



RADIANCEKIT

Brukerhåndbok

Fotorealistisk 3D-rekonstruksjon
via Gaussian Splatting

Versjon 1.5.0 · macOS 26.0+ · Mai 2026

BJOERN KINDLER · KINDLER-DEV.DE

Oversikt

| | |
|---|-----|
| Innledning — Det du bør vite | 3 |
| Hva er RadianceKit? | 3 |
| Hva er Gaussian Splatting? | 3 |
| Kapittel 1 — Menylinje | 5 |
| File-menyen | 5 |
| Mode-menyen | 9 |
| Training-menyen | 10 |
| Viewport-menyen | 14 |
| Export-menyen | 19 |
| Help-menyen | 24 |
| Merknad: Cmd-Z i Edit-menyen | 28 |
| Hurtigtaster i oversikt | 29 |
| Kapittel 2 — Inspector (Expert View) | 30 |
| Look-seksjon (L1–L5) | 33 |
| Presets-seksjon (I1–I11) | 36 |
| Treningskonfigurasjons-seksjon (I12–I22) | 42 |
| Enhancements-seksjon (I26–I29, I42–I44) | 48 |
| Metrikk-seksjon (I30–I38) | 55 |
| Loss-diagram-seksjon (I39–I41) | 61 |
| Når strekke ut etter Inspector? | 64 |
| Kapittel 3 — Innstillinger | 66 |
| General-fanen | 67 |
| AI-Helpers-fanen | 72 |
| Inspector-speilet settings | 75 |
| Når hva? | 76 |
| Kapittel 4 — Hjelpvinduer | 77 |
| User Guide (W1–W4) | 78 |
| Keyboard Shortcuts (W5–W6) | 81 |
| Manage Storage (W7–W12) | 83 |
| Pareto Dashboard (W13–W22) | 87 |
| Holdout Analysis (W23–W29) | 94 |
| BayesOpt Console (W30–W39) | 99 |
| Hovedvindu: loss-forløp og gaussian-count (I39–I41, krysshenvising) | 105 |
| Tommelfingerregel-boks | 106 |
| Kapittel 6 — Trenings-konfigurasjon | 108 |
| Iterasjon (T1–T2) | 110 |
| Learning rates (T3–T10) | 112 |

| | |
|--|-----|
| Densification — Classic (T11–T16) | 117 |
| Loss (T17–T20) | 121 |
| SH-grad-progresjon (T21) | 124 |
| Performance (T22–T25) | 125 |
| Diagnose og punktsky-forberedelse (T26–T30) | 127 |
| Regularisering (T31–T37) | 130 |
| Forfining (T38–T44) | 133 |
| Sky-dome (T45–T48) | 137 |
| Adam + LR-schedule (T49–T55) | 139 |
| Post-processing + Apple AI (T56–T60) | 142 |
| MCMC-densification (T61–T73) | 145 |
| Mip-splatting (Q1.5) (T74–T76) | 151 |
| Adaptive densification (Q5) (T77–T79) | 154 |
| Curriculum (Q6) (T80–T81) | 155 |
| Statistiske presets (TP1–TP9) | 156 |
| Metode: | 159 |
| Hvilket felt til hva? (cheat-sheet) | 160 |
| Farlige felter | 161 |
| Kapittel 7 — Innebygde kvalitets-forhåndsinnstillinger | 162 |
| Når hvilken preset? | 172 |
| Rask sammenligning | 173 |
| Egne forhåndsinnstillinger | 175 |
| Kapittel 8 — Eksportformater | 176 |
| Hvilket format når? | 189 |
| Rask sammenligning | 190 |
| Kapittel 9 — SfM-backends | 191 |
| Hvilken backend når? | 196 |
| Rask sammenligning | 197 |
| Kapittel 10 — Begynnermodus | 198 |
| Z1 — Import (velg bilder & preset) | 198 |
| Z2 — Behandling (SfM + trening) | 206 |
| Z3 — Forhåndsvisning (drei 3D-modellen) | 212 |
| Z4 — Eksport (velg format & lagre) | 215 |
| Bytte til Expert Mode | 220 |
| Ofte stilte spørsmål | 220 |

Slik leser du denne håndboken

Hvert oppslag i håndboken følger det samme skjemaet. På venstre side står betjeningsstiene og de tekniske detaljene; til høyre i en varm sidekolonne finner du alltid den enkle forklaringen. Små ikoner i starten av hver linje forteller deg på et øyeblikk hvilken type informasjon som følger nå.

DE FIRE IKONENE



Hvor finner jeg det? Den konkrete klikkstien gjennom appen — menylinje, Inspector-seksjon eller trinn i begynnermodus. De tilhørende hurtigtastene står også her. Ikonet er en kartnål og viser: Her sitter funksjonen i brukergrensesnittet.



Detaljer. Standardverdier, verdiområder og kodebaner. Du møter dette spesielt ved treningsinnstillingene, som ikke er et menyvalg, men tallparametere. Ikonet viser et lite spesifikasjonskort.



Teknisk. Hva funksjonen gjør internt, hvilke parametere som virker, hva den reagerer på, og hvilke bivirkninger den har. For lesere som vil forstå hva som skjer bak kulissene. Ikonet er en glidebryterblokk og står symbolsk for skruene under panseret.



Kort forklart. Kjernen i klare ord — uten fagspråk, uten kode. Les denne delen først hvis du bare raskt vil vite hva en funksjon er til for, og når du trenger den. Ikonet er en taleboble og står for „rett på sak«. Denne kolonnen er alltid satt i en varm sandtone, slik at øyet finner den med det samme.

KAPITTELFARGER

Hvert kapittel har sin egen aksentfarge, som du kjenner igjen på ID-merket (for eksempel **M1**) til venstre for hver oppslagstittel og på de små ikonene foran dem. Når du blar, ser du straks hvilket kapittel du befinner deg i.

- | | | | | |
|--------------------------------|--------------------|------------------------|-------------------------|------------------|
| 1 Menyer | 2 Inspector | 3 Innstillinger | 4 Hjelpvinduer | 6 Trening |
| 7 Forhåndsinnstillinger | 8 Eksporter | 9 SfM | 10 Begynnermodus | |

NAVIGASJONSTIPS

Hurtigstart. Hvis du bare er interessert i betjeningen, hopper du direkte til **Kapittel 10 – Begynnermodus**. Det er den veiledede varianten med fire trinn og krever ingen forhåndskunnskap.

Dypere innføring. **Kapittel 2 – Inspector** og **Kapittel 7 – Forhåndsinnstillinger** forklarer betjeningselementene og de forhåndskonfigurerte kvalitetsprofilene som du har til rådighet i ekspertmodus.

Oppslag. Innholdsfortegnelsen og PDF-fulltekstsøk hjelper deg med å finne en bestemt funksjon. Du trenger ikke å lese håndboken fra perm til perm.

Innledning — Det du bør vite

Hva er RadianceKit?

RadianceKit er en native macOS-app som av en rekke vanlige fotos eller en video lager en gåbar 3D-rekonstruksjon. Inputet er for eksempel 50 til 500 opptak som du har tatt rundt et objekt, gjennom et rom eller over et landskap. Resultatet er en såkalt Gaussian-Splatting-scene — en 3D-modell som du kan se på Mac-en i sanntid fra et hvilket som helst perspektiv, som lar seg eksportere og bygges inn på nettsider, og som i hovedsak ser fotorealistisk ut.

Appen kjører helt lokalt på din Mac — ingen bilder lastes opp til skyen, ingen pålogging kreves, ingen abonnement. Den utnytter GPU-en i Apple-silicon-Mac-en din (M-serien) intensivt: et fullt treningsforløp kan, avhengig av scene og forhåndsinnstilling, ta mellom to minutter og flere timer. Mens beregningen kjører, kan du arbeide helt normalt videre på Mac-en, RadianceKit kjører videre i bakgrunnen og melder fra når resultatet er klart.

Det finnes to betjeningsmoduser: *Begynnermodus* (Simple Mode) fører deg i fire trinn gjennom arbeidsflyten Import → Velg forhåndsinnstilling → Trening → Eksport. *Ekspertmodus* (Expert Mode) åpner en stor Inspector med alle skruer, et live-forhåndsvisningsvindu og diagnosediagrammer. Du kan når som helst veksle mellom modusene; dataene i scenen blir bevart.

Hva er Gaussian Splatting?

Gaussian Splatting (ofte kort *3DGS* eller bare *Splatting*) er en relativt ny metode for fotorealistisk 3D-fremstilling, presentert i 2023 i et paper fra Graz og INRIA. Idéen: i stedet for å modellere en scene som et klassisk polygonnett (trekanter) eller et voxel-rutenett, settes den sammen av millioner av små, myke 3D-skyer — hver enkelt sky er en 3D-gaussisk fordeling (derav navnet) med egen posisjon, størrelse, form, farge og gjennomsiktighet. Disse skyene trenes slik at de fra alle synsvinkler i inndata-fotoene dine til sammen gir det riktige bildet.

I praksis betyr det: Gaussian Splatting kan vise refleksjoner, høylys, mykt løvverk, hår eller gardiner på en måte som klassisk 3D-modellering ikke kan, eller bare kan med enorm innsats. Til gjengjeld er resultatet ikke en redigerbar 3D-modell i klassisk

forstand — du kan ikke bare flytte en enkelt vegg eller flytte en vase. Det er heller et *frosset opptak* av rommet, som du fritt kan bevege deg gjennom. For mange anvendelser — arkitekturvisualisering, produktpresentasjon, virtuelle turer, kriminalteknikk, kulturarv — er det nettopp den rette styrken.

For at det skal bli en 3D-scene ut av inndata-bildene, kreves to trinn. Først regner appen via en prosess som heter *Structure-from-Motion (SfM)* ut hvor kameraet ditt sto ved hvert foto. Som biprodukt oppstår en grov punktsky av scenen. Deretter starter selve Gaussian-Splatting-treningen: med utgangspunkt i denne grove skyen blir de millioner av 3D-skyene gradvis fordelt, forstørret, forfinet og justert i posisjon og farge, til de fra alle inndata-synsvinkler gir det passende bildet.

Du trenger ikke vite noe om noen av delene for å bruke RadianceKit. Begynnermodus skjuler disse trinnene helt. Men hvis du vil forstå hva diagnose-tallene i ekspertmodus (iterasjon, tap, gaussians, SSIM ...) betyr, eller hvorfor enkelte scener blir penere enn andre, så finner du svarene i de senere kapitlene av håndboken.

KAPITTEL

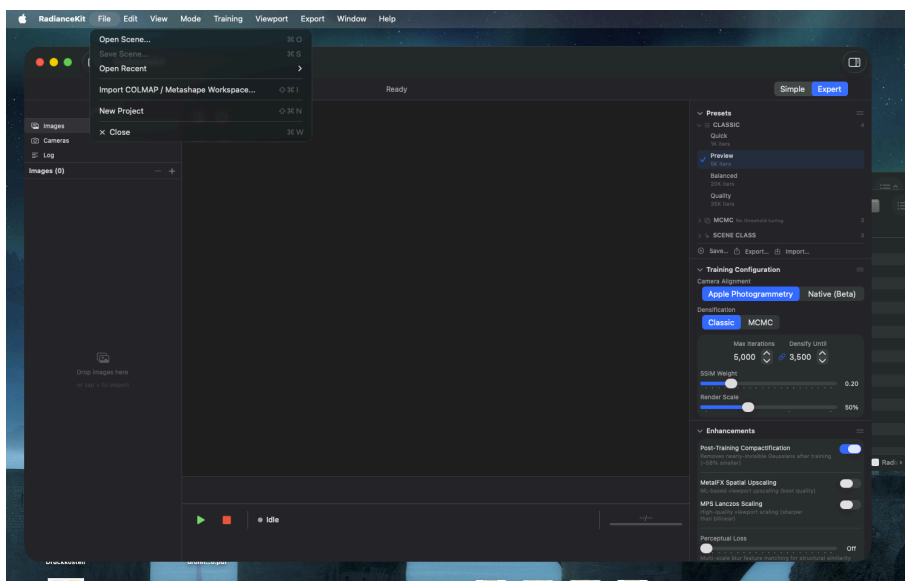
Kapittel 1 — Menylinje

RadianceKits menylinje strukturerer alle funksjoner som ikke ligger direkte i hovedvinduet eller Inspector. Det er først og fremst handlinger som virker på hele scenen (Åpne, Lagre, Nytt prosjekt), styrer treningen (Start, Pause, Fortsett), betjener viewporten (auto-rotasjon, skjermbilde, bakgrunnsfarge) og utløser eksporter til forskjellige 3D- og medieformater. I tillegg kommer hoppunkter til alle hjelpevinduer (User Guide, Pareto Dashboard, Holdout Analysis, BayesOpt Console).

Hurtigtaster står til høyre for menyoppføringen. Konvensjoner: **⌘** betyr Command-tasten (Apple-tasten), **⇧** er Shift, **⌥** er Option (Alt) og **⌘** er Control. Eksempel: **⇧⌘T** står for Shift+Command+T. Alle her dokumenterte snarveier vises dessuten i et eget oversiktsvindu via Help → Keyboard Shortcuts (⌘/).

De følgende 42 oppføringene er dokumentert i inventarets rekkefølge (M1–M42), gruppert etter den tilhørende toppnivåmenyen. Alle oppføringer er verifisert mot den aktuelle kodestanden i (linje 175–477). Ingen oppføringer er fjernet eller foreldet i forhold til inventaret; en ny Edit-menyoppføring (Cmd-Z for „Remove Image«) tas opp av system-NSUndoManager-ramverket og dukker derfor ikke opp i RadianceKitApp- koden (se merknad sist i kapittelet).

File-menyen



Figur 1: File-menyen foldet ut — oppføringer M1 til M6

File-menyen erstatter Apples standard „New Window«-oppføring med prosjektspesifikke handlinger. Den omfatter innlasting/lagring av scener, en dynamisk Recent-liste, workspace-importen og hard nullstilling til tom tilstand.

M1 File > Open Scene...



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



Åpner en filvelger for formatene `RadianceScene` -bundle, `.ply`, `.splat` og `.spz`. Enkeltvalg, kan vise både filer og mapper (for bundle-formatet). Etter valg skrives stien i Recent- listen, og scenen lastes inn asynkront — den forrige erstattes, og trenings-pipelinen initialiseres med den innlastede tilstanden. PLY/SPZ/Splat-filer leses via de respektive format-lasterne; `.radianceScene` -bundlen er en mappe med manifest, cloud-snapshot og SfM-resultater.

KORT FORKLART

Slik laster du en allerede trent scene tilbake i appen. Virker med RadianceKits eget format og med standardformatene PLY, SPLAT og SPZ, som andre splatting-programmer genererer. Bruk det hvis du f.eks. har trent en scene over natten og neste dag vil arbeide videre eller eksportere. Ved åpning erstattes den hittil værende tilstanden i hovedvinduet — så lagre først hvis den nåværende scenen fortsatt er viktig. Stien lander automatisk i „Open Recent« (M3), så du kommer raskere til den neste gang.

M2 File > Save Scene...



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



Åpner en lagre-fil-dialog med content-type `RadianceScene` -bundle og forhåndsutfyllt filnavn `scene.radianceScene`. Skriver en mappepakke med `manifest.json`, den serialiserte gaussian-skyen (PLY-snapshot) og et dump av SfM-resultatet, slik at continue-trening også virker etter gjenåpning. Oppføringen er deaktivert så lenge det ikke finnes gaussians. Lagrer ikke i training-logs-stien, men der lagre-dialogen peker — typisk under `~/Documents/`.

KORT FORKLART

Lagrer den aktuelle scenen som fil (mer presist: som en mappe som ser ut som en fil). Først deretter kan du senere åpne denne scenen igjen med „Open Scene...« (M1). I pakken lander både gaussian-skyen og SfM-resultatet, slik at du også kan henge på continue-trening (M12–M14) senere. Så lenge du ennå ikke har avsluttet en trening, er oppføringen grånet ut. Standardnavnet er `scene.radianceScene` — du kan imidlertid velge ditt eget navn i save-dialogen.

M3 File > Open Recent > [scenenavn]

Menylinjen → File → Open Recent → (liste).



Dynamisk undermeny som genereres ut fra en liste over de senest åpnete stier (lagret i innstillingene). Hver listeoppføring navngis med filnavnet og lastes ved klikk. Hvis listen er tom, vises i stedet det deaktiverte labellet „No Recent Scenes«. Apple-typisk holder listen de N senest åpnete scenene — begrensningen skjer ved skiving til innstillingene, ikke i menybyggeren selv.

 KORT FORKLART

Her ser du de senest åpnete scenene og kan hoppe tilbake til dem med et klikk uten å gå via fildialogen. Hvis du nettopp har begynt, er listen tom og står grå i menyen. Hver scene du åpner via „Open Scene...« (M1), lander automatisk i denne listen. Hvis listen på et tidspunkt blir for full, eller du vil tømme den av personvern hensyn, så bruk „Clear Recent« (M4).

M4 File > Open Recent > Clear Recent

Menylinjen → File → Open Recent → Clear Recent.



Tømmer Recent-listen i innstillingene. Virker umiddelbart uten bekreftelsesdialog. Oppføringen vises kun i undermenyen hvis det i det hele tatt finnes oppføringer i Recent-listen (den står under en divider etter stiene).

 KORT FORKLART

Sletter listen over de senest åpnete scenene. Praktisk hvis du har lekt med et testdatasett og ikke vil se stiene mer. Selve scenefilene slettes ikke — bare forbindelsen i menyen. Handlingen virker umiddelbart uten spørsmål; deretter vises „No Recent Scenes« i undermenyen. Oppføringen dukker bare opp hvis det i det hele tatt er scener i listen — ved tom liste er den ikke synlig.

M5 File > Import COLMAP / Metashape Workspace...

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



Åpner en mappevelger. Forventer en mappe med COLMAP-workspace-layout (f.eks. `sparse/0/cameras.{bin,txt}` pluss `images/`). Etter valg foretas en for-prøve av workspacet — denne gjenkjenner de tre layoutene (`sparse/0/`, `sparse/`, roten), og om rekonstruksjonen er binær (`cameras.bin`) eller ETH3D-tekst (`cameras.txt`). Ved suksess importerer workspacet; ellers vises bare en advarsel i app-loggen. Se også kapittel 9 „SfM-backends«, Q6 for den fulle pipeline-logikken.

KORT FORKLART

Hvis du bruker Metashape, COLMAP, RealityCapture eller lignende programvare for kamerate-rekonstruksjonen og har en eksport, laster du inn mappen her. RadianceKit hopper da over SfM-trinnet og starter direkte med treningen — det sparer timer på store scener. Dra-og-slipp på hovedvinduet virker på samme måte. Forventer en mappe med COLMAP-layout (altså `sparse/0/` med `cameras.*` pluss `images/-mappe`). Mer om de støttede layoutene og workflowene står i kapittel 9 „SfM-backends«.

M6 File > New Project

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

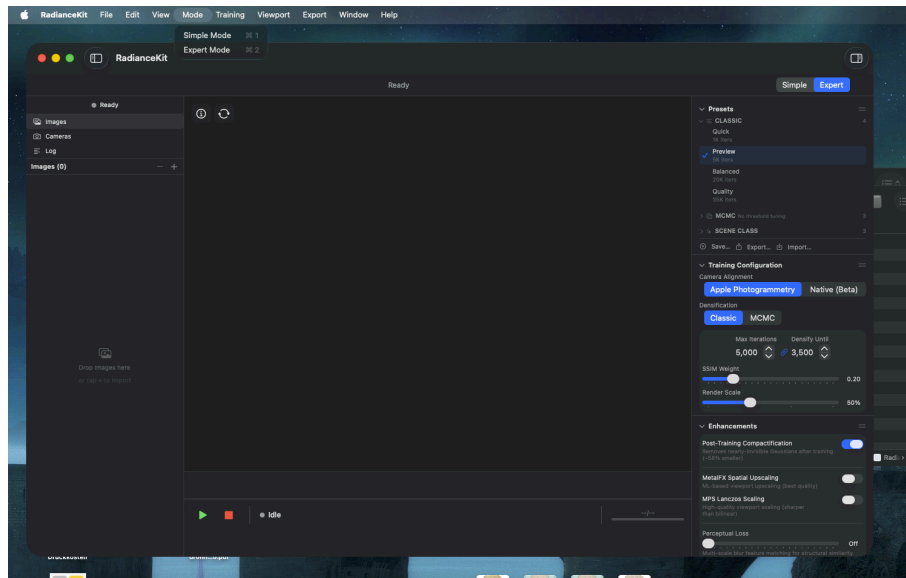


Sjekker om det finnes ulagret arbeid. Hvis ja, dukker en bekreftelsesdialog opp før noe går tapt. Hvis det ikke er noe å lagre, kjører nullstillingen direkte — den tømmer importerte bilder, SfM-resultatet, gaussian-skyen, treningstilstanden og alle avhengige UI-indikatorer. Advarsel: et preset-bibliotek opprettet av brukeren bevares fordi det ligger i app-innstillingene og ikke i prosjekttilstanden.

KORT FORKLART

Nullstiller alt til en tom start — som om du nettopp hadde åpnet appen friskt. Hvis du fortsatt har ulagret arbeid, spør appen først. Bruk det hvis du vil starte med en helt annen scene. Importerte bilder, SfM-resultat, gaussian-sky og treningstilstand tømmes helt. Dine egne presets bevares imidlertid fordi de ligger i app-innstillingene og ikke hører til scenen.

Mode-menyen



Figur 2: Mode-meny med Simple- og Expert-Mode-brytere

To enkle brytere mellom den veiledede Simple Mode (wizard-aktig, 4 trinn) og den fulle Expert Mode (klassisk Inspector-layout med alle håndtak).

M7 Mode > Simple Mode

HVOR

Menylinjen → Mode → Simple Mode (⌘1).

TEKNISK

Skifter app-tilstanden til Simple Mode. Appens hovedområde viser da den veiledede arbeidsflyten i stedet for Expert-layoutet. Modus-tilstanden lagres i innstillingene (se S1 „Default Mode« i kapittel 3 Settings).

