fim sempre que um novo trial chega — assim você acompanha o fluxo ao vivo no rodapé sem rolar.

EM PALAVRAS SIMPLES

Tabela de todos os trials. Valor, fase, estrela para o melhor. Faz scroll automático com novos trials no fim.

Janela principal: curva de loss e Gaussian count (I39–I41, cross-reference)

Três indicadores do Inspetor na janela principal merecem uma explicação à parte, porque ficam visíveis durante o treinamento todo e existem regras práticas importantes sobre quando o curso está saudável. Os indicadores estão na seção „Loss Chart“ do Inspetor (Capítulo 2 — Inspetor) e complementam a Holdout Analysis da janela auxiliar acima.

Quando a curva de loss está saudável? Uma curva sadia mostra três fases: (1) **Warmup** — as primeiras 200–500 iters caem íngreme de alto (típico 0.15–0.25 para L1+SSIM combinados, conforme a cena) para mais ou menos metade. Se o loss NÃO cai nesta fase, em geral a entrada está errada (imagens quebradas, poses SfM ruins, número de Gaussians iniciais baixo demais). (2) **Densification** — entre 500 e densifyUnliteration (classic 15K, MCMC 20K ou 25K), o loss cai mais, com pequenos saltos para baixo conforme densify insere novos Gaussians que o otimizador usa. O Gaussian count sobe nesta fase. (3) **Refinement** — depois disso o loss corre num tail cada vez mais plano. Valores típicos: Tanks & Temples Truck com P4 Quality em L1 \approx 0.023, Horse com Full Classic V546 em L1 \approx 0.0230, Mip-NeRF360 externas piores (0.04–0.07).

O que significa um plateau? Um plateau (curva de loss horizontal por vários milhares de iters) tem duas leituras: (a) o modelo convergiu, treinar mais não traz nada — caso bom. (b) o modelo está stuck (mínimo local, gradiente ruim, cap no buffer) — caso ruim. Os dois parecem iguais no chart. Distinguir: olhe o Gaussian count. Se também está plano e perto do cap MCMC (p. ex. 150K de 150K em `.fullMCMC`), você está no limite — ou aumente o cap ou aceite o plateau. Se o count ainda cresce mas o loss não cai, é stuck.

Cancelar ou continuar? Regra: 10K iters sem melhora do Min loss → cancele, mais iters são desperdiçadas. Antes disso, via Cmd+T (Training → Continue Training → +5K

iters) você pode anexar mais, se houver melhora marginal. Atenção: em MCMC o plateau costuma ser real — o cap é o limite natural.

Plateau de Gaussian count NÃO é sinal de „pronto“. Significa só que MCMC bateu no cap ou Classic exauriu densification. A pergunta „pronto“ mesmo só a Holdout Analysis responde — PSNR/SSIM/LPIPS em test set independente, avaliado na janela Holdout (W23–W29) ou via `--benchmark`.

PSNR/Holdout é a verdade; loss é só proxy. O loss é métrica relativa: cai conforme o modelo se ajusta às training views. Loss baixo não significa modelo bom — se decorou as training images (overfitting), o loss é pequeno mas o PSNR em views não vistas (holdout) é ruim. Por isso: para avaliação final, sempre olhe métricas de holdout, não só end-loss.

Caixa de regras práticas

- User Guide e Keyboard Shortcuts são ajuda estática — rápido para palavra-chave; para profundidade, este manual.
- Manage Storage assim que o disco tiver menos de 10 % livre. Logs e staging de imports são os principais culpados.
- Pareto Dashboard só faz sentido depois de pelo menos três ou quatro reports. X = custos (time/Gs), Y = qualidade (PSNR/SSIM). Pareto front mostra as combinações eficientes.
- Holdout Analysis antes de publicar benchmarks PSNR — garante que seu test set é representativo.
- BayesOpt Console é sobretudo ferramenta de aprendizado e inspeção dos search spaces. Para tuning real de hiperparâmetros por treinamento, use o workflow CLI offline.
- Plateau de loss e plateau de Gaussian count são interpretados separadamente. Cap não é sinal de „pronto“. Qualidade real só holdout PSNR mede.
- 10K iters sem melhora de Min loss → parar.

CAPÍTULO

Capítulo 6 — Configuração de treinamento

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

Figura 26: Predefinição Preview exportada como JSON exibida no TextEdit — campos id/name/category/version/createdAt/description, trainingConfig com todos os parâmetros relevantes (maxIterations 5000, densifyUntilIteration 3500, ssimWeight 0.20, renderScale 0.50, strategy classic, cameraAlignment applePhotogrammetry, densifyGradThreshold 2.0e-06, opacityResetInterval 3000, minOpacity 0.005, seis toggles boolean)

O QUE APARECE NA IMAGEM Um export típico de predefinição em JSON. Campos no topo: id (UUID), name, (classic | mcmc | sceneClass | custom), (versão de schema), (timestamp), (texto livre). O objeto aninhado contém os parâmetros críticos para reprodutibilidade — no import, todo o bloco é deserializado para a struct `TrainingConfig`, e os defaults da versão do app preenchem os campos faltantes (p. ex. após update do app). Quem passa uma predefinição para outro Mac envia apenas esse JSON.

A struct `TrainingConfig` é o coração de todo treinamento no RadianceKit. Ela reúne cada parâmetro que afeta o treinamento — do número máximo de iterações pelas oito learning rates até os campos especiais de MCMC, Mip-Splatting, currículo e a lógica

scene-aware de cap. Você a edita na sidebar na seção Configuração de treinamento (Expert View), salva como predefinição ou repassa como export JSON para outro Mac. No treinamento, esse objeto é congelado e entregue ao backend de GPU.

Este capítulo é material de referência para usuários avançados e autores de script. Lista todos os 81 campos públicos, as 9 predefinições estáticas e o método público único. O arquivo-fonte é `TrainingConfig.swift` — em caso de dúvida, vale o doc comment ali e o default do initializer como source-of-truth.

AVISO · UI VS. PREDEFINIÇÃO/CLI

Apenas 12 dos 81 campos têm slider, toggle ou picker direto no Inspetor (build sandboxed para App Store): **T1, T2, T17, T20, T22, T38, T56–T58, T60, T61, T73**. Os outros 69 campos são configurados pela **predefinição** escolhida (Capítulo 7) e só podem ser sobrescritos diretamente por **flag de CLI** (ver Capítulo 5). Essa separação é proposital: defaults ficam estáveis e testados em produção, e usuários avançados têm uma escape hatch. Se um campo te interessa especialmente: confira primeiro no Capítulo 2 (Inspetor) e no Capítulo 5 (CLI) se você consegue alcançá-lo sem mexer no JSON.

Sumário:

1. Iteração (T1–T2)
2. Learning rates (T3–T10)
3. Densification — Classic (T11–T16)
4. Loss (T17–T20)
5. Progressão de SH degree (T21)
6. Performance (T22–T25)
7. Diagnóstico e preparação da nuvem de pontos (T26–T30)
8. Regularização (T31–T37)
9. Refinement (T38–T44)
10. Sky Dome (T45–T48)
11. Adam + LR Schedule (T49–T55)
12. Pós-processamento + Apple AI (T56–T60)
13. MCMC Densification (T61–T73)
14. Mip-Splatting (Q1.5) (T74–T76)
15. Adaptive Densification (Q5) (T77–T79)
16. Curriculum (Q6) (T80–T81)
17. Predefinições estáticas (TP1–TP9)
18. Método:
19. Qual campo para quê? (cheat sheet)
20. Campos perigosos

Iteração (T1–T2)

T1 maxIterations

DETALHES

Default: 30 000 (Initializer), 35 000 (`.full`), 200 000 (`.fullMCMC`) **Range:** 1 000 – 500 000 (slider UI), sem limite duro na lógica **Defined in:**

TÉCNICO

Total de iterações de treinamento que o backend percorre. Uma iteração denota um forward render de uma única câmera de treinamento, um backward pass por todos os componentes de loss (L1 + SSIM + regularizações opcionais + máscara de céu) e um passo do otimizador Adam. Esse número atua direto sobre outros schedules: LR de posição segue cosine annealing de 0 até T1 ou T49 `positionLRScheduleEndIteration`; densification para em T2 `densifyUntilIteration`; decaimento de noise MCMC termina em T69 `mcmcNoiseDecayEnd`; upgrades de SH degree acontecem nas três marcas definidas em T21. Em densification classic, o sweet spot empírico está em 20 000–35 000 iterações (Sessões 1–32, testes V546); em MCMC, em 60 000–200 000 (V534). Aumentar bem acima dos valores do preset raramente traz mais qualidade — momentum Adam satura e, sem fim de LR decay, o loss estagna. Reduzir abaixo de 5 000 leva a geometrias não convergidas (density control não tem tempo de clone/split).

EM PALAVRAS SIMPLES

Por quanto tempo o app calcula. Mais iterações significa melhor resultado, mas em algum momento sem ganho perceptível e com muito mais tempo. As predefinições já dão um bom valor sem você pensar: Quick 1 000, Preview 5 000, Balanced 20 000, Quality 35 000, MCMC Quality 200 000. Se mexer por conta: MCMC pode ir alto (100K–200K); Classic não passe de 40K — não rende mais.

T2 densifyUntilIteration

DETALHES

Default: 15 000 (Initializer), 5 000 (`.full`), 160 000 (`.fullMCMC`) **Range:** 0 – **Defined in:**

TÉCNICO

Iteração em que a densification para. Até aqui, Gaussians são clonados, splitados e prunidos pelas regras dos campos `T11–T16` (Classic) ou `T67–T70` (MCMC); depois a contagem fica constante e só posições, rotações, escalas, opacidades e SH são otimizados (refinement). No paper original 3DGS o valor fica em 50% de `T1` ; no `.full` do RadianceKit em apenas 14% (5 000 de 35 000) — resultado dos experimentos V310/V338, que mostraram que após 5 000 iters mais densification piora o resultado (mais floaters, mais memória, sem ganho). Em MCMC, a relocação roda até 80 % de `T1` (V504b), porque MCMC não produz floaters daninhos. `T2` pequeno demais (< 1 000) gera poucos Gaussians; grande demais em Classic (> 50 % de `T1`) leva a overgrowth e outliers de RGB saturation (ver Outdoor Overtraining Findings).

EM PALAVRAS SIMPLES

Até quando o app pode criar novos Gaussians. Depois só refina os existentes. Em Classic 35K, 5 000 é o valor certo — acima fica mais embolado. Em MCMC, é 80 % do total (ou seja, 160 000 num 200K). Se mudar Quality, deixe este em paz.

Learning rates (T3–T10)

T3 positionLearningRate

DETALHES

Default: 0.00016 **Range:** 1e-7 – 1e-3 (recomendado) **Defined in:**

TÉCNICO

LR Adam para posição XYZ de cada Gaussian no início do treinamento (iter 0). Segue cosine annealing e desce ao longo do treinamento até T_4 `positionLearningRateFinal`. O default 0.00016 vem do paper 3DGS original (Kerbl et al., 2023) e no RadianceKit não escala com a resolução — posição se move em coordenadas do mundo, não em pixel space. Aumentar muito (> 0.0005) faz Gaussians pularem distâncias grandes e o loss fica instável; diminuir muito (< 0.00005) faz pontos mal inicializados nunca encontrarem o lugar. V414 testou dobrar o init \rightarrow loss 16.8 % pior; V544a confirmou o paper default como ótimo. Nota: em `.fullMCMC`, mantemos esse valor no default propositadamente — MCMC precisa de LR's constantes para sua lógica de relocação; tunar aqui não ajuda.

EM PALAVRAS SIMPLES

Quão rápido os splats podem se mover no espaço. O default é muito bem ajustado e quase nunca precisa mudar. Só se splats „balançam“ pelo bilho ou um canto inteiro fica em branco porque nada se moveu para lá, a learning rate seria um ponto a ajustar — mas tipicamente algo antes está errado (poses, nuvem inicial).

T4 positionLearningRateFinal DETALHES

Default: 0.0000016 (Initializer + paper), 0.000016 (`.full` , `.fullMCMC` — 10x maior) **Range:** 0 – **Defined in:**

 TÉCNICO

Valor final da curva cosine annealing de position LR. É atingido em `T1 maxIterations` ou em `T49 positionLRScheduleEndIteration` (se definido). O preset `.full` do RadianceKit usa 0.000016 — ou seja 10x o paper default 0.0000016. V420 mostrou que 0.5x do final (0.000008) piora o loss em 6.4 %; V414 mostrou que 2x o init piora em 16.8 %. O final alto não é trade-off, é escolha consciente: com decay forte demais, os Gaussians perdem na refinement a capacidade de se ajustar aos novos candidatos de densification. Via extensão V431/V433 a fase de schedule pode ser encurtada (`T49 < T1`), de modo que `T4` é atingido antes do fim e o restante roda em mini-LR constante — configuração típica: `T49 = 20 000` , `T1 = 35 000` , refinement em 0.000016 por 15 000 iters.

 EM PALAVRAS SIMPLES

Quão lenta a LR de posição vira no fim do treinamento. Ajustamos menos agressivo que o paper — splats podem tremer um pouquinho até o fim, deixa mais nítido. Se você mexer: maior = splats mais inquietos no fim; menor = não se ajustam mais quando novos aparecem.

T5 shDCLearningRate DETALHES

Default: 0.0025 (Initializer + paper), 0.005 (`.full` e todas predefinições MCMC — 2x) **Range:** 0.0001 – 0.05 **Defined in:**

 TÉCNICO

LR Adam para a parte DC (degree 0, albedo constante) da cor SH. SH-DC corresponde ao tom base independente de direção de uma Gaussian, a „cor base“. V176 e V188 acharam 2x o paper default ótimo — convergência de cor mais rápida, especialmente em treinos curtos (< 5 000 iters), onde SH-DC senão não toma forma. Diferente das LRs geométricas, SH-DC não tem decay; fica constante ou segue o decay opcional de extended phase em `T51` . V416 testou quadruplicar para 0.01 → 6.4 % pior com Adam beta2=0.99.

 EM PALAVRAS SIMPLES

Quão rápido a cor base de cada splat se ajusta. Quase nunca se muda — as predefinições têm o valor certo. Maior iria mais rápido mas pode dar cores instáveis.

T6 **shRestLearningRate** **DETALHES**

Default: 0.000125 (Initializer + paper), 0.00025 (.full e MCMC — 2×) **Range:** 0.000001 – 0.005
Defined in:

 **TÉCNICO**

LR Adam para SHs de ordem mais alta (degree 1, 2, 3 — as partes de cor dependentes de direção que dão highlights, reflexões e shading suave). 20× menor que T5 por convenção do paper, porque esses coeficientes crescem quadraticamente em número (3 para degree 1, 5 para 2, 7 para 3 → 15 floats totais por Gaussian) e sem LR menor saturaria a imagem. É liberado em dois passos — até a primeira marca de T21 shDegreeUpgradeIterations só degree 0 (só T5), depois 1, depois 2, depois 3. Valores baixos aqui são importantes em cenas com muita iluminação difusa; em superfícies muito brilhantes (laca de carro, água) não vale girar — a representação SH em si é limitada.

 **EM PALAVRAS SIMPLES**

Quão rápido os efeitos de cor dependentes de direção (reflexos, brilho) aprendem. Pequena por padrão, senão tudo brilha. Deixe como está — quem quer mais brilho se beneficia mais de MCMC e tempo maior que dessa LR.

T7 **opacityLearningRate** **DETALHES**

Default: 0.05 (Initializer + paper), 0.1 (.full, MCMC — 2×) **Range:** 0.001 – 1.0 **Defined in:**

 **TÉCNICO**

LR Adam para opacidade logit por Gaussian. O app guarda opacidade como float sem limite e aplica sigmoid em [0, 1]; a LR atua no logit space. O default do paper 0.05 foi restaurado após V50 (best single run L1 0.1664); V71 reverteu o 0.025 de V67. A duplicação V188 para 0.1 torna o pruning mais eficiente — Gaussians mortos caem mais rápido abaixo de T14 pruneOpacityThreshold. V418: 0.05 com Adam beta2=0.99 é 7.1 % pior que 0.1 — a interação com Adam não é trivial. Valores baixos (< 0.01) deixam „dead“ Gaussians para sempre e gastam memória; altos (> 0.5) causam explosão de opacidade, daí o clamp do logit em [-15, 3] no otimizador (ver „Opacity Explosion Prevention“ em CLAUDE.md).

 **EM PALAVRAS SIMPLES**

Quão rápido splats viram transparentes ou sólidos. Importante para a limpeza — splats que não contribuem precisam sumir rápido. Default ok; só profissionais mudam.

T8 `opacityLearningRateFinal` **DETALHES**

Default: 0.0 (= „sem decay“) **Range:** 0 ou 0.001 –
Defined in:

 **TÉCNICO**

Valor final opcional de cosine decay para LR de opacidade (V427). Em 0.0, decay desligado; LR fica constante em T7. V427 testou decay 0.1 → 0.01 → 11.5 % pior; reverted. Hipótese: LR constante de opacidade poderia oscilar na refinement, com splats de transparência certa sendo deslocados por ruído. Empiricamente falso — o clamp do logit já cuida. O campo fica disponível para futuros experimentos; runs MCMC muito longos (> 500K iters) poderiam se beneficiar.

 **EM PALAVRAS SIMPLES**

Se a LR de opacidade diminui ao fim. Padrão: não. Tentamos, ficou pior, deixamos desligado. Fique em 0.

T9 `scaleLearningRate` **DETALHES**

Default: 0.005 (Initializer + paper), 0.01 (`.full` ,
MCMC — 2x) **Range:** 0.0001 – 0.1 **Defined in:**

 **TÉCNICO**

LR Adam para os três componentes de escala em log space (RadianceKit guarda $\log(\text{scale})$ para escala ficar positiva). Paper default 0.005, dobrado no RadianceKit para 0.01 em prol de melhor convergência. V423: 0.005 com Adam beta2=0.99 → 18.7 % pior + poucos Gaussians (density control não clonava porque updates de escala eram lentos). Escala controla extensão de cada Gaussian — aprender rápido demais leva a Gaussians-agulha (T34 `scaleRatioPruneThreshold`); lento demais mantém splats compactos e density control divide demais.

 **EM PALAVRAS SIMPLES**

Quão rápido a forma dos splats se ajusta. Default bom. Se você sobe, dá splats-agulha — gotas longas e finas que viram floaters.

T10 rotationLearningRate DETALHES

Default: 0.001 (Initializer + paper), 0.002 (`.full` , MCMC — 2×) **Range:** 0.0001 – 0.05 **Defined in:**

 TÉCNICO

LR Adam para os quatro componentes de quaternion de cada Gaussian. O quaternion é re-normalizado a cada passo (norma $L2 = 1$); senão a matriz de covariância degenera. RadianceKit duplica o paper default em quality presets porque rotação tem magnitudes menores de gradiente que escala/posição (na esfera unitária cada passo é curto) e sem 2× ficaria sub-convergida em 35 000 iters. V188 documenta. Em cenas NeRF-Blender (Lego, Chair) rotação se destaca — bordas só se alinham depois de 5 000–10 000 iters.

 EM PALAVRAS SIMPLES

Quão rápido os splats aprendem a girar — ou seja, a se alinhar na superfície de um objeto. Default ok. Em outras palavras: se splats parecem blocos tortos em vez de se encaixar na superfície, é tempo de treino curto, não LR baixa.

Densification — Classic (T11–T16)

T11 densifyGradThreshold

DETALHES

Default: 0.000002 (Initializer, calibrado para 0.5x), 0.0000011 (`.full` , calibrado para 1.0x), 0.000004 (`.quickTest` , 0.25x), $2e-7$ (`.fullClassicPaper`)
Range: $1e-8$ – $1e-3$ (dependente da resolução) **Defined in:**

TÉCNICO

Limiar para a norma L2 do gradiente projetado em screen space `dMean2D` , acima do qual um Gaussian é marcado para clone/split. O valor absoluto depende direto da resolução de treino — `dMean2D` escala como $1/\text{resolução}^2$ (mais pixels = gradientes por pixel menores). Por isso cada degrau de T22 `trainingRenderScale` precisa de um limiar calibrado: $0.25x \rightarrow 4e-6$, $0.5x \rightarrow 2e-6$, $1.0x \rightarrow 5e-8$... $1.1e-6$ (`.full`). O paper default 0.0002 é normalizado em NDC e não comparável diretamente ao pipeline em world space do RadianceKit. Com o flag T52 `adaptiveDensifyThreshold` (V440) ativo, o valor sai do p98 da distribuição corrente — mas V440 testou em cenas reais e produziu 63 K Gaussians (perda catastrófica de pruning); o flag fica off. Q5 (T77–T79) traz alternativa via rolling median. **Não é seguro** — metade gera 2–4x mais Gaussians (pressão de memória, risco de OOM); dobrar pode sub-densificar.

EM PALAVRAS SIMPLES

Quão sensível o app é ao decidir se um splat está pouco representado e precisa ser multiplicado. Menor = sensível = mais splats. Maior = menos. É dos campos mais perigosos: baixo demais e o Mac enche de milhões de splats e pode crashar. Deixe em paz ou mude só em passos de 10 %.

T12 densifyFromIteration DETALHES**Default:** 500 **Range:** 100 – 5 000 **Defined in:** TÉCNICO

Primeira iter em que densification fica ativa. Antes, treinamento „nu“ na nuvem inicial SfM, sem novos Gaussians. O default 500 vem do paper 3DGS e dá à inicialização tempo de se estabilizar — com densify desde iter 0, pontos SfM mal posicionados clonariam várias vezes antes mesmo de achar o lugar. V349 testou 1000 → loss um pouco pior; o default é ótimo.

 EM PALAVRAS SIMPLES

Quando o app começa a clonar splats pela primeira vez. Antes ele só aprende com os pontos existentes. 500 é padrão — dá tempo de orientar antes de multiplicar.

T13 densifyInterval DETALHES**Default:** 100 (Initializer, MCMC), 200 (`.full`)**Range:** 50 – 1 000 **Defined in:** TÉCNICO

Quantas iters entre dois passos de densification. Paper default 100 — a cada 100 iters avalia candidatos de densify, clone/split e ao mesmo tempo prune ($\text{sigmoid}(\text{opacity}) < T_{14} \text{pruneOpacityThreshold}$) os mortos. V112: 200 é ótimo para `.full` — alivia a GPU (menos passos de reorganização) e dá a cada Gaussian mais tempo de se acomodar após clone. V417 testou 100 com $\text{beta2}=0.99$ → 5.8 % pior (957K Gaussians, sobredensify). Em MCMC, o mesmo campo é interpretado como intervalo de relocação; ver T67 `mcmcRelocationInterval` para a lógica MCMC.

 EM PALAVRAS SIMPLES

Com que frequência o app procura splats novos. 100 = frequente, 200 = médio. Maior = cada splat tem mais tempo de assentar entre multiplicações. Em 50, a GPU fica em constante reorganização sem ganho mensurável.

T14 `pruneOpacityThreshold` **DETALHES**

Default: 0.005 (Initializer, paper, MCMC), 0.001 (`.full`) **Range:** 0.0001 – 0.1 **Defined in:**

 **TÉCNICO**

Limiar de sigmoid(opacity) abaixo do qual o Gaussian é deletado no próximo passo de densify. Trabalha junto com `T7 opacityLearningRate` e o clamp do logit no otimizador. V393 baixou o default em `.full` de 0.005 para 0.001 — splats que só importam em ângulos exóticos sobrevivem mais tempo e contribuem para detalhe SH. V394 testou 0.0001 → um pouco pior (pouco pruning, memória desperdiçada). Importante: density control DEVE sempre prunar, mesmo quando a capacidade do buffer já está cheia por outros motivos (ver „Density Control Must Always Prune“ em CLAUDE.md) — senão Gaussians mortos acumulam e a contagem trava.

 **EM PALAVRAS SIMPLES**

Quando um splat é „transparente o bastante“ para ser apagado. 0.005 é o padrão do paper; em Quality temos 0.001 — damos mais chance. Isso melhora luz suave e sombras fracas. Subir para acima de 0.01 reduz contagem rápido — útil em pressão de memória, mas custa detalhe.

T15 `opacityResetInterval` **DETALHES**

Default: 3 000 (Initializer + paper), 100 000 (`.full` = efetivamente desativado), 200 000 (`.fullMCMC` = desativado) **Range:** 1 000 – 100 000+ **Defined in:**

 **TÉCNICO**

A cada quantas iters a opacidade de todos os Gaussians é resetada para baixo (0.01) — medida do paper 3DGS para reavaliar splats „congelados“. V194 mostrou que, com warmup + treino estocástico + LRs 2× do RadianceKit, o reset custa 5.5 % de qualidade; o clamp do logit já cobre a função do reset. Por isso desativado de fato em `.full` (100 000 > 35 000 = nunca dispara). V421 testou reset a cada 3 000 com $\beta_2=0.99$ → 4.9 % pior; reverted. Em `.fullClassicPaper` (Q1.5-A, testes paper-faithful) está propositalmente em 3 000 — um dos levers para atingir os budgets de Gaussian do paper.

 **EM PALAVRAS SIMPLES**

A cada quantas iters o app reseta a visibilidade de todos os splats para „quase invisível“ — um botão reset de opacidade. Em nós desativado (valor tão alto que nunca acontece), porque outros mecanismos cuidam. Só ative para experimentos paper-fiel.

T16 maxScreenSize DETALHES

Default: 0.0 (= desativado) **Range:** 0 (off) ou > 0
Defined in:

 TÉCNICO

Tamanho máximo em screen space (pixels projetados) que um Gaussian pode atingir antes de ser splitado. Em 0 (V48 testou e reverted) — density control do RadianceKit usa o threshold de escala world space do `dMean2D` em vez disso. Fica no catálogo porque experimentos futuros com Mip-Splatting (T74–T76) ou estratégias scene-specific podem usar. Ativar (> 0, p. ex. 20) forçaria splats muito grandes na tela a se dividirem — relevante em paredes grandes/lisas onde um splat gigante teria pouco detalhe.

 EM PALAVRAS SIMPLES

Limita o quão grande um único splat pode ficar na tela. Em nós off. Ligado, dividiria splats gigantes chatos (p. ex. parede). Mantenha off salvo experimentos específicos.

Loss (T17–T20)

T17 ssimWeight DETALHES

Default: 0.2 (Initializer + paper + `.full`), 0.05 (todas predefinições MCMC) **Range:** 0.0 – 1.0 **Defined in:**

 TÉCNICO

Peso da parte D-SSIM na função de loss combinada $loss = (1 - \lambda) * L1 + \lambda * D\text{-SSIM}$, com $\lambda =$ T17. O paper default 0.2 é ótimo para Classic — V383 testou 0.3 → 28.9 % pior; V373b confirmou 0.2 como sweet spot. Em MCMC, V521b/V534 acharam independentemente 0.05 ótimo — MCMC, por sua exploração estocástica, precisa de sinal L1 mais forte; pesos SSIM maiores diluiriam decisões de relocação. SSIM é caro (janela 11x11 sobre toda a imagem); RadianceKit usa implementação MPS abaixo de 1 ms por imagem 1080p. Q7-BayesOpt achou ótimos por cena entre 0.05 (`.outdoorPreset` : 0.082) e 0.171 (`.indoorPreset`).

 EM PALAVRAS SIMPLES

Quão importante o app considera „estruturas similares“ ao lado de „cada pixel certo“. 0.2 é padrão e dá uma imagem boa. Menor = mais pixel-perfect, mas transições mais suaves. Maior = mais similar estruturalmente, mas detalhes ficam soft. Confie nas predefinições.

T18 **ssimWeightRefinement** **DETALHES**

Default: 0.0 (= „sem troca, mantém ssimWeight“)

Range: 0 ou 0 – 1.0 **Defined in:**

 **TÉCNICO**

Valor SSIM opcional para a fase de refinement após T2 densifyUntilIteration. V428 testou 0.2 → 0.3 no refinement → 16 % pior (L1 e SSIM ambos pioraram); reverted, daí default 0.0. Hipótese: após densification, mais SSIM maximizaria nitidez estrutural. Empiricamente falso: subir SSIM reduz peso L1, e L1 é o sinal mais informativo no refinement final. O campo fica para futuros experimentos com perceptual loss (T60) ou edge loss (T19), onde composição de loss específica para refinement pode fazer sentido.

