



RADIANCEKIT

Brugervejledning

Fotorealistisk 3D-rekonstruktion
via Gaussian Splatting

Version 1.5.0 · macOS 26.0+ · Maj 2026

BJOERN KINDLER · KINDLER-DEV.DE

Oversigt

Introduktion — Hvad du bør vide	3
Hvad er RadianceKit?	3
Hvad er Gaussian Splatting?	3
Kapitel 1 — Menulinje	5
File-menuen	5
Mode-menuen	9
Training-menuen	10
Viewport-menuen	14
Export-menuen	19
Help-menuen	24
Note: Cmd-Z i Edit-menuen	28
Tastaturgenveje i oversigt	29
Kapitel 2 — Inspector (Expert View)	30
Look-sektion (L1–L5)	33
Presets-sektion (I1–I11)	36
Trænings-konfigurations-sektion (I12–I22)	42
Enhancements-sektion (I26–I29, I42–I44)	48
Metrik-sektion (I30–I38)	55
Loss-diagram-sektion (I39–I41)	61
Hvornår række ud efter Inspektoren?	64
Kapitel 3 — Indstillinger	66
General-fanen	67
AI-Helpers-fanen	72
Inspector-spejls-settings	75
Hvornår hvad?	76
Kapitel 4 — Hjælpevinduer	77
User Guide (W1–W4)	78
Keyboard Shortcuts (W5–W6)	81
Manage Storage (W7–W12)	83
Pareto Dashboard (W13–W22)	86
Holdout Analysis (W23–W29)	93
BayesOpt Console (W30–W39)	98
Hovedvindue: loss-forløb og gaussian-count (I39–I41, krydshenvisning)	104
Tommelfingerregel-boks	105
Kapitel 6 — Trænings-konfiguration	106
Iteration (T1–T2)	108
Learning rates (T3–T10)	110

Densification — Classic (T11–T16)	115
Loss (T17–T20)	119
SH-degree-progression (T21)	122
Performance (T22–T25)	123
Diagnose og punktsky-forberedelse (T26–T30)	125
Regularisering (T31–T37)	128
Refinement (T38–T44)	131
Sky-dome (T45–T48)	135
Adam + LR-schedule (T49–T55)	137
Post-processing + Apple AI (T56–T60)	140
MCMC-densification (T61–T73)	143
Mip-splatting (Q1.5) (T74–T76)	149
Adaptive densification (Q5) (T77–T79)	152
Curriculum (Q6) (T80–T81)	153
Statistiske presets (TP1–TP9)	154
Metoden:	157
Hvilket felt til hvad? (cheat sheet)	158
Farlige felter	159
Kapitel 7 — Indbyggede kvalitets-forudindstillinger	160
Hvornår hvilken preset?	170
Hurtig sammenligning	171
Egne forudindstillinger	173
Kapitel 8 — Eksportformater	174
Hvilket format hvornår?	187
Hurtig sammenligning	188
Kapitel 9 — SfM-backends	189
Hvilken backend hvornår?	194
Hurtig sammenligning	195
Kapitel 10 — Begyndertilstand	196
Z1 — Import (vælg billeder & preset)	196
Z2 — Behandling (SfM + træning)	204
Z3 — Forhåndsvisning (drej 3D-modellen)	210
Z4 — Eksport (vælg format & gem)	213
Skift til Expert Mode	218
Ofte stillede spørgsmål	218

Sådan læser du denne vejledning

Hvert opslag i vejledningen følger samme skema. I venstre side står betjeningsstierne og de tekniske detaljer; til højre i en varm sidekolonne finder du altid den enkle forklaring. Små ikoner i starten af hver linje fortæller dig på et øjeblik, hvilken slags information der følger nu.

DE FIRE IKONER



Hvor finder jeg det? Den konkrete klikvej gennem appen — menulinje, Inspector-sektion eller trin i begyndertilstand. De tilhørende tastaturgenveje står også her. Ikonet er en kortnål og viser: Her sidder funktionen i brugerfladen.



Detaljer. Standardværdier, værdiområder og kodestier. Du møder dette især ved træningsindstillingerne, som ikke er et menupunkt, men talparametre. Ikonet viser et lille specifikationskort.



Teknisk. Hvad funktionen gør internt, hvilke parametre der virker, hvad den reagerer på, og hvilke sideeffekter den har. Til læsere, som vil forstå, hvad der sker bag kulisserne. Ikonet er en skydeknappblok og står symbolsk for skruerne under motorhjelm.



Kort fortalt. Kernen i klare ord — uden fagsprog, uden kode. Læs dette afsnit først, hvis du blot hurtigt vil vide, hvad en funktion er til, og hvornår du har brug for den. Ikonet er en taleboble og står for „skåret ind til benet“. Denne kolonne er altid sat på en varm sandtone, så øjet finder den med det samme.

KAPITELFARVER

Hvert kapitel har sin egen accentfarve, som du genkender på ID-mærket (for eksempel **M1**) til venstre for hver opslagstitel og på de små ikoner foran dem. Når du bladrer, ser du straks, hvilket kapitel du befinder dig i.

- 1 Menuer
- 2 Inspector
- 3 Indstillinger
- 4 Hjælpevinduer
- 6 Træning
- 7 Forudindstillinger
- 8 Eksporter
- 9 SfM
- 10 Begyndertilstand

Hurtig start. Hvis du kun interesserer dig for betjeningen, så spring direkte til **Kapitel 10 – Begyndertilstand**. Det er den guidede variant med fire trin og kræver ingen forhåndsviden.

Dybere indstigning. **Kapitel 2 – Inspector** og **Kapitel 7 – Forudindstillinger** forklarer betjeningselementerne og de forudkonfigurerede kvalitetsprofiler, som du har til rådighed i eksperttilstand.

Opslag. Indholdsfortegnelsen og PDF-fuldtekstsøgningen hjælper dig med at finde en bestemt funktion. Du behøver ikke læse vejledningen fra ende til anden.

Introduktion — Hvad du bør vide

Hvad er RadianceKit?

RadianceKit er en native macOS-app, der ud af en række almindelige fotos eller en video laver en begåelig 3D-rekonstruktion. Inputtet er for eksempel 50 til 500 optagelser, som du har taget omkring et objekt, gennem et rum eller hen over et landskab. Outputtet er en såkaldt Gaussian-Splatting-scene — en 3D-model, som du kan se på Mac'en i realtid fra et hvilket som helst perspektiv, som lader sig eksportere og indlejre på websider, og som i sine hovedaspekter ser fotorealistisk ud.

Appen kører helt lokalt på din Mac — der uploades ingen billeder til skyen, der kræves intet login, intet abonnement. Den udnytter GPU'en i din Apple-silicon-Mac (M-serien) intensivt: et fuldt træningsforløb kan, alt efter scene og forudindstilling, tage mellem to minutter og flere timer. Mens beregningen kører, kan du arbejde helt normalt videre på Mac'en, RadianceKit kører videre i baggrunden og melder sig, når resultatet er klart.

Der findes to betjeningstilstande: *Begyndertilstand* (Simple Mode) fører dig i fire trin gennem arbejdsgangen Import → Vælg forudindstilling → Træning → Eksport. *Eksperttilstand* (Expert Mode) åbner en stor Inspector med alle skruer, et live-forhåndsvisningsvindue og diagnose-diagrammer. Du kan til enhver tid skifte mellem tilstandene; dataene i scenen forbliver intakte.

Hvad er Gaussian Splatting?

Gaussian Splatting (ofte kort 3DGS eller bare *Splatting*) er en relativt ny metode til fotorealistisk 3D-fremstilling, præsenteret i 2023 i et paper fra Graz og INRIA. Idéen: i stedet for at modellere en scene som et klassisk polygonnet (trekanter) eller et voxelgitter sættes den sammen af millioner af små, bløde 3D-skyer — hver enkelt sky er en 3D-gaussisk fordeling (deraf navnet) med egen position, størrelse, form, farve og gennemsigtighed. Disse skyer trænes, så de fra alle synsvinkler i dine input-fotos tilsammen giver det rigtige billede.

I praksis betyder det: Gaussian Splatting kan vise refleksioner, højlys, blødt løv, hår eller gardiner på en måde, som klassisk 3D-modellering ikke kan, eller kun kan med enorm indsats. Til gengæld er resultatet ikke en redigerbar 3D-model i klassisk forstand —

du kan ikke bare flytte en enkelt væg eller flytte en vase. Det er snarere en *frossen optagelse* af rummet, som du frit kan bevæge dig igennem. Til mange anvendelser — arkitektur- visualisering, produktpræsentation, virtuelle ture, retsteknik, kulturarv — er det netop den rette styrke.

For at der bliver en 3D-scene ud af input-billederne, kræves to trin. Først udregner appen via en proces ved navn *Structure-from-Motion (SfM)*, hvor dit kamera stod ved hvert foto. Som biprodukt opstår en grov punktsky af scenen. Derefter starter selve Gaussian-Splatting-træningen: med udgangspunkt i denne grove sky bliver de millioner af 3D-skyer skridtvis fordelt, forstørret, forfinet og finjusteret i position og farve, indtil de fra alle input-synsvinkler giver det passende billede.

Du behøver ikke vide noget om nogen af delene for at bruge RadianceKit. Begynder-tilstand skjuler disse trin helt. Men hvis du vil forstå, hvad diagnose-tallene i eksperttilstand (iteration, tab, gaussians, SSIM ...) betyder, eller hvorfor nogle scener bliver flottere end andre, så finder du svarene i de senere kapitler af vejledningen.

KAPITEL

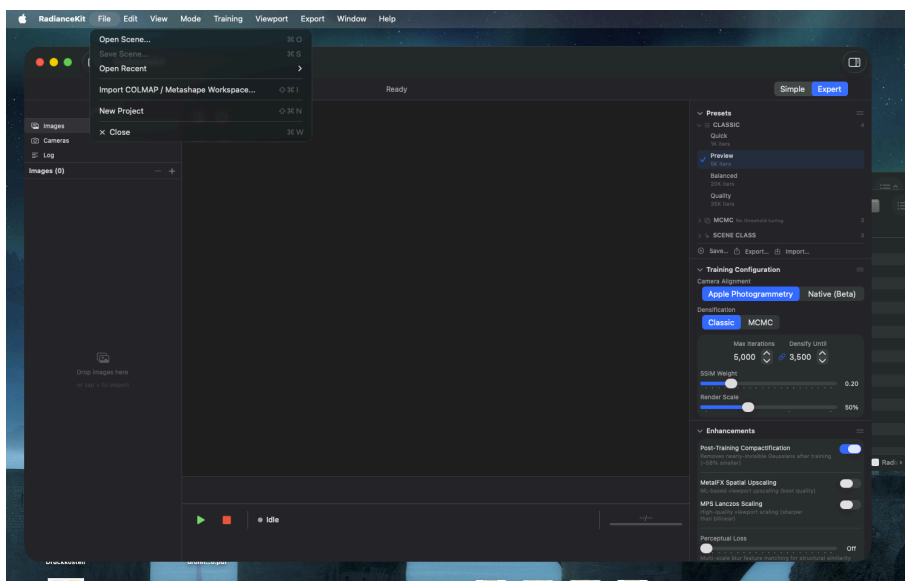
Kapitel 1 — Menulinje

RadianceKits menulinje strukturerer alle funktioner, der ikke ligger direkte i hovedvinduet eller Inspektoren. Det er først og fremmest handlinger, der virker på hele scenen (Åbn, Gem, Nyt projekt), styrer træningen (Start, Pause, Fortsæt), betjener viewporten (auto-rotation, screenshot, baggrundsfarve) og udløser eksporter til forskellige 3D- og medieformater. Dertil kommer springpunkter til alle hjælpevinduer (User Guide, Pareto Dashboard, Holdout Analysis, BayesOpt Console).

Tastaturgenveje står til højre for menuposten. Konventioner: **⌘** betyder Command-tasten (Apple-tasten), **⇧** er Shift, **⌥** er Option (Alt) og **⌘** er Control. Eksempel: **⇧⌘T** står for Shift+Command+T. Alle her dokumenterede genveje vises desuden i et eget oversigtsvindue via **Help** → **Keyboard Shortcuts** (**⌘/**).

De følgende 42 poster er dokumenteret i inventarets rækkefølge (M1–M42), grupperet efter det tilhørende top-niveau-menu. Alle poster er verificeret mod den aktuelle kodestand i (linje 175–477). Ingen poster er fjernet eller forældede i forhold til inventaret; en ny Edit-menupost (Cmd-Z til „Remove Image“) optages af system- NSUndoManager-ramverket og optræder derfor ikke i RadianceKitApp- koden (se note i slutningen af kapitlet).

File-menuen



Figur 1: File-menuen foldet ud — poster M1 til M6

File-menuen erstatter Apples standard-„New Window“-post med projektspecifikke handlinger. Den omfatter indlæsning/gem af scener, en dynamisk Recent-liste, workspace-importen og hård nulstilling til tom tilstand.

M1 File > Open Scene...



Menulinjen → File → Open Scene... (⌘O).



Åbner en filvælger til formaterne `RadianceScene - bundle`, `.ply`, `.splat` og `.spz`. Enkelt-valg, kan vise både filer og mapper (til bundle-formatet). Efter valg skrives stien i Recent-listen, og scenen indlæses asynkront — den forrige erstattes, og trænings-pipelinen initialiseres med den indlæste tilstand. PLY/SPZ/Splat-filer læses via de respektive format-loadere; `.radianceScene`-bundlen er en mappe med manifest, cloud-snapshot og SfM-resultater.

KORT FORTALT

Sådan indlæser du en allerede trænet scene tilbage i appen. Virker med RadianceKits eget format og med standardformaterne PLY, SPLAT og SPZ, som andre splatting-programmer genererer. Brug det, hvis du f.eks. har trænet en scene natten over og næste dag vil arbejde videre eller eksportere. Ved åbning erstattes den hidtidige tilstand i hovedvinduet — så gem først, hvis den nuværende scene stadig er vigtig. Stien lander automatisk i „Open Recent“ (M3), så du kommer hurtigere til den næste gang.

M2 File > Save Scene...



Menulinjen → File → Save Scene... (⌘S).



Åbner en gem-fil-dialog med content-type `RadianceScene - bundle` og fortrykt filnavn `scene.radianceScene`. Skriver en mappe-pakke med `manifest.json`, den serialiserede gaussian-cloud (PLY-snapshot) og et dump af SfM-resultatet, så continue-træning også virker efter genåbning. Posten er deaktiveret, så længe der ikke findes gaussians. Gemmer ikke i training-logs-stien, men dér, hvor gem-dialogen peger — typisk under `~/Documents/`.

KORT FORTALT

Gemmer din aktuelle scene som fil (mere præcist: som en mappe, der ser ud som en fil). Først derefter kan du senere åbne denne scene igen med „Open Scene...“ (M1). I pakken lander både gaussian-cloud'en og SfM-resultatet, så du også kan vedhænge continue-træning (M12–M14) senere. Så længe du endnu ikke har afsluttet en træning, er posten grånet ud. Standardnavnet er `scene.radianceScene` — du kan dog vælge dit eget navn i save-dialogen.

M3 File > Open Recent > [scenenavn]

Menulinjen → File → Open Recent → (liste).



Dynamisk undermenu, der genereres ud fra en liste over de senest åbnede stier (gemt i indstillingerne). Hver liste-post navngives med filnavnet og indlæses ved klik. Hvis listen er tom, vises i stedet det deaktiverede label „No Recent Scenes“. Apple-typisk holder listen de N senest åbnede scener — begrænsningen sker ved skrivning til indstillingerne, ikke i menu-builderen selv.

 KORT FORTALT

Her ser du de senest åbnede scener og kan springe tilbage til dem med et klik uden at gå via fildialogen. Hvis du lige er begyndt, er listen tom og står grå i menuen. Hver scene, du åbner via „Open Scene...“ (M1), lander automatisk i denne liste. Hvis listen på et tidspunkt bliver for fyldt, eller du vil tømme den af privatlivshensyn, så brug „Clear Recent“ (M4).

M4 File > Open Recent > Clear Recent

Menulinjen → File → Open Recent → Clear Recent.



Tømmer Recent-listen i indstillingerne. Virker med det samme uden bekræftelsesdialog. Posten optræder kun i undermenuen, hvis der overhovedet findes poster i Recent-listen (den står under en divider efter stjerne).

 KORT FORTALT

Sletter listen over de senest åbnede scener. Praktisk, hvis du har leget med et test-datasæt og ikke vil se stjerne mere. Selve scene-filerne slettes ikke — kun forbindelsen i menuen. Handlingen virker med det samme uden spørgsmål; derefter optræder „No Recent Scenes“ i undermenuen. Posten dukker kun op, hvis der overhovedet er scener i listen — ved tom liste er den ikke synlig.

M5 File > Import COLMAP / Metashape Workspace... HVOR

Menulinjen → File → Import COLMAP / Metashape Workspace... (⇧⌘I).

 TEKNISK

Åbner en mappe-vælger. Forventer en mappe med COLMAP-workspace-layout (f.eks. `sparse/0/cameras.{bin,txt}` plus `images/`). Efter valg foretages en for-prøve af workspace — denne genkender de tre layouts (`sparse/0/`, `sparse/`, `roden`), og om rekonstruktionen er binær (`cameras.bin`) eller ETH3D-tekst (`cameras.txt`). Ved succes importeres workspacet; ellers vises kun en advarsel i app-loggen. Se også kapitel 9 „SfM-backends“, Q6 for den fulde pipeline-logik.

 KORT FORTALT

Hvis du bruger Metashape, COLMAP, RealityCapture eller lignende software til kamera-rekonstruktionen og har en eksport, indlæser du mappen her. RadianceKit springer så SfM-trinet over og starter direkte med træningen — det sparer timer på store scener. Træk-og-slip på hovedvinduet virker på samme måde. Forventer en mappe med COLMAP-layout (altså `sparse/0/` med `cameras.*` plus `images/`-mappe). Mere om de understøttede layouts og workflows står i kapitel 9 „SfM-backends“.

M6 File > New Project HVOR

Menulinjen → File → New Project (⇧⌘N).

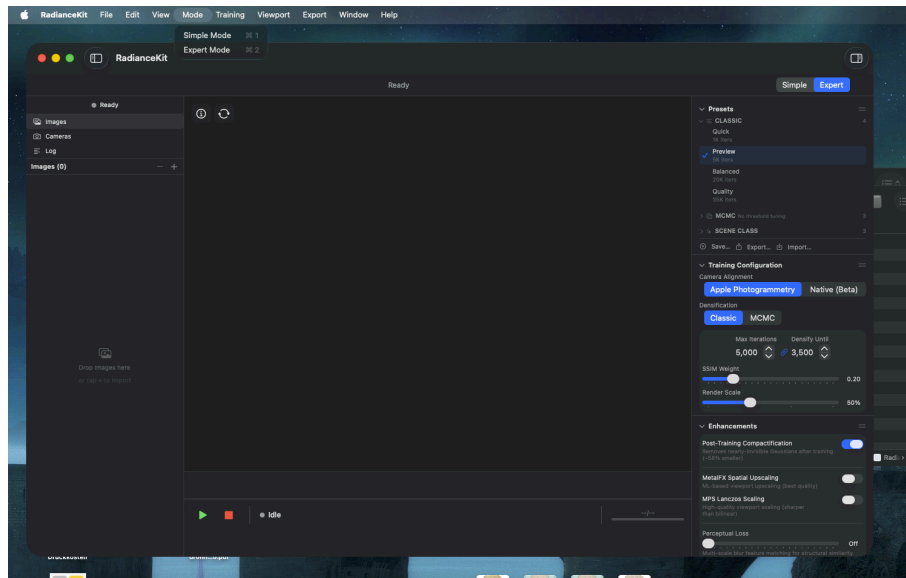
 TEKNISK

Tjekker, om der findes ugemt arbejde. Hvis ja, dukker en bekræftelsesdialog op, før noget går tabt. Hvis der ikke er noget at gemme, kører nulstillingen direkte — den tømmer importerede billeder, SfM-resultatet, gaussian-cloud'en, trænings-state og alle afhængige UI-indikatorer. Advarsel: et preset-bibliotek oprettet af brugeren bevares, fordi det ligger i app-indstillingerne og ikke i projekt-tilstanden.

 KORT FORTALT

Nulstiller alt til en tom start — som om du lige havde åbnet appen frisk. Hvis du stadig har ugemt arbejde, spørger appen først. Brug det, hvis du vil starte med en helt anden scene. Importerede billeder, SfM-resultat, gaussian-cloud og trænings-tilstand tømmes helt. Dine egne presets bevares dog, fordi de ligger i app-indstillingerne og ikke hører til scenen.

Mode-menuen



Figur 2: Mode-menu med Simple- og Expert-Mode-skifter

To enkle kontakter mellem den guidede Simple Mode (wizard-agtig, 4 trin) og den fulde Expert Mode (klassisk Inspector-layout med alle håndtag).

M7 Mode > Simple Mode

HVOR

Menulinjen → Mode → Simple Mode (⌘1).

TEKNISK

Skifter app-tilstanden til Simple Mode. Appens hovedområde viser så den guidede workflow i stedet for Expert-layoutet. Mode-tilstanden gemmes i indstillingerne (se S1 „Default Mode“ i kapitel 3 Settings).

KORT FORTALT

Skifter til skridt-for-skridt-varianten, hvor appen fører dig gennem import, behandling, forhåndsvisning og eksport. Anbefales, hvis du lige er begyndt, eller hvis du hurtigt skal bruge et resultat. De fleste detalje-håndtag er skjult — du arbejder med fornuftige fortrukne indstillinger. Hvis du senere vil dykke dybere, så skift bare til Expert Mode (M8). Hvilken tilstand der er aktiv ved app-start, kan du fastlægge i indstillingerne (kapitel 3, S1).

M8 Mode > Expert Mode



HVOR

Menulinjen → Mode → Expert Mode (⌘2).



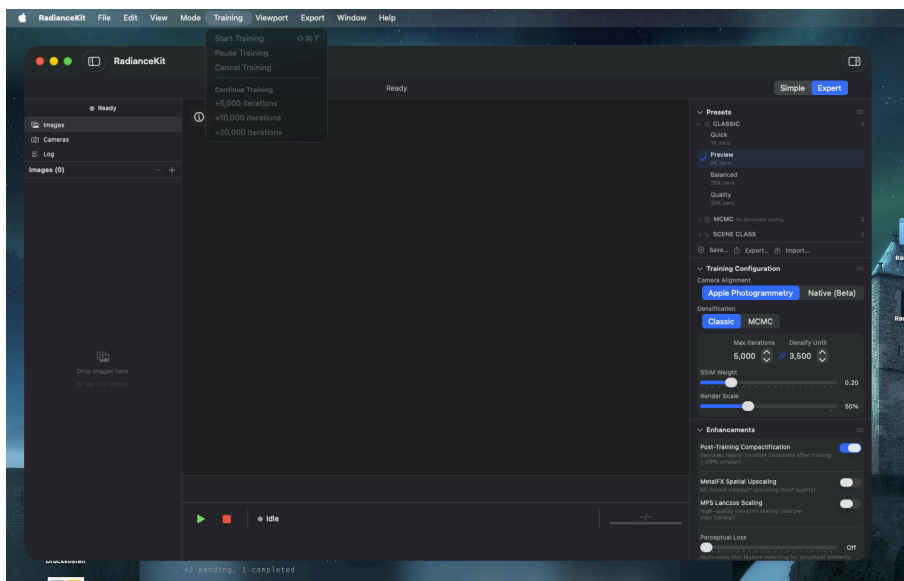
TEKNISK

Skifter app-tilstanden til Expert Mode. Dermed optræder det fulde Inspector-layout med alle sektioner (Presets, TrainingConfig, Enhancements, Metrics, LossChart, ProjectNavigator). I Expert Mode er alle training-parametre, COLMAP-vælger, mid-compact-toggles og diagnostics tilgængelige. Også live-preview fungerer kun i denne tilstand.

KORT FORTALT

Skifter til fuldvisning med alle håndtag. Her ser du loss-charts i realtid, kan finjustere alle parametre og forvalte flere sammenlignings-konfigurationer parallelt via presets. Anbefales, hvis du vil forstå, hvad træningen internt gør, eller hvis du målrettet vil eksperimentere. Også live-preview, COLMAP-vælger og diagnostics er kun tilgængelige her. Hvis du føler dig overvældet, så gå tilbage til Simple Mode via M7 — din scene bevares.

Training-menuen



Figur 3: Training-menuen med Continue-undermenu — poster M9 til M14

Fire handlinger omkring trænings-kørslen: starte, pause, afbryde og forlænge med et angivet antal iterationer. Alle tre continue-poster er IAP-gated (kan ikke klikkes i free-trial-versionen).

M9 Training > Start Training

HVOR

Menulinjen → Training → Start Training (⇧⌘T).



TEKNISK

Starter trænings-pipelinen asynkront. Forudsætning: et SfM-resultat foreligger, og der kører ikke en anden pipeline. Begge betingelser blokerer posten, hvis de ikke er opfyldt. Ved start læses de aktuelle konfigurationsværdier, en ny JSONL-log oprettes under `~/Documents/RadianceKit/Logs/training_YYYY-MM-DD_HHmmss.jsonl`, og afhængigt af strategivalget køres klassisk eller MCMC-stien. Trænings-tilstanden skifter fra „idle“ til „training“.

KORT FORTALT

Trykker på den store grønne knap — så snart du har importeret fotos, og kamera-rekonstruktionen er ovre, begynder den egentlige Gaussian-Splatting-træning. Lad appen køre; afhængigt af preset mellem 1 minut (Quick) og flere timer (MCMC Quality). Posten forbliver grå, så længe der ikke foreligger SfM-resultat, eller en anden pipeline kører. Hver kørsel skriver en log til `~/Documents/RadianceKit/Logs/`, som du senere kan analysere via Pareto Dashboard (M40).

M10 Training > Pause Training

HVOR

Menulinjen → Training → Pause Training.



TEKNISK

Pauser den løbende træning. Aktiveres kun, når trænings-tilstanden er „training“. Pause stopper iterations-loopet ved næste sikkerheds-sync-point, beholder den fulde GPU-state (gaussian-buffers, optimizer-momenter, scheduler-position) og skifter til „paused“. Resume sker via fornyet tryk (post-titlen er statisk — appen skifter dog mellem pause/resume i selve logikken). Pausede træninger overlever ikke app-quit; i så fald gem hellere scenen og udvid den senere via continue-training-post (M12–M14).

KORT FORTALT

Holder kort træningen tilbage uden at miste fremskridtet. Praktisk, hvis du kort skal bruge computeren til noget vigtigere. Klik igen for at fortsætte. Fungerer ikke på tværs af app-genstart — hvis du virkelig vil fortsætte senere, afslut træningen med Cancel (M11), gem scenen med Save Scene (M2) og brug derefter Continue Training (M12–M14). Under pause hviler GPU'en fuldstændigt; hukommelsen forbliver dog optaget.

M11 Training > Cancel Training

Menulinjen → Training → Cancel Training.



Afbryder den løbende træning. Aktiv, når træningstilstanden ikke er „idle“. Sætter cancel-flaget i trænings-engine, hvilket pænt afslutter iterations-loopet ved næste sync-point, skriver det endelige summary-element med i JSONL-loggen og nulstiller tilstanden til „idle“. Den hidtil trænedede cloud bevares (kan gemmes eller eksporteres), men markeres som „cancelled“.

 KORT FORTALT

Afbryder den løbende træning endeligt. Den hidtidige tilstand bevares — så hvis du allerede har et fremvisningsegnet resultat efter et par tusinde iterationer, kan du alligevel eksportere det. Hvis du kun vil afbryde kort, så brug Pause (M10) i stedet. I trænings-loggen markeres kørslen som „cancelled“, den endelige loss-værdi skrives alligevel ned. En afbrudt scene kan du også fortsætte senere via Continue Training (M12–M14), så længe appen ikke er blevet afsluttet i mellemtiden.

M12 Training > Continue Training > +5 000 iterationer

Menulinjen → Training → Continue Training → +5,000 iterations.



Fortsætter træningen med 5 000 iterationer. Aktiv, når en afsluttet træning kan fortsættes, og fuldversionen er oplåst. Fortsætteligheden gælder, når en afsluttet træning findes, og den fulde optimizer-state stadig er i hukommelsen. Ved continue føres Adam-momenterne og LR-scheduleren videre, så fortsættelsen opfører sig som en gennemgående 25K-/45K-/60K-kørsel i stedet for en ny start. JSONL-loggen får en ny config-post med den inkrementelle opsætning. Kun tilgængelig i fuldversionen.

 KORT FORTALT

Hænger 5 000 yderligere trænings-skridt på. Brug det, hvis resultatet efter første kørsel er tæt på, men endnu ikke helt skarpt. Fungerer kun i den betalte fuldversion. I modsætning til en helt ny kørsel bevares optimizer-tilstanden, så fortsættelsen føles som en gennemgående kørsel. Hvis du behøver mere end 5 000 skridt, så tag direkte M13 (+10 000) eller M14 (+20 000).

M13 Training > Continue Training > +10 000 iterationer **HVOR**

Menulinjen → Training → Continue Training → +10,000 iterations.

 **TEKNISK**

Identisk med M12, men med 10 000 yderligere iterationer. Samme forudsætninger, samme LR-scheduler-sti. Anbefales, hvis den indledende træning blev kørt med en mid-tier-preset, og du vil se en signifikant kvalitetsstigning uden at starte kørslen helt forfra.

 **KORT FORTALT**

Forlænger træningen med 10 000 skridt — den midterste af de tre tilgængelige continue-værdier. Godt valg, hvis første kørsel var okay, men du klart vil blive bedre. Som M12 og M14 fortsættes læringsrate-forløbet sømløst i stedet for at starte forfra. Kun tilgængelig i fuldversionen.

M14 Training > Continue Training > +20 000 iterationer **HVOR**

Menulinjen → Training → Continue Training → +20,000 iterations.

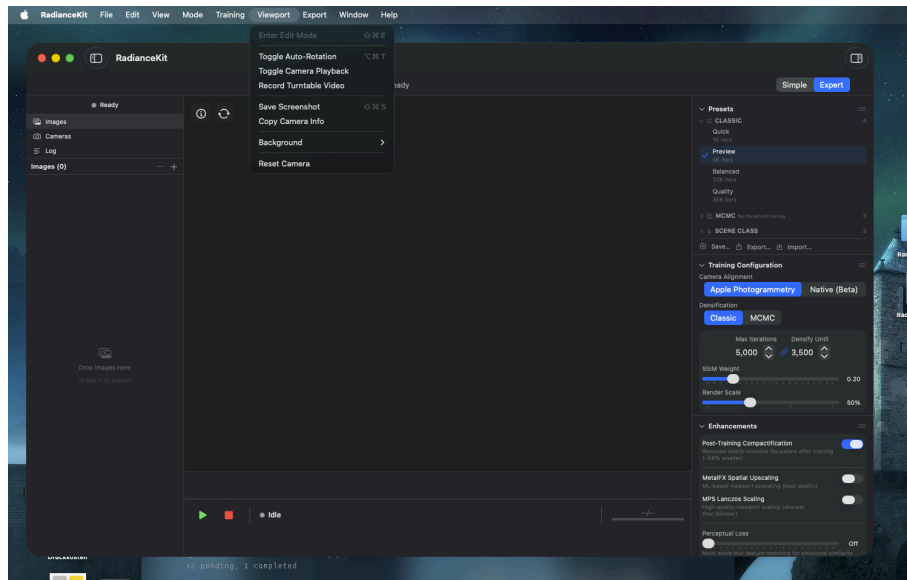
 **TEKNISK**

Identisk med M12 / M13, men med 20 000 yderligere iterationer. Det største foruddefinerede continue-spring. Ved MCMC-træninger er det ofte det, der gør forskellen mellem „passer“ og „benchmark-egnet“; ved Classic fra 35–40K kommer der erfaringsmæssigt ikke meget til.

 **KORT FORTALT**

Hænger 20 000 yderligere trænings-skridt på, den maksimale continue-værdi. Brug det, hvis du virkelig vil hente den sidste kvantitet kvalitet ud. Ved klassisk træning efter 40 000 skridt giver det ofte ikke meget mere — ved MCMC derimod betaler det sig ofte, fordi konvergens sætter ind langsommere der. Regn med markant ekstra køretid afhængigt af scenen. Som M12 og M13 er også denne post kun tilgængelig i fuldversionen.

Viewport-menuen



Figur 4: Viewport-menuen med edit-mode, kamerastyring og baggrund-undermenu

Styrer 3D-viewporten: edit-mode til gaussian-valg og oprydning, kamerastyring (auto-rotation, playback, recording), screenshot, baggrundsfarve og reset.

M15 Viewport > Enter/Exit Edit Mode

HVOR

Menulinjen → Viewport → Enter Edit Mode (eller „Exit Edit Mode“ alt efter tilstand). ⌘⌘E.

TEKNISK

Posttitlen er dynamisk og viser alt efter tilstand „Exit Edit Mode“ eller „Enter Edit Mode“. Ved tryk skiftes edit-mode på viewport-rendereren. Ved at forlade Edit Mode nulstilles desuden det aktuelle valg. Edit Mode aktiverer klik-selektion på gaussians, box-selection og sletning af markerede gaussians (se editor-området i UI'en). Deaktiveret, så længe der ikke er forbundet en viewport-renderer.

KORT FORTALT

Skifter mellem normal 3D-visning og en redigeringsstilstand, hvor du kan markere og slette enkelte gaussians (f.eks. floaters eller outliers i baggrunden). Ved forloden nulstilles valget automatisk. Posten forbliver grå, så længe der endnu ikke er en scene synlig i viewporten. Etiketten skifter alt efter tilstand mellem „Enter Edit Mode“ og „Exit Edit Mode“ — du ser altså altid, hvilken tilstand du er i.

M16 Viewport > Toggle Auto-Rotation

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



Slår den kontinuerlige rotation af viewport-kameraet om en lodret akse gennem scenens center til eller fra. Akse og hastighed kommer fra kamerastyrings-konfigurationen. Auto-rotation er en ren viewport-effekt og påvirker hverken træning eller recording — hvis du parallelt bruger turntable-video-recorderen (M18), leverer auto-rotationen dog præcis den sti, recorderen indfanger.

 KORT FORTALT

Drejer kameraet langsomt rundt om scenen, så du kan se den fra alle sider uden at trække med musen. Klik igen stopper rotationen. Praktisk ved vurdering af færdigtrænede scener eller som baggrundsanimation til en live-demo. Hvis du parallelt optager en video (M18), leverer auto-rotationen præcis den bevægelse, recorderen indfanger.

M17 Viewport > Toggle Camera Playback

Menulinjen → Viewport → Toggle Camera Playback.



Slår kamerasti-playback til/fra. Hvis der findes en optaget kamerasti (f.eks. fra en tidligere recording, eller fordi en `transforms.json` er indlæst), kører stien — viewport- kameraet bevæger sig altså ikke længere efter muse-/trackpad- input, men reproducerer banen frame for frame. Fornyet tryk pauser playbacket.

 KORT FORTALT

Lader en tidligere optaget eller importeret kamerakørsel afspille. Sådan kan du efterse den originale sti, scenen blev optaget med, eller tjekke en planlagt orbit-bevægelse før video-eksporten. Mens playbacket kører, er muse- og trackpad-input deaktiveret — kameraet følger strikt stien. Fornyet klik pauser afspilningen. Hvis du ikke har indlæst eller optaget en kamerasti, sker der ingenting.

M18 Viewport > Record Turntable Video

Menulinjen → Viewport → Record Turntable Video.



Slår viewport-optagelsen til/fra. Ved første tryk starter en frame-optagelse i en midlertidig sti; ved andet tryk afsluttes optagelsen, enkodes og skrives til en MP4-sti (stien spørges via en gem-dialog). I modsætning til Export → Media → Orbit Video (M31), der genererer en fast 360°-sti med indstillelig varighed, optager turntable-recorderen *live* det, du ser i viewporten — du kan altså også optage en manuel kamerakørsel.

KORT FORTALT

Optager en video direkte i viewporten. Lige meget om kameraet drejer automatisk, eller om du selv flytter det med musen — alt, hvad du ser, gemmes i en MP4-fil. I modsætning til „Orbit Video“-eksporten (M31) angiver du selv kamerakørslen. Første klik starter optagelsen, andet klik afslutter den og spørger om gemmesteder. Praktisk, hvis du f.eks. vil vise en bestemt detaljesvingning, som den stive orbitbevægelse ikke kunne klare.

M19 Viewport > Save Screenshot

Menulinjen → Viewport → Save Screenshot (⇧⌘S).



Fanger et enkelt viewport-frame i fuld render-opløsning (altså ikke vinduets pixel-layout, men det fulde render-target- indhold) som PNG-fil. Stien spørges via en gem-dialog. Baggrundsfarven (M21–M23) brændes med ind. MetalFX-/MPS-upscaling- indstillinger fra Enhancements (se I27/I28) virker, hvis de er aktive — screenshottet viser altså den opskalerede output.

KORT FORTALT

Gemmer et øjebliksbillede af din aktuelle 3D-visning som PNG-billede. Praktisk til marketing-materiale eller en hurtig sammenligning. Bemærk: baggrunden er en del af billedet — hvis du har brug for gennemsigtighed, så eksportér hellere en scene-fil. Opløsningen svarer til det interne render-target, ikke din vinduestørrelse — billedet er altså ofte skarpere, end det ser ud i vinduet. Eventuelle upscaling-indstillinger (Inspector → Enhancements) regnes også med.

M20 Viewport > Copy Camera Info

Menulinjen → Viewport → Copy Camera Info.



Læser den aktuelle viewport-kamera-pose (position, look-at-punkt, up-vektor) og FOV-værdier fra kamerastyringen og skriver dem som flerlinje-tekst i udklipsholderen. Formatet er læseligt for mennesker (label = value pr. linje), ikke JSON. Praktisk til at reproducere en specifik visning til debug-formål eller dele den med support.

 KORT FORTALT

Kopierer den aktuelle kamera-position og kigge-retning som tekst i udklipsholderen. Hvis du f.eks. vil vise en medudvikler, hvorfra et sted i scenen ser sært ud, så indsætter du blot teksten i en mail eller et chat-vindue. Formatet er læseligt for mennesker (én linje pr. værdi), ikke JSON. Mest tænkt til bug-rapporter eller support-forespørgsler.

M21 Viewport > Background > Dark Gray

Menulinjen → Viewport → Background → Dark Gray.



Sætter viewport-baggrundsfarven til en mørkegrå (RGB 0.1/0.1/0.1). Rendereren bruger denne farve som baggrund, foran hvilken gaussians compo-siteres. Standardfarven ved app-start styres af Settings-muligheden S3 „Default Viewport Background“.

 KORT FORTALT

Farver 3D-viewportens baggrund mørkegrå. Standardvalget til de fleste scener — giver god kontrast både til lyse og mørke gaussians, uden at øjet hænger fast i en ren sort- eller hvidflade. Farven overtages også i screenshots (M19) og orbit-videoer (M31). Hvis Dark Gray er for kedeligt for dig, så prøv Black (M22) eller White (M23) til sammenligning. Hvilken farve der er aktiv ved app-start, kan du fastlægge i indstillingerne (S3).

M22 Viewport > Background > Black

Menulinjen → Viewport → Background → Black.



Sætter viewport-baggrundsfarven til rent sort (RGB 0/0/0). Hjælper, hvis scenen har mange lyse floaters, og du vil identificere dem, eller til marketing-materiale med mørk look-and-feel.

 KORT FORTALT

Sort baggrund. Godt til meget lyse scener, eller hvis du vil kigge ind i Edit Mode og leder efter små lyse gaussians (floaters), der forsvinder i det grå. Også ideelt til marketing-materiale med mørkt, dramatisk look. Farven brændes ind i screenshots og orbit-videoer — hvis du har brug for genomsigtighed til en senere composit, er sort det dårligste valg. Til mørke floaters skift den anden vej til White (M23).

M23 Viewport > Background > White

Menulinjen → Viewport → Background → White.



Sætter viewport-baggrundsfarven til rent hvidt (RGB 1/1/1). Nyttig, hvis scenen overvejende har mørkt indhold, og du vil se mørke floaters (typisk outdoor-baggrundsstøj).

 KORT FORTALT

Hvid baggrund. Praktisk, hvis motivet hellere kommer til sin ret lyst-på-mørkt, eller for at finde mørke outliers, som du derefter kan fjerne i Edit Mode (M15). Ved outdoor-scener er hvid ofte mere nyttig end sort, fordi de typiske outdoor-floaters er mørke. Som ved de andre baggrundsmuligheder overtages farven i screenshots og videoer.

M24 Viewport > Reset Camera

Menulinjen → Viewport → Reset Camera.

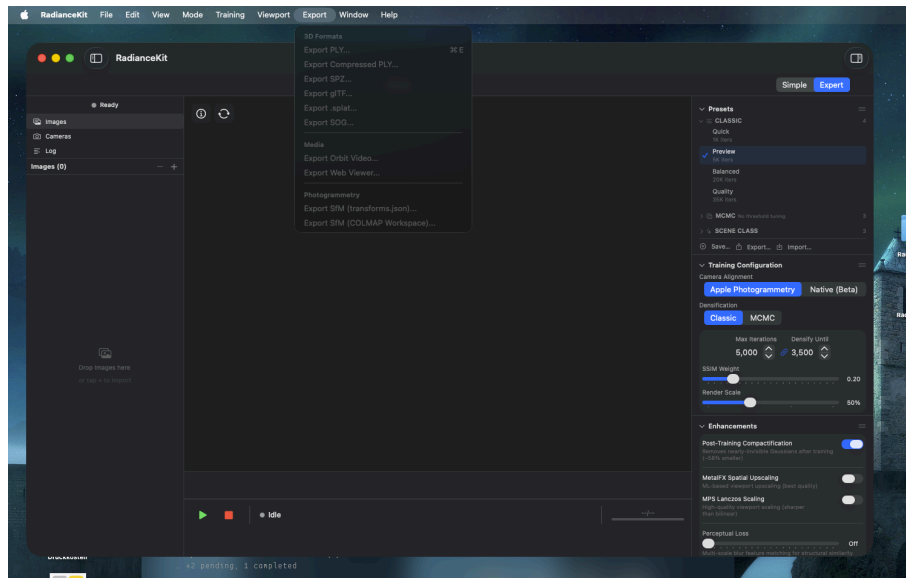


Nulstiller viewport-kameraet, forlader training-camera-visningen og stopper auto-rotationen. Dermed er kameraet tilbage på startpositionen (typisk: foran scenen, let oppefra), auto-rotationen er slukket, og hvis rendereren netop viste training-kameraet (en af SfM-poserne), går den tilbage til free-kameraet.

 KORT FORTALT

Bringer viewport-kameraet tilbage til startpositionen. Hvis du har vildet væk under rundrejsen eller skubbet scenen ud af billedet — et klik her, og du ser igen, hvad du skal se. Slår samtidig auto-rotationen fra, hvis den kører, og vender tilbage fra et fastfrosset training-kamera til den frie visning. Sådan får du i alle tilfælde en ren genstart af visningen.

Export-menuen



Figur 5: Export-menuen med tre undermenu-grupper — 3D Formats, Media og Photogrammetry

Otte eksportmål plus to fotogrammetri-eksporter, grupperet i tre sektioner (3D Formats, Media, Photogrammetry). De første seks bygges via en fælles helper-rutine, som åbner en gem-dialog og registrerer eksporten i format-kataloget. Photogrammetry-posterne har individuel logik. Alle photogrammetry- og nogle 3D-eksporter er kun tilgængelige i fuldversionen.

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



Menulinjen → Export → 3D Formats → PLY (⌘E).



Åbner en gem-dialog med default-filnavn `gaussians.ply`. Ved OK skrives den aktuelle gaussian-cloud i det standardiserede ASCII/binary-PLY-format — kompatibel med SuperSplat, PolyCam, PlayCanvas og alle gængse 3DGS-viewere. Fulde SH-koefficienter, fuld præcision (Float32 pr. felt). Filstørrelse ofte flere hundrede MB ved $\geq 500K$ gaussians.

KORT FORTALT

Gemmer din 3D-scene som standard-PLY-fil. Det er det mest universelle format — næsten enhver software kan indlæse det, fra SuperSplat over PolyCam til PlayCanvas. Filerne bliver dog store, ofte flere hundrede megabyte. Brug PLY, hvis du vil arbejde videre i fuld kvalitet eller arkivere. Hvis du vil dele scenen via nettet, så kig hellere på SPZ (M27) eller Compressed PLY (M26) — de er væsentligt mindre.

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

Menulinjen → Export → 3D Formats → Compressed PLY.

 **TEKNISK**

Skriver gaussian-cloud'en i Compressed-PLY-formatet med custom-kvantisering af position-, scale-, rotation- og SH-felterne. 5–10× mindre filer end ukomprimeret PLY (M25) med minimale visuelle tab. Kompatibel med SuperSplat (som læser Compressed-PLY-standarden) og PlayCanvas. Standardfilnavn `gaussians_compressed.ply`.

 **KORT FORTALT**

Som almindeligt PLY, men 5–10 gange mindre. Kvaliteten forbliver næsten den samme. Brug det, hvis du vil dele filen online eller sender den pr. e-mail. Fungerer direkte med SuperSplat og PlayCanvas. Hvis dit målsystem har brug for endnu mindre filer (mobil, browser-demoer), så vælg SPZ (M27) i stedet — det er endnu mere aggressivt komprimeret. Til fuld redigeringskvalitet vælg det ukomprimerede PLY (M25).

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

Menulinjen → Export → 3D Formats → SPZ.

 **TEKNISK**

Skriver gaussian-cloud'en i SPZ-formatet — det af Niantic udgivne komprimerede splat-format med aggressiv kvantisering (~90 % mindre end ukomprimeret PLY). Optimeret især til web-viewere og mobile apps. Kompatibel med Niantic Splatt3R, `gsplat.js` og Niantic-browser-vieweren.

 **KORT FORTALT**

Et af de mindste formater. Ca. 10× mindre end et almindeligt PLY. Brug det især, hvis du vil vise scenen i en browser eller via mobil-app. Til maksimal kvalitet er PLY det bedre valg. SPZ er udviklet af Niantic og fungerer direkte med `gsplat.js`, Splatt3R og Niantic-web-vieweren. På grund af den stærke komprimering kan du ikke længere uden videre videtræne SPZ-filer — til redigering vælg PLY.

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

Menulinjen → Export → 3D Formats → glTF.



Skriver en `.glb` -fil (binary-glTF) med `KHR_gaussian_splatting`-extensionen. Standardkonform, egnet til pipelines, der bruger glTF-engines som Babylon.js eller Three.js og implementerer `KHR_gaussian_splatting`-extensionen.

 KORT FORTALT

Gemmer scenen i glTF-formatet, som mange 3D-programmer og web-engines forstår — forudsat at de understøtter Gaussian Splatting-udvidelsen. Hvis du har en specifik 3D-pipeline (f.eks. Three.js eller Babylon.js), der forstår det, er det dit format. Filen kommer ud som binær `.glb` — en enkelt pakke, der indeholder alt. Til klassiske splatting-workflows er PLY eller SPZ som regel det bedre valg, fordi flere værktøjer forstår dem direkte.

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

Menulinjen → Export → 3D Formats → .splat.



Skriver Antimatter15- `.splat` -formatet — fast størrelse 32 bytes pr. gaussian (position som 3× Float32, scale som 3× Float32, rotation som 4× Uint8 normaliseret quaternion, RGB+Opacity som 4× Uint8). Ingen SH-koefficienter højere end DC. Mindste fil med browser-direkte-kompatibilitet. Til `gsplat.js` og `antimatter15s` online-demo-viewer.

 KORT FORTALT

Det simpleste web-viewer-format. Lille og straks visningsklart i enhver browser. Mister dog detalje-belysningen (højere SH-koefficienter går tabt — splat'en ser ens ud fra enhver synsvinkel i stedet for at reagere på lyset). Til maksimal web-performance er det godt, til fotorealisme snarere SPZ eller PLY. Fungerer med `antimatter15-online-viewer` og `gsplat.js`. Hver gaussian optager fast 32 bytes, hvilket gør formatet simpelt og kompatibelt — men netop på bekostning af detaljedybden.

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

Menulinjen → Export → 3D Formats → SOG.



Skriver gaussian-cloud'en i SOG-formatet. SOG („Self-Organizing Gaussian“) er PlayCanvas-formatet med texture-atlas-layout og WebP-komprimering af de kvantiserede data. Skalerer med 15–20× bedre størrelsesforhold end PLY. Eksporten kalder internt `cwebp` som eksternt værktøj — derfor i sandbox-varianten (App Store) potentielt begrænset.

 KORT FORTALT

Meget lille format til PlayCanvas-workflows. Ca. 15–20 gange mindre end PLY, fordi dataene pakkes i et texture-atlas-layout og WebP-komprimeres. Hvis du ikke har en PlayCanvas-workflow, er SPZ eller Compressed PLY som regel det bedre valg. Eksporten kalder internt `cwebp` som eksternt værktøj — i App Store-versionen (sandbox) kan dette trin være begrænset.

M31 Export > Media > Export Orbit Video...

Menulinjen → Export → Media → Orbit Video.



Rendrer et 360°-orbit omkring scenens center og enkoder den som MP4 (H.264) eller MOV (HEVC, alt efter systemstandard). I modsætning til M18 (live-recording) er stien her fast — varighed vælges i Settings eller i Simple-Mode-eksport-trinet.

 KORT FORTALT

Genererer automatisk en drejevideo omkring din scene. Ingen manuel bevægelse nødvendig. Godt til sociale medier eller en hurtig demo. Hvis du selv vil styre kameraet, så brug Record Turntable Video (M18) i stedet. Stien er fast: en fuld 360°-orbit omkring scenens center, varighed vælger du i indstillingerne eller i Simple-Mode-eksport-trinet. Videoen udgives alt efter system som H.264-MP4 eller HEVC-MOV.

M32 Export > Media > Export Web Viewer...

Menulinjen → Export → Media → Web Viewer.



Pakker en standalone-HTML-viewer (gsplat.js-baseret) plus gaussian-dataene base64-kodet i en enkelt `.html`-fil. Filen kører offline i enhver moderne browser — ingen server-afhængigheder, ingen eksterne URL'er. Filstørrelsen er ca. faktor 1.3 større end SPZ-varianten (på grund af base64-overhead).

 KORT FORTALT

Gemmer din scene som selv-startende hjemmeside. Dobbeltklik på HTML-filen → browseren åbner → færdig interaktiv 3D-scene. Fungerer uden internet, kan sendes pr. mail, er den enkleste måde at dele resultatet med venner eller kunder. Filen indeholder den komplette gsplat.js-viewer og gaussian-dataene i ét enkelt dokument — intet hentes fra nettet. Filstørrelsen er ca. en tredjedel større end en SPZ-eksport, til gengæld behøver modtageren ikke yderligere software.

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

Menulinjen → Export → Photogrammetry → Export SfM (transforms.json).



Egen eksport-sti (ikke via fælles helper-rutine), fordi ingen gaussian-cloud, men SfM-resultatet eksporteres. Åbner en gem-dialog med `transforms.json` som default og content-type `json`. Ved OK skrives en nerfstudio-kompatibel `transforms.json` med kamera-intrinsics, poser (som 4x4-matrix i NeRF-konvention) og frame-stier. Hjælpetekst i UI'en påmindrer om, at træningsbillederne skal kopieres med som sibling-mappe `images/`. Aktiv kun, når et SfM-resultat foreligger, og fuldversionen er oplåst.

 KORT FORTALT

Hvis du vil bruge SfM-resultatet videre i en anden software som nerfstudio, Brush, gsplat eller OpenSplat, eksporterer du her kamera-positionerne. Læg dine træningsbilleder desuden i en `images/`-mappe ved siden af `transforms.json`-filen — ellers kan målprogrammet ikke knytte billederne. Posten er grånet ud, så længe der ikke findes SfM-resultat, og spærret i free-trial-versionen. Til COLMAP-workspace-workflowen tag M34 i stedet.

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



HVOR

Menulinjen → Export → Photogrammetry → Export SfM (COLMAP Workspace).



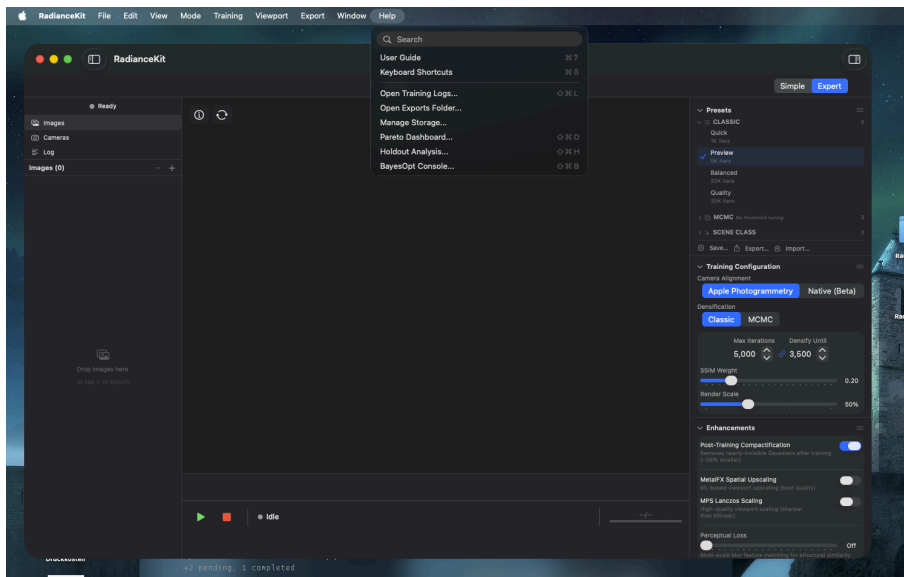
TEKNISK

Åbner en gem-dialog med default-navn `colmap-workspace` (uden extension, fordi det er en mappe). Skriver et standard-COLMAP-workspace med `sparse/0/cameras.bin`, `images.bin`, `points3D.bin`. Tillader at åbne en i RadianceKit beregnet eller importeret SfM-rekonstruktion i andre værktøjer som Postshot, Nerfstudio eller Meshroom, eller ved en A/B-re-run at indlæse den igen som færdigberegnet input i selve RadianceKit (via M5) — sparer regnetid. Aktiv kun, når et SfM-resultat foreligger, og fuldversionen er oplåst.

KORT FORTALT

Som M33, men i COLMAP-formatet i stedet for nerfstudio. Hvis du bruger Postshot, Meshroom, Nerfstudio eller et andet værktøj med COLMAP-workflow, er det din eksport. En praktisk sideeffekt: du kan indlæse denne mappe igen senere via M5 i RadianceKit og spare SfM-regnetiden ved næste kørsel — især ved store scener en tidsgevinst på timer. Som M33 kun tilgængelig, når et SfM-resultat foreligger, og spærret i free-trial-versionen.

Help-menuen



Figur 6: Help-menuen med dokumentations-, mappe- og analyse-poster

Syv poster: to dokumentationsvinduer (User Guide, Keyboard Shortcuts), tre mappegenveje (Training Logs, Exports, Storage) og tre analysevinduer (Pareto Dashboard, Holdout Analysis, BayesOpt Console). Apple-typisk optræder Help-menuen helt til højre. Standard-Help-menuen erstattes helt af RadianceKits egen variant.

M35 Help > User Guide

Menulinjen → Help → User Guide (⌘?).



Åbner User Guide-vinduet. Det viser en navigation med emne-sidebar og scroll-detalj område ved default-størrelse 860×640. Indholdet er statisk hentet (ikke parsed fra Markdown).

 KORT FORTALT

Åbner app-intern vejledning. Hvis du ikke vil slå alt op i denne manual, finder du de vigtigste trin direkte i programmet. Vejledningen er bygget op som et eget vindue med emne-sidebar — du kan altså målrettet hoppe til enkelte emner. Indholdet er kortere end denne håndbog og koncentrerer sig om de hyppigste workflows.

M36 Help > Keyboard Shortcuts

Menulinjen → Help → Keyboard Shortcuts (⌘/).



Åbner Keyboard Shortcuts-vinduet — et simpelt scroll-layout med alle app-geneveje, grupperet efter top-niveau-menu. Default-størrelse 440×560. Indhold er ligeledes statisk hentet.

 KORT FORTALT

Åbner et vindue med den komplette liste over alle tastaturgeneveje. Hvis du f.eks. ikke kan huske, med hvilken tast man starter træningen, så kig der. En sammenfatning står også sidst i dette kapitel. Listen er grupperet efter top-niveau-menu, så du hurtigt hopper til det rigtige område. Hjælpesom, hvis du netop er ved at skifte fra muse- til tastaturstil.

M37 Help > Open Training Logs...

Menulinjen → Help → Open Training Logs... (⇧⌘L).



Beregner log-mappen som `~/Documents/RadianceKit/Logs`, opretter den om nødvendigt og åbner den i Finder. Hver trænings-kørsel skriver en egen JSONL-fil `training_YYYY-MM-DD_HHmss.jsonl` dertil.

KORT FORTALT

Åbner i Finder mappen med alle hidtidige træningsprotokoller. Hvis noget er gået galt, eller du vil se efter, hvornår præcis træningen konvergerede til hvilken værdi, finder du det her i JSONL-filer. Pr. trænings-kørsel oprettes præcis én fil med tidsstempel — den kan du også læse i andre værktøjer eller sende pr. mail til support. Hvis du vil have en grafisk evaluering, er Pareto Dashboard (M40) den bedre indstilling.

M38 Help > Open Exports Folder...

Menulinjen → Help → Open Exports Folder...



Analogt med M37, men med `~/Documents/RadianceKit/Exports`. Oprettede ved første auto-test-kørsel eller ved første klik; derefter lander standardstierne af alle auto-test-eksporter dér (f.eks. `autotest_<timestamp>.ply`). Manuelt via gem-dialogen valgte eksporter går IKKE nødvendigvis herind, men der, hvor brugeren gemmer det — derfor er denne mappe især interessant for auto-tests.

KORT FORTALT

Åbner mappen, hvor appen lægger sine egne eksporter (især auto-test-kørsler). Hvis du har lagt en eksport manuelt et andet sted med gem-dialogen, er den der og ikke i denne mappe. Praktisk til oprydning eller for at se efter, hvor meget plads tidligere test-eksporter optager. Hvis du har brug for et komplet overblik inklusive logs og scene-bundles, så tag i stedet Manage Storage (M39).

M39 Help > Manage Storage...

Menulinjen → Help → Manage Storage...



Åbner storage-browseren (se kapitel 4 Auxiliary Windows, ID'er W7–W12). Lister alle persisterede scener, training-logs, eksporter og caches i `~/Documents/RadianceKit/-mappen` med størrelse, muliggør reveal-in-Finder og move-to-trash pr. post.

KORT FORTALT

Åbner en vindue-browser, der viser dig, hvor meget plads RadianceKit optager på din disk — pr. scene, log og eksport. Du kan slette enkelte ting direkte uden at skulle gå til Finder. Praktisk efter længere brug, når disken bliver fuld — tidligere logs og auto-test-eksporter kan summe op til flere gigabyte. Via reveal-in-Finder kommer du altid også til den klassiske visning.

M40 Help > Pareto Dashboard...

Menulinjen → Help → Pareto Dashboard... (⇧⌘D).



Åbner Pareto-dashboardet (se kapitel 4, ID'er W13–W22). Dashboardet indlæser alle JSONL-training-logs fra `~/Documents/RadianceKit/Logs/`, ordner dem efter scene og preset og tegner et Pareto-scatter-plot (standard: loss vs gaussians, valgfrit loss vs wallclock eller PSNR vs iterationer).

KORT FORTALT

Åbner et overblik over alle hidtidige trænings-kørsler som diagram. Du ser straks, hvilken kørsel der har leveret den bedste balance mellem kvalitet og størrelse. Praktisk, hvis du vil sammenligne forskellige presets med hinanden. Som standard viser diagrammet loss mod gaussian-antal — du kan dog også skifte til wallclock-tid eller PSNR. Dataene kommer fra JSONL-training-loggene (M37); jo flere kørsler du har, desto mere udsigelig bliver evalueringen.

M41 Help > Holdout Analysis...

Menulinjen → Help → Holdout Analysis... (⇧⌘H).



Åbner holdout-analyse-vinduet (se kapitel 4, ID'er W23–W29). Indlæser en `transforms.json`, tegner kameraerne som 3D-globe og tillader train/test-fold-splits (angulært eller lineært, 2–8 folds). Output er en `fold-assignment.json`, som træningen kan bruge i de respektive training-configs som test-sæt.

KORT FORTALT

Hjælper dig med at opdele dine kamera-optagelser i trænings- og test-sæt — så du objektivt kan måle, hvor god din scene er (på billeder, som træningen ikke har set). Snarere et forsknings- og benchmark-værktøj. Kameraerne vises som 3D-globe; du kan vælge mellem 2 og 8 folds, enten jævnt i vinklen eller lineært over rækkefølgen. Resultatet er en lille JSON-fil, som træningen så bruger som test-sæt.