KORT FORKLART

Skifter til steg-for-steg-varian-ten der appen fører deg gjennom import, behandling, forhåndsvisning og eksport. Anbefales hvis du nettopp har begynt, eller hvis du raskt trenger et resultat. De fleste detaljhåndtakene er skjult — du arbeider med fornuftige forhåndsvalgte innstillinger. Hvis du senere vil dykke dypere, så bytt bare til Expert Mode (M8). Hvilken modus som er aktiv ved app-start kan du fastsette i innstillingene (kapittel 3, S1).

M8 Mode > Expert Mode



HVOR

Menylinjen → Mode → Expert Mode (⌘2).



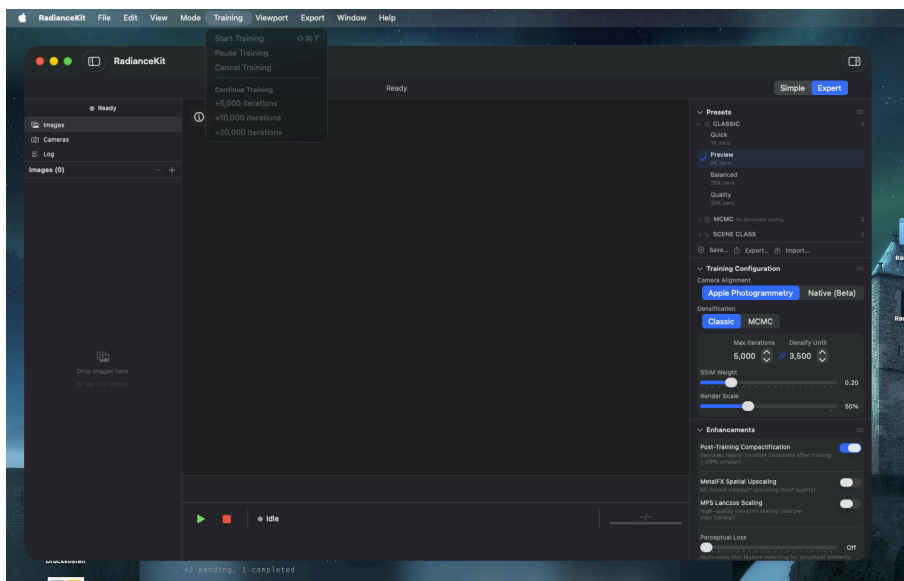
TEKNISK

Skifter app-tilstanden til Expert Mode. Dermed dukker det fulle Inspector-layoutet opp med alle seksjoner (Presets, TrainingConfig, Enhancements, Metrics, LossChart, ProjectNavigator). I Expert Mode er alle training-parametere, COLMAP-velger, mid-compact-toggles og diagnostikk tilgjengelig. Også live-preview fungerer bare i denne modusen.

KORT FORKLART

Skifter til fullvisning med alle håndtak. Her ser du loss-charts i sanntid, kan finjustere alle parametere og forvalte flere sammenlignings-konfigurasjoner parallelt via presets. Anbefales hvis du vil forstå hva treningen internt gjør, eller hvis du målrettet vil eksperimentere. Også live-preview, COLMAP-velger og diagnostikk er kun tilgjengelig her. Hvis du føler deg overveldet, så gå tilbake til Simple Mode via M7 — scenen din bevares.

Training-menyen



Figur 3: Training-menyen med Continue-undermeny — oppføringer M9 til M14

Fire handlinger rundt treningskjøringen: starte, pause, avbryte og forlenge med et angitt antall iterasjoner. Alle tre continue-oppføringer er IAP-gated (kan ikke klikkes i free-trial-versjonen).

M9 Training > Start Training

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



Starter trenings-pipelen asynkront. Forutsetning: et SfM-resultat foreligger, og det kjører ikke en annen pipeline. Begge betingelsene blokkerer oppføringen hvis de ikke er oppfylt. Ved start leses de aktuelle konfigurasjonsverdiene, en ny JSONL-log opprettes under `~/Documents/RadianceKit/Logs/training_YYYY-MM-DD_HHmmss.jsonl`, og avhengig av strategi-valget kjøres klassisk eller MCMC-sti. Trenings-tilstanden skifter fra „idle« til „training».

KORT FORKLART

Trykker på den store grønne knappen — så snart du har importert fotos og kamerarekonstruksjonen er over, begynner selve Gaussian-Splatting-treningen. La appen kjøre; avhengig av preset mellom 1 minutt (Quick) og flere timer (MCMC Quality). Oppføringen forblir grå så lenge det ikke foreligger SfM-resultat, eller en annen pipeline kjører. Hver kjøring skriver en logg til `~/Documents/RadianceKit/Logs/`, som du senere kan analysere via Pareto Dashboard (M40).

M10 Training > Pause Training

Menylinjen → Training → Pause Training.



Pauser den løpende treningen. Aktiveres bare når trenings-tilstanden er „training«. Pause stopper iterasjonsloopet ved neste sikkerhets-sync-punkt, beholder den fulle GPU-tilstanden (gaussian-buffers, optimizer-momenter, scheduler-posisjon) og skifter til „paused«. Resume skjer via fornyet trykk (oppførings-tittelen er statisk — appen skifter imidlertid mellom pause/resume i selve logikken). Pausede treninger overlever ikke app-quit; i så fall lagre heller scenen og utvid den senere via continue-training-oppføringen (M12–M14).

KORT FORKLART

Holder kort treningen tilbake uten å miste fremgangen. Praktisk hvis du kort må bruke datamaskinen til noe viktigere. Klikk igjen for å fortsette. Fungerer ikke på tvers av app-omstart — hvis du virkelig vil fortsette senere, avslutt treningen med Cancel (M11), lagre scenen med Save Scene (M2) og bruk deretter Continue Training (M12–M14). Under pause hviler GPU-en fullstendig; minnet forblir imidlertid opptatt.

M11 Training > Cancel Training

Menylinjen → Training → Cancel Training.



Avbryter den løpende treningen. Aktiv når trenings-tilstanden ikke er „idle«. Setter cancel-flagget i trenings-engine, som pent avslutter iterasjonsloopet ved neste sync-punkt, skriver det endelige summary-elementet i JSONL-loggen og nullstiller tilstanden til „idle«. Den hittil trente skyen bevares (kan lagres eller eksporteres), men markeres som „cancelled«.

 KORT FORKLART

Avbryter den løpende treningen endelig. Den hittil værende tilstanden bevares — så hvis du allerede har et fremvisbart resultat etter et par tusen iterasjoner, kan du likevel eksportere det. Hvis du bare vil avbryte kort, så bruk Pause (M10) i stedet. I trenings-loggen markeres kjøringen som „cancelled«, den endelige loss-verdien skrives likevel ned. En avbrutt scene kan du også fortsette senere via Continue Training (M12–M14), så lenge appen ikke har blitt avsluttet i mellomtiden.

M12 Training > Continue Training > +5 000 iterasjoner

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



Fortsetter treningen med 5 000 iterasjoner. Aktiv når en avsluttet trening kan fortsettes, og fullversjonen er låst opp. Fortsettbarheten gjelder når en avsluttet trening finnes, og den fulle optimizer-tilstanden fortsatt er i minnet. Ved continue føres Adam-momentene og LR-scheduleren videre, slik at fortsettelsen oppfører seg som en sammenhengende 25K-/45K-/60K-kjøring i stedet for en ny start. JSONL-loggen får en ny config-oppføring med den inkrementelle oppsetningen. Bare tilgjengelig i fullversjonen.

 KORT FORKLART

Henger 5 000 ytterligere treningssteg på. Bruk det hvis resultatet etter første kjøring er nær, men ennå ikke helt skarpt. Fungerer bare i den betalte fullversjonen. I motsetning til en helt ny kjøring bevares optimizer-tilstanden, slik at fortsettelsen føles som en sammenhengende kjøring. Hvis du trenger mer enn 5 000 steg, så ta direkte M13 (+10 000) eller M14 (+20 000).

M13 Training > Continue Training > +10 000 iterasjoner

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



Identisk med M12, men med 10 000 ytterligere iterasjoner. Samme forutsetninger, samme LR-scheduler-sti. Anbefales hvis den innledende treningen ble kjørt med en mid-tier-preset, og du vil se en signifikant kvalitetsøkning uten å starte kjøringen helt på nytt.

 KORT FORKLART

Forlenger treningen med 10 000 steg — den midterste av de tre tilgjengelige continue-verdiene. Godt valg hvis første kjøring var grei, men du tydelig vil bli bedre. Som M12 og M14 fortsettes læringsrate-forløpet sømløst i stedet for å starte på nytt. Bare tilgjengelig i fullversjonen.

M14 Training > Continue Training > +20 000 iterasjoner

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

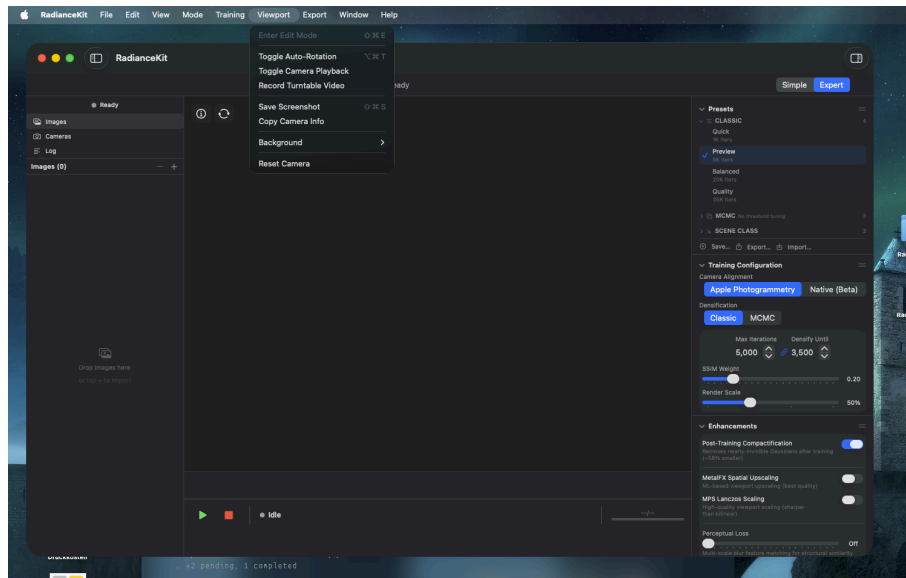


Identisk med M12 / M13, men med 20 000 ytterligere iterasjoner. Det største forhåndsdefinerte continue-spranget. Ved MCMC-treninger er det ofte det som gjør forskjellen mellom „passer« og „benchmark-egnet«; ved Classic fra 35–40K kommer det erfaringsmessig ikke mye til.

 KORT FORKLART

Henger 20 000 ytterligere treningssteg på, den maksimale continue-verdien. Bruk det hvis du virkelig vil hente den siste kvantitet kvalitet ut. Ved klassisk trening etter 40 000 steg gir det ofte ikke mye mer — ved MCMC derimot lønner det seg ofte fordi konvergensen setter inn langsommere der. Regn med markant ekstra kjøretid avhengig av scenen. Som M12 og M13 er også denne oppføringen bare tilgjengelig i fullversjonen.

Viewport-menyen



Figur 4: Viewport-menyen med edit-mode, kamerastyring og bakgrunn-undermeny

Styrer 3D-viewporten: edit-mode for gaussian-valg og opprydding, kamerastyring (auto-rotasjon, playback, recording), skjermbilde, bakgrunnsfarge og reset.

M15 Viewport > Enter/Exit Edit Mode

HVOR

Menylinjen → Viewport → Enter Edit Mode (eller „Exit Edit Mode« avhengig av tilstand). ⌘⌘E.

TEKNISK

Oppførings-tittelen er dynamisk og viser avhengig av tilstand „Exit Edit Mode« eller „Enter Edit Mode«. Ved trykk skiftes edit-mode på viewport-rendereren. Ved å forlate Edit Mode nullstilles dessuten det aktuelle valget. Edit Mode aktiverer klikkseleksjon på gaussians, box-selection og sletting av markerte gaussians (se editor-området i UI-en). Deaktivert så lenge det ikke er koblet en viewport-renderer.

KORT FORKLART

Skifter mellom normal 3D-visning og en redigeringsmodus der du kan markere og slette enkelte gaussians (f.eks. floaters eller outliers i bakgrunnen). Ved forlating nullstilles valget automatisk. Oppføringen forblir grå så lenge det ennå ikke er en scene synlig i viewporten. Etiketten skifter avhengig av tilstand mellom „Enter Edit Mode« og „Exit Edit Mode« — du ser altså alltid hvilken tilstand du er i.

M16 Viewport > Toggle Auto-Rotation

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



Slår den kontinuerlige rotasjonen av viewport-kameraet om en loddrett akse gjennom scenens sentrum av eller på. Akse og hastighet kommer fra kamerastyrings-konfigurasjonen. Auto-rotasjon er en ren viewport-effekt og påvirker verken trening eller recording — hvis du parallelt bruker turntable-video-recorderen (M18), leverer auto-rotasjonen imidlertid nøyaktig den stien recorderen fanger inn.

KORT FORKLART

Drei kameraet langsomt rundt scenen, slik at du kan se den fra alle sider uten å dra med musen. Klikk igjen stopper rotasjonen. Praktisk ved vurdering av ferdigtrente scener eller som bakgrunnsanimasjon for en live-demo. Hvis du parallelt tar opp en video (M18), leverer auto-rotasjonen nøyaktig den bevegelsen recorderen fanger inn.

M17 Viewport > Toggle Camera Playback

Menylinjen → Viewport → Toggle Camera Playback.



Slår kamerasti-playback på/av. Hvis det finnes en opptatt kamerasti (f.eks. fra en tidligere opptak, eller fordi en `transforms.json` er lastet inn), kjører stien — viewport-kameraet beveger seg altså ikke lenger etter muse-/trackpad-input, men reproduserer banen frame for frame. Fornyet trykk pauser playbacket.

KORT FORKLART

Lar en tidligere opptatt eller importert kamerakjøring spilles av. Slik kan du etterkikke den originale stien scenen ble tatt opp med, eller sjekke en planlagt orbit-bevegelse før video-eksporten. Mens playbacket kjører, er muse- og trackpad-input deaktivert — kameraet følger strengt stien. Fornyet klikk pauser avspillingen. Hvis du ikke har lastet inn eller tatt opp en kamerasti, skjer det ingenting.

M18 Viewport > Record Turntable Video

Menylinjen → Viewport → Record Turntable Video.



Slår viewport-opptaket på/av. Ved første trykk starter en frame-opptak i en midlertidig sti; ved andre trykk avsluttes opptaket, enkodes og skrives til en MP4-sti (stien spørres via en lagre-dialog). I motsetning til Export → Media → Orbit Video (M31), som genererer en fast 360°-sti med innstillbar varighet, tar turntable-recorderen *live* opp det du ser i viewporten — du kan altså også ta opp en manuell kamerakjøring.

 KORT FORKLART

Tar opp en video direkte i viewporten. Uansett om kameraet dreier automatisk, eller om du selv flytter det med musen — alt du ser, lagres i en MP4-fil. I motsetning til „Orbit Video«-eksporten (M31) angir du selv kamerakjøringen. Første klikk starter opptaket, andre klikk avslutter det og spør om lagringssted. Praktisk hvis du f.eks. vil vise en bestemt detaljsving som den stive orbit-bevegelsen ikke kunne klare.

M19 Viewport > Save Screenshot

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



Fanger et enkelt viewport-frame i full render-oppløsning (altså ikke vinduets piksel-layout, men det fulle render-target- innholdet) som PNG-fil. Stien spørres via en lagre-dialog. Bakgrunnsfargen (M21–M23) brennes med inn. MetalFX-/MPS-upscaling-innstillinger fra Enhancements (se I27/I28) virker hvis de er aktive — skjermbildet viser altså det oppskalerte output.

 KORT FORKLART

Lagrer et øyeblikksbilde av den aktuelle 3D-visningen som PNG-bilde. Praktisk for markedsføringsmateriell eller en rask sammenligning. Merk: bakgrunnen er en del av bildet — hvis du trenger gjennomsiktighet, så eksporter heller en scenefil. Oppløsningen tilsvarer det interne render-targetet, ikke vindusstørrelsen din — bildet er altså ofte skarper enn det ser ut i vinduet. Eventuelle upscaling-innstillinger (Inspector → Enhancements) regnes også med.

M20 Viewport > Copy Camera Info

Menylinjen → Viewport → Copy Camera Info.



Leser den aktuelle viewport-kamera-posesen (posisjon, look-at-punkt, up-vektor) og FOV-verdier fra kamerastyringen og skriver dem som flerlinjet tekst i utklippstavlen. Formatet er leselig for mennesker (label = value per linje), ikke JSON. Praktisk for å reproducere en spesifikk visning for debug-formål eller dele den med support.

 KORT FORKLART

Kopierer den aktuelle kameraposisjonen og blikkretningen som tekst i utklippstavlen. Hvis du f.eks. vil vise en medutvikler hvorfra et sted i scenen ser rart ut, så limer du bare inn teksten i en e-post eller et chat-vindu. Formatet er leselig for mennesker (én linje per verdi), ikke JSON. Mest tenkt for bug-rapporter eller support-henvendelser.

M21 Viewport > Background > Dark Gray

Menylinjen → Viewport → Background → Dark Gray.



Setter viewport-bakgrunnsfargen til en mørkegrå (RGB 0.1/0.1/0.1). Rendereren bruker denne fargen som bakgrunn foran hvilken gaussians komposites. Standardfargen ved app-start styres av Settings-alternativet S3 „Default Viewport Background«.

 KORT FORKLART

Farger 3D-viewportens bakgrunn mørkegrå. Standardvalget for de fleste scener — gir god kontrast både for lyse og mørke gaussians, uten at øyet henger fast i en ren sort- eller hvitflate. Fargen overtas også i skjermbilder (M19) og orbit-videoer (M31). Hvis Dark Gray er for kjedelig for deg, så prøv Black (M22) eller White (M23) til sammenligning. Hvilken farge som er aktiv ved app-start, kan du fastsette i innstillingene (S3).

M22 Viewport > Background > Black

Menylinjen → Viewport → Background → Black.



Setter viewport-bakgrunnsfargen til rent sort (RGB 0/0/0). Hjelper hvis scenen har mange lyse floaters og du vil identifisere dem, eller for markedsføringsmateriell med mørk look-and-feel.

KORT FORKLART

Sort bakgrunn. Bra for veldig lyse scener, eller hvis du vil kikke inn i Edit Mode og leter etter små lyse gaussians (floaters) som forsvinner i det grå. Også ideelt for markedsføringsmateriell med mørkt, dramatisk look. Fargen brennes inn i skjermbilder og orbit-videoer — hvis du trenger gjennom-siktighet for en senere composite, er sort det dårligste valget. For mørke floaters bytt den andre veien til White (M23).

M23 Viewport > Background > White

Menylinjen → Viewport → Background → White.



Setter viewport-bakgrunnsfargen til rent hvitt (RGB 1/1/1). Nyttig hvis scenen overveiende har mørkt innhold og du vil se mørke floaters (typisk outdoor-bakgrunnsstøy).

KORT FORKLART

Hvit bakgrunn. Praktisk hvis motivet heller kommer til sin rett lyst-på-mørkt, eller for å finne mørke outliers som du deretter kan fjerne i Edit Mode (M15). Ved outdoor-scener er hvit ofte mer nyttig enn sort fordi de typiske outdoor-floaters er mørke. Som ved de andre bakgrunnsalternativene overtas fargen i skjermbilder og videoer.

M24 Viewport > Reset Camera

Menylinjen → Viewport → Reset Camera.

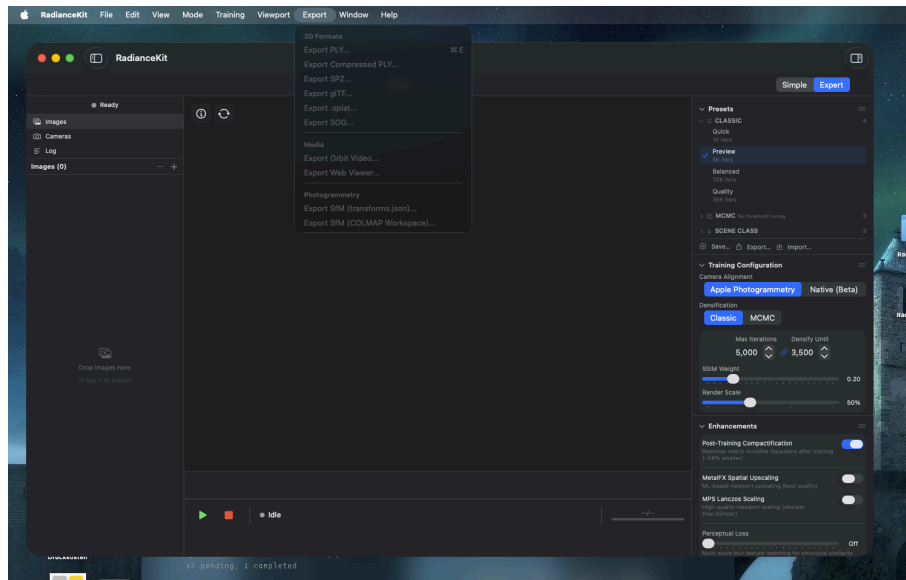


Nullstiller viewport-kameraet, forlater training-camera-visningen og stopper auto-rotasjonen. Dermed er kameraet tilbake på startposisjonen (typisk: foran scenen, lett ovenfra), auto-rotasjonen er av, og hvis rendereren nettopp viste training-kameraet (en av SfM-posene), går den tilbake til free-kameraet.

KORT FORKLART

Bringer viewport-kameraet tilbake til startposisjonen. Hvis du har gått deg vill under rundreisen eller dyttet scenen ut av bildet — et klikk her, og du ser igjen hva du skal se. Slår samtidig av auto-rotasjonen hvis den kjører, og vender tilbake fra et fastfrosset training-kamera til den frie visningen. Slik får du i alle tilfeller en ren omstart av visningen.

Export-menyen



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

Åtte eksportmål pluss to fotogrammetri-eksporter, gruppert i tre seksjoner (3D Formats, Media, Photogrammetry). De første seks bygges via en felles helper-rutine som åpner en lagre-dialog og registrerer eksporten i format-katalogen. Photogrammetry-oppføringerne har individuell logikk. Alle photogrammetry- og noen 3D-eksporter er bare tilgjengelige i fullversjonen.