 **EM PALAVRAS SIMPLES**

Setting especial para a segunda fase de treinamento (refinement após multiplicação). Em 0.0: mesma ponderação SSIM. Mexer não rendeu nada, daí desligado.

T19 **edgeLossWeight** **DETALHES**

Default: 0.0 (= desativado) **Range:** 0 ou 0.001 – 1.0

Defined in:

 **TÉCNICO**

Loss experimental V437: peso de L1 em domínio de gradiente Sobel comparando bordas (Sobel da ground truth vs Sobel do render) adicionalmente a L1+SSIM. Hipótese: borda é pilar perceptual de qualidade; termo explícito encorajaria Gaussians a acertar bordas. Resultado: peso 0.1 → 11 % pior loss, 0.01 → neutro mas 10 % mais lento. O Sobel custa MPS forward extra em GT + render. Por isso permanentemente off. Caso futuro: cenas com bordas artificiais duras (arquitetura, móveis, renders) poderiam ganhar — mas Q7 Scene Class não escolheu, e sim escalou peso SSIM.

 **EM PALAVRAS SIMPLES**

Add-on experimental que dá peso extra a bordas. Empiricamente nada. Fica off.

T20 skyMaskingEnabled **DETALHES**

Default: false (Initializer e todas predefinições)

Range: boolean **Defined in:**

 **TÉCNICO**

Liga sky masking. A região de céu é mascarada em cada imagem via Apple Vision (VNGenerateForegroundInstanceMaskRequest) e o loss nessa área é zerado. Motivo: cenas externas sofrem quando pixels azuis/cinzas/brancos de céu fazem o app posicionar Gaussians lá — percebidos como floaters. Sem máscara de céu, o loss nessa área nunca é zero (céu varia levemente) e o app tenta infinitamente reproduzir com splats. A máscara Vision é calculada uma vez por câmera antes do treinamento e mantida em RAM. Tipicamente ativado junto com `T45 skyDomeEnabled` (lógica UI em Settings). Em cenas internas ou renders sintéticos, deixe off — a máscara identificaria tetos ou paredes como „céu“.

 **EM PALAVRAS SIMPLES**

Liga um modo especial para tomadas externas: o céu é ignorado no treinamento, para o app não tentar reconstruí-lo com splats. Recomendado em cenas externas. Em internas ou renders 3D Blender, deixe off.

Progressão de SH degree (T21)

T21 shDegreeUpgraderIterations

DETALHES

Default: [1_000, 2_000, 3_000] (Initializer),
 [2_000, 5_000, 8_000] (.full , MCMC),
 [1_000, 2_000] (.preview — degree 3 pulado)
Range: [Int] , cada um em [0, maxIterations] ,
 monotonicamente crescente **Defined in:**

TÉCNICO

Iters em que o SH degree ativo passa de 0→1, 1→2, 2→3. Antes da primeira marca só DC (T5 shDCLearningRate); após primeira, DC + 3 coefs degree 1; após segunda, + 5 degree 2; após terceira, todos os 15. Memória por Gaussian cresce em degraus — 4 floats → 16 → 36 → 64. Quality presets atrasam os upgrades vs Initializer (V228) porque a geometria deve estabilizar antes da cor de frequência mais alta. V384 testou [1K, 2K, 3K] em .full → 9.3 % pior — confirma o atraso. .preview para em degree 2 porque degree 3 não converge em 5 000 iters e só gasta otimizador. Q6 (T80–T81) oferece lógica alternativa de currículo que sobrescreve essa lista dinamicamente.

EM PALAVRAS SIMPLES

Quando o app aprende que cores variam por ângulo de visão (highlights, reflexos). Tarde — forma primeiro, cor depois. Os valores nas predefinições funcionam bem. Não mexa salvo se souber por quê.

Performance (T22–T25)

T22 trainingRenderScale

DETALHES

Default: 1.0 (Initializer, `.full`, MCMC, Scene Class), 0.5 (`.preview`), 0.25 (`.quickTest`) **Range:** 0.05 – 2.0 (típico 0.25, 0.5, 1.0) **Defined in:**

TÉCNICO

Resolução de render no treinamento relativa à resolução original das imagens. Em 0.5, cada imagem é reduzida a 50 % largura × 50 % altura (25 % dos pixels) e o render Gaussian roda nessa resolução. Reduz memória e compute quadraticamente. Importante: T11 `densifyGradThreshold` deve combinar — magnitudes escalam com $1/\text{resolução}^2$, por isso `.quickTest` (0.25x) tem threshold muito maior ($4e-6$) que `.full` (1.0x, $1.1e-6$). RadianceKit avisa em imagens grandes e ajusta automaticamente — alvo 3 MP. Em 4K extremo, 0.5 ou 0.25 fazem sentido; senão CPU compaction.

EM PALAVRAS SIMPLES

Quão grandes as imagens no treinamento são. 1.0 = original, 0.5 = metade. Metade = 4x mais rápido, mas detalhes finos somem. As predefinições escolhem; em imagens muito grandes (acima de 12 MP), o app baixa automaticamente.

T23 resolutionWarmupScale

DETALHES

Default: 0.0 (= desativado) **Range:** 0 ou 0.1 – **Defined in:**

TÉCNICO

Otimização V133: treina a fase de densification (iter 0 a T2) numa resolução mais baixa que a refinement. V308 desligou em `.full`, porque com T22 = 1.0 e cosine annealing o ganho de tempo era marginal e a qualidade sofria. Fica no catálogo porque em entradas 4K com treinos longos pode voltar a fazer sentido — Q6 Curriculum (T80) pegou lógica similar mas acoplada ao schedule de LR. Se ativado e T80 `curriculumResolutionRamp` também true, Q6 vence e sobrescreve.

EM PALAVRAS SIMPLES

Special: aprende com imagens pequenas na primeira metade do treinamento, grandes na segunda. Economiza tempo. Off porque a variante Q6 mais nova faz isso melhor.

T24 tileSize **DETALHES****Default:** 16 **Range:** 8, 16, 32 **Defined in:** **TÉCNICO**

Tamanho dos tiles de rasterização em pixels. O rendering Gaussian é tile-based: a imagem é dividida em tiles 16×16; cada tile coleta seus Gaussians relevantes, ordena por profundidade e blendea. 16 é o padrão de quase todas as implementações 3DGS e hard-coded nos kernels Metal; mudar pediria recompilação de shaders — não efetivo no estado atual. Fica como campo caso versão futura suporte tile size dinâmico.

 **EM PALAVRAS SIMPLES**

Parâmetro interno de render. Padrão 16, não mexer.

T25 throttleDelayMs **DETALHES****Default:** 0 (Initializer, `.full`, MCMC, Scene Class), 0 (`.preview`) **Range:** 0 – 100 **Defined in:** **TÉCNICO**

Delay artificial entre iters em ms. 0 = full speed (padrão). Maior torna o Mac mais „usável“ no treinamento porque GPU/CPU pegam pausas — responsividade de outros apps sobe, tempo de treinamento sobe linearmente. Típicos: 1–2 ms (throttling leve, +5 %), 5 ms (médio, +15 %), 10+ ms (eco, potencialmente dobro). Oferecido no Inspecor em „Performance“ mas fora da view padrão — ver backlog `dev_ux-backlog.md` que sugere removê-lo do Expert View porque, mal entendido, alonga muito o treinamento.

 **EM PALAVRAS SIMPLES**

Quantos ms de pausa o app tira entre passos. 0 = sem pausa, mais rápido possível. Maior deixa o Mac mais usável mas alonga o treinamento. Em M3 Ultra ou Mac Studio, deixe 0; em MacBook Air, 2 ou 5 são bons.

Diagnóstico e preparação da nuvem de pontos (T26–T30)

T26 depthDistortionWeight

DETALHES

Default: 0.0 (= desativado) **Range:** 0 ou 0.0001 – 0.05 **Defined in:**

TÉCNICO

Experimental V366: peso de um loss de regularização de distorção de profundidade. Penaliza Gaussians que ao longo de um ray estão escalonados em profundidade mas conceitualmente pertencem à mesma superfície — encoraja distribuições concentradas e reduz floaters. Testes: 0.01 → 4.5 % pior, 0.001 → 8.1 % pior. Vantagem teórica de melhorar consistência multi-view não vira ganho de L1, porque a hipótese implícita é geometria SfM correta. Na prática, SfM é o componente mais fraco, não o empilhamento. Disponível para datasets multi-view com poses muito limpas (Synthetic, Mip-NeRF 360 com ground truth).

EM PALAVRAS SIMPLES

Feature experimental contra splats empilhados no mesmo lugar. Não ativo, porque os testes não ajudaram.

T27 singleViewOverfit

DETALHES

Default: false **Range:** boolean **Defined in:**

TÉCNICO

Flag de diagnóstico: em true, cada iter usa o camera index 0 obrigatoriamente. Sentido: se o modelo não consegue overfitar uma única view (loss em view 0 não vai a zero após 10 000 iters), há bug fundamental no forward/backward. Usado intensamente no desenvolvimento dos shaders Metal e do differentiable rasterizer — fase V42–V47. Hoje só como sanity check para quem modificou backend e quer um regression test. Via CLI: `--single-view`.

EM PALAVRAS SIMPLES

Modo de teste para devs. Verifica se o app consegue aprender de UMA imagem. Para usuários normais irrelevante; deixe off.

T28 maxCameras DETALHES

Default: 0 (= „todas as câmeras“) **Range:** 0 ou 1 – N **Defined in:**

 TÉCNICO

Limite diagnóstico V43: treina só com as primeiras N câmeras, ignora o resto. Hipótese original: muitas câmeras gerariam conflitos de gradiente (muitos sinais de loss contraditórios para a mesma Gaussian). Resultado: nenhum benefício sistemático com limite artificial — mais frames quase sempre trazem mais qualidade. Fica como CLI (`--max-cameras N`) para experimentos pontuais. Sem exposição na UI.

 EM PALAVRAS SIMPLES

Campo diagnóstico para devs — usar só as primeiras N imagens. Usuário normal não precisa, deixe 0 = todas. Mais imagens = melhor (ver `feedback_more-frames-better.md`).

T29 maxInitialPoints DETALHES

Default: 0 (= „todos os pontos SfM“) **Range:** 0 ou 1 000 – 200 000+ **Defined in:**

 TÉCNICO

Salvaguarda V54: limita pontos SfM iniciais. Reconstruções COLMAP densas podem ter > 60 000 pontos, o que com escalas iniciais grandes gera 200–300 Gaussians por pixel-overlap — vira „neblina“ e o treino não converge. Subsampling a 16 000 (hard cap na engine de treinamento) traz densidade inicial ao nível do 3DGS de referência e reduz overlap dramaticamente. Setado automaticamente em SfMs muito densos; via CLI:

`--max-points N` .

 EM PALAVRAS SIMPLES

Quantos pontos iniciais da reconstrução são usados. Em reconstruções muito densas (acima de 60 000), o app limita automaticamente a 16 000 — se não muita neblina no início. Você não precisa setar.

T30 cameraClusterOutlierMultiplier DETALHES

Default: 10.0 (todas predefinições — nunca sobrescrito) **Range:** 1.0 – 100.0 **Defined in:**

 TÉCNICO

Multiplicador para o filtro de outliers de cluster de câmeras, introduzido na Fase 3.10 A.1. Antes do treinamento, a engine calcula o centroide das posições de câmera e a distância máxima de câmera ao centroide. Pontos SfM cuja distância ao centroide excede $\text{multiplier} \times \text{maxCameraDistance}$ são descartados. 10x preserva comportamento pré-Fase 3.10. Bug sutil: SfM mais apertado → menor → menor limiar → mais pontos descartados como outliers. SfM mais largo → maior limiar → menos descartados. É uma das causas da anti-correlação funnel-vs-treinamento da Fase 3.9: melhor SfM pode dar treinamento pior porque mata pontos iniciais demais. Campo via CLI (`--camera-cluster-outlier-multiplier`) para sweeps A.3; não exposto na UI. < 5 muito restritivo; > 20 sem efeito.

 EM PALAVRAS SIMPLES

Filtro especial que descarta pontos da reconstrução longe da nuvem de câmeras. 10 = generoso, mantém quase tudo. Aumentar pode ajudar se pontos distantes (montanhas ao longe) aparecem como grumos flutuantes. Reduzir só em emergência — perde detalhe distante.

Regularização (T31–T37)

T31 coarseToFineBlurRadius DETALHES

Default: 0 (= desativado) **Range:** 0 ou 1 – 10 **Defined in:**

 TÉCNICO

Experimental V369: raio de box blur aplicado à ground truth no início da densification, reduzido linearmente até o fim de densification (T2) para 0. Hipótese: treino coarse to fine — estruturas grossas primeiro, detalhes depois — daria geometria mais estável. Testes: $r=3 \rightarrow 9.6\%$ pior, $r=1 \rightarrow 5.1\%$ pior. Motivo: density control decide com base em gradientes no domínio da imagem; blur reduz justamente os sinais para „clone aqui“. Fica no catálogo para futuros esquemas de density control.

 EM PALAVRAS SIMPLES

Modo experimental „primeiro grosso, depois detalhado“. Não trouxe nada, fica off.

T32 **scaleRegWeight** **DETALHES**

Default: 0.0 (= desativado) **Range:** 0 ou 0.0001 – 0.05 **Defined in:**

 **TÉCNICO**

Experimental V370: regularização L1 sobre escala world space. Pune Gaussians grandes demais — evita „mega splats“ que cobrem paredes inteiras. Testes: 0.01 → 200 % pior loss (2M Gaussians, explosão total); 0.001 → 214 % pior. Motivo: regularização de escala briga com density control — escalas menores significam mais Gaussians necessários, density control divide com frequência, mais compute de gradiente. Disabled mas documentado para experimentos Mip Splatting (T74): nesse contexto um lower bound de escala pode fazer sentido.

 **EM PALAVRAS SIMPLES**

Regularização que força splats a ficarem pequenos. Em testes gerou explosões (milhões de splats). Não ative.

T33 **anisotropyRegWeight** **DETALHES**

Default: 0.0 (= desativado) **Range:** 0 ou 0.0001 – 0.05 **Defined in:**

 **TÉCNICO**

Experimental V445: penalidade sobre razão $\max(\text{scale})/\min(\text{scale})$, para evitar Gaussians-agulha percebidos como floaters. Testes: 0.01 → 69 % pior, 0.001 → 15 % pior. Motivo: regularização força splats para forma „redonda“, o que em superfície plana (parede, mesa, chão) é justo o errado — ali um Gaussian achatado e largo é mais eficiente que esférico. Disabled. V549f ofereceu via T34 `scaleRatioPruneThreshold` uma alternativa mais direcionada que também foi revertida.

 **EM PALAVRAS SIMPLES**

Regularização que pune splats compridos. Parece sensato, mas piorou nos testes. Off.

T34 **scaleRatioPruneThreshold** **DETALHES**

Default: 0.0 (= desativado) **Range:** 0 ou 5.0 – 100.0 (típico 10.0 – 30.0) **Defined in:**

 **TÉCNICO**

Pruning experimental pós-treinamento que apaga Gaussian cuja razão $\max(\text{scale})/\min(\text{scale})$ excede limiar. Foco em floaters tipo „needle/disc“ que regularização sozinha não elimina. Em testes removeu floaters mas também splats úteis em paredes e chãos — a imagem ficou esburacada. Off por padrão; CLI (`--scale-ratio-prune N`) para experimentos. Valores recomendados se testar: 30 (conservador, só extremos), 10 (agressivo, custa detalhe).

 **EM PALAVRAS SIMPLES**

Tentativa de filtrar splats muito longos após o treinamento. Net-negative nos testes — floaters somem, mas detalhe também. Off.

T35 **opacityRegWeight** **DETALHES**

Default: 0.0 (= desativado) **Range:** 0 ou 0.0001 – 0.05 **Defined in:**

 **TÉCNICO**

Experimental V446: penalidade BCE que puxa opacidade para 0 ou 1 (afasta do „meio-transparente“). Hipótese: opacidade mais nítida melhoraria clareza. Combinado com T33 → regularização custa qualidade; ambos desligados. Disabled. Bug 1.4.3-beta: o default do initializer estava errado em 0.01, o que causou mass extinction (460K → 5 Gaussians numa iter). Desde 1.4.4 fixado em 0.0.

 **EM PALAVRAS SIMPLES**

Regularização que torna splats totalmente transparentes ou opacos. Não traz nada e pode ser perigoso (bug mass extinction de 1.4.3). Mantenha em 0.

T36 `opacityDecayFactor` **DETALHES**

Default: 0.0 (Initializer = desativado), 0.9995
(`.full`, `.classicBalanced` — padrão HTGS)

Range: 0 (off) ou 0.95 – 1.0 **Defined in:**

 **TÉCNICO**

Implementação V546 do esquema HTGS (Hierarchical Time Gating, Eurographics 2025): a cada `T37 opacityDecayInterval` iters, `sigmoid(opacity)` é multiplicado por esse fator. 0.9995×100 aplicações = 95 % retenção por fase de densification — pressão suave mas constante para baixo que leva Gaussians fracos confiavelmente ao `T14 pruneOpacityThreshold`. Resultado: 14 % melhor L1 em Horse Full (avg 3 trials V546) vs V438 sem decay. Só ativo na fase de densification (até `T2`); depois roda sem decay, para manter estáveis as opacidades estabelecidas no refinement. Em MCMC não usado (MCMC tem `T67 mcmcRelocationInterval` + `T68 mcmcDeadOpacityThreshold`).

 **EM PALAVRAS SIMPLES**

„Esmacimento suave“ de todos os splats no tempo. Faz splats inativos virarem transparentes mais rápido para sumir na limpeza. Foi o lever Q mais importante do update V546: 14 % melhor. Embutido em Quality. Não recomendamos mexer — calibrado fino.

T37 `opacityDecayInterval` **DETALHES**

Default: 50 **Range:** 10 – 500 **Defined in:**

 **TÉCNICO**

Intervalo de iters de `T36 opacityDecayFactor`. Default HTGS 50, mantido em `.full`. Intervalos longos (>200) cancelam parte do efeito — entre duas aplicações há gradiente suficiente para opacidade subir de novo. Curtos (<20) são agressivos demais. Só ativo na fase de densification.

 **EM PALAVRAS SIMPLES**

Com que frequência o „fade“ é aplicado. 50 = a cada 50 iters um passinho. Ok.

Refinement (T38–T44)

T38 gradientAccumulationSteps

DETALHES

Default: 1 (= „uma view por passo Adam“) **Range:** 1 – 8 **Defined in:**

TÉCNICO

Feature V424: número de views cujos gradientes são acumulados antes de um Adam step. Com `> 1`, app roda numa trilha „unfused“ backward project que soma gradientes em buffer separado; a aplicação final escala por $1/N$ para manter magnitude. V424 testou 2-view → neutro em qualidade mas 10 % mais lento (unfused é mais caro que fused). Reverted para `.full`, mas em MCMC é usado — `.fullMCMC` roda com, e V544a mostrou que com isso o gap a Classic cai para 5 % (em vez de 11 %). No initializer 1, no preset atual 1, fica via CLI (`--accum-steps N`).

EM PALAVRAS SIMPLES

Quantas imagens o app olha antes de ajustar os splats. 1 = cada imagem isolada. Maior = ver várias e aplicar média. No padrão não rende; em MCMC, 2 pode ajudar pouco.

T39 testViewIndices

DETALHES

Default: `[]` (= vazio, todas as views vão para treinamento) **Range:** `Set<Int>`, subconjunto de índices de câmera **Defined in:**

TÉCNICO

Feature V546: set de índices que NÃO vão para treinamento, mas viram holdout para PSNR/SSIM/LPIPS. Setado automaticamente com `--benchmark`: cada 8ª view a partir do índice 0 (padrão LLFF, idêntico ao Mip-NeRF 360 e ao 3DGS paper). Sem benchmark, vazio — treinamento usa todas. **Cuidado:** setar sem entender os índices pode invalidar o benchmark (p. ex. índices $> N$ com só $N-50$ views → sem holdout → sem avaliação). No export de predefinição, `testViewIndices` não persiste, porque depende da cena e poderia deixar valores sem sentido entre datasets.

EM PALAVRAS SIMPLES

Quais imagens ficam de fora do treinamento para usar depois como medidor de qualidade. Você não seta; `--benchmark` faz isso (cada 8ª). Setar manualmente: perigoso, pode falsificar o benchmark.

T40 refinementPruneInterval **DETALHES**

Default: 0 (= desativado) **Range:** 0 ou 100 – 5 000
Defined in:

 **TÉCNICO**

Feature V425: a cada N iters durante refinement (após T2), um passo extra de pruning roda e remove Gaussian com sigmoid(opacity) < T41 refinementPruneOpacityThreshold. Sentido: na densification havia density control regular; depois não — Gaussians cuja opacidade segue caindo ficam no buffer. V425 testou e reverteu: o pruning extra correlacionou com V426 (two-phase densification, também caiu em cascade failure de 0 Gaussians). Disabled. CLI disponível; se ativar, 1 000 ou 2 000 são razoáveis.

 **EM PALAVRAS SIMPLES**

Limpeza extra durante a fase de refinement. Sem ganho, fica off.

T41 refinementPruneOpacityThreshold **DETALHES**

Default: 0.0 (= „usa T14 “) **Range:** 0 ou 0.001 – 0.1
Defined in:

 **TÉCNICO**

V425b: threshold separado para pruning de refinement. Após densification, a maioria dos Gaussians tem opacidade > 0.001, e o T14 pruneOpacityThreshold padrão seria frouxo. Se T40 ativo, este campo define o threshold próprio. Em 0.0, usa T14. Só relevante com T40 > 0.

 **EM PALAVRAS SIMPLES**

Threshold para a limpeza extra do refinement (ver T40). Ambos os campos off, irrelevante.

T42 midTrainingCompactificationIterations **DETALHES**

Default: `[]` (= desativado) **Range:** `[Int]`, valores em (densifyUntilIteration, maxIterations) **Defined in:**

 **TÉCNICO**

Feature V549: iters explícitas na refinement em que roda compactification (mesma lógica de T56 postTrainingCompactification). Sentido: refinements longos acumulam confetti/floateres cujo SH overfita artefatos por view. Configuração típica: `[10000, 20000, 30000]` para Classic 40K. **Mas:** A/B V549 em family-dataset mostrou em todas as configs piores L1: `[10K, 20K, 30K]@0.01` → -48 % count mas +36 % L1; `[20K, 30K]@0.005` → -44 % count mas +45 % L1; `[20K, 30K]@0.001` → -17 % count mas +87 % L1. Daí disabled. CLI `--mid-compact "10000, 20000"` disponível para o trade-off „menos floater no viewport mas imagem mais embolada“.

 **EM PALAVRAS SIMPLES**

Limpezas no meio do caminho. Em testes piorou o resultado (menos floater, menos detalhe). Off, mas ligável via CLI se floater te incomoda mais que matiz amassado.

T43 frustumCullEnabled **DETALHES**

Default: `false` **Range:** boolean **Defined in:**

 **TÉCNICO**

Feature V549b: após treinamento remove Gaussians fora da união dos frusta de câmeras de treinamento. Esses Gaussians nunca foram restringidos pelo loss e são sempre floateres. Efetivo em cenas onde a novel view fica atrás ou ao lado da trajetória (p. ex. costas de drone linear) — floateres ali não aparecem em treino mas aparecem ao navegar em 3D. A/B V549b em drones positivos, daí opt-in. Default false, porque em object captures com orbit completa a união cobre tudo — ofereço em Settings „Floater Reduction“ e implicitamente em P9 Outdoor testado via T44 frustumCullExpansion (Q7-BayesOpt não ativou, porque Outdoor sky dome resolve melhor o mesmo problema).

 **EM PALAVRAS SIMPLES**

Filtro especial para drones ou capturas lineares: após treinamento, deleta splats que não foram „vistos“ em câmera. Opcional em Settings. Em capturas de objeto simples, desnecessário.

T44 frustumCullExpansion **DETALHES****Default:** 1.1 **Range:** 1.0 – 2.0 **Defined in:** **TÉCNICO**

Margem NDC para T43 `frustumCullEnabled`. 1.0 cortaria exato na borda — splats ondulantes ali ficariam cortados demais. 1.1 = 10 % de padding além do framing exato — tolerância para pixels de borda que numa novel view ligeiramente deslocada apareceriam. Acima de 1.2 torna o cull quase inútil (o frustum estendido cobre demais).

 **EM PALAVRAS SIMPLES**

Quão estrito o filtro de cima corta. 1.1 = margem de segurança na borda. Deixe.

Sky Dome (T45–T48)**T45 skyDomeEnabled** **DETALHES****Default:** false (Initializer + todas predefs exceto P9 Outdoor) **Range:** boolean **Defined in:** **TÉCNICO**

Feature V549e: antes do início do treinamento, uma nuvem esférica (fibonacci sphere com T46 pontos) é gerada, posicionada num raio de T47 `skyDomeRadiusMultiplier × scene_extent` em torno do centro e inicializada com cores dos pixels de céu mascarados (ver T20 `skyMaskingEnabled`). Esses Sky Dome Gaussians ficam no começo do buffer e são „congelados“ no treinamento (gradientes de posição/escala/rotação = 0; só SH e opacidade são otimizados). Efeito: em vez de „confetti“ preto distante, usuário vê céu real em novel views. MVP V549e funciona muito bem em drones e landscapes; em P9 Outdoor on por padrão. Em cenas internas off — esfera ficaria sem sentido fora do quarto.

 **EM PALAVRAS SIMPLES**

Liga uma „cúpula de céu“ artificial em torno da cena. Deixa tomadas externas muito mais bonitas: em vez de grumos pretos na borda, mostra o céu real. Obrigatório em drones e landscapes; sem sentido em internas.

T46 skyDomeSampleCount **DETALHES**

Default: 5 000 **Range:** 1 000 – 50 000 (típico 2 000 – 10 000) **Defined in:**

 **TÉCNICO**

Número de pontos fibonacci sphere no Sky Dome. Maior → dome mais denso (melhor em altas resoluções e muito céu visível), mais memória. 5 000 é o sweet spot para 4K; em resoluções menores 2 000–3 000 bastam. Os pontos são inicializados pela distância cosseno ao view vetor de cada câmera e os pixels sky maskeados correspondentes — samples cujo view cone não vê nenhuma câmera ficam com baixa opacidade inicial, mas no treinamento não mudam (congelados).

 **EM PALAVRAS SIMPLES**

Quão denso é o céu artificial. 5 000 pontos basta normalmente. Mais = melhor transição à distância mas custa memória.

T47 skyDomeRadiusMultiplier **DETALHES**

Default: 30.0 (Initializer + maioria das predefinições), 59.0 (P9 Outdoor, ótimo Q7-BayesOpt) **Range:** 5.0 – 200.0 **Defined in:**

 **TÉCNICO**

Raio do Sky Dome relativo à extensão da cena (= distância média entre câmeras). 30 = esfera 30× o diâmetro da nuvem de câmeras. Pequeno demais (< 5) → dome interfere com a cena (p. ex. um splat dome no plano principal); grande demais (> 100) → perda de precisão float32 nas posições, render glitches distantes. Q7-BayesOpt em Bicycle (Mip-NeRF 360) achou 59.0 como ótimo scene-specific para outdoor — indica que 30 é pequeno demais para landscapes profundas; pixels do dome aparecem como „parede“ na borda.

 **EM PALAVRAS SIMPLES**

Quão longe o céu artificial deve estar. 30 = bem longe. Em landscapes grandes, 50–60 é melhor (Outdoor preset faz automaticamente). Pequeno demais é como ter grumos colados na lente.