M42 Help > BayesOpt Console...

Menulinjen → Help → BayesOpt Console... (⇧⌘B).



Åbner BayesOpt-konsollen (se kapitel 4, ID'er W30–W39). Indlæser foruddefinerede søgerum (f.eks. „MCMC scale-reg + opacity-reg + ssim“), kører bayesian-optimization-trials asynkront og viser konvergenskurve og trial-log live.

KORT FORTALT

En indbygget auto-tuner-konsol. I stedet for manuelt at prøve forskellige parametre af kan appen lade det køre selv natten over og foreslå dig de bedste værdier til din scene til sidst. Meget avanceret værktøj — til de fleste workflows er en god preset (se kapitel 7) tilstrækkelig. Du vælger et foruddefineret søgerum (f.eks. „MCMC scale-reg + opacity-reg + ssim“) og ser live konvergenskurven samt trial-loggen. Regn med flere timer til dage afhængigt af opsætningen.

Note: Cmd-Z i Edit-menuen

Siden maj 2026 understøtter Project Navigator i Expert Mode sletning af importerede billeder via minus-knap eller backspace-tast og fortryden via `Cmd-Z`. Denne `Cmd-Z`-handling optræder i macOS-Edit-menuen (som leveres af SwiftUI) som „Undo Remove Image“, så længe et slettet billede stadig kan gendannes. Den registreres via det standardkonforme system, ikke i ; derfor er der ingen egen M-ID-post i inventaret.

Tastaturgenveje i oversigt

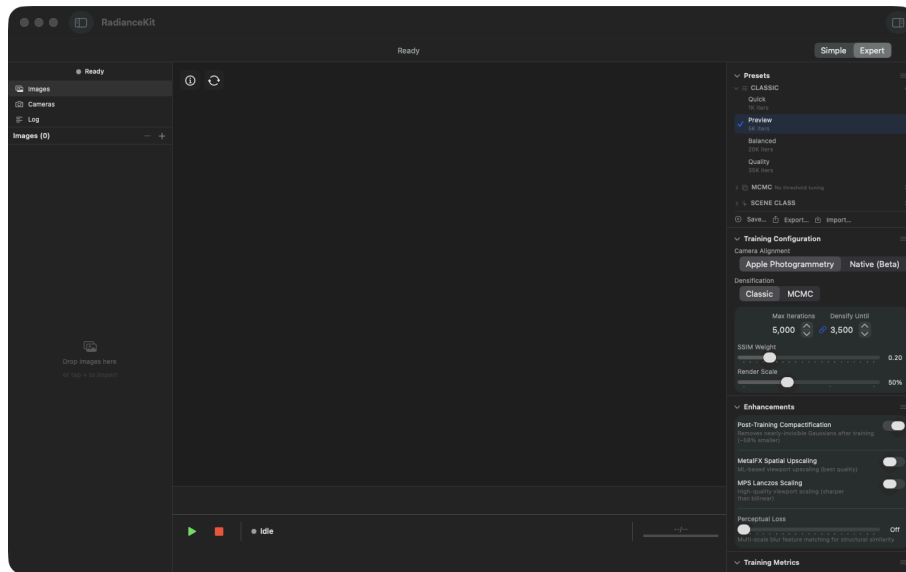
Menupost	Genvej
File > Open Scene...	⌘O
File > Save Scene...	⌘S
File > Import COLMAP / Metashape Workspace...	⇧⌘I
File > New Project	⇧⌘N
Mode > Simple Mode	⌘1
Mode > Expert Mode	⌘2
Training > Start Training	⇧⌘T
Viewport > Enter/Exit Edit Mode	⇧⌘E
Viewport > Toggle Auto-Rotation	⌘⌥T
Viewport > Save Screenshot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

Edit-menu (systemleveret, i Expert Mode ved aktivt Project-Navigator-valg):

Handling	Genvej
Undo Remove Image	⌘Z
Remove Selected Image	Backspace / Delete

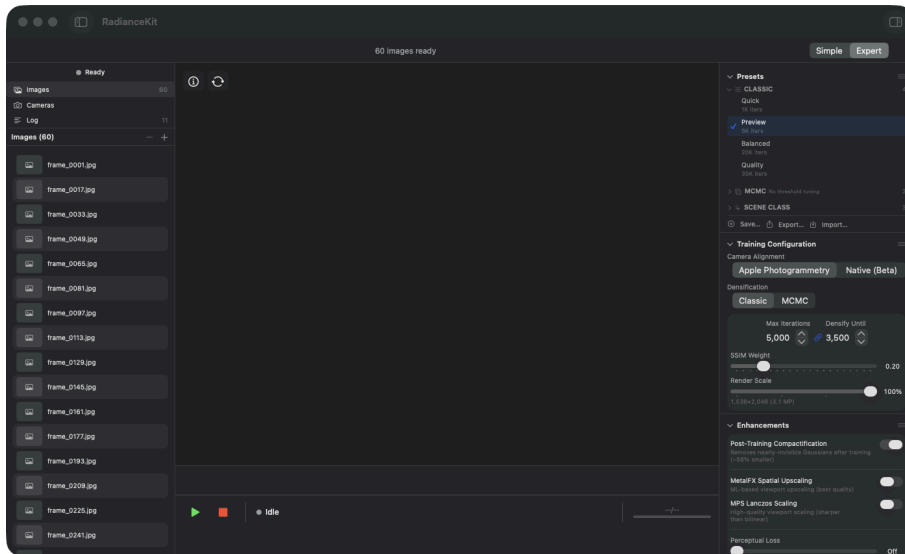
KAPITEL

Kapitel 2 — Inspector (Expert View)



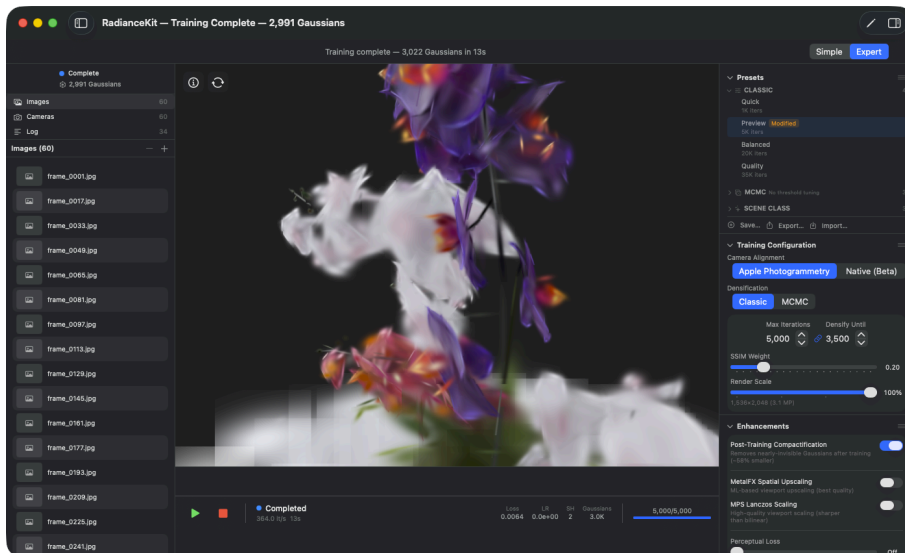
Figur 7: Eksperttilstand tom — Project Navigator til venstre (Images 0, Cameras, Log), tom viewport i midten, Inspector til højre med Presets/Training Configuration/Enhancements/Training Metrics-sektioner

Tom Inspector før import: Venstre sidebar viser images-counter 0 og drop-hint „Drop images here / or tap + to import“. Inspector til højre er fuldt funktionel, men presets er kun informative (ingen aktiv træning). Default-preset „Preview“ (5K iters) er markeret. Camera Alignment på Apple Photogrammetry, Densification Classic, SSIM Weight 0.20, Render Scale 50 %. Empty-states i Training Metrics („Start training to see live metrics“) og Loss History („Loss curve will appear during training“).



Figur 8: Inspector med 60 flowers-billeder indlæst — image-sidebaren viser de første filnavne frame_0001.jpg ff, header „60 images ready“

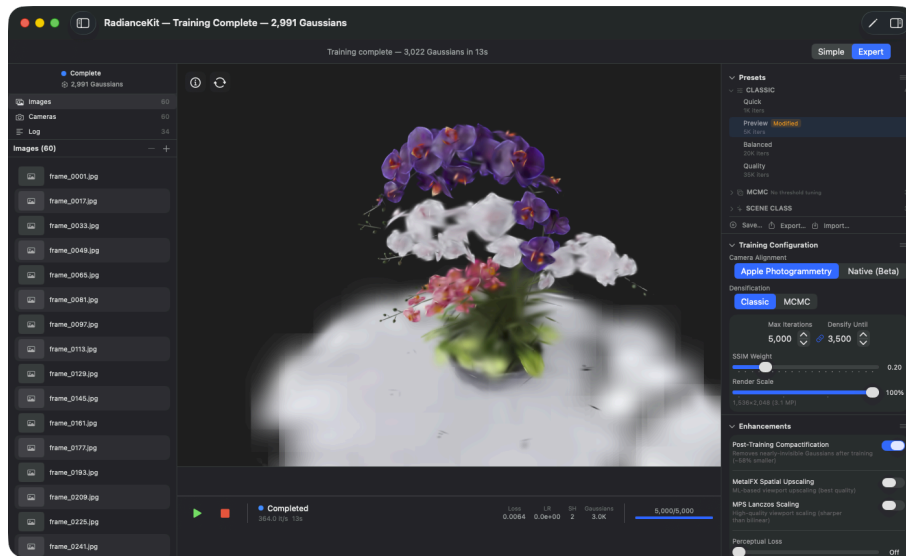
Inspector efter import: Header-status „60 images ready“. Image-sidebaren lister alle 60 importerede frames (frame_0001.jpg til frame_0945.jpg , hver 16. frame fra det 960-cam-buket-datasæt som subset til hurtige iterationer). Auto-render-scale-logikken tjekker billed-opløsningen ($1536 \times 2048 = 3.1$ MP) og tilpasser Render Scale derefter. Play-knappen (grøn, nederst til venstre) er nu aktiv og starter træningen med den aktive preset.



Figur 9: Inspector mid-training — live-viewport viser flowers-buket- rekonstruktion, metrik-bar nederst (Loss / LR / Gaussian-Count / Iterationer), preset-card „Preview“ med „Modified“-badge, hvis parametre er ændret

Inspector under træning: Titellinjen viser globalt fremskridt „RadianceKit — Training NN %“. Viewporten renderer den løbende gaussian-rekonstruktion i realtid (opdateret hver 50 iterationer — live-preview-interval kan justeres i Settings → General → Training → Live Preview). Metrik-bar under viewporten: aktuel loss, learning rate, gaussian-count og iterationer-counter (f.eks. 1.600/5.000 ved Preview-preset). Inspector-preset-

card „Preview“ bærer „Modified“-badge, så snart en parameter afviger fra den indbyggede default. Sidebaren „Log“ indsamler SfM- og training-stage-events.



Figur 10: Inspector efter trænings-afslutning — viewporten viser færdig flowers-buket-rekonstruktion (2.991 gaussians efter 5K iterationer på 13 s), titellinje „Training Complete — 2.991 Gaussians“

Inspector efter træning: Titellinjen viser endeligt gaussian-antal (her 2 991 — meget kompakt, fordi den syntetiske Blender-buket-scene på lys baggrund har enkel geometri). Viewporten viser den færdige punktsky — orbital drag-navigation aktiv (roterer omkring scenens midtpunkt). Training-Metrik-sektionen er nu fyldt med slutværdier, loss-history-charten viser forløbet af de samlede 5 000 iterationer. Eksport-sektionen nederst er nu aktiv (alle format-knapper enabled).

Inspectoren er den højre sidebar i Expert Mode (§2). Den samler alle træningsrelevante parametre i syv sammenklappelige sektioner. Default-rækkefølgen fra øverst til nederst ved første start er: Look, Presets, træningskonfiguration, metrikker, loss-diagram, Enhancements og Export. „Look“-sektionen (post-training-bildjusteringer) er den reelle UI-omdøbning af den tidligere „Finishing“-sektion — dens interne enum-rawValue forbliver af persistens-grunde „Finishing“, den viste overskrift hedder „Look“. Hver sektion kan klappes sammen ved klik på headeren, rækkefølgen kan omarrangeres via træk-og-slip (InspectorView.swift:81–97). **Ved første start er alle syv sektioner sammenklappet** (inspectorCollapsedSections defaulter til Set(InspectorSection.allCases)); app-state gemmer klap- og rækkefølge-præferencerne derefter på tværs af app-start.

En række betjeningslementer fra Inspectoren optræder også i næsten identisk form i indstillingerne (kapitel 3) — typisk SfM-backend, sky-masking og lignende defaults. Adskillelsen er bevidst: indstillingerne leverer den app-globale skabelon for nyligt oprettede projekter, Inspectoren overskriver disse værdier for det aktuelt åbne projekt. Den, der kender betjeningslogikken i den ene side, kan bruge den anden blindt.

Den venstre kolonne i Expert Mode — Project Navigator — hører ikke til Inspectoren, men er dens direkte nabo. Der kan importerede billeder vælges med klik, ses i Quick-Look med mellemrumstasten og slettes via minus-knappen eller delete-tasten (med Cmd-Z til at fortryde). Inspectoren følger det aktuelle sidebar-valg med kontekstspecifikke detalje-informationer, men de syv hovedsektioner forbliver altid tilgængelige.

Look-sektion (L1–L5)

Look-sektionen (intern `rawValue` fortsat „Finishing“) er den øverste Inspector-sektion og samler **post-training** -bildjusteringerne ét sted. Alle regulatorer arbejder **ikke-destruktivt**: Hver slider anvender `FinishingPass` på ny på et uforandret pristine-snapshot (original-DC-farve, -opacity, -skalering) — justeringen er dermed **idempotent**, ikke kumulativ. Resultatet vises **live i viewporten** (WYSIWYG, præcis som den senere eksport) og **bages ind i hver eksport**. Sektionen er først tilgængelig **efter afslutning af en trænings-kørsel** (forinden står „Available after a training run completes.“); dens værdier **nulstilles ved hver ny træning**. Så længe en eksport kører, er alle regulatorer **låst** — et lock-hint „Locked while exporting — the file uses the current settings.“ vises, og GroupBox'en er disabled.

L1 Saturation-slider

HVOR

Inspector → Look-sektion → GroupBox → Saturation.

TEKNISK

Slider 0.5–1.2, visning tocifret (f.eks. „1.00“). Skaleringer SH-DC-chromaen for hver Splat omkring luminans-værdien: 1.0 = uforandret, < 1.0 = afmættet (farve trukket mod gråtone), > 1.0 = kraftigere. Matematisk regnes DC-farven tilbage fra pristine-snapshottet (`desaturateDC`), så gentaget skubning ikke summerer op. Blev valideret på DJI-drone-materiale (Pensford-viadukt), der har tendens til at overtegne — drone-default ligger ved 0.82. Virker kun på farve-basen (SH-grad 0), højere SH-koefficienter forbliver urørte.

KORT FORTALT

Hvor kraftige farverne på den færdige Splat er. 1.00 lader alt stå som trænet, værdier derunder trækker farven mod grå — godt til drone- eller video-materiale, der ofte kommer overmættet ud. Værdier over 1.0 gør det kraftigere. Du kan skubbe frem og tilbage så meget du vil, uden at noget „skruer sig op“, fordi appen altid regner forfra fra det uforandrede original-stade. Synligt live i viewporten og præcis sådan i eksporten.

L2 Splat length-slider



Inspector → Look-sektion → GroupBox → Splat length.



Slider 0.3–1.0, visning tocifret. Trækker de tre skalerings-akser for hver Gaussian i log-rummet hen mod deres middelværdi (`shortenScale` , faktor `alpha`): 1.0 = uforandret, mindre værdier gør aflangte „nåle“-Splats rundere, 0 ville være rene kugler. Angriber nål-agtige, overstrakte Splats uden at ændre den samlede størrelse og reducerer derved typiske „konfetti“-artefakter. Anvendt fra pristine-snapshottet (original-log-skalering), derfor idempotent. Kommuterer med Splat size (L3), fordi begge arbejder i log-rummet.

KORT FORTALT

Gør overlange, splintrede Splats rundere. 1.00 lader formen stå som trænet, lavere værdier støver de langstrakte „nåle“ til rundere klatter — det beroliger kornede rekonstruktioner plaget af konfetti-artefakter. Den samlede størrelse forbliver den samme, det handler kun om aflangheden. Lader sig kombinere risikofrit med Splat size (L3).

L3 Splat size-slider



Inspector → Look-sektion → GroupBox → Splat size.



Slider 0.5–2.0, visning tocifret. Skalerer hver Gaussian på **alle** tre akser uniformt (`sizeScale`): 1.0 = uforandret, < 1.0 = mindre/tættere/skarpere, > 1.0 = større/„fluffigere“ (fylder huller mellem Splats). Da skaleringerne ligger i log-rummet, realiseres multiplikationen som en additiv `log(factor)` -offset — det kommuterer med Splat length (L2), fordi en konstant offset lader afvigelsen-fra-middelværdien være urørt. Fra pristine-snapshottet, altså idempotent. Ny i denne version.

KORT FORTALT

Skalerer alle Splats ensartet større eller mindre. 1.00 er det trænedede stade, værdier derunder gør punktskyen tættere og skarpere, værdier derover dækker huller mellem Splats (virker blødere/„fluffigere“). Praktisk til at lukke en hullet rekonstruktion optisk eller omvendt frilægge mere detalje. Forliges problemfrit med Splat length (L2) — de to regulatorer påvirker ikke hinanden.

L4 Fade far region (med sub-slidere)

HVOR

Inspector → Look-sektion → GroupBox → Toggle „Fade far region“ plus sub-slidere „Fade start xradius“ og „Fade floor“.

TEKNISK

Toggle, der aktiverer et radiale opacity-fald med afstanden fra kamera-tyngdepunktet — de svagt observerede „far-konfetti“ i baggrunden blændes ud. **Kun til orbit-optagelser:** Toggle'en er disabled, hvis `finishingContext.fadeEligible` er false (lineære flyvninger, for få eller degenererede kameraer); så vises i stedet for sub-slidere hintet „Far-fade applies only to orbit captures (not this scene).“ Egnetheden bestemmes via azimuth-dækningen af kamera-positionerne (et orbit kredser om tyngdepunktet og fylder mange kompas-sektorer, en lineær flyvning kun ~2). To sub-slidere styrer geometrien: **Fade start xradius** (1.0–3.0) sætter den indre radius som multiplum af orbit-radius, inden for hvilken fuld opacity gælder; **Fade floor** (0.0–1.0) er opacity-faktoren langt hinsides fade-radius. Vigtigt: Faden **springer sky-dome-området over** (de frozen Gaussians med indeks [0, frozenCount)), så den bevidste baggrunds-kuppel ikke dæmpes med.

KORT FORTALT

Blænder de svampede rester ved scenens yderkant ud — præcis de „far-konfetti“-klumper, der svæver langt bagud ved rundt-om-optagelser. Fungerer kun ved ægte orbit-/omkredsningsoptagelser; ved lige drone-flyvninger eller for få kameraer er kontakten grånet ud, og et hint forklarer hvorfor. Er den aktiv, kommer to finregulatorer til: „Fade start xradius“ fastlægger, fra hvilken afstand (som multiplum af omkreds-radius) udblændingen begynder, „Fade floor“ hvor stærkt de fjerne Splats stadig forbliver synlige til sidst (0 = helt væk, 1 = uforandret). En bevidst rekonstrueret sky-dome (I44) røres aldrig ved undervejs — himlen bevares.

L5 Reset finishing-knap

HVOR

Inspector → Look-sektion → GroupBox → „Reset finishing“ (nederst, lille knap).

TEKNISK

Nulstiller alle Look-settings til defaults (`FinishingPass.Settings() = Saturation 1.0, Fade fra, Splat length 1.0, Splat size 1.0`) og udløser straks en fornyet finishing, så viewporten springer tilbage til det uforandrede trænedede stade. `controlSize(.small)`. Da hele Look-stakken regner idempotent fra pristine-snapshottet, er „tilbage til default“ præcis det oprindelige trænings-output — intet kvalitetstab ved gentaget frem og tilbage. Som alle sektionens regulatorer låst under en igangværende eksport.

KORT FORTALT

Stiller med ét klik alle Look-regulatorer tilbage til standard (Saturation 1.00, Fade fra, begge Splat-slidere på 1.00) — viewporten viser derefter igen præcis det friskt trænedede resultat. Praktisk, hvis du har leget dig væk og vil begynde rent forfra. Fordi appen altid regner fra originalstaden, er der ikke noget kvalitetstab ved det. Mens en eksport kører, er knappen (ligesom sliderne) låst.

Presets-sektion (I1–I11)

Presets-sektionen er den hurtigste måde at anvende en afprøvet konfiguration. Indbyggede presets (Capture Class, Classic, MCMC, Hybrid) leverer reproducerbare startpunkter fra 560+ dokumenterede eksperimenter; egne presets kan gemmes, eksporteres, importeres og deles. Listen er grupperet efter kategorier (Capture Class, Classic, MCMC, Hybrid, Custom), og mere end én kategori kan være foldet ud samtidigt. Via kontekstmenu-mekanismen (højreklik på en linje) er eksport, duplikering og — ved egne presets — sletning tilgængelig.

I1 Save...-knap



Inspector → Presets-sektion → Save...-knap (action-bjælke nederst).



Åbner en popover med tekstfelt og save-/cancel-knapper. Den aktuelle TrainingConfig-tilstand persisteres som ny brugerdefineret preset (JSON-kodet, gemt på tværs af appen). Save-operationen kopierer alle 81 trænings-parametre plus den aktuelle densification-strategi. Presetet lander automatisk i kategorien Custom, uafhængigt af hvilken indbygget preset det blev afledt fra. Tomme navne og rene whitespace-input forkastes. Allerede eksisterende navne afvises ikke — hvert preset har sit eget interne ID, dobbelte navne er teknisk tilladt, men praktisk forvirrende.

KORT FORTALT

Sikrer din aktuelle konfiguration som genbrugelig preset. Tryk på knappen, skriv et navn i popoveren og klik Save — alle 81 parametre inklusive densification-strategi lander under det valgte navn i Custom-kategorien. Brug det, hvis du har gjort dig umage og ikke vil sidde og famle forfra ved næste projekt. Særligt praktisk til tilbagevendende opsætninger som „Drone 4K“ eller „Indendørs hurtig“. Dobbelte navne er teknisk tilladt, men praktisk forvirrende — vælg hellere noget sigende.

I2 Preset Name TextField



Save-popover → tekstfelt „Preset Name“.



Simpelt tekstfelt med afrundet ramme, bred form. Værdien overtages som preset-navn ved klik på save-knappen. Ingen længdebegrænsning i UI'en, men det gemte navn skal være JSON-kodbart og visningsbart i UI-listerne — emoji og specialtegn fungerer. Indholdet nulstilles automatisk til en tom streng ved åbning af popoveren. Save-knappen forbliver deaktiveret, så længe feltet efter trim er tomt. Der er ingen auto-suggest og ingen forudfyldning med navnet på den aktuelt aktive preset.

KORT FORTALT

Her taster du navnet til din preset. Vælg noget sigende som „Drone 4K 30fps“ eller „Indendørs hurtig“ — det hjælper dig senere med at finde det igen i Custom-kategorien. Emoji og specialtegn er tilladt, en hård længdebegrænsning er der ikke. Så længe feltet er tomt eller kun indeholder mellemrum, forbliver save-knappen grånet ud. Ved fornyet åbning af popoveren er feltet tomt igen — der er ingen forudfyldning med det aktive preset-navn.

I3 Cancel-knap (save-dialog)



Save-popover → Cancel-knap (til venstre).



Lukker popoveren uden at gemme. Forkaster tekstfeltets indhold — ved næste åbning nulstilles det igen til tomt via save...-knap-logikken (I1). Standard button-stil, ingen bekræftelsesdialoger, ingen hotkeys. Den aktuelle TrainingConfig forbliver uændret, da save-stien slet ikke blev udført.

KORT FORTALT

Lukker save-popoveren uden at gemme noget. Hvis du har skiftet mening, taster forkert eller åbnede dialogen ved et uheld — bare klik Cancel. Din aktuelle trænings-konfiguration forbliver uændret, fordi der endnu ikke er skrevet noget. Ved næste åbning af popoveren starter navnefeltet tomt igen. Ingen sikkerhedsspørgsmål, ingen hotkey — bare klik og væk.

I4 Save-knap (save-dialog)



Save-popover → Save-knap (til højre, prominent stil).



Udløser den egentlige persistering. Validerer endnu en gang ikke-tomt navn (defensiv check) og skriver derefter den aktuelle TrainingConfig som JSON ind i app-lageret. Lukker derefter popoveren. Blå fremhævet, grånet ud så længe tekstfeltet er tomt. Hvis gemningen mislykkes (f.eks. fordi app-lageret er fuldt — meget usandsynligt), er der i øjeblikket ingen synlig fejl-dialog; presetet ville så bare ikke optræde ved næste app-start.

KORT FORTALT

Med klik på Save overtager du navnet og skriver din aktuelle opsætning væk som ny preset. Popoveren lukker, presetet optræder straks i Custom-kategorien af preset-listen og kan fra nu af aktiveres ved klik. Knappen er blå fremhævet (`borderedProminent`) og forbliver grånet ud, så længe navnefeltet er tomt. Hvis gemningen mislykkes (f.eks. UserDefaults fuld), er der ingen synlig fejl-dialog — presetet ville så bare mangle ved næste app-start.

I5 Export...-knap



Inspector → Presets-sektion → action-bjælke → Export...-knap.



Eksporterer det aktuelt valgte preset som `.radiancepreset` -fil (internt JSON). Deaktiveret, hvis intet preset er valgt. Ved klik åbner appen en save-dialog med forudgivent filnavn (preset-navn + `.radiancepreset` -extension). Det gemte format indeholder den komplette TrainingConfig plus meta-data (navn, kategori, ID, built-in-flag). Dobbeltklik i Finder åbner appen — men **ikke** automatisk importen; brugeren skal bruge import-knappen (I6).

KORT FORTALT

Vælg en preset i listen og klik Export — så kan du gemme den som `.radiancepreset` -fil og f.eks. sende den til en kollega eller overføre den til en anden Mac. Modtageren indlæser den derovre med Import...-knappen (I6). Fungerer lige godt for built-ins og dine egne custom-presets. Knappen er grånet ud, så længe der intet er klikket i listen. Tip: via kontekstmenuen (I8) går det endnu hurtigere — der behøver du ikke vælge presetet først.

I6 Import...-knap



Inspector → Presets-sektion → action-bjælke → Import...-knap.



Åbner en fildialog, der kun tillader `.radiancepreset` -filer (multivalg deaktiveret). Ved valg indlæses JSON-filen, valideres og indsættes i Custom-kategorien — med nyt internt ID, så der ikke opstår kollisioner med built-ins. Importen sætter automatisk kategorien til Custom, selv hvis det eksporterede preset oprindeligt var en built-in. Beskadigede eller inkompatible filer med ældre skema-version afvises i stilhed uden fejl-dialog (konsol-log giver dog besked).

KORT FORTALT

Indlæs en `.radiancepreset` -fil fra disk. Nyttig, hvis nogen sender dig en afprøvet opsætning, eller hvis du selv vil holde dine yndlings-presets synkrone på tværs af flere Mac'er. Importerede presets lander altid i Custom-kategorien — også hvis de oprindeligt blev eksporteret fra built-ins. Beskadigede eller forældede filer ignoreres i stilhed; i konsol- loggen står så grunden. Multivalg i dialogen er deaktiveret, så kun én fil pr. klik.

I7 Preset-linje (klik-aktivering)



Inspector → Presets-sektion → hver preset-linje i hver kategori.



Klik på en preset-linje erstatter alle felter i TrainingConfig med værdierne fra presetet, husker ID for det aktive preset og nulstiller modified-status. Aktiv-fluebenet foran linjen optræder kun, når presetet er valgt OG umodificeret. Så snart en værdi i TrainingConfig ændres (slider, stepper, toggle i de andre Inspector-sektioner), optræder et orange „Modified“-badge bag navnet. Indbyggede presets kan ikke overskrives — ved modifikation skal en egen kopi oprettes via save-knappen (I1).

KORT FORTALT

Klik på en linje aktiverer presetet og overtager alle dér gemte værdier i de aktuelle trænings-indstillinger. Fluebenet foran navnet viser, hvilken preset der er aktiv lige nu. Så snart du derefter justerer en slider, stepper eller toggle i de andre sektioner, optræder et orange „Modified“-badge bag navnet — fordi din opsætning nu afviger fra presetet. Built-in-presets kan ikke overskrives; hvis du vil beholde ændringer, så opret en egen kopi via Save...-knappen (I1) eller dupliker presetet (I9).

I8 Kontekstmenu „Export...“



Højreklik på hver preset-linje → første post „Export...“.



Identisk funktionalitet som I5 (Export...-knap), men mere bekvemt tilgængelig — uden at presetet skal vælges først. Eksporterer direkte det preset, der blev klikket på i linjen. Fungerer ens for alle preset-kategorier (built-in eller Custom), ingen begrænsning. Eksporten indeholder built-in-flag og original-kategori, men ved re-import mappes kategorien som beskrevet under I6 til Custom.

KORT FORTALT

Hurtig vej til eksport — højreklik på det ønskede preset og vælg „Export...“. Sparer omvejen via klik-først og så tryk på Export...-knappen. Fungerer ens for alle kategorier, også for built-ins. Den genererede `.radiancepreset`-fil er identisk med den fra I5; ved senere re-import lander den automatisk i Custom-kategorien.

I9 Kontekstmenu „Duplicate“



Højreklik på hver preset-linje → anden post „Duplicate“.



Kloner presetet ind i Custom-kategorien. Genererer et nyt internt ID, vedhæfter „Copy“ til navnet og gemmer kopien. Fungerer også for built-in-presets — klonen er så redigerbar. Originalen forbliver urørt. TrainingConfig kopieres værdi-for-værdi (JSON-rodtrip), så der ikke består referenceforbindelser mellem original og kopi.

KORT FORTALT

Skaber en redigerbar kopi af et preset i Custom-kategorien. Praktisk, hvis du f.eks. vil have det indbyggede „Quality“-preset som udgangspunkt og så bare lige flytte SSIM-sliden en smule. Workflow: duplicate, omdøb (kontekstmenu eller ny Save...-kørsel), tilpas, færdig. Originalen forbliver urørt — du kan altid vende tilbage til den. Fungerer også for built-ins, hvilket er den eneste måde at overtage deres værdier som basis og samtidig gøre dem redigerbare på.

I10 Kontekstmenu „Delete“



Højreklik på egne preset-linjer → sidste post „Delete“ (rød, destruktiv).



Kun synlig for custom-presets. Built-ins kan ikke slettes. Posten er markeret som destruktiv, vises rød i kontekstmenuen og placeres efter en divider, så den ikke klikkes ved et uheld. Der findes **ingen** bekræftelsesdialog — et klik sletter presetet med det samme. Det slettede preset kan ikke gendannes (Cmd-Z virker ikke her — undo findes i den aktuelle build kun for billedlisten, ikke for preset-operationer). Var det slettede preset netop aktivt, forbliver den aktuelle TrainingConfig uændret, kun det aktive preset-valg nulstilles.

KORT FORTALT

Slette egne presets. Ved built-ins (Quick, Preview, Balanced, Quality, Ultra Detail, Drone / Aerial, 360° Walkaround, Photo / Object osv.) er „Delete“ slet ikke synlig — dem kan du ikke komme til at slette ved et uheld. Advarsel: der er ingen sikkerhedsspørgsmål og ingen undo, ét klik og presetet er væk. Hvis du er i tvivl, så træk først en sikkerhedskopi på disk via Export... (I5/I8) — den kan du altid importere igen. Var presetet netop aktivt, forbliver din TrainingConfig uændret, kun fluebenet forsvinder.

I11 Kategori-header (udfold/sammenklap)



HVOR

Inspector → Presets-sektion → hver kategori-header (Capture Class, Classic, MCMC, Hybrid, Custom).



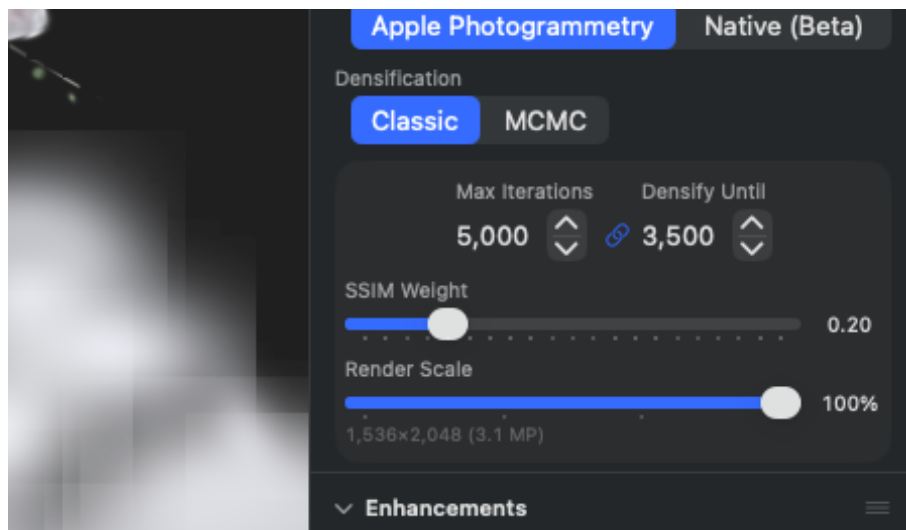
TEKNISK

Klap-status pr. kategori med forskellig default: den kuraterede Capture Class-gruppe starter **foldet ud**, Classic, MCMC, Hybrid og Custom starter **sammenklappet**. Status persisteres ikke — ved app-genstart er alle kategorier igen i default-tilstand. Chevron-pilen roterer animeret. Tallet til højre i headeren viser antallet af presets i denne kategori. Klik-hit-området omfatter hele headerområdet.

KORT FORTALT

Folde kategorier ind og ud for at holde preset-listen overskuelig. Ved app-start er Capture Class-gruppen åben, Classic, MCMC, Hybrid og Custom er lukkede. Klik på headeren (hele området er klikbart), og listen kører op eller i med en kort chevron-animation. Det lille tal til højre viser, hvor mange presets der ligger i kategorien. Efter genstart af appen er default-tilstanden tilbage — appen gemmer bevidst ikke denne klap-indstilling.

Trænings-konfigurations-sektion (I12–I22)



Figur 11: Crop kun training-konfigurations-sektion — Camera Alignment (Apple Photogrammetry aktiv, Native (Beta) inaktiv), Densification (Classic aktiv), Max Iterations 5.000 / Densify Until 3.500 med link-symbol, SSIM Weight slider 0.20, Render Scale slider på 100 % (1.536×2.048 = 3.1 MP)

Her lander de centrale håndtag: hvilket SfM-backend der skal beregne, hvordan densification arbejder, hvor mange iterationer, hvor stor SSIM-vægtningen er. Ved MCMC-strategi optræder to ekstra toggles („MCMC Quality“ og „Auto-scale by scene“), som skjules i Classic-tilstand. Ved Native-SfM-backend kommer FOV-override-feltet til, som kun behøves til video-frames uden EXIF-brændvidde.

I12 Camera Alignment-vælger

Inspector → træningskonfiguration → Camera Alignment (segmenteret vælger øverst).



Segmenteret vælger med to muligheder: Apple Photogrammetry og Native (Beta). Valget bestemmer det anvendte SfM-backend ved næste kamera-rekonstruktion. Det påvirker samtidigt, hvilke andre Inspector-elementer der er synlige: Native viser desuden FOV-override (I13), der kun behøves ved EXIF-løse video-frames. Bemærk: ved meget store outdoor-optagelser kan du indlæse resultatet fra et eksternt værktøj (Metashape eller COLMAP) via workspace-import — se kapitel 1 (M5) og kapitel 9 (Q3, Q6).

KORT FORTALT

Her vælger du, hvordan kamera-positionerne rekonstrueres — den vigtigste kontakt for slutkvaliteten. Apple Photogrammetry er den hurtige standard og rækker fuldt ud til de fleste objekt-scanninger. Native (Beta) er den App Store-konforme egenudvikling, god til orbits og turntable-scener, og kræver ved EXIF-løse video-frames FOV-override (I13). Ved meget store outdoor-sæt kan du alternativt beregne kameraerne i Metashape eller COLMAP og indlæse resultatet via workspace-importen. Detaljer og anbefalinger pr. scenetype finder du i kapitel 9.

I13 FOV Override-felt (Native SfM)

Inspector → træningskonfiguration → FOV Override (kun synlig ved Camera Alignment = Native).



Numerisk tekstfelt (range 0-170°), default 0 = automatisk bestemmelse fra EXIF eller heuristik. Manuel indtastning er nødvendig, hvis input-billederne er udtrukket fra en video, der ikke indeholder brændvidde-metadata. Typiske værdier: iPhone Wide ≈ 73°, DJI Mavic Wide-Crop ≈ 70°, drone med fuldformatsensor ≈ 84°. Værdien klampes til [0, 170] — værdier udenfor staves direkte tilbage. Påvirker kun den native SfM-pipeline (Q4/Q5); Apple Photogrammetry ignorerer denne værdi helt.

KORT FORTALT

Hvis dine billeder ikke har EXIF (typisk ved udtrukne video-frames), så indtast her kameraets horisontale synsfelt i grader. Tommelfingerværdier: iPhone Wide ≈ 73°, DJI Mavic Wide-Crop ≈ 70°, drone med fuldformatsensor ≈ 84°. Et 0 lader appen gætte selv — det går ofte godt, men kan gå skævt ved sjældne optikker. Værdier over 170° staves automatisk tilbage. Feltet er kun synligt og kun virksomt, hvis du har valgt Native som Camera Alignment (I12) — Apple Photogrammetry ignorerer det helt.

I15 **Densification-vælger**

Inspector → træningskonfiguration → Densification (segmenteret vælger, altid synlig).



Skifter mellem de to densification-strategier: Classic (original 3DGS-metode med clone/split/prune og gradient- tærskel) og MCMC (Stochastic Gradient Langevin Dynamics med relocation, NeurIPS 2024). Ved skift fra Classic til MCMC sætter appen automatisk MCMC-specifikke felter til afprøvede default- værdier (reg-weights = 0, MCMC-cap-multiplier 3.0, sample-/ noise-schedule). Uden denne automatiske initialisering led sessioner med gamle presets under 1.4.4-MCMC-collapse-bug'en (460K→5 gaussians, watchdog-kill). Vælger-valget bestemmer desuden, hvilke Inspector-elementer der er synlige — ved MCMC optræder I16/I17. Detaljeret felt-virkning i kapitel 6, T11–T16 (Classic) og T61–T73 (MCMC).

KORT FORTALT

Det centrale strategivalg for, hvordan gaussian-antallet vokser. Classic er godt tunet fra 459 eksperimenter, leverer hurtige resultater af høj kvalitet og behøver ikke at kende MCMC-felterne. MCMC er den nyere tilgang (NeurIPS 2024), mere reproducerbar og undviger manuel threshold-justering — til gengæld regner den ca. 6× længere ved sammenlignelig kvalitet. Ved skift til MCMC sætter appen automatisk sikre defaults, så træningen ikke ender i 1.4.4-collapse. Detaljer om strategifelterne står i kapitel 6 (T11–T16 Classic, T61–T73 MCMC).

I16 **MCMC Quality-toggle**

Inspector → træningskonfiguration → MCMC Quality (kun ved Densification = MCMC).



Slår gradient-accumulation til 2 trin (aktiv) eller 1 trin (inaktiv). Akkumulerer gradienterne fra to på hinanden følgende kamera-views, før optimizer-step udføres. Empirisk (session 33, V544a) reducerer det den endelige L1-fejl med ca. 6 % (0.0246 med Quality vs 0.0261 uden, ved 3-trial-gennemsnit på Horse-Full-MCMC). Prisen: fordoblet træningstid. Ved meget lange træninger (200K iterationer) fører det til yderligere 10+ minutters ventetid — så kun værd, hvis de sidste par procent kvalitet virkelig er nødvendige. Påvirker kun træningen, ikke eksportformatet eller viewport-visningen.

KORT FORTALT

Quality-mode for MCMC med gradient-accumulation over to views. Gør slutresultatet empirisk ca. 6 % bedre (L1 0.0246 i stedet for 0.0261 i Horse-testen), koster til gengæld dobbelt så lang tid. Hvis du alligevel køber en 200K-MCMC-træning (gerne 2 timer), kommer der endnu en knap time oveni. Værd ved endelige showcase-renderinger eller mod slutningen af en quality-sweep-session, i daglig workflow snarere ikke. Kun synlig, når Densification står på MCMC (I15).

I17 Auto-scale by scene-toggle



HVOR

Inspector → træningskonfiguration → Auto-scale by scene (kun ved MCMC).



TEKNISK

Hvis aktiv, skales den effektive max-gaussians-overgrænse med SfM-init-point-count × MCMC-cap-multiplier (default 3.0). Eksempel: SfM leverer 250K initpunkter, basis-cap = 150K, multiplier 3.0 → effektiv overgrænse = $\max(150K, 750K) = 750K$. Hvis deaktiveret, gælder strengt kun basen. Blev indført til v1.4.5, fordi store outdoor-optagelser med over 1000 frames og tilsvarende høj SfM-punkt-densitet med den stive 150K-cap-default udsultede densifikationen — overflødige punkter blev tilbage, nye fik ikke lov at opstå. Default OFF i custom-presets, ON i MCMC-built-ins. Påvirker kun til træningstidspunkt, ikke i eksporten.

KORT FORTALT

Lader maksimumstallet af gaussians vokse med scenestørrelsen (mere præcist: med antallet af SfM-initpunkter). Ved små scener mærker du næsten ingen forskel, ved store outdoor-scener er det ofte afgørende for kvaliteten — ellers „kvæles“ træningen, fordi default-overgrænsen på 150K er alt for lav til scenen. Blev specifikt indført til v1.4.5, efter at meget store outdoor-sæt (over 1000 frames) synligt hang fast på cap'et. Ved MCMC-built-in-presets allerede slået til på forhånd; i egne presets slukket som default.

I18 Max Iterations-stepper



HVOR

Inspector → træningskonfiguration → GroupBox → Max Iterations.



TEKNISK

Stepper med range 1 000–100 000, skridtstørrelse 1 000. Bestemmer det samlede antal optimizer-iterationer. Lineært korreleret med træningstiden (halvering = ca. 50 % tid). Empiriske sweet-spots: 20K (Classic Balanced, $L1 \approx 0.028$), 40K (Classic Quality, $L1 \approx 0.023$), 200K (MCMC Full, $L1 \approx 0.0246$). Over 40K ved Classic giver i gennemsnit næppe forbedring — diminishing returns. Ved ændring trækkes Densify Until proportionalt med, hvis link-funktionen (I19) er aktiv (default-ratio: 0.5, dvs. Densify-Until = $\text{Max}/2$).

KORT FORTALT

Hvor mange trænings-skridt der køres — mere er bedre, men koster også lineært mere tid. Tom-fingerregel: 20 000 for god kvalitet, 40 000 for optimum ved Classic-strategi (derover giver det i gennemsnit næppe mere). MCMC kræver markant mere, 200 000 er her standard. Fordobling af iterationer fordobler groft træningstiden. Ved aktiv link-knap (I19) trækkes Densify Until proportionalt med — praktisk altid det, du vil.

I19 Link/Unlink-knap (Densify ↔ Iterations)



Inspector → træningskonfiguration → GroupBox → lille link-knap mellem Max Iterations og Densify Until.



Toggle-knap, der fryser forholdet mellem Densify Until og Max Iterations. Ved aktiv (link-ikon fremhævet) trækkes Densify Until proportionalt med ved hver ændring af Max Iterations. Ved unlink (link-plus-ikon) forbliver værdierne uafhængige. Default er linked, fordi det afspejler den typiske korrelation — hvis du trækker træningen til dobbelt iterationer, vil du som regel også lade densificationen køre proportionalt længere. Forholdet beregnes ved indstilling af link-knappen ud fra den aktuelle værdi; et typisk forhold er 0.5 (Densify-Until = halve iterations-antal).

KORT FORTALT

Lille klamme-knap mellem Max Iterations og Densify Until. Når aktiv (link-ikon fremhævet), vandrer de to værdier sammen — fordobler du iterations, fordobles også Densify Until i samme forhold. Hvis ikke (link.badge.plus -ikon), kan du sætte dem uafhængigt. Standard er linket, fordi det afspejler den typiske korrelation — længere træning kræver som regel også længere densification-fase. Til 99 % af tilfældene lad det være låst.

I20 Densify Until-stepper



Inspector → træningskonfiguration → GroupBox → Densify Until.



Stepper med range 500–50 000, skridtstørrelse 500. Bestemmer iterations-indekset, hvorfra der ikke længere kommer nye gaussians til via clone/split (Classic) eller relocation (MCMC). Efter opnåelse forfines kun position og farve. Højere værdier = flere gaussians = større fil, længere pr.-iteration-tid (+30-60 % GPU-tid pr. skridt). Typiske værdier: 15K (til 30K max-iter), 20K (til 40K), 100K (til 200K MCMC). Ved aktiv link (I19) skaleres automatisk med. Virker forskelligt ved Classic vs MCMC: Classic stopper væksten helt, MCMC stopper relocation-logikken, men sample-/noise-adaptation kører videre.

KORT FORTALT

Op til hvilken iteration nye gaussians må tilføjes — ved Classic via clone/split, ved MCMC via relocation. Derefter handler det kun om farve- og form-forfining af de eksisterende punkter. Højere = mere detalje, men også større fil og +30-60 % GPU-tid pr. skridt. Typiske værdier: 15K (til 30K max-iter), 20K (til 40K), 100K (til 200K MCMC). Hænger normalt via link (I19) på Max Iterations — sjældent meningsfuldt at afkoble det manuelt.

I21 SSIM Weight-sliden



Inspector → træningskonfiguration → GroupBox → SSIM Weight.



Slider 0.0–1.0 i 0.05-skridt, visning som „0.20“.
Blander L1-loss (0.0) og SSIM-loss (1.0). L1 strammer lysstyrken pr. pixel, SSIM den strukturelle lighed (kanter, lokale statistikker). Default 0.2 er værdien fra det oprindelige 3DGS-paper (Kerbl 2023) og reverse-engineered som robust kompromis i talrige sessioner. Højere værdier (0.5+) foretrækker detaljebevarelse, men kan ignorere lokale lysstyrkefejl. Lavere værdier (< 0.1) fører til detaljetab ved skarpe kanter. SSIM-beregningen kører i shaderen med et 11×11-gaussian-vindue. Performance: ved 0.0 (kun L1) er træningen ca. 8-12 % hurtigere, fordi SSIM-beregningen i shaderen springes over.

KORT FORTALT

Hvor stærkt strukturel billedlighed (kanter, lokale mønstre) vægtes i forhold til ren lysstyrkesammenligning. 0.2 er standarden fra det oprindelige 3DGS-paper og rækker til næsten alle scener. Højere (0.5+) ved fine strukturer som hår, pels eller vegetation — der hjælper mere strukturvægt. Lavere (0.0) gør træningen ca. 8-12 % hurtigere, fordi SSIM-beregningen i shaderen springes over, men koster detalje ved skarpe kanter. Den, der ikke har en god grund til ændring, lader 0.2 stå.

I22 Render Scale-sliden



Inspector → træningskonfiguration → GroupBox → Render Scale.

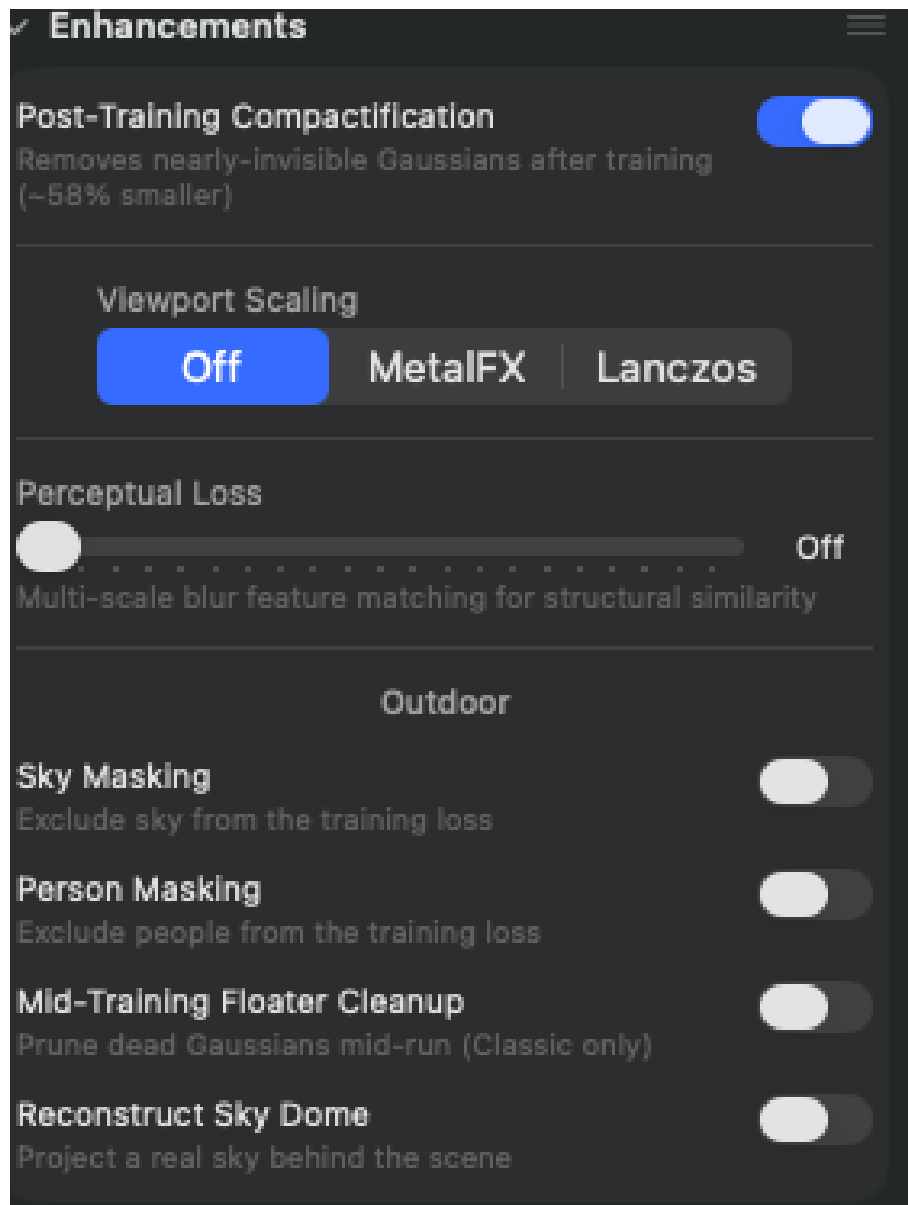


Slider 0.25–1.0 i 0.25-skridt, visning som „100 %“.
Skalerer training-rendering-opløsningen relativt til kildebillede-størrelsen. Største håndtag på performance: 50 % reducerer GPU-tid med ca. 75 % (fordi 4× færre pixel), 25 % med ca. 94 %. Gradient-tærsklen skales automatisk med. Under slideren optræder en live-opløsnings-visning i MP (f.eks. „2304×1296 (3.0 MP)“). Hvis den aktuelle værdi afviger fra den anbefalede, indkobles i orange skrift „— recommended: 50 %“. Anbefalingen sigter på ~3 MP effektiv opløsning — det område, Apple Silicon GPU'er behandler mest effektivt. 4K-kildebilleder får f.eks. automatisk 25 % anbefalet, FullHD-billeder 100 %. En ændring udløser desuden buffer-reallokationen.

KORT FORTALT

Med hvilken opløsning træningen renderer — et af de største performance-håndtag. Fuld (100 %) giver bedste kvalitet, men koster ved store billeder meget GPU-tid. Halv (50 %) sparer ca. 75 % GPU-tid, fordi der beregnes fire gange færre pixel — perfekt til 4K-kilder. Under slideren ser du den effektive opløsning i megapixel; appen sigter på rundt 3 MP, fordi det kører mest effektivt på Apple Silicon. Hvis din værdi afviger fra det, indkobler appen et orange „recommended“-hint — som regel kan det betale sig at følge.

Enhancements-sektion (I26–I29, I42–I44)



Figur 12: Crop kun Enhancements-sektion — tre rækker: Post-Training Compactification (toggle til), Viewport Scaling (segmenteret vælger Off/MetalFX/Lanczos), Perceptual Loss (slider på „Off“). Hver række med undertitel forklarer funktionen

Enhancements-sektionen grupperer tre features, der forbedrer billedkvalitet uden at ændre selve kerne-trænings-loopet. De første to (I26-I27) er **post-training-** eller **viewport-trin**: compactification rydder op efter trænings-slut, viewport-skaleringen er en ren viewport-renderer, der ikke påvirker den løbende træning. Perceptual Loss (I29) er trods sektion-tilhørsforholdet en træningsbestanddel — den aktiveres under træningen som ekstra loss-term, deraf adskillelsen fra viewport-regulatorerne via en divider.

Fra og med v1.6 har sektionen desuden en Outdoor-gruppe (I42–I44: Sky Masking, Mid-Training Floater Cleanup, Reconstruct Sky Dome) — trænings-muligheder mod himmel-floaters, som tidligere lå i indstillingsvinduet og nu sidder her pr. projekt.

I26 Post-Training Compactification-toggle **HVOR**

Inspector → Enhancements → Post-Training Compactification.

 **TEKNISK**

Aktiverer V443-post-processing: efter afslutningen af trænings-iterationerne slettes gaussians med opacity under 0.01 (1 % synlighed). Empirisk reducerer det filstørrelsen med ~55-58 % ved nul synligt kvalitetstab — fordi disse gaussians visuelt alligevel ikke bidrager. Compactificationen kører som GPU-compact-pas og varer afhængigt af gaussian-count brøkdele af sekunder til få sekunder. Påvirker ikke trænings-performance. Hvis denne toggle er slukket, eksporteres også usynlige gaussians — kun relevant, hvis du vil bruge formatet til endnu et trænings-stage (Continue Training), ellers spild af lagerplads.

 **KORT FORTALT**

Rydder op efter træningen i gaussians, du alligevel ikke kan se (opacity under 1 %). Gør eksport-filerne omtrent halvt så store (55-58 % størrelsesreduktion) uden synligt kvalitetstab. Kører som kort GPU-pas efter sidste iteration, varer kun brøkdele af sekunder til få sekunder. Bør praktisk altid være tændt — den eneste grund til at slukke det er, hvis du senere vil fortsætte træningen via Continue Training og skal beholde også usynlige gaussians. Ved normale eksport-workflows lad det bare stå.

I27 Viewport Scaling-vælger **HVOR**

Inspector → Enhancements → Viewport Scaling (segmenteret vælger med tre muligheder: Off, MetalFX, Lanczos).

 **TEKNISK**

En enkelt segmenteret vælger, der vælger viewport-upscaleren — de tre muligheder er **gensidigt udelukkende**. Hvis trænings-opløsningen (via I22 Render Scale) er lavere end viewport-størrelsen, skalerer den valgte mode det rendrede frame op til visningsstørrelsen. **Off** = simpel bilinear strækning. **MetalFX** = Apples ML-baserede Spatial Upscaler, den skarpeste mulighed (ML-modellen er optimeret til skarpe kanter), overhead ca. 1-2 ms pr. frame på M3-GPU'er. **Lanczos** = Apples Metal Performance Shaders med 8-tap-sinc-resampling, klassisk uden ML, minimal overhead (< 0.5 ms), kvalitet under MetalFX, men uden ML-typisk „blødgøring“ af fine linje-strukturer. Renderer-pipelinen omkonfigureres live ved skift — synlig med det samme, uden genstart. **Baggrund:** Tidligere var det to separate toggles (MetalFX + Lanczos), der kunne være tændt samtidigt — en modstridende tilstand, hvor MetalFX stille gik forud for Lanczos. Vælgeren fjerner den tilstand; en eventuel „begge-tændt“-tilstand arvet fra ældre sessioner heler sig selv til MetalFX ved næste skift. Påvirker **kun** live-viewporten, ikke rendrede eksporter (orbit-video, screenshots) — de rendres i fuld kildeopløsning.

 **KORT FORTALT**

Skærper live-billedet op i viewporten — især nyttigt, hvis du arbejder med reduceret trænings-opløsning (Render Scale 50 %, se I22). Tre trin, hvoraf kun ét er aktivt ad gangen: „Off“ strækker bare pixlerne, „MetalFX“ bruger Apples machine-learning til de skarpeste kanter (praktisk altid det bedste valg), „Lanczos“ er det klassiske filter uden ML — tag det som fallback, hvis MetalFX i en scene glatter linjer ud eller viser artefakter. Griber live, uden genstart. Virker kun i live-viewporten, ikke på eksporterede orbit-videoer eller screenshots — de rendres i fuld kildeopløsning. Modsat tidligere kan du ikke længere ved et uheld vælge to modes samtidigt.

I29 Perceptual Loss-slider HVOR

Inspector → Enhancements → Perceptual Loss.

 TEKNISK

Slider 0.0–0.2 i 0.01-skridt, visning ved 0.0 som „Off“, ellers som „0.05“ osv. Aktiverer en ekstra loss-term, der sammenligner multi-skaleret gaussi-an-blur af renderingen med ground-truth-billedet (3 blur-skalaer). Fanger strukturelle forskelle, som L1+SSIM alene ikke genkender. V460-implementering. Empirisk forbedrer en værdi på 0.05-0.1 L1-scoren i sessioner med et par procent, men koster ~5 % træningstid (ekstra forward-pas gennem blur-kernerne). Over 0.15 bliver træningen ustabil, og L1 forværres igen (loss-term'en dominerer optimeringen). Virker **under** træningen, ikke i post-processing — trods placering i „Enhancements“-sektionen er det altså ikke en ren opgradering bagefter.

 KORT FORTALT

En ekstra loss-andel, der tjekker strukturel billedlighed via tre forskellige uskarphedstrin. Hjælper især ved scener med fine strukturer som hår, stof eller vegetation, fordi den fanger mønstre, L1+SSIM alene ikke ser. Mindre værdier er sikrere — 0.05 til 0.1 er sweet spot, over 0.15 bliver træningen ustabil, og loss'en forværres igen. Ved 0 (Off) er funktionen helt slukket og koster intet; aktiv sluger den ca. 5 % træningstid til det ekstra forward-pas gennem blur-kernerne. Virker trods „Enhancements“-sektionen direkte under træningen, ikke først i post-processing.

I42 Sky Masking

HVOR

Inspector → Enhancements (Outdoor-gruppe) → Toggle „Sky Masking“. Bound: `AppState.trainingConfig.skyMaskingEnabled` (pr. projekt, `@DefaultFalse`). Default: `false` .

TEKNISK

Aktiverer pre-training Apple-Vision-baseret sky-pixel-segmentering. Før trænings-start ekstraheres sky-regionen for hvert input-kamera via Apple-Vision-Foreground-Mask (Sky = Background) og tilknyttes det pågældende kamera som pr.-pixel-maske. Under træningen multipliceres loss-bidraget pr. pixel med komplementet af sky-masken — sky-pixel bidrager 0 til gradienten, så gaussians, der projicerer ind i himlen, ikke modtager optimerings-signaler og dermed ikke bliver „tættere“ eller „lyse-re“. Reducerer floaters (mørke klumper i himlen) ved outdoor-/drone-scener signifikant. Koster ~3% L1-regression ved klassisk 40K-træning (se [memory/dev_outdoor-floater-reduction.md](#)). Kun meningsfuld ved outdoor-scener med klart genkendelig himmel; ved indendørs scener eller hvid baggrund identificerer sky-segmenteringen forkerte områder og blokerer valide loss-signaler. Værdien gemmes nu pr. projekt (ikke længere app-globalt) og følger pre-setet / scene-filen.

KORT FORTALT

Ved outdoor-optagelser med himmel i billedet opstår der ofte sorte eller farvede klumper i himlen — såkaldte „floaters“. Denne mulighed registrerer automatisk, hvor himlen er, og siger til træningen: „Lad himlen være i fred.“ Fungerer rigtig godt ved drone-flyvninger og landskabsscener. Ved indendørs rum eller mørke baggrunde kan det forringe billedet — så tænd kun, hvis der faktisk er rigtig himmel at se. Detaljer: [memory/dev_outdoor-floater-reduction.md](#).

I43 Mid-Training Floater Cleanup



Inspector → Enhancements (Outdoor-gruppe) → Toggle „Mid-Training Floater Cleanup“. Bound: `AppState.trainingConfig.floaterCleanupEnabled` (pr. projekt, `@DefaultFalse`). Default: `false`.