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



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



Åpner en lagre-dialog med default-filnavn `gaussians.ply`. Ved OK skrives den aktuelle gaussian-skyen i det standardiserte ASCII/binary-PLY-formatet — kompatibel med SuperSplat, PolyCam, PlayCanvas og alle gjengse 3DGS-viewere. Fulle SH-koeffisienter, full presisjon (Float32 per felt). Filstørrelse ofte flere hundre MB ved $\geq 500K$ gaussians.

KORT FORKLART

Lagrer 3D-scenen din som standard-PLY-fil. Det er det mest universelle formatet — nesten enhver programvare kan laste det inn, fra SuperSplat over PolyCam til PlayCanvas. Filene blir imidlertid store, ofte flere hundre megabyte. Bruk PLY hvis du vil arbeide videre i full kvalitet eller arkivere. Hvis du vil dele scenen via nettet, så se heller på SPZ (M27) eller Compressed PLY (M26) — de er vesentlig mindre.

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

Menylinjen → Export → 3D Formats → Compressed PLY.

 TEKNISK

Skriver gaussian-skyen i Compressed-PLY-formatet med custom-kvantisering av posisjons-, skala-, rotasjons- og SH-feltene. 5–10× mindre filer enn ukomprimert PLY (M25) med minimale visuelle tap. Kompatibel med SuperSplat (som leser Compressed-PLY-standarden) og PlayCanvas. Standard filnavn `gaussians_compressed.ply`.

 KORT FORKLART

Som vanlig PLY, men 5–10 ganger mindre. Kvaliteten forblir nesten den samme. Bruk det hvis du vil dele filen online eller sende den per e-post. Fungerer direkte med SuperSplat og PlayCanvas. Hvis målsystemet ditt trenger enda mindre filer (mobil, browser-demoer), så velg SPZ (M27) i stedet — det er enda mer aggressivt komprimert. For full redigeringskvalitet velg den ukomprimerte PLY (M25).

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

Menylinjen → Export → 3D Formats → SPZ.

 TEKNISK

Skriver gaussian-skyen i SPZ-formatet — det av Niantic publiserte komprimerte splat-formatet med aggressiv kvantisering (~90 % mindre enn ukomprimert PLY). Optimalisert spesielt for web-viewere og mobile apper. Kompatibel med Niantic Splatt3R, `gsplat.js` og Niantic-browser-vieweren.

 KORT FORKLART

Et av de minste formatene. Ca. 10× mindre enn en vanlig PLY. Bruk det spesielt hvis du vil vise scenen i en nettleser eller via mobil-app. For maksimal kvalitet er PLY det bedre valget. SPZ er utviklet av Niantic og fungerer direkte med `gsplat.js`, Splatt3R og Niantic-web-vieweren. På grunn av den sterke komprimeringen kan du ikke lenger uten videre videretrene SPZ-filer — for redigering velg PLY.

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

Menylinjen → Export → 3D Formats → glTF.



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

 KORT FORKLART

Lagrer scenen i glTF-formatet, som mange 3D-programmer og web-engines forstår — forutsatt at de støtter Gaussian-Splatting-utvidelsen. Hvis du har en spesi-fikk 3D-pipeline (f.eks. Three.js eller Babylon.js) som forstår det, er det ditt format. Filen kommer ut som binær `.glb` — en enkelt pakke som inneholder alt. For klassiske splatting-workflows er PLY eller SPZ som regel det bedre valget fordi flere verktøy forstår dem direkte.

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

Menylinjen → Export → 3D Formats → .splat.



Skriver Antimatter15- `.splat` -formatet — fast størrelse 32 bytes per gaussian (posisjon som 3× Float32, skala som 3× Float32, rotasjon som 4× Uint8 normalisert quaternion, RGB+Opacity som 4× Uint8). Ingen SH-koeffisienter høyere enn DC. Minste fil med browser-direkte-kompatibilitet. For `gsplat.js` og `antimatter15s` online-demo-viewer.

 KORT FORKLART

Det enkleste web-viewer-formatet. Lite og straks visningsklart i enhver nettleser. Mister imidlertid detalj-belysningen (høyere SH-koeffisienter går tapt — splat-ten ser lik ut fra enhver synsvinkel i stedet for å reagere på lyset). For maksimal web-ytelse er det bra, for fotorealisme heller SPZ eller PLY. Fungerer med `antimatter15-online-viewer` og `gsplat.js`. Hver gaussian opptar fast 32 bytes, noe som gjør formatet enkelt og kompatibelt — men nettopp på bekostning av detaljdybden.

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

Menylinjen → Export → 3D Formats → SOG.



Skriver gaussian-skyen i SOG-formatet. SOG („Self-Organizing Gaussian«) er PlayCanvas-formatet med texture-atlas-layout og WebP-komprimering av de kvantiserte dataene. Skalerer med 15–20× bedre størrelsesforhold enn PLY. Eksporten kaller internt `cwebp` som eksternt verktøy — derfor i sandbox-varianten (App Store) potensielt begrenset.

 KORT FORKLART

Veldig lite format for PlayCanvas-workflows. Ca. 15–20 ganger mindre enn 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 valget. Eksporten kaller internt `cwebp` som eksternt verktøy — i App Store-versjonen (sandbox) kan dette trinnet være begrenset.

M31 Export > Media > Export Orbit Video...

Menylinjen → Export → Media → Orbit Video.



Rendrer et 360°-orbit rundt scenens sentrum og enkoder det som MP4 (H.264) eller MOV (HEVC, avhengig av systemstandard). I motsetning til M18 (live-recording) er stien her fast — varighet velges i Settings eller i Simple-Mode-eksport-trinnet.

 KORT FORKLART

Genererer automatisk en dreievideo rundt scenen din. Ingen manuell bevegelse nødvendig. Bra for sosiale medier eller en rask demo. Hvis du selv vil styre kameraet, så bruk Record Turntable Video (M18) i stedet. Stien er fast: en full 360°-orbit rundt scenens sentrum, varighet velger du i innstillingene eller i Simple-Mode-eksport-trinnet. Videoen publiseres avhengig av system som H.264-MP4 eller HEVC-MOV.

M32 Export > Media > Export Web Viewer...

Menylinjen → Export → Media → Web Viewer.



Pakker en standalone-HTML-viewer (gsplat.js-basert) pluss gaussian-dataene base64-kodet i en enkelt `.html`-fil. Filen kjører offline i enhver moderne nettleser — ingen server-avhengigheter, ingen eksterne URL-er. Filstørrelsen er ca. faktor 1.3 større enn SPZ-varianten (på grunn av base64-overhead).

 KORT FORKLART

Lagrer scenen din som selvstartende nettside. Dobbelklikk på HTML-filen → nettleseren åpner → ferdig interaktiv 3D-scene. Fungerer uten internett, kan sendes per e-post, er den enkleste måten å dele resultatet med venner eller kunder. Filen inneholder den komplette gsplat.js-vieweren og gaussian-dataene i ett enkelt dokument — ingenting hentes fra nettet. Filstørrelsen er ca. en tredjedel større enn en SPZ-eksport, til gjengjeld trenger motakeren ikke ytterligere programvare.

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

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



Egen eksport-sti (ikke via felles helper-rutine), fordi ingen gaussian-sky, men SfM-resultatet eksporteres. Åpner en lagre-dialog med `transforms.json` som default og content-type `json`. Ved OK skrives en nerfstudio-kompatibel `transforms.json` med kamera-intrinsics, poser (som 4x4-matrise i NeRF-konvensjon) og frame-stier. Hjelpetekst i UI-en påminner om at treningsbildene skal kopieres med som sibling-mappe `images/`. Aktiv bare når et SfM-resultat foreligger, og fullversjonen er låst opp.

 KORT FORKLART

Hvis du vil bruke SfM-resultatet videre i en annen programvare som nerfstudio, Brush, gsplat eller OpenSplat, eksporterer du her kameraposisjonene. Legg treningsbildene dine dessuten i en `images/`-mappe ved siden av `transforms.json`-filen — ellers kan målprogrammet ikke knytte bildene. Oppføringen er grånet ut så lenge det ikke finnes SfM-resultat, og sperret i free-trial-versjonen. For COLMAP-workspace-workflowen ta M34 i stedet.

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



HVOR

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



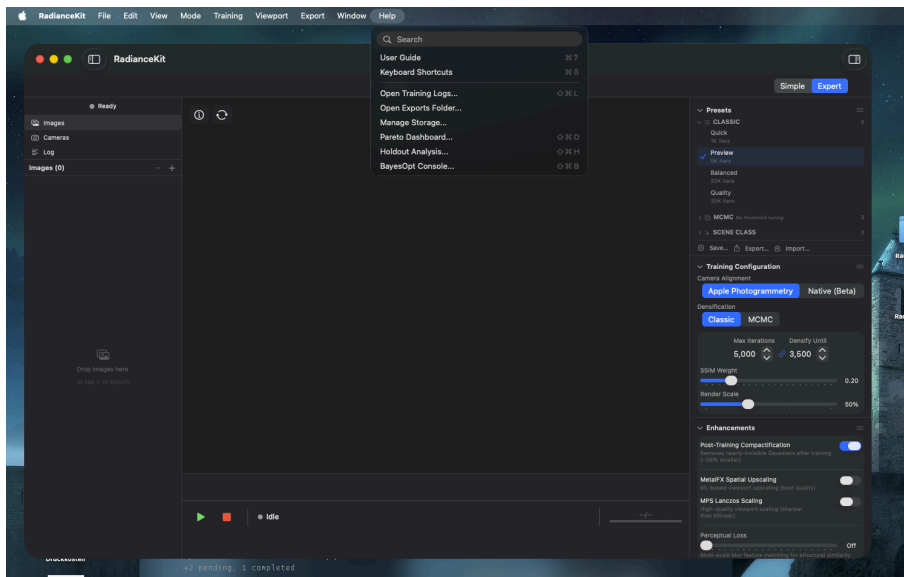
TEKNISK

Åpner en lagre-dialog med default-navn `colmap-workspace` (uten extension fordi det er en mappe). Skriver et standard-COLMAP-workspace med `sparse/0/cameras.bin`, `images.bin`, `points3D.bin`. Tillater å åpne en i RadianceKit beregnet eller importert SfM-rekonstruksjon i andre verktøy som Postshot, Nerfstudio eller Meshroom, eller ved en A/B-re-run å laste den inn igjen som ferdigberegnet input i selve RadianceKit (via M5) — sparer regnetid. Aktiv bare når et SfM-resultat foreligger, og fullversjonen er låst opp.

KORT FORKLART

Som M33, men i COLMAP-formatet i stedet for nerfstudio. Hvis du bruker Postshot, Meshroom, Nerfstudio eller et annet verktøy med COLMAP-workflow, er det din eksport. En praktisk sideeffekt: du kan laste denne mappen inn igjen senere via M5 i RadianceKit og spare SfM-regnetiden ved neste kjøring — spesielt ved store scener en tidsgevinst på timer. Som M33 bare tilgjengelig når et SfM-resultat foreligger, og sperret i free-trial-versjonen.

Help-menyen



Figur 6: Help-menyen med dokumentasjons-, mappe- og analyseoppføringer

Syv oppføringer: to dokumentasjonsvinduer (User Guide, Keyboard Shortcuts), tre mappe-snarveier (Training Logs, Exports, Storage) og tre analyse-vinduer (Pareto Dashboard, Holdout Analysis, BayesOpt Console). Apple-typisk dukker Help-menyen opp helt til høyre. Standard-Help-menyen erstattes helt av RadianceKits egen variant.

M35 Help > User Guide

Menylinjen → Help → User Guide (⌘?).



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

 KORT FORKLART

Åpner app-intern veiledning. Hvis du ikke vil slå alt opp i denne manualen, finner du de viktigste trinnene direkte i programmet. Veiledningen er bygget opp som et eget vindu med emne-sidebar — du kan altså målrettet hoppe til enkelte emner. Innholdet er kortere enn denne håndboken og konsentrerer seg om de hyppigste workflowene.

M36 Help > Keyboard Shortcuts

Menylinjen → Help → Keyboard Shortcuts (⌘/).



Åpner Keyboard Shortcuts-vinduet — et enkelt scroll-layout med alle app-snarveier, gruppert etter toppnivåmeny. Default-størrelse 440×560. Innhold er likeså statisk hentet.

 KORT FORKLART

Åpner et vindu med den komplette listen over alle hurtigtaster. Hvis du f.eks. ikke kan huske med hvilken tast man starter treningen, så se der. En sammenfatning står også sist i dette kapittelet. Listen er gruppert etter toppnivåmeny, slik at du raskt hopper til det rette området. Hjelpsom hvis du nettopp er i ferd med å bytte fra muse- til tastatur-stil.

M37 Help > Open Training Logs...

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



Beregner log-mappen som `~/Documents/RadianceKit/Logs`, oppretter den om nødvendig og åpner den i Finder. Hver treningskjøring skriver en egen JSONL-fil `training_YYYY-MM-DD_HHmss.jsonl` dit.

 KORT FORKLART

Åpner i Finder mappen med alle hittil værende treningsprotokoller. Hvis noe har gått galt, eller du vil se etter når nøyaktig treningen konvergente til hvilken verdi, finner du det her i JSONL-filer. Per treningskjøring opprettes nøyaktig én fil med tidsstempel — den kan du også lese i andre verktøy eller sende per e-post til support. Hvis du vil ha en grafisk evaluering, er Pareto Dashboard (M40) det bedre inngangspunktet.

M38 Help > Open Exports Folder...

Menylinjen → Help → Open Exports Folder...



Analogt med M37, men med `~/Documents/RadianceKit/Exports`. Opprettes ved første auto-test-kjøring eller ved første klikk; deretter ligger standardstiene av alle auto-test-eksporter der (f.eks. `autotest_<timestamp>.ply`). Manuelt via lagre-dialogen valgte eksporter går IKKE nødvendigvis hit, men der brukeren lagrer det — derfor er denne mappen spesielt interessant for auto-tester.

 KORT FORKLART

Åpner mappen der appen legger sine egne eksporter (spesielt auto-test-kjøringer). Hvis du har lagt en eksport manuelt et annet sted med lagre-dialogen, er den der og ikke i denne mappen. Praktisk for opprydding eller for å se etter hvor mye plass tidligere test-eksporter opptar. Hvis du trenger et komplett overblikk inkludert logs og scene-bundles, så ta i stedet Manage Storage (M39).

M39 Help > Manage Storage...

Menylinjen → Help → Manage Storage...



Åpner storage-browseren (se kapittel 4 Auxiliary Windows, ID-er W7–W12). Lister alle persister-te scener, training-logs, eksporter og caches i ~/Documents/RadianceKit/-mappen med størrelse, muliggjør reveal-in-Finder og move-to-trash per oppføring.

 KORT FORKLART

Åpner en vindu-browser som viser deg hvor mye plass RadianceKit opptar på disken — per scene, logg og eksport. Du kan slette enkeltting direkte uten å måtte gå til Finder. Praktisk etter lengre bruk når disken blir full — tidligere logger og auto-test-eksporter kan summere seg opp til flere gigabyte. Via reveal-in-Finder kommer du alltid også til den klassiske visningen.

M40 Help > Pareto Dashboard...

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



Åpner Pareto-dashbordet (se kapittel 4, ID-er W13–W22). Dashbordet laster inn alle JSONL-training-logs fra ~/Documents/RadianceKit/Logs/, ordner dem etter scene og preset og tegner et Pareto-scatter-plot (standard: loss vs gaussians, valgfritt loss vs wallclock eller PSNR vs iterasjoner).

 KORT FORKLART

Åpner et overblikk over alle hittil værende treningskjøringer som diagram. Du ser straks hvilken kjøring som har levert den beste balansen mellom kvalitet og størrelse. Praktisk hvis du vil sammenligne forskjellige presets med hverandre. Som standard viser diagrammet loss mot gaussian-antall — du kan imidlertid også bytte til wallclock-tid eller PSNR. Dataene kommer fra JSONL-training-loggene (M37); jo flere kjøring du har, desto mer utsigelig blir evalueringen.

M41 Help > Holdout Analysis...

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



Åpner holdout-analyse-vinduet (se kapittel 4, ID-er W23–W29). Laster inn en `transforms.json`, tegner kameraene som 3D-globus og tillater train/test-fold-splits (angulært eller lineært, 2–8 folds). Output er en `fold-assignment.json`, som treningen kan bruke i de respektive training-configs som test-sett.

KORT FORKLART

Hjelper deg med å dele kamera-opptakene dine i trenings- og test-sett — slik at du objektivt kan måle hvor god scenen din er (på bilder som treningen ikke har sett). Heller et forsknings- og benchmark-verktøy. Kameraene vises som 3D-globus; du kan velge mellom 2 og 8 folds, enten jevnt i vinkelen eller lineært over rekkefølgen. Resultatet er en liten JSON-fil som treningen så bruker som test-sett.

M42 Help > BayesOpt Console...

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



Åpner BayesOpt-konsollen (se kapittel 4, ID-er W30–W39). Laster inn forhåndsdefinerte søkerom (f.eks. „MCMC scale-reg + opacity-reg + ssim«), kjører bayesian-optimization- trials asynkront og viser konvergenskurve og trial-log live.

KORT FORKLART

En innebygd auto-tuner-konsoll. I stedet for manuelt å prøve forskjellige parametere kan appen la det kjøre selv over natten og foreslå deg de beste verdiene for din scene til slutt. Veldig avansert verktøy — for de fleste workflows er en god preset (se kapittel 7) tilstrekkelig. Du velger et forhåndsdefinert søkerom (f.eks. „MCMC scale-reg + opacity-reg + ssim«) og ser live konvergenskurven samt trial-loggen. Regn med flere timer til dager avhengig av oppsetningen.

Merknad: Cmd-Z i Edit-menyen

Siden mai 2026 støtter Project Navigator i Expert Mode sletting av importerte bilder via minus-knapp eller backspace-tast og angring via `Cmd-Z`. Denne `Cmd-Z`-handlingen dukker opp i macOS-Edit-menyen (som leveres av SwiftUI) som „Undo Remove Image« så lenge et slettet bilde fortsatt kan gjenopprettes. Den registreres via det standardkonforme systemet, ikke i ; derfor er det ingen egen M-ID-oppføring i inventaret.

Hurtigtaster i oversikt

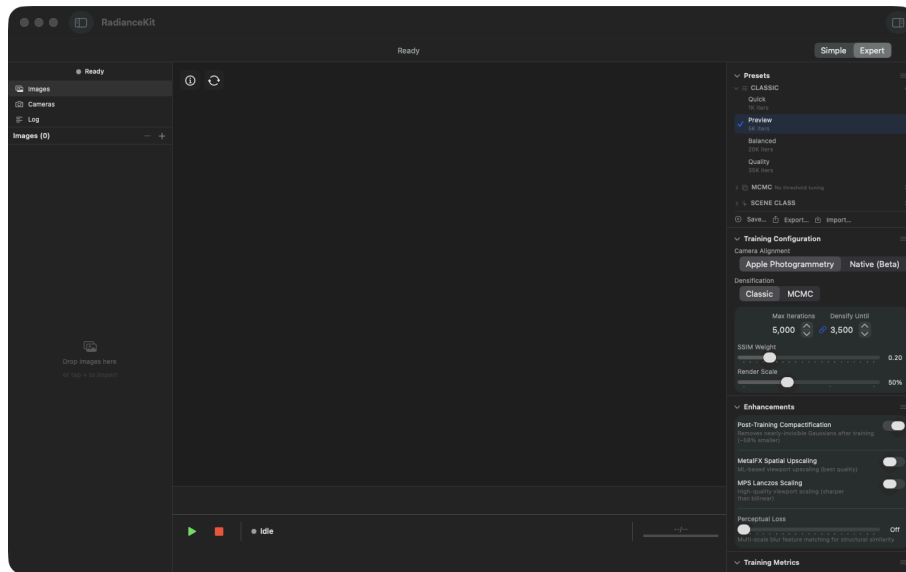
| Menyoppføring | Snarvei |
|---|---------|
| File > Open Scene... | ⌘O |
| File > Save Scene... | ⌘S |
| File > Import COLMAP / Metashape Workspace... | ⇧⌘I |
| File > New Project | ⇧⌘N |
| Mode > Simple Mode | ⌘1 |
| Mode > Expert Mode | ⌘2 |
| Training > Start Training | ⇧⌘T |
| Viewport > Enter/Exit 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-meny (systemlevert, i Expert Mode ved aktivt Project-Navigator-valg):

| Handling | Snarvei |
|-----------------------|--------------------|
| Undo Remove Image | ⌘Z |
| Remove Selected Image | Backspace / Delete |

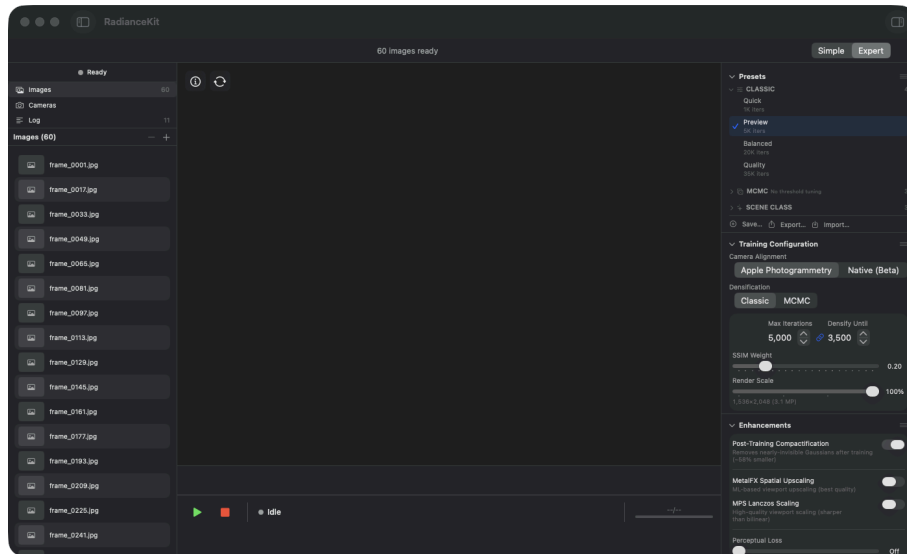
KAPITTEL

Kapittel 2 — Inspector (Expert View)