T48 frozenGaussianCount **DETALHES**

Default: 0 (= sem Gaussians congelados) **Range:** 0 ou 1 – **T46** **Defined in:**

 **TÉCNICO**

Número de Gaussians no começo do buffer cujos gradientes de posição/escala/rotação são zerados no otimizador — permanecem espacialmente rígidos. Density control não os clona/splita/pruna. Usado para injeção de Sky Dome (ver **T45**): quando dome on, este campo é setado automaticamente para **T46** `skyDomeSampleCount`. Setar manualmente é possível (p. ex. congelar nuvem LiDAR pre-posicionada), mas não exposto na UI. Importante: os primeiros N Gaussians do buffer são sempre os congelados — ordem manda, não índice.

 **EM PALAVRAS SIMPLES**

Quantos splats no início ficam fixos. Setado automaticamente para a contagem do Sky Dome quando ele está on. Não precisa mexer.

Adam + LR Schedule (T49–T55)**T49 adamResetIteration** **DETALHES**

Default: 0 (= desativado) **Range:** 0 ou 100 – **Defined in:**

 **TÉCNICO**

Feature V430: iter em que momentos Adam (m1, m2) são zerados. Correção de bias passa a rodar com `(iter - adamResetIteration)` em vez de `iter`. V430 testou reset em 5 000 (pós densification) → 12.8 % pior loss. Motivo: momento Adam acumulado na densification carrega informação sobre magnitudes típicas e acelera o refinement. Descartar custa as primeiras 500 iters de convergência. Disabled. Fica CLI para experimentos.

 **EM PALAVRAS SIMPLES**

Botão reset da „memória“ interna do Adam. Em testes piorou, fica off.

T50 positionLRScheduleEndIteration DETALHES

Default: 0 (Initializer = „usa maxIterations“), 20 000 (`.full` — cosine termina em 20K embora `maxIter=35K`), 30 000 (`.fullClassicPaper`) **Range:** 0 ou 1 000 – **Defined in:**

 TÉCNICO

Feature V431: iter em que o cosine annealing para position LR atinge mínimo. Em 0 = igual a `T1`. Em > 0 , roda até esse valor e depois fica em `T4` constante. Permite uma „extended refinement phase“ com LR mínima mas constante — refina posições devagar sem decay extra. `.full` faz (schedule termina em 20K, treinamento até 35K); V434c/V434d confirmaram: 15K e 25K parecidos, 20K marginalmente ótimo. Em combinação com `T51` permite modificar também as não-position LRs na extended phase.

 EM PALAVRAS SIMPLES

Quando o app para de baixar a LR de posição. Se menor que a iter máxima, depois roda com mini-LR constante — refina devagar mas estável. Embutido em Quality; não precisa mexer.

T51 extendedPhaseLRDecay DETALHES

Default: 0.0 (= desativado, LRs constantes) **Range:** 0 ou 0.01 – 1.0 **Defined in:**

 TÉCNICO

Feature V433: multiplicador mínimo para as LRs não-position (Scale, Rotation, Opacity, SH) na „extended phase“ — depois de `T50` atingido. Em 0.1, escala/rotação/opacity/SH fazem cosine decay de 1.0 (LR padrão) para 0.1× do padrão. Em 0.0 (default), constantes. V457 testou decay total (0.0 = decay até zero) vs sem decay → avg 0.0400 (2 runs) = mesmo loss que V438 sem decay. Comportamento mais limpo com decay, mas não mensuravelmente melhor. Daí disabled. CLI: `--nonpos-lr-scale F`.

 EM PALAVRAS SIMPLES

Na fase tardia, baixar também as LRs de cor e forma. „Estabiliza“ o treinamento mas empiricamente não melhora. Off.

T52 adaptiveDensifyThreshold DETALHES**Default:** false **Range:** boolean **Defined in:** TÉCNICO

Experimental V440: em true, calcula p98 da distribuição corrente de gradientes a cada densify step e usa como threshold dinâmico (clamped a no mínimo 0.5× do T11). Hipótese: adaptação a fase tornaria density control mais robusto. V440 testou e reverteu: drop catastrófico para 63K Gaussians (mass pruning, porque p98 nas primeiras iters é altíssimo e quase nada o ultrapassa). O threshold fixo já é bem calibrado; dinâmico estraga. Q5 (T77) traz lógica adaptive alternativa via rolling median.

 EM PALAVRAS SIMPLES

Versão adaptiva do threshold de densify. Em testes catastrófica (contagem caiu para 63K). Off. Q5 tem variante melhor.

T53 mergeAfterDensification DETALHES**Default:** false (Initializer), true (`.full` , `.classicBalanced` , `.fullClassicPaper`) **Range:** boolean **Defined in:** TÉCNICO

Feature V438: no fim da fase de densification (iter T2), pass único de merge agrupa Gaussians próximos com escala e cor parecidas. Reduz contagem em 5–15 % sem perda visual. Sentido: clones produzem clusters quase idênticos que não contribuem — merge libera capacidade do otimizador. Padrão em quality classic. Em MCMC não usado — relocação não permite que esses clusters se formem.

 EM PALAVRAS SIMPLES

No fim da fase de multiplicação, juntar clones quase idênticos. Reduz dados sem efeito visível. Padrão em Quality.

T54 densifyPhase2FromIteration **DETALHES****Default:** 0 (= desativado) **Range:** 0 ou T2 – T1**Defined in:** **TÉCNICO**

Experimental V426: segunda fase de densification que começa aqui após pausa de refinement e roda até T55 . Hipótese: após refinement, magnitudes de gradiente são estáveis e podem indicar áreas que ainda precisam de Gaussians. V426 testou e revertido: two-phase densification caiu em cascade de 0 Gaussians (com V425 refinement pruning combinado, destruiu o buffer). Disabled. CLI disponível.

 **EM PALAVRAS SIMPLES**

Segunda rodada de multiplicação após pausa. Em testes aniquilou os splats. Off.

T55 densifyPhase2Untilliteration **DETALHES****Default:** 0 **Range:** 0 ou T54 – T1 **Defined in:** **TÉCNICO**

Fim da two-phase densification V426. Só relevante se T54 > 0 . Ambos disabled.

 **EM PALAVRAS SIMPLES**

Fim da segunda rodada (ver T54). Ambos off.

Pós-processamento + Apple AI (T56–T60)

T56 postTrainingCompactification

DETALHES

Default: true (em todas predefinições de produção), false (`.quickTest` , `.preview`) **Range:** boolean **Defined in:**

TÉCNICO

Feature V443: após o fim do treinamento, Gaussians com $\text{sigmoid}(\text{opacity}) < 0.01$ são removidos (já não contribuem). Reduz contagem em 58 % e tamanho de export em 55 % sem perda visual. Padrão em produção — resultado entregue o mais compacto possível. Em `.quickTest` off, porque diagnose não exporta. Diferente de T42 `midTrainingCompactificationIterations` (V549), aqui é só no fim — refinement pode usar todos os Gaussians.

EM PALAVRAS SIMPLES

Limpeza após treinamento: splats quase invisíveis somem. Arquivo de export fica metade sem perda. Feature obrigatório; só off em runs de diagnóstico.

T57 metalFXUpscaling

DETALHES

Default: false **Range:** boolean **Defined in:**

TÉCNICO

Feature V444: ativa Apple MetalFX Spatial Upscaler no output do viewport 3D em vez de bilinear. Se a resolução de treinamento $<$ viewport (p. ex. treino em 0.5x, viewport em plena), MetalFX entrega imagem bem mais nítida. Live no viewport, sem retraining. Mutuamente exclusivo com T58 `mpsLanczosScaling` — MetalFX vence. Recomendação: ligue quando o viewport parece „lavado“ para o detalhe esperado.

EM PALAVRAS SIMPLES

Sharpening baseado em ML da Apple no viewport 3D. Ajuda quando você treinou em resolução menor e mostra em full. Toggle live.

T58 mpsLanczosScaling DETALHES**Default:** false **Range:** boolean **Defined in:** TÉCNICO

Feature V444: MPSImageLanczosScale para upscaling do viewport em vez de bilinear. Lanczos é re-sampling sinc-based clássico, bem mais nítido que bilinear com overhead mínimo. Toggle live. Sobrescrito por T57 quando ambos on.

 EM PALAVRAS SIMPLES

Método clássico de upscaling pelo MPS (Lanczos). MetalFX (T57) é baseado em ML e em geral melhor; Lanczos é alternativa menos agressiva.

T59 livePreviewInterval DETALHES**Default:** 50 (Initializer e maioria das predefinições)**Range:** 0 (off) ou 10 – 5 000 **Defined in:** TÉCNICO

Com que frequência o viewport é atualizado durante o treinamento. 50 = a cada 50 iters um render novo — suficiente para acompanhar progresso sem retardar muito. 0 = não atualiza (treinamento em background, máximo speed). Ajustes típicos: em `.quickTest` baixe para 10 (ver cada passo); em runs MCMC longos suba para 500–2000 (overhead acumula).

 EM PALAVRAS SIMPLES

Com que frequência a vista 3D atualiza durante o treino. 50 = cada 50 iters. Mais alto = menos atualizado, ou seja, um pouco mais rápido, mas vê progresso menos. 0 = sem preview (máxima velocidade).

T60 perceptualLossWeight DETALHES**Default:** 0.0 (= desativado) **Range:** 0 ou 0.001 – 0.5**Defined in:** TÉCNICO

Feature futuro V444: peso de termo perceptual via MPSGraph (pequena rede VGG-like). Capturaria similaridade estrutural/textural em nível semântico acima de L1+SSIM — típico em pipelines de pesquisa que valorizam mais „parece real“ que „pixel perfect“. Implementação pendente (code stub existe; forward pass não). Default 0.0. Fica para ativação futura; CLI `--percep-weight F` reservado.

 EM PALAVRAS SIMPLES

Feature planejado que busca „parece natural“ com ajuda de IA em vez de „pixel exato“. Ainda não totalmente implementado.

MCMC Densification (T61–T73)

T61 densificationStrategy

DETALHES

Default: `.classic` (Initializer + Classic presets),
`.mcmc` (todas predefinições MCMC + Scene Class)

Range: `.classic` ou `.mcmc` **Defined in:**

TÉCNICO

Escolhe entre Classic Densification (clone/split/prune, Kerbl et al., 2023) e MCMC Densification (Stochastic Gradient Langevin Dynamics com re-locução, Kheradmand et al., NeurIPS 2024). Em `.classic`, T11–T16 valem; em `.mcmc`, T62–T73. Cuidado ao trocar: defaults Classic e MCMC têm calibração muito diferente — quem flipa o picker no Expert View sem carregar preset adequado arrisca mass extinction tipo bug 1.4.3 (460K → 5 numa iter, porque MCMC OpacityReg 0.01 mata as opacities Classic). Por isso os defaults init de MCMC são „amaciados“ (todos regs em 0.0).

EM PALAVRAS SIMPLES

Qual algoritmo multiplica splats. Classic = método original (rápido, muitos splats). MCMC = mais novo (mais lento, bem menos splats, mas compactos). As predefinições escolhem certo. Só mude se carregar a predefinição (P5–P7 ou P8–P10).

T62 **mcmcMaxGaussians** **DETALHES**

Default: 150 000 (Initializer + `.fullMCMC` + `.mcmcBalanced`), 100 000 (`.mcmcPreview`), 1 500 000 (`.fullMCMCMip` — variante Mip Splatting com budget 10x), 1.19M (`.renderPreset`), 1.25M (`.outdoorPreset`), 670K (`.indoorPreset`) **Range:** 0 (= „usa capacidade do buffer“) ou 10 000 – 5 000 000 **Defined in:**

 **TÉCNICO**

Limite hard da contagem em MCMC. Cresce gradualmente em `T70 mcmcGrowthRate` (típico 5 %) por passo de relocação até o cap. V473/V531 acharam 150K sweet spot — acima de 200K dilui qualidade (muitos pequenos redundantes), abaixo de 100K sub-densifica. Em cenas grandes (p. ex. 1 545-foto drone com 158K SfM init) 150K é baixo demais — daí a extensão 1.4.5 `T72 mcmcCapMultiplier` + `T73 mcmcAutoScaleByScene`. Q7-BayesOpt achou ótimos por cena entre 670K (Indoor) e 1.25M (Outdoor). Em valor 0, a engine usa a capacidade plena do buffer.

 **EM PALAVRAS SIMPLES**

Máximo de splats em MCMC. 150 000 é padrão e basta para a maioria. Outdoor e Render presets (P8, P9) vão a 1+ M para mais detalhe. Aumentar pode trazer detalhe e custar memória; reduzir é freio de emergência.

T63 **mcmcNoiseScale** **DETALHES**

Default: 0.00005 (5e-5 = paper default) **Range:** 1e-6 – 1e-3 **Defined in:**

 **TÉCNICO**

Multiplicador do ruído gaussiano adicionado à posição de cada Gaussian em cada iter MCMC (lógica SGLD). Maior significa mais exploração (Gaussians erram mais, podem achar bons lugares), menor = mais exploitation (ficam onde já estão bons). V467 e V536 confirmaram 5e-5 ótimo — 1e-5/2e-5 pouca exploração, 1e-4 demais (splats se desfazem). Cosine decayed ao longo do treinamento até `T69 mcmcNoiseDecayEnd` — no fim do decay, ruído 0 e os Gaussians convergem.

 **EM PALAVRAS SIMPLES**

Quanto „balanço“ aleatório o app permite aos splats para acharem o melhor lugar. Default é ótimo. Subir deixa os splats inquietos.

T64 `mcmcOpacityRegWeight` **DETALHES**

Default: 0.0 (= desativado nos defaults RadianceKit; paper: 0.01) **Range:** 0 ou 0.001 – 0.05 **Defined in:**

 **TÉCNICO**

Penalidade L1 MCMC-specific sobre opacidade. Paper default 0.01 (empurra Gaussians sem uso para 0, deixa disponíveis para relocação). V464b mostrou: sem reg melhor em RadianceKit (Sessão 28 confirma). Motivo: o critério de pruning em T68 `mcmcDeadOpacityThreshold` basta — penalidade L1 adicional força até Gaussians de baixa opacidade mas valiosos a morrer. Daí default 0. **Atenção:** em 1.4.3-beta o default no initializer era 0.01 por engano, o que causou mass extinction (ver T61). Desde 1.4.4 fixado em 0.0.

 **EM PALAVRAS SIMPLES**

Regularização MCMC-específica. Off, pois o mecanismo MCMC (threshold T68) já basta. Deixe em 0.

T65 `mcmcScaleRegWeight` **DETALHES**

Default: 0.0 (= desativado; paper: 0.01) **Range:** 0 ou 0.001 – 0.05 **Defined in:**

 **TÉCNICO**

Penalidade L1 MCMC-specific sobre autovalores de escala. Paper default 0.01. V464b: sem reg melhor, mesmo motivo que T64. Disabled em todas predefinições MCMC do RadianceKit. Atenção como em T64: bug 1.4.3.

 **EM PALAVRAS SIMPLES**

Como T64, mas para tamanho. Off.

T66 **mcmcRelocationInterval** **DETALHES**

Default: 100 (Initializer + todas predefinições MCMC, padrão paper), 155 (P9 Outdoor — ótimo Q7-BayesOpt) **Range:** 50 – 500 **Defined in:**

 **TÉCNICO**

Intervalo de iters em que MCMC realoca Gaussians mortos ($\text{sigmoid}(\text{opacity}) < T68 \text{ mcmcDeadOpacityThreshold}$). V537 testou 50 (muito disruptivo, loss oscila) e 200 (marginal pior, MCMC perde reatividade). 100 é ótimo. Q7-BayesOpt em Bicycle achou 155 como ótimo scene-specific outdoor — intervalo levemente maior dá ao Adam tempo de integrar Gaussians recém-posicionados antes do próximo reloc evento.

 **EM PALAVRAS SIMPLES**

A cada quantas iters MCMC desloca os splats mortos. 100 padrão. Não precisa mexer — Outdoor preset já tem ótimo.

T67 **mcmcWarmupIterations** **DETALHES**

Default: 500 **Range:** 100 – 5 000 **Defined in:**

 **TÉCNICO**

Iters iniciais sem relocação MCMC. Só após esse warmup começa relocação. Sentido: nas primeiras iters as opacities ainda não estabilizaram — se reloc começasse direto, Gaussians iriam para o lugar errado e teriam que ser deslocados em seguida, destruindo momento Adam. Paper default 500. V464b confirma robustez.

 **EM PALAVRAS SIMPLES**

Quantas iters MCMC tem para „acomodar“ antes de mexer nos splats. 500 padrão, ok.

T68 `mcmcDeadOpacityThreshold` **DETALHES**

Default: 0.005 (Initializer, paper), 0.01 (`.fullMCMC` e todas predefinições MCMC — ótimo V535)

Range: 0.001 – 0.05 **Defined in:**

 **TÉCNICO**

Threshold de sigmoid(opacity) abaixo do qual Gaussian é considerado „morto“ e elegível para relocação. V535 achou 0.01 ótimo (0.005 marginal, 0.02 pior). Maior = reloc mais agressivo; menor = cuidadoso. 0.01 „0.5 % de visibilidade“. P10 Indoor usa via Q7-BayesOpt 0.0142.

 **EM PALAVRAS SIMPLES**

Em que transparência um splat conta como „morto“ para MCMC realocar. 0.01 é ótimo nos testes. Não mexer.

T69 `mcmcNoiseDecayEnd` **DETALHES**

Default: 0 (Initializer = sem decay), 160 000

(`.fullMCMC` = 80 % de 200K), 96 000

(`.mcmcBalanced` = 80 % de 120K), 40 000

(`.mcmcPreview`) **Range:** 0 ou 1 000 – **Defined in:**

 **TÉCNICO**

Iter em que ruído `T63 mcmcNoiseScale` é totalmente amortecido a zero (cosine decay de iter 0 até aqui). V497c/V502 acharam 80 % de maxiterations ótimo — dá ao MCMC tempo de exploração e reserva os últimos 20 % para convergência sem ruído. 0 = ruído constante por todas as iters (raramente útil, MCMC não converge).

 **EM PALAVRAS SIMPLES**

Quando o „balanço“ aleatório dos splats para. Nas predefinições MCMC em 80 % do total — primeiro exploração, depois convergência. Deixe.

T70 **mcmcGrowthRate** **DETALHES**

Default: 0.05 (padrão paper = 5 %) **Range:** 0.01 – 0.2 **Defined in:**

 **TÉCNICO**

Taxa de crescimento do target populacional MCMC por passo de relocação. Lógica: a cada reloc, o target sobe em $(1 + \text{growthRate})$ até $T62 \cdot \text{mcmcMaxGaussians}$ (ou variante escalada via T72/T73). V512/V522 acharam 0.05 ótimo — maiores levam a crescimento rápido demais (Gaussians inseridos antes do Adam integrar), menores deixam cenas sub-densificadas no fim.

 **EM PALAVRAS SIMPLES**

Quão rápido o número de splats em MCMC cresce. 5 % por passo é ótimo. Deixe.

T71 **mcmcSigmoidK** **DETALHES**

Default: 100.0 **Range:** 10.0 – 500.0 **Defined in:**

 **TÉCNICO**

Parâmetro sigmoid sharpness para a atenuação de ruído MCMC. No passo SGLD, o ruído por Gaussian é amortecido por — Gaussians muito opacos (logit positivo) recebem ruído exponencialmente menor que pouco opacos. $K = 100$ é nítido; transição „full noise“ → „no noise“ passa rápido em torno de opacity 0.5. V484–V487 acharam $K = 100$ ótimo — menores (10–50) deixam Gaussians muito opacos tremerem (destrói convergidos); maiores (> 500) fazem transição artificial e Gaussians mortos nem se movem.

 **EM PALAVRAS SIMPLES**

Parâmetro especial: quão nítido MCMC separa „transparente o bastante para realocar“ e „sólido, não mexer“. Padrão ótimo. Não mexer.

T72 **mcmcCapMultiplier** **DETALHES**

Default: 3.0 (Initializer + `.fullMCMC`), 2.0 (`.mcmcPreview`), 2.5 (`.mcmcBalanced`), 2.98 (P8 Render), 5.32 (P9 Outdoor), 1.76 (P10 Indoor)
Range: 0 (= desativado) ou 1.0 – 10.0 **Defined in:**

 **TÉCNICO**

Feature 1.4.5: scaling adaptivo de cap por cena. Quando `T73 mcmcAutoScaleByScene true`, o cap efetivo é (clamped à capacidade do buffer). Contexto: em cenas grandes (drone 1 545 fotos → 158K SfM init), `T62 = 150 000` é baixo — density control nem cresceria. Com multiplier 3.0, cap escala a 474K (158K × 3.0). Q7-BayesOpt achou ótimos: Outdoor com multiplier alto (5.32 → 830K cap em bicycle 156K init), Indoor com 1.76 (paredes saturam mais cedo). Resolução completa do cap em .

 **EM PALAVRAS SIMPLES**

Multiplicador que adapta o cap de splats ao tamanho da cena. Cena grande = mais pontos iniciais = cap maior. Padrão 3× combina; Outdoor vai a 5× (profundidade grande), Indoor a 1.76× (paredes limitam).

T73 **mcmcAutoScaleByScene** **DETALHES**

Default: true (Initializer + todas predefinições MCMC) **Range:** boolean **Defined in:**

 **TÉCNICO**

Feature 1.4.5: master switch da lógica scene-aware de cap (ver T72 +). Em false, só `T62 mcmcMaxGaussians` vale (comportamento 1.4.4). Padrão on para evitar reincidência dos mass extinctions de 1.4.3 em cenas grandes. Desligar manualmente só se quiser cap rígido — p. ex. treinar variante 150K com tamanho final previsível.

 **EM PALAVRAS SIMPLES**

Liga a adaptação automática do cap ao tamanho da cena. Padrão on. Só desligue se quiser número específico de splats.

Mip-Splatting (Q1.5) (T74–T76)

Status: Q1.5 foi declarado „closed no-win“ em 2026-05-25 após 14 iterações autônomas + overnight 1.5M confidence check (máx $\Delta@2\times = +0.27$ dB; gate original exigia $\geq +1.5$ dB média sobre $0.5\times/2\times$; FAIL 0/11 pair scenes). Os campos ficam **opt-in** para experimentos; todas as predefinições de produção têm. Verdict: docs/plans/2026-05-25-phase-q1.5-final-verdict.md.

T74 useMipSplatting **DETALHES**

Default: false (todas predefinições de produção), true (`.fullMCMCMip` — sibling research) **Range:** boolean **Defined in:**

 **TÉCNICO**

Ativa Mip Splatting (Yu et al., CVPR 2024): filtro de smoothing 3D + filtro 2D + compensação α que limita a frequência por Gaussian ao Nyquist da sampling rate mais densa das câmeras. Objetivo teórico: eliminar aliasing em renderings fora da resolução de treino (0.5x ou 2x). Funcionalidade verificada em Q1.5-D-Test. Mas: gate original ($\Delta@1x \geq +0.3$ dB E $\text{avg}(\Delta@0.5x, \Delta@2x) \geq +1.5$ dB) não atingido em nenhuma das 11 pair scenes. Máximo observado: family 750K classic $\Delta@2x = +0.270$ dB. Cenas externas (Truck, Flowers) ainda PIORARAM em 1x e 0.5x. Hipótese: smoothing 3D briga com relocação MCMC em high-Gs. Campo fica para futura re-eval multi-scale com methodology Mip- NeRF 360 correta (ver O3-backlog no caminho de benchmark).

 **EM PALAVRAS SIMPLES**

Filtro de aliasing de paper de 2024. Teoricamente legal, mas nos testes não trouxe nada e às vezes piorou. Fica disponível para quem quer experimentar; não recomendamos. Deixe off.

T75 mipSmoothing3DScale **DETALHES**

Default: 0.2 (paper default) **Range:** 0.05 – 1.0 **Defined in:**

 **TÉCNICO**

Parâmetro de escala de smoothing 3D (Yu et al. §3.3, paper default 0.2). Maior = mais suavização em world space por Gaussian (= mais anti-aliasing, mais blur na escala default); menor = mais nítido mas suscetível a aliasing. Só consultado se T74 `useMipSplatting = true`. Em Q1.5 não foi otimizado — o A/B gate já perdeu no paper default 0.2; mais sweeps seriam inúteis.

 **EM PALAVRAS SIMPLES**

. Se Mip off, irrelevante.

T76 mipFilter2DVariance DETALHES

Default: 0.3 (= exatamente o comportamento V242 legado) **Range:** 0.1 – 1.0 **Defined in:**

 TÉCNICO

Variância 2D Mip somada à diagonal de Σ_{2D} (variância direta, não quadrada). 0.3 é o valor legado V242 hard-coded no kernel antes de Mip Splatting. Quando T74 useMipSplatting = false, o kernel ignora esse valor por completo e escreve 0.3 hard-coded — garantia Codex Round 1 S3-1 de que a baseline não regride. Quando, o valor setado é usado. Fica no catálogo para sweeps Mip.

 EM PALAVRAS SIMPLES

Mais um parâmetro Mip Splatting. Se Mip off, irrelevante.

Adaptive Densification (Q5) (T77–T79)**T77** adaptiveDensification DETALHES

Default: false **Range:** boolean **Defined in:**

 TÉCNICO

Feature Q5: rolling median tracker como alternativa ao fixo T11 densifyGradThreshold. Em true, cada densify step sobrescreve o threshold com `median(últimos N avgGrad samples) × T79 adaptiveDensifyMultiplier`. `N = T78 adaptiveWindow`. Mais conservador que o p98 do V440 (cuja armadilha de 63K pruning foi catastrófica); `median + 2x` fica em p70–p80 da distribuição em steady state. Q5 testes: sozinho FAIL 0/3 cenas; combinado com Q6 (ver T80/T81) PASS 1/3 cenas — bundle Q5+Q6 passado em 2026-05-25 como opt-in via CLI `--adaptive-densify`. Q6 é „carrier“ do ganho; Q5 contribui mais para estabilidade.

 EM PALAVRAS SIMPLES

Threshold de densify autoajustável. Em vez de sensibilidade fixa, o app se adapta à cena. Sozinho não melhora, junto com Q6 sim. Ligue os dois ou nenhum.

T78 adaptiveWindow DETALHES**Default:** 1 000 **Range:** 100 – 10 000 **Defined in:** TÉCNICO

Window de rolling median em eventos de densificação (NÃO iters — cada T13 `densifyInterval` step dá um sample). Default 1 000 — com isso, as últimas 100 000 iters contribuem, ou seja toda a história até aqui. Fase inicial (antes de T78 samples): tracker retorna nil → fallback no fixo T11 . Só relevante se.

 EM PALAVRAS SIMPLES

Quanto densify steps antigos entram na mediana para T77. Padrão 1000 ok. Só relevante com Q5 adaptive on.

T79 adaptiveDensifyMultiplier DETALHES**Default:** 2.0 **Range:** 1.0 – 4.0 **Defined in:** TÉCNICO

Multiplicador sobre rolling median para threshold adaptive. Default 2.0 p70–p80 da distribuição típica. Menor = mais agressivo (mais clones); maior = mais estrito. Q5 testou em 1.5–3.0; 2.0 é melhor default. Só relevante se.

 EM PALAVRAS SIMPLES

Fator para T77/T78. Padrão 2.0 = mais estrito que a mediana. Não mexer.

Curriculum (Q6) (T80–T81)

T80 curriculumResolutionRamp

DETALHES

Default: false **Range:** boolean **Defined in:**

TÉCNICO

Feature Q6: resolução de treino começa em 0.5× e troca em $T50 \text{ positionLRScheduleEndIteration} / 2$ (ou $T1 \text{ maxIterations} / 2$, se T50 não setado) para T22 `trainingRenderScale`. Usa a infra `resize/restoreImageBuffers` desenvolvida em Q1.5.1. Sobrescreve T23 `resolutionWarmupScale` se ativo. Q6 é „carrier“ do ganho no bundle Q5+Q6 (ver T77) — aumento gradual de resolução dá tempo de achar geometria grossa em resolução menor antes do detalhe fino. Via CLI: `--curriculum-resolution`.