Aktiverer ved klassisk 40K-træning (preset „P4 Quality“) to ekstra density-control-passes: ved iteration 20,000 og ved iteration 30,000. Begge passes gennem søger alle gaussians efter tre kriterier: (a) meget lav opacity (standard 0.005), (b) lille screen-space-størrelse, (c) ingen loss-bidrag i de seneste 1000 iterationer. Gaussians, der opfylder alle tre betingelser, purges. Effekt: ~5–15% færre gaussians ved trænings-slut, synligt færre mørke klumper i himlen ved drone-/outdoor-scener. Koster ~1–3% L1-regression ved nærbillede-indoor-scener, derfor ikke aktiveret som default. Værdien huskes på tværs af genstart (i modsætning til I42). De to cleanup-iterationer (20K, 30K) er hårdt definerede og kan p.t. ikke ændres via UI; ved kortere træninger (f.eks. P2 Preview 5K) har toggle'en ingen effekt, fordi den aldrig når iterations-mærkerne. **Nyt:** Toggle'en kan kun betjenes, når det aktive preset bruger **Classic**-densifieren (`densificationStrategy == .classic`). Ved MCMC eller Hybrid bliver den **disabled**, og et inline-hint vises, fordi disse strategier behandler døde Gaussians selv (MCMC via relocation, Hybrid via kombineret reloc-/noise-logik) — de manuelle cleanup-passes ville der være virkningsløse eller kontraproduktive. Code-referenc: `RadianceKitApp.swift`, General-fanen. Detaljer: `memory/dev_outdoor-floater-reduction.md`.

KORT FORTALT

Under træningen opstår der nogle gange „døde“ gauss-punkter, som ikke længere bidrager til billedkvaliteten, men optager hukommelse. Denne mulighed rydder op to gange under en lang træning (ved 20K og 30K iterationer) og fjerner disse lig. Ved outdoor-scener med himmel er det særligt nyttigt, fordi det er der, de fleste floaters samles. Ved små træninger eller nærbilleder af møbler er det snarere ikke nødvendigt. Kontakten kan kun tændes, hvis dit preset bruger Classic-densifieren — ved MCMC- eller Hybrid-presets er den grånet ud (med en kort forklaring), fordi de selv rydder op i deres døde punkter.

I44 Reconstruct Sky Dome **HVOR**

Inspector → Enhancements (Outdoor-gruppe)
→ Toggle „Reconstruct Sky Dome“. Bound:
AppState.trainingConfig.skyDomeEnabled (pr. pro-
jekt, @DefaultFalse). Default: false .

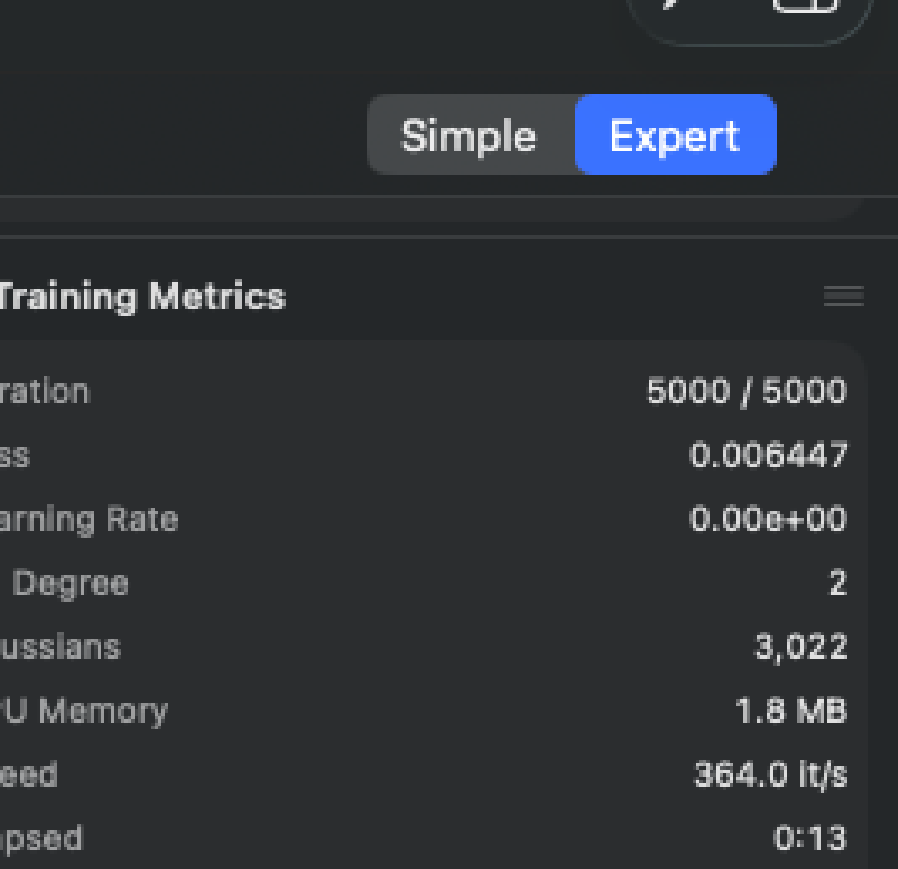
 **TEKNISK**

Aktiverer pre-training sky-dome-projektionen (V549e MVP). Efter SfM og før trænings-start ekstraheres for hvert input-kamera den i I42 fælles brugte Apple-Vision-sky-maske fra billedet, sky-pixlene af-projiceres med kamera-intrinsics på en virtuel kugleoverflade (standard-radius $8 \times$ scene-radius). På denne kugle initialiseres ~5000 nye gaussians med farve-middelværdier fra de projicerede sky-pixler, meget stor skalering (1.0 i scene-enheder) og start-opacity 0.95. Disse 5000 gaussians er ikke en sky-mask i klassisk forstand — de trænes som alle andre, men holdes via den høje start-opacity i et tyndt skal. Resultat: ved 360° -novel-views i outdoor-/drone-scener fremstår faktiske himmelfarver og skyformationer i stedet for mørke konfetti-klumper. Værdien huskes på tværs af genstart. Kun meningsfuld ved outdoor-scener med mindst 360° -kameradækning; ved rene object-captures uden himmel har det ingen effekt. Status: eksperimentel, bredere A/B-validering på tværs af yderligere outdoor-sæt udestår.

 **KORT FORTALT**

I stedet for at træningen forsøger at „gætte“ himlen ud fra de få synlige pixler (hvilket fører til floaters), projicerer RadianceKit himmel-pixlerne direkte på en virtuel kugle omkring scenen, før træningen starter. Når du så drejer den færdige scene i 360° , ser du rigtig himmel i stedet for sorte klumper. Fungerer kun ved outdoor-optagelser, hvor der faktisk er himmel i billedet. Ved stue-scanninger eller studie-optagelser bringer det intet.

Metrik-sektion (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

Figur 13: Crop kun Training Metrics-sektion efter afsluttet træning på buket (5K iterationer, 2 991 gaussians endelig) — tabel med trænings-metrikker (Iteration, Loss, SSIM Loss, Combined Loss, Gaussian Count, Learning Rate, Elapsed, ETA)

Mens en træning kører, viser metrik-sektionen ni live-værdier fra trænings-loopet. Før start af en træning er sektionen tom („Start training to see live metrics“). Alle værdier opdateres hver ~30 iterationer (opdaterings-frekvens for streamen). Sektionen er read-only — intet element kan klikkes eller ændres. Til dybere analyse trækkes JSONL-trænings-loggene under `~/Documents/RadianceKit/Logs/` ind (script `python3 scripts/analyze_logs.py best 5`).

I30 Iteration

Inspector → Metrikker → Iteration. Read-only.



Visning i formatet „4523 / 40000“ — aktuel iteration over total planlagte iterationer. Tæller synkront med trænings-loopet, der pusher værdierne hver ~30 iter. Det andet tal svarer til max-iterations-værdien på start-tidspunktet; det ændres ikke længere, selv hvis brugeren justerer stepperen derefter — den løbende kørsel bruger sin egen snapshot-kopi. Hvis appen via training-menuen lægger iterationer på (Continue Training +5K/+10K/+20K), øges nævneren.

KORT FORTALT

Hvor træningen lige nu står. „4523 / 40000“ betyder: 4523 af 40 000 skridt er igennem, altså ca. 11 % færdig. Tallet til venstre tæller op i sekundtakt; hvis det står stille i minutter, hænger træningen — som regel et tegn på GPU-throttling eller en konkurrerende app. Tallet til højre svarer til max-iterations-værdien (I18) ved træningens start og ændres ikke længere, selv hvis du justerer stepperen senere. Ved Continue Training (+5K/+10K/+20K) vokser den med de ekstra skridt.

I31 Loss

Inspector → Metrikker → Loss. Read-only.



Float-værdi med seks decimaler (f.eks. „0.024385“). Måler den kombinerede L1+SSIM-loss (blanding kontrolleres via I21 SSIM Weight) plus valgfrit Perceptual Loss (I29) og andre regularizere. Skalaen er ikke absolut, men scene-afhængig — kræver for de fleste sammenligninger samme datasæt. Typiske slutværdier ved gode konfigurationer:

- 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 (geometri-betinget dårligere)
- Indendørs lejligheder: 0.018–0.025

Værdier over 0.10 efter 5K iterationer indikerer SfM-problemer (dårlige kamera-positioner) — afbryd og lav SfM på ny.

KORT FORTALT

Hvor langt det rendrede billede stadig afviger fra originalen — kombineret af L1, SSIM og evt. Perceptual Loss. Mindre er bedre. Under 0.03 er som regel rigtig godt, under 0.05 stadig okay, outdoor-scener ligger geometribetinget snarere ved 0.03–0.06. Over 0.10 efter flere tusinde iterationer er et advarselssignal — som regel skyldes det kamera-rekonstruktionen (SfM har ikke klappet pænt). Skalaen er ikke absolut, men scene-afhængig; lav kun sammenligninger inden for samme datasæt. Hvis tallet pludselig springer opad, er der som regel sket en gradient-explosion-event.

I32 Learning Rate



Inspector → Metrikker → Learning Rate. Read-only.



Visning i scientific notation (f.eks. „1.60e-04“). Aktuell læringsrate for position-parametrene (3DGS har seks uafhængige LR'er for position, SH-DC, SH-rest, opacity, scale, rotation — her vises position-LR'en som repræsentativ værdi). Default-startværdi 1.6e-4, der over en exponential-decay synker ned til ~1.6e-6 i træningens slutning. Forfaldet kan justeres via LR-schedule-feltet i trænings-konfigurationen (T-felt i kap. 6). Hvis LR'en bliver usædvanlig høj (f.eks. 1e-3 eller mere efter 10K iterationer), kan det indikere en fejlindlæst konfiguration.

KORT FORTALT

Hvor store optimerings-skridtene er lige nu — konkret læringsraten for gaussian-positionerne. Starter ved 1.60e-04 og synker eksponentielt til ca. 1.60e-06 ved trænings-slut („1.60e-06“ = 0.0000016). Forløbet kører automatisk, du behøver ikke justere her. Hvis værdien efter 10 000+ iterationer stadig er større end 1e-3, er der sandsynligvis indlæst en fejlagtig config — afbryd træning og vælg preset på ny. Internt har 3DGS seks uafhængige læringsrater (position, SH-DC, SH-rest, opacity, scale, rotation); her ser du kun position-LR'en som stedfortræder.

I33 SH Degree



Inspector → Metrikker → SH Degree. Read-only.



Heltal 0-3. Spherical-harmonics-grad for farverepresentationen. Begynder ved 0 (kun DC-komponenten, dvs. retnings-uafhængig farve pr. gaussian — altså bare en RGB-konstant) og stiger progressivt til 3 i træningens forløb. Standard-schedule hæver graden ved 1000/2000/3000 iterationer med 1 hver. SH-3 svarer til 48 farve-koefficienter pr. gaussian (3 RGB-channels × 16 SH-basisfunktioner). Højere SH-grad = mere retnings-afhængig refleksion (blanke overflader ser korrekt forskellige ud under forskellige synsvinkler), men også mere lager og langsommere træning.

KORT FORTALT

Hvor kompleks farvefremstillingen pr. gaussian er lige nu. Starter ved 0 (kun en retnings-uafhængig farve pr. punkt) og trækkes trinvis op til 3 — typisk ved iteration 1000, 2000 og 3000. Trin 3 betyder 48 farve-koefficienter pr. gaussian og tillader retnings-afhængige refleksioner, altså at blanke overflader fra forskellige synsvinkler ser korrekt forskellige ud. Du behøver ikke aktivt røre ved det, schedulen kører automatisk. Højere grad koster mere lager og forsinker træningen en smule — men det er prisen for realistiske højlys.

I34 Gaussians



HVOR

Inspector → Metrikker → Gaussians. Read-only.



TEKNISK

Aktuelt antal gaussians i modellen, formateret med locale-separator (f.eks. „524.318“). Vækst:

- Classic: starter ved SfM-init-punkterne (typisk 50K-300K), vokser via clone/split indtil kort før Densify Until, derefter statisk indtil trænings-slut (modulo pruning)
- MCMC: sample-punkter tilføjes indtil MCMC-cap'et, derefter kun relocation

Healthy slutværdier:

- Classic Quality: 400K-700K (Horse 524K, Garden 800K)
- MCMC Full: præcis på cap'et (default 150K, med auto-scale multiplier × SfM-count alt efter scene 500K-1.5M)

Ved MCMC falder tallet til < 60 % af cap'et → anomali (collapse-indikator, peger på for aggressive regularizere).

KORT FORTALT

Hvor mange gaussian-punkter 3D-modellen har lige nu. Vokser under træningen, indtil Densify Until (I20) er nået; derefter forbliver tallet praktisk konstant. Flere punkter = mere detalje, men også større fil og langsommere rendering i viewporten. 500.000 gaussians er en typisk middelværdi for Classic-Quality på en mellemstor scene; MCMC Full lander alt efter auto-scale (I17) ved 500K til 1.5M. Hvis tallet ved MCMC pludselig falder under 60 % af cap'et, er det en collapse-indikator — som regel for aggressive regularizere.

I35 GPU Memory



HVOR

Inspector → Metrikker → GPU Memory. Read-only.



TEKNISK

Skøn af gaussian-bufferens hukommelsesforbrug som gaussian-count × 616 bytes (formateret i memory-style). 616 bytes er den empiriske størrelse af en fuldt udstyret gaussian (position, skalering, rotation, opacity, SH-koefficienter grad 3, gradient-akkumulator). Visningen fanger **ikke** renderer-overhead (tile-buffer, sort-buffer, backward-buffer) — det reelle GPU-lagerbehov ligger typisk 2-3× over denne værdi. Ved 500K gaussians: vist ~290 MB, reelt ~700 MB. Ved 1.5M gaussians: vist ~880 MB, reelt ~2.5 GB. På M3 Max med 64+ GB unified memory ukritisk, på M3 Pro med 18 GB allerede en grænse.

KORT FORTALT

Et skøn over, hvor meget GPU-lager gaussians selv optager — rundt 616 bytes pr. punkt. Det faktiske GPU-forbrug er 2-3× højere end vist, fordi rendereren også lægger egne tile-, sort- og backward-buffers til. På en MacBook med 16-18 GB unified memory bør du holde dig under 500K gaussians; med M3 Max eller Studio (64+ GB) kan du nemt køre 1.5M og mere. Hvis træningen pludselig crasher, eller systemet swapper, er grænsen som regel nået her — skru Render Scale (I22) ned eller reducer Densify Until (I20).

I36 Speed

Inspector → Metrikker → Speed. Read-only.



Iterationer pr. sekund med en decimal („24.3 it/s“). Beregnes af træneren som glidende gennemsnit over de seneste ~100 iterationer. Typiske værdier:

- Quick-preset (1K iters): 80-120 it/s (kort, ingen steady-state)
- Classic 20K @ 1.0 Render Scale (Truck-scene, 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 (langsommere pga. relocation)
- Ved 1M+ gaussians og fuld opløsning: < 10 it/s

Faldende speed i træningens forløb er normalt — flere gaussians = mere compute pr. iteration. Pludselige fald (f.eks. fra 30 → 5 it/s) indikerer GPU-thermal-throttling eller konkurrerende apps.

KORT FORTALT

Hvor hurtigt træningen kører, i iterationer pr. sekund. Står typisk ved 20-50 it/s, ved reduceret Render Scale (50 %) og små scener også 80-120 it/s. Falder gennem træningen ganske naturligt, fordi flere gaussians = mere arbejde pr. iteration. Pludselige fald (f.eks. 30 → 5 it/s) indikerer GPU-thermal-throttling eller konkurrerende apps — browser-faner med video, Time Machine-backup, Photos-indeksering. At holde appen i forgrunden og lukke baggrundsprogrammer hjælper ofte. Ved 1M+ gaussians og fuld opløsning er under 10 it/s normalt.

I37 Elapsed

Inspector → Metrikker → Elapsed. Read-only.



Allerede forløbet tid som „4:23“ (m:ss) eller „1:23:45“ (h:mm:ss). Format-switch fra 1 time. Måler kun den rene trænings-tid, ikke de forudliggende faser (SfM-beregning, billed-import). Ved pause/resume kører uret videre — det er altså wall-clock, ikke CPU-tid.

KORT FORTALT

Hvor længe træningen allerede har kørt, som ren stopur (wall-clock-tid). Format er „m:ss“ til en time, derefter „h:mm:ss“. Ikke „CPU-tid“, men „hvor længe har vi ventet“ — så også pause-tider tæller med. Måler kun den rene trænings-fase, ikke den forudliggende SfM-beregning eller billed-importen. Hjælpsom til sammenligning med ETA (I38) — hvis Elapsed markant overskrider den oprindelige ETA, er træningen blevet langsommere et sted end planlagt.

I38 ETA HVOR

Inspector → Metrikker → ETA. Read-only.

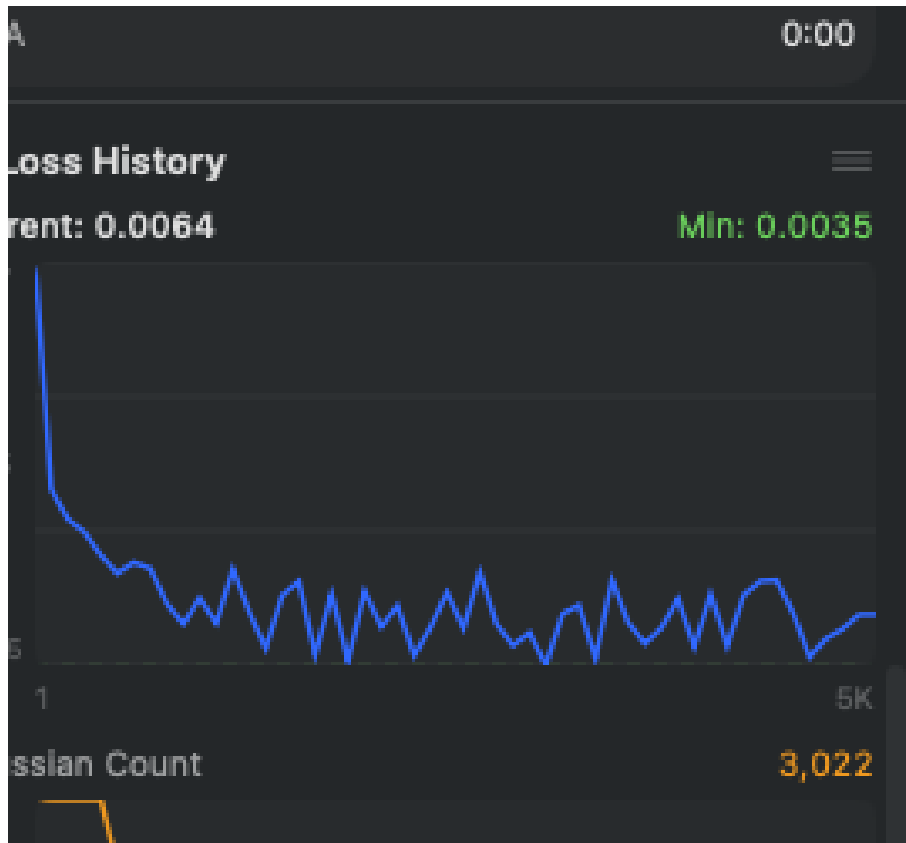
 TEKNISK

Estimeret resttid som „17:42“ eller „1:12:35“. Beregning: $(\text{Max Iterations} - \text{aktuel iteration}) / \text{iterationer-pr.-sekund}$. Viser „-“, når speed lige er nul (helt i starten eller ved pause). Estimatet tilpasses **ikke** den typiske forlangsomning mod trænings-slut — især ved MCMC og Classic med store Densify-Until-værdier har træningen tendens til at blive langsommere, fordi flere og flere gaussians kommer ind i billedet. Reelt bliver det typisk 10-20 % over start-ETA.

 KORT FORTALT

Hvor længe der formentlig stadig skal ventes — beregnet ud fra de tilbageværende iterationer og den aktuelle speed (I36). Et groft skøn: hvis Mac'en pludselig bliver langsommere (flere gaussians fra densify-fasen, thermal-throttling, andre apps), kan det være længere end vist. Appen indregner ikke den typiske forlangsomning mod trænings-slut, så det reelle slut lander som regel 10-20 % over start-ETA. Læg 15 % ekstra ind, så passer det som regel. Viser „-“, når speed lige er 0 (træningens start eller pause).

Loss-diagram-sektion (I39–I41)



Figur 14: Crop kun Loss History-sektion efter afsluttet træning — Current 0.0064, Min 0.0035 (grøn), blå forløb fra 0.027 (iteration 1) til 0.0035 (iteration 5K) med karakteristisk knæk omkring iter 200, derunder Gaussian Count-chart orange

Loss-diagram-sektionen visualiserer trænings-forløbet over tid. Den består af to charts: et loss-curve-chart (stort, øverst, blå) og et gaussian-count-chart (mindre, nederst, orange). Begge bygges op live under træningen og persisterer indtil næste trænings-start. Før første træning er området tomt („Loss curve will appear during training“). Charts er rene SwiftUI-path-tegninger (ingen Swift-Charts-framework), så de også renderer flydende ved 100K+ punkter.

I39 Current Loss (visning)

Inspector → loss-diagram → venstre label-område „Current: 0.0287“. Read-only.



Float-værdi af det sidste loss-sample-punkt, formateret med fire decimaler. Identisk med I31 (Loss i metrics-sektionen), bare her mere kompakt formateret. Kilden er loss-history — en liste, der får en post pr. ~30 iterationer. Kun endelige værdier optages i listen — NaN/Infinity (meget sjældent, i tilfælde af gradient-explosion-bug) filtreres.

 KORT FORTALT

Den aktuelle loss-værdi i kortere skrivemåde end i metrics-sektionen (fire decimaler). Indholdsmæssigt identisk med I31, men her sidder visningen direkte ved loss-charten og giver dig den præcise tal-værdi, mens du iagtager kurven. Opdateres som alle live-metrikker hver 30 iterationer. NaN- eller infinity-værdier (ekstremt sjældne ved gradient-explosion-bugs) filtreres appen automatisk fra. Nyttig for ikke at skulle hoppe til den anden sektion, mens du kigger på diagrammet.

I40 Min Loss (visning)

Inspector → loss-diagram → højre label-område „Min: 0.0245“ (grøn). Read-only.



Minimum af alle nogensinde sete loss-værdier i den aktuelle trænings-kørsel. Genberegnes live ud fra loss-history — ingen separat persistens. Vises med grøn skrift, fordi „Min“ = „best so far“. Den stiplede grønne linje ved chartens nederste rand markerer denne Y-position visuelt. Ved continue-training-sessioner starter min-springen forfra — den gamle history erstattes i UI'en af den nye (ikke vedhæftet). Hvis den aktuelle træning kører dårligere end den foregående, kan min-visningen altså være større end det forrige slutresultat.

 KORT FORTALT

Den laveste loss-værdi, denne træning hidtil har set — grønt vist, fordi „best so far“. Den stiplede grønne linje ved chartens nederste rand markerer også denne position visuelt. Hvis den aktuelle kurve netop ligger markant over, er der med lidt held endnu en forbedring; som regel er min dog det slutresultats-tegn, der senere interesserer dig. Ved continue-training-sessioner starter min-springen forfra, fordi den gamle history erstattes i UI'en af den nye — min-værdien kan dermed se dårligere ud end det forrige slutresultat.

I41 Gaussian Count Chart

HVOR

Inspector → loss-diagram → andet chart derunder (orange). Read-only.

TEKNISK

Linje-diagram af gaussian-antallet over trænings-iterationerne. Kilde: gaussian-count-history (liste af (iter, count)-par, fyldt af træneren hver ~30 iter). Y-skala dynamisk mellem minimum og maksimum af history'en. Ved Classic-strategi ser kurven typisk sådan ud: jævnt stigende indtil Densify Until, derefter flad (med små pruning-udsving). Ved MCMC: stejl stigning indtil cap, derefter horisontal linje (relocation holder tallet konstant). Hvis kurven **falder** trods aktiv træning, prunner densificationen for aggressivt — tegn på forkerte defaults eller en kendt MCMC-collapse-bug (v1.4.4-hotfix-tema).

KORT FORTALT

Hvordan antallet af gaussians udvikler sig over trænings-tiden — det mindre orange chart under loss-kurven. Ved Classic-strategi stiger linjen jævnt, indtil Densify Until (I20) er nået, derefter forbliver den flad med små pruning-udsving. Ved MCMC svirrer den stejlt op til cap'et og forbliver derefter horisontal, fordi relocation holder tallet konstant. Hvis kurven trods aktiv træning pludselig knækker nedad, er densificationen for aggressiv ved pruning — klassisk tegn på MCMC-collapse-bug'en fra v1.4.4. Så hjælper app-opdatering eller skift tilbage til Classic.

Hvordan læser man loss-kurven?

Loss-charten er det vigtigste diagnose-værktøj i Inspektoren — ingen anden indikator viser så direkte, om træningen skrider nyttigt frem eller hænger fast. Den typiske sunde form er et hurtigt fald i de første 1000-3000 iterationer (fra ~0.15 til ~0.05), efterfulgt af et langsomt, jævnt fald til trænings-slut (til 0.020-0.030). Logaritmisk virker kurven som en glat diagonal.

Hvad betyder et plateau ved loss'en? Hvis kurven over flere tusinde iterationer forbliver flad, er der to mulige læsninger: (a) træningen er „konvergeret“ — loss'en kan ikke længere falde signifikant, fordi modellen er så god, som den kan blive med de givne data og indstillinger. Det er ønsket; det er „færdig“. (b) Træningen „hænger“ — loss'en kan egentlig stadig falde, men optimeringen stagnerer (lokalt minimum, læringsrate for lille, densification slukket). At skelne: hvis loss-værdien ligger i et typisk godt område (0.020-0.030 ved indoor/object, 0.040-0.060 ved outdoor), og kurven har været flad i 5K iterationer, er det konvergeret. Hvis værdien er markant højere end ved sammenlignelige scener (f.eks. 0.08), hænger den fast.

Advarsel: gaussian-plateau ≠ loss-plateau. Et plateau i gaussian-antallet betyder **ikke** „træning er færdig“. Det betyder kun, at densificationen er holdt op med at tilføje nye punkter — enten fordi er nået (Classic), eller fordi MCMC-cap'et er fuldt. Træningen kører bagefter videre og forfiner kun de eksisterende punkter. Det egentlige „færdig“-signal læser du af loss-kurven og iteration-visningen (I30), ikke her.

Tommefingerregel til afbrydelse: Hvis loss-kurven efter 5000+ iterationer ligger over 0.08 og næppe falder mere, er SfM-rekonstruktionen med høj sandsynlighed skæv. Afbryd træning, slå op i kapitel 9, om det valgte SfM-backend passer til scenen, skift

evt. til COLMAP/Native, start så på ny. Bedre at investere 10 minutter i bedre SfM end 2 timers træning med dårlig kamera-justering.

Hvornår række ud efter Inspektoren?

Hurtig-reference: hvilken sektion + hvilke controls til hvilket typisk use-case?

Common-task	Sektion	Control-ID'er
Afmætte farverne på den færdige Splat	Look	L1 (Saturation)
Runde nål-/konfetti-Splats	Look	L2 (Splat length)
Fylde hullet sky / forstørre Splats	Look	L3 (Splat size)
Blænde fjerne „far-konfetti“ ud ved orbits	Look	L4 (Fade far region)
Forkaste Look-justeringer	Look	L5 (Reset finishing)
Indlæse forudfremstillet opsætning	Presets	I7 (klik på linje)
Gemme egen opsætning	Presets	I1 → I2 → I4
Dele opsætning med kolleger	Presets	I5 (eksport) eller I6 (import)
Skifte SfM-backend (f.eks. fordi Apple-PG er for ustabil)	træningskonfiguration	I12 (se kap. 9)
Behandle video-frames uden EXIF- brændvidde	træningskonfiguration	I13 (FOV-override)
COLMAP-performance: GLO-MAP i stedet for klassisk	træningskonfiguration	I14
Skifte fra Classic til MCMC	træningskonfiguration	I15
Lade træningen køre længere	træningskonfiguration	I18 (Max iter) + I20 (Densify Until) — koblet via I19
Halvere GPU-tid	træningskonfiguration	I22 (Render Scale til 50 %)
Trænings-kvalitet +6 % (MCMC)	træningskonfiguration	I16 (MCMC Quality)
Outdoor-scene med mange SfM- punkter	træningskonfiguration	I17 (Auto-scale by scene)
Opsætte / skifte COLMAP-sti	træningskonfiguration	I23 / I24 / I25
Gøre eksport-filer mindre	Enhancements	I26 (lad altid være tændt)
Skarpere viewport uden ekstra trænings-tid	Enhancements	I27 (Viewport Scaling → Metal-FX)
MetalFX glatter for meget → alternativ	Enhancements	I27 (Viewport Scaling → Lanczos)
Sidste kvantitet detalje ved fine strukturer	Enhancements	I29 (Perceptual Loss 0.05-0.1)
Overvåge træning	Metrikker	I30 (fremskridt), I36 (tempo), I37 (gennemsnitlig FPS), I38 (gennemsnitlig RAM) → SfM på ny
Se detaljer om SfM-proceduren	Metrikker	I39 (gennemsnitlig FPS) → SfM på ny

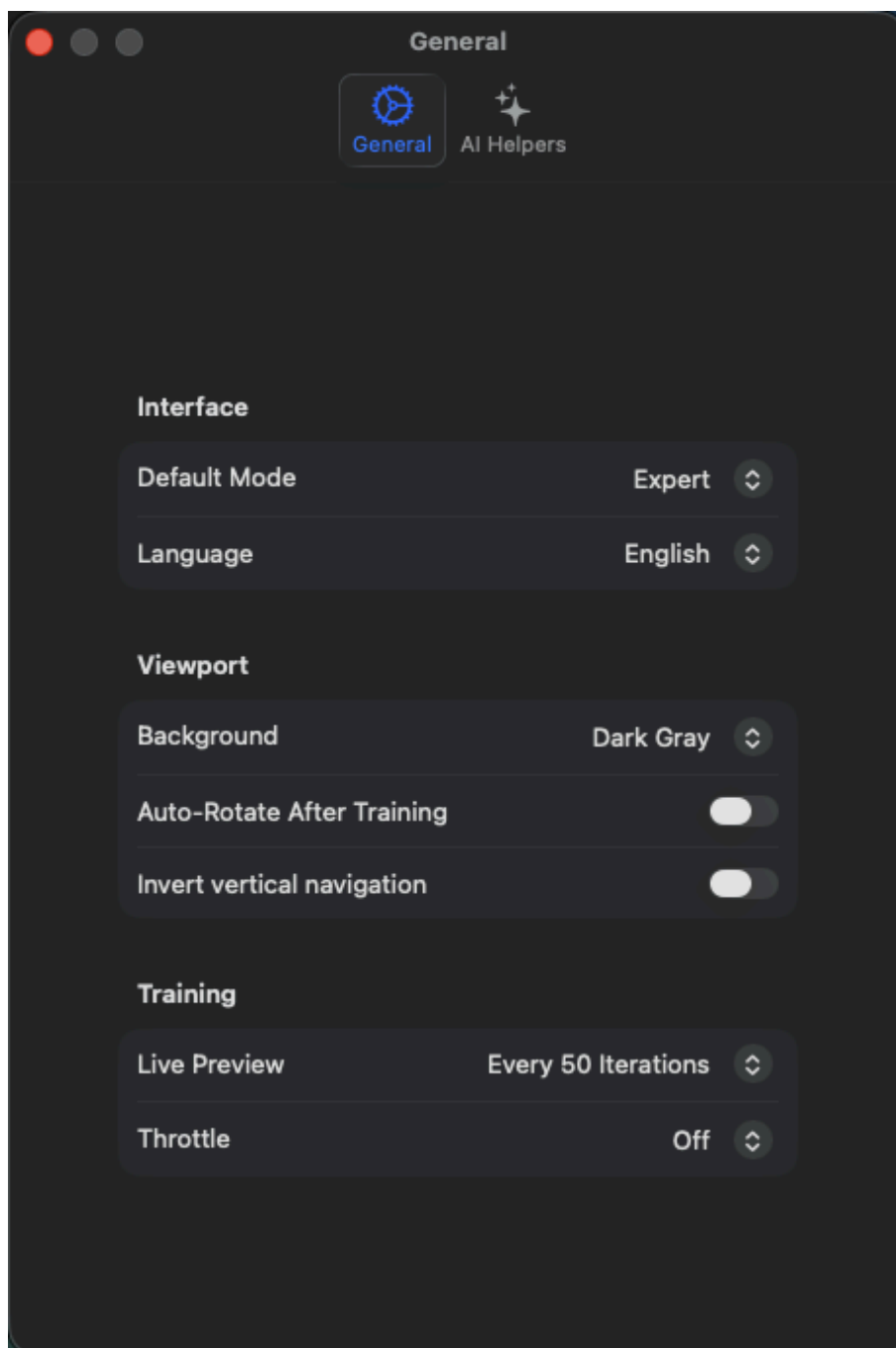
KAPITEL

Kapitel 3 — Indstillinger

Indstillingsvinduet åbnes via `RadianceKit` → `Indstillinger...` eller standardgenvejen `⌘, .` Det indeholder to faner: **General** og **AI Helpers**. I modsætning til Inspector-værdierne fra kapitel 2 virker indstillingerne fra dette vindue **app-globalt** (på tværs af alle projekter) — de persisteres og overlever app-genstart. General-fanen grupperer tre indholdsmæssige sektioner: Interface, Viewport og Training. (De tre Outdoor-Floater-kontakter — Sky Masking, Mid-Training Floater Cleanup, Reconstruct Sky Dome — som tidligere lå her, er fra og med v1.6 flyttet til Expert-Inspectorens Enhancements-sektion, hvor de nu gemmes pr. projekt; se kapitel 2, 142–144.) AI-Helpers-fanen tænder for on-device-machine-learning-hjælperne (Vision, CoreML) til SfM- og trænings-forbehandling.

Tidligere betjeningslementer til samlet aktivering eller deaktivering af alle AI-helpers findes ikke længere i den aktuelle version — de er derfor ikke dokumenteret her. Også det tidligere „Coming Soon“-område til endnu ikke leverede hjælpere er fjernet og refereres ikke her.

General-fanen



Figur 15: Indstillinger → General-fanen med brugerflade, viewport, training og experimental-sektion

S1 Default Mode

HVOR

Settings → General → Interface → Default Mode-vælger. Bound: Default: `.simple`.

TEKNISK

Styrer, hvilken af to UI-tilstande appen åbner i ved næste opstart. „Simple Mode“ er den guide-wizard-workflow i 4 trin (Import → Processing → Preview → Export, dokumenteret i kapitel 10 under Z1–Z4), „Expert Mode“ det klassiske tre-panel-layout med Navigator, 3D-viewport og Expert-Inspector fra kapitel 2. Værdien huskes på tværs af genstart. Identisk virkning som menuen Mode → Simple Mode (⌘1) / Mode → Expert Mode (⌘2), bortset fra at menuen skifter den løbende session, mens denne vælger sætter standarden for fremtidige sessioner. Begge tilstande tilgår samme projekt-state — projekter, kameraer og træningskonfiguration bevares ved tilstandsskift. Tilstandsspecifikke værktøjslinjeknapper gen-rendres med det samme.

KORT FORTALT

Her vælger du, hvilken brugerflade RadiancKit starter op med næste gang. „Simple Mode“ er begyndertilstanden: fire klare trin, forudvalgte indstillinger, næsten ingen valg. „Expert Mode“ er det fulde værktøjskasselayout med alle håndtag, du ser i kapitel 2. Du kan til enhver tid skifte frem og tilbage via menuen „Mode“, uden at billeder eller træningsfremskridt går tabt.

S2 Language

HVOR

Settings → General → Interface → Language-vælger. Bound: Default: `.system` (følger macOS-sproget).

TEKNISK

Vælger visnings sproget for hele app-UI'en, uafhængigt af macOS-systemsproget. RadiancKit er lokaliseret til 17 sprog (`de`, `en`, `pl`, `en-AU`, `ar-SA` plus 12 yderligere). Ved „System“ følger appen macOS-sproget. Ved et eksplicit valg huskes sprogindstillingen på tværs af genstart; fuld virkning kræver som regel en app-genstart, fordi lokaliserings-bundles kun indlæses ved opstart. De 298 dokumenterede lokaliseringsnøgler i projektet medtages alle, inklusive alle tekster i sub-views og hjælpe-tooltips.

KORT FORTALT

Hvis din Mac kører på engelsk, men du hellere vil have den danske RadiancKit-brugerflade (eller omvendt), stiller du det her. De fleste tekster skiftes med det samme. Nogle dialoger optræder først på det nye sprog efter en app-genstart.

S3 Viewport Background

HVOR

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

TEKNISK

Sætter standardbaggrundsfarven for 3D-viewporten. Tre muligheder: „Dark Gray“ (RGB 0.1, 0.1, 0.1 — default), „Black“ (0, 0, 0) og „White“ (1, 1, 1). Indstillingen persisterer standarden for nye projekter og sessioner på tværs af genstart og opdaterer samtidig den kørende Metal-renderer med det samme. Identiske muligheder findes i menuen Viewport → Background (M21, M22, M23), men Settings-vælgeren sætter standarden, mens menuen skifter den løbende visning. Vigtigt for screenshots og demo-videoer: hvide baggrunde fremhæver grønne/blå floaters tydeligere, mørke baggrunde er bedre til rene render-optagelser.

KORT FORTALT

Farven bag dine 3D-modeller i forhåndsvisningsvinduet. Mørkegrå er standard og passer til de fleste scener. Hvid er godt til screenshots, sort virker mere elegant til render-optagelser. Du kan til enhver tid skifte farven via menuen „Viewport → Background“ for den løbende scene — denne indstilling fastlægger kun, hvilken farve der skal være aktiv ved næste åbning.

S4 Auto-Rotate After Training

HVOR

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

TEKNISK

Starter umiddelbart efter trænings-slut en kontinuerlig turntable-rotation af viewport-kameraet om scenens tyngdepunkt (standard-rotationshastighed ~0.3 rad/s). Praktisk nyttig til demo-sessioner, A/B-sammenligninger og til direkte at vurdere fra 360°-perspektiv, om der er opstået „floaters“ ved scene-kanten. Effekten er visuelt identisk med menuen Viewport → Toggle Auto-Rotation (M16, ⌘⌥T), bortset fra at toggle'en her udløser adfærden automatisk efter trænings-slut i stedet for manuelt. Lader sig senere afbryde til enhver tid via menuen eller ved klik i viewporten (som pauser rotationen). Har ingen indflydelse på trænings-performance — rotationen kører først, når træningen er færdig.

KORT FORTALT

Når aktiveret roterer 3D-scenen automatisk, så snart træningen er færdig — som en karrusel. Rart, hvis du om morgenen efter en natlig træning straks ser resultatet i bevægelse uden selv at skulle klikke. Ved lange sessioner, hvor du blot overvåger træningen, så lad hellere være.

S5 Live Preview Interval **HVOR**

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

 **TEKNISK**

Bestemmer, med hvilket iterations-interval det løbende træningssnapshot rendres ind i 3D-viewporten. Fire diskrete værdier: 0 („Off“), 50, 250, 1000 iterationer. Ved aktiv Live Preview kopierer træneren Gaussian-bufferen fra GPU'en til en separat renderbuffer og udløser en viewport-redraw. Ved „Off“ opdateres viewporten først efter trænings-afslutning. Performance-omkostning: hver 50 iterationer ~5–10% langsommere på M3 Ultra, hver 250 iterationer ~1–2% langsommere, hver 1000 iterationer umålelig. Memory-overhead konstant ~2 GB til snapshot-bufferen, uafhængigt af intervallet. Værdien fungerer som standard for nye træninger; efter træningsstart viser trænings-Inspectoren den reelle live-værdi for den pågældende træning. Ved interval 50 er det visuelle indtryk en flydende „opvækst“ af punktskyen, ved 1000 virker det hakkende.

 **KORT FORTALT**

Mens træningen kører, kan du vælge, hvor ofte 3D-visningen opdateres. „Off“ betyder: ingen opdatering under træningen (hurtigst). „Every 50 Iterations“ viser næsten i realtid, hvordan din scene opstår (lidt langsommere). Til afslappet zen-kigning ved små træninger er „Every 250“ et godt kompromis.

S6 Throttle Delay

HVOR

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

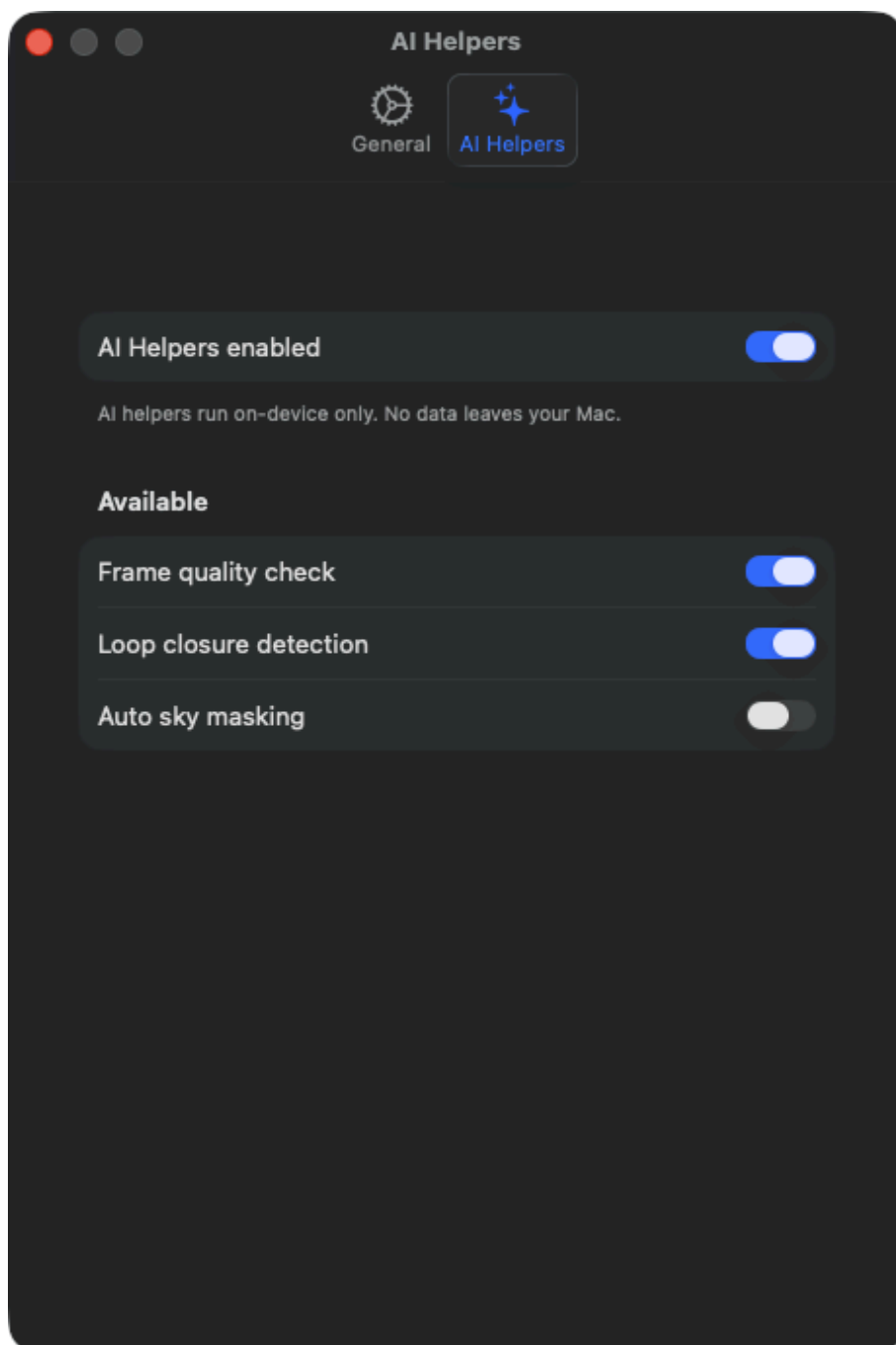
TEKNISK

Indsætter en kunstig forsinkelse i millisekunder mellem trænings-iterationer. Fire diskrete værdier: 0 („Off“), 2 („Light“), 5 („Moderate“), 10 („Eco“). Mening: ved længere træninger (flere timer) bliver GPU'en ellers belastet 100 %, hvilket fører til mærkbart langsommere system-UI (musemarkøren hakker, andre apps bliver trægme). Throttle-forsinkelsen giver GPU'en pauser, hvor andre opgaver kan udføres. Performance-omkostningen er betragtelig: ved 5 ms throttle varer en typisk 40K-træning omkring 50–80% længere end uden throttle. I performance-tilstanden „Eco“ (10 ms) er forsinkelsen pr. iteration længere end selve iterationen — faktor 2–3× langsommere. Ved aktiv throttle vises der under vælgeren en bemærkning: „Throttle is on. Training will be slower than usual.“ Selve appen reagerer ikke mærkbart bedre — kun andre apps har gavn af det.

KORT FORTALT

Hvis din Mac bliver for varm under en lang træning, eller andre programmer bliver for trægme, så slå en bremse til her. „Off“ giver GPU'en fuld gas (hurtigst). „Light“ indlægger en lille pause mellem hvert skridt (lidt langsommere, men systemet reagerer bedre). „Eco“ er den stærkeste bremse — god til nat-træninger på en MacBook, der ikke skal blive for varm.

AI-Helpers-fanen



Figur 16: Indstillinger → AI-helpers-fanen med master-kontakt og sub-toggles

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

Settings → AI Helpers → første sektion → Toggle „AI Helpers enabled“. Bound: Default: `true`.

 **TEKNISK**

Master-kontakt over alle AI-helpers-funktioner i pipelinen. Når slukket, springer import- og SfM-pipelinen alle ML-baserede forbehandlingsstadier helt over — intet Apple-Vision- kald, ingen CoreML-model-load, ingen NPU-opvågning. Når tændt, konsulteres de individuelle sub-toggles (S12–S13). Værdien huskes på tværs af genstart. Påvirker følgende stadier: (a) frame-quality- pre-check før SfM (S12), (b) loop-closure-detektion (S13). Vigtigt: ved slukket er de to sub-toggles deaktiveret og visuelt grånet ud. Footer-bemærkning understreger, at alle AI-helpers strikt kører on-device — ingen billed-upload, ingen sky-behandling. Privatlivsgarantien kommer ved udelukkende brug af Apple-Vision-framework (lokalt på Neural Engine) og CoreML-modeller, der ligger direkte i app-bundlen.

 **KORT FORTALT**

Hovedkontakten for alle funktioner, der internt bruger AI/machine-learning. Standard er „tændt“, fordi hjælperne sparer en masse tid, uden at dine billeder forlader Mac'en. Hvis du vil have dem helt slukket (f.eks. for at spare strøm, eller fordi din Mac ikke har NPU), så sluk dem her — så bliver de to undermuligheder herunder automatisk grå og gør ikke længere noget.

S12 Frame quality check **HVOR**

Settings → AI Helpers → Available-sektion → Toggle „Frame quality check“. Bound: . Default: `true` .

 **TEKNISK**

Aktiverer frame-quality-screeneren (fase 3.11), som før SfM-kaldet analyserer hver importeret frame. Pipeline-trin pr. frame: (a) Laplacian-variance-filer fra Apple Vision (sløringsdetektion — tærskel ~150), (b) histogram-baseret over/under-eksponerings-check (tærskel: >5% pixel ved 0 eller 255), (c) blank-frame-detect (standardafvigelse < 5 over alle pixel). Frames, der består alle tre tjek, går direkte igennem. Frames, der fejler mindst ét tjek, udløser en modal bekræftelsesdialog, der lister hvert problematisk frame med thumbnail og begrundelse og spørger, om det skal fjernes. Vigtigt: ingen automatisk sletning — dialogen er altid påkrævet, brugeren beholder den sidste beslutning. Performance: ~50 ms pr. frame på M3 Ultra, kører parallelt. Ved slukket sendes alle frames uprøvet videre til SfM. Ved deaktiveret master (S11) er denne toggle visuelt grånet ud og uden virkning. Shipped status ifølge memory: SHIPPED 2026-05-23.

 **KORT FORTALT**

Før den egentlige træning kigger appen på hvert foto: er det rystet? helt mørkt eller hvidt? tomt? Hvis ja, spørger den dig, om du vil smide billedet ud — den fjerner aldrig noget automatisk. Det sparer mange timer senere, fordi et eneste totalt rystet billede nogle gange kan ødelægge hele træningen. Standard er „tændt“, fordi indsatsen er næsten nul, og gavnen er stor.

S13 Loop closure detection

HVOR

Settings → AI Helpers → Available-sektion → Toggle „Loop closure detection“. Bound: Default: `true`.

TEKNISK

Aktiverer Apple-Vision-Feature-Print-baseret loop-closure-detektion. For hver importeret frame beregnes en ~768-dimensional feature-vektor, som repræsenterer en neural indlejring af billedindholdet. Derefter sammenlignes alle feature-prints parvis via cosine-similarity. Par med similarity > 0.85 og afstand i frame-index > 50 (altså ikke-nabliggende frames) identificeres som „loop-closure-kandidater“ og skrives i en sidecar-JSONL-fil i projektmappen. Kun informativ — den importerede billed-sekvens modificeres ikke. Mening: giver SfM-solveren (især COLMAP) et hint om, at disse frames hører sammen i klyngen i 3D-rummet. For native SfM er sidecar-informationen p.t. kun dokumenterende; COLMAP bruger hintsene internt via custom matches-file (manuel integration mulig, ikke automatisk koblet). Performance: ~200 ms pr. frame på M3 Ultra, kører parallelt. Ved slukket genereres ingen feature-prints. Ved deaktiveret master (S11) visuelt grånet ud.

KORT FORTALT

Når du går rundt om et objekt og til slut lander ved startpunktet igen, hjælper det computeren enormt at vide det. Denne mulighed registrerer automatisk, hvilke af dine fotos der er optaget „næsten fra samme standpunkt“, og skriver det i en lille hjælpefil. SfM-værktøjer (især COLMAP) kan bruge denne information til at levere en renere 3D-rekonstruktion. Standard er „tændt“, fordi det kører uden din indblanding og ikke ændrer noget ved dine billeder.

Inspector-spejls-settings

De øvrige settings-poster (S17–S33) fra inventartabellen er spejlinger fra Expert-Inspectoren og dokumenteret i kapitel 2 (Inspector-Controls I12–I29). De optræder ikke fysisk i indstillingsvinduet, men er kun listet i inventaret, fordi de kører via `TrainingConfig`-properties, der persisteres via, og dermed formelt har settings-karakter. For indholdsmæssige forklaringer henvises der dertil.

Hvornår hvad?

Setting	Gyldighedsområde	Persistens
S1 Default Mode	App-globalt	App-genstart
S2 Language	App-globalt	App-genstart
S3 Viewport Background	App-globalt (default) + runtime	App-genstart
S4 Auto-Rotate After Training	App-globalt	App-genstart
S5 Live Preview Interval	Default for nye træninger	App-genstart
S6 Throttle Delay	Default for nye træninger	App-genstart
S11 AI Helpers Master	App-globalt	App-genstart
S12 Frame quality check	App-globalt	App-genstart
S13 Loop closure detection	App-globalt	App-genstart

App-globalt = virker på alle projekter. Default for nye træninger = virker kun på den næste oprettede træning, løbende sessioner forbliver uændrede. Aktuell træning = virker med det samme på den løbende trænings-konfiguration, men persisterer ikke uden eksplicit reimport.

KAPITEL

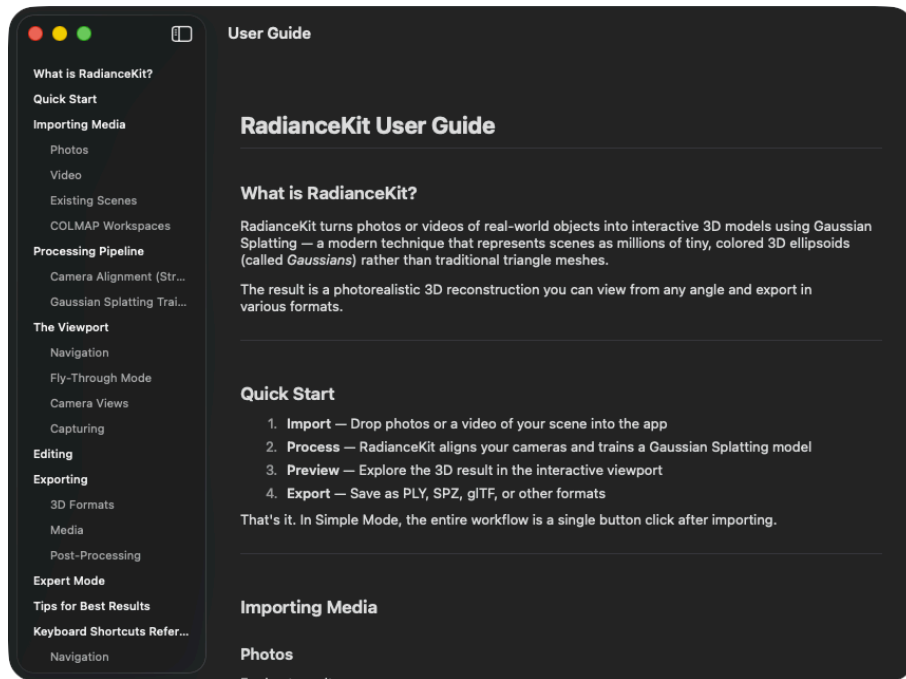
Kapitel 4 — Hjælpevinduer

Foruden hovedvinduet (3D-viewport plus Inspector) forvalter RadianceKit syv yderligere vinduer, der alle åbnes via Help-menuen. Listen er fra top til bund: User Guide (⌘?), Keyboard Shortcuts (⌘/), Open Training Logs... (åbner ikke et app-vindue, men Finder; derfor ikke yderligere behandlet her), Manage Storage..., Pareto Dashboard... (⇧⌘D), Holdout Analysis... (⇧⌘H), BayesOpt Console... (⇧⌘B). Tre af dem — Dashboard, Holdout, BayesOpt — er selvstændige analyse-værktøjer. De har hver deres egen view-model-stack, læser eller skriver JSON-filer på disk, og for hver findes der et CLI-argument, hvor du kan få vinduet til at pege direkte på en bestemt fil ved app-start (`--dashboard-dir` , `--holdout-file` , `--bayesopt-autorun`).

De fire enkle vinduer (User Guide, Keyboard Shortcuts, Manage Storage, plus undermenu-punkterne Open Training Logs / Open Exports Folder) får pr. styreelement en kort post. De tre analyse-vinduer er mere udførligt dokumenteret — hver med en indledning, der forklarer, hvad du ser i vinduet, hvornår du bør åbne det, og hvordan du fortolker det viste billede.

Sidst i kapitlet er der et krydshenvisnings-afsnit til hovedvinduets Inspector: hvad du fornuftigt kan aflæse i live- loss-charten og gaussian-count-visningen under en kørende træning.

User Guide (W1–W4)



Figur 17: User Guide-vindue med sidebar til venstre og rendret markdown- indhold til højre

Hvad det er: Et indbygget hjælpevindue, der renderer den medfølgende `guide_<prog>.md`. Sproget afledes af indstillingerne (fanen General → Language) eller, hvis „System“ står der, af macOS-sprogpræferencerne. Layout er klassisk: til venstre sidebar med alle overskrifter, til højre brødteksten.

HVORNÅR ÅBNE Når du har brug for en hurtig påmindelse om et enkelt punkt — altså som stikordsersrtning. Den udførlige reference er denne manual; det indbyggede hjælpevindue er snarere det, en `--help` på kommandolinjen ville være. Den opdateres ved hver app-udgivelse, men holdes indholdsmæssigt overfladisk.

W1 NavigationSplitView (sidebar + detalje)



Help → User Guide (⌘?).



Tospalte-layout med smal sidebar (mindst 180 pt bred) til indholdstræet og et scrollbart detaljeområde til selve markdown-indholdet. Vinduet har en minimumsstørrelse på 700 × 500 pt. Ved første åbning indlæser vinduet den passende `guide_<lang>.md` fra app-bundlen (fallback `guide_en.md`), parser den til blok-records (overskrifter H1–H4, afsnit, lister, tabeller, separatorlinjer) og udtrækker separat overskriftsstrukturen til sidebar. Inline-formatering (fed, kursiv, code-span) rendres via den indbyggede markdown-engine. Sproget læses fra app-indstillingerne, med specialtilfældet kinesisk (`zh-Hans`) og brasiliansk portugisisk (`pt-BR`), som bevares som fulde locale-tags, fordi disse varianter adskiller sig fra zh og pt.

KORT FORTALT

Den indbyggede hjælpetekst, til venstre emnelisten, til højre indholdet. Sproget indstilles automatisk efter dine system-settings. Fungerer offline, men er bevidst kun en kortversion — den fulde reference er denne manual.

W2 List (heading-sidebar)



Venstre kolonne i User Guide-vinduet.



Liste over alle H2- og H3-overskrifter i det aktuelle markdown-dokument. H2-poster optræder uden indrykning med medium skriftsnit, H3-poster med 16 pt indrykning til venstre og reduceret foreground-stil. H4 og dybere ignoreres, fordi dybden ellers gør sidebar uoverskuelig. Anker-ID'er genereres fra heading-teksten via slugifisering (lowercase + mellemrum til streger + filtrering på bogstaver/tal/streger — samme algoritme, som GitHub bruger til sine markdown-ankre, så også eksterne URL'er til dok potentielt ville lande på samme anker). Listen bruger native macOS-stil.

KORT FORTALT

Navigationsbjælken i venstre side. Tryk på en post, og du hopper til afsnittet.

W3 Button (heading → anker-spring)



Pr. sidebar-linje en knap.



Hver sidebar-post er en knap, der sætter det aktuelle anker, men optisk ser ud som en liste-post. En observator-variabel udløser så scroll-springet til det tilsvarende anker med en blød animation over 0,3 s. Efter springet nulstilles anker-værdien, så næste klik på samme anker udløser igen (ellers ville observatoren ikke udløse på ny, fordi værdien ikke har ændret sig).

KORT FORTALT

Klik fører dig til det rigtige sted i teksten til højre.

W4 ScrollView (detalje-indhold)



Højre kolonne.