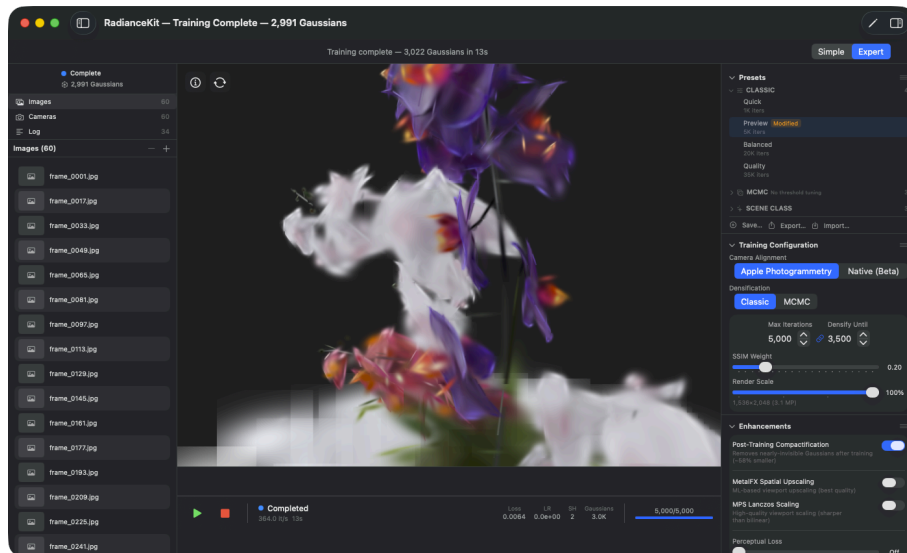
Figur 7: Ekspertmodus tom — Project Navigator til venstre (Images 0, Cameras, Log), tom viewport i midten, Inspector til høyre med Presets/Training Configuration/Enhancements/Training Metrics-seksjoner

Tom Inspector før import: Venstre sidefelt viser images-teller 0 og drop-hint „Drop images here / or tap + to import«. Inspector til høyre er fullt funksjonell, men presets er bare informative (ingen aktiv trening). Default-preset „Preview« (5K iters) er markert. 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 blomsterbilder lastet inn — image-sidefeltet viser de første filnavnene `frame_0001.jpg ff`, header „60 images ready«

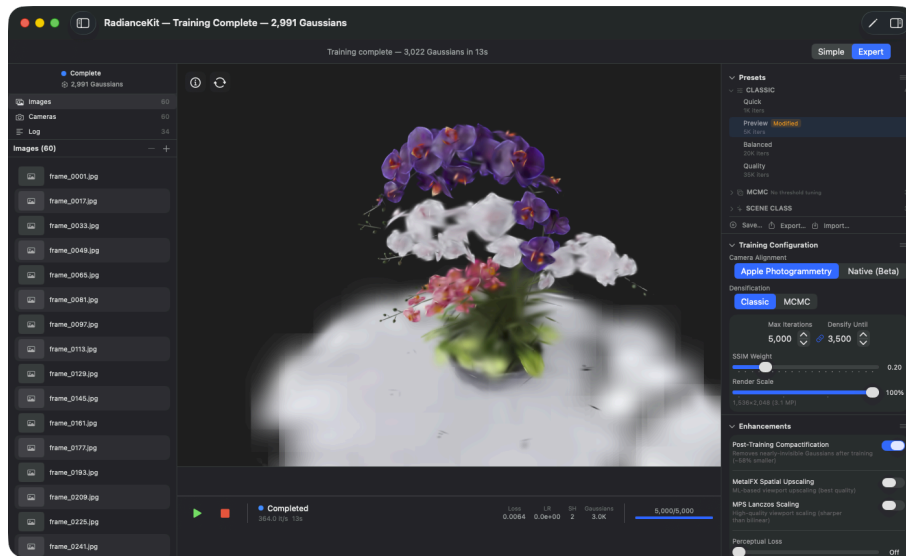
Inspector etter import: Header-status „60 images ready«. Image-sidefeltet lister alle 60 importerte frames (`frame_0001.jpg` til `frame_0945.jpg` , hver 16. frame fra et 960-cam-bukett-datasett som subset for raske iterasjoner). Auto-render-scale-logikken sjekker bildeoppløsningen ($1536 \times 2048 = 3.1$ MP) og tilpasser Render Scale deretter. Play-knappen (grønn, nede til venstre) er nå aktiv og starter treningen med den aktive preseten.



Figur 9: Inspector mid-training — live-viewport viser bukett- rekonstruksjonen, metrikk-bar nederst (Loss / LR / Gaussian-Count / Iterasjoner), preset-card „Preview« med „Modified«- badge hvis parametere er endret

Inspector under trening: Tittelinjen viser global fremgang „RadianceKit — Training NN %«. Viewporten renderer den løpende gaussian-rekonstruksjonen i sanntid (oppdatert hver 50 iterasjoner — live-preview-intervall kan justeres i Settings → General → Training → Live Preview). Metrikk-bar under viewporten: aktuell loss, learning rate, gaussian-count og iterasjoner-teller (f.eks. 1.600/5.000 ved Preview-preset). Inspector-preset-

card „Preview» har „Modified«-badgje så snart en parameter avviker fra den innebygde default. Sidefeltet „Log» samler SfM- og training-stage-events.



Figur 10: Inspector etter treningsavslutning — viewporten viser ferdig bukett-rekonstruksjon (2.991 gaussians etter 5K iterasjoner på 13 s), tittelinje „Training Complete — 2.991 Gaussians«

Inspector etter trening: Tittelinjen viser endelig gaussian-antall (her 2 991 — veldig kompakt, fordi den syntetiske Blender-bukett-scenen på lys bakgrunn har enkel geometri). Viewporten viser den ferdige punktskyen — orbital drag-navigasjon aktiv (roterer rundt scenens midtpunkt). Training-Metrikks-seksjonen er nå fylt med sluttverdier, loss-history-charten viser forløpet av de samlede 5 000 iterasjonene. Eksport-seksjonen nederst er nå aktiv (alle format-knapper enabled).

Inspector er den høyre sidefeltet i Expert Mode (§2). Den samler alle treningsrelevante parametere i sju sammenleggbare seksjoner. Default-rekkefølgen ovenfra og ned ved første oppstart er: Look, Presets, treningskonfigurasjon, metrikker, loss-diagram, Enhancements og Export. „Look«-seksjonen (bildejusteringer etter trening) er den reelle UI-omdøpingen av den tidligere „Finishing«-seksjonen — dens interne enum-rawValue forblir „Finishing« av persistens-grunner, den viste overskriften heter „Look«. Hver seksjon kan klappes sammen ved klikk på headeren, rekkefølgen kan omarrangeres via dra-og-slipp (InspectorView.swift:81–97). **Ved første oppstart er alle sju seksjonene sammenklappet** (inspectorCollapsedSections defaulter til Set(InspectorSection.allCases)); app-state lagrer klapp- og rekkefølge-preferansene deretter på tvers av app-omstart.

En rekke betjeningselementer fra Inspector dukker også opp i nesten identisk form i innstillingene (kapittel 3) — typisk SfM-backend, sky-masking og lignende defaults. Adskillelsen er bevisst: innstillingene leverer den app-globale malen for nyopprettede prosjekter, Inspector overskriver disse verdiene for det aktuelt åpne prosjektet. Den som kjenner betjeningslogikken på den ene siden kan bruke den andre blindt.

Den venstre kolonnen i Expert Mode — Project Navigator — hører ikke til Inspector, men er den direkte naboen. Der kan importerte bilder velges med klikk, ses i Quick-Look med mellomromtasten og slettes via minus-knappen eller delete-tasten (med Cmd-Z

for å angre). Inspector følger det aktuelle sidefelt-valget med kontekstspesifikk detaljinformasjon, men de sju hovedseksjonene forblir alltid tilgjengelige.

Look-seksjon (L1–L5)

Look-seksjonen (intern `rawValue` fortsatt „Finishing«) er den øverste Inspector-seksjonen og samler **bildejusteringene etter trening** på ett sted. Alle reglere arbeider **ikke-destruktivt**: hver slider anvender `FinishingPass` på nytt på et uendret pristine-snapshot (original DC-farge, -opacity, -skalering) — justeringen er dermed **idempotent**, ikke kumulativ. Resultatet vises **live i viewporten** (WYSIWYG, akkurat slik den senere eksporten blir) og **bakes inn i hver eksport**. Seksjonen er først tilgjengelig **etter at et treningsforløp er avsluttet** (før det står „Available after a training run completes.»); verdiene **nullstilles ved hver ny trening**. Så lenge en eksport kjører, er alle reglere **låst** — en lås-hint „Locked while exporting — the file uses the current settings.« vises og GroupBox-en er disabled.

L1 Saturation-slider

HVOR

Inspector → Look-seksjon → GroupBox → Saturation.

TEKNISK

Slider 0.5–1.2, visning med to desimaler (f.eks. „1.00”). Skalerer SH-DC-chromaen til hver Splat rundt luminans-verdien: 1.0 = uendret, < 1.0 = avmettet (fargen trekkes mot gråtone), > 1.0 = kraftigere. Matematisk regnes DC-fargen tilbake fra pristine-snapshotet (`desaturateDC`), slik at gjentatt skyving ikke summeres opp. Ble validert på DJI-drone-materiale (Pensford-viadukten), som har en tendens til å være overtegnet — drone-defaulten ligger på 0.82. Virker bare på farge-basisen (SH-grad 0), høyere SH-koeffisienter forblir urørt.

KORT FORKLART

Hvor kraftige fargene til den ferdige Splat er. 1.00 lar alt være som trent, verdier under trekker fargen mot grått — bra for drone- eller video-materiale, som ofte kommer ut overmettet. Verdier over 1.0 gjør det kraftigere. Du kan skyve fram og tilbake så mye du vil uten at noe „bygger seg opp«, fordi appen alltid regner på nytt fra den uendrede originaltilstanden. Synlig live i viewporten og akkurat slik i eksporten.

L2 Splat length-slider



Inspector → Look-seksjon → GroupBox → Splat length.



Slider 0.3–1.0, visning med to desimaler. Trekker de tre skalerings-aksene til hver gaussian i log-rommet mot gjennomsnittet (`shortenScale` , faktor `alpha`): 1.0 = uendret, mindre verdier gjør avlange „nål«-Splats rundere, 0 ville vært rene kuler. Tar tak i nål-aktige, overstrukne Splats uten å endre totalstørrelsen, og reduserer dermed typiske „konfetti«-artefakter. Anvendes fra pristine-snapshotet (original log-skaling), derfor idempotent. Kommuterer med Splat size (L3), fordi begge arbeider i log-rommet.

KORT FORKLART

Gjør overlange, flisete Splats rundere. 1.00 lar formen være som trent, lavere verdier staker de langstrukne „nålene« til rundere klatter — det roer ned kornete rekonstruksjoner plaget av konfetti-artefakter. Totalstørrelsen forblir den samme, det handler bare om avlangheten. Lar seg trygt kombinere med Splat size (L3).

L3 Splat size-slider



Inspector → Look-seksjon → GroupBox → Splat size.



Slider 0.5–2.0, visning med to desimaler. Skalerer hver gaussian uniformt på **alle** tre aksene (`sizeScale`): 1.0 = uendret, < 1.0 = mindre/tettere/skarpere, > 1.0 = større/„fluffigere« (fyller hull mellom Splaten). Siden skalingene ligger i log-rommet, realiseres multiplikasjonen som en additiv `log(factor)` -offset — det kommuterer med Splat length (L2), fordi en konstant offset lar avviket-fra-gjennomsnittet være urørt. Fra pristine-snapshotet, altså idempotent. Ny i denne versjonen.

KORT FORKLART

Skalerer alle Splats jevnt større eller mindre. 1.00 er den trente tilstanden, verdier under gjør punktskyen tettere og skarpere, verdier over dekker hull mellom Splaten (virker mykere/„fluffigere«). Praktisk for å lukke en hullete rekonstruksjon optisk eller motsatt hente fram mer detalj. Tåler problemfritt Splat length (L2) — begge reglene påvirker ikke hverandre.

L4 Fade far region (med sub-slidere)

HVOR

Inspector → Look-seksjon → GroupBox → toggle „Fade far region« pluss sub-slidere „Fade start xradius« og „Fade floor«.

TEKNISK

Toggle som aktiverer et radiale opacity-fall med avstanden fra kamera-tyngdepunktet — den svakt observerte „far-konfettien« i bakgrunnen blendes ut. **Bare for orbit-opptak:** toggle-en er disabled når `finishingContext.fadeEligible` er false (lineære flyvninger, for få eller degenererte kameraer); da vises i stedet for sub-slidere hinten „Far-fade applies only to orbit captures (not this scene).« Egnetheten avgjøres via azimut-dekning av kamera-posisjonene (en orbit omkretser tyngdepunktet og fyller mange kompass-sektorer, en lineærflyvning bare ~2). To sub-slidere styrer geometrien: **Fade start xradius** (1.0–3.0) setter innerradiusen som et multiplum av orbit-radiusen, innenfor hvilken full opacity gjelder; **Fade floor** (0.0–1.0) er opacity-faktoren langt forbi fade-radiusen. Viktig: faden **hopper over sky-dome-området** (de frozne gaussians med indeksene [0, frozenCount]), slik at den bevisste bakgrunns-kuppelen ikke dimmes med.

KORT FORKLART

Blender ut de svampaktige restene ytterst i scene-kanten — nettopp de „far-konfetti«-klumpene som svever langt bak ved rundt-omkring-opptak. Fungerer bare ved ekte orbit-/omkretsings-opptak; ved rette drone-flyvninger eller for få kameraer er bryteren grået ut og en hint forklarer hvorfor. Er den aktiv, kommer det to fininnstillere til: „Fade start xradius« fastsetter fra hvilken avstand (som et multiplum av omkretsradiusen) utblendingen begynner, „Fade floor« hvor sterkt de fjerne Splatene fortsatt forblir synlige til slutt (0 = helt borte, 1 = uendret). En bevisst rekonstruert sky-dome (I44) røres aldri ved — himmelen beholdes.

L5 Reset finishing-knapp

HVOR

Inspector → Look-seksjon → GroupBox → „Reset finishing« (nederst, liten knapp).

TEKNISK

Nullstiller alle Look-settings til defaultene (`FinishingPass.Settings()` = Saturation 1.0, Fade av, Splat length 1.0, Splat size 1.0) og utløser straks et nytt finishing, slik at viewporten spretter tilbake til den uendrede trente tilstanden. `controlSize(.small)`. Siden hele Look-stacken regner idempotent fra pristine-snapshotet, er „tilbake til default« akkurat det opprinnelige trenings-outputet — ingen kvalitetstap ved gjentatt fram og tilbake. Som alle reglerne i seksjonen låst under et pågående eksport.

KORT FORKLART

Stiller med ett klikk alle Look-reglere tilbake til standard (Saturation 1.00, Fade av, begge Splat-slidere på 1.00) — viewporten viser deretter igjen akkurat det ferskt trente resultatet. Praktisk når du har lekt deg bort og vil starte rent på nytt. Fordi appen alltid regner fra originaltilstanden, blir det ikke noe kvalitets-tap. Mens en eksport kjører, er knappen (som sliderne) låst.

Presets-seksjon (I1–I11)

Presets-seksjonen er den raskeste måten å anvende en utprøvd konfigurasjon. Innebygde presets (Capture Class, Classic, MCMC, Hybrid) leverer reproduerbare startpunkter fra 560+ dokumenterte eksperimenter; egne presets kan lagres, eksporteres, importeres og deles. Listen er gruppert etter kategorier (Capture Class, Classic, MCMC, Hybrid, Custom), og mer enn én kategori kan være foldet ut samtidig. Via kontekstmeny-mekanismen (høyreklikk på en linje) er eksport, duplisering og — ved egne presets — sletting tilgjengelig.

I1 Save...-knapp

 HVOR

Inspector → Presets-seksjon → Save...-knapp (action-bar nederst).

 TEKNISK

Åpner en popover med tekstfelt og save-/cancel-knapper. Den aktuelle TrainingConfig-tilstanden persisteres som ny brukerdefinert preset (JSON-kodet, lagret på tvers av appen). Save-operasjonen kopierer alle 81 treningsparametere pluss den aktuelle densification-strategien. Presetet lander automatisk i kategorien Custom, uavhengig av hvilken innebygd preset det ble avledet fra. Tomme navn og rene whitespace-input forkastes. Allerede eksisterende navn avvises ikke — hvert preset har sin egen interne ID, doble navn er teknisk tillatt, men praktisk forvirrende.

KORT FORKLART

Sikrer den aktuelle konfigurasjonen din som gjenbrukbar preset. Trykk på knappen, skriv et navn i popoveren og klikk Save — alle 81 parametere inkludert densification-strategi lander under det valgte navnet i Custom-kategorien. Bruk det hvis du har gjort deg umak og ikke vil sitte og fomle på nytt ved neste prosjekt. Spesielt praktisk for tilbakevendende oppsett som „Drone 4K« eller „Innendørs rask«. Doble navn er teknisk tillatt, men praktisk forvirrende — velg heller noe sigende.

I2 Preset Name TextField

 HVOR

Save-popover → tekstfelt „Preset Name«.

 TEKNISK

Enkelt tekstfelt med avrundet ramme, bred form. Verdien overtas som preset-navn ved klikk på save-knappen. Ingen lengdebegrensning i UI-en, men det lagrede navnet må være JSON-kodbart og visningsbart i UI-listene — emoji og spesialtegn fungerer. Innholdet nullstilles automatisk til en tom streng ved åpning av popoveren. Save-knappen forblir deaktivert så lenge feltet etter trim er tomt. Det er ingen auto-suggest og ingen forhåndsutfylling med navnet på den aktuelt aktive preseten.

KORT FORKLART

Her taster du navnet på presetet ditt. Velg noe sigende som „Drone 4K 30fps« eller „Innendørs rask« — det hjelper deg senere med å finne det igjen i Custom-kategorien. Emoji og spesialtegn er tillatt, en hard lengdebegrensning er det ikke. Så lenge feltet er tomt eller bare inneholder mellomrom, forblir save-knappen grånet ut. Ved fornyet åpning av popoveren er feltet tomt igjen — det er ingen forhåndsutfylling med det aktive preset-navnet.

I3 Cancel-knapp (save-dialog)



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



Lukker popoveren uten å lagre. Forkaster tekstfeltets innhold — ved neste åpning nullstilles det igjen til tomt via save...-knapp-logikken (I1). Standard button-stil, ingen bekreftelsesdialoger, ingen hotkeys. Den aktuelle TrainingConfig forblir uendret fordi save-stien slett ikke ble utført.

KORT FORKLART

Lukker save-popoveren uten å lagre noe. Hvis du har ombestemt deg, taster feil eller åpnet dialogen ved et uhell — bare klikk Cancel. Den aktuelle treningskonfigurasjonen forblir uendret fordi det ennå ikke er skrevet noe. Ved neste åpning av popoveren starter navnefeltet tomt igjen. Ingen sikkerhets-spørsmål, ingen hotkey — bare klikk og vekk.

I4 Save-knapp (save-dialog)



Save-popover → Save-knapp (til høyre, prominent stil).



Utløser selve persistasjonen. Validerer enda en gang ikke-tomt navn (defensiv sjekk) og skriver deretter den aktuelle TrainingConfig som JSON inn i app-lageret. Lukker deretter popoveren. Blå fremhevet, grånet ut så lenge tekstfeltet er tomt. Hvis lagringen mislykkes (f.eks. fordi app-lageret er fullt — veldig usannsynlig), er det for øyeblikket ingen synlig feildialog; presetet ville da bare ikke dukke opp ved neste app-start.

KORT FORKLART

Med klikk på Save overtar du navnet og skriver den aktuelle oppsetningen din vekk som ny preset. Popoveren lukker, presetet dukker straks opp i Custom-kategorien av preset-listen og kan fra nå av aktiveres ved klikk. Knappen er blå fremhevet (`borderedProminent`) og forblir grånet ut så lenge navnefeltet er tomt. Hvis lagringen mislykkes (f.eks. UserDefaults full), er det ingen synlig feildialog — presetet ville da bare mangle ved neste app-start.

I5 Export...-knapp



Inspector → Presets-seksjon → action-bar → Export...-knapp.



Eksporterer det aktuelt valgte presetet som `.radiancepreset` -fil (internt JSON). Deaktivert hvis ingen preset er valgt. Ved klikk åpner appen en save-dialog med forhånds-gitt filnavn (preset-navn + `.radiancepreset` -extension). Det lagrede formatet inneholder den komplette TrainingConfig pluss metadata (navn, kategori, ID, built-in-flagg). Dobbelklikk i Finder åpner appen — men **ikke** automatisk importen; brukeren må bruke import-knappen (I6).

KORT FORKLART

Velg en preset i listen og klikk Export — så kan du lagre den som `.radiancepreset` -fil og f.eks. sende den til en kollega eller overføre den til en annen Mac. Mottakeren laster den inn der borte med Import...-knappen (I6). Fungerer like godt for built-ins og dine egne custom-presets. Knappen er grånet ut så lenge det ikke er klikket noe i listen. Tips: via kontekstmenyen (I8) går det enda raskere — der trenger du ikke velge presetet først.

I6 Import...-knapp



Inspector → Presets-seksjon → action-bar → Import...-knapp.



Åpner en fildialog som bare tillater `.radiancepreset` -filer (multivalg deaktivert). Ved valg lastes JSON-filen, valideres og settes inn i Custom-kategorien — med ny intern ID, slik at det ikke oppstår kollisjoner med built-ins. Importen setter automatisk kategorien til Custom, selv om det eksporterte presetet opprinnelig var en built-in. Skadde eller inkompatible filer med eldre skjemaversjon avvises i stillhet uten feildialog (konsoll-log gir imidlertid beskjed).