EM PALAVRAS SIMPLES

„Primeiro grosso, depois fino“ para resolução. Metade na primeira metade do treinamento, full na segunda. Ajuda em algumas cenas, em outras não — melhor ligar com T81.

T81 curriculumSHProgression

DETALHES

Default: false **Range:** boolean **Defined in:**

TÉCNICO

Feature Q6: sobrescreve T21 `shDegreeUpgradeIterations` com $[\text{maxIter}/4, \text{maxIter}/2, \text{maxIter} * 3/4]$, distribuindo upgrades SH uniformemente em vez de front-loaded. Hipótese: geometria estável antes de explosão de detalhe de cor posiciona efeitos dependentes de view com mais precisão. Q5+Q6 juntos PASS 1/3 cenas; Q6 é carrier. CLI: `--curriculum-sh`.

EM PALAVRAS SIMPLES

„Forma primeiro, cor depois“ — efeitos de brilho só liberados tarde no treinamento, para os splats acharem posição e tamanho antes. Liga com T80; sozinho rende menos.

Predefinições estáticas (TP1–TP9)

Aqui só as diferenças estruturais ao default do Initializer. A descrição completa de marketing das onze predefinições UI P1–P11 está no Capítulo 7.

TP1 `.preview`

DETALHES

Diagnose/preview para sistemas ≥ 10 GB RAM.

Overrides:

- 30 000 \rightarrow 5 000 - 15 000 \rightarrow 3 500 (70 % de maxIter) - 1.6e-6 \rightarrow 1.6e-5

(10x maior, decay menos agressivo) -,,,, cada 2x (V176) - 3 000 \rightarrow 100 000 (efetivamente off, V172: reset destrói treinos curtos) - [1K, 2K, 3K] \rightarrow [1K, 2K] (V182: degree 3 não converge em 2K iters) - 1.0 \rightarrow 0.5

EM PALAVRAS SIMPLES

qualquer avaliação inicial de uma série recém-importada — 2–3 min de espera, depois basta para a pergunta binária „vale quality run?“.

TP2 `.full`

DETALHES

Production-quality classic. Overrides: - 30 000 \rightarrow 35 000 (V550: testes 40K Truck overtraining +10.7 % Gs em -1.3 % L1) - 15 000 \rightarrow 5 000 (sweet spot V310, V338 7K worse) - todas LRs 2x (V188) - 1.6e-6 \rightarrow 1.6e-5 (V45 10x) - 2e-6 \rightarrow 1.1e-6 (V335) - 100 \rightarrow 200 (V112) - 0.005 \rightarrow 0.001 (V393) - 3 000 \rightarrow 100 000 (V194 disabled, V421 confirmado) - [1K, 2K, 3K] \rightarrow [2K, 5K, 8K] (V228 delayed) - 0.0 \rightarrow 0.9995 (V546 HTGS, 14 % de melhoria) - 50 (inalterado, V546) - false \rightarrow true (V438) - 0 \rightarrow 20 000 (V431) - true (V443, já initializer default para `.full`)

EM PALAVRAS SIMPLES

qualquer foto-captura padrão (objeto, sala pequena, escultura) com < 500 imagens. A 14 % melhora vs V438 anunciada em V546 foi confirmada em avg 3 trials Horse Full.

TP3 `.fullClassicPaper`

DETALHES

Sibling Q1.5-A de TP2, paper-faithful Classic. Overrides vs TP2: - 35 000 \rightarrow 30 000 (paper) - 5 000 \rightarrow 15 000 (paper: 50 % de maxIter) - 1.6e-5 \rightarrow 1.6e-6 (paper) -,,, paper defaults (0.05, 0.005, 0.001) - 1.1e-6 \rightarrow 2e-7 (calibrado para 1-2M Gs em Bicycle) - 200 \rightarrow 100 (paper) - 0.001 \rightarrow 0.005 (paper) - 100 000 \rightarrow 3 000 (paper §5.2, risco — pode disparar V194 regression) - 0.9995 \rightarrow 0.0 (paper sem decay) - 20 000 \rightarrow 30 000 (cosine roda em 100 % de maxIter)

EM PALAVRAS SIMPLES

experimentos Q1.5 que precisam de budgets paper (1–2M Gaussians) para testes Mip Splatting. Após veredito Q1.5 „closed no-win“, a predefinição fica para advanced, não recomendada em produção.

TP4 `.fullMCMC` **DETALHES**

Production-quality MCMC. Overrides vs Initializer:

- 30 000 → 200 000 (V534, MCMC precisa de 5x mais iter que Classic)
- 15 000 → 160 000 (V504b 80 % de maxIter) - 1.6e-6 → 1.6e-5 -

LR schedule como TP2 (todas 2x) - 0.2 → 0.05 (V521b/V534: MCMC precisa de sinal L1 mais forte)

- [1K, 2K, 3K] → [2K, 5K, 8K]
- `.classic` → `.mcmc` - 150 000 (já no initializer, confirmado no

preset) - 5e-5 (V467/V536 ótimo) - 0.005 → 0.01 (V535 ótimo) - 0 → 160 000 (80 % de maxIter, V497c/V502) - 3.0 (já no initializer) - true (já no initializer) - 3 000 → 200 000 (efetivamente off, MCMC usa reloc em vez de reset)

 **EM PALAVRAS SIMPLES**

entrega web, object captures com exigência de detalhe, drones (mesmo assim P9 Outdoor ainda melhor). 71 % menos Gaussians que Classic com L1 comparável.

TP5 `.fullMCMCMip` **DETALHES**

Sibling Q1.5-D de TP4, com Mip Splatting + budget MCMC paper. Overrides vs TP4:

- `mcmcMaxGaussians` 150 000 → 1 500 000 (10x, paper magnitude)
- `useMipSplatting` false → true (Mip on)

 **EM PALAVRAS SIMPLES**

todos os outros campos idênticos ao TP4. Q1.5 D-PASS em Bicycle 2026-05-24 (quebra a streak de 12-iter multi-scale FAIL). Veredito final Q1.5 2026-05-25 ainda closed-no-win — ganho Mip Splatting não reproduzível sobre 11 pair scenes. Preset fica opt-in.

TP6 `.classicBalanced` **DETALHES**

Mid-tier Classic. Overrides vs TP2: - 35 000 → 20 000 (V149: 20K = 30K em 33 % menos tempo) - 20 000 → 0 (cosine roda em maxIter = 20K, sem extended phase)

 **EM PALAVRAS SIMPLES**

casos padrão com espera mais curta. V149 identificou como sweet spot.

TP7 `.mcmcPreview`**DETALHES**

Diagnose MCMC. Overrides vs TP4: - 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 = scaling mais leve)

EM PALAVRAS SIMPLES

ver rápido um resultado MCMC para julgar se TP4 ou Scene Class valem.

TP8 `.mcmcBalanced`**DETALHES**

Mid-tier MCMC. Overrides vs TP4: - 200 000 → 120 000 (V518)

- 160 000 → 96 000 (80 %) - 160 000 → 96 000 (80 %) - 3.0 → 2.5

(entre Preview 2.0 e Full 3.0)

EM PALAVRAS SIMPLES

MCMC sem o run de 200K completo. 120K iters é bom compromisso entre qualidade e espera.

TP9 `.quickTest`**DETALHES**

Teste de funcionamento puro. Overrides vs Inicializar:

- 30 000 → 1 000 - 15 000 → 500 - 2e-6 → 4e-6 (calibrado para 0.25x)
- 100 → 50 - 3 000 → 100 000 (off, curto demais) - 1.0 → 0.25

EM PALAVRAS SIMPLES

sanity check „o treinamento começa direito?“. Duração < 30 s em M3 Ultra. Sai garantidamente embolado.

Método:

Assinatura: `public func resolveMcmcMaxGaussians(initialPointCount: Int, bufferCapacity: Int) -> Int` **Defined in:**

TÉCNICO Única source-of-truth para „quantos Gaussians MCMC pode crescer ao máximo?“. Calcula a partir de três entradas: o configurado `T62 mcmcMaxGaussians` (com mass-extinction floor 150 000 se 0), o (número de pontos SfM iniciais) e o (tamanho pré-alocado do buffer). Lógica:

1. `base = T62 > 0 ? T62: 150_000` (o floor protege contra bugs de initializer default como o incidente 1.4.3)
2. Se `T73 mcmcAutoScaleByScene && initialPointCount > 0 && T72 mcmcCapMultiplier > 0`:
 - `scaled = max(base, ceil(initialPointCount × T72));` **senão**

3. Se `bufferCapacity > 0` : `return min(scaled, bufferCapacity)`

4. Senão `return scaled`

Exemplo: Bicycle (Mip-NeRF 360, 194 frames) → SfM init 156K pontos, `T62 = 150 000`, `T72 = 5.32`, capacidade do buffer 8M. Cap resolvido = $\min(8M, \max(150K, \text{ceil}(156K \times 5.32))) = \min(8M, 830K) = 830K$. Esse é o cap de crescimento efetivo a que a relocação MCMC se atém.

EM PALAVRAS SIMPLES Calcula o número máximo real de splats em MCMC. Pega seu setting, olha quantos pontos sua cena tem no início, e escala com o `Multiplier` se o ajuste automático estiver on. Assim o cap se adapta à cena, em vez de forçar o mesmo valor para uma cena pequena e uma gigante. Você não chama o método diretamente — o treinamento usa internamente.

Qual campo para quê? (cheat sheet)

Objetivo	Campos para girar
Mais detalhe à distância	<code>T62 mcmcMaxGaussians</code> alto, <code>T72 mcmcCapMultiplier</code> 5+
Mais detalhe em geral (Classic)	<code>T1 maxIterations</code> alto ($\leq 40K$), <code>T2 densifyUntilIteration</code> $\leq 14\%$ de <code>T1</code>
Reduzir floaters em drones	<code>T43 frustumCullEnabled</code> on, <code>T20 skyMaskingEnabled</code> on, <code>T45 skyDomeEnabled</code> on
Belo céu em cenas externas	<code>T45 skyDomeEnabled</code> on, <code>T47 skyDomeRadiusMultiplier</code> 30–60
Arquivo de export menor	Estratégia <code>.mcmc</code> (<code>T61</code>), <code>T56 postTrainingCompactification</code> on, <code>T62 mcmcMaxGaussians</code> $\leq 200K$
Treinamento mais rápido	<code>T22 trainingRenderScale</code> 0.5, <code>T1 maxIterations</code> pela metade — mas não ambos!
Melhores high-lights	<code>T21 shDegreeUpgradeIterations</code> com [2K, 5K, 8K] (sem front-load), MCMC + 200K iter
Manter Mac responsivo	<code>T25 throttleDelayMs</code> 5–10 (custa 15% de treinamento)
Live preview mais frequente	<code>T59 livePreviewInterval</code> baixe para 10–20
Transições mais suaves em sombras	<code>T17 ssimWeight</code> um pouco alto (0.15–0.25), mas não acima de 0.3
Internos compactos	<code>P10 Indoor</code> (, <code>T72 = 1.76</code>)

Campos perigosos

Estes campos podem, mal configurados, levar a OOM, crash, mass extinction de Gaussians ou benchmarks inúteis. Tratar com cuidado:

- T11 `densifyGradThreshold` — metade pode gerar 2–4x mais Gaussians e estourar memória de GPU. Também: deve combinar com T22 `trainingRenderScale` (1.0x → 1e-6, 0.5x → 2e-6, 0.25x → 4e-6).
- T72 `mcmcCapMultiplier` — em cenas grandes com > 200K pontos iniciais SfM e multiplier > 5, o cap resolvido sobe a milhões de Gaussians. Em Macs com 36 GB, possível OOM. Outdoor 5.32 funciona porque Mip-NeRF 360 Bicycle tem 156K init → 830K.
- T39 `testViewIndices` — setar manualmente pode invalidar benchmark (indices > N → sem holdouts). Deixe o `--benchmark` setar.
- T64 `mcmcOpacityRegWeight` e T65 `mcmcScaleRegWeight` — em 1.4.3-beta setados a 0.01 causaram mass extinction (460K → 5 Gaussians). Desde 1.4.4 fixados em 0.0, mas aumentar manualmente reproduz o problema.
- T15 `opacityResetInterval` — se não em 100 000+ (efetivamente off) e treinamento < 10 000 iters, o reset destrói convergência. `.preview` por isso 100 000 mesmo com `maxIterations = 5 000`.
- T54/T55 `densifyPhase2*` — two-phase densification caiu em cascade de 0 Gaussians em testes. Mantenha em 0.
- T74 `useMipSplatting` — Q1.5 closed no-win 2026-05-25, pode até PIORAR PSNR em outdoor. Default off; opt-in só para pesquisa.

Se um campo nesta lista te interessa: backup do preset atual (export JSON) antes e pense em como medir o resultado de forma reproduzível — senão não saberá se melhorou ou piorou.

CAPÍTULO

Capítulo 7 — Predefinições de qualidade embutidas

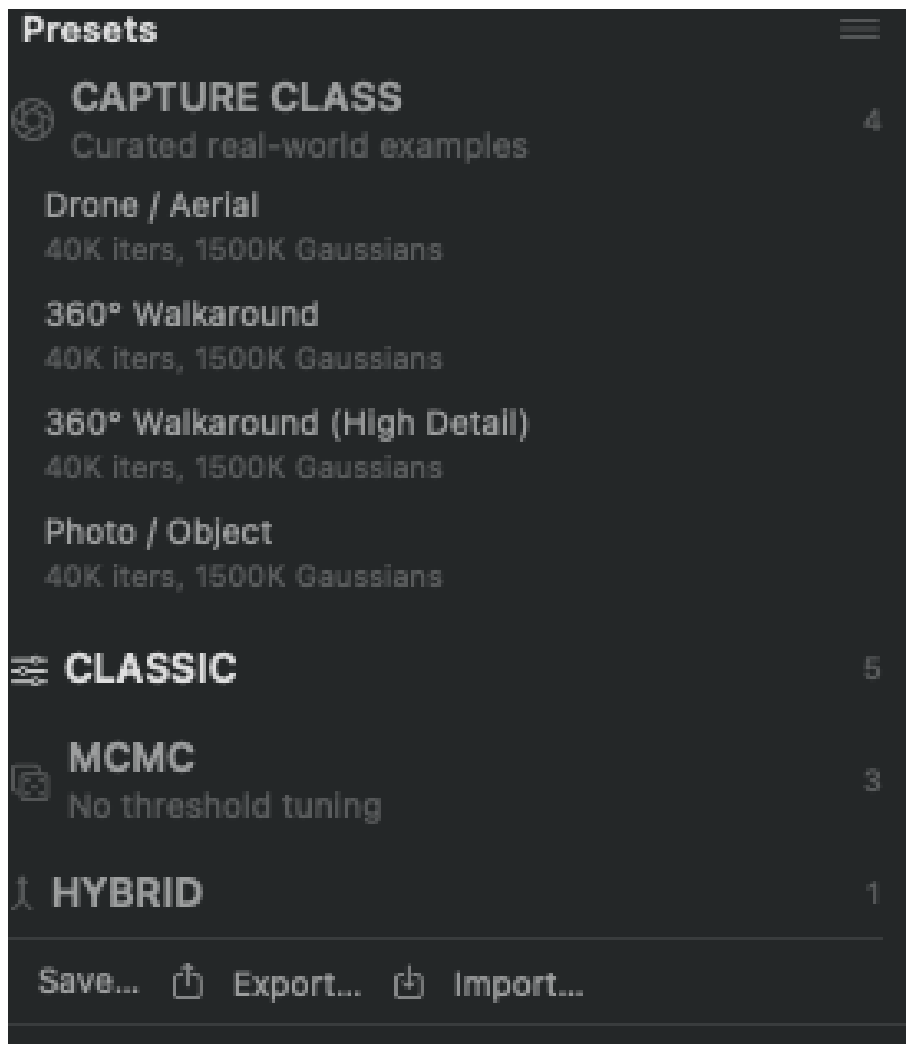


Figura 27: Seção de predefinições com os quatro grupos expandidos — CAPTURE CLASS (4 presets: Drone/Aerial, 360° Walkaround, 360° Walkaround (High Detail), Photo/Object), CLASSIC (5 presets: Quick/Preview/Balanced/Quality/Ultra Detail), MCMC (3 presets, aviso „No threshold tuning“), HYBRID (1 preset: Balanced (Hybrid))

O QUE APARECE NA IMAGEM Seção de predefinições no Inspetor, todos os quatro grupos expandidos. CAPTURE CLASS com as quatro predefinições reais curadas (Drone / Aerial, 360° Walkaround, 360° Walkaround (High Detail), Photo / Object) — é o grupo primário e, no Modo Iniciante, o único visível. CLASSIC com Quick (1K iters), Preview

(5K iters, seleção ativa com check azul), Balanced (20K iters), Quality (35K iters) e Ultra Detail (35K iters). MCMC com subtítulo „No threshold tuning“ — MCMC dispensa threshold de densify-until: Preview (60K iters, 100K Gaussians), Balanced (120K, 150K), Quality (200K, 150K). HYBRID com o Balanced (Hybrid) (20K iters, 150K Gaussians). Linha de ações no rodapé: Save..., Export..., Import...

Uma predefinição (preset) é uma configuração pronta para o treinamento. O RadianceKit traz treze predefinições embutidas em quatro grupos: quatro predefinições **Capture-Class** (P9–P12) — receitas curadas, validadas a olho em material real da comunidade para tipos de captura reais (drone, walkaround 360°, foto-objeto) e o eixo primário desde a v1.6 —, cinco predefinições Classic (P1–P5: Quick/Preview/Balanced/Quality/Ultra Detail), três predefinições MCMC (P6–P8) e uma predefinição Híbrida (P13) que combina as estratégias Classic e MCMC. As antigas predefinições „Scene Class“ (Render/3D, Outdoor, Indoor, ajustadas na Fase Q7 academicamente contra cenas Mip-NeRF 360 e NeRF-Blender) foram recolhidas na v1.6 como grupo visível — a Capture Class validada a olho em material real é agora o eixo primário; as configurações ajustadas na Q7 permanecem apenas internamente. Você escolhe as predefinições na sidebar na área **Presets** ou no Modo Iniciante durante o import. Os botões **+** abrem diálogos para criar suas próprias predefinições ao lado — as treze embutidas não podem ser apagadas, mas podem ser duplicadas.

No Expert View as predefinições aparecem agrupadas por tipo de captura e estratégia (Capture Class / Classic / MCMC / Híbrido). Um clique numa entrada grava a configuração de treinamento armazenada no estado atual. Isso não é um snapshot — se você depois mexer nos controles, o estado muda, mas a predefinição em si permanece inalterada; um aviso colorido então mostra „modified“.

Qual predefinição é a certa em cada momento depende sobretudo do tipo de cena e do hardware. As três tabelas no fim do capítulo resumem isso.

I P1 — Quick



ONDE

Inspetor → Seção de predefinições → Grupo „Classic“ → entrada „Quick“. Sufixo de UUID ...001 .



TÉCNICO

Predefinição de diagnóstico com 1 000 iterações, estratégia clássica (adaptativa) de densification e uma escala de resolução de treinamento de 0.25x (imagem de entrada é reduzida a 25 % antes do treinamento). Não pretende entregar uma cena final, mas rapidamente verificar se o setup (poses de câmera, nuvem de pontos, série de imagens) mostra movimento útil nos valores de loss. Em um M3 Ultra tipicamente menos de 30 segundos em 50–200 imagens. A resolução pequena disfarça a qualidade real de imagem, mas mantém o consumo de memória e o custo de render bem baixos. Também é escolhida automaticamente como padrão no primeiro início se o sistema tiver menos de 10 GB de RAM.

EM PALAVRAS SIMPLES

Teste rápido de funcionamento. Joga as imagens, espera meio minuto, vê se o esboço da cena aparece. Se a imagem no visualizador parecer um borrão amassado — ok, é assim mesmo. Mas se você só ver pontos escuros ou uma forma totalmente distorcida, provavelmente as poses de câmera estão erradas (ver Capítulo 9). Para um resultado apresentável você precisa depois de pelo menos P2 ou P3.

I P2 — Preview (Classic)



ONDE

Inspetor → Seção de predefinições → Grupo „Classic“ → entrada „Preview“. Sufixo de UUID ...002 .



TÉCNICO

5 000 iterações de densification clássica, escala de resolução 0.5x, taxas de aprendizado duplicadas em relação ao padrão. A densification (clone + split) fica ativa nas primeiras 2 500 iterações, depois somente pruning. Predefinição padrão para sistemas com ≥ 10 GB de RAM. Em um M3 Ultra tipicamente 90 segundos a 3 minutos para uma cena de 200 imagens. Entrega uma impressão útil da geometria e da pose da câmera, mas as texturas ficam visivelmente suavizadas — a resolução de 0.5x não pode ser contornada diretamente treinando depois com P3 ou P4, porque as taxas de aprendizado estão calibradas para meia resolução.

EM PALAVRAS SIMPLES

O padrão para „dar uma olhada rápida“. Quando você acabou de importar imagens e quer ver se a cena é minimamente reconstrutível, é o nível certo. Cerca de 2–3 minutos de espera, e depois você pode girar no visualizador 3D e julgar se vale a pena investir em mais treinamentos. Só quando o preview já parece bom é que compensa Balanced ou Quality.

| P3 — Balanced (Classic)



ONDE

Inspetor → Seção de predefinições → Grupo „Classic“ → entrada „Balanced“. Sufixo de UUID `...005` .



TÉCNICO

20 000 iterações de densification clássica em resolução de imagem total. A densification roda nas primeiras 15 000 iterações; a partir da iter 3 000 com intervalo de densify de 100. Empiricamente o „sweet spot“ das sessões de treinamento documentadas: com densification clássica em Horse Full e Truck o L1-loss estabiliza entre as iters 18 000 e 22 000; treinar mais que isso, abaixo de Quality (P4), não traz melhora significativa. Em um M3 Ultra tipicamente 30–60 segundos em 200 imagens, 5–8 minutos em 1 000+ imagens.

EM PALAVRAS SIMPLES

O „bom compromisso“. A maioria das cenas já fica bonita assim, sem você ter de esperar uma hora. Se o objetivo é mostrar o resultado em algum lugar (rede social, site, demo para cliente), em geral basta. Só quando você quer dar zoom no modelo splat ou precisa de detalhes da textura de superfície é que vale o salto para P4 Quality ou P7 MCMC.

| P4 — Quality (Classic)



ONDE

Inspetor → Seção de predefinições → Grupo „Classic“ → entrada „Quality“. Sufixo de UUID `...003` .



TÉCNICO

35 000 iterações de densification clássica com V546-„Opacity Decay“ (HTGS, Eurographics 2025): após cada ciclo de densify, a opacidade de todos os Gaussians existentes é multiplicada por um fator < 1.0 , o que remove com confiabilidade Gaussians que ficaram inativos durante o pruning, alcançando assim um L1-loss 14 % melhor que a corrida clássica de 35 000 iters com o mesmo número de iterações. SSIM-loss está ativo (`ssimWeight=0.05`). Em um M3 Ultra tipicamente 2–4 minutos em 200 imagens. Em NeRF-Blender (Lego, Chair, Drums) entrega L1 final ≈ 0.023 — melhor variante Classic nos 560+ experimentos documentados. Atenção: precisa de ~3–5 GB de memória de GPU; em sistemas com 8 GB, P3 é a escolha segura.

EM PALAVRAS SIMPLES

A melhor variante clássica. Entrega textura nítida e geometria fina, sobretudo em capturas de objeto (uma escultura, uma cadeira, um vaso). Em cenas externas grandes ou ambientes, em compensação, você quase não percebe diferença para Balanced — ali vale mais a mudança para uma predefinição MCMC (P6–P8) ou para uma Capture Class (P9–P12) do que o salto de P3 para P4. Quem quer o máximo absoluto da família Classic usa P5 Ultra Detail.

I P5 — Ultra Detail (Classic)

ONDE

Inspetor → Seção de predefinições → Grupo „Classic“ → entrada „Ultra Detail“. Sufixo de UUID `...008` .

TÉCNICO

Cerca de 35 000 iterações de densification clássica — o vencedor da corrida held-out da Quality-Matrix (2026-06-10). Nas três cenas Mip-NeRF 360 testadas, Ultra Detail bate a predefinição embutida MCMC „Quality“ (P8) com tempo de wall-clock comparável por cerca de +0.94 dB de PSNR. É assim a predefinição de qualidade mais forte do grupo Classic e a variante Classic mais nítida que o RadianceKit entrega. Em um M3 Ultra tipicamente no mesmo intervalo de tempo que P4 Quality (2–5 minutos em 200 imagens), mas precisa de um pouco mais de memória de GPU; em sistemas com 8 GB, P3 continua sendo a escolha segura.

EM PALAVRAS SIMPLES

O nível Classic mais nítido e o vencedor held-out dos nossos testes de qualidade: em cenas reais, cerca de um decibel melhor que a variante MCMC „Quality“ — com tempo de espera parecido. Se você quer máxima fidelidade de detalhe com a densification clássica consagrada e tem memória de GPU suficiente, é a primeira escolha. Se a memória não for suficiente ou você precisar de um arquivo de export o menor possível, fique com P4 Quality ou uma predefinição MCMC.

I P6 — Preview (MCMC)

ONDE

Inspetor → Seção de predefinições → Grupo „MCMC“ → entrada „Preview“. Sufixo de UUID `...006` .

TÉCNICO

60 000 iterações de densification MCMC (3DGS-MCMC, NeurIPS 2024) com cap de 100 000 Gaussians. MCMC substitui a lógica heurística de clone/split por relocação Markov-chain Monte-Carlo: Gaussians mortos são reposicionados por profundidades de amostragem ponderadas por sigmoide, o que produz uma quantidade controlada e reproduzível de Gaussians. O cap limita o número máximo de forma dura em 100K — economiza memória e tempo de render, mas custa detalhe. Em um M3 Ultra tipicamente 5–8 minutos em 200 imagens. Adequada como „teste de funcionamento MCMC“ — ajuda a julgar se trocar de Classic para MCMC faz sentido, antes de você investir mais tempo em P7 ou P8.

EM PALAVRAS SIMPLES

Como P2 Preview, mas com o método MCMC mais novo. Costuma entregar nuvens splat mais compactas e mais uniformes que a variante clássica. Para uma primeira inspeção de uma cena, os 5–8 minutos geralmente bastam. Se você gostar do preview, o próximo passo é P7 (Balanced) ou direto P8 (Quality MCMC).

| P7 — Balanced (MCMC)

ONDE

Inspetor → Seção de predefinições → Grupo „MCMC“ → entrada „Balanced“. Sufixo de UUID `...007`.

TÉCNICO

120 000 iterações MCMC com cap de 150 000 Gaussians. O nível intermediário do MCMC — quase o número final de Gaussians de P8 Quality, mas só 60 % das iterações. Empiricamente o L1-loss nas sessões de treinamento documentadas fica em 0.026–0.028 em Horse Full, contra 0.0246 do P8 — ou seja, cerca de 7 % mais alto, com metade do tempo de espera. Em um M3 Ultra tipicamente 8–15 minutos em 200 imagens. Usa um método que escala o cap efetivo de Gaussians de acordo com a densidade de pontos da nuvem SfM de entrada (ver T75 no Capítulo 6).

EM PALAVRAS SIMPLES

MCMC com profundidade de detalhe decente, mas sem a longa corrida completa do P8. Para a maioria das cenas basta, sobretudo se você quer espremer um MCMC dentro do tempo da pausa de almoço. Se a memória estiver curta (p. ex. em chips M com apenas 16 GB), fique aqui — P8 precisa de mais memória de GPU.

| P8 — Quality (MCMC)

ONDE

Inspetor → Seção de predefinições → Grupo „MCMC“ → entrada „Quality“. Sufixo de UUID `...004`.