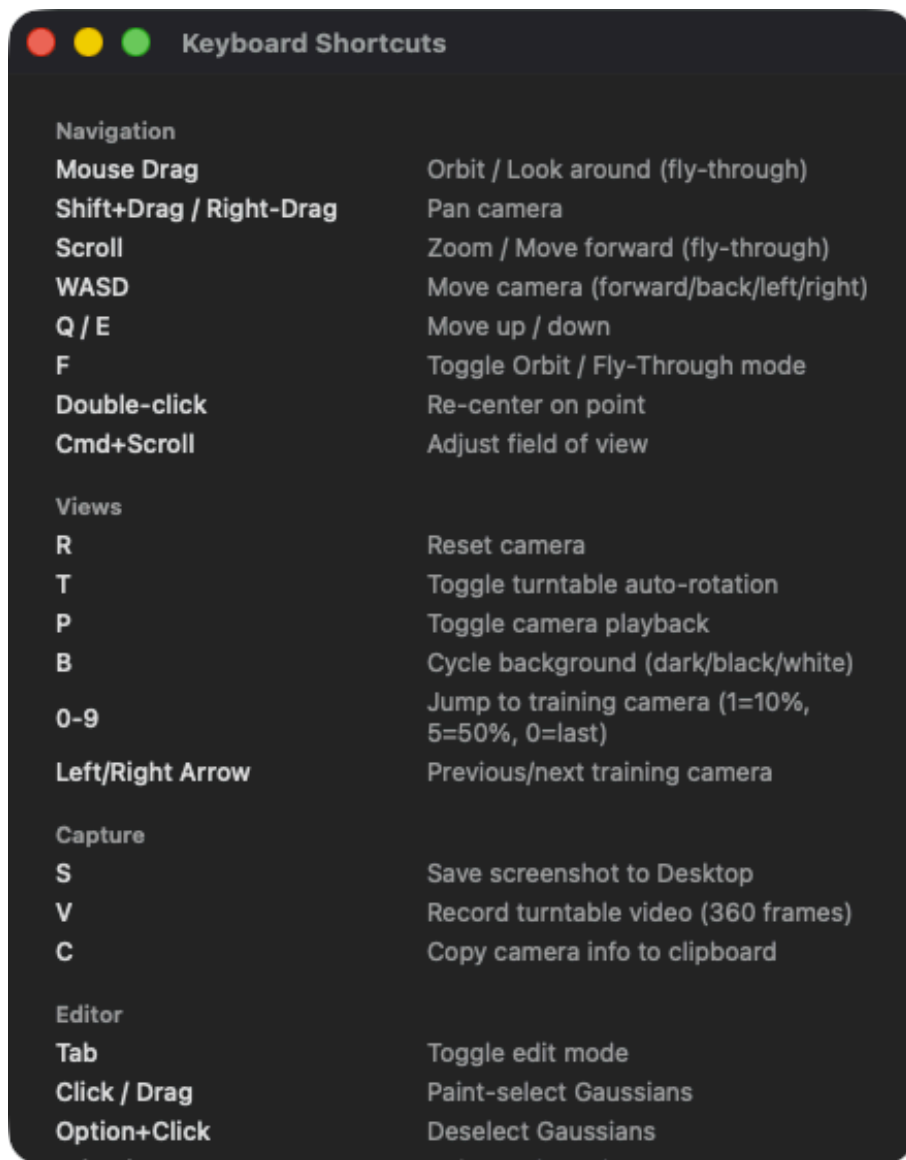


Scrollbart, vertikalt stablende indholdsområde med lazy rendering, fordi længere guides nemt kan have over 200 markdown-blokke — en ikke-lazy variant ville instantiere dem alle samtidigt. Hver blok får et eget ID, enten heading-ankeret (for hopbare H1–H3) eller en index-pladsholder. Maksimumbredde er 720 pt, padding 32 horisontalt / 24 vertikalt, så lange linjer bevarer et godt læsbart layout. Tabeller rendres celle-vis med horisontale stacks og separatorlinjer; inline-kode af den indbyggede markdown-engine. Egentlige kodeblokke behandles aktuelt som paragraph — en kendt begrænsning ved hjælpevinduet.

KORT FORTALT

Selve hjælpeteksten. Scrollbar, godt læselig bredde, klar typografi.

Keyboard Shortcuts (W5–W6)



Figur 18: Keyboard Shortcuts-vindue — fem grupper Navigation/Views/Capture/ Editor/Training med hotkey-kolonne til venstre og beskrivelse til højre

HVAD BILLEDET VISER Statisk reference-liste i fem sektioner. **Navigation:** Mouse Drag (orbit/fly), Shift+Drag/Right-Drag (pan), scroll (zoom), WASD (fly-through- bevægelse), Q/E (op/ned), F (toggle orbit/fly), dobbeltklik (re-center), Cmd+scroll (FoV-justering). **Views:** R (reset camera), T (auto-rotation), P (camera playback), B (background-cycle), 0–9 (spring til training-cam 1=10 %/5=50 %/0=last), venstre/højre pil (prev/next cam). **Capture:** S (screenshot til desktop), V (turntable-video), C (copy camera info). **Editor:** Tab (edit-mode), klik/træk (paint-select), Option+klik (deselect), X / Delete (slet selektion), Cmd-Z (fortryd sidste sletning), [/] (pensel mindre/større), Esc (ophæv selektion). **Training:** Start, Pause/Resume, Cancel, Continue +5K/+10K/+20K via menu-genvejene i M9–M14.

Hvad det er: En enkel statisk oversigt over alle tastaturgenveje — Navigation, Views, Capture, Editor, Training. Indhold er hårdkodet i, ingen markdown-loading.

HVORNÅR ÅBNE Når du leder efter den hurtigste vej til at gøre noget i viewporten. WASD-fly-through, R til camera-reset, B til background-cycling — alle står her.

W5 ScrollView (indholdsområde)

 HVOR

Help → Keyboard Shortcuts (⌘/).

 TEKNISK

Et enkelt scroll-område med en vertikal liste i. Padding 20 hele vejen rundt, ingen sidebar-navigations-tree (listen er kort nok). Indhold er grupperet i fem sektioner (Navigation, Views, Capture, Editor, Training). Pr. tastekombination en linje med oversættelig tekst i begge kolonner. Venstre kolonne (tastekode) fastgjort til 180 pt bredde, så beskrivelserne til højre forbliver vertikalt aligned. Ingen interaktion ud over scrolling — klik på en linje udløser intet, tastegenvejene er ægte tastatur-modifiers i menuen og på viewporten.

 KORT FORTALT

Tabel over alle shortcut-taster. Statisk cheat sheet til hurtigt opslag.

W6 VStack (shortcut-sektioner)

 HVOR

Inden for ScrollView.

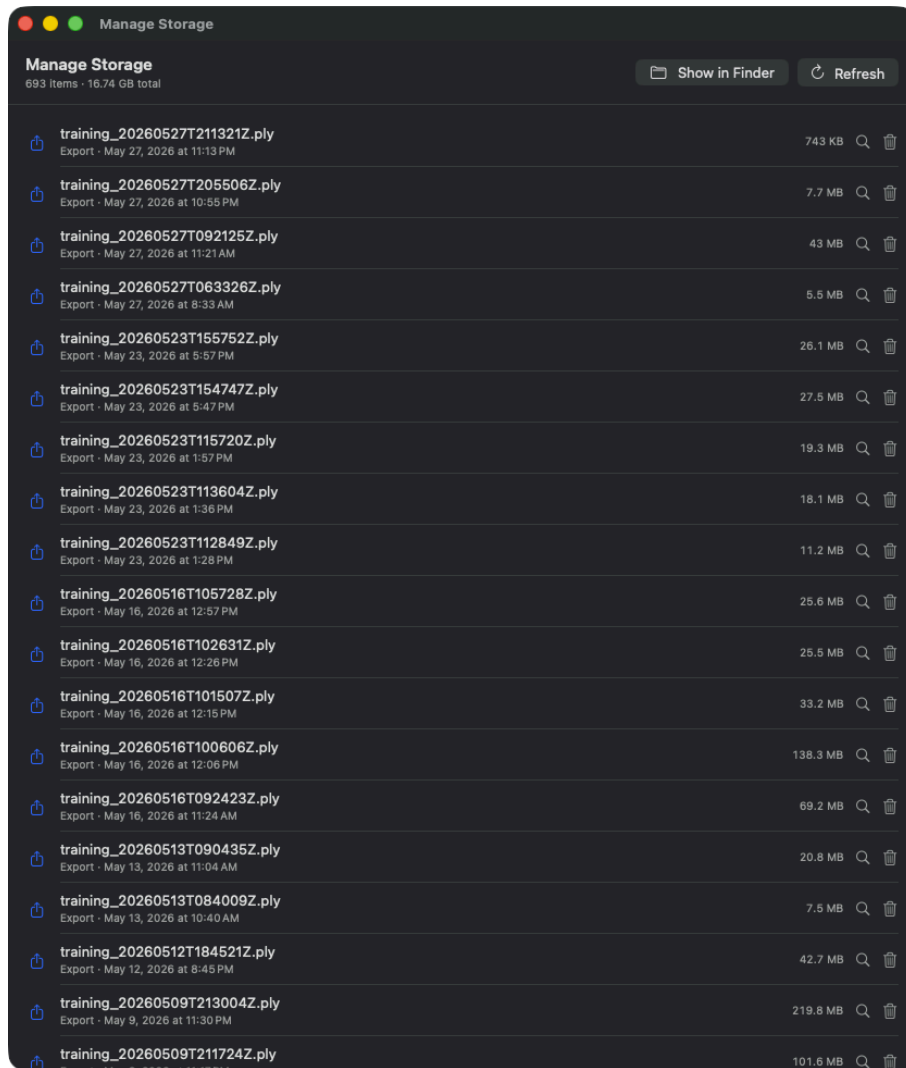
 TEKNISK

Venstrejusteret stablede sektioner med 16 pt afstand. Inden for de fem sektioner hver heading + linje-følge. Headings bruger en sekundær subheadline-stil — bevidst ikke title-format, fordi sektionerne ikke behøver at være navigerbare. Indhold er bevidst fladt (ingen disclosure, ingen search, ingen filter), så komponenten kører uændret på enhver macOS-version, og filen forbliver læselig.

 KORT FORTALT

Grupperingen af taster efter funktion (Navigation, Views, Editor og så videre).

Manage Storage (W7–W12)



Figur 19: Manage Storage-vindue — header viser „693 items · 16.74 GB total“, tabel med export-PLY-filer sorteret efter dato, hver med format-pille + filnavn + størrelse + dato

HVAD BILLEDET VISER Tabelvisning af alle filer, som RadiancKit forvalter. Header tæller 693 items, 16.74 GB samlet størrelse. Toolbar øverst: „Show in Finder“ + „Refresh“. Hver linje: PLY-ikon, filnavn (f.eks. training_20260527T211321Z.ply), export-dato, størrelse (varierer 7 KB til 218 MB), forstørrelses-ikon (reveal) og papirkurv-ikon (move to trash). Filer er sorteret efter dato, nyeste øverst. I denne demo-optagelse dominerer PLY-eksporter, fordi der er arbejdet meget med `--benchmark`.

Hvad det er: En disk-usage-oversigt for alt, hvad RadiancKit lægger under ~/Documents/RadiancKit/ — logs, exports, scenes, capture-bundles (fra iOS-companion), imports (staging-kopier af input-billederne). Pr. post en størrelse i bytes og to knapper: „vis i Finder“ og „flyt til papirkurv“. Det er INGEN automatisk oprydning — appen sletter intet selv; du beslutter pr. post.

HVORNÅR ÅBNE Når disken bliver fuld. Især logs samles (en JSONL pr. trænings-forsøg, plus `_qualityMetrics.json`); eksporterne også (PLY 100 % rå data, en pr. eksport). Også

nyttig efter et crash, hvor imports-staging-mappen har gamle kopier af input-billederne liggende (se „disk-pressure incident“ i `dev_v549f-needle-reduction.md`).

W7 Knappen „Show in Finder“

 HVOR

Header øverst til højre i storage-browser-vinduet.

 TEKNISK

Åbner hele RadianceKit-mappen (`~/Documents/RadianceKit/`) i Finder, så du kan se mappestrukturen direkte og også manipulere den med Finder selv. Handlingen åbner et nyt Finder-vindue og skifter ikke til `app-sandbox-containeren` — `~/Documents/RadianceKit/` er det regulært app-tilgængelige `Documents`-domæne, ingen `sandboxed-container-sti`.

 KORT FORTALT

Åbner mappen i Finder, så du selv kan håndtere filerne.

W8 Knappen „Refresh“

 HVOR

Header, ved siden af Finder-knappen.

 TEKNISK

Udløser en baggrunds-scanning, der kører på en bruger-initieret asynkron task, så scanningen af store mappetræer ikke blokerer UI'en. Selve gennemgangen går hver kendt undermappe (Logs, Exports, Scenes, Captures, Imports) igennem og genererer en storage-post pr. direkte barn. Pr. post bestemmes den rekursive størrelse — helst det faktiske diskforbrug (inklusive APFS-hardlinks-sharing) med fallback på den logiske filstørrelse.

 KORT FORTALT

Læser listen på ny, hvis du i mellemtiden har slettet eller tilføjet noget i Finder.

W9 List (storage-poster)

Hovedindhold under headeren.



Liste med dette layout pr. linje: kategori- specifikt SF-symbol-ikon (dokument for logs, upload-pil for exports, terning for scenes, tray for imports), navn + undertitel (kind-label + formateret modifikationsdato), bytes-counter til højre (højrejusteret, monospaced), reveal-knap (forstørrelses-symbol), trash-knap (papirkurv). Sortering: primært efter kind (scenes først, så exports, logs, captures, imports, other), sekundært efter modifikationsdato faldende (nyeste øverst). Hvis scanningen stadig kører, viser stedet i stedet et „Scanning..“ - fremskridt. Hvis intet blev fundet, en empty-state-visning med tray-ikon.

 KORT FORTALT

Liste over alle dine RadianceKit-data, sorteret efter type og aktualitet. Pr. post ser du størrelsen og kan slette direkte.

W10 Row-knap „Reveal in Finder“

Pr. linje, forstørrelses-symbol til højre.



Åbner Finder og vælger den specifikke post (fil eller mappe). Forskel fra W7: W7 åbner rod-mappen; W10 markerer præcis denne ene post. Praktisk workflow: identificér en stor post, klik på forstørrelsen, kopiér den så f.eks. til et eksternt volume.

 KORT FORTALT

Springer i Finder direkte til denne post, så du hurtigt finder den.

W11 Row-knap „Move to Trash“

Pr. linje, papirkurv-symbol til højre for forstørrelsen.



Udløser bekræftelses-dialog-boksen (W12). Først efter bekræftelse kører macOS-standard-operationen „flyt til papirkurv“ (altså reversibel, ingen direkte sletning). Efter vellykket trash fjernes posten fra listen, og total-byte-counteren opdateres. Ved fejl indkobles en modal fejl-dialog.

 KORT FORTALT

Flytter posten til papirkurven. Dialog spørger først.

W12 ConfirmationDialog (slette-bekræftelse)**HVOR**

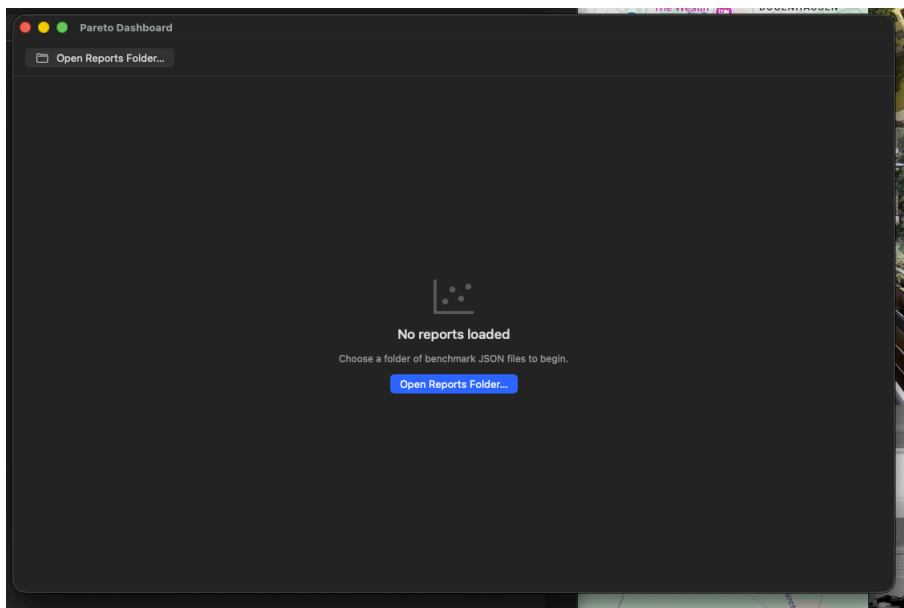
Udløses af W11, vises som macOS-sheet.

TEKNISK

Standard bekræftelsesdialog med dynamisk titel „Delete <navn>?” og en besked-linje, der eksplicit påmindes om, at posten lander i papirkurven og kan gendannes derfra (indtil papirkurven tømmes). To knapper: „Move to Trash” som destruktiv handling (vist rødt) og „Cancel” med automatisk Esc-binding. Dialogen er non-modal i den forstand, at den kun blokerer dette vindue, ikke hele appen — det er macOS-standard for reversible sletninger.

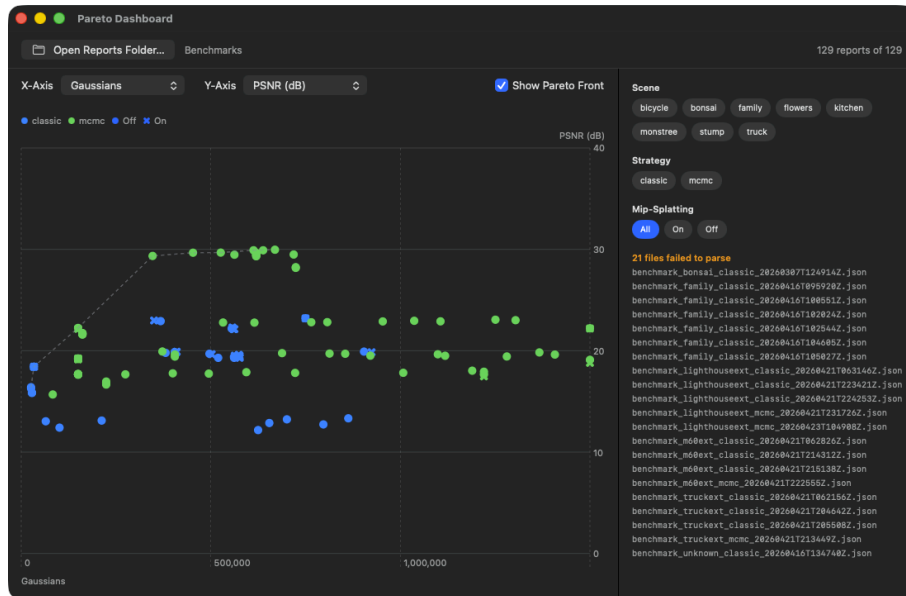
KORT FORTALT

Sikkerhedsspørgsmål før sletning. „Move to Trash” er reversibelt — så længe papirkurven ikke er tømt.

Pareto Dashboard (W13–W22)

Figur 20: Pareto Dashboard — tom tilstand før report-import

Tom tilstand (efter første åbning) — empty-state med call-to-action „Open Reports Folder...”. Datapunkterne optræder, så snart trænings-reports er indlæst, se næste shot.



Figur 21: Pareto Dashboard med 129 indlæste benchmark-reports — Gaussians vs PSNR med Pareto-front, Scene/Strategy/Mip-filter

HVAD BILLEDET VISER Header-toolbar viser „129 reports of 129“ (alle reports i den valgte mappe blev parsed med succes — 21 yderligere filer kunne ikke parses pga. ældre format, se hint-listen til højre). Akser: X-Axis-vælger på Gaussians, Y-Axis-vælger på PSNR (dB). Scatter-plot: grønne punkter = Classic-strategi, blå punkter = MCMC. Den stiplede Pareto-front- linje løber langs de bedst opnåede PSNR-værdier og plateauer omkring PSNR≈30 dB fra ca. 500K gaussians. Filter-chips til højre: 7 scener (bicycle, bonsai, family, flowers, kitchen, stump, truck), 2 strategier (classic, mcmc), 3 Mip-splating-muligheder (All, On, Off). I øjeblikket er alle filtre åbne, derfor den tætte punkt-klynge.


Hvad det er: Et multi-run-sammenligningsværktøj. Du har i fortiden trænet flere scener eller samme scene med forskellige presets — hver af disse træningskørsler producerer (hvis du har givet `--benchmark` med eller kaldt via `benchmark`-funktionen) en JSON-report-fil, der bl.a. indeholder final-PSNR, SSIM, LPIPS, gaussian-count og wallclock-tid. Dashboardet indlæser en hel mappe af sådanne reports samtidigt og plotter dem som 2D-scatter med valgbare akser. Derudover tegnes Pareto-fronten (mængden af ikke-dominerede punkter) som stiplede linje.

HVORNÅR ÅBNE Efter du har oprettet mindst tre eller fire trænings-reports. Med færre punkter er frontier-linjen ikke sigende. Typisk use-case: du har prøvet at rekonstruere en outdoor-scene og har spillet P3 Balanced (Classic), P4 Quality (Classic), P7 MCMC Quality og P9 Outdoor (tuned) igennem efter hinanden — nu vil du vide, hvilken konfiguration der leverer bedste PSNR pr. sekund træningstid, eller hvilken der kræver færrest gaussians til given PSNR.

SÅDAN FORTOLKES DET Begge akser kan frit vælges (X-akse:,, psnr , ssim , lpips , ...; Y-akse ligeså). Pareto-front-logikken i `ParetoFront2D.indices` ved for hver metrik, om „mindre = bedre“ (f.eks. LPIPS, Loss, Time) eller „større = bedre“ (PSNR, SSIM) — linjen løber altså alt efter akse-valget fra nederst til venstre til øverst til højre eller fra øverst til venstre til nederst til højre, altid langs den bedste opnåede kombination. Et

punkt er Pareto-optimalt, hvis INTET andet punkt i BEGGE dimensioner er mindst lige så godt (altså intet andet dominerer det). Pareto-optimale punkter ligger på linjen, andre punkter til højre/over (alt efter akseorientering). Punkter PÅ linjen er de ægte kandidater til „bedste preset“; punkter LANGT fra linjen er spildt træningstid.

FILTER-CHIPS Du kan begrænse valget til en bestemt scene (hvis du f.eks. kun vil sammenligne outdoor-runs), til en bestemt strategi (Classic eller MCMC), eller Mip-splattung til/fra (relevant efter fase Q1.5, hvor Mip forbliver som opt-in advanced flag).

 Du har tre reports for „truck“-scenen under ~/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). Sæt X-aksen til trainingTime, Y-aksen til PSNR. Run B ligger øverst til højre, Run C endnu længere mod øverst til højre, Run A nederst til venstre. Pareto-fronten forbinder A og C — begge ikke-dominerede. Run B er „lost“ (C er bedre i både time OG PSNR). Indsigt: for „truck“ er MCMC-default'en ikke værd det; enten hurtig+ok (A) eller lang+meget god (C). Gem konfigurationen fra C som egen preset (Inspector → I1 Save Preset).

Næste handling: Gem bedste konfiguration som preset. Konkret: kig på Pareto-punkterne (hover viser PSNR/SSIM/LPIPS/Gs/ Time i tooltipet), beslut hvilken time-vs-quality-tradeoff der passer dig bedst, åbn den tilhørende report (filnavnet indeholder run-timestampet), kopiér dens trænings-konfiguration i en ny run eller gem den efter næste trænings-session som preset via Inspektoren.

W13 Knappen „Open Reports Folder...“

 HVOR

Toolbar øverst til venstre.

 TEKNISK

Åbner en mappe-vælger med opfordringen „Select a folder containing benchmark .json reports“. Efter bekræftelse kører en baggrunds-task, der parserer alle `.json`-filer i mappen sekventielt. Fejlagtige reports (defekt JSON, forkert skema) samles og vises nederst i sidebaren som „N file failed to parse“ — intet crash. Hvis der kommer et andet klik, mens en første load stadig kører, afbrydes den tidligere task, så der ikke skrives to resultater ind i state samtidigt.

Også via CLI: `--dashboard-dir /sti/til/reports` indlæser mappen direkte ved app-start.

KORT FORTALT

Vælger mappen, hvor dine benchmark-reports ligger. Standardstien er ~/Documents/RadianceKit/Reports/. Indlæser så alle JSON'er på en gang.

W14 Vælger „X-Axis“

HVOR

Over charten, til venstre.



TEKNISK

Menu-vælger med alle tilgængelige metrik-akser i dashboard-modulet (PSNR, SSIM, LPIPS, gaussian-count, træningstid og så videre). Default er gaussian-count. Ved skift nulstilles det hoverede punkt, fordi en hidtil highlightet position i det gamle aksekoordinatsystem ikke længere giver mening efter akse-skift. Vælgeren er begrænset til indholdsbredden, så den ikke spænder over hele bredden.

KORT FORTALT

Hvilken metrik der skal stå på den vandrette akse. Som regel „træningstid“ eller „gaussian-antal“, fordi det er de „omkostninger“, du vil sammenligne.

W15 Vælger „Y-Axis“

HVOR

Over charten, ved siden af X-Axis.



TEKNISK

Identisk med W14, bortset fra at default er PSNR. Akse-valget gemmes uafhængigt, så brugeren også kan vælge sludder-kombinationer (X=PSNR, Y=PSNR — ville kaste alle punkter ned på en diagonal). Sådanne kombinationer fanges dog ikke; bevidst beslutning, fordi en sammenligning „SSIM vs PSNR“ sagtens kan være interessant for at se, hvor konsistent metrikkerne opfører sig.

KORT FORTALT

Hvad der står på den lodrette akse. Normalt „PSNR“ eller „SSIM“ som kvalitetsmål.

W16 Toggle „Show Pareto Front“

HVOR

Til højre for akse-vælgerne.



TEKNISK

Standard macOS-toggle. Når aktiv, tegnes ud over punktskyen i Pareto-charten en linje med den beregnede 2D-Pareto-front. Stil: stiple (stregmønster 4–4), grå halvtransparent, linjestyrke 1,5 pt. Pareto-beregningen kører på hovedtråden — ved det typiske antal reports ($\leq \sim 50$) er det problemfrit hurtigt. Når toggle er fra, udelades linjen, så kun de nøgne punkter står.

KORT FORTALT

Viser linjen, der går gennem de „hidtil bedste“ punkter. Hvis linjen er i vejen (f.eks. fordi du kun vil sammenligne de enkelte trades), så sluk den.

W17 Chips „Scene“-filter

Højre sidebar i dashboard-vinduet.



Filter-chips for hver scene, der optræder i de indlæste reports. Eget flow-layout, der automatisk ompakker chips i flere linjer, så snart bredden er udnyttet. Aktive chips får accent-baggrund, inaktive en neutral standard-material-baggrund. Multivalg er muligt (set-semantic); hvis ingen chip er valgt, gælder alle scener som „lukket ind“ — dvs. set-logikken er „tomt valg = alt“, ikke „tomt valg = intet“.

 KORT FORTALT

Klik på et scene-navn filtrerer punkterne til kun denne scene. Multivalg muligt. Tom = alle scener.

W18 Chips „Strategy“-filter

Under scene-filteret i sidebaren.



Præcis som W17, men for trænings-strategier — typisk de to værdier „classic“ og „mcmc“, afledt fra strategy- feltet i benchmark-report-JSON'erne. Hjælpesom, hvis du har blandet reports fra begge strategier og kun vil se den ene type (f.eks. „vis kun MCMC-runs, fordi jeg har udelukket Classic“).

 KORT FORTALT

Filter efter Klassisk eller MCMC. Som standard er begge aktive.

W19 Chips „Mip-Splatting“-filter

Under strategy-filteret i sidebaren.



Treværdigt filter (i stedet for set som W17/W18): „All“ / „On“ / „Off“. Baggrund: Mip-splatting blev evalueret i fase Q1.5 som eksperimentel multi-skalering, og den endelige verdict var „ingen pæn win gennemgående; beholdt som opt-in flag“. Når du laver Mip-on/off-sammenligninger, vil du ofte kunne adskille meget skarpt. Derfor det dedikerede ternære filter med tilstandene „lad alt igennem“, „kun Mip til“, „kun Mip fra“. Sidebar-sektionen indkobles kun, hvis der er mindst én Mip-report OG mindst én ikke-Mip-report i data-mængden (ellers giver filtreringen ingen mening).

 KORT FORTALT

Hvis du vil sammenligne Mip-splatting til/fra, treværdigt filter her. Ellers ignorer.

W20 ChipButton (filter-toggle, all/on/off)

Helper-component, bruges i W17/W18/W19.



Minimalistisk knap-wrapper. Indhold: label-tekst med caption-skriftgrad og padding 10 horisontalt / 5 vertikalt. Baggrund betinget: når aktiv → app-accentfarve med hvid tekst; ellers neutral standard-material-baggrund med sort tekst. Formen er en capsule (pilleformet). Plain-buttonstyle, så capsule-materialet ikke overdækkes af en system-border.

 KORT FORTALT

De runde filter-knapper selv. Optisk som et iOS-tag.

W21 Chart (Pareto-scatter)

Midten af dashboardet.



Swift-Charts-diagram med to lag: 1. et punkt pr. report — position fra de valgte X- og Y-metrikker, farve efter strategy, symbol efter Mip-status. Symbol-størrelse normalt 80, highlightet 200 (når ID svarer til den aktuelt hoverede report).
2. en linje for Pareto-fronten, kun hvis toggle er tændt.

Chart-overlay: et transparent rektangel registrerer musebevægelse; pr. frame bestemmes den euklidiske nærmeste punktposition i plot-frame, og den hoverede report opdateres, hvis distancen er under 24 px (ellers nulstillet). Sådan får du tooltippet uden klik — hover rækker.

KORT FORTALT

Det egentlige scatter-diagram. Hvert punkt er en trænings-kørsel. Hover for detalje-tooltip.

W22 Tooltip (hover-detalle)

Under charten, indkoblet ved hover.

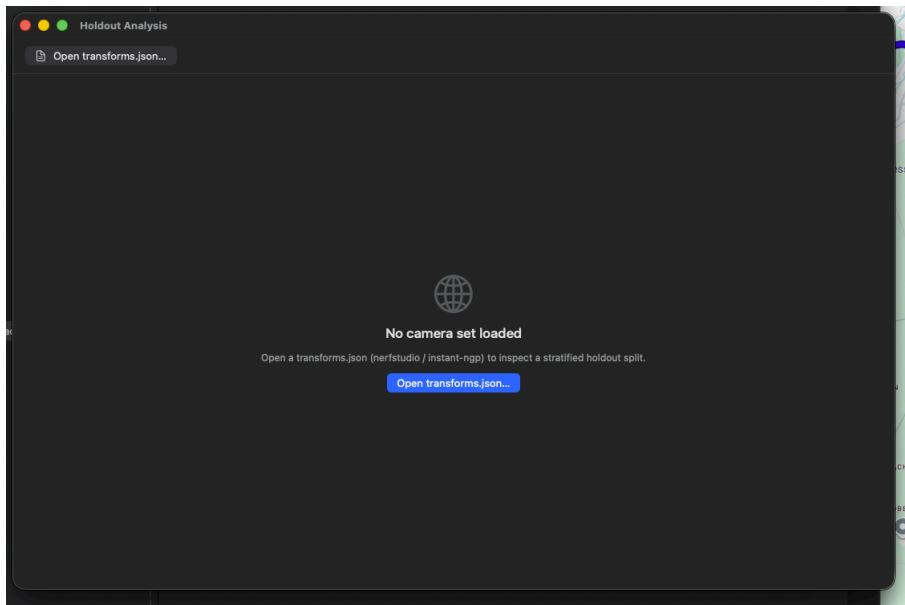


Horizontalt stack: scene-navn (headline), strategy-tag (caption), separator-linje, så PSNR/SSIM/LPIPS/Gs/Time-metrikker i hver en lille vertikal gruppe (label + monospaced værdi). Hvis Mip var aktiveret, desuden et „Mip“-capsule-tag i accentfarve. Baggrund halvtransparent blur, afrundet rektangel med 8 pt radius. Viser kun, når musen faktisk er over et punkt. Forsvinder automatisk ved forlading.

KORT FORTALT

Detalje-kortet nederst, når du fører musen over et punkt. Viser alle kvalitetsmetrikker og run-konfigurationen på en gang.

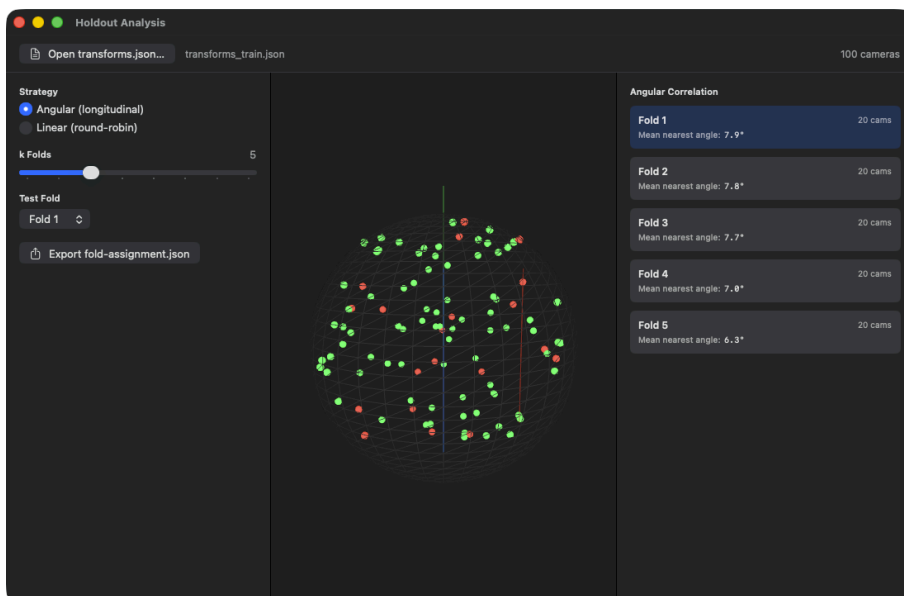
Holdout Analysis (W23–W29)



Figur 22: Holdout Analysis — tom tilstand før indlæsning af en transforms.json

Tom tilstand med empty-state og call-to-action „Open transforms.json...“. Accepterer NeRF-Studio- og Instant-NGP-format.

Tom tilstand (efter første åbning) — kamera-markører optræder, så snart en transforms.json er indlæst, se næste shot.



Figur 23: Holdout-globe med 100 NeRF-Blender-mic-kameraer, 5 folds à 20 kameraer, Angular-strategi aktiv

HVAD BILLEDET VISER Header viser indlæst fil (transforms_train.json) og cam-count („100 cameras“). Venstre sidebar: strategy-vælger med to muligheder — Angular (longitudinal) aktiv (justerer folds efter længde-/breddesektorer på sfæren, så hver test-fold er geometrisk tæt) vs Linear

(`round-robin`) (rækkefølge-baseret, alle k 'te frames som test-sæt). k -Folds-slider står på 5, test-fold-vælger på fold 1. Export-knap genererer en `fold-assignment.json` til Nerfstudio/Instant-NGP. Midter-panel: 3D-globe-projektion af alle 100 kameraer — grønne punkter = train, røde punkter = aktuel test-fold (fold 1 med 20 kameraer). Højre sidebar (Angular Correlation): pr. 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°) — mindre værdi betyder, at kameraerne inden for denne fold ligger tæt sammen, altså at holdout-splittet er rumligt kohærent.



Hvad det er: En 3D-visualizer for din kamera-arrangement med cross-validation-logik. Du indlæser en `transforms.json` (standardformatet fra Nerfstudio / Instant-NGP for kameraposer), appen læser alle kameraer, projicerer deres kigge-retninger på en enhedskugle og viser dem som små kugle-markører på en virtuel globus. Derefter opdeler den kameraerne i k folds (efter valgt strategi: angular eller linear), markerer grønt for træningsdelen og rødt for testdelen (holdout) og beregner pr. fold en angular-correlation-score, der fortæller dig, hvor langt test-folden ligger fra trænings-folden i kigge-vinkel-rummet.

HVORNÅR ÅBNE Når du vil lave holdout-evaluering — altså: hvor godt generaliserer din model til usete kigge-vinkler? Standard i træningen er „every-8th view som holdout“ (Mip-NeRF360-konvention), men det er en meget lineær opdeling. Hvis dine billeder f.eks. er tidsmæssigt klyngede (først én side af objektet, så den anden), er „every-8th“ ikke repræsentativ — en tilfældig sekvens-position lander i testen, men alle dens naboer er i træningen, det er for nemt. Med „angular“ stratificerer man i stedet over kigge-vinkel-rummet: hver fold indeholder kameraer fra alle dele af orbitten, så testen virkelig prøver generaliserings-huller.

SÅDAN FORTOLKES DET Angular vs Linear:

- Angular (standard): opdeler kameraerne efter longitudinalvinkel (ϕ -koordinat omkring Y-aksen) i k lige sektorer. Fold 0 er kameraer med $\phi \in [0^\circ, 360/k^\circ)$, fold 1 de næste, og så videre. Fordel: hver fold dækker en deleget af orbitten; test-folden er rumligt kompakt, men bredt fordelt over verdens-datasættet. God til klassiske orbit-optagelser.
- Linear (round-robin): `fold-index = (image_index modulo k)`. Det er den simple „every- k -th“-opdeling. Fungerer, hvis billedrækkefølgen ingen spatial bias har (f.eks. tilfældigt sorterede droneoptagelser). Fungerer dårligt, når billederne klynger tidsmæssigt.

I 3D-globussen ser du straks: grønne punkter (træning) og røde punkter (test). Hvis de røde punkter alle klynger sig i et hjørne, er holdout dårlig (ingen god generaliserings-test). Hvis de ligger jævnt mellem de grønne, er den god. Angular-correlation-scoren pr. fold (højre sidebar, i grader) fortæller derudover: mindre værdi = testen er tæt på træningen (hver test-kamera har en nær trænings-kamera, let test); større værdi = testen er langt fra træningen (hårdere generalisering).

 Du har optaget din truck-scene med 251 billeder, eksporterer via menupunkt M33 (Export SfM `transforms.json`) en nerfstudio-fil. Åbn holdout-vinduet ()H), indlæs JSON'en via „Open `transforms.json`...“, kig på globussen. $k=5$ (default) giver dig 5 folds. Klik på „Fold 3“ — se, om de røde markører er nogenlunde jævne. Hvis ja: „Export `fold-assignment.json`“, læg den eksporterede fil i reports-mappen, og ved næste training-run med `--benchmark` (eller tilsvarende Inspector-indstillinger) bruges præcis denne fold-opdeling som test-holdout — i stedet for standarden „every-8th“.

W23 Knappen „Open transforms.json...”

Toolbar øverst til venstre.



Åbner en filvælger, begrænset til JSON-filer. Efter bekræftelse indlæser holdout-modulet filen. Loaderen parser både nerfstudio-formatet (kamera-intrinsics plus liste over frames med billedstier og transform-matrix) og instant-ngp-formatet (samme opbygning). Pr. frame udtrækkes kigge-retningen fra transform-matricen (kameraets z-akse i lokalbasis) og gemmes. Hvis parsning fejler, vises en fejlmeddelelse i status-området.

Også via CLI: `--holdout-file /sti/til/transforms.json` starter vinduet direkte med indlæst fil.

 KORT FORTALT

Indlæser din kamera-pose-JSON. Standard er Nerfstudio- og Instant-NGP-eksporter. Radian-ceKit selv kan eksportere transforms.json via menu → Export → SfM.

W24 Vælger „Strategy“ (angular/linear)

Venstre sidebar, øverst.



Radio-vælger med to muligheder: Angular og Linear. Strategy-skift udløser automatisk en gen-beregning af folds. Kigge-retningerne er en liste over 3D-enhedsvektorer på sfæren; angular-strategien projicerer dem på longitudinalvinklen ϕ og sorterer, linear-strategien laver bare en modulo-opdeling over frame-index.

 KORT FORTALT

Angular til jævne orbit-optagelser (standard, sikkert), Linear kun hvis dine billeder ikke klynger rumligt.

W25 Slider „k Folds“

Venstre sidebar, i midten.



Slider fra 3 til 10, skridtstørrelse 1. Ved ændring udløses fold-beregningen automatisk, så folds-listen, train/test-indices og pr.-fold-scoren beregnes med det samme. Den valgte værdi vises som monospaced-digit-tekst til højre for labelet.

Tommelfingerregel: $k=5$ er standard (giver dig 20 % test pr. fold, hvilket er almindeligt for cross-validation). $k=10$ hvis du har meget data og brug for flere folds til statistisk sigende værdi. $k=3$ hvis du har lidt data.

 KORT FORTALT

Hvor mange folds opdelingen har. 5 er standard og passer næsten altid.

W26 Vælger „Test Fold“

Venstre sidebar, under k-sliden.



Menu-vælger. Muligheder er dynamisk $0..<k$, label „Fold 1“ til „Fold N“ (altså 1-indexed i UI'en, 0-indexed internt). Hvis den tidligere valgte index er $\geq k$ (f.eks. fordi du har reduceret k fra 10 til 5), nulstilles den automatisk til 0. Den valgte test-fold vises rødt i globussen, alle andre grønt.

 KORT FORTALT

Hvilken fold der lige nu er test-folden. Du kan klikke dig igenem og se, hvordan hver enkelt fold ser ud i globussen.

W27 Knappen „Export fold-assignment.json“

Venstre sidebar, nederst.



Åbner en gem-dialog med default-filnavn `fold-assignment.json`. Efter bekræftelse koder holdout-modulet den aktuelle opdeling til et JSON-skema (per-frame fold-tildeling plus strategy-meta-blok). Denne fil kan så gives med til næste træning med `--benchmark`, så samme holdout bruges til den endelige metrik-evaluering. Skrivefejl vises som fejltekst; succes i grøn tekst som „Saved to (filename)“.

 KORT FORTALT

Gemmer den aktuelle train/test-opdeling som JSON. Denne fil kan du så give direkte med ved træningen, så samme test-sæt bruges igen.

W28 SCNView (3D camera globe)

Midter-panel i holdout-vinduet.



SceneKit-globus-view. Scenen består af: en wireframe-kugle (radius 1.0, 36 segmenter, mørkegrå), tre farvede akse-stumper (rød/grøn/blå for X/Y/Z, hver 1.2 lang) og pr. kamera en lille markerkugle (radius 0.03) ved den tilsvarende kigge-retnings-position på enhedskuglen (let udenfor, så den ikke forsvinder i wireframe-kuglen). Markørerne genopbygges IKKE ved hver fold-ændring — genopbygning er kun nødvendig, når frame-listen ændres (altså når en ny JSON indlæses). I stedet kører en in-place-opdatering af materialefarverne pr. update: rødt for test-index, grønt for træning, lysgrå hvis hverken eller. Sådan forbliver slider-tikker performante også ved $N > 1000$ kameraer.

Kamera-styringen er aktiveret — du kan rotere globussen med musen, zoome, panorere. Belysning sørger for, at markørerne ikke ser flade ud. Baggrund er mørkegrå.

 KORT FORTALT

3D-globussen med kamera-positionerne. Grøn = træning, rød = test, lysegrå = ikke tildelt (forekommer ikke, alle kameraer hører til et sted). Med musen kan du rotere globussen og zoome.

W29 FoldCard (tap to select fold)

Højre sidebar, „Angular Correlation“-sektion.



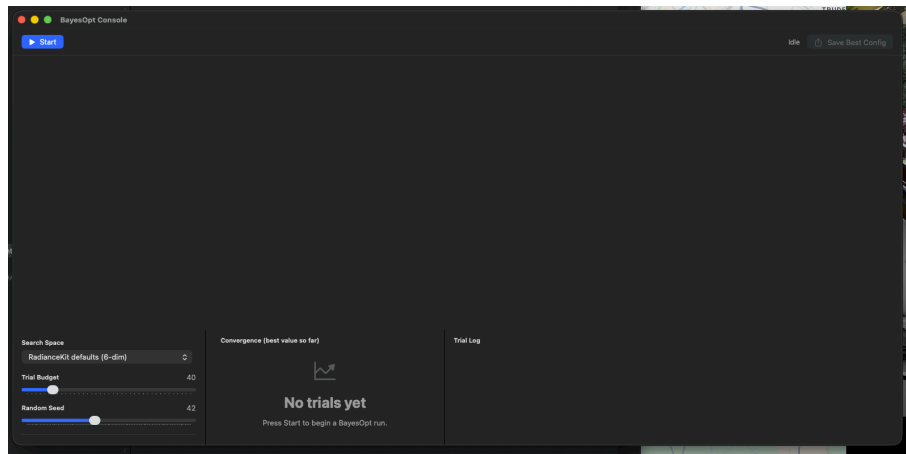
Pr. fold en kort-view — afrundet rektangel med 6 pt radius, padding 10, vertikalt layout med to linjer (øverst „Fold N“ + kamera-antal, nederst „Mean nearest angle:“ + værdi i grader). Baggrundsfarve betinget: aktiv fold = accentfarve halvtransparent, inaktive = neutralt standard-material. Tap vælger folden, og globussen farver om live.

„Mean nearest angle“-scoren er den gennemsnitlige mindste vinkel pr. test-kamera til den nærmeste trænings-kamera (i radian internt beregnet, vist i grader i UI'en).

 KORT FORTALT

Pr. fold et lille kort til højre med antal kameraer og gennemsnitsafstanden til den nærmeste trænings-kamera. Klik på det vælger denne fold som test.

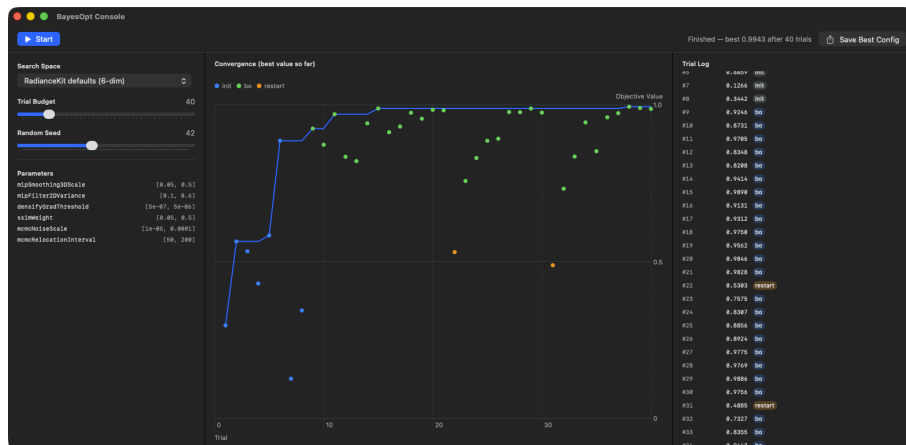
BayesOpt Console (W30–W39)



Figur 24: BayesOpt-konsol — tom tilstand før trial-start

Tom tilstand med search-space-vælger (RadianceKit defaults (6-dim)), trial-budget-slider (default 40), random-seed (42) og tre empty-panels for convergence-chart, trial log og search-space-parameter-liste.

Tom tilstand (efter første åbning) — convergence-chart og trial-tabel fylder sig, så snart en run er startet, se næste shot.



Figur 25: BayesOpt-konsol efter 40 trials — convergence-chart stiger stejlt indtil trial 15, Best Value 0.9943, trial log med init/bo/restart-tags

HVAD BILLEDET VISER Status øverst til højre „Finished — best 0.9943 after 40 trials“.

Venstre sidebar: search-space- vælger på RadianceKit defaults (6-dim), trial-budget 40, random seed 42. Parameter-liste viser de seks hyperparametre, der skal tunes, med deres værdiområder: 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]. Midt: convergence-chart (X = trial-index 1–40, Y = objective value 0–1) — grå punkter = initial samples (LHS), blå punkter = BayesOpt-acquisition, orange punkter = restart-trials (#22 og #31). Bedste-værdi-linjen stiger stejlt til trial ~7, derefter kun marginal forbedring til trial 15, fra da af fladt plateau ved

0.99+. Højre sidebar: trial-log #1–#34 med score + tag (init/bo/restart). Save- Best-Config-knappen øverst til højre skriver `bayesopt-best.json`.

Hvad det er: En Bayes-optimerings-konsol til hyperparameter-søgning. Bayes-Opt er en automatisk metode, der forsøger at finde det optimale punkt for en ukendt funktion med så få eksperimenter som muligt — typisk: „hvilken kombination af `mcmcMaxGaussians`, `capMultiplier`, `ssimWeight` og `gradThreshold` giver den bedste PSNR for min scene-klasse?“ I stedet for et grid på $6^4 = 1296$ trials prøver Bayes-Opt ca. 40–100 informerede trials og kommer dermed tæt på optimum.

Vigtigt: Den aktuelt leverede version i appen kører ikke optimeringen mod ægte trænings-runs (det ville tage dage), men mod et syntetisk demo-objektiv — et multi-modalt landskab med hill-climbing-karakter plus let noise. Det er bevidst sådan: vinduet skal vise dig optimerens adfærd (konvergens-forløb, sample-punkter, best-so-far) og lade dig forstå search-space-definitionerne. Til ægte trænings-drevne BayesOpt-kørsler (som udført i fase Q7 for scene-class-presets) bruges et separat offline CLI-workflow; vinduet er live-UI-varianten.


HVORNÅR ÅBNE Tre anvendelses-tilfælde: 1. Du vil forstå, hvordan BayesOpt arbejder — så start en demo-run og iagttag convergence-charten. 2. Du planlægger en ny scene-klasse (f.eks. „akvarier“ eller „antikke møbler“), som de indbyggede 10 presets ikke passer perfekt til. Definér mentalt et søgerum, tjek det her med „Bowl demo“ eller „Densify“-preset, eksportér så best-config som JSON og brug det som startpunkt for en ægte trænings-run.

3. Du vil inspicere de default-search-spaces, der er defineret i

RKBayesOpt-pakken (Mip-subset, RadianceKit defaults) — de listes i parameter-panelet i venstre sidebar.

SÅDAN FORTOLKES DET

- **Convergence-chart** (midter-kolonnen): Y = bedste hidtil opnåede objective-funktion-værdi. X = trial-index. I starten stejlt stigende (BayesOpt prøver de initial-samples tilfældigt, nogle af dem er heldige), derefter tiltagende fladt, fordi nær-optimum-regionen er udtømt. Hvis linjen forbliver flad i 20+ trials, kan du stoppe kørslen — flere trials giver intet mere. De enkelte punkter i charten er de individuelle trial-værdier (altså ikke „best so far“), farvet efter fase: grå = initial sample, blå = bayesopt acquisition, orange = restart.
- **Trial-tabel** (højre kolonne): #1, #2, #3, ... hver med værdi og fase-tag. Det hidtil bedste trial er markeret med en gul stjerne. Ud fra tabellen kan du identificere best-trial og se dets parameter-værdier senere ved eksport.
- **Search-space-inspector** (venstre sidebar): viser for det valgte preset alle parameter-navne og deres søgeområder `[lo, hi]`. Hvis du står ved presetet „RadianceKit defaults (6-dim)“, ser du f.eks. „densifyGradThreshold [5e-7, 5e-6]“ — altså log-uniform mellem disse to værdier.

 Vælg preset „RadianceKit defaults (6-dim)“, trial-budget 40, seed 42. Klik „Start“. Iagttag: de første 8 trials er grå (initial samples, LHS-Latin-Hypercube), de følgende blå (BayesOpt-erhvervet). Convergence-charten bliver stejl til trial ~15, derefter flader den ud. Ved trial ~30–40 stabiliseres den bedste værdi. Klik „Save Best Config“ —

en `bayesopt-best.json` gemmes med preset-navn, trial-index, værdi og de dekodede parameter-værdier. Denne JSON kan du så manuelt overtage i din preset-definition.

W30 Knappen „Start“



Toolbar til venstre, i idle/finished-state.



Nulstiller trial-listen, skifter til running-state, genererer en ny run-ID (til stale-detection ved flere start-klik) og opretter en frisk pause-gate. Derefter starter en baggrunds-task, der kører optimereren som asynkron stream. Initial- samples-størrelsen følger af $\min(8, \text{budget} / 4 + 1)$ — altså typisk 8 Latin-Hypercube-samples ved $\text{budget} \geq 28$, færre ved lille budget. Trial-updates modtages inkrementelt og vedhæftes listen. Stale-run-beskyttelse: hvis et andet start-klik i mellemtiden sætter run-ID på ny, kastes updates fra den gamle run væk.

Primary-action-stil til den prominente knap-look.

KORT FORTALT

Starter en frisk optimerings-kørsel med det aktuelle søgerum, budget og seed.

W31 Knappen „Pause“



Toolbar til venstre, i running-state.



Sætter pause-gate aktiv og skifter til paused-state. Den egentlige effekt: runneren venter i et 50-ms-polling-loop, før den evaluerer næste objective-funktion. Det betyder, at et igangværende trial føres til ende (det er syntetisk og varer kun mikrosekunder), men intet yderligere trial startes. Så snart resume kører, fortsætter den, hvor den slap.

KORT FORTALT

Holder kørslen tilbage. Aktuell beregning løber til ende, så pauser den.

W32 Knappen „Stop“

Toolbar til venstre, i running- og paused-state.



Afbryder runner-tasken, nuller referencen, løser pause-gate (hvis stadig paused), og skifter til finish-state (hvis trials findes) eller idle-state (hvis ingen). De allerede beregnede trials forbliver synlige i listen — stop sletter dem ikke. Destruktiv knap-rolle viser knappen rød, fordi den afbryder kørslen.

KORT FORTALT

Afbryder kørslen endeligt. Trials forbliver synlige, du kan stadig eksportere best-config.

W33 Knappen „Resume“

Toolbar til venstre, i paused-state.



Løser pause-gate og skifter tilbage til running-state. Runner-tasken kører allerede (den venter jo i polling-loopet); så snart loopet bemærker, at pausen er ophævet, kører den videre og starter næste trial.

KORT FORTALT

Fortsætter en paused kørsel.

W34 Knappen „Save Best Config“

Toolbar til højre, altid synlig (men deaktiveret, hvis ingen bestTrial findes).



Åbner en gem-dialog med default-filnavn `bayesopt-best.json`, begrænset til JSON. Efter bekræftelse bygges en payload-dictionary: preset-navn, trial-index, værdi (objective-score), parametre (dictionary af dekodede parameter-navne → værdier). Dekodningen projicerer de normaliserede søgerum-kordinater i $[0,1]^d$ tilbage til det oprindelige værdiområde (med log-uniform/linear/integer-skalaer tilsvarende). JSON-output er pretty-printed med sorterede keys. Ved skrivefejl ignoreres det (i den aktuelle demo-version) stille — ingen error-UI, fordi det er en demo-sti.

Knappen forbliver grå, så længe ingen trial er kørt.

KORT FORTALT

Gemmer parameter-værdierne for det hidtil bedste trial som JSON. Du kan så manuelt overtage disse værdier i din preset-konfiguration.

W35 Vælger „Search Space“-preset

HVOR

Venstre sidebar, øverst.



TEKNISK

Menu-vælger med fire preset-muligheder:

- „RadianceKit defaults (6-dim)“ — det fulde standard-søgerum med alle Q7-hyperparametre.
- „Mip subset (2-dim)“ — kun `mipSmoothing3DScale` [0.05, 0.5] log-uniform og `mipFilter2DVariance` [0.1, 0.6] linear. Nyttig, hvis du vil tune Mip-splating for en scene-klasse.
- „densify-until + ssim-weight + grad-thresh“ — tre densify-relevante parametre (`densifyGradThreshold` log-uniform, `ssimWeight` linear, `densifyUntilIter` integer).
- „Bowl demo (1-dim)“ — pædagogisk single-parameter-søgerum til „sådan virker BayesOpt“-demoer.

Mens en kørsel er aktiv, kan søgerummet ikke skiftes (ville forvirre optimeren).

KORT FORTALT

Hvilket hyperparameter-søgerum BayesOpt gennemsøger. Standard er „RadianceKit defaults“. Til målrettede Mip-tuning-forsøg „Mip subset“. Til at forstå, hvordan BayesOpt arbejder, „Bowl demo“.

W36 Slider „Trial Budget“

HVOR

Venstre sidebar, under search-space-vælgeren.



TEKNISK

Slider fra 10 til 200, skridtstørrelse 5. Default 40. Det betyder: BayesOpt må maksimalt lave N trials. Heraf er de første par initial samples (Latin-Hypercube), resten er ægte BayesOpt-trials. Tommelfingerregler i praksis: et søgerum med d dimensioner kræver ca. $10d$ til $20d$ trials for et godt optimum. Ved 6-dim defaults altså 60–120, ved 2-dim Mip-subset 20–40, ved 1-dim Bowl-demo 10–20.

Under kørslen er slideren deaktiveret.

KORT FORTALT

Hvor mange optimerings-forsøg maksimalt. Flere forsøg = bedre løsning, men koster mere tid. 40 er god default for demo-objective.

W37 Slider „Random Seed“

Venstre sidebar, under budget-sliden.



Slider fra 1 til 100, skridtstørrelse 1. Default 42. Seedet sendes både til de initiale Latin-Hypercube-samples og

til noise-komponenten i demo-objective. Reproducerbarhed: samme seed + samme søgerum + samme budget giver præcis identisk trial-sekvens. Nyttig til „får alle dine kolleger samme kørsel, hvis de bygger demoen efter?“. Under kørslen deaktiveret.

KORT FORTALT

Styrer den tilfældige generator. Samme seed = samme kørsel — til at reproducere.

W38 Chart (convergence)

Midt-kolonnen i vinduet.



Swift-Charts-diagram med to lag: 1. en linje for „best-value-so-far“ pr. trial — en monotont stigende eller konstant kurve i accentfarve. 2. et punkt pr. trial med den individuelle objective-værdi, farvet efter fase. Symbol-størrelse

40. Tre fase-labels: „init“ (grå), „bo“ (blå), „re-start“ (orange).

En lille legende viser fase-farverne øverst til venstre. Hvis trial-listen er tom (før første start), vises i stedet en empty-state-visning med chart-ikon og hintet „Press Start to begin a BayesOpt run.“.

KORT FORTALT

Forløbs-charten. Den ubrudte linje er „bedste hidtil fundne løsning“; punkterne er de enkelte forsøg. Hvis linjen forbliver flad i lang tid, har BayesOpt fundet optimum.

W39 Table (trial log)



HVOR

Højre kolonne i vinduet.



TEKNISK

Scroll-område med lazy stablede trial-linjer. Pr. linje et horisontalt stack: trial-nummer (3-cifret monospaced, til venstre), værdi (monospaced, højrejusteret, 70 pt bred), fase-tag (capsule, fyldt med fasefarve ved 25 % opacity), valgfrit en gul stjerne, hvis dette trial er det aktuelt bedste. En auto-scroll-mekanisme springer automatisk til enden, så snart et nyt trial kommer til — så du kan medlæse live-forløbet i skærmens bund uden selv at scrolle.

KORT FORTALT

Tabellen over alle forsøg. Værdi, fase, stjerne for det bedste. Scroller automatisk med, nye trials optræder nederst.

Hovedvindue: loss-forløb og gaussian-count (I39–I41, krydshenvisning)

Tre af Inspector-visningerne i hovedvinduet fortjener en egen forklaring, fordi de ses konstant under en kørende træning, og der er vigtige tommefingerregler for, hvornår forløbet ser sundt ud. Visningerne er i Inspektoren under „Loss Chart“-sektionen (se kapitel 2 — Inspector) og supplerer holdout-analysen fra hjælpevinduet ovenfor.

Hvornår er loss-kurven sund? En sund loss-kurve viser tre faser: (1) **Warmup** — i de første 200–500 iterationer falder loss'en stejl fra højt (typisk 0.15–0.25 for L1+SSIM-kombineret afhængigt af scenen) til ca. halvdelen. Hvis loss'en IKKE falder i denne fase, er input som regel forkert (billeder defekte, SfM-positioner dårlige, antal initial-gaussians for lille). (2) **Densification** — mellem ~500 og densifyUntilIteration (klassisk 15K, MCMC til 20K eller 25K) falder loss'en videre, ofte med små spring nedad, når densify-operationerne indsætter nye gaussians, og optimeren udnytter dem. Gaussian-count stiger i denne fase. (3) **Refinement** — derefter kører loss'en ud i en fladere hale. Typiske slutværdier: Tanks-&-Temples Truck med P4 Quality lander ved $L1 \approx 0.023$, Horse med Full Classic V546 ved $L1 \approx 0.0230$, outdoor-Mip-NeRF360-scener ofte dårligere (0.04–0.07).

Hvad betyder et plateau? Et plateau (loss-kurven løber horisontalt over flere tusinde iterationer) har to fortolkninger: (a) modellen er konvergeret, yderligere træning giver intet mere — det er den gode situation. (b) Modellen er stuck (lokalt minimum, dårlig gradient-information, et cap ved buffer-grænsen) — den dårlige situation. Begge ser identiske ud i charten. Skellen: kig på gaussian-count. Hvis den også er flad OG tæt på MCMC-cap'et (f.eks. 150K af 150K ved `.fullMCMC`), er du ved grænsen — enten hæv cap'et eller acceptér plateauet. Hvis gaussian-count stadig vokser, men loss'en ikke falder, hænger den fast.

Hvornår afbryde vs videretræne? Tommelfingerregel: 10K iterationer uden forbedring af min-loss → afbryd, yderligere iterationer er spildt. Før: kan du via Cmd+T (training-menuen → Continue Training → +5K iterations) stadig hænge en forlængelse på, hvis du ser grænsemæssig forbedring. Advarsel: ved MCMC er plateauet ofte ægte — cap'et er den naturlige grænse.

Gaussian-count-plateau er IKKE et „færdig“-signal. Det betyder kun, at MCMC har nået cap'et, eller at Classic densification er udtømt. Det egentlige „færdig“-spørgsmål stilles først af holdout-analysen — PSNR/SSIM/LPIPS på et uafhængigt test-sæt, evalueret i holdout-vinduet (W23–W29) eller via `--benchmark` -flag.