KORT FORKLART

Laste en `.radiancepreset` -fil fra disk. Nyttig hvis noen sender deg en utprøvd oppsetning, eller hvis du selv vil holde favorittpresetene dine synkron på tvers av flere Mac'er. Importerte presets lander alltid i Custom-kategorien — også hvis de opprinnelig ble eksportert fra built-ins. Skadde eller foreldede filer ignoreres i stillhet; i konsoll-loggen står da grunnen. Multivalg i dialogen er deaktivert, så bare én fil per klikk.

I7 Preset-linje (klikkaktivering)



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



Klikk på en preset-linje erstatter alle felter i TrainingConfig med verdiene fra presetet, husker ID for det aktive presetet og nullstiller modified-status. Aktiv-haken foran linjen dukker bare opp når presetet er valgt OG umodifisert. Så snart en verdi i TrainingConfig endres (slider, stepper, toggle i de andre Inspector-seksjonene), dukker det opp et oransje „Modified«-badgje bak navnet. Innebygde presets kan ikke overskrives — ved modifikasjon må en egen kopi opprettes via save-knappen (I1).

KORT FORKLART

Klikk på en linje aktiverer presetet og overtar alle der lagrede verdier i de aktuelle treningsinnstillingene. Haken foran navnet viser hvilken preset som er aktiv akkurat nå. Så snart du deretter justerer en slider, stepper eller toggle i de andre seksjonene, dukker det opp et oransje „Modified«-badgje bak navnet — fordi oppsetningen din nå avviker fra presetet. Built-in-presets kan ikke overskrives; hvis du vil beholde endringer, så opprett en egen kopi via Save...-knappen (I1) eller dupliser presetet (I9).

I8 Kontekstmeny „Export...«



Høyreklikk på hver preset-linje → første oppføring „Export...«.



Identisk funksjonalitet som I5 (Export...-knapp), men mer bekvemt tilgjengelig — uten at presetet må velges først. Eksporterer direkte det presetet som ble klikket på i linjen. Fungerer likt for alle preset-kategorier (built-in eller Custom), ingen begrensning. Eksporten inneholder built-in-flagg og originalkategori, men ved re-import mappes kategorien som beskrevet under I6 til Custom.

KORT FORKLART

Rask vei til eksport — høyreklikk på det ønskede presetet og velg „Export...«. Sparer omveien via klikk-først og så trykk på Export...-knappen. Fungerer likt for alle kategorier, også for built-ins. Den genererte .radiancepreset -filen er identisk med den fra I5; ved senere re-import lander den automatisk i Custom-kategorien.

I9 Kontekstmeny „Duplicate«



Høyreklikk på hver preset-linje → andre oppføring „Duplicate«.



Kloner presetet inn i Custom-kategorien. Genererer en ny intern ID, henger på „ Copy« til navnet og lagrer kopien. Fungerer også for built-in-presets — klonen er da redigerbar. Originalen forblir urørt. TrainingConfig kopieres verdi-for- verdi (JSON-round-trip), slik at det ikke består referanseforbindelser mellom original og kopi.

KORT FORKLART

Skaper en redigerbar kopi av en preset i Custom-kategorien. Praktisk hvis du f.eks. vil ha det innebygde „Quality«-presetet som utgangspunkt og så bare flytte SSIM-glidebryteren litt. Workflow: duplicate, gi nytt navn (kontekstmeny eller ny Save...-kjøring), tilpass, ferdig. Originalen forblir urørt — du kan alltid vende tilbake til den. Fungerer også for built-ins, noe som er den eneste måten å overta verdiene deres som basis og samtidig gjøre dem redigerbare.

I10 Kontekstmeny „Delete«



Høyreklikk på egne preset-linjer → siste oppføring „Delete« (rød, destruktiv).



Bare synlig for custom-presets. Built-ins kan ikke slettes. Oppføringen er markert som destruktiv, vises rød i kontekstmenyen og plasseres etter en divider, slik at den ikke klikkes ved et uhell. Det finnes **ingen** bekreftelsesdialog — ett klikk sletter presetet umiddelbart. Det slettede presetet kan ikke gjenopprettes (Cmd-Z virker ikke her — undo finnes i den aktuelle buildet bare for bildelisten, ikke for preset-operasjoner). Var det slettede presetet nettopp aktivt, forblir den aktuelle TrainingConfig uendret, bare det aktive preset-valget nullstilles.

KORT FORKLART

Slette egne presets. Ved built-ins (Quick, Preview, Balanced, Quality, Ultra Detail, Drone / Aerial, 360° Walkaround, Photo / Object osv.) er „Delete« slett ikke synlig — dem kan du ikke komme til å slette ved et uhell. Advarsel: det er ingen sikkerhetsspørsmål og ingen undo, ett klikk og presetet er borte. Hvis du er i tvil, så ta først en sikkerhetskopi til disk via Export... (I5/I8) — den kan du alltid importere igjen. Var presetet nettopp aktivt, forblir TrainingConfig uendret, bare haken forsvinner.

I11 Kategori-header (utfold/sammenlegg)



HVOR

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



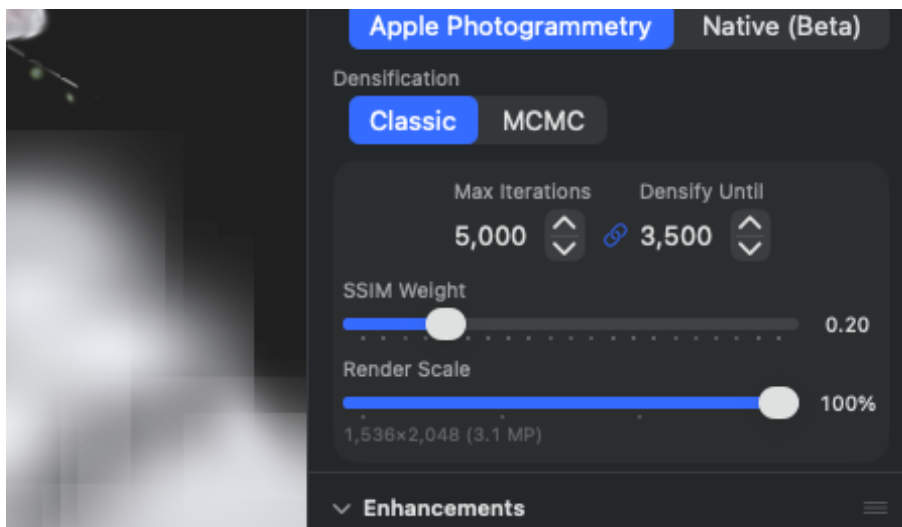
TEKNISK

Klapp-status per kategori med forskjellig default: den kuraterte Capture Class-gruppen starter **foldet ut**, Classic, MCMC, Hybrid og Custom starter **sammenfoldet**. Status persisteres ikke — ved app-omstart er alle kategorier igjen i default-tilstand. Chevron-pilen roterer animert. Tallet til høyre i headeren viser antallet presets i denne kategorien. Klikk-hit-arealet omfatter hele headerområdet.

KORT FORKLART

Folde kategorier inn og ut for å holde preset-listen oversiktlig. Ved app-start er Capture Class-gruppen åpen, Classic, MCMC, Hybrid og Custom er lukket. Klikk på headeren (hele området er klikkbart), og listen kjører opp eller igjen med en kort chevron-animasjon. Det lille tallet til høyre viser hvor mange presets som ligger i kategorien. Etter omstart av appen er default-tilstanden tilbake — appen lagrer bevisst ikke denne klapp-innstillingen.

Treningskonfigurasjons-seksjon (I12–I22)



Figur 11: Crop kun training-konfigurasjons-seksjon — 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 sentrale håndtakene: hvilket SfM-backend som skal beregne, hvordan densification arbeider, hvor mange iterasjoner, hvor stor SSIM-vektingen er. Ved MCMC-strategi dukker det opp to ekstra toggles („MCMC Quality« og „Auto-scale by scene») som skjules i Classic-modus. Ved Native-SfM-backend kommer FOV-override-feltet til, som bare trengs for video-frames uten EXIF-brennvidde.

I12 Camera Alignment-velger



Inspector → treningskonfigurasjon → Camera Alignment (segmentert velger øverst).



Segmentert velger med to alternativer: Apple Photogrammetry og Native (Beta). Valget bestemmer det benyttede SfM-backendet ved neste kamerarekonstruksjon. Det påvirker samtidig hvilke andre Inspector-elementer som er synlige: Native viser dessuten FOV-override (I13), som bare trengs ved EXIF-løse video-frames. Merk: ved veldig store outdoor-opptak kan du laste inn resultatet fra et eksternt verktøy (Metashape eller COLMAP) via workspace-import — se kapittel 1 (M5) og kapittel 9 (Q3, Q6).

KORT FORKLART

Her velger du hvordan kameraposisjonene rekonstrueres — den viktigste bryteren for sluttkvaliteten. Apple Photogrammetry er den raske standarden og holder fullt ut for de fleste objektskanninger. Native (Beta) er den App Store-konforme egenutviklingen, bra for orbits og turntable-scener, og krever ved EXIF-løse video-frames FOV-override (I13). Ved veldig store outdoor-sett kan du alternativt beregne kameraene i Metashape eller COLMAP og laste resultatet via workspace-importen. Detaljer og anbefalinger per scenetype finner du i kapittel 9.

I13 FOV Override-felt (Native SfM)



Inspector → treningskonfigurasjon → FOV Override (bare synlig ved Camera Alignment = Native).



Numerisk tekstfelt (range 0-170°), default 0 = automatisk bestemmelse fra EXIF eller heuristikk. Manuell inntasting er nødvendig hvis input-bildene er trukket ut fra en video som ikke inneholder brennvidde-metadata. Typiske verdier: iPhone Wide ≈ 73°, DJI Mavic Wide-Crop ≈ 70°, drone med fullformatsensor ≈ 84°. Verdien klampes til [0, 170] — verdier utenfor stues direkte tilbake. Påvirker bare den native SfM-pipelinen (Q4/Q5); Apple Photogrammetry ignorerer denne verdien helt.

KORT FORKLART

Hvis bildene dine ikke har EXIF (typisk ved uttrukne video-frames), så tast inn her kameraets horisontale synsfelt i grader. Tommelfingerverdier: iPhone Wide ≈ 73°, DJI Mavic Wide-Crop ≈ 70°, drone med fullformatsensor ≈ 84°. En 0 lar appen gjette selv — det går ofte bra, men kan gå skjevt ved sjeldne optikker. Verdier over 170° stues automatisk tilbake. Feltet er bare synlig og bare virksomt hvis du har valgt Native som Camera Alignment (I12) — Apple Photogrammetry ignorerer det helt.

I15 Densification-velger



HVOR

Inspector → treningskonfigurasjon → Densification (segmentert velger, alltid synlig).



TEKNISK

Bytter mellom de to densification-strategiene: Classic (original 3DGS-metode med clone/split/prune og gradient- terskel) og MCMC (Stochastic Gradient Langevin Dynamics med relocation, NeurIPS 2024). Ved bytte fra Classic til MCMC setter appen automatisk MCMC-spesifikke felter til utprøvde default-verdier (reg-weights = 0, MCMC-cap-multiplier 3.0, sample-/ noise-schedule). Uten denne automatiske initialiseringen led sesjoner med gamle presets under 1.4.4-MCMC-collapse-buggen (460K→5 gaussians, watchdog-kill). Velger-valget bestemmer dessuten hvilke Inspector-elementer som er synlige — ved MCMC dukker I16/I17 opp. Detaljert feltvirkning i kapittel 6, T11–T16 (Classic) og T61–T73 (MCMC).

KORT FORKLART

Det sentrale strategivalget for hvordan gaussian-antallet vokser. Classic er godt tunet fra 459 eksperimenter, leverer raske resultater av høy kvalitet og trenger ikke å kjenne MCMC-feltene. MCMC er den nyere tilnærmingen (NeurIPS 2024), mer reproduserbar og unngår manuell threshold-justering — til gjengjeld regner den ca. 6× lengre ved sammenlignbar kvalitet. Ved bytte til MCMC setter appen automatisk trygge defaults, slik at treningen ikke ender i 1.4.4-collapse. Detaljer om strategifeltene står i kapittel 6 (T11–T16 Classic, T61–T73 MCMC).

I16 MCMC Quality-toggle



HVOR

Inspector → treningskonfigurasjon → MCMC Quality (bare ved Densification = MCMC).



TEKNISK

Slår gradient-accumulation til 2 trinn (aktiv) eller 1 trinn (inaktiv). Akkumulerer gradientene fra to påfølgende kamera-views før optimizer-step utføres. Empirisk (sesjon 33, V544a) reduserer det den endelige L1-feilen med ca. 6 % (0.0246 med Quality vs 0.0261 uten, ved 3-trial-gjennomsnitt på Horse-Full-MCMC). Prisen: doblet treningstid. Ved veldig lange treninger (200K iterasjoner) fører det til ytterligere 10+ minutters venting — så bare verdt det hvis de siste par prosent kvalitet virkelig er nødvendige. Påvirker bare treningen, ikke eksportformatet eller viewport-visningen.

KORT FORKLART

Quality-mode for MCMC med gradient-accumulation over to views. Gjør sluttresultatet empirisk ca. 6 % bedre (L1 0.0246 i stedet for 0.0261 i Horse-testen), koster til gjengjeld dobbelt så lang tid. Hvis du likevel kjører en 200K-MCMC-trening (gjerne 2 timer), kommer det enda en knapp time oppå. Verdt det ved endelige showcase-renderinger eller mot slutten av en quality-sweep-sesjon, i daglig workflow heller ikke. Bare synlig når Densification står på MCMC (I15).

I17 Auto-scale by scene-toggle



HVOR

Inspector → treningskonfigurasjon → Auto-scale by scene (bare ved MCMC).



TEKNISK

Hvis aktiv, skaleres den effektive max-gaussians-øvre grensen med SfM-init-point-count × MCMC-cap-multiplier (default 3.0). Eksempel: SfM leverer 250K initpunkter, basis-cap = 150K, multipliar 3.0 → effektiv øvre grense = $\max(150K, 750K) = 750K$. Hvis deaktivert, gjelder strengt bare basen. Ble innført til v1.4.5 fordi store outdoor-opptak med over 1000 frames og tilsvarende høy SfM-punkttetthet med den stive 150K-cap-default sultet ut densificationen — overflødigde punkter ble igjen, nye fikk ikke lov til å oppstå. Default OFF i custom-presets, ON i MCMC-built-ins. Påvirker bare ved treningstidspunkt, ikke i eksporten.

KORT FORKLART

Lar maksimumstallet av gaussians vokse med scenestørrelsen (mer presist: med antallet SfM-initpunkter). Ved små scener merker du nesten ingen forskjell, ved store outdoor-scener er det ofte avgjørende for kvaliteten — ellers „kveles« treningen fordi default-øvre-grensen på 150K er altfor lav for scenen. Ble spesifikt innført til v1.4.5 etter at veldig store outdoor-sett (over 1000 frames) synlig hang seg fast på cap-et. Ved MCMC-built-in-presets allerede slått på på forhånd; i egne presets slått av som default.

I18 Max Iterations-stepper



HVOR

Inspector → treningskonfigurasjon → GroupBox → Max Iterations.



TEKNISK

Stepper med range 1 000–100 000, skrittstørrelse 1 000. Bestemmer det samlede antallet optimizer-iterasjoner. Lineært korrelert med treningstiden (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 gir i gjennomsnitt knapt forbedring — diminishing returns. Ved endring trekkes Densify Until proporsjonalt med hvis link-funksjonen (I19) er aktiv (default-ratio: 0.5, dvs. Densify-Until = $\text{Max}/2$).

KORT FORKLART

Hvor mange treningssteg som kjøres — mer er bedre, men koster også lineært mer tid. Tommelfingerregel: 20 000 for god kvalitet, 40 000 for optimum ved Classic-strategi (over det gir det i gjennomsnitt knapt mer). MCMC krever markant mer, 200 000 er her standard. Dobling av iterasjoner dobler grovt treningstiden. Ved aktiv link-knapp (I19) trekkes Densify Until proporsjonalt med — praktisk alltid det du vil.

I19 Link/Unlink-knapp (Densify ↔ Iterations)



Inspector → treningskonfigurasjon → GroupBox → liten link-knapp mellom Max Iterations og Densify Until.



Toggle-knapp som fryser forholdet mellom Densify Until og Max Iterations. Ved aktiv (link-ikon fremhevet) trekkes Densify Until proporsjonalt med ved hver endring av Max Iterations. Ved unlink (link-plus-ikon) forblir verdiene uavhengige. Default er linked fordi det gjenspeiler den typiske korrelasjonen — hvis du trekker treningen til doble iterasjoner, vil du som regel også la densificationen kjøre proporsjonalt lengre. Forholdet beregnes ved innstilling av link-knappen ut fra den aktuelle verdien; et typisk forhold er 0.5 (Densify-Until = halvparten av iterasjonsantallet).

KORT FORKLART

Liten klammeknapp mellom Max Iterations og Densify Until. Når aktiv (link-ikon fremhevet) vander de to verdiene sammen — dobler du iterations, dobles også Densify Until i samme forhold. Hvis ikke (link.badge.plus-ikon), kan du sette dem uavhengig. Standard er linket fordi det gjenspeiler den typiske korrelasjonen — lengre trening krever som regel også lengre densification-fase. For 99 % av tilfellene la det være låst.

I20 Densify Until-stepper



Inspector → treningskonfigurasjon → GroupBox → Densify Until.



Stepper med range 500–50 000, skrittstørrelse 500. Bestemmer iterasjonsindeksen hvor det ikke lenger kommer nye gaussians til via clone/split (Classic) eller relocation (MCMC). Etter oppnåelse forfines kun posisjon og farge. Høyere verdier = flere gaussians = større fil, lengre per-iterasjon-tid (+30-60 % GPU-tid per skritt). Typiske verdier: 15K (til 30K max-iter), 20K (til 40K), 100K (til 200K MCMC). Ved aktiv link (I19) skaleres automatisk med. Virker forskjellig ved Classic vs MCMC: Classic stopper veksten helt, MCMC stopper relocation-logikken, men sample-/noise-adaptation kjører videre.

KORT FORKLART

Opp til hvilken iterasjon nye gaussians får legges til — ved Classic via clone/split, ved MCMC via relocation. Deretter handler det bare om farge- og form-forfining av de eksisterende punktene. Høyere = mer detalj, men også større fil og +30-60 % GPU-tid per skritt. Typiske verdier: 15K (til 30K max-iter), 20K (til 40K), 100K (til 200K MCMC). Henger normalt via link (I19) på Max Iterations — sjelden meningsfullt å koble fra det manuelt.

I21 SSIM Weight-sliden



Inspector → treningskonfigurasjon → GroupBox → SSIM Weight.



Slider 0.0–1.0 i 0.05-skritt, visning som „0.20“. Blander L1-loss (0.0) og SSIM-loss (1.0). L1 strammer lysstyrken per piksel, SSIM den strukturelle likheten (kanter, lokale statistikker). Default 0.2 er verdien fra det opprinnelige 3DGS-paperet (Kerbl 2023) og reverse-engineered som robust kompromiss i talrike sesjoner. Høyere verdier (0.5+) foretrekker detaljbevaring, men kan ignorere lokale lysstyrkefeil. Lavere verdier (< 0.1) fører til detaljtap ved skarpe kanter. SSIM-beregningen kjører i shaderen med et 11×11-gaussian-vindu. Ytelse: ved 0.0 (bare L1) er treningen ca. 8-12 % raskere fordi SSIM-beregningen i shaderen hoppes over.

KORT FORKLART

Hvor sterkt strukturell bildelighet (kanter, lokale mønstre) vektet i forhold til ren lysstyrkesammenligning. 0.2 er standarden fra det opprinnelige 3DGS-paperet og holder for nesten alle scener. Høyere (0.5+) ved fine strukturer som hår, pels eller vegetasjon — der hjelper mer strukturvekt. Lavere (0.0) gjør treningen ca. 8-12 % raskere fordi SSIM-beregningen i shaderen hoppes over, men koster detalj ved skarpe kanter. Den som ikke har en god grunn til endring, lar 0.2 stå.

I22 Render Scale-sliden



Inspector → treningskonfigurasjon → GroupBox → Render Scale.

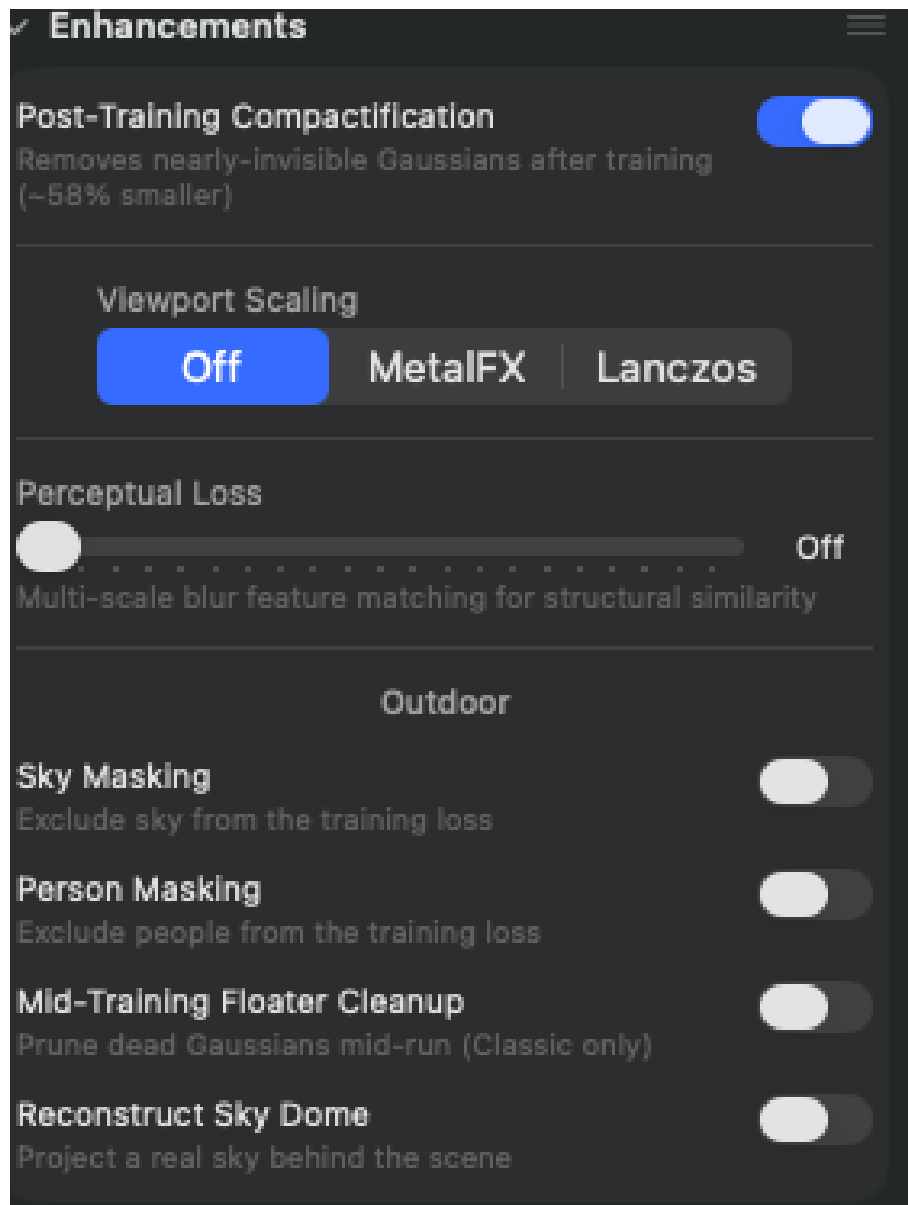


Slider 0.25–1.0 i 0.25-skritt, visning som „100 %«. Skalerer training-rendering-oppløsningen relativt til kildebilde størrelsen. Største håndtak på ytelse: 50 % reduserer GPU-tid med ca. 75 % (fordi 4× færre piksler), 25 % med ca. 94 %. Gradient-terskelen skales automatisk med. Under slideren dukker det opp en live-oppløsnings-visning i MP (f.eks. „2304×1296 (3.0 MP)«). Hvis den aktuelle verdien avviker fra den anbefalte, slås det inn i oransje skrift „— recommended: 50 %«. Anbefalingen sikter på ~3 MP effektiv oppløsning — det området som Apple Silicon GPU-er behandler mest effektivt. 4K-kildebilder får f.eks. automatisk 25 % anbefalt, FullHD-bilder 100 %. En endring utløser dessuten buffer-realloseringen.