TÉCNICO

200 000 iterações MCMC com cap de 150 000 Gaussians, SSIM-loss 0.05, decaimento de noise MCMC ao longo de 80 % das iterações. Melhor L1 single-run nos 560+-experimentos: 0.0238 em Horse Full; média de 3 trials 0.0246 (contra 0.0230 do P4 na mesma cena). O MCMC entrega ainda 71 % menos Gaussians (150K vs ~524K) — decisivo se você quer publicar o resultado na web, porque a nuvem menor produz arquivos de export bem menores. Tempo de treinamento em um M3 Ultra tipicamente 20–35 minutos em 200 imagens; em sets de 1 000+ imagens, mais perto de 1–2 horas. Melhor escolha quando se busca máxima qualidade de imagem com tamanho final mínimo.

EM PALAVRAS SIMPLES

A melhor variante MCMC. Entrega nuvens splat muito limpas e compactas — ideal se você quer depois embutir o resultado num visualizador 3D para web ou enviar como arquivo (o arquivo é menor que o do P4 Quality com qualidade visual equivalente). Mas para isso você precisa de paciência — em capturas grandes, mais de uma hora de espera. Planeje mais como uma corrida „durante a noite“.

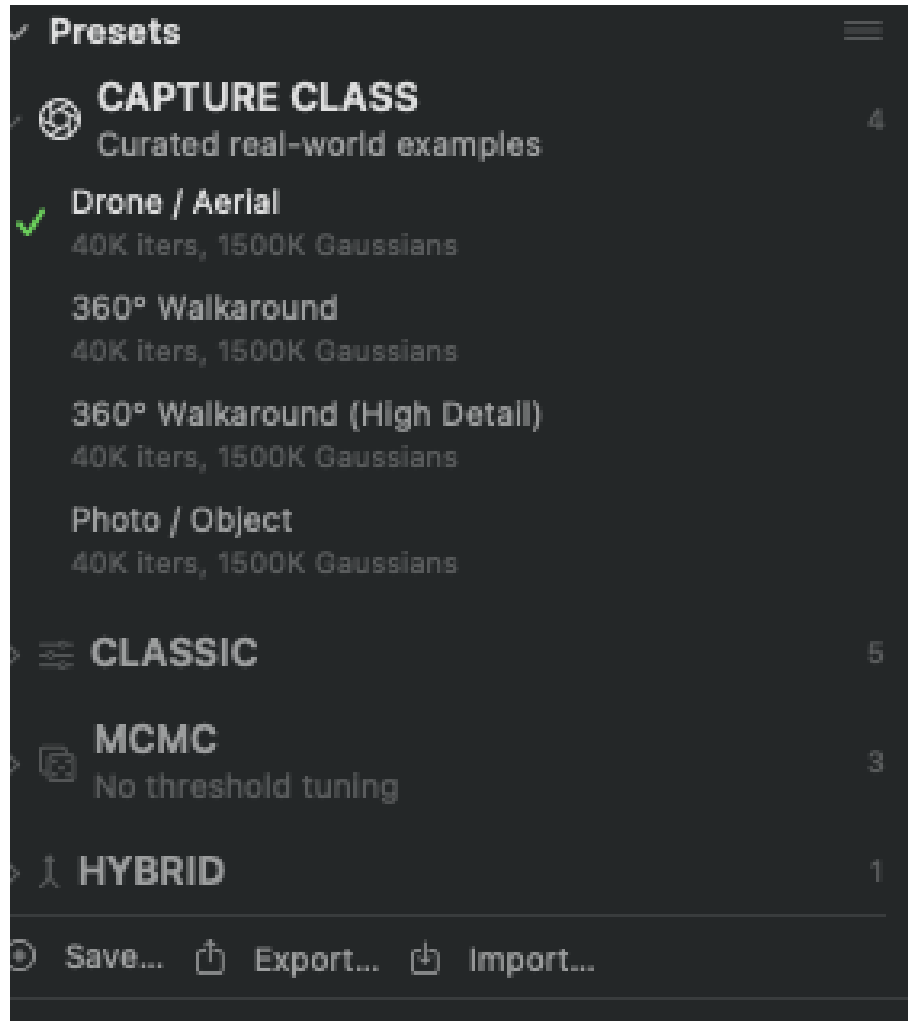


Figura 28: Grupo CAPTURE CLASS expandido com as quatro predefinições reais curadas — Drone / Aerial (MCMC, 40K iters, Cap 1,5 M), 360° Walkaround (MCMC, 40K, Cap 1,5 M), 360° Walkaround (High Detail) (Híbrido, 40K, Cap 1,5 M, opt-in) e Photo / Object (Híbrido, 40K, Cap 1,5 M). Este grupo fica bem no topo e, no Modo Iniciante, é o único visível.

O QUE APARECE NA IMAGEM Inspetor com o grupo CAPTURE CLASS expandido — o grupo primário de predefinições desde a v1.6, no Modo Iniciante o único exibido. Cada entrada é uma receita validada a olho em material real da comunidade para um tipo concreto de captura (drone, walkaround 360°, foto-objeto), não um valor otimizado contra um conjunto de teste acadêmico. A seleção por clique grava a configuração de treinamento armazenada no estado atual.

| P9 — Drone / Aerial

ONDE

Inspetor → Seção de predefinições → Grupo „Capture Class“ → entrada „Drone / Aerial“. Sufixo de UUID ...010 .

TÉCNICO

Predefinição Capture Class para órbitas aéreas e de drone de edifícios e paisagens. Densifier MCMC, 40 000 iterações, cap de 1,5 milhão de Gaussians, SSIM-loss 0.5 mais termo edge-aware 0.1. O decisivo é a penalidade de anisotropia com peso 0.003 num limiar de razão de 6 — o „matador de espagete“ contra os artefatos tipicamente em forma de agulha que o footage de drone produz. Validado num voo de drone DJI 4K real sobre o viaduto de Pensford (verificado a olho, não apenas por métrica).

EM PALAVRAS SIMPLES

Para tomadas do alto — voos de drone ao redor de um edifício, sobre uma paisagem, ao longo de uma fachada. A forte penalidade de anisotropia limpa os artefatos em forma de agulha ou espagete que o material de drone gosta de produzir. Se o seu material foi capturado do chão, encaixa mais Photo / Object ou uma predefinição Classic.

| P10 — 360° Walkaround

ONDE

Inspetor → Seção de predefinições → Grupo „Capture Class“ → entrada „360° Walkaround“. Sufixo de UUID ...011 .

TÉCNICO

Predefinição Capture Class para vídeos de walkaround 360°. Densifier MCMC, 40 000 iterações, cap de 1,5 milhão de Gaussians, SSIM-loss 0.5 mais termo edge-aware 0.1, penalidade de anisotropia suave (peso 0.001 num limiar de razão 15). Máscaras de pessoas e de céu estão ativas. A predefinição espera um vídeo equirretangular 360° que é reprojeto internamente em crops perspectivos de cerca de 90° de largura antes de o treinamento começar. Validado em walkarounds 360° 8K com pau de selfie (cena Monument, verificado a olho).

EM PALAVRAS SIMPLES

Para vídeos de walkaround 360° — você caminha com uma câmera 360° ou pau de selfie por um ambiente ou ao redor de um objeto. O RadianceKit decompõe ele mesmo o panorama esférico em ângulos de visão normais e mascara transeuntes e céu. Para máxima nitidez no mesmo material, experimente também a variante High Detail (P11).

I P11 — 360° Walkaround (High Detail)



ONDE

Inspetor → Seção de predefinições → Grupo „Capture Class“ → entrada „360° Walkaround (High Detail)“. Sufixo de UUID `...013` (opt-in).



TÉCNICO

Predefinição Capture Class opt-in para vídeos de walkaround 360° com detalhe máximo. Densifier Híbrido (clone/split clássico por abs-gradiente + ruído MCMC + relocação), 40 000 iterações, cap de 1,5 milhão de Gaussians, penalidade de anisotropia 0.0015 num limiar de razão 15, SSIM-loss 0.2 e termo edge-aware 0 — a receita travada „r50“ de screen-split. Em material 360° bate a predefinição MCMC padrão „360° Walkaround“ (P10) em PSNR, LPIPS e confete visível, e isso com cerca de um terço da contagem de splats. Fica deliberadamente opt-in *ao lado* da predefinição 360 padrão, até ser validada em mais cenas.

EM PALAVRAS SIMPLES

A alternativa mais nítida à predefinição 360 padrão (P10): mais detalhe, menos confete, arquivo bem menor. Fica deliberadamente ao lado em vez de substituí-la — até agora confirmada em uma mão cheia de cenas. Se o seu walkaround 360° foi capturado de forma limpa, experimente esta predefinição primeiro e compare o resultado com P10.

I P12 — Photo / Object



ONDE

Inspetor → Seção de predefinições → Grupo „Capture Class“ → entrada „Photo / Object“. Sufixo de UUID `...012`.



TÉCNICO

Predefinição Capture Class para órbitas de objeto a partir de fotos individuais nítidas (sem vídeo). Densifier Híbrido-t1 (com relocação), 40 000 iterações, cap de 1,5 milhão de Gaussians, SSIM-loss 0.5 mais termo edge-aware 0.1, penalidade de anisotropia suave (peso 0.001 num limiar de razão 15), opacity decay 0.9995 a cada 50 iterações, **sem** masking. Validado em 163 fotos de alta resolução de 41 MP de um esqueleto (verificado a olho). Poucas views (até cerca de 600) ficam abaixo do limiar de collapse do Híbrido.

EM PALAVRAS SIMPLES

Para capturas de objeto a partir de fotos individuais nítidas — você circunda uma escultura, um modelo, um produto com a câmera e faz fotos em vez de vídeo. Sem masking, porque fotos nítidas costumam ter fundo limpo. Para fontes em vídeo, use em vez disso uma predefinição 360° ou Drone.

I P13 — Equilibrado (Híbrido)

ONDE

Inspetor → Seção de predefinições → Grupo „Híbrido“ → entrada „Equilibrado (Híbrido)“. Sufixo de UUID ...009 .

TÉCNICO

20 000 iterações com a estratégia de densificação Híbrida: o clone/split clássico guiado por gradiente coloca capacidade onde o loss precisa, o ruído SGLD do MCMC continua explorando, e Gaussians mortos são realocados em vez de se perderem no pruning. O opacity decay (V546) substitui os resets de opacidade; uma penalidade de anisotropia (peso 0.001, limiar de razão 15) mantém sob controle os splats em forma de agulha. O cap de Gaussians escala com a cena (base de 150K, scene-aware $\times 3.0$). Validado em cinco cenas contra MCMC puro com o mesmo orçamento: em média +0.45 dB de PSNR com 20–30 % menos Gaussians (stonehenge +1.23, family +0.82, garden +0.47 dB). Num M3 Ultra, tipicamente 5–10 minutos com 200 imagens.

EM PALAVRAS SIMPLES

Uma forte primeira escolha para um resultado final: detalhe mais nítido que as predefinições MCMC com um arquivo de tamanho parecido, numa fração do tempo de treinamento do P8. Se você só tem tempo para uma corrida de qualidade e nenhuma das Capture-Classes encaixa claramente, comece por aqui. As predefinições Classic continuam melhores para testes rápidos, e as predefinições Capture-Class (P9–P12) são a primeira escolha quando a sua cena claramente corresponde a um desses tipos de captura.

Qual predefinição, quando?

Cenário	Teste inicial	Corrida principal
Teste de funcionamento de imagens novas, < 30s	P1 Quick	—
Órbita de objeto a partir de fotos individuais nítidas	P2 Preview	P12 Photo / Object
Scan de objeto único (vídeo), < 500 fotos	P2 Preview	P4 Quality ou P8 Quality MCMC
Vídeo de walkaround 360°	P6 Preview MCMC	P10 360° Walkaround (nítido: P11 High Detail)
Órbita aérea / de drone, paisagem	P6 Preview MCMC	P9 Drone / Aerial
Distribuição na web (pequeno, compacto)	P2	P8 Quality MCMC (arquivo menor com qualidade plena)
Detalhe nítido em pouco tempo, exportação compacta	P2 ou P6	P13 Equilibrado (Híbrido)
Máxima fidelidade de detalhe, estratégia Classic	P3 ou P6	P5 Ultra Detail
Impressão, marketing, detalhe pleno	P3 ou P6	P4 Quality (Classic) ou P5 Ultra Detail

Comparação rápida

Pre-set	Estratégia	Iters	Max-Gs	Escala de render	Tempo típico (200 imagens, M3 Ultra)	Q-Sweep
P1 Quick	Classic	1 000	∞	0.25x	~30 s	—
P2 Preview	Classic	5 000	∞	0.5x	2–3 min	—
P3 Balanced	Classic	20 000	∞	1.0x	30–60 s	—
P4 Quality	Classic	35 000	∞	1.0x	2–4 min	V546 HTGS
P5 Ultra Detail	Classic	~35 000	∞	1.0x	2–5 min	Matrix $\Delta+0.94$ dB
P6 Preview MCMC	MCMC	60 000	100 K	1.0x	5–8 min	—
P7 Balanced MCMC	MCMC	120 000	150 K	1.0x	8–15 min	—
P8 Quality MCMC	MCMC	200 000	150 K	1.0x	20–35 min	V544a
P9 Drone / Aerial	MCMC	40 000	1.5 M	1.0x	10–25 min	Olho / Via-duto
P10 360° Walkaround	MCMC	40 000	1.5 M	1.0x	10–25 min	Olho / Monument
P11 360° Walkaround (High Detail)	Híbrido	40 000	1.5 M	1.0x	10–25 min	Olho (opt-in)
P12 Photo / Object	Híbrido	40 000	1.5 M	1.0x	10–25 min	Olho / Esqueleto
P13 Equilibrado (Híbrido)	Híbrido	20 000	150 K	1.0x	5–10 min	Matrix $\Delta+0.45$ dB

Predefinições próprias

Pelo botão **Save...** na seção de predefinições (I1 no Capítulo 2), você salva a configuração de treinamento atual como uma predefinição própria. Predefinições próprias não são „Built-in“ e podem ser renomeadas, exportadas (como JSON), compartilhadas por arrastar/soltar, duplicadas e apagadas. As treze predefinições embutidas P1–P13 ficam intocadas pelo botão de apagar.

Regra prática: se você muda algo numa predefinição que quiser usar com mais frequência — Sky Dome ligado, peso de SSIM maior para uma certa classe de cena, número de iters diferente —, salve a variante como predefinição própria. Assim, na próxima corrida você já sabe que é uma configuração diferente do padrão.

CAPÍTULO

Capítulo 8 — Formatos de exportação

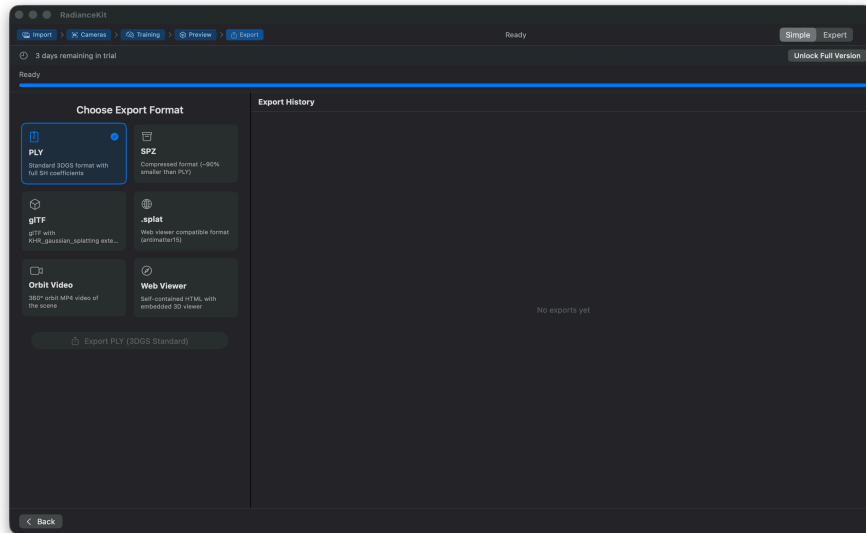


Figura 29: Seleção de formato de exportação no Modo Iniciante — seis cards de formato

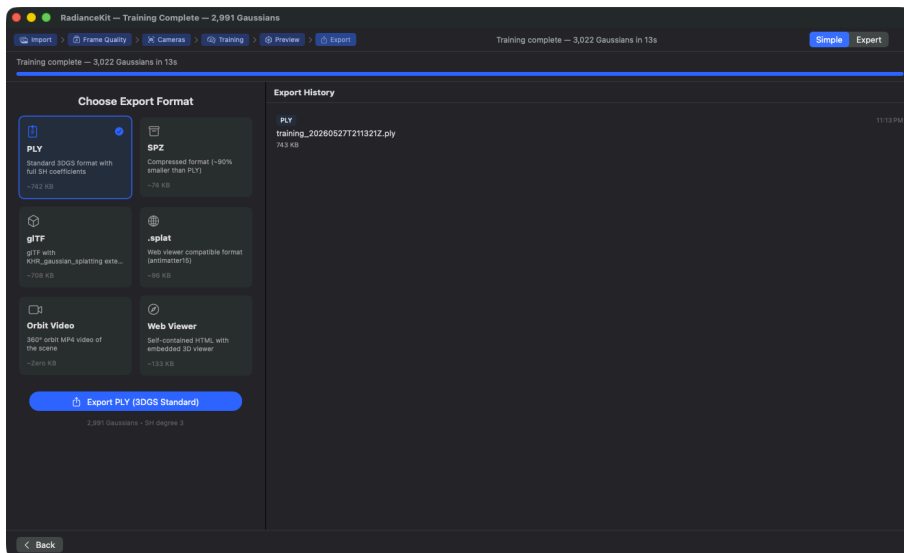


Figura 30: Grid de formatos de exportação ao vivo após treinamento de 5K iters num buquê — todos os seis cards com cálculo dinâmico de tamanho (PLY 742 KB selecionado, SPZ 74 KB, glTF 708 KB, .splat 96 KB, Orbit Video ~Zero KB, Web Viewer 133 KB), Export History à direita com PLY já salvo

O que aparece na imagem (2 991 Gaussians, SH degree 3, buquê sintético em Blender como conjunto de teste IP-clean): os tamanhos sob cada card de formato

são calculados ao vivo a partir do número atual de Gaussians e do overhead do formato — não estão fixos. De 2 991 Gaussians (SH degree 3) saem 742 KB de PLY, 74 KB de SPZ (fator ~10x menor por quantização), 708 KB de glTF (com extensão `KHR_gaussian_splatting`, por isso quase equivalente ao PLY), 96 KB de `.splat` (formato compactado de 24 bytes por Gaussian). Orbit Video mostra „~Zero KB” porque o tamanho só é conhecido depois do encoding do MP4. Web Viewer (133 KB) empacota um arquivo HTML autônomo com viewer WebGL embutido e dados de `splat` comprimidos — maior que `.splat` puro por causa do overhead do viewer. Export History à direita lista um export de PLY já concluído („`training_20260527T211321Z.ply`, 743 KB, 23:13”) com pill do formato e ação Reveal no Finder.

Um treinamento concluído entrega uma nuvem de Gaussians — uma coleção de algumas centenas de milhares a milhões de distribuições gaussianas 3D que juntas reconstroem a cena. O RadianceKit conhece dez formas de gravar essa nuvem em disco. Seis delas são formatos de dados 3D puros (PLY, Compressed PLY, SPZ, SOG, glTF, `.splat`); um empacota a nuvem junto com um viewer HTML pronto (Web Viewer); um renderiza um MP4 a partir de uma trajetória orbital (Orbit Video); e dois exportam não o conteúdo Gaussian em si, mas apenas o resultado de SfM (poses de câmera e nuvem de pontos grosseira) para reuso em outros pipelines de treinamento (`transforms.json` + COLMAP-Workspace).

Qual formato é o certo em cada momento depende do objetivo. Para arquivar dados completos sem perda de qualidade, use PLY. Para viewers web na sua própria página, geralmente basta `.splat` ou o Web Viewer embutido. Quando o arquivo precisa ser mínimo, vale SPZ ou SOG. Para reusar o resultado de SfM em Nerfstudio, Postshot ou Brush, os caminhos certos são `transforms.json` e o COLMAP-Workspace.

Todas as funções de exportação estão no menu „Export” e também no Modo Iniciante, no último passo do assistente. A maioria dos formatos é totalmente compatível com sandbox e funciona na versão da App Store. Só o SOG exige um binário externo (`cwebp`), que pode não estar disponível no build da App Store — detalhes em E4.

I E1 — PLY (.ply)

ONDE

Barra de menus → Export → 3D Formats → Export PLY... (⌘E). Modo Iniciante: passo do assistente Export → card „PLY“. **Tamanho:** tipicamente 100 % (valor de referência). **Compatível com:** SuperSplat, PolyCam, todos os viewers 3DGS.

TÉCNICO

PLY é o formato canônico de armazenamento para 3D Gaussian Splatting. O RadianceKit grava um arquivo binário little-endian com o layout padronizado de propriedades 3DGS: por Gaussian, posição de três componentes, três normais sempre em zero, três coeficientes DC-SH (`f_dc_0..2`) para a cor base RGB, depois até 45 coeficientes SH adicionais (`f_rest_0..44`) no arranjo channel-major transposto definido no artigo Kerbl-2023 (primeiro todos os coeficientes do canal R, depois G, depois B), seguidos de opacidade logit (valores pré-sigmoide brutos), três escalas em log space e uma quaternion wxyz de rotação. O grau SH máximo exportado é clampeado para o mínimo entre o desejado pelo usuário e o realmente aprendido; padrão é 3 (45 coeficientes de resto). Antes da escrita, o tamanho do payload é calculado em inteiros de 64 bits para capturar overflow em nuvens muito grandes. O arquivo é gravado de forma atômica, o que ocupa temporariamente o dobro de espaço em disco para nuvens grandes.

EM PALAVRAS SIMPLES

Esse é o „arquivo original“. Maior arquivo, compatibilidade máxima, sem perdas. Se você não sabe qual formato escolher, escolha PLY — abre em quase todas as ferramentas 3DGS. Para 1 milhão de Gaussians, dependendo do grau SH, são entre 200 e 800 MB. Se o arquivo ficar grande demais, dê uma olhada em E2 (PLY comprimido) ou E3 (SPZ).

I E2 — Compressed PLY (.ply)

ONDE

Barra de menus → Export → 3D Formats → Export Compressed PLY... Modo Iniciante: card „Compressed PLY“. **Tamanho:** cerca de 10–20 % do PLY (compressão de 5 a 10×). **Compatível com:** SuperSplat, engine PlayCanvas, viewers baseados em web.

TÉCNICO

A variante PlayCanvas do formato PLY com quantização em chunks. Os Gaussians são agrupados em chunks de 256. Por chunk são gravados no header bounds mín/máx separados para posição, escala e cor; os Gaussians individuais referenciam seus valores relativos a esses bounds e são comprimidos em 32 bits cada: posição e escala com packing 11-10-11 bits, rotação como quaternion „smallest three“ 2-10-10-10 bits, cor como RGBA 8-8-8-8. Coeficientes SH mais altos são quantizados com apenas 8 bits por componente (`shCoeffCount * 3` uchar por Gaussian). O formato em si ainda é PLY com header ASCII e portanto validável com ferramentas PLY, mas as propriedades de vertex são declaradas como campos `uint`. O grau SH padrão é 0 (sem coeficientes de resto), para maximizar a compressão — graus SH mais altos podem ser escolhidos explicitamente.

EM PALAVRAS SIMPLES

A variante PLY econômica em espaço. Compatibilidade idêntica do PLY normal com as engines, mas 5 a 10 vezes menor. SuperSplat e PlayCanvas leem nativamente. Para deploy em web, quase sempre melhor que PLY normal. A perda de qualidade pela quantização normalmente não é perceptível visualmente, desde que a cena não contenha detalhes extremamente de alta frequência.

| E3 — SPZ (.spz)

ONDE

Barra de menus → Export → 3D Formats → Export SPZ... Modo Iniciante: card „SPZ“. **Tamanho:** cerca de 10 % do PLY (90 % menor). **Compatível com:** Niantic Scaniverse, Niantic Spatial Fields, MetalSplatler.

TÉCNICO

Formato SPZ v2 da Niantic. Posições empacotadas como ponto fixo de 24 bits (resolução de cerca de 0,25 mm), escalas como quantização de 8 bits em log space, rotações como „smallest three“ de 8 bits (em v2 só são gravados xyz, w é derivado pelo decoder a partir da norma do quaternion), opacidades como valores de 8 bits passados por sigmoide. DC-SH é gravado com uma fórmula de pack específica do SPZ ($dc_raw * 0.15 * 255 + 0.5 * 255$); bandas SH mais altas com 5 bits (banda 1) ou 4 bits (bandas 2–3) por coeficiente. Todo o blob binário empacotado é então comprimido com gzip padrão (RFC 1952), o que resulta em um contêiner gzipped com magic bytes `1f 8b`. O RadianceKit chama o `gzip` do sistema porque a API zlib embutida da Apple gera um framing proprietário Apple que não seria compatível com os leitores SPZ em Spatial Fields ou MetalSplatler. O `gzip` do sistema permanece spawnável também dentro do sandbox macOS.

EM PALAVRAS SIMPLES

O menor arquivo padrão. Se você conhece o Scaniverse da Niantic — é o formato que o app usa. Muito pequeno, amigável de carregar para apps móveis. Diretamente utilizável no Cloud Viewer da Niantic (Spatial Fields). Cerca de 90 % menor que um PLY com os mesmos dados, sendo opticamente quase indistinguível para a maioria das cenas.

| E4 — SOG (.sog)

ONDE

Barra de menus → Export → 3D Formats → Export SOG.... Modo Iniciante: card „SOG“. **Tamanho:** cerca de 5–6 % do PLY (compressão de 15 a 20× — a menor opção). **Compatível com:** engine PlayCanvas, editor SuperSplat.

TÉCNICO

„Spatially Ordered Gaussians“ — um formato PlayCanvas que armazena a nuvem GPU-ready em várias imagens WebP sem perda. Primeiro, todos os Gaussians são ordenados espacialmente por código Morton 3D (Z-Order de 30 bits, 10 bits por eixo), o que dá às imagens cache locality posterior no renderer. Depois, posições são quantizadas para 16 bits com transformação log simétrica (para melhor alcance dinâmico) e divididas em duas imagens RGBA (`means_l.webp` para os 8 bits inferiores, `means_u.webp` para os superiores). Rotações são codificadas como „smallest three“ com 3×8 bits mais 2 bits de modo numa imagem RGBA (modo vai para alpha como `252 + largest`). Escalas e DC-SH são quantizadas com um codebook de 256 entradas cada (baseado em percentis distribuído sobre todos os valores), os índices vão para `scales.webp` e `sh0.webp` . As cinco imagens mais um `meta.json` com codebooks e bounds são empacotadas num ZIP (encoder próprio, porque o sandbox bloqueia o `zip` do sistema) e salvas com a extensão `.sog` .

Atenção sandbox: SOG é a única opção de formato que exige um binário externo. O estágio de encoder WebP chama `cwebp` de `/usr/local/bin/cwebp` ou `/opt/homebrew/bin/cwebp`. Se nenhum binário `cwebp` for encontrado, o código cai para encoding PNG bruto — mas: **o fallback PNG não funciona no SuperSplat**. Na versão da App Store, avalia a disponibilidade pela variante de build; na variante de desenvolvimento, `cwebp` deve estar instalado via Homebrew (`brew install webp`).

EM PALAVRAS SIMPLES

O menor formato 3DGS de todos, bem menor que SPZ. Mas: precisa da ferramenta `cwebp` no seu Mac, porque o RadianceKit em si não consegue gerar todos os formatos de imagem. Instale uma vez com Homebrew e tudo roda. Na versão da App Store, talvez não funcione 100 % — se na exportação sair PNG em vez de WebP, você não consegue abrir o arquivo direto no SuperSplat. Quem trabalha sem Homebrew, use SPZ (E3).

| E5 — glTF (.glb)

ONDE

Barra de menus → Export → 3D Formats → Export glTF... Modo Iniciante: card „glTF“. **Tamanho:** comparável ao PLY. **Compatível com:** viewers glTF com extensão `KHR_gaussian_splatting` (rascunho de padrão Khronos).