PSNR/holdout er sandheden, loss kun proxy. Loss'en er en relativ metrik: den falder, mens din model tilpasser sig trænings-views. En lav loss betyder dog ikke automatisk god model — hvis modellen har lært trænings-billederne udenad (overfitting), ville loss'en være lille, men PSNR på usete views (holdout) ville være dårlig. Derfor: til den endelige kvalitetsbedømmelse kig altid på holdout-metrikker, ikke på end-loss alene.

Tommelfingerregel-boks

- User Guide og Keyboard Shortcuts er statisk hjælp — ved stikordsspørgsmål hurtigt, til dybde brug denne manual.
- Manage Storage åbnes, så snart disken falder under 10 % fri plads. Logs og imports-staging er de sædvanlige syndere.
- Pareto Dashboard er først meningsfuldt efter mindst tre eller fire trænings-reports. X-akse = omkostninger (time / Gs), Y-akse = kvalitet (PSNR / SSIM). Pareto-fronten viser de effektive kombinationer.
- Holdout Analysis bruges, før du offentliggør PSNR-benchmarks med andre — det sikrer dig, at dit test-sæt virkelig er repræsentativt.
- BayesOpt Console er primært et lærings- og inspektions-værktøj til søgerum-definitioner. Til ægte trænings-drevet hyperparameter-tuning brug offline CLI-workflowen.
- Loss-plateau og gaussian-count-plateau skal fortolkes adskilt. Cap-grænse er ikke et „færdig“-signal. Ægte kvalitet måler kun holdout-PSNR.
- 10K iterationer uden min-loss-forbedring → stop træning.

KAPITEL

Kapitel 6 — Trænings-konfiguration

```
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
  }
}
```

Figur 26: Preview-preset eksporteret som JSON og vist i TextEdit — felter id/name/category/version/createdAt/description, trainingConfig med alle relevante parametre (maxIterations 5000, densifyUntilIteration 3500, ssimWeight 0.20, renderScale 0.50, strategy classic, cameraAlignment applePhotogrammetry, densifyGradThreshold 2.0e-06, opacityResetInterval 3000, minOpacity 0.005, seks bool-toggles)

HVAD BILLEDET VISER En typisk preset-JSON-eksport. Top-level-felter: `id` (UUID), `name`, (`classic` | `mcmc` | `sceneClass` | `custom`), (skema-version), (timestamp), (fri tekst). Det indlejrede -objekt indeholder de for reproducerbarhed kritiske parametre — ved import deserialiseres hele blokken til `TrainingConfig`-strukturen, og defaults fra app-versionen fylder de felter, der mangler i JSON'en (f.eks. efter app-opdatering). Den, der overdrager en preset til en anden Mac, sender simpelthen denne JSON-fil.

`TrainingConfig`-strukturen er hjertet i hver trænings-kørsel i RadianceKit. Den samler hver parameter, der påvirker træningen — fra max iterations-antallet over de otte læringsrater til specialfelter for MCMC, Mip-splatting, curriculum og scene-aware caplogik. Du redigerer den i sidebaren under trænings-konfigurations-sektionen (Expert View), gemmer den som preset eller giver den videre som JSON-eksport til en anden Mac. Ved træning fryses præcis dette objekt og overgives til GPU-backend.

Dette kapitel er reference-materiale til power-users og script-forfattere. Det lister alle 81 offentlige felter, de 9 statiske presets og den ene offentlige metode. Kildefil er `TrainingConfig.swift` — ved tvivl gælder den dér gemte doc-comment og `initializer-default`'en som `source-of-truth`.

BEMÆRK · UI VS. PRESET/CLI

Kun 12 af de 81 felter har en direkte slider, toggle eller vælger i Inspektoren (sandboxed App Store-build): **T1, T2, T17, T20, T22, T38, T56–T58, T60, T61, T73**. De resterende 69 felter sættes via den valgte **preset** (kapitel 7) og kan kun overskrives direkte via **CLI-flag** (se kapitel 5). Denne adskillelse er tilsigtet: defaults forbliver stabile og produktionsafprøvede, power-users har alligevel en flugtluge. Hvis et felt særligt interesserer dig: kig først i kapitel 2 (Inspector) og kapitel 5 (CLI), om du kan nå det uden JSON-pillerier.

Indholdsfortegnelse:

1. Iteration (T1–T2)
2. Learning rates (T3–T10)
3. Densification — Classic (T11–T16)
4. Loss (T17–T20)
5. SH-degree-progression (T21)
6. Performance (T22–T25)
7. Diagnose og punktsky-forberedelse (T26–T30)
8. Regularisering (T31–T37)
9. Refinement (T38–T44)
10. Sky-dome (T45–T48)
11. Adam + LR-schedule (T49–T55)
12. Post-processing + Apple AI (T56–T60)
13. MCMC-densification (T61–T73)
14. Mip-splatting (Q1.5) (T74–T76)
15. Adaptive densification (Q5) (T77–T79)
16. Curriculum (Q6) (T80–T81)
17. Statiske presets (TP1–TP9)
18. Metoden:
19. Hvilket felt til hvad? (cheat sheet)
20. Farlige felter

Iteration (T1–T2)

T1 maxiterations

DE TALJER

Default: 30 000 (initializer), 35 000 (`.full`), 200 000 (`.fullMCMC`) **Range:** 1 000 – 500 000 (UI-sli-der), ingen hård øvre grænse i logikken **Defined in:**

TEKNISK

Samlet antal trænings-iterationer, som backend gennemløber. En iteration består af et forward-render af et enkelt trænings-kamera, et backward-pas over alle loss-komponenter (L1 + SSIM + valgfrie regulariseringer + sky-mask) og et Adam-optimer-skridt. Dette tal virker direkte ind på andre schedules: position-læringsraten følger en cosine-annealing-kurve fra 0 til enten T1 selv eller til T49 `positionLRScheduleEndIteration`; densification stopper ved T2 `densifyUntilIteration`; MCMC-noise-decay slutter ved T69 `mcmcNoiseDecayEnd`; SH-degree-opgraderinger sker ved de tre i T21 definerede markeringer. Ved klassisk densification ligger det empirisk fundne sweet spot ved 20 000–35 000 iterationer (sessioner 1–32, V546-tests), ved MCMC ved 60 000–200 000 (V534). En drastisk forhøjelse ud over de i presets gemte værdier giver sjældent ekstra kvalitet — Adam-momentum mætter, og uden LR-decay-ende stagnerer loss'en. Omvendt fører underskridelse af ~5 000 til ufuldstændigt konvergerede geometrier (density-control har for lidt tid til klon/split).

KORT FORTALT

Hvor længe appen regner. Flere iterationer = bedre resultat, men på et tidspunkt ikke længere mærkbart bedre, til gengæld meget længere. Presets er valgt sådan, at du uden at tænke har en god værdi: Quick 1 000, Preview 5 000, Balanced 20 000, Quality 35 000, MCMC Quality 200 000. Hvis du selv drejer på det, gælder: ved MCMC må du gerne højt op (100 000–200 000), ved Classic ikke over 40 000 — giver så ikke mere.

T2 densifyUntilIteration

DETALJER

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

TEKNISK

Iteration, hvorfra densification holder op. Indtil da kloner, splittes og prunes gaussians efter reglerne parametriseret i `T11–T16` (Classic) eller `T67–T70` (MCMC); derefter forbliver gaussian-antallet konstant, og kun positioner, rotationer, skaler, opaciteter og SH-koefficienter optimeres (refinement-fase). I 3DGS-originalpapiret ligger værdien ved 50 % af `T1`, i Radiances `.full`-preset ved kun ~14 % (5 000 af 35 000) — følge af V310/V338-eksperimenter, der viste, at videre densificering efter 5 000 iterationer snarere forværrer resultatet (flere floaters, mere lagerbehov, ingen kvalitetsgevinst). MCMC derimod lader relocation køre til 80 % af `T1` (V504b), fordi MCMC ikke producerer skadelige floaters. Hvis `T2` vælges for lille (< 1 000), opstår der for få gaussians; for stor ved Classic (> 50 % af `T1`) fører til overgrowth og RGB-saturation-outliers (se outdoor-overtraining-findings).

KORT FORTALT

Hvor længe appen må generere nye gaussians. Derefter forfines kun det, der allerede er. Ved klassisk træning med 35 000 iterationer er 5 000 her den rigtige værdi — alt derover gør scenen mere plumret. Ved MCMC er det 80 % af samlede iterationer (altså 160 000 ved 200 000-kørsel). Hvis du ændrer Quality-preset, lad hellere dette felt være.

Learning rates (T3–T10)

T3 positionLearningRate

DETALJER

Default: 0.00016 **Range:** 1e-7 – 1e-3 (anbefalet)

Defined in:

TEKNISK

Adam-læringsrate for XYZ-positionen af hver gaussian ved trænings-start (iteration 0). Følger en cosine-annealing-kurve og synker i træningens forløb til T4 `positionLearningRateFinal`. Default 0.00016 stammer fra 3DGS-originalpapiret (Kerbl et al. 2023) og kan i RadiancKit ikke skaleres med billedopløsningen — positionen bevæger sig i verdenskoordinatsystemet, ikke i pixel-rummet. En markant forhøjelse (> 0.0005) bevirker, at gaussians springer over lange distancer, og loss'en bliver ustabil; værdier markant under (< 0.00005) fører til, at fejl-initialiserede punktskyer aldrig finder deres plads. V414 testede en fordobling af init-værdien \rightarrow 16.8 % dårligere L1-loss; V544a-tuning bekræftede paper-default'en som optimal. Bemærk: ved `.fullMCMC` lader vi denne værdi bevidst stå ved default — MCMC har brug for konstante læringsrater til sin relocation-logik, så tuning her giver intet.

KORT FORTALT

Hvor hurtigt splat-punkterne må bevæge sig i rummet. Standardværdien er meget veljusteret og kræver i virkeligheden ingen ændring. Kun hvis splats i billedet „flagrer“ eller et helt hjørne mangler, fordi intet vil bevæge sig dertil, ville læringsraten være et punkt at dreje på — men så passer typisk noget andet allerede ikke (kamera-positioner, initial-punktsky).

T4 positionLearningRateFinal DETALJER

Default: 0.0000016 (initializer + paper), 0.000016 (`.full`, `.fullMCMC` — 10× højere) **Range:** 0 – **Defined in:**

 TEKNISK

Slutværdi for position-LR-cosine-annealing-kurven. Når enten ved `T1 maxIterations` eller, hvis sat, ved `T49 positionLRScheduleEndIteration`. RadiancEKits `.full` -preset bruger 0.000016 — altså 10× højere end paper-default'en 0.0000016. V420-eksperimenter viste, at 0.5× af slutværdien (0.000008) forværrer loss'en med 6.4 %; V414 viste, at 2× init-værdi forværrer den med 16.8 %. Den høje slutværdi er ikke tradeoff, men bevidst valg: ved for stærk decay mister gaussians under refinement-fasen evnen til at tilpasse sig nytilkomne densification-kandidater. Via V431/V433-udvidelsen kan schedule-fasen forkortes (`T49 < T1`), så `T4` allerede nås før trænings-slut, og resten af træningen kører ved konstant mini-LR — typisk konfiguration: `T49 = 20 000`, `T1 = 35 000`, refinement altså ved 0.000016 i 15 000 iterationer.

 KORT FORTALT

Hvor langsom position-læringsraten bliver i slutningen af træningen. Vi har bevidst sat den mindre aggressivt end originalpapiret — splats kan stadig wackele lidt til slutningen, det gør dem skarpere. Hvis du drejer på det: højere = mere urolige splats i slutningen, lavere = splats kan ikke længere tilpasse sig, når nye dukker op.

T5 shDCLearningRate

DETALJER

Default: 0.0025 (initializer + paper), 0.005 (`.full` og alle MCMC-presets — 2x) **Range:** 0.0001 – 0.05
Defined in:

TEKNISK

Adam-læringsrate for DC-andelen (degree 0, altså konstant albedo) af spherical-harmonic-farven. SH-DC svarer til den retnings-uafhængige grundtone af en gaussian, så at sige „basisfarven“. V176- og V188-eksperimenter fandt 2x højere end paper-default'en optimal — hurtigere farve-konvergens, særligt fordi SH-DC ved kort træning (5 000 iterationer) ellers ikke kommer i form. I modsætning til de geometriske LR'er har SH-DC ingen decay; læringsraten forbliver konstant over alle iterationer (eller følger kun den valgfri extended-phase-decay fra T51). V416 testede en firedobling til 0.01 → 6.4 % dårligere loss ved beta2=0.99-Adam.

KORT FORTALT

Hvor hurtigt grundfarven for hver splat tilpasser sig. Værdien ændrer man næsten aldrig selv — presets har den rigtige værdi. Højere ville gå hurtigere, men kan føre til ustabile farver.

T6 shRestLearningRate

DETALJER

Default: 0.000125 (initializer + paper), 0.00025 (`.full` og MCMC — 2x) **Range:** 0.000001 – 0.005
Defined in:

TEKNISK

Adam-læringsrate for SH-koefficienterne af højere orden (degree 1, 2, 3 — altså de view-direction-afhængige farveandele, der sørger for højlys, refleksioner og blød skygning). 20x mindre end T5 ifølge paper-konvention, fordi disse koefficienter vokser kvadratisk i antal (3 for degree 1, 5 for degree 2, 7 for degree 3 → i alt 15 floats pr. gaussian) og uden mindre læringsrate ville overmættte billedet. Frigives i to trin — indtil første markering i T21 `shDegreeUpgradeIterations` er kun degree 0 aktiv (altså kun T5), derefter 1, senere 2, til sidst 3. Lave værdier her er særligt vigtige på scener med meget diffus belysning; ved meget blanke overflader (billa, vand) giver det ingen mening at dreje på det — SH-repræsentationen selv er begrænset.

KORT FORTALT

Hvor hurtigt de retnings-afhængige farve-effekter (refleksioner, glans) lærer. Som standard meget lille, fordi ellers blanker alt. Værdien lader man bedre stå — den, der vil få bedre højlys, er bedre tjent med MCMC og længere træningstid end med denne LR.

T7 opacityLearningRate

DETALJER

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

TEKNISK

Adam-læringsrate for logit-opaciteten af hver gaussian. Appen gemmer opacity som ubegrænset float-værdi og transformerer den med sigmoid til $[0, 1]$; LR'en virker i logit-space. Paper-default 0.05 er efter V50-tests (best single-run L1 0.1664) genoprettet, V71 reverterede V67's 0.025. V188-fordoblingen til 0.1 gør pruning mere effektiv — døde gaussians falder hurtigere under `T14 pruneOpacityThreshold`. V418 viste: 0.05 med `beta2=0.99`-Adam er 7.1 % dårligere end 0.1 — vekselvirkningen med Adam-konfigurationen er ikke trivial. Lave værdier (< 0.01) fører til, at „døde“ gaussians ligger evigt rundt og bruger lager; for høje værdier (> 0.5) kan føre til opacity-explosion, derfor klampes logit-værdien i optimeren til $[-15, 3]$ (se note „Opacity Explosion Prevention“ i CLAUDE.md).

KORT FORTALT

Hvor hurtigt splats bliver gennemsigtige eller uigennemsigtige. Vigtigt for opryddningen — splats, der ikke bidrager med noget, skal forsvinde hurtigt, så der ikke opstår en slør. Standardværdien passer, kun professionelle ændrer den.

T8 opacityLearningRateFinal

DETALJER

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

TEKNISK

Valgfri cosine-decay-slutværdi for opacity-LR'en (V427). Når 0.0, er decay deaktiveret, og opacity-LR'en forbliver konstant ved T7 over hele træningen. V427 testede en decay $0.1 \rightarrow 0.01$ — resultat 11.5 % dårligere loss; reverteret, deraf default'en „fra“. Hypotesen bag feltet: i refinement-fasen kunne konstant opacity-LR føre til oscillation, så splats, der allerede har nået det rigtige mål for transparens, forskydes igen af tilfældige gradient-udsving. Empirisk bekræftes det ikke — logit-clamping-logikken fanger det alligevel. Feltet forbliver tilgængeligt for fremtidige eksperimenter; også meget lange MCMC-kørsler ($> 500K$ iterationer) kunne drage nytte af det.

KORT FORTALT

Om opacity-læringsraten skal blive mindre mod slutningen. Standard: nej. Vi har prøvet det, var dårligere, lader det deaktivere. Bliv ved 0.

T9 scaleLearningRate

DETALJER

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

TEKNISK

Adam-læringsrate for de tre skala-komponenter af hver gaussian i log-space (RadianceKit gemmer $\log(\text{scale})$, så skalaerne forbliver positive). Paper-default 0.005, i RadianceKit fordoblet til 0.01 for bedre skala-konvergens ved de optimerede læringsrate-konfigurationer. V423-eksperiment: 0.005 med $\text{beta2}=0.99$ -Adam \rightarrow 18.7 % dårligere loss og synligt for få gaussians (density-control kunne ikke klone, fordi skala- updates var for langsomme). Skala kontrollerer udstrækningen af hver gaussian — for hurtig læring fører til „needle“-gaussians (ekstremt lange tynde splats, se T34 `scaleRatioPruneThreshold`), for langsom læring lader splats forblive for kompakte, og density-control skal splitte for ofte.

KORT FORTALT

Hvor hurtigt formen for splats tilpasser sig. Standard er god. Hvis du skruer det op, får du gerne „nåle“-splats — ekstreme lange tynde dråber, der får billedet til at floate.

T10 rotationLearningRate

DETALJER

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

TEKNISK

Adam-læringsrate for de fire quaternion-komponenter af hver gaussian. Quaternion'en normaliseres igen i hvert optimizer-skridt efter Adam-update'en ($L2\text{-norm} = 1$) — ellers ville kovariansmatricen blive degenereret. RadianceKit fordobler paper-default'en i quality-presets, fordi rotation har mindre absolute gradient-magnituder end skala / position (på enhedssfæren forbliver hvert skridt kort), og uden 2x ville rotationen i 35 000-iterations-vinduet være markant under-konvergeret. V188 dokumenterer. På NeRF-Blender-scener (Lego, Chair) virker rotation særligt — objekternes kanter retter sig først korrekt ud efter 5 000–10 000 iterationer.

KORT FORTALT

Hvor hurtigt splats lærer at dreje — altså komme i den rette retning på overfladen af et objekt. Standard passer. Sagt anderledes: hvis splats ser ud som skrå klodser i stedet for at lægge sig om overfladen, er træningstiden snarere for kort, ikke denne læringsrate for lav.

Densification — Classic (T11–T16)

T11 densifyGradThreshold

DETLJER

Default: 0.000002 (initializer, kalibreret for 0.5× opløsning), 0.0000011 (`.full` , kalibreret for 1.0×), 0.000004 (`.quickTest` , kalibreret for 0.25×), 2e-7 (`.fullClassicPaper`) **Range:** 1e-8 – 1e-3 (opløsningsafhængig) **Defined in:**

TEKNISK

Tærskel for L2-normen af den skærmrums-projicerede gradient `dMean2D` , over hvilken en gaussian markeres til klon eller split. Den absolutte værdi afhænger direkte af trænings-opløsningen — `dMean2D` skalerer ca. som $1/\text{opløsning}^2$ (flere pixel = mindre per-pixel-gradienten). Derfor kræver hvert T22 `trainingRenderScale`-trin en kalibreret tærskel: 0.25× → 4e-6, 0.5× → 2e-6, 1.0× → 5e-8 ... 1.1e-6 (`.full`). Paper-default 0.0002 er NDC-normaliseret og ikke direkte sammenlignelig i Radiances Kits world-space-pipeline. Med V440 indkøbet T52 `adaptiveDensifyThreshold`-flag kan værdien beregnes i runtime ud fra p98 i den aktuelle gradient-fordeling — men V440 testede det på rigtige scener og producerede 63 K gaussians (katastrofalt pruning-tab); flaget forbliver fra. Q5 (T77–T79) leverer en alternativ adaptiv logik via rolling median. **Dette felt er ikke ufarligt** — halvering genererer 2–4× flere gaussians (lager-pres, OOM-risiko); fordobling kan under-densificere scenen.

KORT FORTALT

Hvor følsom appen er, når den skal beslutte, om et splat er for lidt repræsenteret og skal mangfoldiggøres. Lav værdi = mere følsom = flere splats. Højere = færre splats. Det er en af de farligste værdier overhovedet: for lav, og Mac'en fyldes med millioner af splats og crasher måske. Lad feltet være, eller ændr det kun i skridt på 10 %.

T12 densifyFromIteration

DETLJER

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

TEKNISK

Første iteration, fra hvilken densification bliver aktiv. Før det sker kun „nøgen“ læring på den initiale SfM-punktsky, uden at nye gaussians genereres. Default 500 stammer fra 3DGS-papiret og giver initialiseringen tid til at stabilisere sig — hvis der allerede densificeres fra iteration 0, kloner forkert positionerede SfM-punkter mange gange, før de overhovedet finder deres rigtige plads. V349 testede 1000 → let dårligere loss; default'en er optimal.

KORT FORTALT

Hvornår appen første gang begynder at kloner splats. Før det lærer den kun på de allerede tilstedeværende punkter. 500 er standardværdien — giver appen tid nok til først at orientere sig, før den mangfoldiggør.

T13 densifyInterval

DETLJER

Default: 100 (initializer, MCMC), 200 (`.full`) **Range:** 50 – 1 000 **Defined in:**

TEKNISK

Hvor mange iterationer der ligger mellem to densification-skridt. Ved paper-default 100 — hver 100 iterationer evalueres listen af densify-kandidater, kloner/ splittes, og samtidig fjernes listen af prune-kandidater ($\text{sigmoid}(\text{opacity}) < T_{14} \text{pruneOpacityThreshold}$). V112-tests fandt 200 som optimal for `.full` — det aflaster GPU'en, fordi færre reorganisations-pas kører, og giver hver gaussian mere tid til at falde til ro efter en klon-handling. V417 testede 100 med $\text{beta2}=0.99$ → 5.8 % dårligere (957 K gaussians, overdensificering). Ved MCMC fortolkes samme felt som relocation-interval; se T67 `mcmcRelocationInterval` for den MCMC-specifikke logik.

KORT FORTALT

Hvor ofte appen kigger efter nye splats. 100 = ofte, 200 = mellem. Højere betyder: hvert splat har længere tid til at indrette sig, før det mangfoldiggøres igen. Det er godt. At sænke til 50 kan optage GPU'en vedvarende uden, at det bliver nævneværdigt bedre.

T14 `pruneOpacityThreshold` DETALJER

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

 TEKNISK

Sigmoid-opacitets-tærskel, under hvilken en gaussian slettes ved næste densification-step. Virker sammen med T7 `opacityLearningRate` og `logit-clamp`-logikken i optimeren. V393 sænkede default'en fra 0.005 til 0.001 i `.full` — følge: splats, som kun spiller en rolle under eksotiske kigge-vinkler, bevares længere og bidrager til SH-detajle. V394 testede 0.0001 → let dårligere (for lidt pruning, lager spildt). Vigtigt: `density-control` skal ALTID `prune`, selv hvis `buffer-kapaciteten` allerede er fuld af andre tiltag (se „Density Control Must Always Prune“ i CLAUDE.md) — ellers akkumulerer døde gaussians, og `count` fryser fast.

 KORT FORTALT

Hvornår et splat regnes som „gennemsigtigt nok“ til at blive slettet. 0.005 er paper-standard, vi har i Quality 0.001 — altså vi giver splats en længere chance. Det gør blødt lys og svage skygger bedre at fremstille. At sætte højere (over 0.01) lader splat-antallet hurtigt synke — kan være meningsfuldt ved lager-knaphed, men koster detalje.

T15 `opacityResetInterval` DETALJER

Default: 3 000 (initializer + paper), 100 000 (`.full` = reelt deaktiveret), 200 000 (`.fullMCMC` = deaktiveret) **Range:** 1 000 – 100 000+ **Defined in:**

 TEKNISK

Hver hvor mange iterationer nulstilles opaciteten af alle gaussians til en lav værdi (~0.01) — en foranstaltning fra 3DGS-papiret for at vurdere „frosne“ splats på ny. V194 viste, at med `RadianceKits` warmup + stochastic-trænings-setup + 2× lærings-rater koster `opacity-reset` 5.5 % kvalitet, og `logit-clamp` dækker `reset-funktionen` allerede. Derfor i `.full` praktisk deaktiveret (100 000 > 35 000 = aldrig udløst). V421 testede `reset` hver 3 000 med `beta2=0.99` → 4.9 % dårligere; reverteret. Ved `.fullClassicPaper` (Q1.5-A, paper-tro test) er det bevidst sat tilbage til 3 000 — det var en af de håndtag, hvormed paper-magnitude-gaussian-budgetterne skulle nås.

 KORT FORTALT

Hver hvor mange iterationer appen nulstiller synligheden af alle splats til „næsten usynlig“ — en slags `reset-knap` for `opacity`. Ved os deaktiveret (værdi så høj, at det aldrig sker), fordi andre mekanismer gør det overflødig. Kun ved paper-tro eksperimenter at slå til.

T16 maxScreenSize DETALJER

Default: 0.0 (= deaktiveret) **Range:** 0 (off) eller > 0
Defined in:

 TEKNISK

Maksimal skærums-størrelse (i projicerede pixel), som en gaussian må nå, før den tvunget splittes. Værdien er sat til 0 (V48 testede og reverterede) — RadianceKits density-control bruger i stedet world-space-skala-tærsklen fra `dMean2D` -logikken. Forbliver i feltkataloget, fordi fremtidige eksperimenter med Mip-splatting (T74–T76) eller scene-specifikke splatting-strategier kunne drage nytte af det. Aktivering (værdi > 0, f.eks. 20) ville tvinge splats, der er blevet meget store på skærmen, til at dele sig — relevant ved store, glatte vægoverflader, hvor et enkelt riesensplat tilbyder for lidt detalje.

 KORT FORTALT

Begrænsning på, hvor stort et enkelt splat må blive på skærmen. Ved os fra. Tændt ville det betyde, at kæmpe flade splats (f.eks. på en væg) tvinges til at deles i flere små. Lad det være fra, hvis ikke der udtrykkeligt eksperimenteres med det.

Loss (T17–T20)

T17 `ssimWeight`

DE TALJER

Default: 0.2 (initializer + paper + `.full`), 0.05 (alle MCMC-presets) **Range:** 0.0 – 1.0 **Defined in:**

TEKNISK

Vægt af D-SSIM-andelen i den kombinerede loss-funktion $loss = (1 - \lambda) * L1 + \lambda * D\text{-SSIM}$, hvor $\lambda =$ T17. 3DGS-paper-default 0.2 er optimal for Classic-densification — V383 testede 0.3 → 28.9 % dårligere, V373b bekræftede 0.2 som sweet spot. For MCMC blev det i V521b/V534 uafhængigt fastslået: 0.05 er optimal, fordi MCMC via sin stokastiske eksploration har brug for en stærkere L1-signal-andel — højere SSIM-vægte ville udvande relocation-beslutningerne. SSIM er markant dyrere at beregne end L1 (lokale 11×11-vinduer over hele billedet); RadianceKit bruger en MPS-accelereret implementering, der bliver under 1 ms pr. 1080p-billede. Q7-BayesOpt-sweeps fandt scene-specifikke optima mellem 0.05 (`.outdoorPreset` : 0.082) og 0.171 (`.indoorPreset`).

KORT FORTALT

Hvor vigtigt appen ud over „hver pixel stemmer“ også finder „strukturer er ens“. 0.2 er standarden og leverer et godt billede. Lavere = pixel-nøjagtigere, men kan få blødere overgange. Højere = mere strukturligt, men detaljer bliver blødere. Lad presets afgøre.

T18 **ssimWeightRefinement** **DETALJER**

Default: 0.0 (= „intet skift, behold ssimWeight“)

Range: 0 eller 0 – 1.0 **Defined in:**

 **TEKNISK**

Valgfri SSIM-værdi for refinement-fasen efter T2 densifyUntilIteration. V428 testede 0.2 → 0.3 i refinement → 16 % dårligere loss (både L1 og SSIM forværredes); reverteret, deraf default 0.0. Hypotesen bag feltet var, at efter densification — når der ikke længere opstår nye gaussians — en stærkere SSIM-andel ville maksimere den strukturelle skarp-hed. Empirisk forkert: at hæve SSIM-vægten betyder indirekte at sænke L1-vægten, og L1 er det markant mere sigende signal i final-refinement-fasen. Feltet forbliver tilgængeligt for fremtidige eksperimenter med perceptual loss (T60) eller edge-loss (T19), hvor en refinement-specifik loss-komposition kunne være meningsfuld.

 **KORT FORTALT**

Specialindstilling til anden trænings-fase (refinement efter splot-mangfoldiggørelse). Ved 0.0: samme SSIM-vægtning som før. At dreje giver empirisk intet, derfor fra.

T19 **edgeLossWeight** **DETALJER**

Default: 0.0 (= deaktiveret) **Range:** 0 eller 0.001 – 1.0 **Defined in:**

 **TEKNISK**

V437-eksperimental-loss: vægt af en Sobel-gradient-domæne-L1-loss, der sammenligner billedkanter direkte (ground-truth-Sobel vs render-Sobel) ud over L1+SSIM. Hypotese: kant-information er en perceptuel hjørneste af billedkvalitet, og en eksplisit term burde opmuntre gaussians til bedre at træffe kanterne. Test-resultater: vægt 0.1 → 11 % dårligere loss, 0.01 → kvalitets-neutral, men 10 % langsommere. Sobel-passet koster et yderligere MPS-forward på ground-truth og render. Derfor permanent deaktiveret. Fremtidigt use-case: scener med hårde kunstige kanter (arkitektur, møbler, renderinger) kunne profitere — Q7-scene-class-presets pickede det dog ikke, men skalerede i stedet SSIM-vægten.

 **KORT FORTALT**

Eksperimentel tilføjelse, der gør kanter ekstra vigtige. Giver empirisk intet. Forbliver fra.

T20 skyMaskingEnabled DETALJER

Default: false (initializer og alle presets) **Range:** boolean **Defined in:**

 TEKNISK

Slår sky masking til. Derved maskeres sky-regionen i hvert billede via Apple Vision Framework (VNGenerateForegroundInstanceMaskRequest), og loss'en i dette område sættes til nul. Mening: outdoor-scener lider ofte under, at blå/grå/hvide sky-pixler får appen til at placere gaussians præcis dér — hvilket opfattes som „floater“. Uden sky-mask vil loss'en i dette område aldrig blive nul, fordi himlen i billedet varierer let, og appen evigt forsøger at genskabe det med splats. Vision-masken beregnes en gang pr. kamera før træningen og holdes i RAM. Aktiveres typisk sammen med T45 skyDomeEnabled (UI-logik i settings-viewet). Ved indendørs-scener eller syntetiske renderinger lader man stå fra — masken ville dér fejlagtigt genkende lofter eller vægge som „sky“.

 KORT FORTALT

Slår en specialtilstand til for udendørs-optagelser: himlen ignoreres under træningen, så der ikke forsøges genskabt med splats. Anbefales for hver udendørs scene. Lad være ved indendørs eller ved 3D-renderinger fra Blender.

SH-degree-progression (T21)

T21 shDegreeUpgraderIterations

DE TALJER

Default: [1_000, 2_000, 3_000] (initializer), [2_000, 5_000, 8_000] (.full , MCMC), [1_000, 2_000] (.preview — degree 3 sprunget over) **Range:** [Int] , hver værdi i [0, maxIterations] , monotont stigende **Defined in:**

TEKNISK

Iterationer, hvor den aktive SH-degree skiftes op fra 0→1, 1→2, 2→3. Før første markering er kun DC-komponenterne aktive (altså T5 shDCLearningRate), efter første markering DC + 3 degree-1-koefficienter, efter anden markering + 5 degree-2-koefficienter, efter tredje markering alle 15 koefficienter. Lager-behovet pr. gaussian vokser i trin — 4 floats → 16 floats → 36 floats → 64 floats. Quality-presets udskyder optrapningerne i forhold til initializer-defaults (V228), fordi geometrien skal stabilisere sig først, før farve- detaljerne med deres højere frekvens kommer på. V384 testede [1K, 2K, 3K] for .full → 9.3 % dårligere — bekræfter forsinkelsen. .preview kapper ved degree 2, fordi degree 3 ikke konvergerer i 5 000 iterationer og kun bruger optimer-kapacitet. Q6 (T80–T81) tilbyder en alternativ curriculum-logik, der dynamisk overskriver denne liste.

KORT FORTALT

På hvilke punkter i træningen appen lærer, at farver kan se forskellige ud fra forskellige synsvinkler (højlys, refleksioner). Først sent — så formen først stemmer, så farven. Værdierne i presets er sat sådan, at det fungerer godt. Lad være med at ændre noget, medmindre du ved nøjagtigt hvorfor.

Performance (T22–T25)

T22 trainingRenderScale

DETALJER

Default: 1.0 (initializer, `.full`, MCMC, scene-class), 0.5 (`.preview`), 0.25 (`.quickTest`) **Range:** 0.05 – 2.0 (typisk 0.25, 0.5, 1.0) **Defined in:**

TEKNISK

Render-opløsning ved træning relativt til trænings-billedernes oprindelige opløsning. Ved 0.5 nedskales hvert billede til 50 % bredde × 50 % højde (altså 25 % af pixlerne), og gaussian-renderingen sker i denne mindre opløsning. Reducerer både lager- og regneindsats kvadratisk. Vigtigt: `T11 densifyGradThreshold` skal passe til den valgte opløsning — gradient-magnituderne skalerer med $1/\text{opløsning}^2$, derfor har `.quickTest` (0.25×) en meget højere tærskel ($4e-6$) end `.full` (1.0×, $1.1e-6$). RadianceKit advarer ved meget store billeder og tilpasser automatisk — 3-MP-mål-opløsning. Ved ekstreme 4K-input-billeder ville 0.5 eller endda 0.25 være meningsfuldt, ellers kører selv Mac'en kun i CPU-compaction.

KORT FORTALT

Hvor store billederne er ved træningen. 1.0 = original, 0.5 = halv størrelse. Halv størrelse = fire gange hurtigere, men de fineste detaljer mangler. Presets vælger den rigtige værdi; ved ekstremt store input-billeder (over 12 megapixel) skifter appen automatisk ned.

T23 resolutionWarmupScale

DETALJER

Default: 0.0 (= deaktiveret) **Range:** 0 eller 0.1 – **Defined in:**

TEKNISK

V133-optimering: træn densification-fasen (iter 0 til `T2`) i en lavere opløsning end refinement-fasen. V308 har deaktiveret den igen for `.full`, fordi tids-gevinsten ved `T22 = 1.0` og cosine-annealing var marginal, og kvaliteten led minimalt. Forbliver i feltkataloget, fordi den kunne blive meningsfuld igen ved 4K-input og lange trænings-kørsler — Q6 curriculum (T80) har taget en lignende logik op, men dér er den koblet til LR-schedulen. Hvis aktiveret og `T80 curriculumResolutionRamp` ligeledes true, vinder Q6 og overskriver denne værdi.

KORT FORTALT

Spesial-feature: i første træningshalvdel læres med mindre billeder, i den anden med store. Sparer tid. Fra, fordi den nyere Q6-variant løser det bedre.

T24 tileSize DETALJER**Default:** 16 **Range:** 8, 16, 32 **Defined in:** TEKNISK

Størrelse af rasteriserings-tiles i pixel. Gaussian-splattung-renderingen er tile-baseret: billedet opdeles i 16×16-pixel-fliser, hver flise samler de for den relevante gaussians, sorterer dem efter dybde og blander dem ind. 16 er standard, som praktisk talt alle 3DGS-implementeringer bruger, og er hardkodet i RadianceKits Metal-kerneler; en ændring af denne værdi ville kræve re-kompilering af shaderne og er ikke effektiv i den aktuelle stand. Forbliver som felt, hvis en fremtidig engine-version understøtter tile-size dynamisk.

 KORT FORTALT

Intern render-parameter. Standard 16, lad være at ændre.

T25 throttleDelayMs DETALJER**Default:** 0 (initializer, `.full`, MCMC, scene-class), 0 (`.preview`) **Range:** 0 – 100 **Defined in:** TEKNISK

Kunstig forsinkelse mellem trænings-iterationer i millisekunder. 0 = fuld hastighed (standard). Højere værdier gør Mac'en mere „brugelig“ under træningen ved, at GPU/CPU regelmæssigt får pusterum — andre apps' brugervenlighed stiger, men træningstiden også lineært med forsinkelsen. Typiske værdier: 1–2 ms („let“ throttling, +5 % træningstid, Mac føles mere responsive), 5 ms („mellem“, +15 % træningstid), 10+ ms („Eco“, potentielt fordoblet træningstid). Tilbydes i Inspektoren under „Performance“, men er ikke i standard-visningen — se backlog `dev_ux-backlog.md`, der foreslår at fjerne den fra Expert View, fordi den misforstået forlænger træningstiden dramatisk.

 KORT FORTALT

Hvor mange millisekunder pause appen tager mellem trænings-skridt. 0 = ingen pause, hurtigst muligt. Højere værdier gør Mac'en bedre brugelig under træningen — men træningen varer så også længere. På en M3 Ultra eller Mac Studio kan du lade den stå på 0; på en MacBook Air ville 2 eller 5 være en god værdi.

Diagnose og punktsky-forberedelse (T26–T30)

T26 depthDistortionWeight

DETALJER

Default: 0.0 (= deaktiveret) **Range:** 0 eller 0.0001 – 0.05 **Defined in:**

TEKNISK

V366-eksperimental: vægt af en depth-distortion-regulariserings-loss. Straffer gaussians, der langs en render-stråle ganske vist er dybt stablede, men konceptuelt hører til samme overflade — det opmuntrer koncentrerede dybdefordelinger og reducerer floaters. Tests: 0.01 → 4.5 % dårligere, 0.001 → 8.1 % dårligere. Den teoretiske fordel — multi-view-konsistens forbedres — afspejler sig ikke i L1-loss'en, fordi hypotesen implicit antager, at SfM-geometrien er korrekt, og gaussians kun skal „stables“. I praksis er SfM-punktskyen som regel den svageste komponent, ikke stablingen. Forbliver tilgængelig for multi-view-datasæt med særligt rene poser (synthetic, Mip-NeRF 360 med ground truth).

KORT FORTALT

Eksperimentel feature til at undgå flere splats bagved hinanden på samme sted. Ikke aktiveret, fordi testene ikke har givet noget.

T27 singleViewOverfit

DETALJER

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

TEKNISK

Diagnose-flag: hvis true, bruges i hver træningsiteration tvunget kamera-index 0 i stedet for en tilfældig fra camera-pool'en. Mening: hvis modellen ikke engang kan overfitte en enkelt view (altså loss'en på view 0 ikke går mod nul selv efter 10 000 iterationer), er der en fundamental bug i forward/backward-passet. Denne kontakt blev brugt intensivt under udviklingen af Metal-shaderne og differentiable-rasterizer-kernels — V42–V47-fasen. I dag kun tilgængelig som sanity-check, hvis nogen har modificeret backend-kode og vil lave en regression test. Via CLI med `--single-view`.

KORT FORTALT

Test-tilstand for udviklere. De kan dermed tjekke, om appen overhovedet kan lære fra ÉT billede. Irrelevant for almindelige brugere, lad altid være fra.

T28 maxCameras **DETALJER**

Default: 0 (= „brug alle kameraer“) **Range:** 0 eller 1 – N **Defined in:**

 **TEKNISK**

Diagnose-grænse fra V43: træn kun med de første N kameraer, ignorer alle yderligere. Mening oprindeligt: teste hypotese, at for mange kameraer skaber gradient-konflikter (for mange modstridende loss-signaler for samme gaussian). Test-resultat: ingen systematisk fordel ved kunstig begrænsning — flere frames bringer faktisk talt altid mere kvalitet. Forbliver som CLI-flag (`--max-cameras N`) til målrettede eksperimenter, f.eks. „fungerer træningen på de første 100 billeder af en 1 500-billed-droneflyvning?“ Ikke eksponeret i UI.

 **KORT FORTALT**

Diagnose-felt for udviklere — brug kun de første N billeder, ignorer resten. Almindelig bruger har ikke brug for det, værdi 0 = alle billeder. Flere billeder = bedre resultat (se `feedback_more-frames-better.md`).

T29 maxInitialPoints **DETALJER**

Default: 0 (= „brug alle SfM-punkter“) **Range:** 0 eller 1 000 – 200 000+ **Defined in:**

 **TEKNISK**

V54-sikring: begrænser antallet af initiale SfM-punkter, som træningen starter med. Tætte COL-MAP-rekonstruktioner kan producere > 60 000 punkter, hvilket ved store initial-skalaer fører til 200–300 gaussians pr. pixel-overlap — det giver et „tågefelt“, hvor træningen ikke konvergerer. Sub-sampling til ~16 000 punkter (hard-cap-logik i trænings-engine) bringer initial-densiteten på niveau med, hvad reference-3DGS bruger, og reducerer overlap dramatisk. Sættes automatisk ved meget tætte SfM'er; via CLI med `--max-points N`.

 **KORT FORTALT**

Hvor mange start-punkter fra kamera-rekonstruktionen der bruges. Ved meget tætte rekonstruktioner (mere end 60 000) begrænser appen automatisk til 16 000 — ellers er der for meget tåge i starten. Du behøver ikke sætte det; appen klarer det.

T30 cameraClusterOutlierMultiplier DETALJER

Default: 10.0 (alle presets — aldrig overskrevet)

Range: 1.0 – 100.0 **Defined in:**

 TEKNISK

Multiplikator for camera-cluster-outlier-filteret, indført i fase 3.10 A.1. Før træningen beregner trænings-engine centroidet for alle kamera-positioner og den maksimale distance for et kamera fra centroidet. SfM-punkter, hvis distance fra centroidet overskrider $\text{multiplier} \times \text{maxCameraDistance}$, kasseres som outliers. Default 10× bevarer adfærden fra før fase 3.10. En subtil bug: tighter SfM (kameraer tættere sammen) → mindre → mindre tærskel → flere punkter kasseres som outliers. Looser SfM → større tærskel → færre punkter kasseres. Dette er en af årsagerne til fase-3.9-funnel-vs-training-anti-korrelationen: bedre SfM kan downstream føre til dårligere træning, fordi for mange initial-punkter dræbes. Feltet ligger som CLI-override (`--camera-cluster-outlier-multiplier`) til A.3-sweeps; ikke eksponeret i UI. Værdier under 5 er som regel for restriktive, over 20 virkningsløse.

 KORT FORTALT

Specielt filter, der kasserer punkter fra rekonstruktionen, som ligger langt væk fra kamera-skyen. 10 = appen er generøs, beholder næsten alt. Forhøjelse kan være meningsfuld, hvis langt fjertliggende punkter (bjerge i det fjerne) i billedet ser ud som svævende klumper. At sætte lavere kun i nødtilfælde — du mister dybde-detajle.

Regularisering (T31–T37)

T31 coarseToFineBlurRadius

DE TALJER

Default: 0 (= deaktiveret) **Range:** 0 eller 1 – 10 **De-
fined in:**

TEKNISK

V369-eksperimental: box-blur-radius, der i starten af densification-fasen anvendes på ground-truth-billedet og lineært reduceres til slutningen af densification (T2) til 0. Hypotese: coarse-to-fine-træning — først lære grove strukturer, så detaljer — skulle give stabilere geometri. Tests: $r=3 \rightarrow 9.6\%$ dårligere, $r=1 \rightarrow 5.1\%$ dårligere. Grunden til mislykket: densification beslutter baseret på billed-domæne-gradienter, og blur reducerer netop de signaler, der er vigtige for „her skal klones“. Forbliver i feltkataloget til fremtidige tests med andet density-control-skema.

KORT FORTALT

Eksperimentel „først-grov-så-detajleret“-tilstand. Har intet givet, forbliver fra.

T32 scaleRegWeight

DE TALJER

Default: 0.0 (= deaktiveret) **Range:** 0 eller 0.0001 – 0.05 **De-
fined in:**

TEKNISK

V370-eksperimental: L1-regularisering på verdensrumlig skala. Straffer gaussians, der bliver for store — forhindrer „mega-splats“, der dækker hele væg-overflader med en gaussian. Tests: $0.01 \rightarrow 200\%$ dårligere loss (2 M gaussians, total eksplosion), $0.001 \rightarrow 214\%$ dårligere. Grunden: skala-regularisering kommer i konflikt med density-control — mindre skalaer betyder, at flere gaussians bruges, så density-control splitter oftere, hvilket igen betyder mere gradient-arbejde. Disabled, men dokumenteret til Mip-splatting-eksperimenter (T74): i denne kontekst kunne en skala-undergrænse være meningsfuld.

KORT FORTALT

Regularisering, der tvinger splats til at forblive små. Har udløst splat-eksplosioner i tests (millioner af splats). Ikke aktivere.

T33 anisotropyRegWeight DETALJER

Default: 0.0 (= deaktiveret) **Range:** 0 eller 0.0001 – 0.05 **Defined in:**

 TEKNISK

V445-eksperimentel: penalty på $\max(\text{scale})/\min(\text{scale})$ -forholdet, skal forhindre ekstremt langstrakte „needle“-gaussians, der opfattes som floaters. Tests: 0.01 → 69 % dårligere, 0.001 → 15 % dårligere. Grunden: regulariseringen tvinger splats mod „runde“ former, hvilket på en flad overflade (væg, bord, gulv) er netop forkert — dér er en flad, bred gaussian mere effektiv end en kugleformet. Disabled. V549f tilbød med T34 `scaleRatioPruneThreshold` en alternativ, mere målrettet tilgang, som ligeledes blev reverteret.

 KORT FORTALT

Regularisering, der straffer for lange tynde splats. Lyder fornuftigt, var dog dårligere i tests. Fra.

T34 scaleRatioPruneThreshold DETALJER

Default: 0.0 (= deaktiveret) **Range:** 0 eller 5.0 – 100.0 (typisk 10.0 – 30.0) **Defined in:**

 TEKNISK

Eksperimentel post-training-pruning, der sletter hver gaussian, hvis $\max(\text{scale})/\min(\text{scale})$ -forhold overskrider den her satte lineære tærskel. Sigter på ekstremt langstrakte „needle/disc“-floaters, der ikke kan elimineres ved regularisering alene. I testen fjernede pruning floaters som håbet, men samtidig også fornuftige flade splats på vægge og gulve — billedet blev hullet. Derfor fra som default, CLI-flaget (`--scale-ratio-prune N`) forbliver tilgængeligt til målrettede eksperimenter. Anbefalede værdier, hvis man alligevel vil teste: 30 (meget konservativ, fjerner kun ekstreme outliers), 10 (aggressiv, koster detalje).

 KORT FORTALT

Forsøg på at filtrere helt langstrakte splats fra efter træningen. Var netto-negativ i tests — floaters væk, men også detalje væk. Fra.

T35 **opacityRegWeight** **DETALJER**

Default: 0.0 (= deaktiveret) **Range:** 0 eller 0.0001 – 0.05 **Defined in:**

 **TEKNISK**

V446-eksperimental: binary-cross-entropy-penalty, der trækker opacity mod 0 eller 1 (altså væk fra „halv-transparent“). Hypotese: skarpere opacitetsfordeling ville forbedre billedklarhed. Test med T33 kombineret → regularisering koster kvalitet, begge deaktiveret. Disabled. Advarsel: i 1.4.3-beta dukkede en bug op, der havde præcis dette felt med en default-værdi-ændring (initializer = 0.01), hvilket førte til mass-extinction af gaussian-count (460 K → 5 på en iteration). Siden 1.4.4 fast på 0.0 som default.

 **KORT FORTALT**

Regularisering, der gør splats enten helt transparente eller helt solide. Giver intet, kan endda blive farligt (1.4.3-bug mass-extinction). Lad stå på 0.

T36 **opacityDecayFactor** **DETALJER**

Default: 0.0 (initializer = deaktiveret), 0.9995 (`.full`, `.classicBalanced` — HTGS-standard) **Range:** 0 (off) eller 0.95 – 1.0 **Defined in:**

 **TEKNISK**

V546-implementering af HTGS-skemaet (Hierarchical Time-Gating, Eurographics 2025): hver `T37` `opacityDecayInterval` iterationer multipliceres sigmoid-opaciteten af hver gaussian med denne faktor. 0.9995×100 anvendelser giver ~95 %-tilbage pr. densification-fase — et let, men stabilt nedadgående tryk på alle opacities, der pålideligt lader svagt bidragende gaussians synke mod `T14` `pruneOpacityThreshold`. Resultat: 14 % bedre L1-loss på Horse Full (3-trial-avg V546) i forhold til V438 uden decay. Kun aktiv under densification-fasen (indtil `T2`), derefter kører træningen videre uden decay, så de i refinement etablerede opaciteter forbliver stabile. Bruges ikke ved MCMC (MCMC har egne mekanismer via `T67` `mcmcRelocationInterval` 1. `T68` `mcmcDeadOpacityThreshold`).

 **KORT FORTALT**

„Blid falmen“ af alle splats over træningstiden. Gør inaktive splats hurtigere transparente, så de fjernes ved opryddning. Var den vigtigste quality-håndtag i V546-opdateringen: 14 % bedre. Indbygget i Quality-preset. Anbefales ikke at dreje på selv, fordi præcist afbalanceret.

T37 opacityDecayInterval DETALJER**Default:** 50 **Range:** 10 – 500 **Defined in:** TEKNISK

Iterations-interval, hvor T36 opacityDecayFactor anvendes. HTGS-paper-default 50, beholdt i `.full`. Lange intervaller (>200) ophæver effekten delvist, fordi der mellem to anvendelser sker nok gradient-updates til, at opacity stiger igen. Korte intervaller (<20) gør decay for aggressiv. Kun aktiv i densification-fasen.

 KORT FORTALT

Hvor ofte „falmen“ anvendes. 50 = hver 50 iterationer et lille falme-skridt. Passer.

Refinement (T38–T44)**T38** gradientAccumulationSteps DETALJER**Default:** 1 (= „en view pr. Adam-skridt“) **Range:** 1 – 8 **Defined in:** TEKNISK

V424-feature: antal views, hvis gradienter akkumuleres, før et Adam-update udføres. Ved `> 1` kører appen på en separat, „unfused“ backward-projectsti, der summerer gradienterne i en separat buffer; den endelige anvendelse skalerer med $1/N$ for at holde magnituden konstant. V424 testede 2-view → kvalitets-neutral, men 10 % langsommere (fordi unfused-stien er dyrere end fused-stien). Reverteret for `.full`, men bevidst brugt til MCMC — `.fullMCMC` kører med, men V544a-tests viste, at quality-gappet til Classic skrumper til 5 % (i stedet for 11 %). I initializer-default 1, i aktuelle preset 1, forbliver CLI-flag (`--accum-steps N`).

 KORT FORTALT

Hvor mange billeder appen ser, før den tilpasser splats. 1 = hvert billede enkeltvis. Højere = se flere billeder samtidigt og anvend så en middelværdi. Giver intet i standardtilfældet; ved MCMC kan 2 hjælpe lidt.

T39 testViewIndices DETALJER

Default: `[]` (= tom, alle views bruges til træning)

Range: `Set<Int>`, vilkårlig delmængde af camera-indices **Defined in:**

 TEKNISK

V546-feature: sæt af camera-indices, som IKKE bruges til træning, men gemmes som holdout til PSNR/SSIM/LPIPS- evaluering. Sættes automatisk, når `--benchmark` -CLI-flaget er aktivt: så hver ottende view fra index 0 (LLFF-standard, identisk med Mip-NeRF-360- og 3DGS-paper-konventioner). Uden benchmark tom — træningen bruger alle views. **Forsigtighed:** manuel indstilling af dette felt uden forståelse for indekserne kan gøre benchmarket ubrugeligt (f.eks. hvis alle indices over N sættes, mens der kun er N-50 views → ingen holdouts → ingen evaluering). Ved egen preset-eksport persisteres testViewIndices ikke, fordi det er scene-afhængigt og ellers ville efterlade meningsløse værdier mellem forskellige datasæt.

 KORT FORTALT

Hvilke billeder der „udelades“ ved træningen for senere at bruges til kvalitetsmåling. Du sætter ikke det selv; `--benchmark` -flaget gør det automatisk (hvert ottende billede er test). Hvis du sætter egne indekser: farligt, kan forfalske benchmarket.

T40 refinementPruneInterval DETALJER

Default: 0 (= deaktiveret) **Range:** 0 eller 100 – 5 000 **Defined in:**

 TEKNISK

V425-feature: hver N iterationer i refinement-fasen (efter `T2`) køres et yderligere prune-pas, der fjerner gaussians med `sigmoid(opacity) < T41 refinementPruneOpacityThreshold`. **Mening:** under densification er der regelmæssige density-control-kald, derefter ikke længere — gaussians, hvis opacity fortsætter med at synke, forbliver dog i bufferen. V425 testede og reverterede: den ekstra pruning korrelerede med V426 (two-phase densification, ligeledes afsluttet i 0-gaussians-cascade failure). Disabled. CLI-flag tilgængelig til eksperimenter; hvis aktiveret, er 1 000 eller 2 000 fornuftige værdier.

 KORT FORTALT

Yderligere opryddning under refinement-fasen. Giver intet, forbliver fra.

T41 refinementPruneOpacityThreshold **DETALJER**

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

 **TEKNISK**

V425b: separat opacity-tærskel til refinement- pruning. Efter densification har de fleste gaussians nået en markant højere opacity (> 0.001), så standard-T14 `pruneOpacityThreshold` ville være for slap. Hvis T40 aktiv, bestemmer dette felt den egne tærskel. Ved 0.0 anvendes T14 fortsat. Kun relevant, hvis T40 > 0.

 **KORT FORTALT**

Tærskel for den yderligere refinement- opryddning (se T40). Begge felter ikke aktive, altså irrelevant.

T42 midTrainingCompactificationIterations **DETALJER**

Default: [] (= deaktiveret) **Range:** [Int], værdier i (`densifyUntilIteration`, `maxIterations`) **Defined in:**

 **TEKNISK**

V549-feature: eksplicitte iterations-punkter under refinement-fasen, hvor et compactification-pas køres (fjerner sigmoid(opacity) < 0.01 + outlier-skala-gaussians, samme logik som T56 `postTrainingCompactification`). Mening: lange refinement-faser kan vise confetti-/floater-akkumulation, hvis SH så overfitter på view-specifikke artefakter. Typisk konfiguration hvis aktiveret: [10000, 20000, 30000] for 40K Classic. **MEN:** V549-A/B-tests på Family-dataset viste i alle konfigurationer dårligere L1: [10K, 20K, 30K]@0.01 → -48 % count, men +36 % L1; [20K, 30K]@0.005 → -44 % count, men +45 % L1; [20K, 30K]@0.001 → -17 % count, men +87 % L1. Derfor disabled. CLI-flag `--mid-compact "10000, 20000"` tilgængelig, hvis man foretrækker den visuelle floater-tradeoff (mindre confetti i viewporten) over loss-regressionen.

 **KORT FORTALT**

Imellem-opryddninger under træningen. I tests gjorde opryddningen slutresultatet dårligere (ganske vist færre floaters, men også mindre detalje). Fra, kan tændes via CLI, hvis floaters generer dig mere end et lidt mere plumret billede.

T43 frustumCullEnabled **DETALJER****Default:** false **Range:** boolean **Defined in:** **TEKNISK**

V549b-feature: efter træning fjernes alle gaussians, der ligger uden for foreningen af alle træningskamera-frusta. Sådanne gaussians blev aldrig begrænset af loss-signalet og er altid floaters. Særligt effektivt for scener, hvor novel-viewet ligger bag eller ved siden af kamerastien (f.eks. bagsiden af en lineær dronflyvning) — floaters dér bliver aldrig synlige i træningsfasen, men meget vel ved senere bevægelse i 3D-vieweren. V549b A/B på dronflyvninger positive resultater, derfor tilgængelig som opt-in. Default false, fordi ved object-captures med fuld orbit-coverage omfatter frustumforeningen hele scenen, og featuren fjerner intet — tilbydes i settings under „Floater Reduction“ og også i Q9 Outdoor-preset implicit testet over T44 `frustumCullExpansion` (Q7-BayesOpt aktiverede det dog ikke, fordi outdoor-sky-dome løser samme problem bedre).

 **KORT FORTALT**

Specialfilter til dronflyvninger eller lineære optagelser: efter træningen slettes splats, der ikke blev „set“ i noget kamera. Valgfri tænd i settings. Ved enkle objekt-optagelser unødvendigt.

T44 frustumCullExpansion **DETALJER****Default:** 1.1 **Range:** 1.0 – 2.0 **Defined in:** **TEKNISK**

NDC-margin for T43 `frustumCullEnabled`. 1.0 ville skære præcis ved billedkanten, hvilket ville beskære vakkende splats ved billedkanten for meget. 1.1 = 10 % padding ud over den præcise kamera-framing — giver lidt tolerance for randpixler, der i en let forskudt novel-view alligevel kunne blive synlige. Værdier > 1.2 gør cullen praktisk talt virkningsløs, fordi det udvidede frustum omfatter meget mere rum.

 **KORT FORTALT**

Hvor strengt det ovenfor beskrevne filter beskærer. 1.1 = lidt sikkerhedsafstand til billedkanten. Lad værdien stå.

Sky-dome (T45–T48)

T45 skyDomeEnabled

DE TALJER

Default: false (initializer + alle presets undtagen P9 Outdoor) **Range:** boolean **Defined in:**

TEKNISK

V549e-feature: før trænings-start genereres en kugleformet punktsky (Fibonacci-sphere med [T46](#) sample-points), placeret i en radius af $T47 \text{ skyDomeRadiusMultiplier} \times \text{scene_extent}$ omkring scenens midtpunkt og initialiseret med farverne fra de sky-maskerede pixler i alle træningskameraer (se [T20](#) skyMaskingEnabled). Disse sky-dome-gaussians indsættes i begyndelsen af gaussian-bufferen og „fryses“ under træningen (position/skala/rotation-gradient = 0, kun SH og opacity forbliver optimerbare). Effekt: i stedet for sorte „confetti“-områder i det fjerne ser brugeren en rigtig himmel i novel-views. V549e-MVP fungerer meget godt på drone- og landskabsscener; i P9 Outdoor-preset default-on. Ved indendørs scener lad stå fra — sfæren ville hænge meningsløst uden for rummet.

KORT FORTALT

Slår en kunstig „himmelkuppel“ til omkring scenen. Gør uden-dørs-optagelser meget pænere: i stedet for sorte klumper ved billedkanten viser appen den rigtige himmel. Pligt for droneflyvninger og landskaber, meningsløs for indendørs rum.

T46 skyDomeSampleCount

DE TALJER

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

TEKNISK

Antal Fibonacci-sphere-sample-punkter på sky-dome- sfæren. Højere værdier → tættere sky-dome (bedre ved store opløsninger og meget synlig himmel), men mere lagerbehov. 5 000 er sweet spot for 4K-renderinger; ved lavere opløsninger rækker 2 000–3 000. Punkterne initialiseres efter cosine-distance til hver træningskamera-view-vektor med de tilsvarende sky-maskerede pixler — sample-points, hvis view-cone ikke ser noget kamera, forbliver med lav opacitets-startværdi bagved, men ændres ikke i træningen (frosset).

KORT FORTALT

Hvor tæt den kunstige himmel er. 5 000 punkter rækker normalt. Mere = bedre overgang i det fjerne, men koster lidt lager.

T47 skyDomeRadiusMultiplier **DETALJER**

Default: 30.0 (initializer + de fleste presets), 59.0 (P9 Outdoor, Q7-BayesOpt-optimum) **Range:** 5.0 – 200.0 **Defined in:**

 **TEKNISK**

Radius af sky-dome-sfæren relativt til scenens udstrækning (= middel-distance mellem kamera-positionerne). 30 = kuglen har 30 gange diameteren af kamera-skyen. For lille (< 5) → sky-dome interfererer med selve scenen (f.eks. lander et sky-dome-splat i forgrunden); for stor (> 100) → float32-precisionstab på sky-dome-positionerne, hvilket udløser render-glitches i det fjerne. Q7-BayesOpt på Bicycle (Mip-NeRF 360) fandt 59.0 som scene-specifikt optimum for outdoor — det tyder på, at standard-30.0 er for lille til dybe landskaber, og sky-dome-pixlerne i billedkant-områder renderer synligt som „væg“.

 **KORT FORTALT**

Hvor langt væk den kunstige himmelkuppel skal være. 30 = ret langt. Ved store landskaber er 50–60 bedre (Outdoor-preset gør det automatisk). For lille ville være, som om man har klumper direkte foran linsen.