KORT FORKLART

Med hvilken oppløsning treningen renderer — ett av de største ytelseshåndtakene. Full (100 %) gir beste kvalitet, men koster ved store bilder mye GPU-tid. Halv (50 %) sparer ca. 75 % GPU-tid fordi det beregnes fire ganger færre piksler — perfekt for 4K-kilder. Under slideren ser du den effektive oppløsningen i megapiksler; appen sikter på rundt 3 MP fordi det kjører mest effektivt på Apple Silicon. Hvis verdien din avviker fra det, slår appen på et oransje „recommended«-hint — som regel lønner det seg å følge.

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



Figur 12: Crop kun Enhancements-seksjon — tre rader: Post-Training Compactification (toggle på), Viewport Scaling (segmentert picker Off/MetalFX/Lanczos), Perceptual Loss (slider på „Off«). Hver rad med undertittel forklarer funksjonen

Enhancements-seksjonen grupperer tre features som forbedrer bildekvaliteten uten å endre selve kjerne-trenings-loopet. De første to (I26–I27) er **post-training-** eller **viewport-trinn**: compactification rydder opp etter treningslutt, viewport-skaleringen er en ren viewport-renderer som ikke påvirker den løpende treningen. Perceptual Loss (I29) er tross seksjonstilhørigheten en treningsbestanddel — den aktiveres under treningen som ekstra loss-term, derav adskillelsen fra viewport-reglerne via en divider.

Fra og med v1.6 har seksjonen også en Outdoor-gruppe (I42–I44: Sky Masking, Mid-Training Floater Cleanup, Reconstruct Sky Dome) — treningsalternativer mot himmelfloaters som tidligere lå i Innstillinger-vinduet og nå sitter her per prosjekt.

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

Inspector → Enhancements → Post-Training Compactification.

 **TEKNISK**

Aktiverer V443-post-processing: etter avslutningen av treningsiterasjonene slettes gaussians med opacity under 0.01 (1 % synlighet). Empirisk reduserer det filstørrelsen med ~55-58 % ved null synlig kvalitetstap — fordi disse gaussians visuelt likevel ikke bidrar. Compactificationen kjører som GPU-compact- pass og varer avhengig av gaussian-count brøkdeler av sekunder til få sekunder. Påvirker ikke trenings-ytelsen. Hvis denne toggle er av, eksporteres også usynlige gaussians — bare relevant hvis du vil bruke formatet til enda et trenings-stage (Continue Training), ellers sløsing med lagringsplass.

 **KORT FORKLART**

Rydder opp etter treningen i gaussians du likevel ikke kan se (opacity under 1 %). Gjør eksport-filene omtrent halvparten så store (55-58 % størrelsesreduksjon) uten synlig kvalitetstap. Kjører som kort GPU-pass etter siste iterasjon, varer bare brøkdeler av sekunder til få sekunder. Bør praktisk alltid være på — den eneste grunnen til å slå det av er hvis du senere vil fortsette treningen via Continue Training og må beholde også usynlige gaussians. Ved normale eksport-workflows la det bare stå.

I27 Viewport Scaling-picker

Inspector → Enhancements → Viewport Scaling (segmentert picker med tre alternativer: Off, MetalFX, Lanczos).



En enkelt segmentert picker som velger viewport-upscaler — de tre alternativene er **gjensidig utelukkende**. Hvis trenings-oppløsningen (via I22 Render Scale) er lavere enn viewport-størrelsen, skalerer den valgte modusen det rendrede frame opp til visningsstørrelsen. **Off** = enkel bilineær strekking. **MetalFX** = Apples ML-baserte spatiale upscaler, det skarpeste alternativet (ML-modellen er optimalisert for skarpe kanter), overhead ca. 1-2 ms per frame på M3-GPU-er. **Lanczos** = Apples Metal Performance Shaders med 8-tap-sinc-resampling, klassisk uten ML, minimal overhead (< 0.5 ms), kvalitet under MetalFX, men uten den ML-typiske „mykgjøringen« av fine linje-strukturer. Renderer-pipelinen omkonfigureres live ved bytte — synlig umiddelbart, uten omstart. **Bakgrunn:** tidligere var dette to separate toggles (MetalFX

1. Lanczos) som kunne være på samtidig — en motstridende tilstand

der MetalFX stille gikk forbi Lanczos. Pickeren fjerner denne tilstanden; en eventuell „begge-på«-tilstand arvet fra eldre sesjoner heler seg selv til MetalFX ved neste bytte. Virker **bare** på live-viewporten, ikke på rendrede eksporter (orbit-video, skjermbilder) — de rendres i full kildeoppløsning.

 KORT FORKLART

Skjerper live-bildet i viewporten — spesielt nyttig hvis du arbeider med redusert trenings-oppløsning (Render Scale 50 %, se I22). Tre trinn der bare ett er aktivt om gangen: „Off« strekker bare pikslene, „MetalFX« bruker Apples machine-learning for de skarpeste kantene (praktisk alltid det beste valget), „Lanczos« er det klassiske filteret uten ML — ta det som fallback hvis MetalFX glatter ut linjer eller viser artefakter i en scene. Griper live, uten omstart. Virker bare i live-viewporten, ikke på eksporterte orbit-videoer eller skjermbilder — de rendres i full kildeoppløsning. I motsetning til før kan du ikke lenger ved et uhell velge to modi samtidig.

I29 Perceptual Loss-slider

Inspector → Enhancements → Perceptual Loss.



Slider 0.0–0.2 i 0.01-skritt, visning ved 0.0 som „Off«, ellers som „0.05» osv. Aktiverer en ekstra loss-term som sammenligner multi-skalert gaussian-blur av renderingen med ground-truth-bildet (3 blur-skalaer). Fanger strukturelle forskjeller som L1+SSIM alene ikke gjenkjenner. V460-implementasjon. Empirisk forbedrer en verdi på 0.05–0.1 L1-scoren i sesjoner med et par prosent, men koster ~5 % treningstid (ekstra forward-pass gjennom blur-kjernene). Over 0.15 blir treningen ustabil og L1 forverres igjen (loss-termen dominerer optimeringen). Virker **under** treningen, ikke i post-processing — tross plassering i „Enhancements«-seksjonen er det altså ikke en ren oppgradering etterpå.

 KORT FORKLART

En ekstra loss-andel som sjekker strukturell billedlikhet via tre forskjellige uskarphetsnivåer. Hjelper spesielt ved scener med fine strukturer som hår, stoff eller vegetasjon, fordi den fanger mønstre L1+SSIM alene ikke ser. Mindre verdier er tryggere — 0.05 til 0.1 er sweet spot, over 0.15 blir treningen ustabil og loss-en forverres igjen. Ved 0 (Off) er funksjonen helt av og koster ingenting; aktiv sluker den ca. 5 % treningstid for det ekstra forward-passet gjennom blur-kjernene. Virker tross „Enhancements«-seksjonen direkte under treningen, ikke først i post-processing.

I42 Sky Masking

HVOR

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

TEKNISK

Aktiverer pre-training Apple-Vision-basert himmel-piksel-segentering. Før treningsstart ekstraheres himmelregionen for hvert input-kamera via Apple-Vision-Foreground-Mask (Sky = Background) og tilknyttes det respektive kameraet som per-piksel-maske. Under treningen multipliseres loss-bidraget per piksel med komplementet av himmelmasken — himmelpiksler bidrar 0 til gradienten, slik at gaussians som projiserer inn i himmelen ikke mottar optimeringssignaler og dermed ikke blir „tettere« eller „lyser«. Reduserer floaters (mørke klumper i himmelen) ved outdoor-/drone-scener signifikant. Koster ~3% L1-regresjon ved klassisk 40K-trening (se [memory/dev_outdoor-floater-reduction.md](#)). Bare meningsfull ved outdoor-scener med klart gjenkjennelig himmel; ved innendørs scener eller hvit bakgrunn identifiserer himmel-segenteringen feilaktige områder og blokkerer gyldige loss-signaler. Verdien lagres nå per prosjekt (ikke lenger app-globalt) og følger presetet / scenefilen.

KORT FORKLART

Ved outdoor-opptak med himmel i bildet oppstår det ofte sorte eller fargede klumper i himmelen — såkalte „floaters«. Dette alternativet registrerer automatisk hvor himmelen er, og sier til treningen: „La himmelen være i fred.« Fungerer veldig bra ved droneflyginger og landskapsscener. Ved innendørs rom eller mørke bakgrunner kan det forringe bildet — så slå bare på hvis det faktisk er ekte himmel å 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` (per prosjekt, `@DefaultFalse`). Default: `false`.



Aktiverer ved klassisk 40K-trening (preset „P4 Quality«) to ekstra density-control-passes: ved iterasjon 20,000 og ved iterasjon 30,000. Begge passes gjennomfører alle gaussians etter tre kriterier: (a) veldig lav opacity (standard 0.005), (b) liten screen-space-størrelse, (c) ingen loss-bidrag i de siste 1000 iterasjonene. Gaussians som oppfyller alle tre betingelsene, purges. Effekt: ~5–15% færre gaussians ved treningsslutt, synlig færre mørke klumper i himmelen ved drone-/outdoor-scener. Koster ~1–3% L1-regresjon ved nærbilde-indoor-scener, derfor ikke aktivert som default. Verdien huskes på tvers av omstart (i motsetning til S7). De to cleanup-iterasjonene (20K, 30K) er hardt definert og kan p.t. ikke endres via UI; ved kortere treninger (f.eks. P2 Preview 5K) har toggle-en ingen effekt fordi den aldri når iterasjonsmerkene. **Ny:** toggle-en er bare betjenbar når det aktive presetet bruker **Classic**-densifieren (`densificationStrategy == .classic`). Ved MCMC eller Hybrid blir den **disabled** og en inline-hint vises, fordi disse strategiene behandler døde gaussians selv uansett (MCMC via relocation, Hybrid via kombinert reloc-/noise-logikk) — de manuelle cleanup-passene ville der vært virkningsløse eller kontraproduktive. Kode-referanse: `RadianceKitApp.swift`, General-fanen. Detaljer: `memory/dev_outdoor-floater-reduction.md`.

KORT FORKLART

Under treningen oppstår det noen ganger „døde« gauss-punkter som ikke lenger bidrar til bildekvaliteten, men opptar minne. Dette alternativet rydder opp to ganger under en lang trening (ved 20K og 30K iterasjoner) og fjerner disse likene. Ved outdoor-scener med himmel er det spesielt nyttig fordi det er der de fleste floaters samles. Ved små treninger eller nærbilder av møbler er det heller ikke nødvendig. Bryteren lar seg bare slå på når presetet ditt bruker Classic-densifieren — ved MCMC- eller Hybrid-presets er den grånet ut (med kort forklaring), fordi de rydder opp de døde punktene sine selv.

I 44 Reconstruct Sky Dome**HVOR**

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

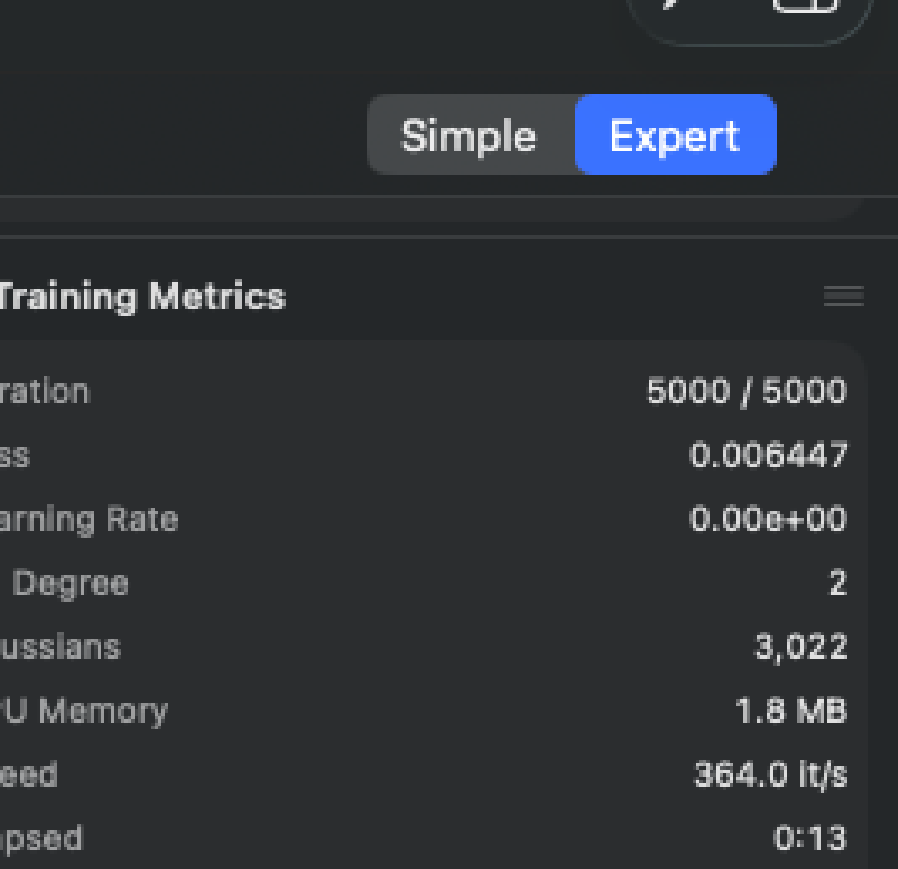
TEKNISK

Aktiverer pre-training sky-dome-projeksjonen (V549e MVP). Etter SfM og før treningsstart ekstraheres for hvert input-kamera den i S7 felles brukte Apple-Vision-himmelmasken fra bildet, himmelpikslene av-projiseres med kamera-intrinsics på en virtuell kuleoverflate (standard radius 8x sceneradius). På denne kulen initialiseres ~5000 nye gaussians med fargemiddelveier fra de projiserte himmelpikslene, veldig stor skalering (1.0 i sceneenheter) og start-opacity 0.95. Disse 5000 gaussians er ikke en sky-mask i klassisk forstand — de trenes som alle andre, men holdes via den høye start-opacity i et tynt skall. Resultat: ved 360°-novel-views i outdoor-/drone-scener fremstår faktiske himmelfarger og skyformasjoner i stedet for mørke konfettiklumper. Verdien huskes på tvers av omstart. Bare meningsfull ved outdoor-scener med minst 360°-kameradekning; ved rene object-captures uten himmel har det ingen effekt. Status: eksperimentell, bredere A/B-validering på tvers av ytterligere outdoor-sett gjenstår.

KORT FORKLART

I stedet for at treningen prøver å „gjette« himmelen ut fra de få synlige pikslene (noe som fører til floaters), projiserer RadianceKit himmelpikslene direkte på en virtuell kule rundt scenen før treningen starter. Når du så dreier den ferdige scenen i 360°, ser du ekte himmel i stedet for sorte klumper. Fungerer bare ved outdoor-opptak der det faktisk er himmel i bildet. Ved stueskanninger eller studioopptak bringer det ingenting.

Metrikk-seksjon (I30–I38)



| Metric | Value |
|---------------|-------------|
| 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-seksjon etter avsluttet trening på bukett (5K iterasjoner, 2 991 gaussians endelig) — tabell med treningsmetriker (Iteration, Loss, SSIM Loss, Combined Loss, Gaussian Count, Learning Rate, Elapsed, ETA)

Mens en trening kjører, viser metrikk-seksjonen ni live-verdier fra trenings-loopet. Før start av en trening er seksjonen tom („Start training to see live metrics«). Alle verdier oppdateres hver ~30 iterasjoner (oppdateringsfrekvens for strømmen). Seksjonen er read-only — ingen elementer kan klikkes eller endres. For dypere analyse trekkes JSONL-treningsloggene under `~/Documents/RadianceKit/Logs/` inn (skript `python3 scripts/analyze_logs.py best 5`).

I30 Iteration



HVOR

Inspector → Metrikker → Iteration. Read-only.



TEKNISK

Visning i formatet „4523 / 40000” — aktuell iterasjon over totalt planlagte iterasjoner. Teller synkront med trenings-loopet som pusher verdiene hver ~30 iter. Det andre tallet svarer til max-iterations-verdien på start-tidspunktet; det endres ikke lenger, selv om brukeren justerer stepperen deretter — den løpende kjøringen bruker sin egen snapshot-kopi. Hvis appen via training-menyen legger på iterasjoner (Continue Training +5K/+10K/+20K), økes nevneren.

KORT FORKLART

Hvor treningen står akkurat nå. „4523 / 40000” betyr: 4523 av 40 000 skritt er igjennom, altså ca. 11 % ferdig. Tallet til venstre teller opp i sekundtakt; hvis det står stille i minutter, henger treningen — som regel et tegn på GPU-throttling eller en konkurrerende app. Tallet til høyre svarer til max-iterations-verdien (I18) ved treningens start og endres ikke lenger, selv om du justerer stepperen senere. Ved Continue Training (+5K/+10K/+20K) vokser den med de ekstra skrittene.

I31 Loss



HVOR

Inspector → Metrikker → Loss. Read-only.



TEKNISK

Float-verdi med seks desimaler (f.eks. „0.024385”). Måler den kombinerte L1+SSIM-tapet (blandingen kontrolleres via I21 SSIM Weight) pluss valgfritt Perceptual Loss (I29) og andre regularizers. Skalaen er ikke absolutt, men sceneavhengig — krever for de fleste sammenligninger samme datasett. Typiske sluttverdier ved gode konfigurasjoner:

- 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)
- Innendørs leiligheter: 0.018–0.025

Verdier over 0.10 etter 5K iterasjoner indikerer SfM-problemer (dårlige kameraposisjoner) — avbryt og gjør SfM på nytt.

KORT FORKLART

Hvor mye det rendrede bildet fortsatt avviker fra originalen — kombinert av L1, SSIM og evt. Perceptual Loss. Mindre er bedre. Under 0.03 er som regel veldig bra, under 0.05 fortsatt greit, outdoor-scener ligger geometribetinget heller ved 0.03–0.06. Over 0.10 etter flere tusen iterasjoner er et varselssignal — som regel skyldes det kamerarekonstruksjonen (SfM har ikke klaffet pent). Skalaen er ikke absolutt, men sceneavhengig; gjør bare sammenligninger innen samme datasett. Hvis tallet plutselig hopper oppover, har det som regel oppstått en gradient-explosion-hendelse.

I32 Learning Rate

Inspector → Metrikker → Learning Rate. Read-only.



Visning i scientific notation (f.eks. „1.60e-04”). Aktuell læringsrate for posisjons-parametrene (3DGS har seks uavhengige LR-er for posisjon, SH-DC, SH-rest, opacity, scale, rotation — her vises posisjons-LR-en som representativ verdi). Default-startverdi 1.6e-4 som over en eksponentiell decay synker ned til $\sim 1.6e-6$ i treningens slutt. Forfallet kan justeres via LR-schedule-feltet i treningskonfigurasjonen (T-felt i kap. 6). Hvis LR-en blir uvanlig høy (f.eks. 1e-3 eller mer etter 10K iterasjoner), kan det indikere en feillastet konfigurasjon.

KORT FORKLART

Hvor store optimeringsskrittene er akkurat nå — konkret læringsraten for gaussian-posisjonene. Starter ved 1.60e-04 og synker eksponentielt til ca. 1.60e-06 ved treningslutt („1.60e-06” = 0.0000016). Forløpet kjører automatisk, du trenger ikke justere her. Hvis verdien etter 10 000+ iterasjoner fortsatt er større enn 1e-3, er det sannsynligvis lastet en feilaktig config — avbryt trening og velg preset på nytt. Internt har 3DGS seks uavhengige læringsrater (posisjon, SH-DC, SH-rest, opacity, scale, rotation); her ser du bare posisjons-LR-en som stedfortreder.

I33 SH Degree

Inspector → Metrikker → SH Degree. Read-only.