TÉCNICO

Grava um arquivo binário `.glb` autocontido (sem anexo separado de arquivo bin) conforme a especificação da extensão `KHR_gaussian_splatting`. Posições são gravadas como dados de vertex glTF `POSITION` regulares (float3); todos os demais atributos (rotação como float4, escala como float3, opacidade como float, coeficientes SH como float3 x `shCoeffCount`) ficam em atributos de vertex adicionais referenciados pela extensão. Importante: glTF usa sistema de coordenadas Y-up destrímano, enquanto COLMAP/3DGS trabalham Y-down/Z-forward. Por isso o exporter aplica uma rotação de 180° em torno do eixo X — posições são reescritas como `(x, -y, -z)`; quaternions são ajustados para `(w, x, -y, -z)`. Isso produz uma representação geometricamente correta e destrímana (não espelhada) em viewers glTF. Chunks JSON e binário são pad-ed para alinhamento de 4 bytes, como exigido pelo padrão GLB.

EM PALAVRAS SIMPLES

O formato padrão oficial Khronos para dados 3D, na nova extensão para Gaussian Splats. Vantagem: glTF é difundido em todas as grandes engines 3D (Babylon.js, Three.js, Unity, Unreal). Desvantagem: a extensão ainda está em rascunho em 2026, e muitos viewers ainda não a suportam. Útil sobretudo se você integra dados de splat num pipeline glTF existente ou escreve um viewer que já é compatível com glTF.

I E6 — Splat (.splat)

ONDE

Barra de menus → Export → 3D Formats → Export .splat... Modo Iniciante: card „.splat“. **Tamanho:** exatamente 32 bytes por Gaussian. **Compatível com:** gsplat.js, viewers baseados em web (referência antimatter15), a maioria das demos de 3DGS em browser.

TÉCNICO

O formato `.splat` do antimatter15 — 32 bytes por Gaussian, sem header, sem indireção. Layout por entrada: 3 × float32 posição (coordenadas mundo), 3 × float32 escala (transformação exp a partir do log space do buffer interno), 4 × uint8 cor RGBA (coeficiente DC-SH escalado com `SH_C0 = 0.282...` e clampeado para `[0,255]`), 4 × uint8 quaternion (w,x,y,z, normalizado e codificado em byte como `128 + 128*q`). Só o DC-SH é gravado — bandas SH mais altas são descartadas. Isso torna o formato extremamente compacto, mas custa as variações de cor dependentes de vista que aparecem em reflexões ou highlights especulares. A ordem de escrita é exatamente a ordem dos índices da nuvem (sem ordenação espacial); viewers web como `gsplat.js` renderizam a partir disso.

EM PALAVRAS SIMPLES

O formato preferido se você quer mostrar o splat num viewer web próprio com `gsplat.js`. Muito compacto (32 bytes/Gaussian), mas sem grau SH mais alto — ou seja, sem reflexos brilhantes ou alterações sutis de cor por ângulo. Para a maioria das aplicações web sem problema, porque a cor DC basta e a ausência de dependência de vista quase não chama a atenção.

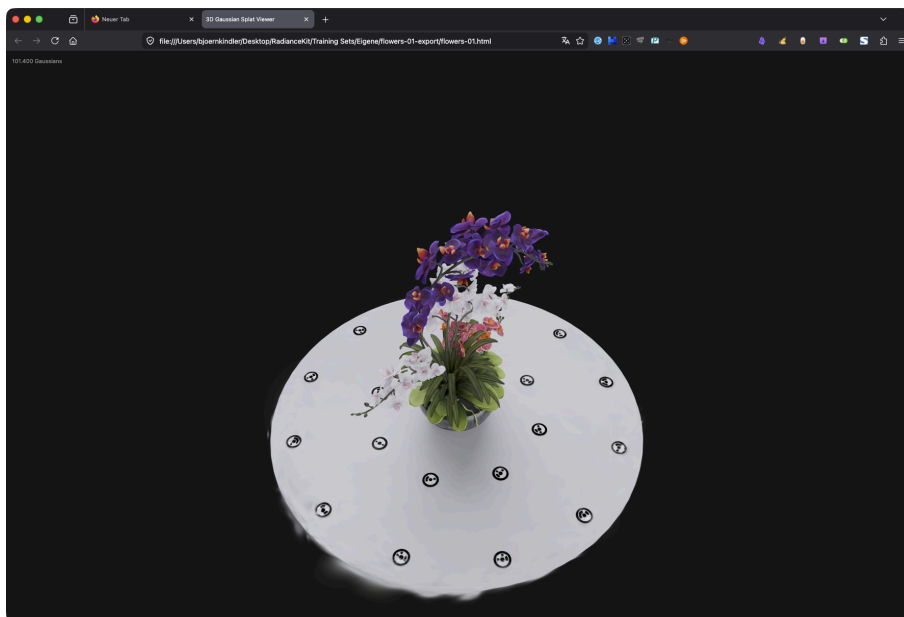


Figura 31: Web Viewer aberto no Firefox — splat do buquê renderizado com esferas marcadoras de câmera ao redor, barra de abas do browser visível em cima, sem CDN/servidor. Arquivo independente `flowers-01.html` aberto direto do Finder por duplo clique no browser padrão — o programa WebGL2 embutido renderiza a nuvem Gaussian imediatamente, sem rede ou servidor. Os marcadores pretos em volta do buquê são as câmeras de treinamento, opcionais de exibir. Drag do mouse rotaciona, scroll dá zoom.

| E7 — Web Viewer (.html)

ONDE

Barra de menus → Export → Media → Export Web Viewer... Modo Iniciante: card „Web Viewer“. **Tamanho:** dados de splat codificados em base64 (\approx overhead 4/3) + cerca de 5 KB de shell HTML/JS. **Compatível com:** qualquer browser moderno com WebGL2 (todos os desktops, iOS 15+, Android 5+).

TÉCNICO

Empacota a nuvem Gaussian junto com um renderer WebGL2 totalmente inline num único arquivo `.html`. Sem dependências de CDN, sem WASM, sem segundo arquivo. A nuvem é internamente codificada primeiro como binário `.splat` (mesma lógica de 32 bytes de E6), embutida em base64 e decodificada com `atob` no browser. O renderer embutido faz a sua própria ordenação WebGL2, controle orbital com mouse e ordenação CPU por frame; todo o código JS (shaders, matemática, loop) é visível no HTML de saída. A convenção de eixos na fronteira armazenamento-renderer é exatamente a mesma de E5: posição $(x, -y, -z)$, quaternion $(w, x, -y, -z)$. Opcionalmente pode aparecer uma sobreposição de branding (chave do free tier). Como tudo é inline, o arquivo funciona também direto do protocolo `file://` — sem necessidade de servidor web local para testar.

EM PALAVRAS SIMPLES

Um único arquivo HTML que você pode mandar por e-mail ou embutir num site. Duplo clique no Finder e o browser mostra a cena com rotação pelo mouse. Sem upload para a nuvem, sem segundo arquivo, sem servidor. Ideal para apresentações a clientes, portfólio, anexos de e-mail. Desvantagem: o arquivo fica cerca de um terço maior que um `.splat` puro por causa da codificação base64 — para cenas muito grandes vale hospedar o arquivo `.splat` separadamente com um viewer padrão.

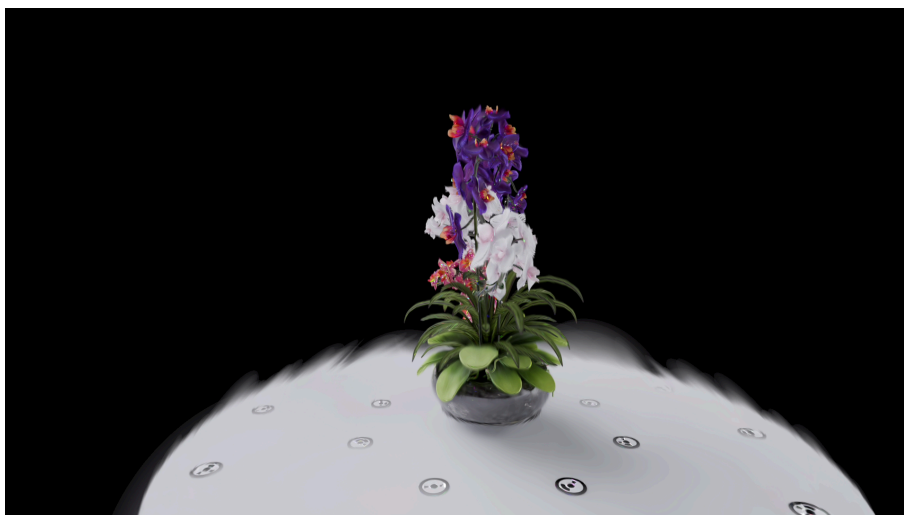


Figura 32: Frame único extraído de flowers-01.mp4 — buquê em render de perfil, plataforma branca com marcadores de câmera visíveis, fundo preto (background padrão do viewport, ajustável em Configurações). A câmera circula a cena por um caminho paramétrico (elevação + distância fixas, yaw rotacional), duração típica 6–10 segundos a 30 ou 60 fps. Resolução de frame escalável de 480p até 8K via VideoPreset.

| E8 — Orbit Video (.mp4/.mov)

ONDE

Barra de menus → Viewport → Record Turntable Video OU barra de menus → Export → Media → Export Orbit Video.... Modo Iniciante: card „Orbit Video“ com slider de duração 3–30 s. **Tamanho:** depende de duração, resolução, bitrate. **Compatível com:** todas as plataformas (H.264 e HEVC são padrão Apple).

TÉCNICO

Renderiza a nuvem Gaussian ao longo de uma trajetória orbital paramétrica e codifica cada frame via AVAssetWriter em arquivo MP4 ou MOV. A configuração orbital controla rotações (voltas), distância, elevação, FOV, duração e fator de ease in/out. A exportação de orbit video roda pelo ForwardPass PRÓPRIO do RadianceKit com avaliação completa de SH — pixel-idêntica ao viewport in-app (WY-SIWYG). Por frame, a matriz de alinhamento de mundo (calculada pelo renderer para girar as coordenadas internas para o mundo orbital Y-up) é multiplicada pela câmera; em seguida aplica-se um flip de convenção de câmera (camFlip: orbital Y-up → COLMAP Y-down). O alvo de render offscreen é puxado via IOSurface para um CVPixelBuffer para o encoder. O encoder suporta H.264 e HEVC, bitrate e resolução configuráveis de 480p a 8K. Antes do primeiro frame o renderer espera 200 ms para que a ordenação inicial dos splats termine. Esta exportação é GPU-bound — em 8K com milhões de Gaussians, o tempo de render por frame fica em vários segundos, ou seja, tempos totais de 10–30 minutos para um vídeo de 6 s são possíveis.

EM PALAVRAS SIMPLES

Um arquivo MP4 pronto com uma rotação em torno da sua cena. Perfeito para redes sociais, marketing, apresentações. Você pode definir duração (3–30 segundos), sentido e velocidade de rotação. O arquivo embute direto no YouTube, Instagram, PowerPoint e qualquer outro lugar. Pode demorar um pouco porque o app precisa renderizar cada frame por completo — para um vídeo 8K planeje cinco a trinta minutos, dependendo do número de Gaussians.

| E9 — SfM Transforms (transforms.json)

ONDE

Barra de menus → Export → Photogrammetry → Export SfM (transforms.json).... **Tamanho:** tipicamente 1–10 KB (só poses + intrínsecas, sem imagens, sem Gaussians). **Compatível com:** nerfstudio, Brush, gsplat, OpenSplat, Meshroom, todos os treinadores feed forward 3DGS modernos.

TÉCNICO

Grava o formato `transforms.json` do nerfstudio com uma lista de poses de câmera mais intrínsecas compartilhadas. Para cada câmera, a matriz de view (interno do RadianceKit: World-to-Camera em convenção COLMAP) é invertida; em seguida os vetores base Y e Z em coordenadas locais da câmera são espelhados para converter para a convenção do nerfstudio (estilo OpenGL, câmera olha ao longo de `-Z`, `+Y` para cima). A matriz 4×4 final vai como array aninhado row-major de doubles no campo `transform_matrix` de cada frame. As intrínsecas são gravadas em top level (focal x/y, principal point x/y, largura/altura da imagem, `camera_model = "OPENCV"`, mais os coeficientes de distorção `k1`, `k2`, `p1`, `p2`) — exceto quando o exporter detecta vários conjuntos de intrínsecas diferentes; aí são gravados por frame. Caminhos de imagem são escritos como `images/<filename>` relativos ao JSON; o usuário deve criar uma pasta sibling `images/` com as fotos de treinamento.

EM PALAVRAS SIMPLES

Esse JSON descreve, para cada foto, onde a câmera estava e para onde ela olhava. O arquivo sozinho é pequeno e inútil — ele é usado junto com as imagens originais numa pasta. Nerfstudio, Brush e algumas outras ferramentas leem exatamente esse formato, e com isso você passa seus resultados de SfM do RadianceKit para outra ferramenta sem que ela precise recalcular a reconstrução de câmera. Em cenas grandes economiza horas.

I E10 — COLMAP Workspace (sparse/0/)

ONDE

Barra de menus → Export → Photogrammetry → Export SfM (COLMAP Workspace)... **Tamanho:** três arquivos binários, no total tipicamente 4–8 MB — `points3D.bin` domina (uma linha por ponto 3D da sparse cloud), `images.bin` e `cameras.bin` cada um bem abaixo de 100 KB. **Compatível com:** o próprio COLMAP, Nerfstudio, Postshot, Meshroom, todas as ferramentas que esperam um diretório `sparse/` do COLMAP.

TÉCNICO

Grava o layout padrão COLMAP `sparse/0/` com três arquivos binários: `cameras.bin`, `images.bin`, `points3D.bin`. A referência de formato é a documentação oficial do COLMAP. `cameras.bin` contém a lista deduplicada de intrínsecas (câmeras com intrínsecas + tamanho de imagem idênticos viram uma única entrada); o modelo de câmera usado é `OPENCV` (modelo 4), com `fx/fy/cx/cy` mais os quatro coeficientes de distorção `k1/k2/p1/p2`. `images.bin` lista por imagem a pose como quaternion `wxyz` mais translation, seguida do ID da câmera e do nome do arquivo; sem correspondências 2D-3D armazenadas. `points3D.bin` contém a nuvem de pontos SfM com posição, cor (RGB 0-255) e valores padrão para reprojeção e track length. Tudo é gravado em little-endian. O reimport no RadiancKit funciona pelo menu File → „Import COLMAP/Metashape Workspace...” (ver Q3 no capítulo de backends de SfM).

EM PALAVRAS SIMPLES

O formato COLMAP oficial. Se você quer continuar o treinamento no Postshot, Nerfstudio ou outra ferramenta compatível com COLMAP, é o caminho. Três arquivos pequenos mais suas imagens originais, e o programa-alvo aceita como se o COLMAP em si tivesse sido o programa fonte. Mais programas entendem isso do que o formato `transforms.json` (E9), ao mesmo tempo um pouco menos prático porque é binário em vez de texto.

Qual formato, quando?

Objetivo	Formato
Viewer web na própria página	E7 Web Viewer (.html)
Viewer web com <code>gsplat.js</code>	E6 Splat (.splat)
Reuso no pipeline Postshot / Nerfstudio	E9 transforms.json + E10 COLMAP Workspace
Edição no SuperSplat	E1 PLY ou E2 Compressed PLY
Niantic Scaniverse / Spatial Fields	E3 SPZ
Compressão máxima	E4 SOG (cwebp necessário)
Vídeo de marketing / redes	E8 Orbit Video

Comparação rápida

For- mato	Extensão	Sandbox	Tamanho (1M Gauss)	Melhor uso
E1 PLY	<code>.ply</code>	sim	~250 MB	Arquivo, maior com- patibilidade
E2 Com- pressed PLY	<code>.ply</code>	sim	~40 MB	Web + Su- perSplat
E3 SPZ	<code>.spz</code>	sim (spawn de gzip)	~40 MB	Niantic + mobile
E4 SOG	<code>.sog</code>	condi- cional (cwebp)	~20 MB	Compres- são máxima
E5 glTF	<code>.glb</code>	sim	~250 MB	Pipeline Kh- ronos
E6 Splat	<code>.splat</code>	sim	~32 MB	Viewer web gsplat.js
E7 Web Viewer	<code>.html</code>	sim	~45 MB	Arquivo standalone para brow- ser
E8 Orbit Video	<code>.mp4 / .mov</code>	sim	variável	Redes/mar- keting
E9 SfM Trans- forms	<code>.json</code>	sim	~5 KB	Repassé de poses
E10 COLMAP Works- pace	diretório	sim	~4–8 MB	Repassé de poses biná- rio

A coluna de tamanho são valores aproximados para 1 milhão de Gaussians com grau SH 3. Valores reais variam conforme a compressibilidade da cena; grau SH 0 reduz PLY/glTF por um fator de 4.

CAPÍTULO

Capítulo 9 — Backends de SfM

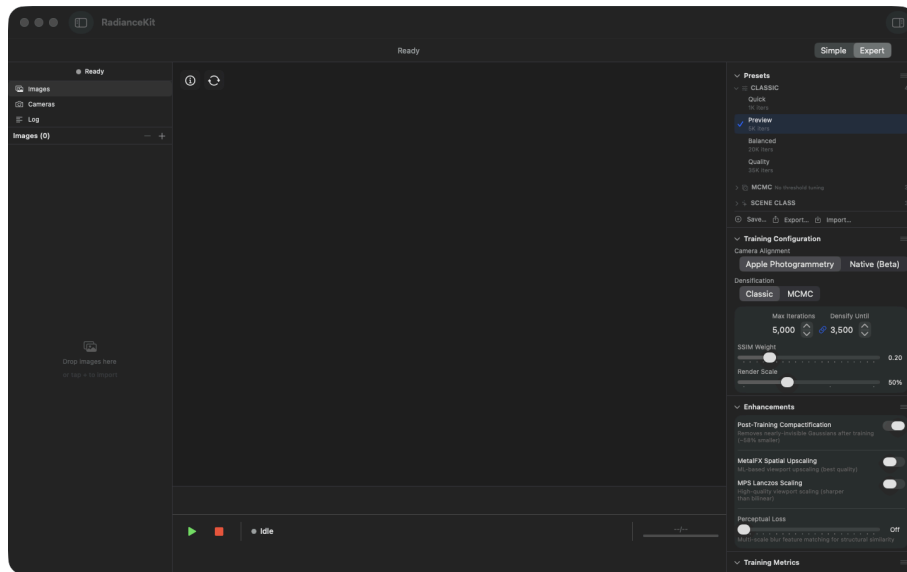


Figura 33: Modo Especialista com seletor Camera Alignment no Inspetor (Apple Photogrammetry / Native (Beta))

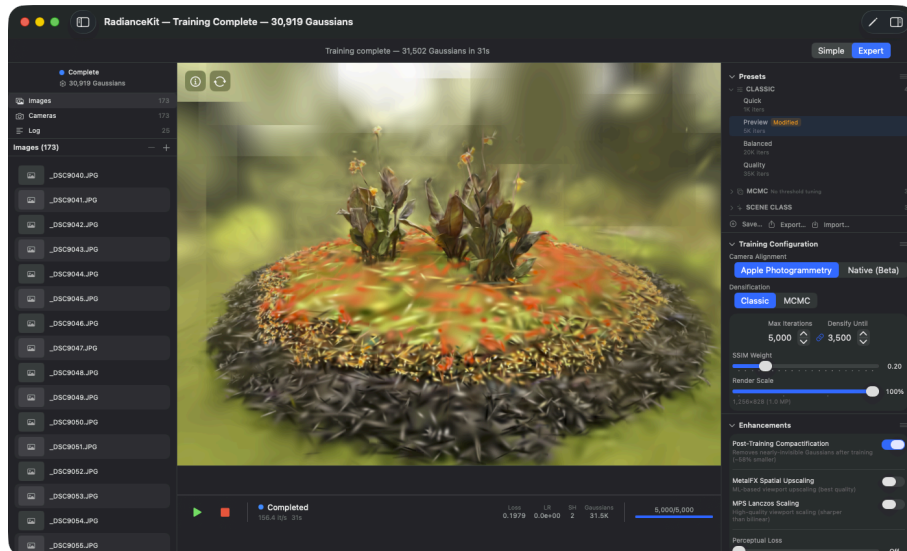


Figura 34: Inspetor com Native (Beta) ativo — segunda opção do seletor Camera Alignment selecionada, todos os outros parâmetros de configuração de treinamento inalterados

O QUE APARECE NA IMAGEM O seletor Camera Alignment no Inspetor é um segmented control com duas opções — Apple Photogrammetry (padrão para builds da App Store, totalmente compatível com sandbox) e Native (Beta) (o backend de pipeline próprio do

RadianceKit baseado em FAST+BRIEF+GLOMAP, desenvolvido nas Fases 3.8/3.9, stand 2026-05). Native (Beta) está validado apenas para orbit e é mais rápido com $\geq 1\ 000$ frames que Apple Photogrammetry, mas ainda não atinge o gate de qualidade Phase-3-§5 (finalLoss ≤ 0.0115) — daí a tag Beta. Resultados externos de SfM do Metashape, COLMAP ou de outro software de fotogrametria podem ser importados adicionalmente pelo menu File (Q3 formato texto COLMAP, Q6 import de workspace) — o seletor não muda, mas as poses importadas substituem o resultado do SfM.

SfM significa **Structure from Motion**. A partir de um conjunto de fotos sobrepostas, o software reconstrói para cada imagem a posição e a direção de visão da câmera em um sistema de coordenadas 3D comum. Para isso é gerada uma nuvem grosseira de pontos 3D, que inicializa o treinamento por Gaussian Splatting. O resultado de SfM é a entrada do treinamento propriamente dito e determina decisivamente a qualidade de imagem posterior.

O RadianceKit oferece cinco caminhos de SfM: dois backends embutidos no app (Q1 Apple Photogrammetry, Q4/Q5 Native), dois caminhos de import de ferramentas externas (Q3 formato texto COLMAP, Q6 import binário de workspace) e ainda Q2 COLMAP-Binary, disponível apenas em builds de desenvolvimento fora da App Store. Qual é o certo depende do tipo de cena (orbit em torno de um objeto, ambiente interno, voo de drone) e se um software externo já entrega uma reconstrução.

I Q1 — Apple Photogrammetry

ONDE

Expert View → Inspetor → Configuração de Treinamento → seletor Camera Alignment, entrada „Apple Photogrammetry“.

TÉCNICO

Encapsula o framework Photogrammetry embutido da Apple, originalmente desenvolvido para Object Capture. A Apple extrai internamente features com um pipeline proprietário (os passos não são documentados publicamente), os verifica via multi-view matching e resolve bundle adjustment na Neural Engine + GPU do Apple Silicon. O backend é totalmente compatível com a App Store (sem binário externo, Sandbox=true, on-device), mas entrega apenas poses de câmera mais uma nuvem grosseira de pontos — sem métricas de diagnóstico como comprimento de track ou erro de reprojeção. Escala, segundo recomendação da Apple, até algumas centenas de imagens. Acima de ~500 frames em voos lineares de drone ou grandes cenas externas, foram observados de forma reprodutível travamentos ou descarte silencioso de câmeras individuais.

EM PALAVRAS SIMPLES

É o caminho mais fácil. Joga as imagens, o app calcula. Funciona muito bem em scans clássicos de objeto — quando você anda em volta de um móvel ou escultura e tira 50–200 fotos. Em voos de drone sobre paisagens ou com muitas imagens (acima de 500), o método da Apple costuma ficar instável. Para essas cenas, teste o backend Native (Q4/Q5) ou calcule as câmeras no Metashape e traga pelo import de workspace (Q6).

USUÁRIO AVANÇADO

Q2 COLMAP-Binary — dispara o programa COLMAP externo como subprocesso e por isso **não está disponível** na versão da App Store (sandbox). Funciona apenas em builds de desenvolvimento fora da App Store. Para a qualidade que o COLMAP entrega, existe na versão da App Store o import de workspace (Q3 ou Q6): calcule o SfM no COLMAP ou no Metashape externamente e carregue o resultado aqui.

Q3 — Formato texto COLMAP (Metashape / ETH3D) **ONDE**

Menu „File → Import COLMAP / Metashape Workspace...” (Cmd+⇧+I) OU arrastar e soltar uma pasta com `sparse/0/cameras.txt`.

 **TÉCNICO**

Lê a exportação padronizada em texto do COLMAP — três arquivos de texto `cameras.txt`, `images.txt`, `points3D.txt` na subpasta `sparse/0/` — e converte para o modelo interno de resultado de SfM. Mesma definição de formato que a exportação binária do COLMAP, só que em ASCII em vez de binário. É gerada por Agisoft Metashape, RealityCapture, PolyCam e pelo benchmark ETH3D exatamente nesse layout. O parser compartilha a detecção de modelo de câmera com o parser binário (todos os modelos usuais: SIMPLE_PINHOLE, PINHOLE, OPENCV, OPENCV_FISHEYE, FULL_OPENCV). Robusto a linhas de comentário e linhas vazias. Em testes escala até ~1 400 câmeras (ETH3D Tunnel) sem problemas.

 **EM PALAVRAS SIMPLES**

Se você já trabalhou com Metashape, RealityCapture ou outro software comercial de foto-3D e exportou o resultado — esse export você carrega direto no RadianceKit, sem que o app precise calcular tudo de novo. Isso economiza horas de espera. Carregue a pasta inteira pelo menu File ou arraste para a janela.

I Q4 — SfM Nativo (incremental)

ONDE

Expert View → Inspetor → Configuração de Treinamento → seletor Camera Alignment, entrada „Native (Beta)“. Incremental é o modo padrão desse backend — não há seletor de mapper separado no Inspetor. Pela CLI, o modo pode ser definido explicitamente com `--native-sfm` ou `--sfm-mapper incremental`.

TÉCNICO

Implementação própria com aceleração de GPU de todo o pipeline de SfM: features FAST+BRIEF OU SuperPoint+LightGlue via CoreML (com `--coreml-features`), seguido de matching Hamming-KNN, RANSAC com matriz fundamental, track building, seleção de par inicial, two-view bootstrap (F→E mais DLT), mapper incremental greedy com registro PnP e triangulação multi-view e bundle adjustment final via Levenberg-Marquardt reduzido por Schur com loss de Huber e jacobianos analíticos via Cholesky solve. Totalmente compatível com a App Store: sem binário externo, Sandbox=true. Com o detector de R2-collapse entregue na Fase 3.10: se o app registra menos de 60 % dos frames de entrada ou se a taxa de pontos por câmera cai abaixo de 13, automaticamente se desvia para o mapper global (Q5). Empiricamente limpo em cenas orbit/turntable; em movimentos mais gerais (voo de drone, ambientes internos com geometria complexa) a taxa de sucesso é mais baixa — o detector pega esses casos. Escala até ~200 câmeras de forma confiável, mais com tempo de execução bem maior.

EM PALAVRAS SIMPLES

As forças da Apple (compatível com App Store, rápido em orbits com valores de diagnóstico adicionais. Funciona especialmente bem quando você anda em torno de um motivo, como num Object Capture. Em capturas mais complicadas (voo de drone ou sala), o RadianceKit reconhece automaticamente que vai dar ruim e pula para o método global. Marcado „Beta“ porque ainda está em testes — a recomendação padrão continua sendo Apple Photogrammetry para scans simples de objeto e o import de workspace (Q3 ou Q6) para conjuntos externos exigentes.

I Q5 — SfM Nativo (global)

ONDE

É chamado automaticamente quando o mapper incremental (Q4) dispara o detector de collapse (menos de 60 % dos frames de entrada registrados ou taxa de pontos por câmera abaixo de 13). Forçável manualmente apenas via CLI `--sfm-mapper global`. No Inspector, o método global não está acessível por nenhum seletor separado — o app decide sozinho quando trocar.