T48 frozenGaussianCount **DETALJER**

Default: 0 (= ingen frosne gaussians) **Range:** 0 eller 1 – T46 **Defined in:**

 **TEKNISK**

Antal gaussians i begyndelsen af bufferen, hvis position/skala/rotation-gradienten sættes til nul i optimeren — de forbliver rumligt stive over hele træningen. Density-control må ikke klone, splitte eller prune dem. Bruges til sky-dome-injection (se T45): hvis sky-dome er tændt, sættes dette felt automatisk til T46 skyDomeSampleCount. Manuel indstilling er mulig (f.eks. for at fryse en forhånds-placeret punktsky fra en LiDAR-scanning), men ikke direkte tilgængelig i UI. Vigtigt: de første N gaussians i bufferen er altid de frosne — rækkefølgen i bufferen afgør, ikke et explicit index.

 **KORT FORTALT**

Hvor mange splats i begyndelsen der er faste og ikke må bevæge sig. Sættes automatisk til sky-dome-antallet, når sky-dome er tændt. Du behøver ikke selv dreje på det.

Adam + LR-schedule (T49–T55)

T49 adamResetIteration

DETALJER

Default: 0 (= deaktiveret) **Range:** 0 eller 100 – **Defined in:**

TEKNISK

V430-feature: iteration, hvor Adam-optimizer-momentum-akkumulatorerne (m1, m2) nulstilles. Bias-korrektion derefter kører med (`iter - adamResetIteration`) i stedet for med `iter`. V430 testede reset ved 5 000 (efter densification- slut) → 12.8 % dårligere loss. Grund: Adam-momentum, der har bygget sig op under densification, bærer information om de typiske gradient-magnituder og accelererer refinement-fasen. At kaste det væk koster de første ~500 iterationer refinement i konvergens. Disabled. Forbliver CLI-flag til forskningseksperimenter.

KORT FORTALT

Reset-knap for det interne Adam-optimizeres „hukommelse“. Har skadet i tests, forbliver fra.

T50 positionLRScheduleEndIteration

DETALJER

Default: 0 (initializer = „brug maxIterations“), 20 000 (`.full` — cosine slutter ved 20K trods `maxIter=35K`), 30 000 (`.fullClassicPaper`) **Range:** 0 eller 1 000 – **Defined in:**

TEKNISK

V431-feature: iteration, hvor cosine-annealing-kurven for position-LR når sit minimum. Hvis 0, er det identisk med `T1 maxIterations`. Hvis > 0, kører schedulen til denne værdi og forbliver derefter konstant ved `T4`

`positionLearningRateFinal`. Det tillader en „extended refinement phase“ med minimal, men konstant læringsrate — forfiner positioner langsomt uden fornyet decay. `.full` gør det (schedule- slut ved 20K, træning kører til 35K), V434c/V434d bekræftede: 15K og 25K begge ca. lige gode, 20K minimalt optimal. Bruges i forbindelse med `T51` videre for også at modificere ikke-position-LR'erne i extended phase.

KORT FORTALT

Hvornår appen holder op med yderligere at sænke position-læringsraten. Hvis lavere end maksimum-iteration, kører derefter med konstant mini-rate — det forfiner meget langsomt, men meget stabilt. Indbygget i Quality-preset, du behøver ikke dreje.

T51 extendedPhaseLRDecay DETALJER

Default: 0.0 (= deaktiveret, konstante LR'er) **Range:** 0 eller 0.01 – 1.0 **Defined in:**

 TEKNISK

V433-feature: minimal multiplikator for ikke-position-LR'erne (skala, rotation, opacity, SH) i „extended phase“ — altså: efter T50 er nået, og position-LR er allerede ved T4. Hvis 0.1, cosine-decays skala/rotation/opacity/SH for deres del fra 1.0 (= deres standard-LR) til 0.1× af deres standard. Hvis 0.0 (default), forbliver de konstante. V457 testede fuldt decay (0.0 = decay-til-nul) mod intet-decay og fandt: avg 0.0400 (2 runs) = samme loss som V438 uden decay. Adfærd renere med decay, men ikke målbart bedre. Derfor disabled. Forbliver i CLI som `--nonpos-lr-scale F`.

 KORT FORTALT

I den sene refinement-fase også gøre farve- og form-læringsrater mindre. Gør træningen „stabilere“, men empirisk ikke bedre. Fra.

T52 adaptiveDensifyThreshold DETALJER

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

 TEKNISK

V440-eksperimental: hvis true, beregner appen i hvert densification-skridt p98 af den aktuelle gradient-fordeling og bruger det som dynamisk tærskel (klamptet til mindst 0.5× af den konfigurerede værdi fra T11, så det ikke skejser for langt ud). Hypotese: automatisk tilpasning til aktuel scene-fase ville gøre density-control mere robust — f.eks. strengere pruning i starten, slappere senere, eller omvendt. V440 testede og reverterede: katastrofalt fald til 63 K gaussians (mass-pruning, fordi p98 i de første iterationer er ekstremt højt, og så overskrider næsten intet tærsklen). Den faste tærskel er allerede godt kalibreret, dynamisk tilpasning skader mere end den gavner. Q5 (T77) tilbyder en alternativ adaptiv logik via rolling median, der omgår problemet.

 KORT FORTALT

Adaptiv version af densify-tærsklen. I tests katastrofal (splat-tallet styrt dykkede til 63K). Fra. Q5 har en bedre variant af det.

T53 mergeAfterDensification **DETALJER**

Default: false (initializer), true (`.full` , `.classicBalanced` , `.fullClassicPaper`) **Range:** boolean **Defined in:**

 **TEKNISK**

V438-feature: ved slutningen af densification-fasen (iter `T2`) udføres et engangs-merge-pass, der sammenfatter nærtliggende gaussians med lignende skala og farve. Reducerer gaussian-antallet med typisk 5–15 % uden synligt kvalitetstab. Mening: efter intensiv kloning opstår klynger af kvasi-identiske gaussians, der ikke bidrager med noget nyt — merging frigør optimerer-kapacitet til andre områder. Standard i Classic-quality-presets. Bruges ikke ved MCMC, fordi MCMC via sin relocation-logik slet ikke lader sådanne klynger opstå.

 **KORT FORTALT**

I slutningen af splat-mangfoldiggørelses-fasen sammenfattes kloner, der er næsten identiske. Reducerer datamængde uden synlig effekt. Som standard tændt i Quality- preset.

T54 densifyPhase2FromIteration **DETALJER**

Default: 0 (= deaktiveret) **Range:** 0 eller `T2` – `T1` **Defined in:**

 **TEKNISK**

V426-eksperimental: muliggør en anden densification-fase, der efter refinement-pausen starter ved denne iteration og kører til `T55` . Hypotese: efter en refinement-fase har gradient-akkumulatorerne stabilere magnitudes og kan mere præcist sige, hvilke områder der stadig har brug for yderligere gaussians. V426 testede og reverterede: two-phase densification faldt i 0-gaussians-cascade-failure (kombineret med V425 refinement-pruning ødelagde det bufferen). Disabled. CLI-flag tilgængelig til eksperimenter.

 **KORT FORTALT**

Anden mangfoldiggørelses-runde efter pause. Har udryddet splat-beholdningen i tests. Fra.

T55 densifyPhase2Untilliteration DETALJER

Default: 0 **Range:** 0 eller T54 – T1 **Defined in:**

 TEKNISK

Slutning af V426-two-phase-densification. Kun relevant hvis `T54 > 0`. Begge felter sammen disabled.

 KORT FORTALT

Slutning af anden mangfoldiggørelses-runde (se T54). Begge fra.

Post-processing + Apple AI (T56–T60)

T56 postTrainingCompactification DETALJER

Default: true (i alle production-presets), false (`.quickTest`, `.preview`) **Range:** boolean **Defined in:**

 TEKNISK

V443-feature: efter trænings-slut fjernes gaussians med `sigmoid(opacity) < 0.01` hårdt (de bidrager praktisk talt ikke længere til billedet). Reducerer gaussian-count med typisk 58 % og eksport-filstørrelse med 55 % uden synligt kvalitetstab. Som standard aktiv i production-presets — slutresultatet skal kunne leveres så kompakt som muligt. I `.quickTest` fra, fordi en diagnose-kørsel alligevel ikke eksporteres. I modsætning til T42 `midTrainingCompactificationIterations` (V549) finder compactificationen først sted ved slutningen — refinement kan indtil da bruge alle gaussians.

 KORT FORTALT

Opryddning efter træningen: næsten usynlige splats fjernes. Gør eksport-filen ca. halvt så stor uden kvalitetstab. Pligt-feature, lad fra kun i diagnose-kørsler.

T57 metalFXUpscaling DETALJER**Default:** false **Range:** boolean **Defined in:** TEKNISK

V444-feature: aktiverer Apples MetalFX spatial upscaler i stedet for bilineær interpolation i 3D-viewer-output. Hvis træningsopløsning < viewport-størrelse (f.eks. træning på 0.5x, viewport-visning i fuld opløsning), kan MetalFX levere et markant skarpere billede. Ændrer sig live i viewporten, ingen genoptræning nødvendig. Udelukker T58 `mpsLanczosScaling` — MetalFX har forrang. Anbefaling: tænd, hvis billedet i vieweren virker „udvasket“ sammenlignet med den forventede detalje.

 KORT FORTALT

Apple-ML-baseret billede-skarpning i 3D-vieweren. Hjælper, hvis du har trænet i lavere opløsning og viser resultatet i fuld skærm. Live-toggle, prøv.

T58 mpsLanczosScaling DETALJER**Default:** false **Range:** boolean **Defined in:** TEKNISK

V444-feature: `MPSImageLanczosScale` til viewport-skalering i stedet for bilineær interpolation. Lanczos er en klassisk sinc-baseret resampling-metode, der leverer markant skarpere resultater end bilineær med minimal overhead. Live- toggle. Overskrives af T57, hvis begge er tændt.

 KORT FORTALT

Klassisk skarpmingsmetode til 3D-vieweren (Lanczos). MetalFX (T57) er ML-baseret og som regel bedre; Lanczos er et mindre aggressivt alternativ.

T59 livePreviewInterval DETALJER

Default: 50 (initializer og de fleste presets) **Range:** 0 (off) eller 10 – 5 000 **Defined in:**

 TEKNISK

Hvor ofte under træningen 3D-vieweren opdateres med de aktuelle gaussians. 50 = hver 50 iterationer et nyt render i vieweren — godt nok til at iagttage fremskridt uden at sinke træningen. 0 = vieweren opdateres slet ikke (baggrunds- træning, max hastighed). Typisk tilpasning: ved `.quickTest` ned til 10 (man vil se hvert skridt), ved lange MCMC-kørsler op til 500–2000 (update-overhead i sum mærkbar).

 KORT FORTALT

Hvor ofte 3D-forhåndsvisningen opdateres under træningen. 50 = hver 50 iterationer. Højere = sjældnere = lidt hurtigere, men du ser sjældnere fremskridt. 0 = ingen forhåndsvisning (for maksimal hastighed).

T60 perceptualLossWeight DETALJER

Default: 0.0 (= deaktiveret) **Range:** 0 eller 0.001 – 0.5 **Defined in:**

 TEKNISK

V444-future-feature: vægt af en perceptuel loss-term via MPSGraph (VGG-lignende lille netværk). Ville fange strukturel og teksturel lighed på et højere semantisk niveau end L1+SSIM — typisk i forsknings-pipelines, hvor „pixel-perfect“ er mindre vigtigt end „ser realistisk ud“. Implementering stadig udestående (kode-stub findes, men forward-pass ikke implementeret). Default 0.0. Forbliver i feltkataloget til fremtidig aktivering; CLI-flag `--percep-weight F` reserveret.

 KORT FORTALT

Planlagt feature, der med AI-hjælp tilstræber „naturligt udseende“ i stedet for „pixel-præcis“. Endnu ikke færdigimplementeret.

MCMC-densification (T61–T73)

T61 densificationStrategy

DETLJER

Default: `.classic` (initializer + Classic-presets),
`.mcmc` (alle MCMC-presets + scene-class) **Range:**
`.classic` eller `.mcmc` **Defined in:**

TEKNISK

Vælger mellem Classic-densification (klon/split/prune, Kerbl et al. 2023) og MCMC-densification (stochastic gradient Langevin dynamics med relocation, Kheradmand et al. NeurIPS 2024). Ved `.classic` evalueres T11–T16, ved `.mcmc` T62–T73. Forsigtighed ved skift: Classic-defaults og MCMC-defaults er totalt anderledes kalibreret — den, der flipper vælgeren i Expert View uden at indlæse en passende preset, risikerer 1.4.3-bugstil mass-extinction (460 K → 5 i en iteration, fordi MCMC-OpacityReg på 0.01 dræber Classic-opacities). Derfor er MCMC-init-defaults bevidst „blødgjorte“ (alle reg-værdier 0.0).

KORT FORTALT

Hvilken algoritme der bruges til at mangfoldiggøre splats. Classic = oprindelig metode (hurtigt, mange splats). MCMC = nyere metode (langsommere, langt færre splats, til gengæld mere kompakt). Presets vælger den rigtige. Selv kun omstille, hvis du også indlæser den passende preset (P5–P7 eller P8–P10).

T62 mcmcMaxGaussians **DETALJER**

Default: 150 000 (initializer + `.fullMCMC` + `.mcmcBalanced`), 100 000 (`.mcmcPreview`), 1 500 000 (`.fullMCMCMip` — Mip-splattung-variant med 10x budget), 1.19 M (`.renderPreset`), 1.25 M (`.outdoorPreset`), 670 K (`.indoorPreset`) **Range:** 0 (= „brug buffer-kapacitet“) eller 10 000 – 5 000 000 **Defined in:**

 **TEKNISK**

Hård overgrænse for antallet af gaussians ved MCMC-strategi. Antallet vokser gradvist med `T70 mcmcGrowthRate` (typisk 5 %) pr. relocation-step op til dette cap. V473/V531 fandt 150 K som sweet spot — over 200 K udvander splat-kvaliteten (for mange små, overflødige gaussians), under 100 K forbliver scenen under-densificeret. Ved meget store scener (f.eks. 1 545-foto-droneflyvning med 158 K SfM-init) er 150 K for lavt — derfor 1.4.5-udvidelsen `T72 mcmcCapMultiplier` + `T73 mcmcAutoScaleByScene`. Q7-BayesOpt fandt scene-specifikke optima mellem 670 K (Indoor) og 1.25 M (Outdoor). Ved værdi 0 bruger engine den fulde buffer-kapacitet som cap.

 **KORT FORTALT**

Maksimum antal splats ved MCMC. 150 000 er standarden og rækker til de fleste scener. Outdoor- og render-presets (P8, P9) går op til 1+ million for mere detalje-rige scener. At sætte op kan bringe detalje, koster lager; ned er snarere en nødbremse.

T63 mcmcNoiseScale **DETALJER**

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

 **TEKNISK**

Multiplikator for det gaussiske støj, der i hver MCMC-iteration adderes til positionen af hver gaussian (SGLD- logik). Højere = mere eksploration (gaussians vandrer mere, finder potentielt bedre pladser), lavere = mere udnyttelse (gaussians forbliver, hvor de allerede er gode). V467 og V536 bekræftede 5e-5 som optimal — 1e-5/2e-5 for lidt eksploration, 1e-4 for meget (splats løber ud). Cosine-decayes over træningstiden til `T69 mcmcNoiseDecayEnd` — ved slutningen af decay-området er støj reelt 0, og gaussians konvergerer.

 **KORT FORTALT**

Hvor meget tilfældigt „wackeln“ appen tillader splats, så de selv finder den bedste plads. Standardværdien er optimalt testet. Hvis du skruer det op, bliver splats urolige.

T64 **mcmcOpacityRegWeight** **DETALJER**

Default: 0.0 (= deaktiveret i RadianceKit defaults, paper: 0.01) **Range:** 0 eller 0.001 – 0.05 **Defined in:**

 **TEKNISK**

MCMC-specifik L1-penalty på opacity. Paper-default 0.01 (trykker ubrugte gaussians mod nul, gør dem tilgængelige for relocation). V464b viste dog: uden reg er det målbart bedre i RadianceKit (session 28 bekræftet). Grund: pruning-kriteriet defineret med T68 `mcmcDeadOpacityThreshold` rækker alene — en yderligere L1-penalty tvinger også værdifulde, lav-opacity- gaussians til at dø. Derfor default 0. **Advarsel:** i 1.4.3-beta-build var initializer-default'en fejlagtigt 0.01, hvilket resulterede i mass-extinction-bug'en (se T61-forklaring); siden 1.4.4 fastsat på 0.0.

 **KORT FORTALT**

MCMC-specialregularisering. Fra, fordi den anden MCMC-mekanisme (tærskel i T68) allerede dækker det. Lad stå på 0.

T65 **mcmcScaleRegWeight** **DETALJER**

Default: 0.0 (= deaktiveret, paper: 0.01) **Range:** 0 eller 0.001 – 0.05 **Defined in:**

 **TEKNISK**

MCMC-specifik L1-penalty på skala-egenværdier. Paper-default 0.01. V464b: uden reg bedre, samme begrundelse som T64. Disabled i alle RadianceKit-MCMC-presets. Advarsel som ved T64: 1.4.3-bug.

 **KORT FORTALT**

Som T64, men for splat-størrelse. Fra.

T66 **mcmcRelocationInterval** **DETALJER**

Default: 100 (initializer + alle MCMC-presets, pa-per-standard), 155 (P9 Outdoor — Q7-BayesOpt-optimum) **Range:** 50 – 500 **Defined in:**

 **TEKNISK**

Iterations-interval, hvor MCMC reloacerer døde gaussians ($\text{sigmoid}(\text{opacity}) < T_{68}$ `mcmcDeadOpacityThreshold`) til nye positioner. V537 testede 50 (for forstyrrende, loss svinger) og 200 (marginalt dårligere, MCMC mister reaktionsevne). 100 er optimal. Q7-BayesOpt på Bicycle fandt 155 som scene- specifikt optimum for outdoor — de let længere intervaller giver Adam mere tid til at integrere nyplacerede gaussians, før næste reloc-event sætter dem under pres.

 **KORT FORTALT**

Hver hvor mange iterationer MCMC flytter de døde splats et andet sted hen. 100 er standard. Du behøver ikke dreje selv — Outdoor-preset har allerede den optimale værdi.

T67 **mcmcWarmupIterations** **DETALJER**

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

 **TEKNISK**

Antal initial-iterationer, hvor ingen MCMC- relocation endnu sker. Først efter denne warmup begynder reloc- logikken. Mening: i de første iterationer er opacity-værdierne endnu ikke afbalancerede — hvis der startedes direkte med reloc, ville gaussians blive placeret de forkerte steder og straks skulle flyttes igen, hvilket ødelægger Adam-momentum. Paper-default 500. RadianceKit overtager denne værdi, fordi V464b viste, at den er robust.

 **KORT FORTALT**

Hvor mange iterationer MCMC først „lander“, før det begynder at omplacere splats. 500 er standard og passer.

T68 **mcmcDeadOpacityThreshold** **DETALJER**

Default: 0.005 (initializer, paper-standard), 0.01 (`.fullMCMC` og alle MCMC-presets — V535-optimum) **Range:** 0.001 – 0.05 **Defined in:**

 **TEKNISK**

sigmoid(opacity)-tærskel, under hvilken en gaussian regnes som „død“ og kommer i betragtning til relocation. V535 fandt 0.01 som optimal (0.005 marginal, 0.02 dårligere). Højere = mere aggressiv reloc (flere gaussians flyttes), lavere = mere forsigtig. 0.01 svarer ca. til „0.5 % visuel synlighed“. P10 Indoor bruger via Q7-BayesOpt 0.0142 som optimum.

 **KORT FORTALT**

Fra hvilken gennemsigtighed et splat regnes som „dødt“, så MCMC flytter det et andet sted hen. 0.01 er optimal i vores tests. Du behøver ikke dreje selv.

T69 **mcmcNoiseDecayEnd** **DETALJER**

Default: 0 (initializer = „ingen decay“), 160 000 (`.fullMCMC` = 80 % af 200K), 96 000 (`.mcmcBalanced` = 80 % af 120K), 40 000 (`.mcmcPreview`) **Range:** 0 eller 1 000 – **Defined in:**

 **TEKNISK**

Iteration, hvor `T63 mcmcNoiseScale` -støj dæmpes fuldstændigt til nul (cosine-decay fra iter 0 til her). V497c/V502 fandt 80 % af `maxIterations` optimal — giver MCMC nok eksplorations-tid, men lader de sidste 20 % gå til konvergens uden støj. 0 = konstant støj over alle iterationer (sjældent meningsfuldt, MCMC kan så ikke konvergere).

 **KORT FORTALT**

Hvornår det tilfældige „wackeln“ af splats holder op. I MCMC-presets ved 80 % af samlede iterationer — først eksploration, så konvergens. Lad værdien stå.

T70 **mcmcGrowthRate** **DETALJER**

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

 **TEKNISK**

Vækstrate af MCMC-populations-targetet pr. relocation-step. Logikken: ved hver reloc-event øges mål-populations-størrelsen med $(1 + \text{growthRate})$, indtil T62 `mcmcMaxGaussians` (eller den per T72/T73 skalerede variant) er nået. V512/V522 fandt 0.05 som optimal — højere værdier fører til for hurtig vækst (gaussians indsættes, før Adam-momentum kan integrere dem), lavere til under-densificerede scener til sidst.

 **KORT FORTALT**

Hvor hurtigt splat-antallet ved MCMC vokser. 5 % pr. skridt er optimal. Lad værdien stå.

T71 **mcmcSigmoidK** **DETALJER**

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

 **TEKNISK**

Sigmoid-sharpness-parameter for MCMC-noise-attenuation. I SGLD-skridtet dæmpes per-gaussian-støjen med — højt-opake gaussians (hvis logit er positiv) får eksponentielt mindre støj end lavt-opake. $K = 100$ er skarp, sprich overgangen fra „fuld-noise“ til „ingen-noise“ sker meget hurtigt omkring opacity 0.5. V484–V487 fandt $K = 100$ optimal — mindre værdier (10–50) lader også højt-opake gaussians wackle med (ødelægger konvergerede gaussians), større (> 500) gør overgangen kunstigt hård, og døde gaussians flyttes slet ikke mere.

 **KORT FORTALT**

Specialparameter, der bestemmer, hvor skarpt MCMC adskiller mellem „gennemsigtig nok til at flytte“ og „solid, rør ikke“. Standardværdien er optimal. Ikke dreje på.

T72 mcmcCapMultiplier DETALJER

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 (= deaktiveret) eller 1.0 – 10.0 **Defined in:**

 TEKNISK

1.4.5-feature: scene-adaptiv cap-skalering. Hvis `T73 mcmcAutoScaleByScene true`, beregnes det effektive cap som (klampet til buffer-kapacitet). Baggrund: ved store scener (f.eks. 1 545-foto-droneflyvning → 158 K SfM-init) er `T62 = 150 000` for lavt — density-control ville slet ikke kunne vokse. Med multiplier 3.0 skales cap'et i dette eksempel til 474 K (158 K × 3.0). Q7-BayesOpt fandt scene-specifikke optima: outdoor profiterer af høj multiplier (5.32 → ~830 K cap ved 156 K bicycle-init), indoor nøjes med 1.76 (vægge mætter hurtigere). Fuld opløsning af cap'et se -metoden.

 KORT FORTALT

Multiplikator, der tilpasser splat-cap'et automatisk til scene-størrelsen. Stor scene = flere startpunkter = højere cap. Standard 3× passer til de fleste scener; Outdoor-preset går op til 5× (store dybde-områder), Indoor til 1.76× (vægge begrænser alligevel).

T73 mcmcAutoScaleByScene DETALJER

Default: true (initializer + alle MCMC-presets) **Range:** boolean **Defined in:**

 TEKNISK

1.4.5-feature: master-switch for scene-aware cap-logikken (se T72 +). Hvis false, bruges udelukkende `T62 mcmcMaxGaussians` som cap (tilbage til 1.4.4-adfærd). Som standard tændt, fordi mass-extinction-problemerne ved store scener fra 1.4.3 ellers kommer tilbage. Manuelt deaktivere kun, hvis du eksplisit vil sætte et hårdt cap — f.eks. for at træne en 150 K-variant, hvis slutstørrelse kan planlægges.

 KORT FORTALT

Slår automatisk tilpasning af splat-cap'et til scene-størrelsen til. Som standard tændt. Lad fra kun, hvis du selv vil have nøjagtigt et bestemt splat-antal.

Mip-splatting (Q1.5) (T74–T76)

Status: Q1.5 blev 2026-05-25 efter 14 autonome iterationer

1. overnight-1.5M-confidence-check kasseret som „closed no-win“

(max $\Delta@2x = +0.27$ dB, original-gate krævede $\geq +1.5$ dB middelværdi over $0.5x/2x$, FAILT på 0/11 pair-scenes). Felterne forbliver **opt-in** til forskningseksperimenter; alle production- presets har. Se verdict: docs/plans/2026-05-25-phase-q1.5-final-verdict.md.

T74 useMipSplatting

DE TALJER

Default: false (alle production-presets), true (`.fullMCMCmip` — forsknings-sibling) **Range:** boolean **Defined in:**

TEKNISK

Aktiverer Mip-splatting (Yu et al. CVPR 2024): 3D-smoothing-filter + 2D-filter + α -kompensation, der begrænser per-gaussian-frekvensen til Nyquist-grænsen af den tætteste træningskamera-samlingsrate. Teoretisk mål: eliminering af aliasing ved rendering i off-training-skalaer ($0.5x$ eller $2x$ af træningsopløsningen). Aktiveret i preprocess- og backward- projection-shaderne, funktionelt korrekt verificeret i Q1.5-D- test. Men: det originale acceptance-gate ($\Delta@1x \geq +0.3$ dB OG $\text{avg}(\Delta@0.5x, \Delta@2x) \geq +1.5$ dB) blev ikke nået på nogen af de 11 pair-scenes. Maksimalt observeret: family 750K classic $\Delta@2x = +0.270$ dB. Outdoor-scener (Truck, Flowers) viste endda forværring $1x$ og $0.5x$. Hypotese: 3D-smoothing konkurrerer med MCMC-relocation ved high-Gs. Feltet forbliver til fremtidig multi-scale-re-eval med korrekt Mip-NeRF-360-metodologi (se O3-backlog i benchmark-stien).

KORT FORTALT

Aliasing-filter fra et 2024-paper. Teoretisk fedt, praktisk har det intet givet i vores tests og nogle gange endda skadet. Forbliver tilgængelig for eksperimentører, men vi anbefaler det ikke. Lad fra.

T75 mipSmoothing3DScale **DETALJER**

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

 **TEKNISK**

3D-smoothing-skala-parameter (Yu et al. §3.3, paper- default 0.2). Større = mere world-space-udjævning pr. gaussian (= mere anti-aliasing, men også mere blur i default-skalaen), mindre = skarpere, men mere modtagelig for aliasing. Konsulteres kun, hvis T74 `useMipSplatting = true`. I Q1.5-tests ikke yderligere optimeret — A/B-gate'en havde allerede tabt med paper-default 0.2, yderligere sweeps ville være nytteløse.

 **KORT FORTALT**

. Hvis du ikke har tændt Mip, irrelevant.

T76 mipFilter2DVariance **DETALJER**

Default: 0.3 (= præcis V242-legacy-adfærd) **Range:** 0.1 – 1.0 **Defined in:**

 **TEKNISK**

2D-Mip-filter-varians, der adderes til Σ_{2D} -diagonalen (varians direkte, ikke kvadreret). 0.3 er præcis V242-legacy-værdien, der før Mip-splatting var hardkodet i kerneler. Hvis T74 `useMipSplatting = false`, ignorerer kernel denne værdi fuldstændigt og skriver det hardkodede 0.3 — så baselinen ikke kan regredere (Codex-round-1-S3-1-garanti). Hvis, bruges den her satte værdi. Forbliver i feltkataloget til Mip-sweeps.

 **KORT FORTALT**

Yderligere Mip-splatting-parameter. Ved Mip-fra: irrelevant.

Adaptive densification (Q5) (T77–T79)

T77 adaptiveDensification

DE TALJER

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

TEKNISK

Q5-feature: rolling-median-tracker som alternativ til faste T11 densifyGradThreshold. Hvis true, overskrives den aktuelle tærskel i hvert densify-step med $\text{median}(\text{seneste } N \text{ avgGrad-samples}) \times T79 \text{ adaptiveDensifyMultiplier}$. $N = T78 \text{ adaptiveWindow}$. Strengere end V440 p98 (den katastrofale 63 K-pruning-fælde), median + 2× sidder omtrent ved p70–p80 af gradient-fordelingen i steady state. Q5-tests: alenestående FAIL 0/3 scener, men sammen med Q6 (se T80/T81) PASS 1/3 scener — bundtet Q5+Q6 blev 2026-05-25 godkendt som opt-in og aktiveres via CLI `--adaptive-densify`. Q6 er her „carrier“ af kvalitetsgevinsten, Q5 bidrager snarere til stabilitet.

KORT FORTALT

Selvlærende densify-tærskel. I stedet for en fast indstillet følsomhed tilpasser appen sig scenen. Alene testet ikke bedre, sammen med curriculum'et fra Q6 dog ja. Tænd begge sammen, eller begge fra.

T78 adaptiveWindow

DE TALJER

Default: 1 000 **Range:** 100 – 10 000 **Defined in:**

TEKNISK

Rolling-median-vindue i densification-events (IKKE iterationer — hvert T13 densifyInterval-step leverer et sample). Default 1 000 — betyder ved, at de seneste 100 000 trænings-iterationer bidrager til medianen, altså typisk hele træningshistorikken indtil her. Tidlig fase (før T78 samples): tracker returnerer nil → fallback på fast tærskel T11. Kun relevant hvis.

KORT FORTALT

Hvor mange gamle densify-skridt der indgår i medianen for T77. Standard 1000 er god. Kun relevant hvis Q5- adaptive tændt.

T79 adaptiveDensifyMultiplier DETALJER**Default:** 2.0 **Range:** 1.0 – 4.0 **Defined in:** TEKNISK

Multiplikator på rolling-median for den adaptive tærskel. Default 2.0 svarer ca. til p70–p80 af den typiske gradient-fordeling. Lavere = mere aggressiv vækst (flere kloner), højere = strengere (færre kloner). Q5-tests i range 1.5–3.0 — 2.0 bedste default. Kun relevant hvis.

 KORT FORTALT

Faktor for T77/T78. Standard 2.0 = strengere end typisk median. Ikke dreje på.

Curriculum (Q6) (T80–T81)**T80** curriculumResolutionRamp DETALJER**Default:** false **Range:** boolean **Defined in:** TEKNISK

Q6-feature: trænings-opløsning starter ved 0.5x og skifter ved $T50 \text{ positionLRScheduleEndIteration} / 2$ (eller $T1 \text{ maxIterations} / 2$, hvis T50 ikke er sat) til $T22 \text{ trainingRenderScale}$. Bruger den i Q1.5.1 udviklede `resize/restoreImageBuffers`-infrastruktur. Overskriver $T23 \text{ resolutionWarmupScale}$, hvis aktiveret. Q6 er godkendt som „carrier af kvalitetsgevinsten“ i Q5+Q6-bundtet (se T77) — den trinvis opløsningsforøgelse giver appen tid til at finde grov geometri på den lavere opløsning, før den går over til det fine detalje-arbejde. Via CLI: `--curriculum-resolution`.

 KORT FORTALT

„Først grov, så fin“ for trænings-opløsningen. Halv opløsning i første halvdel, så fuld opløsning. Hjælper i visse scener, i andre ikke — bedst tændt sammen med T81.

T81 curriculumSHProgression DETALJER**Default:** false **Range:** boolean **Defined in:** TEKNISK

Q6-feature: overskriver

T21 shDegreeUpgradeIterations med [maxIter/4, maxIter/2, maxIter*3/4], fordeler altså SH- op-trapningerne jævnt over træningstiden i stedet for at front- load dem. Hypotese: stabil geometri etableres før color-detail- eksplosion, hvilket positionerer view-direction-afhængige glans- effekter mere præcist. Q5+Q6 sammen PASS 1/3 scener, Q6 som carrier af gevinsten (Q5 alene FAIL). Via CLI:

`--curriculum-sh .` KORT FORTALT

„Først form, så farve“ — glans-effekterne frigives først sent i træningen, så splats først finder deres position og størrelse. Kan tændes sammen med T80; alene bringer det ikke helt så meget.

Statiske presets (TP1–TP9)

Her kun de strukturelle forskelle til initializer-default'en. Den fulde marketing-beskrivelse af de elleve UI-presets P1–P11 finder du i kapitel 7.

TP1 .preview DETALJER

Diagnose-/forhåndsvisnings-preset til systemer \geq 10 GB RAM. Overrides i forhold til initializer: - 30 000 \rightarrow 5 000 - 15 000 \rightarrow 3 500 (70 % af maxIter) - $1.6e-6 \rightarrow 1.6e-5$ (10 \times højere, mindre aggressiv decay) -,,,, hver 2 \times (V176) - 3 000 \rightarrow 100 000 (reelt fra, V172: reset ødelægger korte træninger) - [1K, 2K, 3K] \rightarrow [1K, 2K] (V182: degree 3 konvergerer ikke i 2K iter) - 1.0 \rightarrow 0.5

 KORT FORTALT

enhver vilkårlig initial vurdering af en nyligt importeret billed-serie — 2–3 min ventetid, derefter rækker resultatet til et binært spørgsmål „er Quality-kørsel det værd?“.

TP2 `.full` **DETALJER**

Production-Quality Classic. Overrides: - 30 000 → 35 000 (V550: 40K-tests Truck-overtraining +10.7 % Gs ved -1.3 % L1) - 15 000 → 5 000 (V310 sweet spot, V338 7K worse) - Alle LR'er 2x (V188) - $1.6e-6$ → $1.6e-5$ (V45 10x) - $2e-6$ → $1.1e-6$ (V335) - 100 → 200 (V112) - 0.005 → 0.001 (V393) - 3 000 → 100 000 (V194 disabled, V421 confirmed) - [1K, 2K, 3K] → [2K, 5K, 8K] (V228 delayed) - 0.0 → 0.9995 (V546 HTGS, 14 % forbedring) - 50 (uændret, V546) - false → true (V438) - 0 → 20 000 (V431) - true (V443, allerede initializer-default for `.full`)

 **KORT FORTALT**

enhver standard-foto-optagelse (objekt, lille rum, skulptur) med < 500 billeder. Den i V546 annoncerede 14 % loss- forbedring i forhold til V438 blev 3-trial-gennemsnit bekræftet på Horse Full.

TP3 `.fullClassicPaper` **DETALJER**

Q1.5-A-test-sibling af TP2, paper-tro Classic. Overrides i forhold til TP2: - 35 000 → 30 000 (paper-standard) - 5 000 → 15 000 (paper: 50 % af maxlter) - $1.6e-5$ → $1.6e-6$ (paper-default) -, tilbage til paper defaults (0.05, 0.005, 0.001) - $1.1e-6$ → $2e-7$ (kalibreret for ~1-2M Gs på Bicycle) - 200 → 100 (paper) - 0.001 → 0.005 (paper-default) - 100 000 → 3 000 (paper §5.2, risikabelt — kan udløse V194-regression) - 0.9995 → 0.0 (paper har ingen decay) • 20 000 → 30 000 (cosine kører til 100 % af maxlter)

 **KORT FORTALT**

Q1.5-forskningseksperimenter, der har brug for paper-magnitudo-gaussian-budgetter (1–2 M) til Mip-splatting- tests. Efter Q1.5-„closed no-win“-verdict forbliver presetet tilgængeligt for advanced users, men er ikke production-anbefalet.

TP4 `.fullMCMC` **DETALJER**

Production-Quality MCMC. Overrides i forhold til initializer:

- 30 000 → 200 000 (V534, MCMC har brug for 5x flere iter

end Classic) - 15 000 → 160 000 (V504b 80 % af maxlter) - 1.6e-6 → 1.6e-5 - LR-schedule som TP2 (alle 2x) - 0.2 → 0.05 (V521b/V534: MCMC har brug for stærkere L1-signal) - [1K, 2K, 3K] → [2K, 5K, 8K] - `.classic` → `.mcmc` - 150 000 (allerede i initializer, bekræftet i preset) - 5e-5 (V467/V536 optimal) - 0.005 → 0.01 (V535 optimal) - 0 → 160 000 (80 % af maxlter, V497c/V502) - 3.0 (allerede i initializer) - true (allerede i initializer) - 3 000 → 200 000 (reelt fra, MCMC bruger reloc i stedet for reset)

 **KORT FORTALT**

Web-levering, object-captures med detalje-krav, droneflyvninger (også når P9 Outdoor er endnu bedre). 71% færre gaussians end Classic ved sammenlignelig L1.

TP5 `.fullMCMCMip` **DETALJER**

Q1.5-D-test-sibling af TP4, med Mip-splatting + paper-magnitude-MCMC-budget. Overrides i forhold til TP4:

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

 **KORT FORTALT**

Alle andre felter identiske med TP4. Q1.5 D-PASS på Bicycle 2026-05-24 (bryder 12-iter-multi-scale-FAIL-streak). Q1.5-slutdom 2026-05-25 alligevel closed-no-win — Mip-splatting-gevinst ikke reproducerbar over 11 pair-scenes. Preset forbliver optin.

TP6 `.classicBalanced` **DETALJER**

Mid-tier Classic. Overrides i forhold til TP2: - 35 000 → 20 000 (V149: 20K = 30K ved 33 % mindre tid) - 20 000 → 0 (cosine kører til maxlter = 20K, ingen extended phase)

 **KORT FORTALT**

Standardtilfælde med kortere ventetid. V149 identificeret som sweet spot.

TP7 `.mcmcPreview`**DETALJER**

MCMC-diagnose. Overrides i forhold til 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 = lighter scaling)

KORT FORTALT

hurtigt se et MCMC-resultat for at vurdere, om TP4 eller en scene-class-preset er det værd.

TP8 `.mcmcBalanced`**DETALJER**

Mid-tier MCMC. Overrides i forhold til TP4: - 200 000 → 120 000 (V518) - 160 000 → 96 000 (80 %) - 160 000 → 96 000 (80 %) - 3.0 → 2.5 (mellem Preview 2.0 og Full 3.0)

KORT FORTALT

MCMC uden den fulde 200K-kørsel. ~120 K iterationer er et godt kompromis mellem kvalitet og ventetid.

TP9 `.quickTest`**DETALJER**

Ren funktionstest. Overrides i forhold til initializer: - 30 000 → 1 000 - 15 000 → 500 - 2e-6 → 4e-6 (kalibreret for 0.25× opløsning) - 100 → 50 - 3 000 → 100 000 (fra, da alt for kort) - 1.0 → 0.25

KORT FORTALT

Sanity-check „starter træningen overhovedet meningsfuldt?”. Værrighed < 30 s på M3 Ultra. Ser garanteret plumret ud.

Metoden:

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

TEKNISK Eneste source-of-truth for spørgsmålet „hvor mange gaussians må MCMC maksimalt lade vokse?”. Beregnes ud fra tre input: det konfigurerede `T62 mcmcMaxGaussians` (med `mass-extinction-floor` 150 000, hvis 0), (antal SfM- init-punkter) og (forhåndsallokeret gaussian- buffer-størrelse). Logik:

1. `base = T62 > 0 ? T62: 150_000` (mass-extinction-floor beskytter mod initializer-default-bugs som 1.4.3-mass-extinction- hændelsen)
2. Hvis `T73 mcmcAutoScaleByScene && initialPointCount > 0 && T72 mcmcCapMultiplier > 0:`
 - `scaled = max(base, ceil(initialPointCount × T72))` ellers
3. Hvis `bufferCapacity > 0:` `return min(scaled, bufferCapacity)`
4. Ellers `return scaled`

Eksempel: Bicycle (Mip-NeRF 360, 194 foto-frames) → SfM-init ~156 K punkter, `T62 = 150 000`, `T72 = 5.32`, buffer-kapacitet 8 M. Resolved cap = $\min(8M, \max(150K, \text{ceil}(156K \times 5.32))) = \min(8M, 830K) = 830 K$. Det er det effektive vækst-cap, som MCMC-relocation-logikken holder sig til.

KORT FORTALT Beregner det reelle maksimum-splat-antal ved MCMC. Tager din indstilling, ser, hvor mange punkter din scene har i starten, og skalerer med multipliseren, hvis automatisk tilpasning er tændt. Sådan tilpasser cap'et sig scenen i stedet for at tvinge samme værdi til en lille og en gigantisk scene. Du behøver ikke selv kalde metoden — træningen bruger den internt.

Hvilket felt til hvad? (cheat sheet)

Mål	Felter at dreje på
Mere detalje i det fjerne	<code>T62 mcmcMaxGaussians</code> højt, <code>T72 mcmcCapMultiplier</code> 5+
Mere detalje generelt (Classic)	<code>T1 maxIterations</code> højt ($\leq 40K$), <code>T2 densifyUntilIteration</code> $\leq 14\%$ af T1
Reducere floaters i dronflyvninger	<code>T43 frustumCullEnabled</code> til, <code>T20 skyMaskingEnabled</code> til, <code>T45 skyDomeEnabled</code> til
Pæn himmel i udendørs scener	<code>T45 skyDomeEnabled</code> til, <code>T47 skyDomeRadiusMultiplier</code> 30–60
Mindre eksport-fil	Strategi <code>.mcmc</code> (T61), <code>T56 postTrainingCompactification</code> til, <code>T62 mcmcMaxGaussians</code> $\leq 200K$
Hurtigere træning	<code>T22 trainingRenderScale</code> 0.5, <code>T1 maxIterations</code> halvere — men ikke begge!
Bedre højlys	<code>T21 shDegreeUpgradeIterations</code> med <code>[2K, 5K, 8K]</code> (ingen early-front-load), MCMC + 200K iter
Holde Mac responsiv	<code>T25 throttleDelayMs</code> 5–10 (koster ~15 % træningstid)
Live-forhåndsvisning oftere	<code>T59 livePreviewInterval</code> ned til 10–20
Blødere overgange ved skygger	<code>T17 ssimWeight</code> lidt højt (0.15–0.25), men ikke over 0.3
Holde indendørs rum kompakt	<code>P10 Indoor-preset</code> (, <code>T72 = 1.76</code>)

Farlige felter

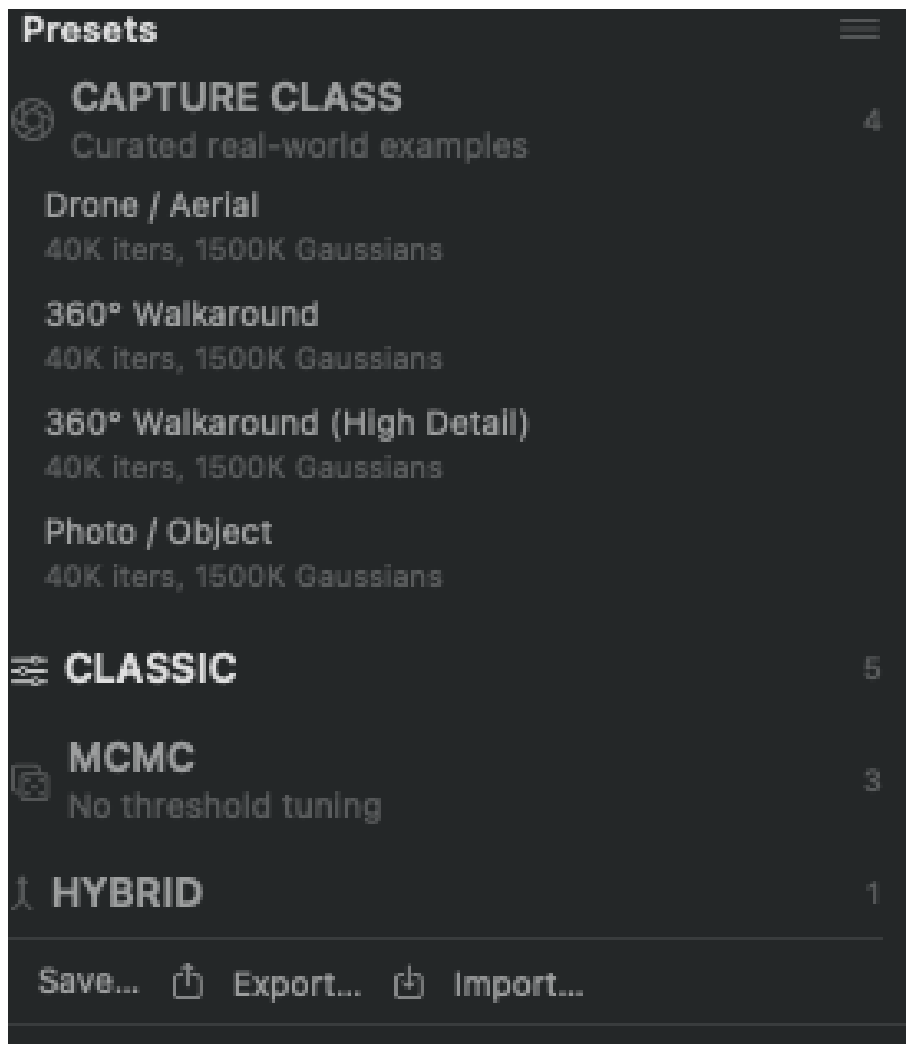
Disse felter kan ved fejl-konfiguration føre til OOM, app-crash, mass-extinction af gaussians eller ubrugelige benchmark-data. Skal håndteres med forsigtighed:

- T11 `densifyGradThreshold` — en halvering kan generere 2–4× så mange gaussians, hvilket hurtigt sprænger GPU-lageret. Bemærk også: skal passe til T22 `trainingRenderScale` (1.0× → 1e-6, 0.5× → 2e-6, 0.25× → 4e-6).
- T72 `mcmcCapMultiplier` — ved store scener med > 200 K SfM-init-punkter og multiplier > 5 opstår et resolved-cap på millioner af gaussians. På 36-GB-RAM-Mac'er OOM muligt. Outdoor-preset 5.32 fungerer kun, fordi Mip-NeRF-360-Bicycle har 156 K init-punkter → 830 K cap.
- T39 `testViewIndices` — manuel indstilling kan gøre benchmarket ubrugeligt (alle indekser > N → ingen holdouts). Lad `--benchmark -flaget` sætte det.
- T64 `mcmcOpacityRegWeight` **og** T65 `mcmcScaleRegWeight` — i 1.4.3-beta sat til 0.01, hvilket førte til mass-extinction (460 K → 5 gaussians på en iteration). Siden 1.4.4 fastsat på 0.0, men manuel forhøjelse kan reproducere problemet.
- T15 `opacityResetInterval` — hvis ikke 100 000+ (reelt fra) og træningen er kortere end 10 000 iterationer, ødelægger reset konvergensen. `.preview` har det derfor på 100 000 trods `maxIterations = 5 000`.
- T54/T55 `densifyPhase2*` — two-phase densification er i tests afsluttet i 0-gaussians-cascade. Lad begge stå på 0.
- T74 `useMipSplatting` — Q1.5 closed-no-win 2026-05-25, kan endda forværre PSNR på nogle outdoor-scener. Default off, opt-in kun til forskning.

Hvis et felt står på denne liste, og du vil ændre det, så lav først en sikkerhedskopi af din nuværende preset (eksport som JSON), og overvej, om du kan måle resultatet reproducerbart — ellers ved du bagefter ikke, om du har fremkaldt en forbedring eller forværring.

KAPITEL

Kapitel 7 — Indbyggede kvalitetsforudindstillinger



Figur 27: Forudindstillingssektion med alle fire grupper foldet ud — 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, bemærkning „No threshold tuning“), HYBRID (1 preset: Balanced (Hybrid))

HVAD BILLEDET VISER Forudindstillingssektion i Inspector, alle fire grupper foldet ud. CAPTURE CLASS med de fire kuraterede Real-World-presets (Drone / Aerial, 360° Walkaround, 360° Walkaround (High Detail), Photo / Object) — det er den primære gruppe og i Simple Mode den eneste synlige. CLASSIC med Quick (1K iters), Preview (5K

iters, aktivt valg med blå flueben), Balanced (20K iters), Quality (35K iters) og Ultra Detail (35K iters). MCMC med undertitel „No threshold tuning“ — MCMC har ikke brug for densify-til-tærskel: Preview (60K iters, 100K gaussians), Balanced (120K, 150K), Quality (200K, 150K). HYBRID med Balanced (Hybrid) (20K iters, 150K gaussians). Footer-action-row: Save..., Export..., Import...

En forudindstilling er en forberedt konfiguration til træningen. RadianceKit leveres med tretten indbyggede forudindstillinger i fire grupper: fire **Capture-Class**-presets (P9–P12) — kuraterede, på rigtigt community-materiale per øje validerede opskrifter til reelle optage-arter (drone, 360°-rundgang, foto-objekt) og den primære akse siden v1.6 —, fem Classic-presets (P1–P5: Quick/Preview/Balanced/Quality/Ultra Detail), tre MCMC-presets (P6–P8) og en Hybrid-preset (P13), der kombinerer Classic- og MCMC-strategierne. De tidligere „Scene-Class“-presets (Render/3D, Outdoor, Indoor, i fase Q7 akademisk tunet mod Mip-NeRF-360- og NeRF-Blender-scener) blev i v1.6 trukket tilbage som synlig gruppe — den per øje på rigtigt footage validerede Capture-Class er nu den primære akse; de Q7-tunede konfigurationer bevares kun internt. Du vælger presets i sidebaren under **Presets** eller i Simple Mode ved import. **+** -knapperne åbner dialoger til at oprette egne forudindstillinger ved siden af — de tretten indbyggede kan ikke slettes, men gerne duplikeres.

I Expert-View vises forudindstillingerne grupperet efter optage-art og strategi (Capture Class / Classic / MCMC / Hybrid). Et klik på en post skriver den gemte træningskonfiguration ind i den aktuelle tilstand. Det er ikke et snapshot — hvis du derefter drejer på skydeknapper, ændres tilstanden, men selve presetet forbliver uændret; en farvet bemærkning viser så „modified“.

Hvilken preset der er den rette hvornår, afhænger mest af scenetypen og hardwaren. De tre tabeloversigter sidst i kapitlet sammenfatter det.

| P1 — Quick



Inspector → Presets-sektion → gruppe „Classic“ → post „Quick“. UUID-suffiks `...001`.



Diagnose-preset med 1 000 iterationer, klassisk (adaptiv) densification-strategi og en træningsop-løsnings-skalering på 0.25× (input-billedet formind-skes til 25 % før træningen). Skal ikke bruges til levering af en scene, men hurtigt finde ud af, om opsætningen (kamerapositioner, punktsky, billed-serie) overhovedet viser meningsfuld bevægelse i loss-værdierne. På en M3 Ultra typisk under 30 sekunder på 50–200 billeder. Den lille opløsning tilslører den ægte billedkvalitet, men holder hukom-melses-forbrug og render-arbejde meget lavt. Væl-ges også automatisk som default ved første start, hvis systemet har mindre end 10 GB RAM.

KORT FORTALT

Hurtig funktionstest. Billeder ind, vent et knap halvt minut, kig om scenens rå omrids dukker op. Hvis billedet i vieweren ligner en plumret klat — så er det fint, det skal det. Hvis du derimod kun ser mørke punkter eller en totalt forvrænget form, er kameraposi-tionerne nok forkerte (se kapitel 9). Til et resultat, der kan vises, har du derefter brug for mindst P2 eller P3.

| P2 — Preview (Classic)



Inspector → Presets-sektion → gruppe „Classic“ → post „Preview“. UUID-suffiks `...002`.



5 000 iterationer Classic-densification, 0.5× opløs-nings-skalering, dobbelte læringsrater i forhold til standard. Densification (klon + split) er aktiv over de første 2 500 iterationer, derefter kun pruning. Default-preset for systemer med ≥ 10 GB RAM. På en M3 Ultra typisk 90 sekunder til 3 minutter på en 200-billed-scene. Giver et brugbart indtryk af geo-metrien og kamera-posen, men teksturer er tydeligt blødtegnet — 0.5× render-opløsningen lader sig ik-ke direkte omgå ved at træne igen med P3 eller P4, fordi læringsraterne er kalibreret til den halve opløs-ning.

KORT FORTALT

Standarden for „lige kigge en gang“. Hvis du netop har impor-teret nye billeder og vil se, om scenen overhovedet kan rekon-strueres, er det det rigtige trin. Cirka 2–3 minutters ventetid, derefter kan du dreje i 3D-viewe-ren og vurdere, om det er værd at investere i flere trænings-pas. Først når forhåndsresultatet alle-rede ser godt ud, betaler Balan-ced eller Quality sig.

| P3 — Balanced (Classic)



HVOR

Inspector → Presets-sektion → gruppe „Classic“ → post „Balanced“. UUID-suffiks `...005`.



TEKNISK

20 000 iterationer Classic-densification ved fuld billed-opløsning. Densification kører over de første 15 000 iterationer, fra iter 3 000 med et densify-interval på 100. Empirisk det „sweet spot“ fra de dokumenterede trænings-sessioner: ved klassisk densification på Horse Full og Truck stabiliseres L1-loss mellem iter 18 000 og 22 000, en længere træning giver ikke længere signifikant forbedring under Quality (P4). På en M3 Ultra typisk 30–60 sekunder på 200 billeder, 5–8 minutter på 1 000+ billeder.

KORT FORTALT

Det „gode kompromis“. De fleste scener ser allerede gode ud her, uden at du skal vente en time. Hvis du vil vise slutresultatet et sted (sociale medier, hjemmeside, en kunde-demo), er det ofte nok. Først når du vil zoome ind i splat-modellen eller har brug for detaljer i overfladeteksturen, er det værd at hoppe op til P4 Quality eller P7 MCMC.

| P4 — Quality (Classic)



HVOR

Inspector → Presets-sektion → gruppe „Classic“ → post „Quality“. UUID-suffiks `...003`.



TEKNISK

35 000 iterationer Classic-densification med V546-„Opacity Decay“ (HTGS, Eurographics 2025): efter hver densify-cyklus multipliceres opacity af alle eksisterende gaussians med en faktor < 1.0 , hvilket pålideligt fjerner gaussians, der er blevet inaktive, ved pruning og dermed opnår 14 % bedre L1-loss end den klassiske 35 000-kørsel ved samme iter-antal. SSIM-loss er aktiveret (`ssimWeight=0.05`). På en M3 Ultra typisk 2–4 minutter på 200 billeder. Leverer på NeRF-Blender (Lego, Chair, Drums) final L1 ≈ 0.023 — bedste Classic-variant i de 560+ dokumenterede eksperimenter. Bemærk: kræver ~3–5 GB GPU-hukommelse; på 8-GB-systemer er P3 det sikre valg.

KORT FORTALT

Den bedste klassiske variant. Leverer skarp tekstur og fin geometri, særligt på objekt-optagelser (en skulptur, en stol, en vase). Ved store outdoor-scener eller rum mærker du derimod næsten ingen forskel fra Balanced — der er det mere værd at skifte til en MCMC-preset (P6–P8) eller en Capture-Class-preset (P9–P12) end at hoppe fra P3 til P4. Vil du have det absolutte maksimum af Classic-familien, så tag P5 Ultra Detail.

| P5 — Ultra Detail (Classic)



HVOR

Inspector → Presets-sektion → gruppe „Classic“ → post „Ultra Detail“. UUID-suffiks `...008` .



TEKNISK

Cirka 35 000 iterationer Classic-densification — vinderen af held-out-kørslen i kvalitets-matrixen (2026-06-10). På alle tre testede Mip-NeRF-360-scener slår Ultra Detail den indbyggede MCMC-„Quality“-preset (P8) ved sammenlignelig wall-clock-tid med rundt +0.94 dB PSNR. Dermed er det den stærkeste Quality-preset i Classic-gruppen og den skarpeste Classic-variant, RadianceKit leverer. På en M3 Ultra typisk i samme tidsramme som P4 Quality (2–5 minutter på 200 billeder), men kræver lidt mere GPU-hukommelse; på 8-GB-systemer er P3 fortsat det sikre valg.

KORT FORTALT

Det skarpeste Classic-trin og held-out-vinderen af vores kvalitets-tests: på rigtige scener rundt en decibel bedre end MCMC-„Quality“-varianten — ved tilsvarende ventetid. Vil du have maksimal detaljetroskab med den gennemprøvede klassiske densification og har nok GPU-hukommelse, er det førstevalget. Rækker hukommelsen ikke, eller har du brug for en så lille eksport-fil som muligt, så bliv ved P4 Quality eller en MCMC-preset.

| P6 — Preview (MCMC)



HVOR

Inspector → Presets-sektion → gruppe „MCMC“ → post „Preview“. UUID-suffiks `...006` .



TEKNISK

60 000 iterationer MCMC-densification (3DGS-MCMC, NeurIPS 2024) ved et cap på 100 000 gaussians. MCMC erstatter den heuristiske klon/split-logik med Markov-Chain-Monte-Carlo-relocation: døde gaussians placeres på ny via sigmoid-vægtede sampling-dybder, hvilket giver et kontrolleret og reproducerbart antal gaussians. Cap'et lægger låg på det maksimale antal hårdt ved 100K — det sparer hukommelse og render-tid, men koster detalje. På en M3 Ultra typisk 5–8 minutter på 200 billeder. Egnede som „MCMC-funktionstest“ — hjælper med at vurdere, om et skift fra Classic til MCMC ville være meningsfuldt, før du investerer mere tid i P7 eller P8.

KORT FORTALT

Som P2 Preview, men med den nyere MCMC-metode. Leverer ofte lidt mere kompakte, jævne fordelte splot-skyer end Classic-varianten. Til en første vurdering af en scene rækker de 5–8 minutter som regel. Hvis du kan lide forhåndsresultatet, er næste skridt P7 (Balanced) eller direkte P8 (Quality MCMC).

| P7 — Balanced (MCMC)



Inspector → Presets-sektion → gruppe „MCMC“ → post „Balanced“. UUID-suffiks ...007 .



120 000 iterationer MCMC ved et cap på 150 000 gaussians. Det mellemste MCMC-trin — næsten det endelige gaussian-antal fra P8 Quality, men kun 60 % af iterationerne. Empirisk ligger L1-loss i de dokumenterede trænings-sessioner på 0.026–0.028 på Horse Full, mod P8 med 0.0246 — altså rundt 7 % højere, til gengæld halv ventetid. På en M3 Ultra typisk 8–15 minutter på 200 billeder. Bruger en metode, der skalerer det effektive gaussian-cap til punktdensiteten i input-SfM-punktskyen (se T75 i kapitel 6).

KORT FORTALT

MCMC med ordentlig detaljerighed, men uden den lange fuldkørsel fra P8. Til de fleste scener rækker det, særligt hvis du vil presse en MCMC-kørsel ind i frokostpause-tidsrammen. Hvis hukommelsen er knap (f.eks. på M-processorer med kun 16 GB), så bliv her — P8 kræver mere GPU-hukommelse.

| P8 — Quality (MCMC)



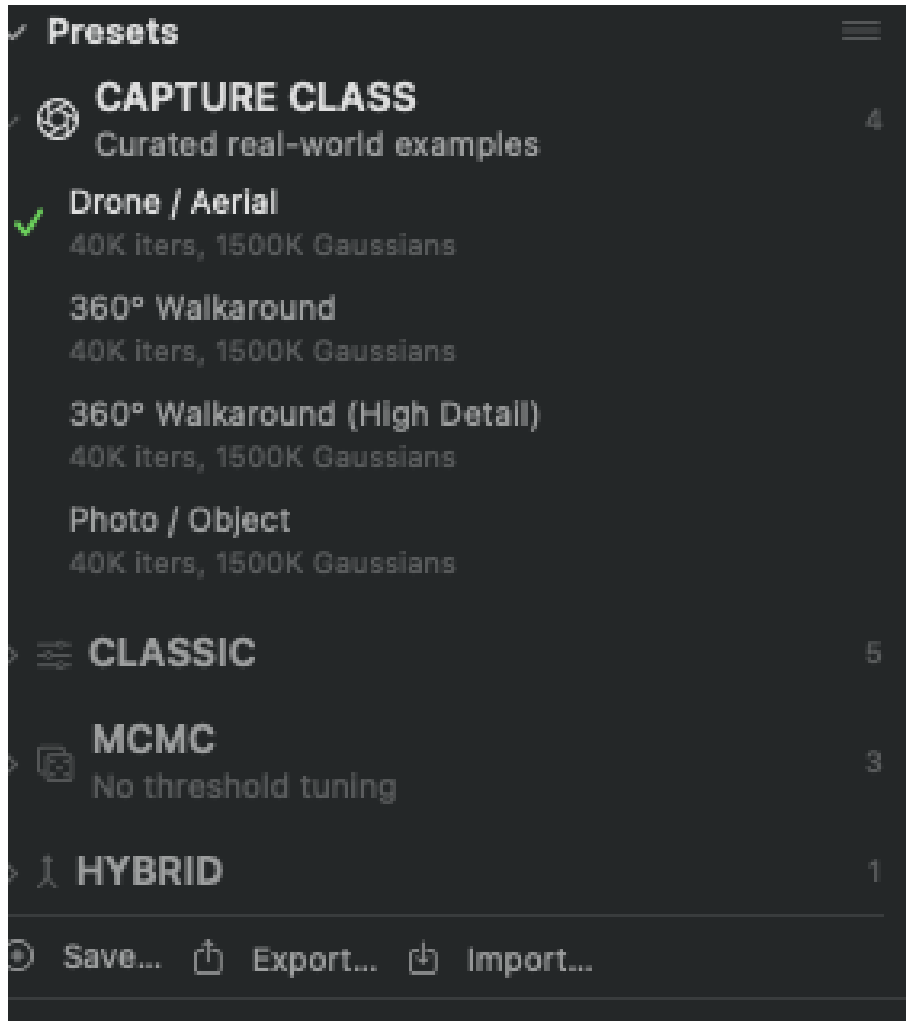
Inspector → Presets-sektion → gruppe „MCMC“ → post „Quality“. UUID-suffiks ...004 .