Heltall 0-3. Spherical-harmonics-grad for fargepresentasjonen. Begynner ved 0 (bare DC-komponenten, dvs. retningsuavhengig farge per gaussian — altså bare en RGB-konstant) og stiger progressivt til 3 i treningens forløp. Standard-schedule hever graden ved 1000/2000/3000 iterasjoner med 1 hver. SH-3 tilsvarer 48 farge-koeffisienter per gaussian (3 RGB-channels × 16 SH-basisfunksjoner). Høyere SH-grad = mer retningsavhengig refleksjon (blanke overflater ser korrekt forskjellige ut under forskjellige synsvinkler), men også mer lagring og langsommere trening.

KORT FORKLART

Hvor kompleks fargefremstillingen per gaussian er akkurat nå. Starter ved 0 (bare en retningsuavhengig farge per punkt) og trekkes trinnvis opp til 3 — typisk ved iterasjon 1000, 2000 og 3000. Trinn 3 betyr 48 farge-koeffisienter per gaussian og tillater retningsavhengige refleksjoner, altså at blanke overflater fra forskjellige synsvinkler ser korrekt forskjellige ut. Du trenger ikke aktivt røre ved det, schedulen kjører automatisk. Høyere grad koster mer lagring og forsinker treningen litt — men det er prisen for realistiske høylys.

I34 Gaussians



HVOR

Inspector → Metrikker → Gaussians. Read-only.



TEKNISK

Aktuelt antall gaussians i modellen, formatert med locale-separator (f.eks. „524.318”). Vekst:

- Classic: starter ved SfM-init-punktene (typisk 50K-300K), vokser via clone/split inntil kort før Densify Until, deretter statisk inntil treningslutt (modulo pruning)
- MCMC: sample-punkter legges til inntil MCMC-cap-et, deretter bare relocation

Healthy sluttverdier:

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

Ved MCMC faller tallet til < 60 % av cap-et → anomali (collapse-indikator, peker på for aggressive regularizers).

KORT FORKLART

Hvor mange gaussian-punkter 3D-modellen har akkurat nå. Vokser under treningen inntil Densify Until (I20) er nådd; deretter forblir tallet praktisk talt konstant. Flere punkter = mer detalj, men også større fil og langsommere rendering i viewporten. 500.000 gaussians er en typisk middelvei for Classic-Quality på en mellomstor scene; MCMC Full lander avhengig av auto-scale (I17) ved 500K til 1.5M. Hvis tallet ved MCMC plutselig faller under 60 % av cap-et, er det en collapse-indikator — som regel for aggressive regularizers.

I35 GPU Memory



HVOR

Inspector → Metrikker → GPU Memory. Read-only.



TEKNISK

Estimat av gaussian-bufferens minnebruk som gaussian-count × 616 bytes (formatert i memory-style). 616 bytes er den empiriske størrelsen på en fullt utstyrt gaussian (posisjon, skalering, rotasjon, opacity, SH-koeffisienter grad 3, gradient-akkumulator). Visningen fanger **ikke** renderer-overhead (tile-buffer, sort-buffer, backward-buffer) — det reelle GPU-lagerbehovet ligger typisk 2-3× over denne verdien. 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 grense.

KORT FORKLART

Et estimat over hvor mye GPU-lager gaussians selv opptar — rundt 616 bytes per punkt. Det faktiske GPU-forbruket er 2-3× høyere enn vist, fordi rendereren også legger egne tile-, sort- og backward-buffers til. På en MacBook med 16-18 GB unified memory bør du holde deg under 500K gaussians; med M3 Max eller Studio (64+ GB) kan du lett kjøre 1.5M og mer. Hvis treningen plutselig krasjer, eller systemet swapper, er grensen som regel nådd her — skru Render Scale (I22) ned eller reduser Densify Until (I20).

I36 Speed

Inspector → Metrikker → Speed. Read-only.



Iterasjoner per sekund med en desimal („24.3 it/s«). Beregnes av treneren som glidende gjennomsnitt over de siste ~100 iterasjonene. Typiske verdier:

- 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 full oppløsning: < 10 it/s

Fallende speed i treningens forløp er normalt — flere gaussians = mer compute per iterasjon. Plutselige fall (f.eks. fra 30 → 5 it/s) indikerer GPU-thermal-throttling eller konkurrerende apper.

KORT FORKLART

Hvor raskt treningen kjører, i iterasjoner per sekund. Står typisk ved 20-50 it/s, ved redusert Render Scale (50 %) og små scener også 80-120 it/s. Faller gjennom treningen ganske naturlig fordi flere gaussians = mer arbeid per iterasjon. Plutselige fall (f.eks. 30 → 5 it/s) indikerer GPU-thermal-throttling eller konkurrerende apper — nettleserfaner med video, Time Machine-backup, Photos-indeksering. Å holde appen i forgrunnen og lukke bakgrunnsprogrammer hjelper ofte. Ved 1M+ gaussians og full oppløsning er under 10 it/s normalt.

I37 Elapsed

Inspector → Metrikker → Elapsed. Read-only.



Allerede forløpt tid som „4:23" (m:ss) eller „1:23:45" (h:mm:ss). Format-bytte fra 1 time. Måler bare den rene treningstiden, ikke de forutgående fasene (SfM-beregning, bilde-import). Ved pause/resume kjører klokken videre — det er altså wall-clock, ikke CPU-tid.

KORT FORKLART

Hvor lenge treningen allerede har kjørt, som ren stoppeklokke (wall-clock-tid). Format er „m:ss« til en time, deretter „h:mm:ss«. Ikke „CPU-tid«, men „hvor lenge har vi ventet« — så også pausetider teller med. Måler bare den rene treningsfasen, ikke den forutgående SfM-beregningen eller bilde-importen. Hjelpsom for sammenligning med ETA (I38) — hvis Elapsed markant overskrider den opprinnelige ETA, har treningen blitt langsommere et sted enn planlagt.

I38 ETA HVOR

Inspector → Metrikker → ETA. Read-only.

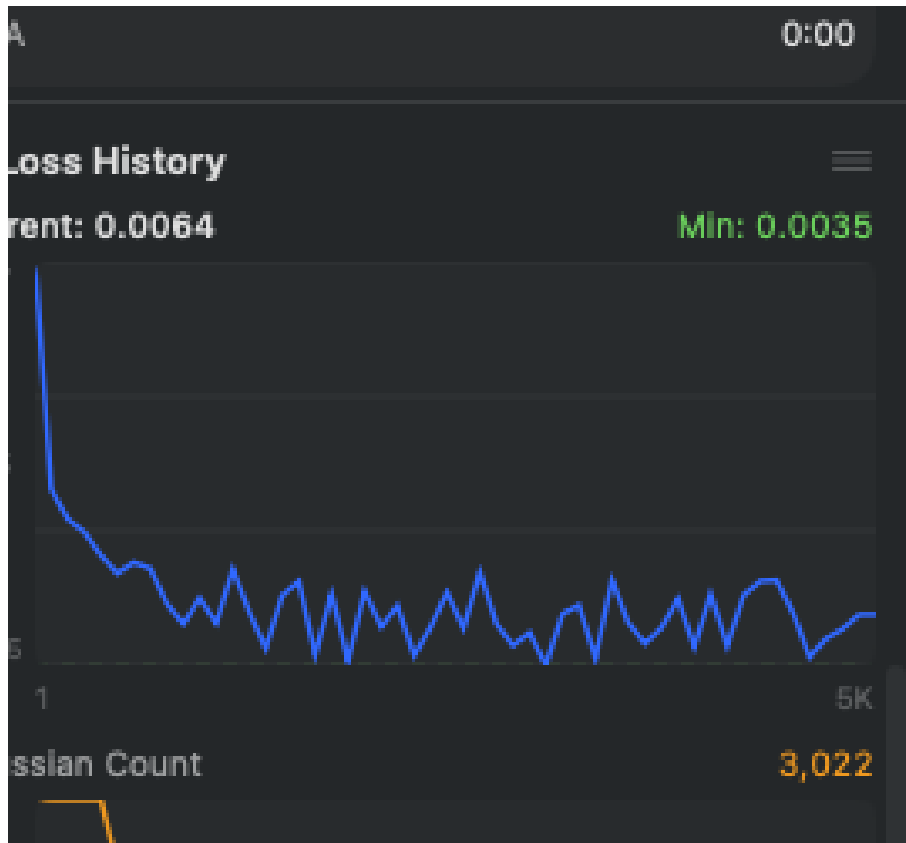
 TEKNISK

Estimert resttid som „17:42“ eller „1:12:35“. Beregning: $(\text{Max Iterations} - \text{aktuell iterasjon}) / \text{iterasjoner-per-sekund}$. Viser „-« når speed nettopp er null (helt i starten eller ved pause). Estimaten tilpasses **ikke** den typiske forlangsommings mot treningsslutt — spesielt ved MCMC og Classic med store Densify-Until-verdier har treningen tendens til å bli langsommere fordi flere og flere gaussians kommer inn i bildet. Reelt blir det typisk 10-20 % over start-ETA.

 KORT FORKLART

Hvor lenge det formentlig fortsatt må ventes — beregnet ut fra de gjenværende iterasjonene og den aktuelle speed (I36). Et grovt estimat: hvis Mac-en plutselig blir langsommere (flere gaussians fra densify-fasen, thermal-throttling, andre apper), kan det vare lengre enn vist. Appen regner ikke inn den typiske forlangsommings mot treningsslutt, så det reelle sluttet lander som regel 10-20 % over start-ETA. Legg 15 % ekstra inn, så stemmer det som regel. Viser „-« når speed nettopp er 0 (treningens start eller pause).

Loss-diagram-seksjon (I39–I41)



Figur 14: Crop kun Loss History-seksjon etter avsluttet trening — Current 0.0064, Min 0.0035 (grønn), blått forløp fra 0.027 (iterasjon 1) til 0.0035 (iterasjon 5K) med karakteristisk knekk omkring iter 200, derunder Gaussian Count-chart oransje

Loss-diagram-seksjonen visualiserer treningsforløpet over tid. Den består av to charts: et loss-curve-chart (stort, øverst, blått) og et gaussian-count-chart (mindre, nederst, oransje). Begge bygges opp live under treningen og persisterer inntil neste treningsstart. Før første trening er området tomt („Loss curve will appear during training«). Charts er rene SwiftUI-path-tegninger (ingen Swift-Charts-framework), slik at de også renderer flytende ved 100K+ punkter.

I39 Current Loss (visning)

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



Float-verdi av det siste loss-sample-punktet, formateret med fire desimaler. Identisk med I31 (Loss i metrics-seksjonen), bare her mer kompakt formateret. Kilden er loss-history — en liste som får en oppføring per ~30 iterasjoner. Bare endelige verdier tas opp i listen — NaN/Infinity (veldig sjelden, i tilfelle av gradient-explosion-bugg) filtreres.

KORT FORKLART

Den aktuelle loss-verdien i kortere skrivemåte enn i metrics-seksjonen (fire desimaler). Innholdsmessig identisk med I31, men her sitter visningen direkte ved loss-charten og gir deg den presise tall-verdien mens du iakttar kurven. Oppdateres som alle live-metrikker hver 30 iterasjoner. NaN- eller infinity-verdier (ekstremt sjeldne ved gradient-explosion-bugs) filtreres oppen automatisk bort. Nyttig for ikke å måtte hoppe til den andre seksjonen mens du kikker på diagrammet.

I40 Min Loss (visning)

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



Minimum av alle noensinne sette loss-verdier i den aktuelle treningskjøringen. Reberegnes live ut fra loss-history — ingen separat persistens. Viser med grønn skrift fordi „Min« = „best so far«. Den stiplede grønne linjen ved chartens nedre rand markerer denne Y-posisjonen visuelt. Ved continue-training-sesjoner starter min-springen på nytt — den gamle historikken erstattes i UI-en av den nye (ikke vedhengt). Hvis den aktuelle trainingen kjører dårligere enn den foregående, kan min-visningen altså være større enn det forrige sluttresultatet.

KORT FORKLART

Den laveste loss-verdien denne trainingen hittil har sett — grønt vist fordi „best so far«. Den stiplede grønne linjen ved chartens nedre rand markerer også denne posisjonen visuelt. Hvis den aktuelle kurven akkurat ligger markant over, er det med litt hell enda en forbedring; som regel er min imidlertid det sluttresultatet som senere interesserer deg. Ved continue-training-sesjoner starter min-springen på nytt fordi den gamle historikken erstattes i UI-en av den nye — min-verdien kan dermed se dårligere ut enn det forrige sluttresultatet.

I41 Gaussian Count Chart

HVOR

Inspector → loss-diagram → andre chart derunder (oransje). Read-only.

TEKNISK

Linjediagram av gaussian-antallet over trenings- iterasjonene. Kilde: gaussian-count-history (liste av (iter, count)-par, fylt av treneren hver ~30 iter). Y-skala dynamisk mellom minimum og maksimum av historikken. Ved Classic- strategi ser kurven typisk slik ut: jevnt stigende inntil Densify Until, deretter flat (med små pruning-utsving). Ved MCMC: bratt stigning inntil cap, deretter horisontal linje (relocation holder tallet konstant). Hvis kurven **faller** tross aktiv trening, pruner densificationen for aggressivt — tegn på feilaktige defaults eller en kjent MCMC-collapse-bugg (v1.4.4-hotfix-tema).

KORT FORKLART

Hvordan antallet gaussians utvikler seg over treningstiden — det mindre oransje chartet under loss-kurven. Ved Classic-strategi stiger linjen jevnt inntil Densify Until (I20) er nådd, deretter forblir den flat med små pruning-utsving. Ved MCMC svirrer den bratt opp til cap-et og forblir deretter horisontal fordi relocation holder tallet konstant. Hvis kurven tross aktiv trening plutselig knekker nedover, er densificationen for aggressiv ved pruning — klassisk tegn på MCMC-collapse-buggen fra v1.4.4. Da hjelper app-oppdatering eller bytte tilbake til Classic.

Hvordan leser man loss-kurven?

Loss-charten er det viktigste diagnoseverktøyet i Inspector — ingen annen indikator viser så direkte om treningen skrider nyttig frem eller henger seg fast. Den typiske sunne formen er et raskt fall i de første 1000-3000 iterasjonene (fra ~0.15 til ~0.05), etterfulgt av et langsomt, jevnt fall til treningsslutt (til 0.020-0.030). Logaritmisk virker kurven som en glatt diagonal.

Hva betyr et platå ved loss-en? Hvis kurven over flere tusen iterasjoner forblir flat, er det to mulige lesninger: (a) treningen er „konvergt« — loss-en kan ikke lenger falle signifikant fordi modellen er så god som den kan bli med de gitte dataene og innstillingene. Det er ønsket; det er „ferdig«. (b) Treningen „henger« — loss-en kan egentlig fortsatt falle, men optimeringen stagnerer (lokalt minimum, læringsrate for liten, densification av). Å skille: hvis loss-verdien ligger i et typisk godt område (0.020-0.030 ved indoor/object, 0.040-0.060 ved outdoor), og kurven har vært flat i 5K iterasjoner, er det konvergt. Hvis verdien er markant høyere enn ved sammenlignbare scener (f.eks. 0.08), henger den fast.

Advarsel: gaussian-platå ≠ loss-platå. Et platå i gaussian-antallet betyr **ikke** „trening er ferdig«. Det betyr bare at densificationen har sluttet å legge til nye punkter — enten fordi er nådd (Classic), eller fordi MCMC-cap-et er fullt. Treningen kjører etterpå videre og forfiner bare de eksisterende punktene. Selve „ferdig«-signalet leser du av loss-kurven og iteration-visningen (I30), ikke her.

Tommefingerregel for avbrytelse: Hvis loss-kurven etter 5000+ iterasjoner ligger over 0.08 og knapt faller mer, er SfM-rekonstruksjonen med stor sannsynlighet skjev. Avbryt trening, slå opp i kapittel 9 om det valgte SfM-backendet passer til scenen, bytt

evt. til COLMAP/Native, start så på nytt. Bedre å investere 10 minutter i bedre SfM enn 2 timers trening med dårlig kamera-justering.

Når strekke ut etter Inspector?

Hurtigreferanse: hvilken seksjon + hvilke controls for hvilket typisk use-case?

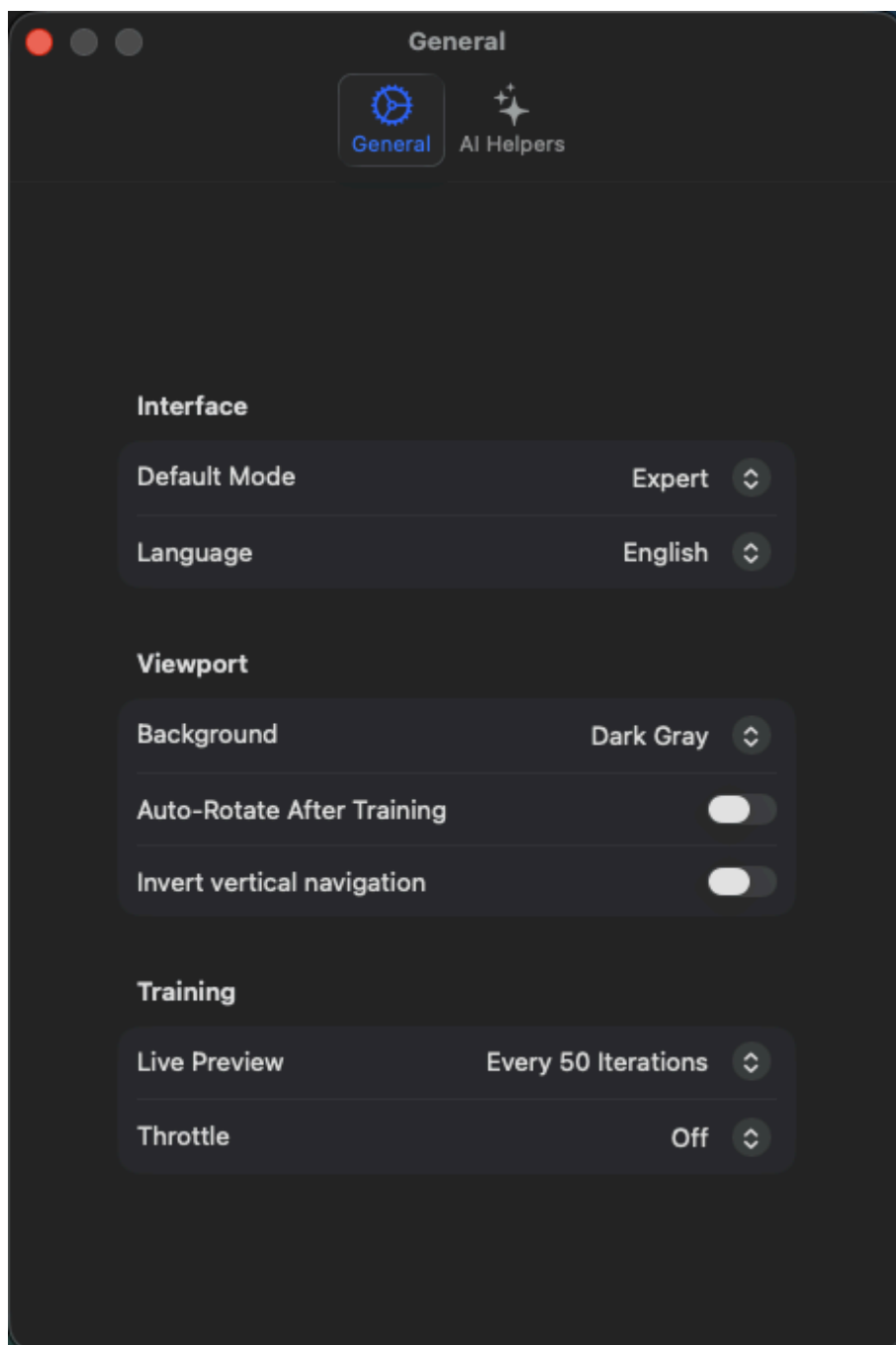
KAPITTEL

Kapittel 3 — Innstillinger

Innstillingsvinduet åpnes via `RadianceKit` → `Innstillinger...` eller standardsnarveien `⌘, .` Det inneholder to faner: **General** og **AI Helpers**. I motsetning til Inspector-verdiene fra kapittel 2 virker innstillingene fra dette vinduet **app-globalt** (på tvers av alle prosjekter) — de persisteres og overlever app-omstart. General-fanen grupperer tre innholdsmessige seksjoner: Interface, Viewport og Training. (De tre Outdoor-Floater-bryterne — Sky Masking, Mid-Training Floater Cleanup, Reconstruct Sky Dome — som tidligere lå her, ble fra og med v1.6 flyttet inn i Enhancements-seksjonen i Expert-Inspector, der de nå lagres per prosjekt; se kapittel 2, 142–144.) AI-Helpers-fanen slår på on-device-machine-learning-hjelperne (Vision, CoreML) for SfM- og treningsforberedelse.

Tidligere betjeningselementer for samlet aktivering eller deaktivering av alle AI-helpers finnes ikke lenger i den aktuelle versjonen — de er derfor ikke dokumentert her. Også det tidligere „Coming Soon«-området for ennå ikke leverte hjelpere er fjernet og refereres ikke her.

General-fanen



Figur 15: Innstillinger → General-fanen med brukergrensesnitt, viewport, training og experimental-seksjon

S1 Default Mode



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



Styrer hvilken av to UI-modus appen åpner i ved neste oppstart. „Simple Mode« er den veiledede wizard-arbeidsflyten i 4 trinn (Import → Processing → Preview → Export, dokumentert i kapittel 10 under Z1–Z4), „Expert Mode« det klassiske trepanellayoutet med Navigator, 3D-viewport og Expert-Inspector fra kapittel 2. Verdien huskes på tvers av omstart. Identisk virkning som menyen Mode → Simple Mode (⌘1) / Mode → Expert Mode (⌘2), bortsett fra at menyen skifter den løpende sesjonen, mens denne velgeren setter standarden for fremtidige sesjoner. Begge modus aksesserer samme prosjekt-state — prosjekter, kameraer og treningskonfigurasjon bevares ved modusskifte. Modusspesifikke verktøylinjeknapper gjenrendres umiddelbart.

KORT FORKLART

Her velger du hvilket brukergrensesnitt RadianceKit starter med neste gang. „Simple Mode« er begynnermodusen: fire klare trinn, forhåndsvalgte innstillinger, nesten ingen valg. „Expert Mode« er det fulle verktøykasselayoutet med alle håndtak, du ser i kapittel 2. Du kan når som helst veksle frem og tilbake via menyen „Mode«, uten at bilder eller treningsfremgang går tapt.