TÉCNICO

Variante global do pipeline nativo. Primeiro extração de features + matching como em Q4, depois estimativa de pose relativa para todos os pares verificados, em seguida rotation averaging (sincroniza todas as rotações de câmera no sistema mundial) e translation averaging (baseado em LSQR sobre uma formulação esparsa sem matriz, para evitar overflow de inteiro em grandes conjuntos de câmeras). Em princípio escala para ~5 000 câmeras; na prática a qualidade degrada acima de algumas centenas de câmeras — a medição do gate de aceitação Phase-3.8-§5 em K-1351 deu finalLoss 0.07 em vez do objetivo 0.0115. É tratado como „tier fallback“: entra em ação quando o mapper incremental degenera, mas não é ele próprio reavaliado em qualidade.

EM PALAVRAS SIMPLES

O plano B da engine nativa. É chamado automaticamente quando o caminho incremental mais rápido falha. Entrega um resultado utilizável, mas em cenas muito grandes ou difíceis costuma não ser tão preciso quanto o que você obtém do Metashape ou de uma instalação externa do COLMAP. Se Native virar seu workflow padrão, nesses casos vale o desvio pelo import de workspace (Q3 ou Q6).

I Q6 — Import de Workspace Metashape / COLMAP-Text (Fase Q7)

ONDE

Menu File → „Import COLMAP / Metashape Workspace...” (Cmd+⇧+I). Arrastar e soltar uma pasta com `sparse/0/cameras.{bin,txt}` e `images/`.

TÉCNICO

Detecta automaticamente se uma pasta escolhida por arrastar/soltar ou pelo open panel corresponde a um dos três layouts de workspace COLMAP (`sparse/0/`, `sparse/` ou raiz) e se a reconstruction está em binário (`cameras.bin`) ou texto (`cameras.txt`). O caminho binário usa o parser binário COLMAP, o caminho texto usa o loader ETH3D — ambos produzem o mesmo modelo de resultado de SfM e o restante do pipeline (importar imagens, iniciar treinamento MCMC) é agnóstico em relação à origem. As imagens são abertas com `security-scope` via o sistema de bookmarks do sandbox do app, de modo que o import funciona também na versão da App Store. Pensado especificamente para o caso „exportação de Metashape sem recalculer a reconstrução”. A detecção mencionada na entrada do menu File avisa no log do app caso a pasta escolhida não seja um workspace reconhecível.

EM PALAVRAS SIMPLES

A função especial para usuários do Metashape. Se você tem uma licença do Metashape ou Reality-Capture e fez ali a reconstrução de câmeras, é só arrastar a pasta do export para cá e começar o treinamento imediatamente. Em cenas grandes economiza várias horas de cálculo, porque o RadianceKit não precisa fazer o SfM por conta própria.

Qual backend, quando?

Cenário	Backend recomendado
Scan de objeto, 50–200 fotos	Q1 Apple Photogrammetry
Outdoor grande / drone / >500 imagens	Q6 import de workspace (calcular no Metashape ou COLMAP e depois carregar)
Já existe export do Metashape/RealityCapture	Q6 import (sem SfM necessário)
Conjunto acadêmico em texto COLMAP / ETH3D	Q3 import texto COLMAP
Estritamente compatível com App Store + cena orbit	Q4 Native incremental
Q4 falha	Q5 Native global (automático)
Dados de benchmark ETH3D	Q3 (autotest precomputed)

Comparação rápida

Bac- kend	App Store	Sand- box	Binário externo	Melhor uso	Máx ~câ- meras
Q1 Apple PG	✓	✓	—	Objeto-or- bit	~300
Q2 COL- MAP Bi- nary	✗ (só build de dev)	—	colmap/glomap	Outdoor grande	~5 000
Q3 Im- port texto COLMAP	✓	✓	—	Bench rigs	~1 500
Q4 Na- tive in- cremen- tal	✓	✓	—	Objeto-or- bit	~200
Q5 Na- tive glo- bal	✓	✓	—	Fallback do Q4	~1 351
Q6 Im- port de works- pace	✓	✓	—	Reuso de Me- tashape	pela fonte

CAPÍTULO

Capítulo 10 — Modo Iniciante

O Modo Iniciante (Simple Mode, Cmd+1) é o fluxo guiado para quem reconstrói pela primeira vez uma cena 3D Gaussian Splatting. Em vez de mostrar uma sidebar cheia de campos do Inspetor, o app conduz você em quatro passos: primeiro importar imagens ou um vídeo e escolher uma predefinição de qualidade; depois roda o processamento (SfM + treinamento); em seguida você pode inspecionar a cena pronta numa pré-visualização 3D; e por fim exporta no formato desejado. Uma barra de progresso estreita no topo da janela mostra sempre em qual passo você está.

Em comparação com o Expert Mode (Cmd+2), que exibe todos os controles ao mesmo tempo, o Modo Iniciante oculta opções não usadas, traz avisos de validação para imagens insuficientes ou ruins e oferece em cada passo apenas os botões úteis no momento. Você pode alternar entre Iniciante e Expert (Cmd+1 / Cmd+2) a qualquer hora; todo o estado — imagens, predefinição, treinamento em andamento, nuvem pronta — é preservado e fica imediatamente disponível no outro modo.

Z1 — Import (imagens & predefinição)

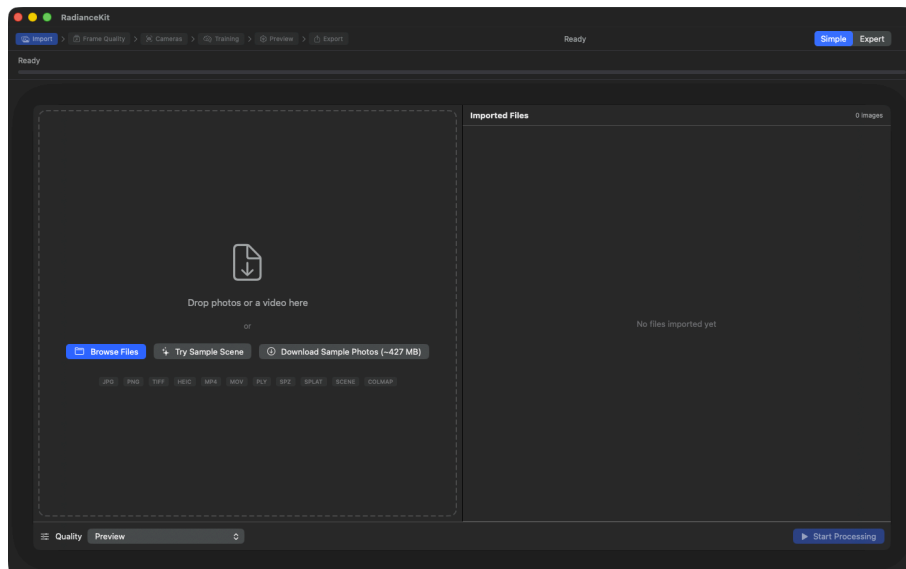


Figura 35: Modo Iniciante passo 1 — drop zone vazia antes do import, breadcrumb no topo (Import → Frame Quality → Cameras → Training → Preview → Export), pills de formato JPG/PNG/TIFF/HEIC/MP4/MOV/PLY/SPZ/SPLAT/SCENE/COLMAP

O QUE APARECE NA IMAGEM Breadcrumb (Import ativo) mostra o fluxo em quatro passos. Drop zone à esquerda com três CTAs: „Browse Files“ (NSOpenPanel), „Try Sample

Scene" (demo bundled), „Download Sample Photos (427 MB)" (subset Mip-NeRF 360 flowers). Pills de formato listam tipos aceitos. À direita „Imported Files" com contador „0 images" e empty state „No files imported yet". Embaixo seletor de qualidade (padrão: Preview) e „Start Processing" (desativado enquanto não há imagens).

O primeiro passo é dar ao app material de imagens. Por drag e drop no campo grande tracejado no centro, pelo „Browse Files" ou clicando na cena de exemplo distribuída com o app. À direita aparece uma lista de todas as imagens importadas com resolução e tamanho; embaixo na barra flutuante você escolhe a predefinição e inicia a pipeline com „Start Processing". Avisos de validação (vermelho com <3 ou <10 imagens, laranja com 10–19) indicam se o app espera reconstrução útil.

C-01 ProgressIndicator (indicador de passo)

ONDE

No topo, acima do fluxo, sempre visível.

TÉCNICO

Mostra uma barra horizontal de progresso de toda a pipeline (Frame Quality → SfM → Training) com alocação por estágio: Frame Quality ocupa 0–5 % (Fase 3.11, muito curta), SfM 0–30 % da barra, Training 30–100 %. Ao lado, texto de status e porcentagem nominada por fase („SfM 41 %", „Training 12 500/20 000"), para o usuário não ler o aparente retrocesso „41 % SfM → 25 % Training" como erro — a barra mostra o progresso total, não o sub-stage. ETA começa quando há velocidade suficiente medida (tipicamente após as primeiras 100 iters). Mesma exibição no Expert Mode acima do Inspecor.

EM PALAVRAS SIMPLES

A barra estreita lá em cima é seu mapa do fluxo. Diz não só o que o app está fazendo (alinhar câmeras, treinar etc.), mas também o quanto já avançou no total. A divisão é proposital: o cálculo de câmeras ocupa o primeiro terço; o treinamento, os outros dois terços — senão pareceria que após SfM voltou ao zero. Pode relaxar; um olhar na barra basta para ver a etapa. O texto ao lado diz se você está em SfM (p. ex. „SfM 41 %") ou Training (p. ex. „Training 12 500/20 000"), para os números não confundirem. Sem ETA: o treinamento é jovem demais — o app estima depois que tem velocidade.

C-03 DropZoneView (drag and drop) **ONDE**

Lado esquerdo do passo de import, retângulo grande tracejado com símbolo. No Modo Iniciante o label é „Drop photos or a video here“.

 **TÉCNICO**

Drop zone que faz o símbolo pular brevemente e colore o fundo quando há drag em cima. Aceita JPG, PNG, TIFF, HEIC, MP4, MOV, PLY, SPZ, .splat, bundles .radiancescene e diretórios. Routing por tipo: imagens coletadas e ordenadas; vídeos dispararam frame sampling; arquivos splat abrem direto a pré-visualização; bundles de cena são lidos. Diretórios são enumerados e todas as imagens importadas. Bookmarks security-scoped (sandbox) são gerenciados. Extensões não suportadas viram banner por 5 s.

 **EM PALAVRAS SIMPLES**

O campo grande tracejado é o controle principal do primeiro passo. Arraste fotos ou um vídeo, ou uma pasta — o app pega tudo que conhece e ignora o resto. Quando o campo fica azul e o símbolo pula, o drag foi reconhecido. Solte e o import começa: imagens vão para a lista, vídeos dispararam o sampling de frames, splats/spz/ply prontos abrem direto a pré-visualização. Se o formato não combinar (p. ex. PDF ou BMP), aparece um aviso no topo — nada é engolido em silêncio.

C-05 Browse Files Button **ONDE**

Dentro da drop zone, botão de destaque.

 **TÉCNICO**

Botão que abre o file dialog do macOS com multi-seleção e tipos JPG, PNG, TIFF, MP4, MOV, pastas e o formato Scene do app. URLs vindas são security-scoped e seguem os mesmos caminhos de drag-and-drop. Se o usuário escolhe uma pasta, enumeração recursiva por imagens.

 **EM PALAVRAS SIMPLES**

Se drag-and-drop for inconveniente, clique este e navegue. Pode escolher vários arquivos (Cmd-clique) ou uma pasta inteira — o app varre recursivamente todas as imagens suportadas. Útil quando capturas estão em subpastas (p. ex. „shoot-day1/“, „shoot-day2/“). O botão faz exatamente o mesmo que o drag-and-drop; use o caminho que preferir.

C-06 Try Sample Scene Button

Dentro da drop zone, visível só se o bundle do app trazer a sample scene E não houver imagens/splats ainda.



Aparece se: (a) `sample-scene.splat`, `.spz` ou `.ply` existe no bundle E (b) sem imagens/vídeos importados E sem nuvem pronta. Ao clicar, carrega a nuvem (prefere o menor formato — splat 3 MB, spz 1.4 MB, fallback ply) e em 400 ms seta valores de câmera hardcoded para uma perspectiva agradável.

EM PALAVRAS SIMPLES

Se você abriu o app pela primeira vez e quer só ver o resultado — clique. Abre uma cena de buquê já treinada para você girar e exportar, sem que o app calcule. A câmera vem em uma perspectiva agradável. Perfeito para testar a controle 3D e o passo de export antes de mexer com suas capturas. Assim que você importar imagens, o botão some — ele só aparece se o projeto está totalmente vazio.

C-07 Download Sample Photos Button

Dentro da drop zone, ao lado de „Try Sample Scene“; mesmas condições de visibilidade.



Dispara download (repo github.com/bkindler/radiancekit-sample-photos) de 427 MB com 960 frames em resolução plena e alimenta o app. Durante o download, botão desabilitado. O progresso entra na barra superior como „Downloading X %“ em estágio próprio, porque mantém escala 0–100 % e não se sobrepõe com SfM.


EM PALAVRAS SIMPLES

Como a sample scene, só que com as fotos originais em vez do resultado pronto. Você roda a pipeline completa por conta e vê quanto SfM e treinamento de fato demoram no seu Mac. O download é grande (cerca de meio DVD = 427 MB), mas só uma vez — depois local; rode quantas predefinições quiser. Durante o download, a barra superior mostra o status. Dica: WiFi rápido ou cabo — 427 MB demora.

C-09 Quality Presets Picker ONDE

Barra inferior flutuante do overlay de import, à esquerda do Start.

 TÉCNICO

Controle com label „Quality“ agrupa predefinições por categoria (Classic / MCMC / Custom). Predefinições built-in agrupadas por categoria; cabeçalhos hardcoded. Custom só visível se há. Locked state: predefinições fora da free list (Quick + Preview) recebem sufixo „“ se o usuário não comprou; ao selecionar, o picker volta para Preview e abre o purchase sheet automaticamente. Selecionar aplica a predefinição e substitui toda a TrainingConfig.

 EM PALAVRAS SIMPLES

Aqui você escolhe quão preciso e quanto tempo o app vai calcular. „Quick“ e „Preview“ funcionam sem comprar e dão um primeiro resultado em poucos minutos — ideal para testar se as imagens fazem sentido. „Balanced“ e „Quality“ exigem versão completa e entregam modelos bem mais limpos, mas em horas em vez de minutos. MCMC é estratégia diferente com menos splats — bom para exportar compacto ou para a web. Predefinições premium têm um cadeado; tocar uma sem licença volta para Preview e abre o purchase. Regra: comece com Preview, veja o resultado, depois decida.

C-10 Start Processing Button ONDE

Barra inferior flutuante do import, à direita do picker de predefinição.

 TÉCNICO

Botão cinza enquanto não há imagens ou vídeo importado. Ao clicar, inicia a pipeline e coloca a stage machine em Frame Quality → SfM → Training. Sem mais status; o processamento em curso aparece em tela separada.

 EM PALAVRAS SIMPLES

O „começar“ botão. Cinza enquanto faltam imagens ou vídeo. Após arrastar fotos, ativa — clique para disparar SfM e Training em sequência. O app conduz o fluxo e você cai automaticamente na tela de processamento (Z2). Sem mais cliques — só após o fim do treinamento o app passa para a pré-visualização (Z3). Se mudar de ideia, dá para Cancelar.

C-11 Video Sampling Slider



ONDE

Lista de imagens à direita; só com vídeo importado.



TÉCNICO

Slider 0.5 fps – 30 fps em passos 0.5. Mudar atualiza densidade de frames e calcula número-alvo (mínimo 10) a partir de densidade e duração. Fora da lista de imagens porque itens de lista bloqueariam eventos do slider. Abaixo, frames calculados („247 frames“) e duração do vídeo („1m23s video“). Tooltip avisa: „Doubling the density doubles the number of frames and increases SfM time by 100 %.“.

EM PALAVRAS SIMPLES

Se importou vídeo em vez de fotos, este slider escolhe quantos frames extrair. Mais frames = melhor qualidade, mas linearmente mais tempo. Para um orbit de 30 s, 5 fps (150 imagens) é bom começo; em 1 min de captura, 3 fps costuma bastar. Sob o slider, o app mostra ao vivo quantos frames sairão — mire na faixa útil 100–300. Se sair ruim, suba; mas dobrar fps dobra SfM.

C-12 Clear All Button



ONDE

Lista de imagens à direita, embaixo à direita; só com imagens importadas.



TÉCNICO

Botão vermelho. Clique abre um diálogo „Clear all imported files?“ com mensagem „N images will be removed“. Confirmar limpa imagens/vídeos, staging, nuvem, status de treinamento, resultado SfM e caches; stage volta a Import. Em Cancel, nada muda. Diálogo configurado como caminho padrão não-destrutivo (destrutivo em vermelho).

EM PALAVRAS SIMPLES

Para começar do zero. A confirmação aparece porque apaga todos os imports atuais e qualquer câmera/treino já calculado — sem desfazer. Útil quando quer trocar tudo de material ou se livrar de projeto velho. Para tirar uma imagem isolada, use a lista à direita (próximo item), não esse botão. Os arquivos no disco não são removidos — o app só esquece referências.

C-13 Lista de arquivos ForEach (remover imagem individual)**ONDE**

Lista de imagens à direita, cada entrada.

**TÉCNICO**

Lista das imagens importadas com swipe-to-delete. Por imagem: ícone, nome, resolução („1920 x 1080”) e tamanho (KB/MB). Resolução vem de um cache de metadados preenchido assíncrono dos headers para não bloquear UI. Apagar oferece swipe (trackpad esquerda) e Delete na seleção. Aviso: a remoção estendida com botão de menos explícito, Backspace e Cmd-Z foi adicionada *apenas no Expert Mode* (Project Navigator); no Modo Iniciante segue swipe.

EM PALAVRAS SIMPLES

A lista à direita mostra cada imagem com resolução e tamanho — útil para perceber misturas de alta e baixa resolução. Para tirar uma imagem, deslize com dois dedos para a esquerda no trackpad — como no iOS Mail — ou selecione e aperte Delete. O arquivo no disco não é apagado; só sai do projeto. Se precisa de botão de menos ou Cmd-Z, vá ao Expert Mode (Cmd+2). No Modo Iniciante fica o padrão simples de swipe.

C-15 Avisos de validação (3 níveis)**ONDE**

Sob a lista de imagens, acima do Clear All.

**TÉCNICO**

Três limiares por contagem de imagens (só com imagens e sem vídeo):

- < 3: banner vermelho (red octagon), texto „At least 3

images are required. Camera alignment cannot be computed from fewer images.”

- 3–9: banner vermelho, „With fewer than 10 images, SfM often

fails and the trained scene tends to overfit [...].

15–20 minimum recommended; 30+ for object captures.”

- 10–19: banner laranja (warning triangle), „Workable, but

quality usually improves with 20+ images and good coverage around the scene.”

A partir de 20, o banner some. Limiares hard-coded com base em 560+ experimentos empíricos.

EM PALAVRAS SIMPLES

O app olha quantas imagens você tem e dá um diagnóstico colorido. Vermelho: provavelmente não dá — ou SfM não calcula câmeras ou o treinamento overfita por pouco material. Laranja: pode dar, mas sem topo de qualidade. Sem banner: bom material. Para modelos muito limpos, mire em pelo menos 30–50 fotos distribuídas; em externas ou ambientes grandes, bem mais. Pode iniciar com aviso, mas não se surpreenda se SfM falhar ou o modelo ficar furado.

C-16 Detecção de COLMAP Workspace**ONDE**

Ao soltar uma pasta — sem botão visível, lógica de detecção.

**TÉCNICO**

Ao soltar diretório, checa um de três layouts: `sparse/0/cameras.bin`, `sparse/cameras.bin` ou `cameras.bin` na raiz. Se sim, a enumeração padrão é cancelada e um alerta modal pergunta se usa a reconstrução existente ou refaz por Apple Photogrammetry. Mesma lógica para text format (`cameras.txt`) e exports ETH3D. Ver Capítulo 9 backend Q6. Funciona igual em Iniciante e Expert.

**EM PALAVRAS SIMPLES**

Se você já calculou câmeras em Metashape, RealityCapture ou COLMAP, basta arrastar a pasta. O RadianceKit detecta automaticamente (`sparse/0/`, `cameras.bin` etc.) e pergunta se reusa ou refaz. Reusar economiza horas em cenas grandes — o treinamento começa direto. Text format (`cameras.txt`) e ETH3D também são detectados. Disponível tanto no Iniciante quanto no Expert; detalhes no Capítulo 9 backend Q6.

Quando passar à próxima etapa?

Você pode clicar Start Processing assim que (a) pelo menos uma imagem ou vídeo está importado E (b) o banner está laranja ou sumiu. Com banner vermelho o app deixa começar, mas é provável que você cancele logo. Recomendado: pelo menos 20 imagens nítidas, com overlap claro entre adjacentes, de distâncias parecidas ao motivo. Escolha antes a predefinição: com 30 imagens e Quick, sai em minutos; com Quality, leva 1–2 horas.

Z2 — Processamento (SfM + treinamento)

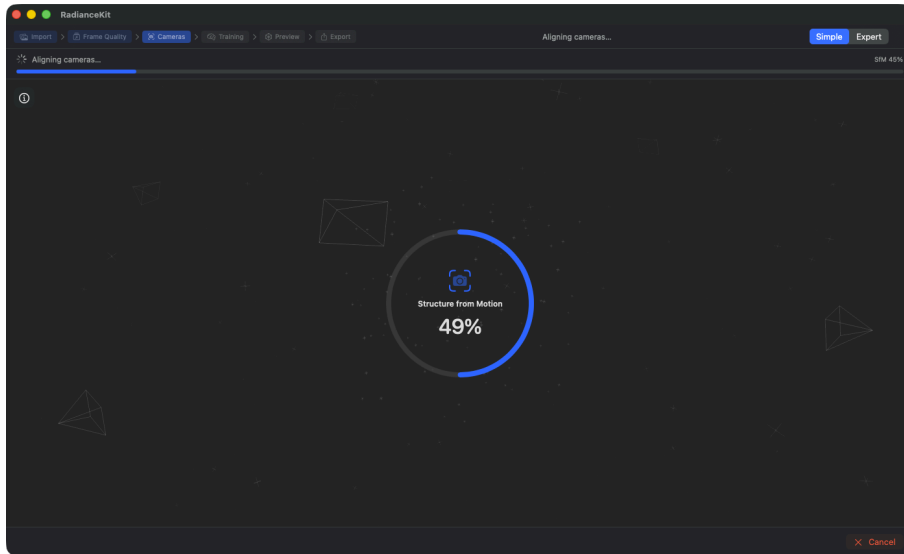


Figura 36: Z2 fase SfM — ícone „Structure from Motion“ com 41 % no círculo grande, barra de status superior em „SfM 25 %“, botão Cancel embaixo à direita

Fase SfM (câmeras se alinham): o grande círculo de progresso mostra o sub-stage (41 % da sessão Apple Photogrammetry). Texto „Aligning cameras...“ acima à esquerda. Breadcrumb marca „Cameras“ como ativo. Barra superior mostra total da pipeline (25 %) — SfM ocupa a primeira metade. Câmeras wireframe flutuantes ao fundo sugerem estimação de poses.

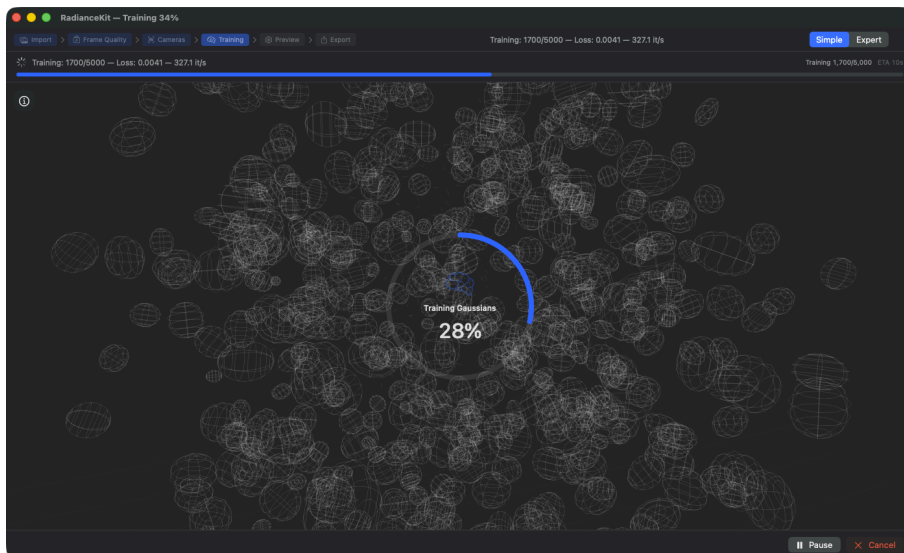


Figura 37: Z2 fase de treinamento — ícone „Training Gaussians“ com 6 %, métricas ao vivo no topo (Training: 400/5000 — Loss: 0.1642 — 138.7 it/s), ETA 33 s, Pause/Cancel embaixo

Fase Treinamento (Gaussians otimizados): sub-stage muda para „Training Gaussians“; % conta iters do preset (aqui 400/5 000 para Preview = 8 % do stage). Linha de métricas mostra Loss (0.1642), iters por segundo (138.7 it/s) e ETA (33 s). Total da pipeline sobe de 50 % a 100 %. Botão Pause (em vez de só Cancel da fase SfM) permite Resume depois; Cancel descarta o resultado e volta a Z1.

Quando a pipeline roda, o app esconde o overlay de import e mostra uma tela cheia de processamento. No centro, círculo grande (220×220 px) com ícone de stage, texto e %; ao fundo, animação discreta de splats sugere computação. À esquerda em cima, painel info com métricas ao vivo de treinamento e SfM. Embaixo: Pause/Resume, Cancel e, em erro, Retry.

C-18 SplatTrainingView (animação de fundo)

ONDE

Fundo completo atrás do círculo, oculto em cancel ou falha.

TÉCNICO

Animação decorativa que, conforme o progresso (0...1), renderiza um número crescente de partículas splat. A fonte é um valor calculado de progresso: SfM mapeado em 0–0.2, treinamento em 0.2–1.0 (Frame Quality em 0–0.05). Splats „crescem“ visivelmente durante o treinamento. Apenas decorativa — não mostra resultados reais (isso seria Live Preview no Expert). Em cancel/erro some, fica só o círculo.

EM PALAVRAS SIMPLES

Ao fundo roda uma animação de pontos dançantes, para a tela não ficar vazia durante o cálculo. Não é seu modelo — esse aparece em Z3. Mas tem a mesma „densidade“ visual, dando uma ideia do avanço do treinamento. No começo poucos pontos; no fim, denso. Indicador visual extra ao lado do %. Se atrapalha (querendo trabalhar ao lado), troque para Expert, onde some.

C-19 Círculo grande de progresso

ONDE

Centro da tela de processamento, 220×220 px.

TÉCNICO

Dois anéis sobrepostos: anel de track suave fora; anel de progresso com acento ou vermelho dentro (vermelho em erro). No meio: ícone de stage (cérebro para Training, câmera para SfM, filme para extração de frames, sparkles para Frame Quality), título e % animada em 32 pt rounded. Ícone pulsa enquanto ativo. Interpolação a 30 Hz suaviza em direção ao progresso real, com creep constante (0.0003/frame) + parte proporcional (4 % do gap) e soft ceiling em 80 % do próximo milestone esperado (para SfM de tabela hardcoded). Sensação suave mesmo se updates reais vêm a cada poucos segundos.

EM PALAVRAS SIMPLES

O círculo grande no centro é o display principal durante o cálculo. Enche suavemente mesmo quando os updates reais demoram — dá a sensação de „algo acontece“ em vez de ver % travada. O símbolo dentro varia: filme (extrair frames), câmera (alinhar), cérebro (treinar). A % é da sub-fase atual; o total da pipeline está na barra fina acima. Em erro, o anel fica vermelho e o ícone para de pulsar.

C-22 Botão Info (exibe métricas)

ONDE

Acima à esquerda na tela, 32×32 px.



TÉCNICO

Botão simples com material. Alterna o painel info. Ícone muda entre info-outline e info-filled quando ativo. Animação suave. Tooltip „Show detailed processing metrics“.