200 000 iterationer MCMC ved et cap på 150 000 gaussians, SSIM-loss 0.05, MCMC-noise-decay over 80 % af iterationerne. Best-single-run-L1 i de 560+ eksperimenter: 0.0238 på Horse Full, gennemsnit over 3 trials 0.0246 (mod P4 0.0230 på samme scene). MCMC leverer samtidig 71 % færre gaussians (150K vs ~524K) — afgørende, hvis du vil levere resultatet på nettet, fordi den mindre sky giver markant mindre eksport-filer. Træningstid på en M3 Ultra typisk 20–35 minutter på 200 billeder; på 1 000+ billed-sæt nærmere 1–2 timer. Bedste valg, hvis maksimal billedkvalitet ved minimal slutstørrelse ønskes.

KORT FORTALT

Den bedste MCMC-variant. Leverer meget rene, kompakte splat-skyer — ideelt, når du senere vil indlejre resultatet som web-3D-viewer eller sende det som fil (filen er mindre end ved P4 Quality med sammenlignelig optisk kvalitet). Det kræver dog tålmodighed — på store optagelser over en times ventetid. Planlæg det snarere som et „natlig kørsel“.



Figur 28: CAPTURE CLASS-gruppen foldet ud med alle fire kuraterede Real-World-presets — Drone / Aerial (MCMC, 40K iters, Cap 1,5 M), 360° Walkaround (MCMC, 40K, Cap 1,5 M), 360° Walkaround (High Detail) (Hybrid, 40K, Cap 1,5 M, opt-in) og Photo / Object (Hybrid, 40K, Cap 1,5 M). Denne gruppe står øverst og er i Simple Mode den eneste synlige.

HVAD BILLEDET VISER Inspector med CAPTURE CLASS-gruppen foldet ud — den primære preset-gruppe siden v1.6, i Simple Mode den eneste viste. Hver post er en på rigtig community-materiale per øje valideret opskrift til en konkret optage-art (drone, 360°-rundgang, foto-objekt), ikke en værdi optimeret mod et akademisk test-sæt. Valg ved klik skriver den gemte træningskonfiguration ind i den aktuelle tilstand.

| P9 — Drone / Aerial



Inspector → Presets-sektion → gruppe „Capture Class“ → post „Drone / Aerial“. UUID-suffiks `...010`.



Capture-Class-preset til luft- og drone-orbits af bygninger og landskaber. MCMC-densifier, 40 000 iterationer, cap 1,5 mio. gaussians, SSIM-loss 0.5 plus edge-aware-term 0.1. Afgørende er anisotropi-straffen med vægt 0.003 ved en ratio-tærskel på 6 — „spaghetti-dræberen“ mod de typisk nåleformede artefakter, som drone-footage frembringer. Valideret på en ægte DJI-4K-droneflyvning over Pensford-viadukten (per øje testet, ikke kun metrisk).

KORT FORTALT

Til optagelser fra luften — drone-flyvninger om en bygning, over et landskab, langs en facade. Den kraftige anisotropi-straf rydder de nåle- eller spaghettiformede artefakter væk, som drone-materiale gerne frembringer. Hvis dit materiale er optaget fra jorden, passer snarere Photo / Object eller en Classic-preset.

| P10 — 360° Walkaround



Inspector → Presets-sektion → gruppe „Capture Class“ → post „360° Walkaround“. UUID-suffiks `...011`.



Capture-Class-preset til 360°-walkaround-videoer. MCMC-densifier, 40 000 iterationer, cap 1,5 mio. gaussians, SSIM-loss 0.5 plus edge-aware-term 0.1, mild anisotropi-straf (vægt 0.001 ved ratio-tærskel 15). Person- og himmel-maske er aktive. Presetet forventer en 360°-equirect-video, der internt repro-jiceres til rundt 90° brede perspektiv-crops, før træningen starter. Valideret på 8K-360°-rundgange med selfie-stick (Monument-scenen, per øje testet).

KORT FORTALT

Til 360°-rundgang-videoer — du går med et 360°-kamera eller en selfie-stick gennem et rum eller rundt om et objekt. RadianceKit opdeler selv kugle-panoramaet i normale synsvinkler og maskerer forbipasserende og himmel væk. For maksimal skarphed på samme materiale, prøv desuden High-Detail-varianten (P11).

| P11 — 360° Walkaround (High Detail)



HVOR

Inspector → Presets-sektion → gruppe „Capture Class“ → post „360° Walkaround (High Detail)“. UUID-suffiks ...013 (Opt-in).



TEKNISK

Opt-in-Capture-Class-preset til 360°-walkaround-videoer med maksimal detalje. Hybrid-densifier (klassisk abs-gradient-klon/split

1. MCMC-noise + relocation), 40 000 iterationer, cap 1,5 mio. gaussians,

anisotropi-straf 0.0015 ved ratio-tærskel 15, SSIM-loss 0.2 og edge-aware-term 0 — den lockede „r50“-screen-split-opskrift. På 360°-materiale slår det standard-MCMC-presetet „360° Walkaround“ (P10) ved PSNR, LPIPS og synligt konfetti, og det med rundt en tredjedel af splat-antallet. Står bevidst opt-in *ved siden af* standard-360-presetet, indtil det er valideret på flere scener.

KORT FORTALT

Det skarpere alternativ til standard-360-presetet (P10): mere detalje, mindre konfetti, markant mindre fil. Står bevidst ved siden af i stedet for at erstatte det — hidtil bekræftet på en håndfuld scener. Hvis din 360°-rundgang er rent optaget, så prøv dette preset først og sammenlign resultatet med P10.

| P12 — Photo / Object



HVOR

Inspector → Presets-sektion → gruppe „Capture Class“ → post „Photo / Object“. UUID-suffiks ...012 .



TEKNISK

Capture-Class-preset til objekt-orbits fra skarpe enkeltfotos (ingen video). Hybrid-t1-densifier (med relocation), 40 000 iterationer, cap 1,5 mio. gaussians, SSIM-loss 0.5 plus edge-aware-term 0.1, mild anisotropi-straf (vægt 0.001 ved ratio-tærskel 15), opacity-decay 0.9995 hver 50. iteration, **ingen** masking. Valideret på 163 højopløselige 41-MP-fotos af et skelet (per øje testet). Få views (op til cirka 600) bliver dermed under hybrid-collapse-tærsklen.

KORT FORTALT

Til objekt-optagelser fra skarpe enkeltfotos — du omkredser en skulptur, en model, et produkt med kameraet og tager fotos i stedet for video. Ingen masking, fordi skarpe fotos som regel har ren baggrund. Til video-kilder, tag i stedet en 360°- eller Drone-preset.

| P13 — Balanceret (Hybrid)

HVOR

Inspector → Presets-sektion → gruppe „Hybrid“ → post „Balanceret (Hybrid)“. UUID-suffiks `...009`.

TEKNISK

20 000 iterationer med Hybrid-densification-strategien: klassisk gradient-drevet clone/split placerer kapacitet dér, hvor loss'en har brug for den, MCMC-SGLD-noise bliver ved med at udforske, og døde gaussians flyttes i stedet for at gå tabt ved pruning. Opacity-decay (V546) erstatter opacity-resets; en anisotropi-straf (vægt 0.001, ratio-tærskel 15) holder nåleformede splats i skak. Gaussian-cap'en skalerer med scenen (150K basis, scene-aware $\times 3.0$). Valideret på fem scener mod ren MCMC ved samme budget: i gennemsnit +0.45 dB PSNR ved 20–30 % færre gaussians (stonehenge +1.23, family +0.82, garden +0.47 dB). På en M3 Ultra typisk 5–10 minutter på 200 billeder.

KORT FORTALT

Et stærkt førstevalg til et endeligt resultat: skarpere detaljer end MCMC-forudindstillingerne ved en tilsvarende kompakt fil, på en brøkdel af P8-træningstiden. Hvis du kun har tid til én kvalitetskørsel og ingen af Capture-klasserne klart passer, så start her. Classic-forudindstillingerne er fortsat bedre til hurtige tests, og Capture-Class-presets (P9–P12) er førstevalget, når din scene klart matcher en af disse optage-arter.

Hvornår hvilken preset?

Scenarie	Første-test	Hovedkørsel
Funktionstest af nye billeder, < 30s	P1 Quick	—
Objekt-orbit fra skarpe enkeltfotos	P2 Preview	P12 Photo / Object
Enkelt-objekt-scan (video), < 500 fotos	P2 Preview	P4 Quality eller P8 Quality MCMC
360°-walkaround-video	P6 Preview MCMC	P10 360° Walkaround (skarp: P11 High Detail)
Luft-/drone-orbit, landskab	P6 Preview MCMC	P9 Drone / Aerial
Web-levering (lille, kompakt)	P2	P8 Quality MCMC (mindste fil ved fuld kvalitet)
Skarpe detaljer på kort tid, kompakt eksport	P2 eller P6	P13 Balanceret (Hybrid)
Maksimal detaljetroskab, Classic-strategi	P3 eller P6	P5 Ultra Detail
Print, marketing, fuld detalje	P3 eller P6	P4 Quality (Classic) eller P5 Ultra Detail

Hurtig sammenligning

Pre-set	Strategi	Iters	Max-Gs	Render-skala	Typisk tid (200 billeder, 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 HT-GS
P5 Ultra Detail	Classic	~35 000	∞	1.0x	2–5 min	Matrix Δ+0.94 dB
P6 Preview MCMC	MCMC	60 000	100 K	1.0x	5–8 min	—
P7 Balanced MCMC	MCMC	120 000	150 K	1.0x	8–15 min	—
P8 Quality MCMC	MCMC	200 000	150 K	1.0x	20–35 min	V544a
P9 Drone / Aerial	MCMC	40 000	1.5 M	1.0x	10–25 min	Øje / via-duk
P10 360° Walkaround	MCMC	40 000	1.5 M	1.0x	10–25 min	Øje / monument
P11 360° Walkaround (High Detail)	Hybrid	40 000	1.5 M	1.0x	10–25 min	Øje (opt-in)
P12 Photo / Object (Hy-	Hybrid	20 000	1.5 M	1.0x	10–25 min	Øje / Matrix Δ+0.45 dB

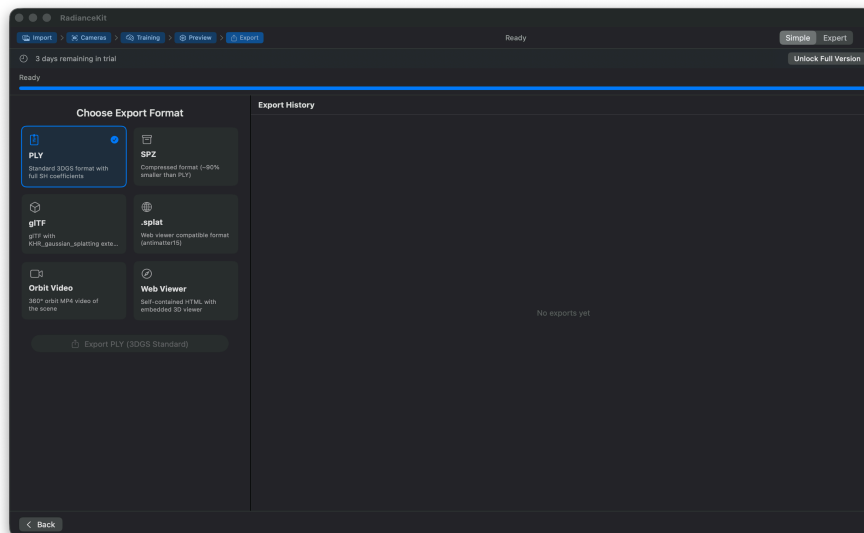
Egne forudindstillinger

Via knappen **Save...** i forudindstillingssektionen (I1 i kapitel 2) gemmer du den aktuelle træningskonfiguration som en egen preset. Egne presets er ikke „Built-in“ og kan omdøbes, eksporteres (som JSON), deles via træk-og-slip, dupliceres og slettes. De tretten indbyggede presets P1–P13 forbliver upåvirkede af slet-knappen.

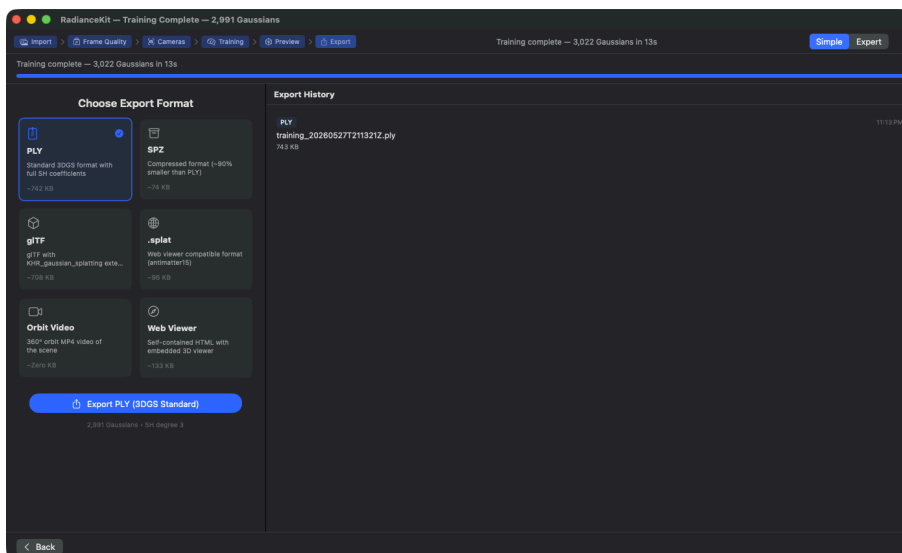
Tommelfingerregel: Hvis du ændrer noget ved en preset, som du vil bruge oftere — sky-dome tændt, højere SSIM-vægt for en bestemt scene-klasse, afvigende iter-antal — så gem varianten som en egen preset. Så ved du straks ved næste kørsel, at det er en konfiguration, der afviger fra standarden.

KAPITEL

Kapitel 8 — Eksportformater



Figur 29: Eksportformat-valg i Simple Mode — seks format-kort



Figur 30: Eksportformat-grid live efter 5K-iter-træning på flowers-buket — alle seks kort med dynamisk størrelsesberegning (PLY 742 KB valgt, SPZ 74 KB, glTF 708 KB, .splat 96 KB, Orbit Video ~Zero KB, Web Viewer 133 KB), Export History til højre med allerede gemt PLY

Hvad billedet viser (2 991 gaussians, SH degree 3, Bjoerns syntetiske Blender-buket som IP-clean test-sæt): Størrelsesangivelserne under hvert format-kort beregnes live ud fra aktuelt gaussian-antal og format-overhead — ikke hårdkodet. Ud af

2 991 gaussians (SH degree 3) opstår 742 KB PLY, 74 KB SPZ (faktor ~10× mindre via kvantisering), 708 KB glTF (med `KHR_gaussian_splating-extension`, derfor næsten PLY-ækvivalent), 96 KB `.splat` (komprimeret 24-byte-pr.-gaussian-format). Orbit Video viser „~Zero KB“, fordi størrelsen først kendes efter MP4-encoding. Web Viewer (133 KB) samler en selvstændig HTML-fil med indlejret WebGL-viewer og komprimerede `splat-data` — større end rent `.splat` på grund af viewer-overhead. Export History til højre lister allerede afsluttet PLY-eksport („`training_20260527T211321Z.ply`“, 743 KB, 23:13“) med format-pille og `reveal-i-Finder-action`.

En afsluttet træning leverer en gaussian-cloud — en samling på nogle hundredtusinde til millioner af 3D-gauss-fordelinger, der tilsammen rekonstruerer scenen. RadianceKit kender ti måder at skrive denne sky til disk på. Seks af dem er rene 3D-dataformater (PLY, Compressed PLY, SPZ, SOG, glTF, `.splat`), et bundler skyen sammen med en færdig HTML-viewer (Web Viewer), et renderer en MP4-fil ud af en orbit-kamerakørsel (Orbit Video), og to eksporterer ingen gaussian-indhold, men kun SfM-resultatet (kamera-positioner og grov punktsky) til genbrug i andre trænings-pipelines (`transforms.json` + `COLMAP-workspace`).

Hvilket format der er det rette hvornår, afhænger af målet. Til arkivering af de fulde data uden kvalitetstab tager man PLY. Til web-viewer på egen side rækker som regel `.splat` eller den indbyggede web-viewer. Hvis filen skal være minimal, betaler SPZ eller SOG sig. Til genbrug af SfM-resultatet i Nerfstudio, Postshot eller Brush er `transforms.json` og `COLMAP-workspace` de rigtige veje.

Alle eksportfunktioner ligger i menuen „Export“ og i Simple Mode på det sidste wizard-trin. De fleste formater er fuldt sandbox-konforme og virker i App Store-versionen. Kun SOG kræver en ekstern binary (`cwebp`), som ikke nødvendigvis er til stede i App Store-buildet — detaljer se E4.

I E1 — PLY (.ply)

HVOR

Menulinjen → Export → 3D Formats → Export PLY... (⌘E). Simple Mode: wizard-trin Export → format-kort „PLY“. **Størrelse:** typisk 100 % (referenceværdi). **Kompatibel med:** SuperSplat, PolyCam, alle 3DGS-viewere.

TEKNISK

PLY er det kanoniske lagringsformat for 3D Gaussian Splatting. RadianceKit skriver en binær Little-Endian-fil med det standardiserede 3DGS-property-layout: pr. gaussian trekomponent position, tre normaler altid sat til nul, tre DC-SH-koefficienter (`f_dc_0..2`) for basis-RGB-farven, derefter op til 45 yderligere SH-koefficienter (`f_rest_0..44`) i den i Kerbl-2023-papiret definerede transponerede channel-major-ordning (først alle R-kanal-koefficienter, så alle G, så alle B), efterfulgt af logit-opacitet (rå pre-sigmoid-værdier), tre log-space-skaler og en wxyz-quaternion-rotation. Den maksimalt eksporterede SH-grad klampes til minimum af brugerønske og faktisk lært grad; default er 3 (45 rest-koefficienter). Før skrivning beregnes payload-størrelsen i 64-bit integer for at fange overflow ved ekstremt store skyer. Filen skrives atomisk, hvilket ved store skyer kortvarigt optager dobbelt diskplads.

KORT FORTALT

Det er „originalfilen“. Største fil, højeste kompatibilitet, ingen tab. Hvis du ikke ved, hvilket format du skal vælge, så tag PLY — det åbnes i næsten alle 3DGS-værktøjer. Til 1 million gaussians bliver det alt efter SH-grad mellem 200 og 800 MB. Hvis filen bliver for stor, så kig på E2 (komprimeret PLY) eller E3 (SPZ).

| E2 — Compressed PLY (.ply)

HVOR

Menulinjen → Export → 3D Formats → Export Compressed PLY... Simple Mode: format-kort „Compressed PLY“. **Størrelse:** ca. 10–20 % i forhold til PLY (5- til 10-gange komprimering). **Kompatibel med:** SuperSplat, PlayCanvas-engine, webbaserede viewere.

TEKNISK

PlayCanvas-varianten af PLY-formatet med chunked kvantisering. Gaussians grupperes i 256-stykker-chunks. Pr. chunk lægges min/max-bounds for position, scale og color separat i headeren; de enkelte gaussians refererer deres værdier relativt til disse bounds og komprimeres til hver 32 bit: position og skala med 11-10-11-bit-pakning, rotation som 2-10-10-10-bit „smallest-three“- quaternion, farve som 8-8-8-8-RGBA. Højere SH-koefficienter kvantiseres med kun 8 bit pr. komponent (`shCoeffCount * 3` uchar pr. gaussian). Selve formatet er stadig ASCII-header-PLY og derfor principielt validerbart med PLY-værktøjer, men vertex-properties er erklæret som `uint`-felter. SH-grad er pr. default 0 (ingen rest-koefficienter) for at maksimere komprimeringen — højere SH-grader kan vælges eksplicit.

KORT FORTALT

Den pladsbesparende PLY-variant. Identisk engine-kompatibilitet som almindeligt PLY, men 5 til 10 gange mindre. SuperSplat og PlayCanvas læser det nativt. Til web-deployment næsten altid bedre end almindeligt PLY. Kvalitetstabet via kvantisering er som regel ikke visuelt mærkbart, så længe scenen ikke indeholder ekstremt højfrekvente detaljer.

| E3 — SPZ (.spz)

HVOR

Menulinjen → Export → 3D Formats → Export SPZ...
Simple Mode: format-kort „SPZ“. **Størrelse:** ca. 10 % i forhold til PLY (90 % mindre). **Kompatibel med:** Niantic Scaniverse, Niantic Spatial Fields, MetalS-platter.

TEKNISK

Niantics SPZ-v2-format. Positioner pakkes som 24-bit-fixed-point (det giver ca. 0,25 mm opløsning), skaler som 8-bit-kvantisering i log-rum, rotationer som 8-bit-smallest-three (i v2 gemmes kun xyz, w afledes i decoderen af quaternion-normen), opaciteter som sigmoideret 8-bit-værdier. DC-SH gemmes med en SPZ-specifik pakke-formel ($dc_raw * 0.15 * 255 + 0.5 * 255$), højere SH-bånd med 5 bit (bånd 1) henholdsvis 4 bit (bånd 2-3) pr. koeficient. Hele den pakkede binær-blob komprimeres derefter med standard-gzip (RFC 1952), hvilket giver et gzipped-container-format med magic bytes `1f 8b`. RadianceKit kalder hertil `system-gzip`, fordi Apples indbyggede zlib-API producerer proprietær Apple-framing, der ikke ville være kompatibel med SPZ-readerne i Spatial Fields eller MetalS-platter. `System-gzip` kan stadig spawnes inden for macOS-sandboxen.

KORT FORTALT

Den mindste standardfil. Hvis du kender Niantics Scaniverse — det er det format, appen bruger. Meget lille, meget indlæsningsvenlig for mobile apps. Direkte brugbar i Niantics egen cloud-viewer (Spatial Fields). Cirka 90 % mindre end et PLY med samme data, samtidig næsten ikke optisk skelnelig for de fleste scener.

E4 — SOG (.sog)

HVOR

Menulinjen → Export → 3D Formats → Export SOG...
Simple Mode: format-kort „SOG“. **Størrelse:** ca. 5–6 % i forhold til PLY (15- til 20-gange komprimering — den mindste mulighed). **Kompatibel med:** PlayCanvas-engine, SuperSplat-editor.

TEKNISK

„Spatially Ordered Gaussians“ — et PlayCanvas-format, der gemmer skyen GPU-ready i flere lossless-WebP-billeder. Først sorteres alle gaussians rumligt via 3D-Morton-code (30-bit Z-order, 10 bit pr. akse), hvilket giver billederne senere cache-lokalitet i rendereren. Så kvantiseres positioner med symmetrisk log-transformation (for bedre dynamikomfang) til 16-bit-værdier og splittes i to RGBA-billeder (`means_l.webp` for de nederste 8 bit, `means_u.webp` for de øverste). Rotationer kodes som `smallest-three` med 3×8-bit plus 2-bit-mode i et RGBA-billede (mode lander i alpha som `252 + largest`). Skaler og DC-SH kvantiseres hver med en 256-indgangs-codebook (percentil-baseret fordelt over alle værdier), indekserne lander i `scales.webp` og `sh0.webp` . De fem billeder plus en `meta.json` med codebooks og bounds pakkes i en ZIP-fil (custom-encoder, fordi sandboxen blokerer system- `zip`) og gemmes med endelsen `.sog` .

Sandbox-advarsel: SOG er den eneste format-mulighed, der kræver en ekstern binary. WebP-encoder-trinet kalder `cwebp` fra `/usr/local/bin/cwebp` eller `/opt/homebrew/bin/cwebp`. Hvis ingen `cwebp`-binary findes, falder koden tilbage på rå PNG-encoding — men: **PNG-fallback fungerer ikke i SuperSplat**. I App Store-versionen evaluér tilgængeligheden ud fra build-varianten; i developer-varianten skal `cwebp` være installeret via Homebrew (`brew install webp`).

KORT FORTALT

Det allermindste 3DGS-format, markant mindre end SPZ. Men: kræver `cwebp`-værktøjet på din Mac, fordi RadianceKit selv ikke kan generere alle billedformater. Installer det en gang med Homebrew, så kører alt. I App Store-versionen muligvis ikke fuldt funktionel — hvis der kommer PNG i stedet for WebP ud ved eksport, kan du ikke åbne filen direkte i SuperSplat. Hvis du arbejder uden Homebrew, så vælg i stedet SPZ (E3).

E5 — glTF (.glb)

HVOR

Menulinjen → Export → 3D Formats → Export glTF... Simple Mode: format-kort „glTF“. **Størrelse:** sammenlignelig med PLY. **Kompatibel med:** glTF-viewere med KHR_gaussian_splatting-extension (Khronos-draft-standard).

TEKNISK

Skriver en selvstændig `.glb`-binærfil (ingen separat bin-fil-vedhæftning) i overensstemmelse med KHR_gaussian_splatting-extension-specifikationen. Positioner gemmes som regulære glTF- `POSITION` -vertex-data (float3), alle andre attributter (rotation som float4, scale som float3, opacity som float, SH-koefficienter som float3 × shCoeffCount) ligger i yderligere vertex-attributter og refereres via extensionen. Vigtigt: glTF bruger højrehåndet Y-up-koordinatsystem, COLMAP/3DGS arbejder Y-down/Z-forward. Eksportøren anvender derfor en 180-graders rotation om X-aksen — positioner skrives om med $(x, -y, -z)$, quaternioner tilpasses til $(w, x, -y, -z)$. Det giver en geometrisk korrekt, håndet (ikke spejlvendt) gengivelse i glTF-viewere. JSON- og binærchunks paddes til 4-byte-alignment, som GLB-standarden kræver.

KORT FORTALT

Det officielle Khronos-standard-format for 3D-data, i den fri-ske udvidelse til Gaussian Splats. Fordel: glTF er udbredt i alle store 3D-engines (Babylon.js, Three.js, Unity, Unreal). Ulempe: udvidelsen er i 2026 stadig i draft-stadie, mange viewere kan den ikke endnu. Meningsfuld især, hvis du integrerer splat-data i en eksisterende glTF-pipeline eller skriver en viewer, der allerede er glTF-i-stand.

E6 — Splat (.splat)

HVOR

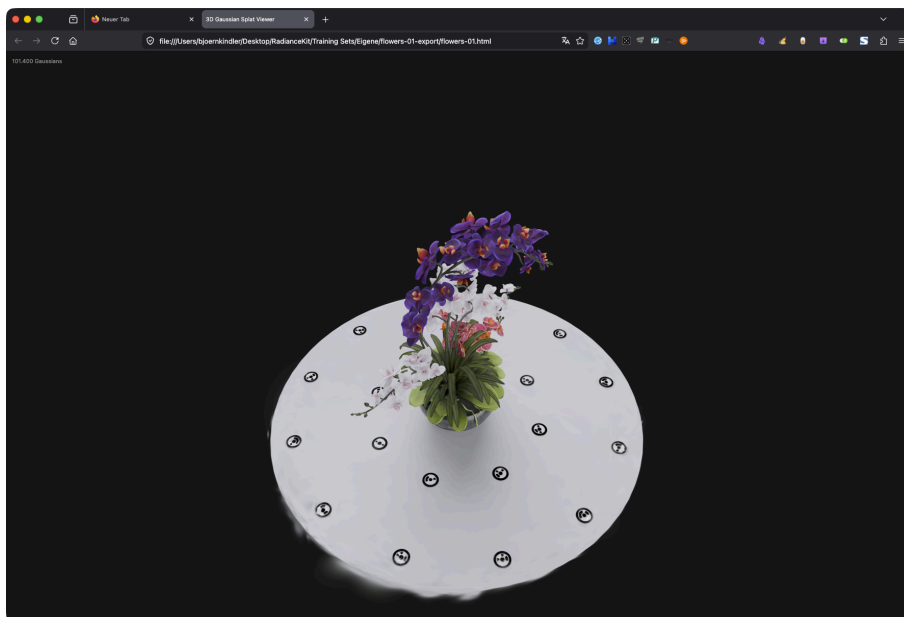
Menulinjen → Export → 3D Formats → Export .splat... Simple Mode: format-kort „.splat“.
Størrelse: præcis 32 bytes pr. gaussian. **Kompatibel med:** gsplat.js, webbaserede viewere (antimatter15-reference), de fleste browser-3DGS-demoer.

TEKNISK

antimatter15- .splat -formatet — 32 bytes pr. gaussian, ingen header, ingen indirektion. Layout pr. post: 3 × float32 position (verdens-koordinater), 3 × float32 scale (exp-transformeret fra log-space i den interne buffer), 4 × uint8 RGBA-farve (DC-SH-koefficient skaleret med `SH_C0 = 0.282...` og klampet til [0,255]), 4 × uint8 quaternion (w,x,y,z, normaliseret og kodet som `128 + 128*q` i byte-området). Kun DC-SH gemmes — højere SH-bånd kasseres. Det gør formatet ekstremt kompakt, men koster de view-afhængige farveændringer, der opstår ved refleksioner eller spekulære højlys. SkrIVERÆKKEFØLGEN er præcis cloudens index-rækkefølge (ingen rumlig sortering), web-viewere som `gsplat.js` renderer ud fra det.

KORT FORTALT

Formatet at vælge, hvis du vil vise splat'en i en egen web-viewer med `gsplat.js`. Meget kompakt (32 bytes/gaussian), men ingen højere SH-grad — altså ingen blanke refleksioner eller fine farveændringer afhængigt af synsvinkel. For de fleste web-anvendelser ingen sag, fordi DC-farve fuldstændigt rækker, og den manglende view-afhængighed næsten ikke ses.



Figur 31: Web Viewer åbnet i Firefox — Bjoerns buket-splat renderet med omgivende kamera-markør-kugler, browser-tab-bar synlig øverst, ingen CDN-/server-opsætning nødvendig. Selvstændig `flowers-01.html` åbnet direkte fra Finder ved dobbeltklik i standardbrowseren — det indlejrede WebGL2-program renderer gaussian-cloud'en med det samme uden netværk eller server. De sorte markører omkring buketten er trænings-kameraerne, valgfrit indkoblelige. Mus-træk roterer, scroll zoomer.

E7 — Web Viewer (.html)

HVOR

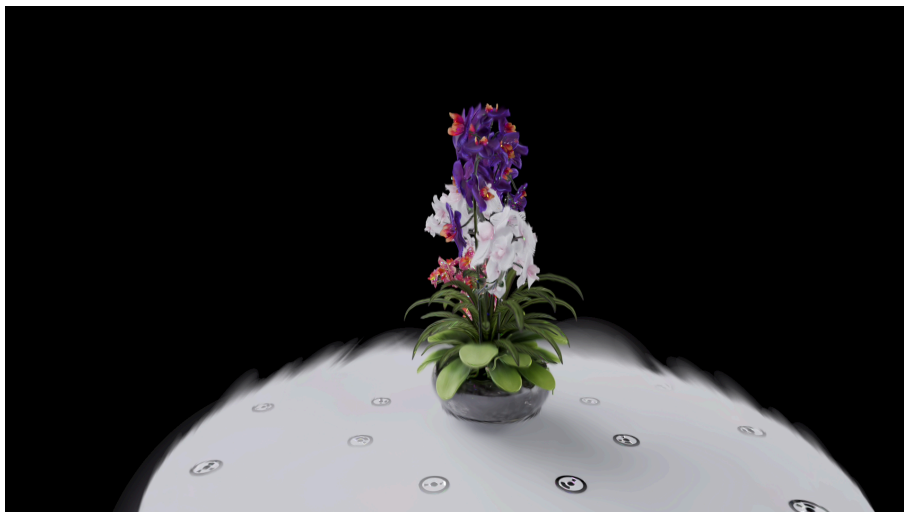
Menulinjen → Export → Media → Export Web Viewer... Simple Mode: format-kort „Web Viewer“.
Størrelse: splat-data base64-kodet ($\approx 4/3$ overhead) + ca. 5 KB HTML/JS-shell. **Kompatibel med:** enhver moderne browser med WebGL2 (alle desktops, iOS 15+, Android 5+).

TEKNISK

Bundler gaussian-cloud'en sammen med en fuldstændig inline skrevet WebGL2-renderer i en enkelt .html -fil. Der er ingen CDN-afhængigheder, ingen WASM, ingen anden fil. Cloud'en kodes internt først som .splat -binær (samme 32-byte-logik som E6), derefter base64-indlejret, derefter dekodet i browseren med `atob`. Den indbyggede renderer laver egen WebGL2-sortering, mus-orbit-styring og CPU-sortering pr. frame; hele JS-koden (shadere, matematik, loop) kan ses i output-HTML'en. Akse-konventionen på lagring-til-renderer-grænsen er præcis den samme som i E5: position $(x, -y, -z)$, quaternion $(w, x, -y, -z)$. Valgfrit kan et branding-overlay indkobles (free-tier-kontakt). Da alt er inline, fungerer filen også direkte fra `file://`-protokollen — ingen lokal webserver nødvendig til test.

KORT FORTALT

En enkelt HTML-fil, du kan sende til nogen pr. mail eller indlejre på en hjemmeside. Dobbeltklik i Finder, og browseren viser din scene med musedrejning. Ingen upload til en sky nødvendig, ingen anden fil, ingen server. Ideel til kunde-præsentationer, portfolio, mail-vedhæftninger. Ulempe: filen bliver ca. en tredjedel større end et rent .splat på grund af base64-kodningen — til meget store scener kan det derfor betale sig med separat hosting af .splat -filen sammen med en standard-viewer.



Figur 32: Enkelt frame udtrukket fra `flowers-01.mp4` — Bjoerns buket i profil-render, hvid platform med kamera-markører synlig, sort baggrund (default-viewport-baggrund, kan ændres i Settings). Kameraet kredser om scenen på en parametriske bane (elevation + afstand fast, yaw roterer), varighed typisk 6–10 sekunder ved 30 eller 60 fps. Frame-opløsning skalérbar fra 480p til 8K via VideoPreset.

| E8 — Orbit Video (.mp4/.mov)

HVOR

Menulinjen → Viewport → Record Turntable Video
ELLER menulinjen → Export → Media → Export Orbit Video.... Simple Mode: format-kort „Orbit Video“ med varigheds-slider 3–30 s. **Størrelse:** afhænger af varighed, opløsning, bitrate. **Kompatibel med:** alle platforme (H.264 og HEVC er Apple-standard).

TEKNISK

Rendrer gaussian-cloud'en langs en parametriske orbit-kamerakørsel og encoderer hvert frame via AVASetWriter til en MP4- eller MOV-fil. Orbit-konfigurationen styrer rotationshastighed (omdrejninger), afstand, elevation, FOV, varighed og ease-in/out-faktor. Orbit-video-eksporten kører gennem RadianceKits EGEN ForwardPass med fuld SH-evaluering — pixel-identisk med den indbyggede viewport (WYSIWYG). Pr. frame multipliceres verdens-tilpasningsmatricen (beregnet af rendereren for at dreje de interne koordinater ind i Y-up-orbit-verdenen) med kameraet, derefter anvendes en kamera-konventions-spejling (camFlip: orbit Y-up → COLMAP Y-down). Offscreen-render-target'et trækkes via IOSurface til en CVPixelBuffer for encoderen. Encoderen understøtter H.264 og HEVC, konfigurerbar bitrate og opløsning fra 480p til 8K. Før første frame venter rendereren 200 ms, så den indledende splat-sortering er afsluttet. Denne eksport er GPU-bound — ved 8K og millioner af gaussians ligger render-tiden pr. frame på flere sekunder, altså samlede render-tider på 10–30 minutter for 6 s video muligt.

KORT FORTALT

En færdig MP4-fil med en drejning omkring din scene. Perfekt til sociale medier, marketing, præsentationer. Du kan indstille varighed (3–30 sekunder), drejning og hastighed. Filen kan indlejres direkte på YouTube, Instagram, i PowerPoint og alle andre steder. Går nogle gange langsomt, fordi appen skal render hvert frame komplet — til en 8K-video kan du regne med fem til tredive minutter, alt efter antal gaussians.

I E9 — SfM Transforms (transforms.json)

HVOR

Menulinjen → Export → Photogrammetry → Export SfM (transforms.json)... **Størrelse:** typisk 1–10 KB (kun positioner + intrinsics, ingen billeder, ingen gaussians). **Kompatibel med:** nerfstudio, Brush, gsplat, OpenSplat, Meshroom, alle moderne feed-forward 3DGS-trænere.

TEKNISK

Skriver nerfstudio- `transforms.json` -formatet med en liste over kamerapositioner plus delte intrinsics. Pr. kamera inverteres view-matricen (RadianceKit-internt: world-to-camera i COLMAP-konvention), derefter spejles kameralokale Y- og Z-basisvektorer for at konvertere til nerfstudio-konventionen (OpenGL-stil, kamera kigger langs `-Z`, `+Y` er op). Den endelige 4×4-matrix lander som row-major nested array af doubles i `transform_matrix`-feltet i hvert frame. Intrinsics gemmes på top-niveau (brændvidde `x/y`, hovedpunkt `x/y`, billedbredde/-højde, `camera_model = "OPENCV"`, plus distortion-coefficients `k1, k2, p1, p2`) — bortset fra hvis eksportøren genkender flere forskellige intrinsics-sæt, så skrives de pr. frame. Billedstier skrives som `images/<filename>` relativt til JSON-filen; brugeren skal oprette en `sibling-images/`-mappe med træningsfotoene.

KORT FORTALT

Denne JSON-fil beskriver for hvert foto, hvor kameraet stod, og hvor det kiggede hen. Filen alene er lille og ubrugelig — den anvendes sammen med originalbillederne i en mappe. Nerfstudio, Brush og et par andre trænere læser præcis dette format, og du kan dermed overdrage dine RadianceKit-SfM-resultater til et andet værktøj, uden at kamerarekonstruktionen skal beregnes på ny der. Sparer timer på store scener.

I E10 — COLMAP Workspace (sparse/0/)

HVOR

Menulinjen → Export → Photogrammetry → Export SfM (COLMAP Workspace).... **Størrelse:** tre binære-filer sammen typisk 4–8 MB — `points3D.bin` dominerer (én linje pr. 3D-punkt i sparse-cloud'en), `images.bin` og `cameras.bin` er hver markant under 100 KB. **Kompatibel med:** COLMAP selv, Nerfstudio, Postshot, Meshroom, alle værktøjer, der forventer et COLMAP-sparse/-katalog.

TEKNISK

Skriver standard-COLMAP-sparse/0/-layoutet med tre binære filer: `cameras.bin`, `images.bin`, `points3D.bin`. Format-reference er den officielle COLMAP-dokumentation. `cameras.bin` indeholder den deduplikerede intrinsics-liste (kameraer med identiske intrinsics + billedstørrelse samles til en enkelt post); det anvendte camera-model er `OPENCV` (model 4), med `fx/fy/cx/cy` plus de fire distorsion-coefficients `k1/k2/p1/p2`. `images.bin` lister pr. billede positionen som `wxyz`-quaternion plus translation, efterfulgt af kamera-ID og filnavn; ingen 2D-3D-korrespondancer gemmes. `points3D.bin` indeholder SfM-punktskyen med position, farve (0-255 RGB) og default-værdier for reprojektion og track-length. Alt skrives i Little-Endian. Re-import i RadianceKit fungerer via File-menuen → „Import COLMAP/Metashape Workspace...” (se Q3 i SfM-bækend-kapitlet).

KORT FORTALT

Det officielle COLMAP-format. Hvis du vil fortsætte din træning i Postshot, Nerfstudio eller en anden COLMAP-i-stand software, er det vejen. Tre små filer plus dine originalbilleder, og målprogrammet accepterer det, som om COLMAP selv havde været kildeprogrammet. Flere programmer forstår det end `transforms.json`-formatet (E9), samtidig lidt mindre handy, fordi det er binært i stedet for tekstbaseret.

Hvilket format hvornår?

Mål	Format
Web-viewer på egen side	E7 Web Viewer (.html)
Web-viewer med <code>gsp1at.js</code>	E6 Splat (.splat)
Pipeline-genbrug i Postshot / Nerf-studio	E9 transforms.json + E10 COLMAP Workspace
SuperSplat-redigering	E1 PLY eller E2 Compressed PLY
Niantic Scaniverse / Spatial Fields	E3 SPZ
Maksimal komprimering	E4 SOG (cwebp påkrævet)
Marketing-/social-video	E8 Orbit Video

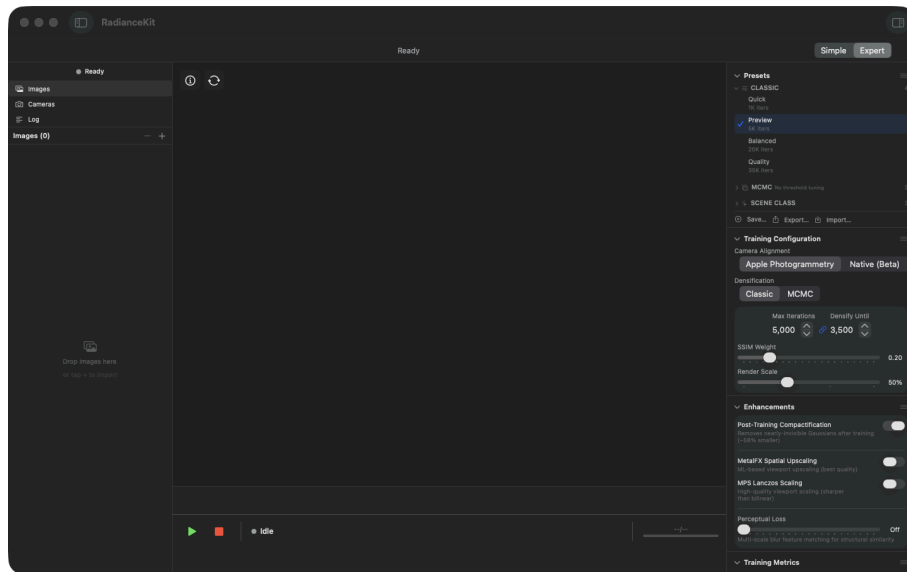
Hurtig sammenligning

Format	Endelse	Sandbox	Størrelse (1M gauss)	Bedst til
E1 PLY	.ply	ja	~250 MB	Arkiv, højeste kompatibilitet
E2 Compressed PLY	.ply	ja	~40 MB	Web + SuperSplat
E3 SPZ	.spz	ja (gzip-spawn)	~40 MB	Niantic + mobil
E4 SOG	.sog	betinget (cwebp)	~20 MB	Maksimal komprimering
E5 glTF	.glb	ja	~250 MB	Khronos-pipeline
E6 Splat	.splat	ja	~32 MB	gsplat.js web-viewer
E7 Web Viewer	.html	ja	~45 MB	Standalone browser-fil
E8 Orbit Video	.mp4 / .mov	ja	variabel	Social/marketing
E9 SfM Transforms	.json	ja	~5 KB	Pose-overdragelse
E10 COLMAP Workspace	Katalog	ja	~4–8 MB	Pose-overdragelse binær

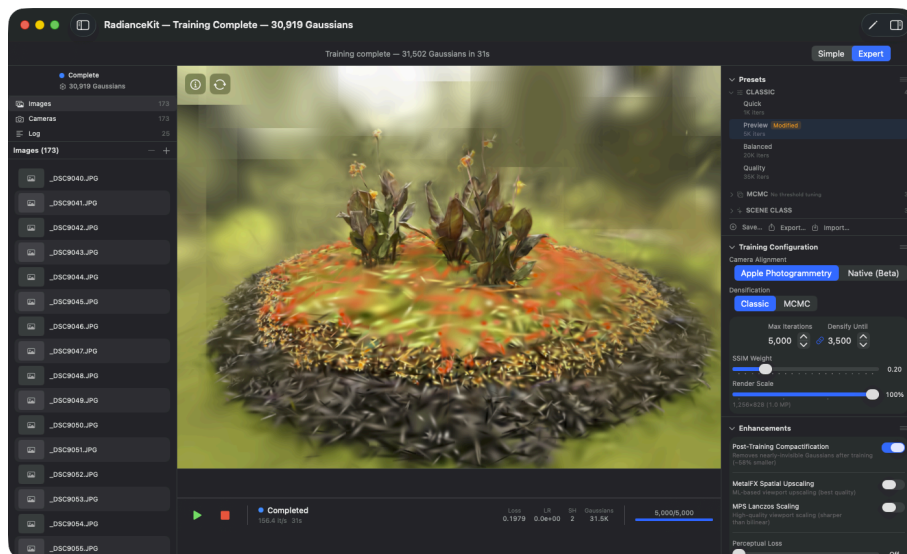
Størrelses-kolonnen er grove pejlemærker for 1 mio. gaussians ved SH-grad 3. Reelle værdier varierer alt efter scenens komprimerbarhed; SH-grad 0 reducerer PLY/glTF med faktor 4.

KAPITEL

Kapitel 9 — SfM-backends



Figur 33: Eksperttilstand med Camera Alignment-vælger i Inspector (Apple Photogrammetry / Native (Beta))



Figur 34: Inspector med Native (Beta) aktiv — Camera Alignment-vælgerens anden mulighed valgt, alle andre træningskonfigurationsparametre uændret

HVAD BILLEDET VISER Camera Alignment-vælgeren i Inspector er en segmenteret kontrol med to muligheder — Apple Photogrammetry (standard for App Store-builds, fuldt sandbox-konform) og Native (Beta) (RadianceKits eget FAST+BRIEF+GLOMAP-pipeline-

backend, udviklet i fase 3.8/3.9, pr. 2026-05). Native (Beta) er orbit-only valideret og hurtigere ved $\geq 1\ 000$ frames end Apple Photogrammetry, men opfylder endnu ikke fase-3-§5-kvalitetsgaten ($\text{finalLoss} \leq 0.0115$) — deraf Beta-mærket. Eksterne SfM-resultater fra Metashape, COLMAP eller anden fotogrammetri-software kan desuden importeres via File-menuen (Q3 COLMAP-tekstformat, Q6 Workspace-import) — vælgeren skifter ikke, men de importerede positioner erstatter SfM-resultatet.

SfM står for **Structure from Motion**. Ud fra en mængde overlappende fotos rekonstruerer softwaren for hvert billede kameraets position og kiggeretning i et fælles 3D-kordinatsystem. Undervejs genereres en grov 3D-punktsky, der initialiserer træningen med Gaussian Splatting. SfM-resultatet er input til selve træningen og er afgørende for den senere billedkvalitet.

RadianceKit tilbyder fem SfM-veje: to backends indbygget i appen (Q1 Apple Photogrammetry, Q4/Q5 Native), to import-stier fra eksterne værktøjer (Q3 COLMAP-tekstformat, Q6 binær Workspace-import) samt Q2 COLMAP-binary, der kun er tilgængelig i developer-builds uden for App Store. Hvilken der er den rette, afhænger af scenetyperen (orbit omkring et objekt, indendørs rum, dronedeflyvning), og af om en ekstern software allerede leverer en rekonstruktion.

I Q1 — Apple Photogrammetry

HVOR

Expert View → Inspector → Træningskonfiguration
→ Camera Alignment-vælger, posten „Apple Photogrammetry“.

TEKNISK

Indkapsler Apples indbyggede fotogrammetri-framework, som oprindeligt blev udviklet til Object Capture. Apple udtrækker internt features med en proprietær pipeline (trin er ikke offentligt dokumenteret), verificerer dem via multi-view-matching og løser bundle-adjustment på Apple Silicon Neural Engine + GPU. Backend'en er fuldt App Store-konform (ingen ekstern binary, Sandbox=true, on-device), men leverer kun kamerapositioner plus en grov punktsky — ingen diagnose-metrikker som sporelængde eller reprojektionsfejl. Skalerer ifølge Apples anbefaling op til et par hundrede billeder. Ved mere end ~500 frames i lineære dronedeflyvninger eller store outdoor-scener er der reproducerbart observeret crashes eller stilstoppende frasortering af enkelte kameraer.

KORT FORTALT

Det er den enkleste vej. Billeder ind, app regner. Fungerer rigtig godt ved klassiske objekt-scanninger — når du går rundt om et møbel eller en skulptur og tager 50–200 fotos. Ved dronedeflyvninger over landskaber eller ved rigtig mange billeder (over 500) bliver Apples metode dog gerne ustabil. Til den slags scener kan du afprøve Native-backend'en (Q4/Q5) eller regne kameraerne i Metashape og indlæse dem via Workspace-importen (Q6).

POWER-USER

Q2 COLMAP-binary — starter det eksterne COLMAP-program som underproces og er derfor **ikke tilgængelig** i App Store-versionen (sandbox). Virker kun i developer-builds uden for App Store. For den kvalitet, COLMAP leverer, findes der i App Store-versionen Workspace-importen (Q3 eller Q6): kør SfM i COLMAP eller Metashape uden for appen, og indlæs resultatet.

Q3 — COLMAP-tekstformat (Metashape / ETH3D) **HVOR**

Menuen „File → Import COLMAP / Metashape Workspace...” (Cmd+⇧+I) ELLER træk-og-slip af en mappe med `sparse/0/cameras.txt`.

 **TEKNISK**

Læser den standardiserede COLMAP-teksteksport — tre tekstfiler `cameras.txt`, `images.txt`, `points3D.txt` i undermappen `sparse/0/` — og konverterer til den interne SfM-resultat-model. Samme formatdefinition som COLMAP-binær-eksporten, bare som ASCII i stedet for binær. Udskrives af Agisoft Metashape, RealityCapture, PolyCam og ETH3D-benchmark'en i præcis dette layout. Parseren deler kamera-model-detektion med binary-parseren (alle gængse modeller: SIMPLE_PINHOLE, PINHOLE, OPENCV, OPENCV_FISHEYE, FULL_OPENCV). Robust over for kommentarlinjer og tomme linjer. Skalerer i tests op til ~1 400 kameraer (ETH3D Tunnel) uden problemer.

 **KORT FORTALT**

Hvis du allerede har arbejdet med Metashape, RealityCapture eller en anden kommerciel foto-3D-software og har eksporteret resultatet — kan du indlæse denne eksport direkte i RadiancKit uden, at appen selv skal regne på ny. Det sparer timers ventetid. Indlæs blot hele mappen via File-menuen eller træk den ind i vinduet.

I Q4 — Native SfM (inkrementel)

HVOR

Expert View → Inspector → Træningskonfiguration → Camera Alignment-vælger, posten „Native (Beta)“. Inkrementel er standardtilstanden for denne backend — der findes ingen separat mapper-vælger i Inspector. Via CLI kan tilstanden sættes eksplicit med `--native-sfm` eller `--sfm-mapper incremental`.

TEKNISK

Egen GPU-accelereret implementering af hele SfM-pipelinen: FAST+BRIEF-features EL-LER SuperPoint+LightGlue via CoreML (med `--coreml-features`), efterfulgt af Hamming-KNN-matching, RANSAC-fundamentalmatrix, track-building, initial-par-udvælgelse, two-view-bootstrap (F→E plus DLT), grådig inkrementel mapper med PnP-registrering og multi-view-triangulering og endelig bundle-adjustment via Schur-reduceret Levenberg-Marquardt med Huber-loss og analytiske jacobianer over Cholesky-løsning. Fuldt App Store-konform: ingen ekstern binary, Sandbox=true. Med R2-collapse-detektoren, der blev leveret i fase 3.10: registrerer appen mindre end 60 % af input-frames eller falder points-per-camera-raten under 13, skiftes der automatisk til den globale mapper (Q5). Empirisk ren på orbit-/turntable-scener; ved mere generelle bevægelser (droneflyvning, indendørs rum med kompleks geometri) er successraten lavere — detektoren fanger dog disse tilfælde. Skalerer pålideligt op til ~200 kameraer, højere med markant længere kørselstid.

KORT FORTALT

Apples styrker (App Store-kompatibel, hurtig til orbits) med yderligere diagnoseværdier. Fungerer særligt godt, når du som ved en Object Capture går rundt om et motiv. Ved mere komplicerede optagelser (droneflyvning eller stue) genkender RadianceKit automatisk, at det ikke lykkes, og hopper over til den globale metode. Mærket „Beta“, fordi den stadig afprøves — standardanbefalingen er fortsat Apple Photogrammetry til simple objekt-scanninger og Workspace-importen (Q3 eller Q6) til krævende outdoor-sæt.

I Q5 — Native SfM (global)

HVOR

Kaldes automatisk, når den inkrementelle mapper (Q4) udløser collapse-detektoren (mindre end 60 % af input-frames registreret eller points-per-camera-rate under 13). Manuelt tvangs-aktiverbar kun via CLI `--sfm-mapper global`. I Inspector er den globale metode ikke tilgængelig via en separat vælger — appen beslutter selv, hvornår den skifter.

TEKNISK

Global variant af den native pipeline. Først feature-ekstraktion + matching som i Q4, derefter relativ pose-estimering for alle verificerede par, efterfulgt af rotation-averaging (synkroniserer alle kamera-rotationer i verdens-koordinatsystemet) og translation-averaging (LSQR-baseret på en matrix-fri sparse-formulering for at undgå integer-overflow ved store kamera-mængder). Skalerer i princippet til ~5 000 kameraer, i praksis kvalitetsforringet over nogle hundrede kameraer — fase-3.8-§5-akzeptanz-gatemålingen på K-1351 gav finalLoss 0.07 i stedet for de tilstræbte 0.0115. Håndteres som „fallback-tier“: kommer i spil, når den inkrementelle mapper degenererer, men kontrolleres ikke selv på ny for kvalitet.

KORT FORTALT

Plan-B-stien for den native engine. Kaldes automatisk, når den hurtigere inkrementelle sti svigter. Leverer et brugbart resultat, men er ved meget store eller svære scener sjældent så præcis som det, du får fra Metashape eller en ekstern COLMAP-installation. Hvis Native bliver din standardarbejdsgang, kan det i sådanne tilfælde betale sig at gå omvejen via Workspace-importen (Q3 eller Q6).

I Q6 — Metashape / COLMAP-tekst-workspace-import (fase Q7)

HVOR

File-menuen → „Import COLMAP / Metashape Workspace...“ (Cmd+⇧+I). Træk-og-slip af en mappe med `sparse/0/cameras.{bin,txt}` og `images/`.

TEKNISK

Genkender automatisk, om en mappe valgt via træk-og-slip eller åbn-panelet svarer til et af de tre COLMAP-workspace-layouts (`sparse/0/`, `sparse/` eller `roden`), og om rekonstruktionen findes som binær (`cameras.bin`) eller tekst (`cameras.txt`). Den binære sti bruger COLMAP-binær-parseren, tekststien ETH3D-loaderen — begge producerer samme SfM-resultat-model, og resten af pipelinen (importér billeder, start MCMC-træning) er agnostisk over for kilden. Billederne åbnes via app-sandbox-bookmark-systemet `security-scoped`, så importen også virker i App Store-versionen. Specifikt tænkt til situationen „Metashape-eksport uden at regne rekonstruktionen om“. Detektion, der nævnes i File-menu-posten, advarer i app-loggen, hvis den valgte mappe ikke er et genkendeligt workspace.

KORT FORTALT

Helt specifikt funktionen for Metashape-brugere. Hvis du har en licens til Metashape eller RealityCapture og har lavet kamera-rekonstruktionen dér, kan du blot trække eksport-mappen ind her og straks starte træningen. Sparrer flere timers regnetid på store scener, fordi RadianceKit så ikke selv skal lave SfM'en.

Hvilken backend hvornår?

Scenarie	Anbefalet backend
Objekt-scan, 50–200 fotos	Q1 Apple Photogrammetry
Stor outdoor / drone / >500 billeder	Q6 Workspace-import (regn i Metashape eller COLMAP, indlæs så)
Metashape/RealityCapture-eksport foreligger	Q6 Import (ingen SfM nødvendig)
ETH3D / akademisk COLMAP-tekstsæt	Q3 COLMAP-tekst-import
Strengt App Store-konform + orbit-scene	Q4 Native inkrementel
Q4 fejler	Q5 Native global (automatisk)
ETH3D-benchmark-data	Q3 (autotest precomputed)

Hurtig sammenligning

Back-end	App Store	Sand-box	Ekstern binary	Bedst til	Max ~kameraer
Q1 Apple PG	✓	✓	—	Orbit-objekt	~300
Q2 COL-MAP Binary	✗ (kun developer-build)	—	colmap/glomap	Outdoor stor	~5 000
Q3 COL-MAP-tekst-import	✓	✓	—	Bench-rigs	~1 500
Q4 Native inkrementel	✓	✓	—	Orbit-objekt	~200
Q5 Native global	✓	✓	—	Q4-fallback	~1 351
Q6 Workspace-import	✓	✓	—	Metashape-genbrug	pr. kilde

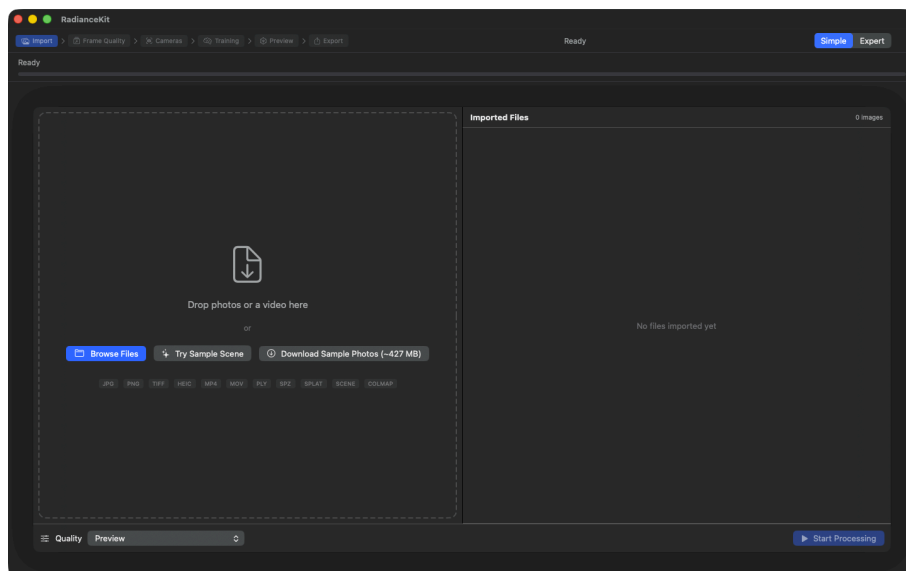
KAPITEL

Kapitel 10 — Begyndertilstand

Begyndertilstanden (engelsk Simple Mode, Cmd+1) er den guidede arbejdsgang for alle, der for første gang rekonstruerer en 3D-Gaussian-Splatting-scene. I stedet for at vise en sidebar fyldt med Inspector-felter fører appen igennem fire trin: først importeres billeder eller en video, og en kvalitets-preset vælges, så kører behandlingen (SfM + træning), bagefter kan den færdige scene vurderes i en 3D-forhåndsvisning, og til sidst eksporteres til det ønskede format. En smal fremskridtsbjælke øverst i vinduet viser til enhver tid, hvilket trin du befinder dig på.

Sammenlignet med Expert Mode (Cmd+2), som viser alle betjeningsfelter samtidig, skjuler begyndertilstanden ubrugte muligheder, giver validerings-advarsler ved for få eller dårlige billeder og tilbyder på hvert trin kun de knapper, der er meningsfulde i den aktuelle tilstand. Du kan til enhver tid skifte mellem begynder- og Expert Mode (Cmd+1 / Cmd+2), hele tilstanden — importerede billeder, valgt preset, igangværende træning, færdig punktsky — bevares og er straks tilgængelig i den anden tilstand.

Z1 — Import (vælg billeder & preset)



Figur 35: Simple Mode trin 1 — tom drop-zone for billed-import, crumb-trail øverst (Import → Frame Quality → Cameras → Training → Preview → Export), format-pills JPG/PNG/TIFF/HEIC/MP4/MOV/PLY/SPZ/SPLAT/SCENE/COLMAP

HVAD BILLEDET VISER Crumb-trail (Import aktiv) viser den fire-trins-arbejdsgang. Venstre drop-zone med tre CTAs: „Browse Files“ (NSOpenPanel), „Try Sample Scene“ (bundled

demo), „Download Sample Photos (~427 MB)“ (Mip-NeRF360 flowers subset). Format-pills herunder lister alle accepterede filtyper. Til højre „Imported Files“ med counter „0 images“ og empty-state „No files imported yet“. Nederst quality-vælger (default: Preview) og „Start Processing“ (deaktiveret, så længe ingen billeder er der).