EM PALAVRAS SIMPLES

A tela é deliberadamente limpa — só o círculo grande, mais nada. Se você quer ver detalhes (iter, loss, contagem de Gaussians), clique no ícone „i“ no canto. Um painel abaixo desdobra com valores ao vivo. Clicar de novo oculta. A configuração não persiste — a cada novo run o painel volta oculto, de propósito para não assustar iniciantes.

C-23 Painel Info (métricas ao vivo)

ONDE

Embaixo à esquerda na tela, só com `showProcessingInfo == true`.



TÉCNICO

Painel em duas colunas com material ultra-thin. Coluna esquerda: linhas info por estágio — para SfM status e %; para Training iteração, loss combinado, L1, D-SSIM, Gaussian count (laranja), speed (it/s), elapsed, ETA, SH degree e LR. Coluna direita: status, time info, mini loss chart inline (ver C-28) e nudge de discoverability (ver C-32). Valores vêm do status de treinamento atualizado a cada tick.

EM PALAVRAS SIMPLES

O painel mostra os valores ao vivo que no Expert ficariam permanentemente na sidebar do Inspetor: iteração, loss (menor = melhor), número de Gaussians, velocidade, restante, SH degree e LR. À direita roda uma micro-curva de loss — num olhar você vê se o treinamento vai na direção certa. Se o treinamento parece travado, olhe aqui — loss que não cai mais, ETA que não baixa = problema. Loss explodindo ou NaN = treinamento instável: Cancel + Retry ou outra predefinição.

C-25 Botão Pause/Resume **ONDE**

Barra inferior, só durante Training (NÃO durante SfM) e enquanto processamento roda.

 **TÉCNICO**

Botão bordered. Chama Pause ou Resume. Label alterna entre „Pause“ (ícone pause) e „Resume“ (play). Em SfM, não exibido — Apple Photogrammetry não tem semântica de pause. Pause preserva iteração, status de Gaussians e momento Adam inteiro — Resume continua exatamente de onde parou.

 **EM PALAVRAS SIMPLES**

Enquanto treina, você pode pausar e continuar depois. Útil quando precisar do GPU para outra coisa pesada — edição de vídeo, jogo, render em outro app. Pause, faça o que precisa, Resume, treinamento segue exatamente. Iter, contagem e momento Adam preservados; pause não custa qualidade. Na fase SfM, Pause não aparece — Apple Photogrammetry não pausa; use Cancel se necessário.

C-26 Botão Cancel **ONDE**

Barra inferior, durante processamento (SfM ou Training).

 **TÉCNICO**

Botão bordered vermelho. Abre diálogo „Stop and discard progress?“ com „Discard Progress“ (destrutivo) e „Keep Running“ (cancel). Confirmar seta flag de cancel, termina a task, mata o subprocesso SfM se necessário e grava summary com cancel no log JSONL. Diferente de Pause, buffers e status são descartados.

 **EM PALAVRAS SIMPLES**

O botão de aborto. Diferente de Pause é definitivo — ao recomençar, tudo zera. Útil quando você errou a predefinição, o treinamento está lento demais ou produz lixo. O app pergunta antes para não perder horas por acidente. Para pausar só um pouquinho, use Pause.

C-27 Botão Retry **ONDE**

Barra inferior, se a pipeline falhou (status SfM começa com „SfM failed“ ou Training em erro).

 **TÉCNICO**

Botão de acento. Reinicia toda a pipeline. Checa imagens importadas antes. Logs de erro anteriores permanecem; novo run grava novo JSONL com timestamp atual.

 **EM PALAVRAS SIMPLES**

Se SfM ou Training falham com erro, tente de novo. Às vezes ajuda — muitos passos (RANSAC, densification) têm aleatoriedade. Toda a pipeline roda do zero em novo log. Se o segundo run também falha, em geral as imagens são o problema (poucas, pouco overlap, motion blur, luz ruim); use Back e troque o material. Dica: veja logs em paralelo (Help → Open Training Logs) — ali está mais detalhado onde travou.

C-28 Loss chart inline **ONDE**

No painel info, coluna direita; só durante treinamento com histórico não vazio.

 **TÉCNICO**

Área de desenho compacta (40 px alta), desenha loss history como linha de 1 px em cor de destaque. Filtra para finitos (proteção NaN). Min/Max sobre toda história — auto-zoom. Último valor de loss acima à direita do chart. O history é preenchido a cada tick (100 iters).

 **EM PALAVRAS SIMPLES**

Uma micro-curva de loss que num olhar mostra se o treinamento „converge“ (linha cai à direita) ou está travado ou explode (linha plana ou sobe). Em treinamento saudável, cai íngreme no início e achata — como meia exponencial. Auto-zoom para mostrar até melhorias pequenas no fim. Se a linha sobe ou trava, algo está errado — material problemático ou outra predefinição melhor. No painel info que abre com o „i“.

C-32 Nudge de descoberta (dica do Expert Mode)**ONDE**

No painel info, coluna direita embaixo; só durante treinamento E no Modo Iniciante.

TÉCNICO

Linha pequena com ícone de olho e caption „Switch to Expert Mode (⌘2) for live splat preview“, em tonalidade discreta e 10 pt. Não é interativo — o usuário aperta Cmd+2 ou o menu Mode → Expert Mode.

EM PALAVRAS SIMPLES

Aviso discreto: no Expert, durante o treinamento, a versão atual do modelo 3D já fica visível no viewport. No Modo Iniciante, deliberadamente oculta para manter a UI calma — mas muitos não sabem que existe; daí o aviso. Pressione Cmd+2 e o treinamento segue em background enquanto você vê o modelo se montar diante dos seus olhos. Bom para após alguns milhares de iters julgar se vai dar certo. Cmd+1 volta ao Iniciante.

Quando passar à próxima etapa?

O app passa automaticamente para Z3 (preview) ao terminar o treinamento com sucesso — nada a clicar. A barra inferior troca de Pause/Cancel para Back (volta ao import) e Export (vai para Export). Em erro (mensagem vermelha, ícone X), aparece Retry; decida se tenta de novo ou volta ao import com Back para trocar material.

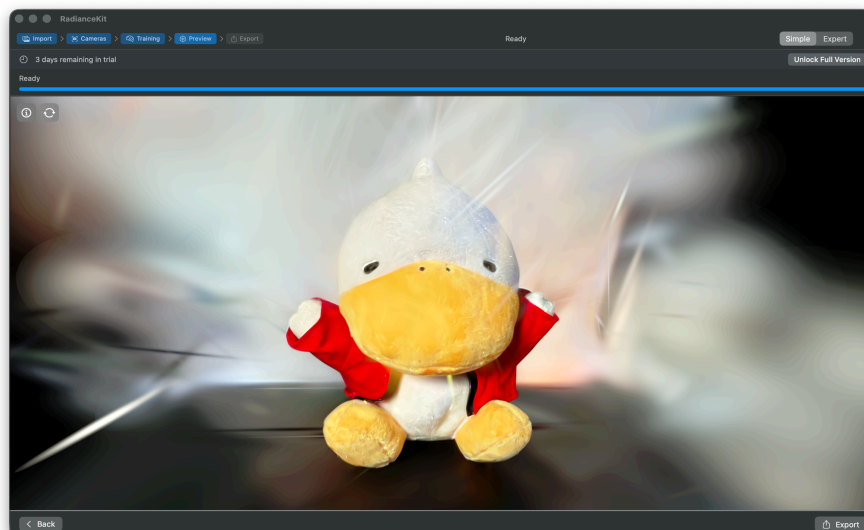
Z3 — Pré-visualização (girar o modelo 3D)

Figura 38: Modo Iniciante passo Preview com viewer 3D

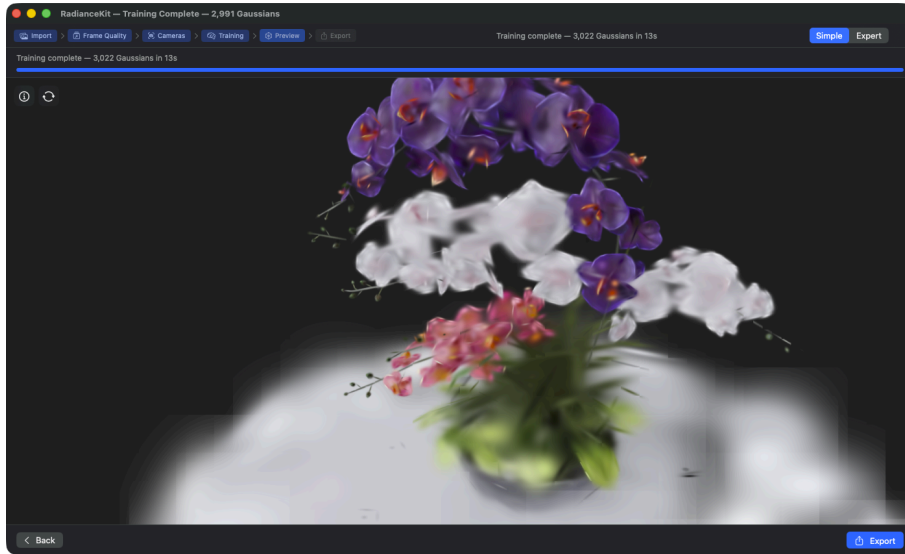


Figura 39: Z3 Preview após conclusão — buquê reconstruído, cabeçalho „Training complete — 3 022 Gaussians in 13 s”, botões Back e Export embaixo

O QUE APARECE NA IMAGEM Breadcrumb marca „Preview” como ativo. Viewport 3D em tela cheia renderiza a cena buquê pronta (set de teste sintético em Blender, subset de 60 frames de 960 hemisféricos). Cabeçalho: „Training complete — 3 022 Gaussians in 13 s” (número final + tempo). Drag rotaciona (yaw/pitch); scroll dá zoom. Botão Back (embaixo à esquerda) volta a Z2 para resume ou re-run; Export (embaixo à direita, primário) leva a Z4.

Após o treinamento, o app cai automaticamente em preview. Aqui você vê o modelo Gaussian Splatting pronto em tela cheia Metal e gira/dá zoom/pan com mouse e trackpad. No topo do viewport fica um pequeno overlay com controle de câmera e info — rotação automática, estatísticas, reset. Antes do próximo passo (Export), vale verificar o modelo em vários ângulos para garantir reconstrução limpa.

C-36 SplatViewportView (vista 3D principal)

Fundo cheio do passo preview.



Viewport 3D Metal que renderiza a nuvem. O renderer é o ForwardPass PRÓPRIO do RadianceKit — o mesmo que já renderiza os splats durante o treinamento — portanto é WYSIWYG de verdade (o que é treinado é exibido e exportado exatamente igual). Pipeline tile-based com order-independent transparency. Se o renderer não inicializa (p. ex. Metal indisponível), aparece fundo preto com „Metal not available“. A vista ignora safe area, modelo vai até a borda.

EM PALAVRAS SIMPLES

O viewport principal. Aqui você vê seu modelo 3D reconstruído de suas fotos, renderizado em tempo real na GPU. Click+drag esquerdo para girar. Scroll ou gesto de trackpad de dois dedos para zoom. Direito ou Cmd+drag para pan. O modelo são dezenas de milhares de elipsoides 3D semi-transparentes („Gaussians“) que reconstituem a cena fotorrealisticamente — cada um com posição, orientação, forma e cor aprendidos. Caso raro: Mac sem suporte a Metal — fundo preto e aviso; RadianceKit precisa de GPU Metal.

C-37 CameraControlsOverlay (overlay de controle)

Acima do viewport, flutuante.



Overlay compacto com botões para auto-rotation (turntable), reset camera, escolha de background (Gray/Black/White), save screenshot, toggle info. Liga aos parâmetros de câmera (distância, azimuth, elevation, target, FOV) e controla o turntable. Durante o treinamento (caso Expert com viewport rodando ao vivo) o overlay mostra também linha de status compacta.

EM PALAVRAS SIMPLES

A barrinha flutuante sobre o modelo. Aqui você inicia a auto-rotação (modelo gira sozinho, bom para screenshots e demos curtas), reseta a câmera (se se perdeu), troca o fundo (cinza neutro, preto para contraste, branco para modelos claros), tira screenshots direto (salvos em / Pictures). Útil para mostrar um detalhe específico sem exportar o modelo inteiro. A auto-rotação também testa se o modelo está bom de todos os lados ou tem um „lado feio“ por falta de capturas.

C-38 Botão Export (barra de nav)



Barra inferior em Z3.



Botão de acento com label „Export“ e ícone de share. Clique muda para Z4. Antes, a view-pai checa se a versão completa está desbloqueada — senão, em vez do palco de export, aparece a tela de bloqueio (ver U-06).

EM PALAVRAS SIMPLES

Se você gostou do resultado, clique Export e cai no último passo: escolha formato e salve. Sem versão completa, em vez disso vê uma tela de bloqueio com „Unlock Full Version“ — o app não força a venda, mas o export é Premium. Após comprar, segue desbloqueado direto para o export. Se mudou de ideia, Back volta à preview para continuar girando.

Quando passar à próxima etapa?

Antes de exportar, gire o modelo todo e veja: todas as áreas que você cobriu nas fotos estão presentes? Há „floaters“ flutuando no ar? O fundo/céu está limpo ou borrado? Problemas grandes só se corrigem com novo treinamento — mais fotos, outra predefinição ou Expert com settings de floater reduction.

Z4 — Export (escolher formato & salvar)

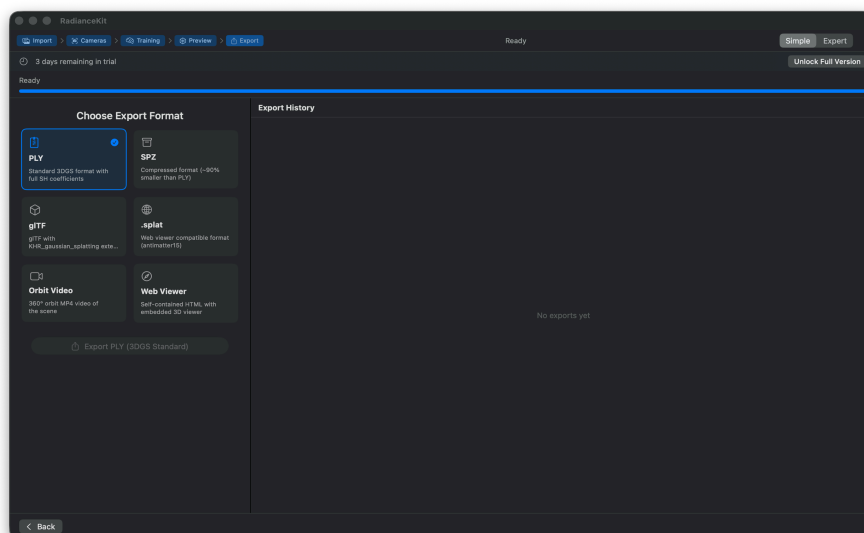


Figura 40: Modo Iniciante passo Export com cards de formato

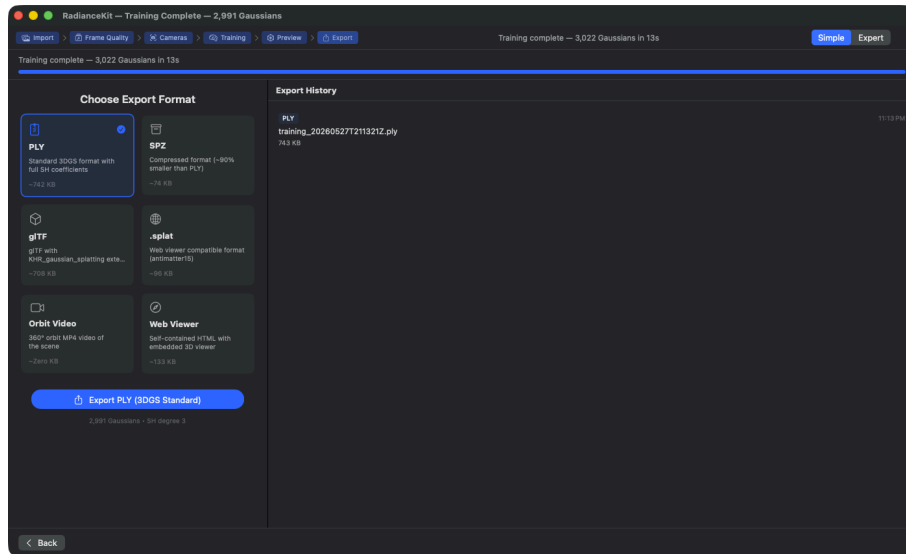


Figura 41: 24 cards de Export — 6 formatos (PLY 742 KB selecionado, SPZ 74 KB, glTF 708 KB, .splat 96 KB, Orbit Video, Web Viewer 133 KB), sidebar Export History à direita com PLY já salvo

O QUE APARECE NA IMAGEM Breadcrumb marca „Export“ como ativo. Grid esquerdo „Choose Export Format“ com seis opções: PLY (3DGS padrão, 742 KB, SH completos — aqui pré-selecionado com check azul), SPZ (3DGS compactado, 90 % menor, 74 KB), glTF (com extensão `KHR_gaussian_splatting`, 708 KB), .splat (Web Viewer via `antimatter15`, 96 KB), Orbit Video (MP4 360°, tamanho calculado ao vivo), Web Viewer (HTML autônomo com viewer embutido, 133 KB). Tamanhos calculados ao vivo a partir do Gaussian count e overhead de formato. À direita „Export History“ lista exports concluídos com pill de formato, nome de arquivo e timestamp — clique revela no Finder. CTA primário embaixo: „Export PLY (3DGS Standard)“ com subtítulo „2,991 Gaussians · SH degree 3“.

No último passo você escolhe entre 6 formatos (PLY, SPZ, glTF, .splat, orbit video, web viewer) num grid de 2 colunas, clica Export e escolhe o destino no diálogo macOS. À direita roda a history de exports — ao selecionar cada card aparece o tamanho estimado, ajudando, p. ex., a preferir SPZ para web (pequeno) e PLY para outros softwares (SuperSplat, Postshot, Blender via plugin, grande e completo).

C-39 Grid de formatos em 2 colunas

ONDE

Lado esquerdo principal do passo Export.



TÉCNICO

Grid de cards em duas colunas flex com 12 pt de espaço. Itera sobre os formatos oferecidos no Iniciante — subconjunto filtrado dos 6 mais importantes: PLY, SPZ, glTF, .splat, Orbit Video, Web Viewer. Compressed PLY e SOG estão SÓ no Expert.

EM PALAVRAS SIMPLES

Um grid com os 6 formatos relevantes no Iniciante: PLY (padrão para outras ferramentas 3D), SPZ (compactado para web), glTF (padrão oficial Web3D), .splat (viewer antimatter15), Orbit Video (MP4 pronto) e Web Viewer (HTML autônomo). Cobre 90 % dos casos. Se precisa de um menos usual (Compressed PLY ou SOG para compressão extrema), Expert Mode tem os 8. Aqui mais compacto, proposital, para iniciantes não se perderem.

C-40 Botão de card de formato

ONDE

Cada card no grid.



TÉCNICO

Botão simples com layout em card: ícone (p. ex. zipper para PLY, archive box para SPZ, vídeo para Orbit Video) em cima, nome em headline, caption descritiva (2 linhas), embaixo tamanho estimado (live a partir de formato, Gaussian count e SH degree em KB/MB). Clique seleciona; card selecionado fica com acento, borda e check à direita. Tooltip = descrição.

EM PALAVRAS SIMPLES

Um card por formato. Clique para selecionar — destacado com cor de acento e check; o botão Export abaixo ajusta o texto („Export PLY“, „Export SPZ“ etc.). Cada card tem ícone, nome, mini-descrição de 2 linhas e tamanho estimado no seu resultado atual. O tamanho ajuda a decidir — mandar por email pede a variante menor (em geral SPZ ou .splat); continuar em outro 3D pede PLY. Hover mostra descrição maior no tooltip se você não tem certeza da diferença.

C-41 Slider de duração de vídeo

ONDE

Sob o grid, só com formato de vídeo (Orbit Video ou Social Video).



TÉCNICO

Slider 3–30 s em passos 1, ligado ao comprimento de vídeo no estado. Largura máxima 300 px. Só aparece com vídeo selecionado. Em formatos não-vídeo, sumir totalmente — sem espaço morto.

EM PALAVRAS SIMPLES

Se escolher Orbit Video, define a duração aqui. 3 s = rotação muito rápida, 30 s = devagar e calma. Para reels (Instagram, TikTok), 6–10 s é ideal — mostra mas não cansa. Para apresentações ou portfólio, 15–20 s ok. O slider só aparece com vídeo selecionado; em PLY/SPZ ficaria sem sentido.

C-42 Botão Export

ONDE

Sob o grid (e sob o slider de duração, se vídeo).



TÉCNICO

Botão grande com cor de acento. Label: „Export {nome do formato}”, ícone share. Clique abre diálogo de salvar do macOS com extensão correspondente e nome padrão „scene.{ext}”; ao confirmar, grava no URL escolhido. Desativado se não há resultado de treinamento ou export já roda.

EM PALAVRAS SIMPLES

Clique, escolha onde salvar no diálogo macOS, pronto — o app grava no formato escolhido. Nome padrão é „scene.{ext}” (p. ex. „scene.ply” ou „scene.spz”); pode trocar no diálogo antes de salvar. Cinza enquanto não há resultado (não deveria acontecer aqui, já estaria em Export) ou outro export roda. Durante o export, barra de progresso aparece embaixo; o app fica usável e você pode preparar o próximo.

C-43 Barra de progresso do export

ONDE

Sob o botão Export, só durante export.



TÉCNICO

Progresso com largura máx 300 px, abaixo caption „Exporting... N %”. Valor 0 a 1, atualizado durante gravação — em PLY em chunks de 10 000 Gaussians, em SPZ após quantização, em Orbit Video por intervalos de frame.

EM PALAVRAS SIMPLES

Durante o export você vê o progresso como barra estreita + %. PLY normalmente em segundos — gravação binária direta. SPZ um pouco mais — quantização e compressão. Orbit Video é o mais demorado — cada frame renderizado de novo; conforme resolução e duração, pode levar um minuto ou mais. App fica usável durante o export; já pode preparar o próximo formato ou clicar no viewport.

C-44 Display de erro de export ONDE

Sob a barra, só se o último export deu erro.

 TÉCNICO

Linha vermelha com warning icon e texto do erro. Fundo vermelho com 8 % opacity, cantos arredondados. Larg máx 400 px. Causas comuns: SOG esperando `cwebp` no PATH (não sandbox); falha de gravação com disco cheio; sandbox recusando destino fora da área permitida.

 EM PALAVRAS SIMPLES

Se o export falha, aqui aparece em vermelho uma descrição curta. Geralmente óbvia — disco cheio, sem permissão na pasta, destino fora do sandbox. SOG especificamente falha sem `cwebp` ; aí use SPZ. Se a mensagem não está clara, veja o log (Help → Open Training Logs), bem mais detalhado. Em dúvida, salve em outra pasta — p. ex. Desktop.

C-46 Lista Export History ONDE

Lado direito do passo Export.

 TÉCNICO

Lista da history (persistida como JSON em UserDefaults, mantida após cada export bem-sucedido). Linha mostra badge de formato (pequeno, acento), timestamp (HH:mm), nome de arquivo (1 linha) e tamanho formatado. Clique abre Finder com arquivo selecionado. Empty state „No exports yet“.

 EM PALAVRAS SIMPLES

Lista dos exports anteriores — formato, hora, nome, tamanho, em ordem cronológica. Clique numa linha e o arquivo é revelado no Finder, sem você navegar. Útil para achar uma hora depois sem lembrar onde salvou — a history guarda. Se nunca exportou nada, aviso amigável „No exports yet“. A lista sobrevive a reinícios do app porque está em UserDefaults.

C-48 Menu de contexto da History (clique direito) ONDE

Clique direito numa linha da history.

 TÉCNICO

Menu de contexto por linha com duas ações: „Reveal in Finder“ (igual ao clique simples) e „Copy Path“ (coloca o caminho completo na área de transferência). O segundo é útil para drag-and-drop em outros apps ou para terminal.

 EM PALAVRAS SIMPLES

Clique direito numa linha da history abre um mini-menu com duas ações. „Reveal in Finder“ faz o mesmo que clique normal — abre o Finder com arquivo selecionado. „Copy Path“ coloca o caminho completo na área de transferência — útil para terminal, outros apps ou notas. Particularmente prático para encaixar o export ou abrir em programa que pede caminho. Pequeno detalhe que segue padrões Mac.

Quando o fluxo termina?

Após um export bem-sucedido, você tem o modelo 3D no disco e a history mostra nova entrada. Não há botão „Done“ — pode anexar quantos exports quiser em formatos diferentes, sem retreinar. Para voltar à preview (p. ex. ver outro ângulo), use Back na barra inferior. Para iniciar cena totalmente nova, Back até Z1 e Clear All, ou File → New Project (Cmd+⇧+N).

Mudança para o Expert Mode

Aperte Cmd+2 a qualquer momento ou escolha Mode → Expert Mode (M8). Todo o estado é preservado: imagens importadas, predefinição, treinamento em andamento ou pronto, nuvem, history de export, até o estágio atual. No Expert, em vez do palco de 4 passos, aparece a sidebar de Inspetor completa com 150 controles. Em particular: o Project Navigator (Capítulo 2) traz operações estendidas de imagem (botão de menos, Backspace delete, Cmd-Z undo, quick look), live preview no viewport durante treinamento, e todos os parâmetros de loss, MCMC, densification e Mip Splatting. Cmd+1 volta ao Iniciante — também sem perder estado.

Perguntas frequentes**Por que meu Start Processing fica cinza?**

Você não importou imagem nem vídeo. Arraste pelo menos um arquivo na drop zone ou use „Browse Files“. Quando a lista direita tem ao menos um item, o botão ativa. (Com 1–2 imagens começa, mas SfM aborta com erro — ver banner vermelho.)

Por que meu Export Button está bloqueado?

No Iniciante há dois níveis: (a) sem pipeline de treinamento terminada e sem , o botão está desativado — precisa terminar Z2. (b) sem versão completa comprada (`PurchaseManager.hasAccess == false`), em vez do palco de export aparece tela de bloqueio com cadeado e „Unlock Full Version“ que abre o purchase sheet. Quick e Preview permitem treinamento grátis, mas export é premium.

Por que não consigo escolher uma predefinição?

Pode escolher — mas se tocar uma premium (Balanced, Quality, variantes MCMC) sem versão completa, o picker volta a Preview e abre o purchase. Quick e Preview são as únicas grátis.

Por que minha drop zone fica vazia e cinza-tracejada mesmo com imagens arrastadas?

Provavelmente UTI mismatch. O app aceita JPG, PNG, TIFF, HEIC, MP4, MOV mais os splats próprios. Outros (BMP, GIF, WebP, RAW) NÃO são reconhecidos. Se tem certeza que seu tipo está na lista, confira a extensão — o app vai primeiro pela extensão, não pelo conteúdo.

Por que SfM demora tanto mesmo com só 30 imagens?

Apple Photogrammetry não escala linear — em algumas constelações (internos com texturas complexas, motion blur, luz ruim), demora bem mais que a contagem sugere. Se SfM trava 10+ minutos em 30 imagens, cancele e tente com material melhor, ou vá ao Expert e tente COLMAP/Native SfM (Cmd+2 → Inspetor → Camera Alignment).

Onde estão meus training logs?

Help → Open Training Logs (Cmd+⇧+L). Abre ~/Documents/RadianceKit/Logs/. Cada sessão grava um JSONL próprio com timestamp — primeira linha é a configuração, depois linha de progresso a cada 100 iters, última linha é summary com final loss e success flag.



COLOFÃO

*Composto em SF Pro · Código em SF Mono ·
Typst 0.14 · 22. June 2026*

© 2026 Bjoern Kindler · Bischofshofener Str. 9, 82008 Unterhaching, Alemanha

Made with ❤️ in Unterhaching