Første trin består i at give appen billedmateriale. Via træk-og-slip i det store, stiplede felt i midten, via „Browse Files“-knappen eller via klik på den medleverede sample-scene. Til højre optræder en liste over alle importerede billeder med opløsning og filstørrelse; nederst i den svævende værktøjslinje vælger du kvalitets-presetet og starter pipeline med „Start Processing“. Validerings-advarsler (rød ved < 3 eller < 10 billeder, orange ved 10–19) viser, om appen forventer en meningsfuld rekonstruktion eller ej.

C-01 ProgressIndicator (trin-visning)

HVOR

Øverst over arbejdsgangen, altid synlig.

TEKNISK

Viser en horisontal fremskridtsbjælke over hele pipeline (frame-quality → SfM → træning) med stage-allokering: frame-quality optager 0–5 % (fase 3.11, meget kort), SfM optager 0–30 % af bjælken, træning 30–100 %. Ved siden status-tekst og fase-navngivet procent-visning („SfM 41 %“, „Training 12 500/20 000“), så brugere ikke læser det tilsyneladende tilbageskridt „41 % SfM → 25 % træning“ som fejl — bjælken viser hele pipeline-fremskridtet, ikke sub-stage. ETA-beregning starter, så snart der er målt nok trænings-tempo (typisk efter de første 100 iterationer). Samme visning bruges også i Expert Mode over Inspektoren.

KORT FORTALT

Den smalle bjælke helt oppe er dit kort gennem arbejdsgangen. Den fortæller dig ikke kun, hvad appen laver lige nu (justerer kameraer, træning kører, ...), men også hvor langt den samlet allerede er. Opdelingen er bevidst sådan, at kamera-beregningen optager den første tredjedel af bjælken, og selve træningen de to bageste tredjedele — ellers ville fremskridtet efter SfM virke som om det pludselig var tilbage ved nul. Du kan altså læne dig tilbage, et blik på bjælken rækker til at se den grove etape. Teksten ved siden af fortæller dig, om du lige er i SfM-trinet (f.eks. „SfM 41 %“) eller i træningen (f.eks. „Training 12 500/20 000“), så talene ikke virker forvirrende. Hvis du ikke får vist ETA, er træningen bare endnu for ung — appen skønner først, så snart den har målt nok tempo.

C-03 DropZoneView (træk-og-slip-område)

HVOR

Venstre side af import-trinet, stort stiplede rektangel med symbol. Viser i begyndertilstand med labellet „Drop photos or a video here“.

TEKNISK

Drop-område, der lader symbolet kort hoppe og farver baggrunden, så snart drag-items svæver over feltet. Accepterer JPG, PNG, TIFF, HEIC, MP4, MOV, PLY, SPZ, .splat, .radiance-scene-bundles og mapper. Drop-routing efter type: billeder samles og overgives sorteret, videoer udløser frame-sampling-stien, splat-filer åbner direkte forhåndsvisningen, scene-bundles indlæses. Mapper enumereres, og alle indeholdte billeder importeres. Security-scoped bookmarks til sandbox-konform adgang optages og frigives korrekt. Ikke-understøttede endelser vises som advarselsbanner i 5 sekunder.

KORT FORTALT

Det store stiplede felt er hovedbetjeningen i første trin. Træk bare fotos eller en video derind, eller en hel mappe — appen tager alt, hvad den kender, og ignorerer resten. Når feltet bliver blå, og symbolet kort hopper, har appen genkendt drag'en. Slip, og importen starter med det samme: billeder vandrer i listen til højre, videoer udløser automatisk frame-sampling-trinet, og allerede trænedede .ply / .spz / .splat -filer åbner direkte forhåndsvisningen. Hvis et format slet ikke passer (f.eks. PDF eller BMP), optræder et kort hint øverst — appen sluger ikke ukendt materiale i stilhed.

C-05 Browse Files-knap

HVOR

Inden i drop-zonen, prominent knap.

TEKNISK

Knap, der åbner macOS-fildialogen med multivalg og filtyperne JPG, PNG, TIFF, MP4, MOV, mapper samt app'ens eget scene-format. Resultat-URL'er er security-scoped og videreføres gennem samme import-stier som træk-og-slip. Hvis brugeren vælger en mappe, enumereres den rekursivt for billeder.

KORT FORTALT

Hvis træk-og-slip er ubekvem for dig, så klik bare denne knap og navigér i macOS-fildialogen til dine fotos. Du kan vælge flere filer samtidig (Cmd-klik på de enkelte billeder) eller vælge en hel mappe — appen gennemser så mappen rekursivt for alle understøttede billedtyper. Det er særligt praktisk, hvis dine optagelser ligger indkapslet i undermapper (f.eks. „shoot-day1“, „shoot-day2“) — et klik på hovedmappen rækker. Funktionelt gør knappen præcis det samme som træk-og-slip; vælg bare den vej, der er mest bekvem for dig.

C-06 Try Sample Scene-knap **HVOR**

Inden i drop-zonen, kun synlig hvis app-bundlen indeholder sample-scenen, og der endnu ikke er importeret billeder/splats.

 **TEKNISK**

Optræder kun, hvis (a) en `sample-scene.splat`, `.spz` eller `.ply` findes i app-bundlen OG (b) endnu ingen billeder/videoer er importeret, og endnu ingen punktsky findes. Ved klik indlæses den færdige punktsky (foretrukket det mindste format — `.splat` ~3 MB, `.spz` ~1.4 MB, fallback `.ply`) og sætter efter 400 ms hardkodede kamera-værdier fra den originale metadata af blomster-scenen til et æstetisk meningsfuldt indgangs-perspektiv.

 **KORT FORTALT**

Hvis du starter appen for første gang og bare vil se, hvad der kommer ud til sidst — klik her. Det åbner en færdig trænet blomster-scene, som du straks kan dreje og eksportere, uden at appen skal regne. Kameraet er forindstillet til et æstetisk meningsfuldt indgangsperspektiv, så du straks ser noget pænt. Perfekt til at prøve 3D-styringen og eksport-trinet risikofrit, før du går løs på egne optagelser. Så snart du importerer egne billeder, forsvinder knappen automatisk — den vises kun, så længe projektet er helt tomt.

C-07 Download Sample Photos-knap **HVOR**

Inden i drop-zonen, ved siden af „Try Sample Scene“; samme synligheds-betingelser.

 **TEKNISK**

Udløser et download (repo github.com/bkindler/radiancekit-sample-photos), der indlæser ca. 427 MB med 960 fuld-opløsnings-frames og fodrer appen. Under downloadet er knappen deaktiveret. Fremskridtet vises i den øvre progress-bar som „Downloading X %“ i en egen stage, fordi denne stage beholder sin egen 0–100 %-skala og ikke overlapper det senere SfM-stage.

 **KORT FORTALT**

Nøjagtigt som sample-scenen, kun med udgangsfotos i stedet for med det færdige resultat. Sådan kan du selv køre hele pipeline igennem en gang og se, hvor lang tid SfM og træning faktisk tager på din Mac. Downloadet er stort (omtrent en halv DVD = 427 MB), men sker kun en gang — bagefter er fotos lokale, og du kan genstarte pipeline vilkårligt mange gange med forskellige presets. Mens downloadet kører, viser den øvre fremskridtsbjælke den aktuelle download-status i procent, så du kan vurdere, hvornår det går i gang. Tip: tag helst et hurtigt WLAN eller kablet net dertil — de 427 MB trækker ellers ud.

C-09 Quality Presets-vælger **HVOR**

Svævende nedre værktøjslinje af import-overlayet, til venstre for start-knappen.

 **TEKNISK**

Betjeningselement med label „Quality“ grupperer de tilgængelige presets efter kategori (Classic / MCMC / Custom). Indbyggede presets grupperes efter kategori; afsnits-headerne er hardkodede. Custom-presets kun synlige, hvis nogen findes. Locked-state: presets, der ikke er i free-listen (Quick + Preview), får et „“-suffiks på navnet, hvis brugeren ikke har købt; ved valg springer vælgeren tilbage på Preview og åbner automatisk purchase-sheetet. Ved valg anvendes presetet, hvilket erstatter hele trænings-konfigurationen.

 **KORT FORTALT**

Her vælger du, hvor præcist og hvor længe appen skal regne. „Quick“ og „Preview“ kan bruges uden køb og leverer et første resultat på få minutter — ideelt til at teste, om dine billeder overhovedet er meningsfulde. „Balanced“ og „Quality“ kræver fuldversionen og leverer markant rene modeller, men tager til gengæld timer i stedet for minutter. MCMC er en anden strategi, der klarer sig med færre gauss-plats — godt, hvis du senere vil eksportere modellen kompakt eller lægge på nettet. Premium-presets genkender du på det lille låse-symbol ved navnet; trykker du på en uden licens, springer vælgeren tilbage på Preview, og købs-sheetet åbnes automatisk. Tommelfingerregel: start altid med Preview, kig på resultatet, og beslut så, om det er værd at lave en længere kørsel.

C-10 Start Processing-knap **HVOR**

Svævende nedre værktøjslinje af import-overlayet, til højre for preset-vælgeren.

 **TEKNISK**

Knap, der forbliver grå, så længe hverken billeder eller en video er importeret. Ved klik startes pipelinen, og stage-maskinen skifter til rækkefølgen frame-quality → SfM → træning. Knappen selv har ingen yderligere status; en kørende behandling optræder i stedet som separat behandlingsskærm.

 **KORT FORTALT**

„Sæt i gang“-knappen. Så længe den er grå, mangler der stadig input-billeder eller en video. Så snart du har trukket fotos ind, bliver den aktiv, og du klikker den for at starte SfM og træning efter hinanden. Derfra overtager appen hele forløbet, og du lander automatisk på behandlingsskærmen (Z2). Du behøver ikke klikke yderligere — først efter trænings- slut skifter appen tilbage til forhåndsvisningen (Z3). Hvis du ombestemmer dig, kan du også derefter til enhver tid afbryde via Cancel.

C-11 Video Sampling-slides**HVOR**

Højre billedliste, kun synlig hvis en video (i stedet for billeder) er importeret.

**TEKNISK**

Skydeknapp 0.5 fps – 30 fps i 0.5-skridt. Ved ændring opdateres frame-densiteten, og desuden beregnes antallet af mål-frames (mindst 10) ud fra densitet og videolængde. Skydeknappen ligger uden for billedlisten, fordi liste-elementer ville blokere muse-events fra skydeknapper. Under skydeknappen står de beregnede mål-frames („247 frames“) og videolængden („1m23s video“). Tooltip advarer: „Doubling the density doubles the number of frames and increases SfM time by ~100 %.“

**KORT FORTALT**

Hvis du har importeret en video i stedet for fotos, beslutter denne skydeknapp, hvor mange enkelt-billeder appen skal trække ud af videoen. Flere billeder = bedre kvalitet, men lineært mere optagelsetid. Til en 30-sekunders orbitvideo er 5 fps (150 billeder) en god start; ved 1-minuts-optagelser rækker 3 fps ofte fuldt ud. Under regulatoren viser appen live, hvor mange frames der kommer ud ved den aktuelle indstilling — så ser du straks, om du rammer det fornuftige område på ca. 100–300 billeder. Hvis resultatet bliver dårligt, så træk regulatoren til højre og prøv igen; fordobling af frame-raten fordobler dog også groft SfM-varigheden.

C-12 Clear All-knap**HVOR**

Højre billedliste, nederst til højre; kun synlig hvis billeder er importeret.

**TEKNISK**

Rød knap. Klik åbner en bekræftelsesdialog med titel „Clear all imported files?“ og besked „N images will be removed.“. Bekræftelse tømmer alle importerede billeder/videoer, staging-mapper, punktskyen, trænings-status, SfM-resultatet og alle caches; stagen springer tilbage til import. Ved Cancel bevares alt. Dialogen er konfigureret som ikke-destruktiv default-sti (destruktiv knap markeret rød).

**KORT FORTALT**

Hvis du vil starte helt forfra, så klik her. Bekræftelses-spørgsmålet optræder, fordi sletningen kasserer alle aktuelle importer inklusive evt. allerede beregnede kameraer og trænings-resultater — du kan ikke fortryde det. Meningsfuldt, hvis du vil udskifte det valgte billedmateriale helt eller komme af med et gammelt projekt, før du starter et nyt. Bemærk: at tage et enkelt billede ud sker via listen til højre (se næste punkt), ikke via denne knap. Dine filer på disken slettes ikke — appen glemmer kun sine referencer.

C-13 File List ForEach (enkelt-billede-fjern)**HVOR**

Højre billedliste, hver post.

**TEKNISK**

Liste over de importerede billeder med swipe-to-delete. Pr. billede en linje med ikon, filnavn, opløsning („1920 × 1080“) og filstørrelse (formateret KB/MB). Opløsningen kommer fra en metadata-cache, der asynkront fyldes fra billed-headerne, så brugerfladen ikke blokeres. Slet-handlingen tilbyder macOS-typisk swipe-delete (trackpad-swipe til venstre på en linje) samt tastatur-delete ved markeret linje. Bemærk: den udvidede image-delete-sti med eksPLICIT minus-knap, backspace og Cmd-Z til at fortryde blev *kun i Expert Mode* tilføjet i Project Navigator — i begyndertilstand bliver det ved swipe-delete.

KORT FORTALT

Listen til højre viser hvert importeret billede med opløsning og filstørrelse — praktisk for med ét blik at se, om du har blandet højopløsningsmateriale med lavopløsningsmateriale. For at tage et enkelt billede ud, så swipe det til venstre med to fingre på trackpadet — som i iOS Mail — eller vælg det og tryk Delete. Appen sletter ikke selve filen; den tager den blot ud af det aktuelle projekt. Hvis du har brug for en rigtig minus-knap eller Cmd-Z-fortryd, så skift til Expert Mode (Cmd+2), der findes det i Project Navigator. I begyndertilstanden bliver det bevidst ved det enkle swipe-pattern.

C-15 Validation Warnings (3-trins-tier)**HVOR**

Under billedlisten, over Clear-All-knappen.

**TEKNISK**

Tre efter hinanden følgende tærskler baseret på antallet af importerede billeder (kun aktiv, hvis billeder findes, og ingen video): - < 3 billeder: rødt banner (red octagon), tekst „At least 3 images are required. Camera alignment cannot be computed from fewer images.“ - 3–9 billeder: rødt banner, tekst „With fewer than 10 images, SfM often fails and the trained scene tends to overfit [...]. 15–20 images minimum recommended; 30+ for object captures.“ - 10–19 billeder: orangefarvet banner (warning triangle), tekst „Workable, but quality usually improves with 20+ images and good coverage around the scene.“ Fra 20 billeder forsvinder banneret. Tærskelværdier er hardkodet og baseret på empiriske 560+-trænings-eksperimenter.

KORT FORTALT

Appen kigger på, hvor mange billeder du har importeret, og giver dig en farvet vurdering. Rødt betyder: det bliver med høj sandsynlighed ikke til noget — enten kan SfM ikke beregne kameraler, eller træningen overfitter på for lidt materiale. Orange betyder: kan måske lykkes, men regner ikke med top-kvalitet, fordi algoritmen finder lidt overlap mellem billederne. Intet banner betyder: gode forudsætninger, du har nok materiale. Hvis du vil have virkeligt rene modeller, så sigt efter mindst 30–50 jævnt fordelte optagelser omkring dit motiv — gerne markant flere ved udenørs-scener eller store rum. Du kan starte trods advarsel, men bliv ikke overrasket, hvis SfM afbryder uden kommentar, eller modellen ser hullet ud.

C-16 COLMAP Workspace Detection**HVOR**

Ved drop af en mappe — ingen synlig knap, men detektions-logik.

**TEKNISK**

Ved drop af en mappe tjekkes det, om den indeholder et af de tre kanoniske workspace-layouts: `sparse/0/cameras.bin`, `sparse/cameras.bin` eller direkte `cameras.bin` i roden. Hvis det er tilfældet, afbrydes standard-billed-enumerationen, og i stedet åbnes en modal alert, der spørger brugeren, om den eksisterende rekonstruktion skal bruges, eller billederne skal sendes igennem Apple Photogrammetry på ny. Samme sti også for tekst-format-workspaces (`cameras.txt`) og ETH3D-eksporter. Se kapitel 9 backend Q6 for detaljer. Virker i begyndertilstand præcis som i Expert Mode.

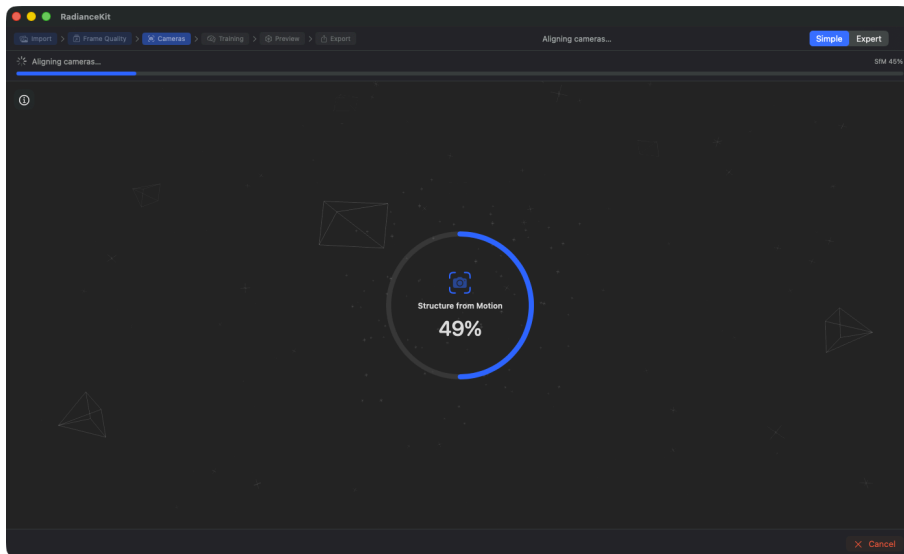
**KORT FORTALT**

Hvis du allerede har arbejdet med Metashape, RealityCapture eller COLMAP og har ladet kamera-beregningen køre der, kan du bare trække eksport-mappen ind her. RadianceKit genkender automatisk på indholdet, at det er et COLMAP-workspace (det tjekker for `sparse/0/`, `cameras.bin` osv.) og spørger dig, om det skal overtage den færdige beregning eller selv regne på ny. At overtage sparer timers ventetid på store scener, fordi SfM springes helt over — træningen starter straks. Også tekst-format-workspaces (`cameras.txt`) og ETH3D-eksporter genkendes. Denne funktion er tilgængelig i begyndertilstand ligesom i Expert Mode; flere detaljer står i kapitel 9 under backend Q6.

Hvornår til næste trin?

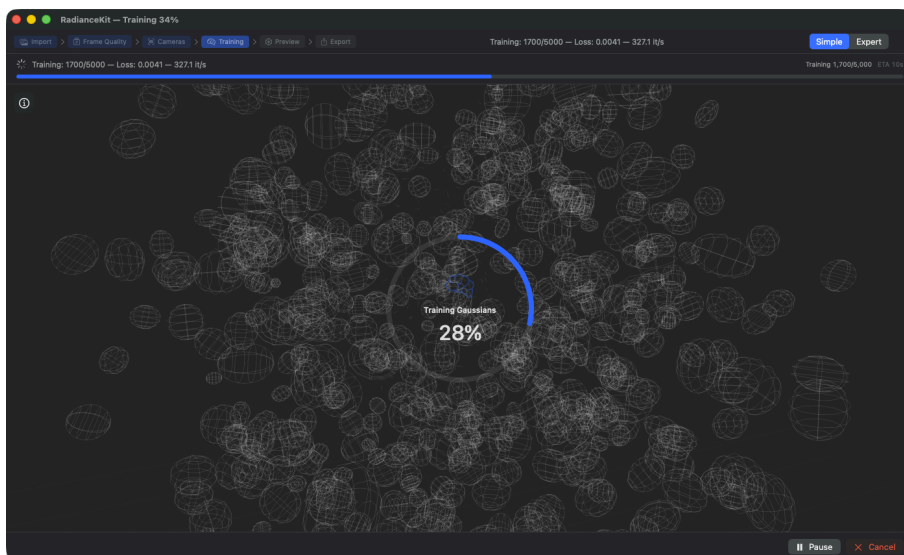
Du kan klikke Start Processing, så snart (a) mindst ét billede eller en video er importeret, og (b) validation-banneret er orange eller forsvundet. Ved rødt banner lader appen dig dog starte alligevel, men du kan med høj sandsynlighed afbryde behandlingen igen straks. Anbefales: mindst 20 billeder, skarpe, med tydeligt overlap mellem på hinanden følgende optagelser, alle fra omtrent samme distance til motivet. Vælg før start en preset, der passer til dit tidsbudget — ved 30 billeder og Quick-preset er du igennem på få minutter, ved Quality varer det snarere 1–2 timer.

Z2 — Behandling (SfM + træning)



Figur 36: Z2 SfM-fase — stage-ikon „Structure from Motion“ med 41 % i den store cirkel, øvre statusbjælke ved „SfM 25 %“, Cancel- knap nederst til højre

SfM-fase (kameraer justeres): Stor fremskridtscirkel viser sub-stage-fremskridt (her 41 % af den kørende Apple- Photogrammetry-session). Status-tekst „Aligning cameras...“ øverst til venstre. Crumb-trail markerer „Cameras“ som aktivt trin. Øvre statusbjælke viser pipeline-samlet-fremskridt (25 %) — SfM optager den første halvdel af bjælken. Svævende wireframe- kameraer i baggrunden antyder, at positioner skønnes.



Figur 37: Z2 træningsfase — stage-ikon „Training Gaussians“ med 6 %, live-metrikker øverst (Training: 400/5000 — Loss: 0.1642 — 138.7 it/s), ETA 33 s, Pause/Cancel nederst

Træningsfase (gaussians optimeres): Sub-stage-ikon skifter til „Training Gaussians“, procent tæller iterationer ud fra valgt preset (her 400 / 5 000 for Preview-preset = 8 % af stagen). Live-metrik-linjen viser loss-værdi (0.1642), iterationer pr. sekund (138.7 it/s) og ETA (33 s). Pipeline-samlet-fremskridt klatrer fra 50 % til 100 % under denne

fase. Pause-knappen (i stedet for Cancel-only i SfM-fase) tillader resume senere; Cancel kasserer trænings-resultatet og vender tilbage til Z1.

Så snart pipelineen kører, fader appen import-overlayet ud og viser en fuldskræms-behandlingsskærm. I midten kører en stor fremskridtscirkel (220 × 220 pixel) med stage-ikon, status-tekst og procent-tal; i baggrunden visualiserer en diskret splat-animation symbolsk den kørende beregning. Øverst til venstre kan et info-panel indkobles, der viser live-metrikker fra træning og SfM. Nederst er der pause/resume, Cancel og i fejltilfælde en retry-knap.

C-18 SplatTrainingView (baggrunds-animation)

HVOR

Fuldskræms-baggrund bag progress-cirklen, fades ud ved afbrydelse eller fejl.

TEKNISK

Dekorativ animation, der afhængigt af pipeline-fremskridtet (0..1) renderer et stigende antal små animerede splat-partikler. Kilden er en beregnet fremskridtsværdi, der mapper SfM-faser til 0–0.2 og træning til 0.2–1.0 (frame-quality til 0–0.05). Sådan „bygger“ splats sig synligt op, mens træningen kører. Udelukkende dekorativt — visningen viser ingen ægte mellemresultater af den aktuelle træning (det ville være live-preview i Expert Mode). Ved afbrydelse eller fejl fader den ud, og kun status-cirklen forbliver synlig.

KORT FORTALT

I baggrunden kører en lille animation af dansende punkter, så skærmen ikke virker så tom under beregningen. Det er ikke din ægte 3D-model — den ser du først efter træningen i trin Z3. Animationen har dog samme tonalitet, så du ud fra det omtrentlige fortætningsniveau kan aflæse, hvor langt træningen er nået. I starten ses kun få punkter, mod slutningen fyldes baggrunden tydeligt tættere — en sød visuel indikator ud over procentvisningen i cirklen. Hvis animationen generer dig (f.eks. fordi du vil arbejde ved siden af i baggrunden), kan du skifte til Expert Mode, hvor den falder væk.

C-19 Stor progress-cirkel

I midten af behandlingsskærmen, 220 × 220 pixel.



To over hinanden rendrede ringe: udvendigt en dæmpet track-ring, indvendigt en udfyldt frem-skriftsring med accent- eller rød-stroke (rød ved fejl). Inden i cirklen et stage-ikon (hjerne for træning, kamera for SfM, film for video-frame-ekstraktion, sparkles for frame-quality), stage-titel og det live-animerede procent-tal i 32-punkts-rounded-font. Ikonet pulserer blødt, så længe behandlingen er aktiv. Visningen interpolerer på en 30-Hz-timer blødt mod det aktuelle ægte fremskridt — med konstant kryb (0.0003/frame) plus proportional andel (4 % af gappet) og et soft-ceiling, der sættes til 80 % af næste forventede milestone (for SfM fra en hardkodet milestone-tabel). Sådan virker fremskridtet flydende, selv hvis de ægte SfM-updates kun ankommer hvert par sekunder.

 KORT FORTALT

Den store cirkel i midten er din hoved-visning, mens appen regner. Den fyldes blødt, selv når de ægte beregnings-updates kun kommer hvert par sekunder — det giver dig følelsen af, at der sker noget, i stedet for at stirre minutter på en fastfrossen procent. Symbolet i midten skifter alt efter, om der lige nu ekstraheres frames (film-ikon), kameraer justeres (kamera-ikon) eller gaussians trænes (hjerne-ikon). Procent-tallet refererer til det aktuelle deltrin — den samlede pipeline ser du i den smalle bjælke helt oppe. Ved en fejl farves ringen rød i stedet for blå, og ikonet pulserer ikke længere, så du straks bemærker, at noget er gået galt.

C-22 Info-knap (vis metrikker)

Øverst til venstre på behandlingsskærmen, 32 × 32 pixel.



Simpel knap med material-baggrund. Slår info-panelet til eller fra. Ikonet skifter mellem info-cirkel-outline og info-cirkel-fyldt, når aktiv. Blød fade-in-animation. Tooltip „Show detailed processing metrics“.

 KORT FORTALT

Som standard er skærmen bevidst ryddelig — kun den store fremskridts-cirke, mere ser du ikke i første omgang. Hvis du som teknisk interesseret bruger vil vide mere præcist, hvad der sker (hvilken iteration, hvor høj loss, hvor mange gaussians), så klik på i-symbolet øverst til venstre. Et lille panel folder ud nederst og viser alle live-værdier. Endnu et klik fader det ud igen. Indstillingen er ikke persistent — ved hvert nyt trænings-forløb er panelet først skjult igen, hvilket bevidst er sådan for ikke at skræmme begyndere.

C-23 Info-panel (live-metrikker) HVOR

Nederst til venstre på behandlingsskærmen, kun synlig hvis `showProcessingInfo == true`.

 TEKNISK

Tospaltet panel med ultra-thin-material-baggrund. Venstre kolonne: stage-specifikke info-linjer — for SfM status-tekst og procent; for træning iteration, kombineret loss, L1-loss, D-SSIM-loss, gaussian-count (orange farvet), speed (it/s), elapsed-time, beregnet ETA, SH-degree og learning-rate. Højre kolonne: status-tekst, time-info-string, inline loss-chart (se C-28) og et discoverability-nudge (se C-32). Alle værdier læses fra trænings-status, der opdateres ved hver trænings-tick.

 KORT FORTALT

Info-panelet viser alle live-værdier, der i Expert Mode ville stå permanent i Inspector-sidebaren: aktuel iteration, loss-værdi (mindre = bedre), antal gaussians, hastighed, estimeret resttid, SH-degree og learning-rate. På højre side kører desuden en lille loss-kurve med, som med ét blik fortæller dig, om træningen kører i den rigtige retning. Hvis træningen virker træg, hjælper et blik her — en loss, der ikke længere falder, eller en ETA, der ikke længere synker, indikerer problemer. Hvis loss'en eksploderer (pludselig bliver gigantisk) eller viser NaN, er træningen blevet ustabil, og en Cancel + Retry eller skift til en anden preset er meningsfuldt.

C-25 Pause/Resume-knap HVOR

Nedre navigationslinje, kun synlig under træningsstagen (IKKE under SfM), og så længe behandlingen kører.

 TEKNISK

Bordered knap. Kalder afhængigt af status pause eller resume. Label skifter mellem „Pause“ (med pause-ikon) og „Resume“ (play-ikon). Under SfM-trinet vises knappen ikke, fordi Apple Photogrammetry ikke kender pause-semantik. Pause-tilstanden bevarer iteration, gaussian-status og optimizer-momentum fuldstændigt — resume fortsætter, hvor det blev pauset.

 KORT FORTALT

Mens træningen kører, kan du til enhver tid holde den tilbage og fortsætte senere. Meningsfuldt, hvis du i mellemtiden vil lave noget andet på Mac'en, der kræver meget GPU — f.eks. video-redigering, spil-test eller en render-eksport fra en anden app. Klik Pause, lav din ting, klik Resume, træningen kører videre præcis, hvor den var. Iterations-tæller, gaussian-antal og optimizer-momentum bevares fuldstændigt, pause-state koster dig intet i kvalitet. Under SfM-fasen er pause ikke tilgængelig — Apple Photogrammetry kender ikke til en pause-funktion, der må du i nødstilfælde arbejde med Cancel.

C-26 Cancel-knap **HVOR**

Nedre navigationslinje, synlig mens behandlingen kører (SfM eller træning).

 **TEKNISK**

Rød bordered-knap. Åbner en bekræftelsesdialog med titel „Stop and discard progress?“, knapper „Discard Progress“ (destruktiv) og „Keep Running“ (Cancel). Ved bekræftelse sættes cancel-flaget, trænings-tasken afsluttes, SfM-underprocessen om nødvendigt afsluttes, og en summary-linje med afbryd-status skrives i JSONL-loggen. I modsætning til pause kasseres trænings-buffer og status.

 **KORT FORTALT**

Afbryd-knappen. I modsætning til pause er det endeligt — hvis du derefter vil starte på ny, kører behandlingen forfra, alle allerede træned iterationer er tabt. Meningsfuldt, hvis du har valgt forkert preset, træningen kører alt for langsomt, eller appen åbenlyst producerer skidt-resultater, og du ikke vil vente. Før den faktiske afbrydelse spørger appen i en bekræftelsesdialog en gang til, så du ikke ved et uheld mister timers regnetid. Hvis du kun kort vil afbryde, så tag hellere pause.

C-27 Retry-knap **HVOR**

Nedre navigationslinje, synlig hvis pipeline er fejlet (SfM-status starter med „SfM failed“, eller træningen er i fejl-tilstand).

 **TEKNISK**

Accent-knap. Starter hele pipeline forfra. Før start tjekkes, om der stadig findes importerede billeder/videoer. Tidligere fejl-logs bevares i JSONL-mappen; en ny run skriver en ny log-fil med aktuel timestamp.

 **KORT FORTALT**

Hvis SfM eller træning afbryder med en fejlmeddelelse, kan du prøve igen her. Nogle gange hjælper det, fordi mange trin (RANSAC, densification) har tilfældighedsdele, og et andet forsøg kan lykkes, hvor det første mislykkedes. Hele pipeline kører så igen forfra — SfM og træning, i en frisk JSONL-logfil. Hvis også andet forsøg fejler, er input-billederne som regel problemet (for få, for lidt overlap, bevægelses-uskarphed, dårligt lys); gå så tilbage med Back og udskift dit materiale. Tip: kig samtidig i training-loggene (Help → Open Training Logs), der står mere detaljeret, hvor det konkret gik galt.

C-28 Inline Loss Chart **HVOR**

I info-panelet, højre kolonne, kun synlig under træning med ikke-tom forløbs-historik.

 **TEKNISK**

Kompakt tegne-område (40 pixel højt), tegner loss-history som 1-pixel-linje i accent-farve. Data filtreres til finite-værdier (NaN-beskyttelse for ustabile træninger). Min/max beregnes over hele historikken — charten auto-zoomer altså på værdiområdet. Den sidste loss-værdi står øverst til højre over charten. Selve historikken bygges op i app-tilstanden ved hvert trænings-tick (typisk hver 100 iterationer).

 **KORT FORTALT**

En lille loss-kurve, der med ét blik viser dig, om træningen „konvergerer“ (linje falder mod højre), eller om den hænger fast eller eksploderer (linje flad eller stigende). Ved en sund træning falder linjen stejlt i starten og fladegør så — det er det forventede forløb, lignende en halveringskurve. Charten zoomer automatisk på det aktuelle værdiområde, så også små forbedringer i slutningen af træningen forbliver synlige. Hvis linjen pludselig skyder opad eller fryser, er det et godt signal om, at noget går galt — enten er materialet problematisk, eller en anden preset ville være bedre egnet. Charten finder du i info-panelet, som du folder ud øverst til venstre med i-symbolet.

C-32 Discoverability Nudge (Expert-Mode-hint) **HVOR**

I info-panelet, højre kolonne nederst, kun synlig under træning OG i begyndertilstand.

 **TEKNISK**

Lille linje med øjne-ikon og caption-tekst „Switch to Expert Mode (⌘2) for live splat preview“, i tilbageholdt tone og 10-punkts-skrift. Intet interaktivt element, kun hint. Reagerer ikke på klik — brugeren skal faktisk trykke Cmd+2 eller klikke menuen Mode → Expert Mode.

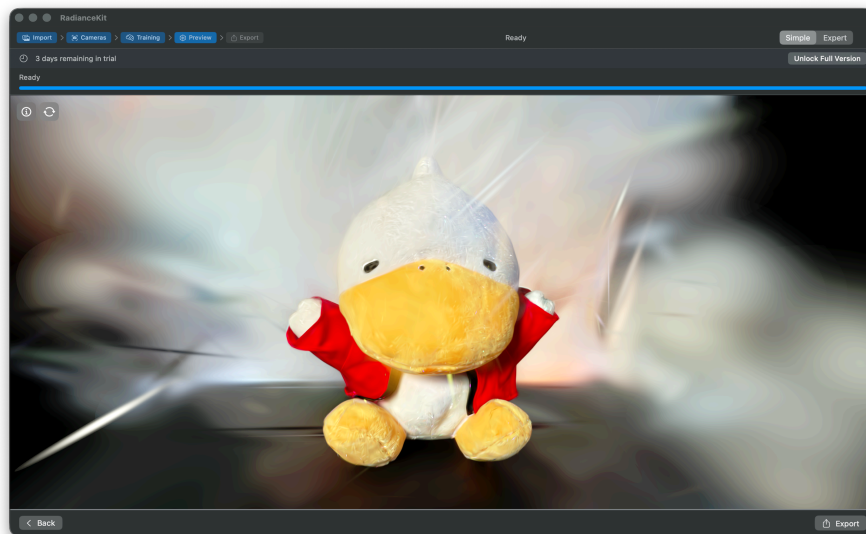
 **KORT FORTALT**

Et diskret hint om, at i Expert Mode er den aktuelle mellemversion af din 3D-model live at se i viewporten under træningen. I begyndertilstand er det bevidst skjult for at holde brugerfladen rolig — men mange brugere ved slet ikke, at funktionen findes, så vi peger blidt på det her. Tryk Cmd+2, og træningen kører videre i baggrunden, mens du kan se på, hvordan din model bygger sig op for dine øjne. Det er også et godt værktøj til allerede efter få tusinde iterationer at vurdere, om resultatet bliver til noget, eller om du hellere afbryder og starter på ny. Cmd+1 bringer dig til enhver tid tilbage til begynder-visningen.

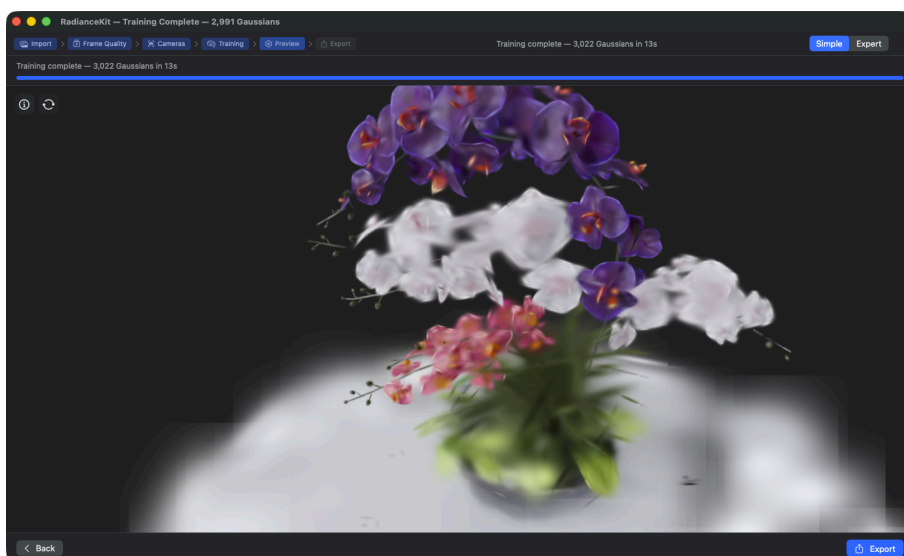
Hvornår til næste trin?

Appen skifter automatisk til Z3 (forhåndsvisning), så snart træningen er afsluttet med succes — du behøver ikke klikke. Den nedre navigationslinje skifter så fra Pause/Cancel til en Back-knap (tilbage til import) og en Export-knap (frem til eksport). I fejltilfælde (rød fejlmeddelelse, stage-ikon er X) optræder Retry i stedet, og du må beslutte, om du starter igen eller går tilbage til import med Back for at ændre billedmateriale.

Z3 — Forhåndsvisning (drej 3D-modellen)



Figur 38: Simple Mode forhåndsvisnings-trin med 3D-viewer



Figur 39: Z3 forhåndsvisning efter trænings-afslutning — Bjoerns Blender-buket rekonstrueret, header viser „Training complete — 3,022 Gaussians in 13s“, Back- og Export-knapper nederst

HVAD BILLEDET VISER Crumb-trail markerer „Preview“ som aktivt trin. Fuldskræms-3D-viewport renderer den færdig trænede buket-scene (syntetisk Blender-test-sæt fra Bjoern, 60-frame-subset fra 960 hemisfæriske cams). Header-statusbjælke: „Training complete — 3 022 Gaussians in 13 s“ — giver endeligt gaussian-antal og træningstid. Drag i viewporten roterer kameraet (yaw/pitch); scroll-wheel zoomer langs view-direction. „Back“-knap (nederst til venstre) vender tilbage til Z2 for resume eller re-run; „Export“-knap (nederst til højre, primary) navigerer videre til Z4.

Efter afslutning af træningen lander appen automatisk i forhåndsvisningen. Her ser du din færdige Gaussian-splatting- model i en fuldskræms-metal-visning og kan dreje, zoome og panorere den med mus og trackpad. På oversiden af viewporten ligger en lille overlay med kamerastyring og info — auto- rotation, trænings-statistik, reset-knap. Før næste trin (eksport) er det værd at tjekke modellen fra forskellige vinkler for at sikre, at rekonstruktionen er ren.

C-36 SplatViewportView (3D-hovedvisning)

HVOR

Fuldskræms-baggrund af forhåndsvisnings-trinet.

TEKNISK

Metal-baseret 3D-viewport, der renderer den færdige punktsky. Rendereren er RadianceKits EGEN ForwardPass- rasterizer — den samme, der allerede renderer splats under træningen — så det er ægte WYSIWYG (det, der trænes, vises og eksporteres præcis ens). Tile-baseret rendering-pipeline med order-independent transparency. Hvis rendereren ikke kan initialiseres (f.eks. fordi Metal ikke er tilgængeligt på systemet), optræder i stedet en sort baggrund med „Metal not available“-tekst. Visningen ignorerer safe-area, så modellen rækker til vinduets kant.

KORT FORTALT

Hovedviewporten. Her ser du din færdige 3D-model rekonstrueret af dine fotos, renderet på GPU'en i realtid. Klik og træk med venstre museknap for at rotere. Scroll-hjul eller trackpad-gestus med to fingre for at zoome. Højre museknap eller Cmd+træk til at panorere. Modellen består af titusindvis af semi-transparente 3D-ellipsoider („gaussians“), der rekonstruerer din scene fotorealistisk — hver enkelt har en position, retning, form og farve, som træningen har lært. I det sjældne tilfælde, at din Mac ikke understøtter Metal, ser du i stedet en sort baggrund med en hint-meddelelse — RadianceKit kræver tvingende en Metal-istandsat GPU.

C-37 CameraControlsOverlay (styrings-overlay)

Over viewporten, svævende.



Kompakt UI-overlay med knapper til auto-rotation (turntable), reset-camera, baggrunds-valg (Gray/Black/White), save-screenshot, toggle-info-panel. Binder til kamera-parametrene (distance, azimut, elevation, target, FOV) og styrer auto-turntable. Under træningen (når brugeren i Expert Mode vil se viewporten køre med) viser overlayet desuden en kompakt trænings-status-linje.

 **KORT FORTALT**

Den lille svævelinje over modellen. Her starter du auto-rotationen (modellen drejer af sig selv, godt til screenshots og korte demoer), nulstiller kameraet via reset til startpositionen (hvis du har vildet væk), skifter baggrund (grå til neutral, sort til maksimal kontrast, hvid til lyse modeller) og laver screenshots direkte, der gemmes under /Pictures. Praktisk, hvis du vil vise en bestemt detalje fra en helt bestemt vinkel uden at skulle eksportere hele modellen. Auto-rotationen er også en god test for, om modellen ser lige god ud fra alle sider, eller om der er en „beskidt side“, opstået af manglende optagelser.

C-38 Export-knap (navigationslinje)

Nedre navigationslinje i Z3.



Accent-knap med label „Export“ og share-ikon. Klik udløser skift til Z4. Først tjekker den overordnede visning, om fuldversionen er låst op — hvis ikke, vises i stedet for eksport-stagen låsevisningen (se U-06).

 **KORT FORTALT**

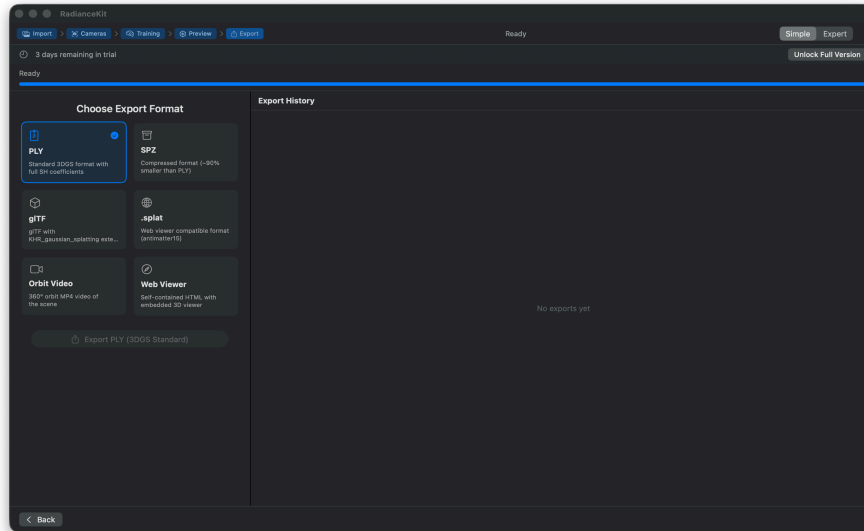
Hvis du er tilfreds med resultatet, så klik Export, og du lander i sidste trin, hvor du vælger format og gemmer. Uden købt fuldversion lander du i stedet på en skærmlåsning med unlock-hint og købs-knap — appen vil ikke påtvinge dig en fuldversion, men eksport er en af premium-funktionerne. Så snart du har afsluttet købet, kører appen direkte videre i låst-op-tilstand, og du lander i den vante eksport-stage. Hvis du ombestemmer dig, kommer du via Back- knappen tilbage til forhåndsvisningen og kan dreje videre på modellen.

Hvornår til næste trin?

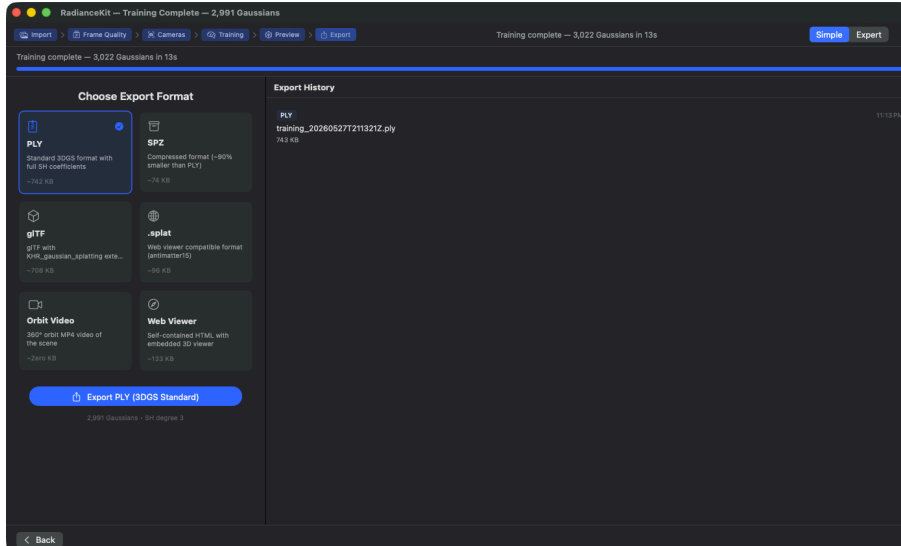
Før du eksporterer, så drej modellen en gang helt rundt og tjek: er alle områder, du har dækket i dine input-billeder, til stede? Findes der svævende „floaters“ (frit svævende gauss-splat-skyer i luften)? Virker baggrunden/himlen ren eller udtværet? Alvorlige pro-

blemer kan kun fikses ved nytræning — enten med flere billeder, anden preset eller i Expert Mode med floater- reduction-settings.

Z4 — Eksport (vælg format & gem)



Figur 40: Simple Mode eksport-trin med format-kort



Figur 41: Z4 eksport-kort — 6 formater (PLY 742 KB valgt, SPZ 74 KB, gITF 708 KB, .splat 96 KB, Orbit Video, Web Viewer 133 KB), Export-History-sidebar til højre med allerede eksporteret PLY

HVAD BILLEDET VISER Crumb-trail markerer „Export“ som aktivt trin. Venstre kort-grid „Choose Export Format“ med alle seks muligheder: PLY (standard-3DGS, 742 KB, med fulde SH- koefficienter — her forvalgt med blå flueben), SPZ (komprimeret 3DGS-format, ~90 % mindre end PLY, 74 KB), gITF (med KHR_gaussian_splatting-extension, 708 KB), .splat (web-viewer-kompatibel via antimatter15, 96 KB), Orbit Video (360°-MP4 af

scenen, live-størrelsesberegning), Web Viewer (selvstændig HTML med indlejret 3D-viewer, 133 KB). Størrelsesangivelser beregnes live ud fra aktuelt gaussian-count og format-overhead. Til højre „Export History“ lister allerede afsluttede eksporter med format-pille, filnavn og timestamp — klik afslører i Finder. Primary-CTA nederst til venstre: „Export PLY (3DGS Standard)“ med gaussian-undertitel „2,991 Gaussians · SH degree 3“.

I sidste trin vælger du blandt 6 eksport-formater (PLY, SPZ, glTF, .splat, orbit-video, web-viewer) via 2-spaltet kort-grid, klikker Export og vælger lagringssted i macOS-dialogen. Til højre kører en history over alle hidtidige eksporter — ved kort-valget vises straks under hvert kort den anslåede filstørrelse, så du f.eks. foretrækker SPZ, hvis du vil på nettet (lille), og PLY, hvis du vil importere til en anden software (SuperSplat, Postshot, Blender via plugin) (stor og fuldstændig).

C-39 2-Column Format Grid

HVOR

Venstre hovedside af eksport-trinet.

TEKNISK

Kort-raster med to fleksible kolonner og 12 punkt afstand. Itererer over de formater, der tilbydes i begyndertilstand — en filtreret delmængde af den fulde formatliste, der kun indeholder de 6 vigtigste formater: PLY, SPZ, glTF, .splat, orbit-video, web-viewer. Compressed-PLY og SOG tilbydes KUN i Expert Mode.

KORT FORTALT

Et kort-raster med de 6 formater, der er relevante i begyndertilstand: PLY (standardformat for andre 3D-værktøjer), SPZ (komprimeret variant til web), glTF (officiel web3D-standard), .splat (til antimatter15-web-viewer), orbit-video (færdig MP4 til at vise frem) og Web Viewer (selvstændig HTML-fil med indlejret 3D-afspiller). Dermed dækker du 90 % af anvendelses-tilfældene. Hvis du har brug for et af de mindre gængse formater (Compressed-PLY eller SOG til ekstrem komprimering), så skift til Expert Mode, der er alle 8 formater tilgængelige. Det kompakte valg her er bevidst, så begyndere ikke overvældes af mangfoldigheden.

C-40 Format Card Button**HVOR**

Hvert kort i griddet.

**TEKNISK**

Simpel knap med kort-layout: ikon (f.eks. dokumentzipper for PLY, arkivboks for SPZ, video-ikon for orbit-video) øverst, format-navn som headline, beskrivelses- caption (2-linjes afkortet), derunder den anslåede filstørrelse (live beregnet ud fra format, gaussian-count og SH-degree og formateret som KB/MB). Ved klik vælges formatet. Det valgte kort får accent-baggrund, accent-border og et flueben-ikon øverst til højre. Tooltip er format-beskrivelsen.

**KORT FORTALT**

Et kort pr. format. Klik en, det fremhæves med accent-farve og et flueben, og export-knappen herunder tilpasser sin tekst („Export PLY“, „Export SPZ“ osv.). Hvert kort viser et passende symbol, navnet, en tolinjet kortforklaring og den anslåede filstørrelse ved dit aktuelle trænings-resultat. Størrelsen hjælper dig med fornuftigt at vælge — hvis du vil sende resultatet pr. mail, så tag den mindste variant (som regel SPZ eller .splat); hvis du vil arbejde videre i en anden 3D-software, så tag den med bedste kompatibilitet (typisk PLY). Ved hover over et kort viser tooltipet en mere udførlig beskrivelse, hvis du finder forskellene mellem formaterne uklare.

C-41 Video Duration-slider**HVOR**

Under format-griddet, kun synlig hvis et video-format er valgt (orbit-video eller social-video).

**TEKNISK**

Skydeknapp 3–30 sekunder i 1-sekund-skridt, binder til video-længden i app-tilstanden. Maks-bredde 300 pixel. Vises kun, hvis et video-format er valgt. Ved ikke-video-formater fjernes skydeknappen helt fra visningen — ingen død plads.

**KORT FORTALT**

Hvis du vælger en orbit-video som eksport, kan du her bestemme længden. 3 sekunder = meget hurtig drejning, 30 sekunder = langsom, rolig drejning om din model. Til social-media-reels (Instagram, TikTok) er som regel 6–10 sekunder ideelt — længe nok til at vise modellen, kort nok til, at seerne ikke springer fra. Ved præsentationer eller portfolio-videoer må du gerne tage 15–20 sekunder. Slidern dukker kun op, hvis et video-format er valgt; ved fil-formater som PLY eller SPZ ville den være meningsløs og er skjult.

C-42 Export-knap **HVOR**

Under format-griddet (og under duration-slideren, hvis video valgt).

 **TEKNISK**

Stor accent-knap. Label: „Export {format-navn}“, share-ikon. Ved klik åbnes macOS-gem-dialogen med format- passende endelse og default-filnavn „scene.{ext}“; ved bekræftelse skrives eksporten til den valgte URL. Deaktiveret, hvis intet træningsresultat findes, eller en eksport allerede kører.

 **KORT FORTALT**

Klik, vælg lagringssted i macOS-dialogen, færdig — appen skriver filen i det valgte format på det valgte sted. Defaultnavn er „scene.{endelse}“ (f.eks. „scene.ply“ eller „scene.spz“), den kan du ændre vilkårligt i dialogen, før du gemmer. Knappen er grå, så længe intet træningsresultat foreligger (skulle aldrig ske her, fordi du ellers slet ikke ville være i eksport-trinet), eller en anden eksport allerede kører. Så snart eksporten kører, optræder herunder en fremskridtsvisning; appen forbliver betjenbar, så du kan allerede forberede næste eksport.

C-43 Export Progress Bar **HVOR**

Under export-knappen, kun synlig mens en eksport kører.

 **TEKNISK**

Fremskridtsvisning med maks-bredde 300 pixel, herunder caption „Exporting... N %“. Værdien går fra 0 til 1 og opdateres under skrivningen — ved PLY i chunks på 10 000 gaussians, ved SPZ engang efter kvantisering, ved orbit-video i frame-intervaller.

 **KORT FORTALT**

Mens eksporten kører, ser du her fremskridtet som smal bjælke plus procent-visning. PLY er som regel færdig i løbet af sekunder, fordi filen simpelthen skrives væk binært. SPZ tager lidt længere, fordi dataene undervejs kvantiseres og komprimeres. Orbit-video er den mest tidkrævende eksport — her rendres hver enkelt frame på ny; alt efter opløsning og længde kan det vare et minut eller længere. Under eksporten forbliver appen betjenbar, så du allerede kan forberede næste format eller klikke videre i viewporten.

C-44 Export Error Display **HVOR**

Under progress-baren, kun synlig hvis der opstod en fejl ved sidste eksport.

 **TEKNISK**

Rød linje med warning-ikon og fejltekst. Rød 8 %-baggrunds-opacity, afrundede hjørner. Maks-bredde 400 pixel. Hyppige fejlårsager: SOG forventer `cwebp` i system-PATH (ikke App Store-konform); skrivefejl ved fuldt diskplads; sandbox- fejl ved lagringsmål uden for det tilladte område.

 **KORT FORTALT**

Hvis eksporten går galt, optræder her i rødt en kort klartekst-beskrivelse af problemet. Som regel er årsagen indlysende — ingen plads på disken, ingen skriverettigheder til målmappen eller et mål uden for sandbox-tilladte områder. Specielt ved SOG-formatet sker det, at `cwebp` mangler i systemet; i så fald kan SOG ikke bruges, og du må vige til SPZ. Hvis fejlmeddelelsen er uklar, så kig i log-mappen (Help → Open Training Logs), der står mere udførligt, hvad der gik galt. I tvivl hjælper det at vælge et andet lagringssted — f.eks. skrivebordet.

C-46 Export History List **HVOR**

Højre side af eksport-trinet.

 **TEKNISK**

Liste over eksport-historikken (gemt persistent som JSON i UserDefaults, plejet efter hver vellykket eksport). Hver linje viser format-badge (lille, accentfarvet), timestamp (HH:mm), filnavn (1 linje afkortet) og formateret filstørrelse. Klik på en linje åbner Finder med valgt fil. Empty-state: „No exports yet“.

 **KORT FORTALT**

En liste over dine hidtidige eksporter — format, klokkeslæt, filnavn, størrelse, i kronologisk rækkefølge. Klik på en linje, og filen fremhæves i Finder, uden at du selv skal navigere gennem mapper. Praktisk, hvis du en time senere igen skal bruge den sidste eksport og ikke længere ved, hvor du gemte den — historikken husker det. Hvis du aldrig før har eksporteret noget, står her et venligt hint „No exports yet“. Listen overlever genstarter af appen, fordi den er gemt i UserDefaults.

C-48 History Context Menu (højreklik)



HVOR

Højreklik på en history-linje.



TEKNISK

Kontekstmenu på hver liste-post med to handlinger: „Reveal in Finder“ (åbner Finder med valgt fil, som et almindeligt klik) og „Copy Path“ (lægger den fulde fil- sti som tekst i udklipsholderen). Sidstnævnte er nyttig til træk-og-slip i andre apps eller overdragelse til kommandolinjen.

KORT FORTALT

Højreklik på en history-post åbner en lille menu med to handlinger. „Reveal in Finder“ gør det samme som et almindeligt klik — åbner Finder med valgt fil, så du straks ser den. „Copy Path“ lægger den komplette filsti i udklipsholderen, så du kan indsætte den f.eks. i terminal- kommandoer, i andre apps eller i en note. Særligt praktisk, hvis du vil give eksporten videre til nogen eller åbne den i et andet program, der arbejder med sti-input. Funktionen er lille, men hjælpsom detalje, der følger Mac-typiske betjenings-patterns.

Hvornår er arbejdsgangen afsluttet?

Efter en vellykket eksport har du din 3D-model som fil på disken, og historikken viser en ny post. Der er ingen „Done“- knap — du kan vedhænge vilkårligt mange eksporter i forskellige formater uden at træne på ny. Hvis du vil tilbage til forhåndsvisningen (f.eks. for igen at tjekke et kamera- perspektiv), så brug Back-knappen i den nedre navigationslinje. Hvis du vil starte en helt ny scene, så gå via Back tilbage til Z1 og brug der Clear All, eller File → New Project (Cmd+⇧+N).

Skift til Expert Mode

Tryk til enhver tid Cmd+2 eller vælg Mode → Expert Mode (M8). Hele tilstanden bevares: importerede billeder, valgt preset, kørende eller færdig træning, færdig punkt-sky, eksport-historik, endda den aktuelle stage. I Expert Mode vises i stedet for det fire-trins-stage hele Inspector-sidebaren med alle ~150 betjeningsfelter. Især: Project Navigator (se kapitel 2) tilbyder de udvidede billed-operationer (minus-knap, backspace-delete, Cmd-Z-undo, Quick-Look-forhåndsvisning), live-preview i viewporten under træningen samt alle loss-, MCMC-, densification- og Mip-splattung-parametre. Cmd+1 skifter tilbage til begyndertilstanden — også det taber ingen tilstand.

Ofte stillede spørgsmål

Hvorfor forbliver min Start Processing-knap grå?

Du har endnu ikke importeret billeder eller en video. Træk mindst en fil ind i drop-zonen, eller brug „Browse Files“. Så snart billedlisten til højre indeholder mindst en post, bliver knappen aktiv. (Ved kun 1–2 billeder starter den ganske vist, men SfM afbryder straks med fejl — se det røde validation- banner.)

Hvorfor er min Export-knap låst?

I begyndertilstand er der to trin: (a) Hvis trænings-pipelinen endnu ikke er færdig, og du ingen har, er knappen deaktiveret — du skal først afslutte Z2. (b) Hvis du endnu ikke har købt fuldversionen (`PurchaseManager.hasAccess == false`), ser du i stedet for eksport-stagen en låsevisning med låse-ikon og „Unlock Full Version“-knap, der åbner purchase-sheetet. Quick- og Preview-presets tillader træning gratis, men eksport er premium.

Hvorfor kan jeg ikke vælge en preset?

Du kan vælge den — men hvis du trykker på en premium-preset (Balanced, Quality, MCMC-varianter) uden købt fuldversion, springer vælgeren automatisk tilbage på Preview, og purchase-sheetet åbnes. Quick og Preview er de eneste gratis brugbare presets.

Hvorfor er min drop-zone tom og stiplet-grå, selv om jeg trækker billeder ind?

Sandsynligvis en UTI-type-mismatch. Appen accepterer JPG, PNG, TIFF, HEIC, MP4, MOV plus app'ens egne splat-formater. Andre billedformater (BMP, GIF, WebP, RAW-formater) genkendes IKKE. Hvis du er sikker på, at din billedtype burde være der, så tjek filnavn- endelsen — appen går primært efter extension, ikke efter fil-indhold.

Hvorfor varer SfM så længe, selv om jeg kun har 30 billeder?

Apple Photogrammetry skalerer ikke lineært — ved visse billed-konstellationer (indendørs rum med komplekse teksturer, bevægelses-uskarphed, dårligt lys) tager det markant længere, end billed-count lader formode. Hvis SfM efter 10+ minutter ved 30 billeder stadig hænger, så afbryd og prøv igen med bedre materiale, eller skift til Expert Mode og prøv COLMAP/Native- SfM (Cmd+2 → Inspector → Camera Alignment).

Hvor finder jeg mine training-logs?

Help → Open Training Logs (Cmd+⇧+L). Det åbner `~/Documents/RadianceKit/Logs/`. Hver trænings-session skriver en egen JSONL-fil med timestamp i filnavnet — første linje er konfigurationen, derefter følger en progress-linje hver 100 iterationer, sidste linje er summary med final-loss og success- flag.



KOLOFON

*Sat i SF Pro · Kode i SF Mono · Typst 0.14 · 22.
June 2026*

© 2026 Bjoern Kindler · Bischofshofener Str. 9, 82008 Unterhaching, Tyskland

Made with ❤️ in Unterhaching