S2 Language



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



Velger visningsspråk for hele app-UI-en, uavhengig av macOS-systemspråket. RadianceKit er lokalisert til 17 språk (`de`, `en`, `pl`, `en-AU`, `ar-SA` pluss 12 ytterligere). Ved „System« følger appen macOS-språket. Ved et eksplisitt valg huskes språkinnstillingen på tvers av omstart; full virkning krever som regel en app-omstart, fordi lokaliseringsbundles bare lastes ved oppstart. De 298 dokumenterte lokaliseringsnøkene i prosjektet tas alle med, inkludert alle tekster i sub-views og hjelpetooltips.

KORT FORKLART

Hvis Mac-en din kjører på engelsk, men du heller vil ha det norske RadianceKit-brukergrensesnittet (eller omvendt), stiller du det inn her. De fleste tekstene byttes umiddelbart. Noen dialoger dukker først opp på det nye språket etter en app-omstart.

S3 Viewport Background

HVOR

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

TEKNISK

Setter standard bakgrunnsfarge for 3D-viewporten. Tre alternativer: „Dark Gray« (RGB 0.1, 0.1, 0.1 — default), „Black« (0, 0, 0) og „White« (1, 1, 1). Innstillingen persisterer standarden for nye prosjekter og sesjoner på tvers av omstart og oppdaterer samtidig den kjørende Metal-rendereren umiddelbart. Identiske alternativer finnes i menyen Viewport → Background (M21, M22, M23), men Settings-velgeren setter standarden, mens menyen skifter den løpende visningen. Viktig for skjermbilder og demovideoer: hvite bakgrunner fremhever grønne/blå floaters tydeligere, mørke bakgrunner er bedre for rene render-opptak.

KORT FORKLART

Fargen bak 3D-modellene dine i forhåndsvisningsvinduet. Mørkegrå er standard og passer til de fleste scener. Hvit er bra for skjermbilder, sort virker mer elegant for render-opptak. Du kan når som helst skifte fargen via menyen „Viewport → Background« for den løpende scenen — denne innstillingen fastsetter bare hvilken farge som skal være aktiv ved neste åpning.

S4 Auto-Rotate After Training

HVOR

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

TEKNISK

Starter umiddelbart etter treningslutt en kontinuerlig turntable-rotasjon av viewport-kameraet om scenens tyngdepunkt (standard rotasjonshastighet ~0.3 rad/s). Praktisk nyttig for demo-sesjoner, A/B-sammenligninger og for direkte å vurdere fra 360°-perspektiv om det har oppstått „floaters« ved scenekanten. Effekten er visuelt identisk med menyen Viewport → Toggle Auto-Rotation (M16, $\text{Æ}\text{⌘}T$), bortsett fra at toggle-en her utløser atferden automatisk etter treningslutt i stedet for manuelt. Lar seg senere avbryte når som helst via menyen eller ved klikk i viewporten (som pauser rotasjonen). Har ingen innflytelse på treningsytelsen — rotasjonen kjører først når treningen er ferdig.

KORT FORKLART

Når aktivert roterer 3D-scenen automatisk så snart treningen er ferdig — som en karusell. Hyggelig hvis du om morgenen etter en natttrening straks ser resultatet i bevegelse uten selv å måtte klikke. Ved lange sesjoner der du bare overvåker treningen, så la heller være.

S5 Live Preview Interval

HVOR

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

TEKNISK

Bestemmer med hvilket iterasjonsintervall det løpende treningsøyeblikksbildet rendres inn i 3D-viewporten. Fire diskrete verdier: 0 („Off«), 50, 250, 1000 iterasjoner. Ved aktiv Live Preview kopierer treneren Gaussian-bufferen fra GPU-en til en separat render-buffer og utløser en viewport-redraw. Ved „Off» oppdateres viewporten først etter treningsavslutning. Ytelseskostnad: hver 50. iterasjon ~5–10% langsommere på M3 Ultra, hver 250. iterasjon ~1–2% langsommere, hver 1000. iterasjon ikke målbar. Memory-overhead konstant ~2 GB for snapshot-bufferen, uavhengig av intervallet. Verdien fungerer som standard for nye treninger; etter treningsstart viser trenings-Inspector den reelle live-verdien for den aktuelle treningen. Ved intervall 50 er det visuelle inntrykket en flytende „oppvekst« av punktskyen, ved 1000 virker det hakkete.

KORT FORKLART

Mens treningen kjører, kan du velge hvor ofte 3D-visningen oppdateres. „Off« betyr: ingen oppdatering under treningen (raskest). „Every 50 Iterations» viser nesten i sanntid hvordan scenen din oppstår (litt langsommere). For avslappet zen-kikking ved små treninger er „Every 250” et godt kompromiss.

S6 Throttle Delay

HVOR

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

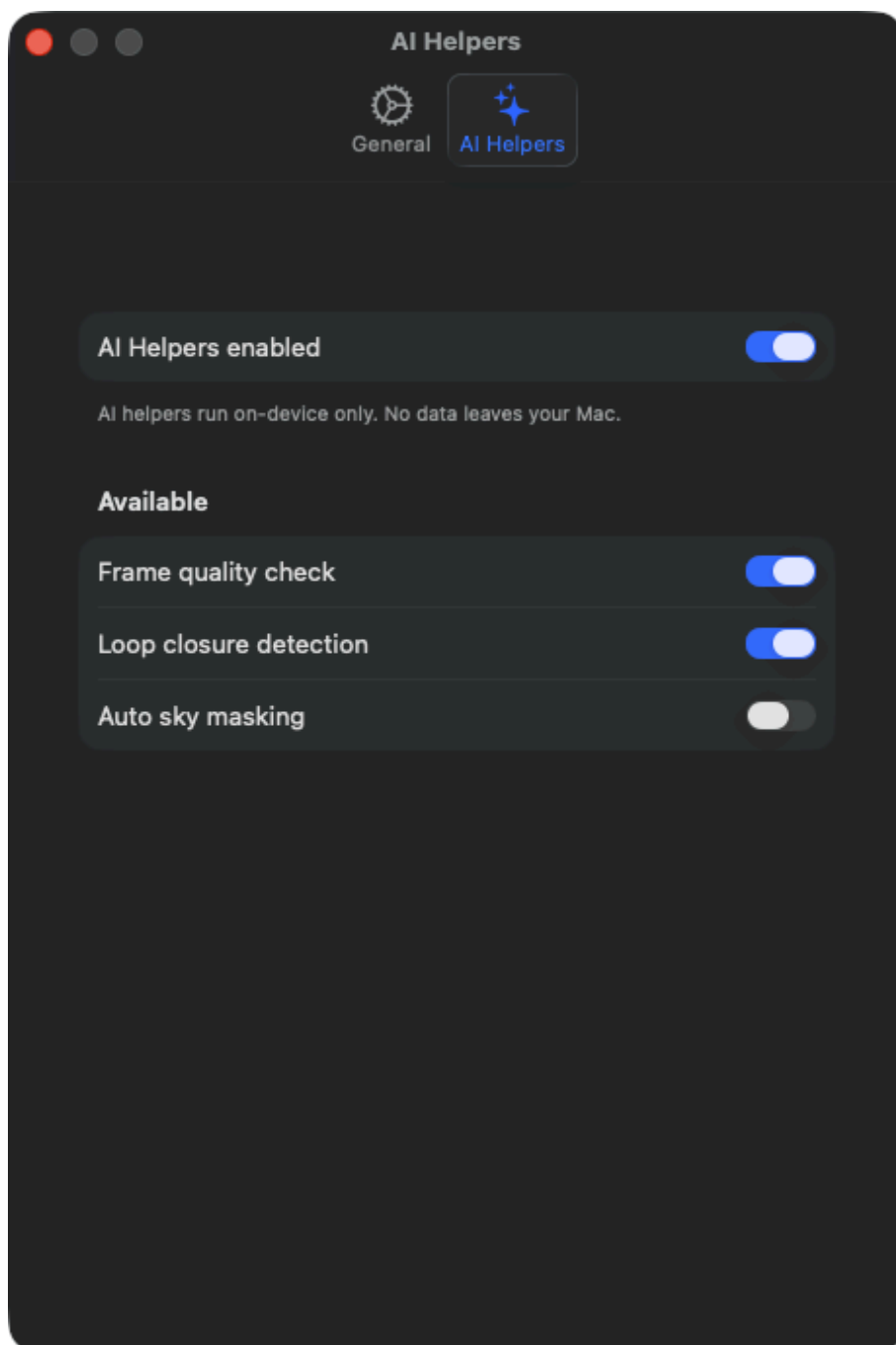
TEKNISK

Setter inn en kunstig forsinkelse i millisekunder mellom treningsiterasjoner. Fire diskrete verdier: 0 („Off«), 2 („Light«), 5 („Moderate«), 10 („Eco«). Mening: ved lengre treninger (flere timer) blir GPU-en ellers belastet 100 %, noe som fører til merkbart langsommere system-UI (musepekeren hakker, andre apper blir trege). Throttle-forsinkelsen gir GPU-en pauser der andre oppgaver kan utføres. Ytelses-kostnaden er betydelig: ved 5 ms throttle varer en typisk 40K-trening rundt 50–80% lenger enn uten throttle. I ytelsesmodus „Eco« (10 ms) er forsinkelsen per iterasjon lengre enn selve iterasjonen — faktor 2–3× langsommere. Ved aktiv throttle vises det under velgeren en merknad: „Throttle is on. Training will be slower than usual.» Selve appen reagerer ikke merkbart bedre — bare andre apper har nytte av det.

KORT FORKLART

Hvis Mac-en din blir for varm under en lang trening, eller andre programmer blir for trege, så slå på en brems her. „Off« gir GPU-en full gass (raskest). „Light« legger inn en liten pause mellom hvert skritt (litt langsommere, men systemet reagerer bedre). „Eco« er den sterkeste bremsen — bra for nattreninger på en MacBook som ikke skal bli for varm.

AI-Helpers-fanen



Figur 16: Innstillinger → AI-helpers-fanen med masterbryter og sub-toggles

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

Settings → AI Helpers → første seksjon → Toggle „AI Helpers enabled«. Bound: Default: `true`.

 **TEKNISK**

Masterbryter over alle AI-helpers-funksjoner i pipeline. Når avslått, hopper import- og SfM-pipeline over alle ML-baserte forbehandlingsstadier helt — ingen Apple-Vision- kall, ingen CoreML-modellinnlasting, ingen NPU-oppvåkning. Når slått på konsulteres de individuelle sub-toggles (S12–S13). Verdien huskes på tvers av omstart. Påvirker følgende stadier: (a) frame-quality- pre-check før SfM (S12), (b) loop-closure-deteksjon (S13). Viktig: ved avslått er de to sub-toggles deaktivert og visuelt grånet ut. Footer-merknad understreker at alle AI-helpers strengt kjører on-device — ingen bildeopplasting, ingen skybehandling. Personverngarantien kommer ved utelukkende bruk av Apple-Vision-framework (lokalt på Neural Engine) og CoreML-modeller som ligger direkte i app-bundlen.

 **KORT FORKLART**

Hovedbryteren for alle funksjoner som internt bruker AI/machine-learning. Standard er „på«, fordi hjelperne sparer mye tid uten at bildene dine forlater Macen. Hvis du vil ha dem helt avslått (f.eks. for å spare strøm, eller fordi Macen din ikke har NPU), så slå dem av her — da blir de to underalternativene nedenfor automatisk grå og gjør ikke noe lenger.

S12 Frame quality check **HVOR**

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

 **TEKNISK**

Aktiverer frame-quality-screeneren (fase 3.11), som før SfM-kallet analyserer hver importerte frame. Pipeline-trinn per frame: (a) Laplacian-variance-fil-ter fra Apple Vision (uskarphetsdeteksjon — terskel ~150), (b) histogram-basert over/under-eksponerings-sjekk (terskel: >5% piksel ved 0 eller 255), (c) blank-frame-detect (standardavvik < 5 over alle piksler). Frames som består alle tre tester går direkte igjennom. Frames som feiler minst én test, utløser en modal bekreftelsesdialog som lister hver problematiske frame med thumbnail og begrunnelse og spør om den skal fjernes. Viktig: ingen automatisk sletting — dialogen er alltid påkrevd, brukeren beholder den siste avgjørelsen. Ytelse: ~50 ms per frame på M3 Ultra, kjører parallelt. Ved avslått sendes alle frames uprøvd videre til SfM. Ved deaktivert master (S11) er denne toggle visuelt grånet ut og uten virkning. Shipped status ifølge memory: SHIPPED 2026-05-23.

 **KORT FORKLART**

Før selve treningen ser appen på hvert foto: er det rystet? helt mørkt eller hvitt? tomt? Hvis ja, spør den deg om du vil kaste bildet — den fjerner aldri noe automatisk. Det sparer mange timer senere, fordi et eneste totalt rystet bilde noen ganger kan ødelegge hele treningen. Standard er „på«, fordi innsatsen er nesten null og nytten stor.

S13 Loop closure detection

HVOR

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

TEKNISK

Aktiverer Apple-Vision-Feature-Print-basert loop-closure-deteksjon. For hver importerte frame beregnes en ~768-dimensjonal feature-vektor som representerer en nevralt innleiring av bildeinnholdet. Deretter sammenlignes alle feature-prints parvis via cosine-similarity. Par med similarity > 0.85 og avstand i frame-index > 50 (altså ikke-nabostilte frames) identifiseres som „loop-closure-kandidater« og skrives i en sidecar-JSONL-fil i prosjektmappen. Bare informativ — den importerte bildesekvensen modifiseres ikke. Mening: gir SfM-solveren (spesielt COLMAP) et hint om at disse frames hører sammen i klyngen i 3D-rommet. For native SfM er sidecar-informasjonen p.t. bare dokumenterende; COLMAP bruker hintene internt via custom matches-file (manuell integrasjon mulig, ikke automatisk koblet). Ytelse: ~200 ms per frame på M3 Ultra, kjører parallelt. Ved avslått genereres ingen feature-prints. Ved deaktivert master (S11) visuelt grånet ut.

KORT FORKLART

Når du går rundt et objekt og til slutt lander ved startpunktet igjen, hjelper det datamaskinen enormt å vite det. Dette alternativet registrerer automatisk hvilke av fotoene dine som er tatt „nesten fra samme ståsted«, og skriver det i en liten hjelpefil. SfM-verktøy (spesielt COLMAP) kan bruke denne informasjonen til å levere en renere 3D-rekonstruksjon. Standard er „på«, fordi det kjører uten din innblanding og ikke endrer noe ved bildene dine.

Inspector-speilet settings

De øvrige settings-oppføringene (S17–S33) fra inventartabellen er speilinger fra Expert-Inspector og dokumentert i kapittel 2 (Inspector-Controls I12–I29). De dukker ikke fysisk opp i innstillingsvinduet, men er bare listet i inventaret fordi de kjører via `TrainingConfig` -properties som persisteres via, og dermed formelt har settings-karakter. For innholdsmessige forklaringer henvises det dit.

Når hva?

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

App-globalt = virker på alle prosjekter. Default for nye treninger = virker bare på den neste opprettede treningen, løpende sesjoner forblir uendret. Aktuell trening = virker umiddelbart på den løpende treningskonfigurasjonen, men persisterer ikke uten eksplisitt reimport.

KAPITTEL

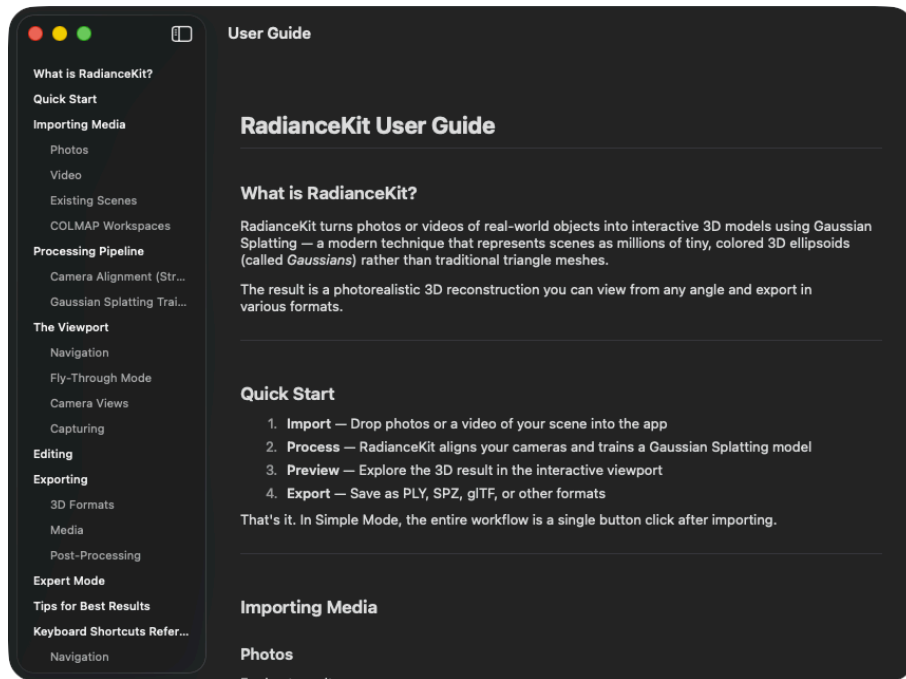
Kapittel 4 — Hjelpvinduer

Foruten hovedvinduet (3D-viewport pluss Inspector) forvalter RadianceKit syv ytterligere vinduer, som alle åpnes via Help-menyen. Listen er fra topp til bunn: User Guide (⌘?), Keyboard Shortcuts (⌘/), Open Training Logs... (åpner ikke et app-vindu, men Finder; derfor ikke ytterligere behandlet her), Manage Storage..., Pareto Dashboard... (⇧⌘D), Holdout Analysis... (⇧⌘H), BayesOpt Console... (⇧⌘B). Tre av dem — Dashboard, Holdout, BayesOpt — er selvstendige analyseverktøy. De har hver sin egen view-model-stack, leser eller skriver JSON-filer på disk, og for hver finnes det et CLI-argument hvor du kan få vinduet til å peke direkte på en bestemt fil ved app-start (`--dashboard-dir` , `--holdout-file` , `--bayesopt-autorun`).

De fire enkle vinduene (User Guide, Keyboard Shortcuts, Manage Storage, pluss undermeny-punktene Open Training Logs / Open Exports Folder) får per styreelement en kort oppføring. De tre analysevinduerne er mer utførlig dokumentert — hver med en innledning som forklarer hva du ser i vinduet, når du bør åpne det, og hvordan du tolker det viste bildet.

Sist i kapittelet er det et krysshenvisningsavsnitt til hovedvinduets Inspector: hva du fornuftig kan lese av i live- loss-charten og gaussian-count-visningen under en kjørende trening.

User Guide (W1–W4)



Figur 17: User Guide-vindu med sidefelt til venstre og rendret markdown- innhold til høyre

Hva det er: Et innebygd hjelpevindu som renderer den medfølgende `guide_<sprog>.md`. Språket avledes av innstillingene (fanen General → Language) eller, hvis „System« står der, av macOS-språkpreferansene. Layout er klassisk: til venstre sidefelt med alle overskrifter, til høyre brødteksten.

NÅR ÅPNE Når du trenger en rask påminnelse om et enkelt punkt — altså som stikkordserstatning. Den utførlige referansen er denne manualen; det innebygde hjelpevinduet er heller det en `--help` på kommandolinjen ville være. Det oppdateres ved hver app-utgivelse, men holdes innholdsmessig overflatisk.

W1 NavigationSplitView (sidefelt + detalj)



Help → User Guide (⌘?).



Tospalters layout med smalt sidefelt (minst 180 pt bredt) for innholdstreet og et scrollbart detaljområde for selve markdown-innholdet. Vinduet har en minimumsstørrelse på 700 × 500 pt. Ved første åpning laster vinduet den passende `guide_<lang>.md` fra app-bundlen (fallback `guide_en.md`), parser den til blokk-records (overskrifter H1–H4, avsnitt, lister, tabeller, separator-linjer) og trekker ut separat overskriftsstrukturen for sidefeltet. Inline-formatting (fet, kursiv, code-span) rendres via den innebygde markdown-engine. Språket leses fra app-innstillingene, med spesialtilfellet kinesisk (`zh-Hans`) og brasiliansk portugisisk (`pt-BR`), som bevares som fulle locale-tags fordi disse variantene skiller seg fra zh og pt.

KORT FORKLART

Den innebygde hjelpeteksten, til venstre emnelisten, til høyre innholdet. Språket settes automatisk etter system-settings dine. Fungerer offline, men er bevisst kun en kortversjon — den fulle referansen er denne manualen.

W2 List (heading-sidefelt)



Venstre kolonne i User Guide-vinduet.



Liste over alle H2- og H3-overskrifter i det aktuelle markdown-dokumentet. H2-oppføringer dukker opp uten innrykk med medium skriftsnitt, H3-oppføringer med 16 pt innrykk til venstre og redusert foreground-stil. H4 og dypere ignoreres fordi dybden ellers gjør sidefeltet uoversiktlig. Anker-ID-er genereres fra heading-teksten via slugifisering (lowercase + mellomrom til streker + filtrering på bokstaver/tall/streker — samme algoritme som GitHub bruker til markdown-ankrene sine, slik at også eksterne URL-er til doc potensielt ville lande på samme anker). Listen bruker native macOS-stil.

KORT FORKLART

Navigasjonsbjelken på venstre side. Trykk på en oppføring, og du hopper til avsnittet.

W3 Button (heading → anker-sprang)



Per sidefelt-linje en knapp.



Hver sidefelt-oppføring er en knapp som setter det aktuelle ankeret, men optisk ser ut som en liste-oppføring. En observatør-variabel utløser så scroll-spranget til det tilsvarende ankeret med en myk animasjon over 0,3 s. Etter spranget nullstilles anker-verdien slik at neste klikk på samme anker utløser igjen (ellers ville observatøren ikke utløse på nytt fordi verdien ikke har endret seg).

KORT FORKLART

Klikk fører deg til det rette stedet i teksten til høyre.

W4 ScrollView (detalj-innhold)



Høyre kolonne.

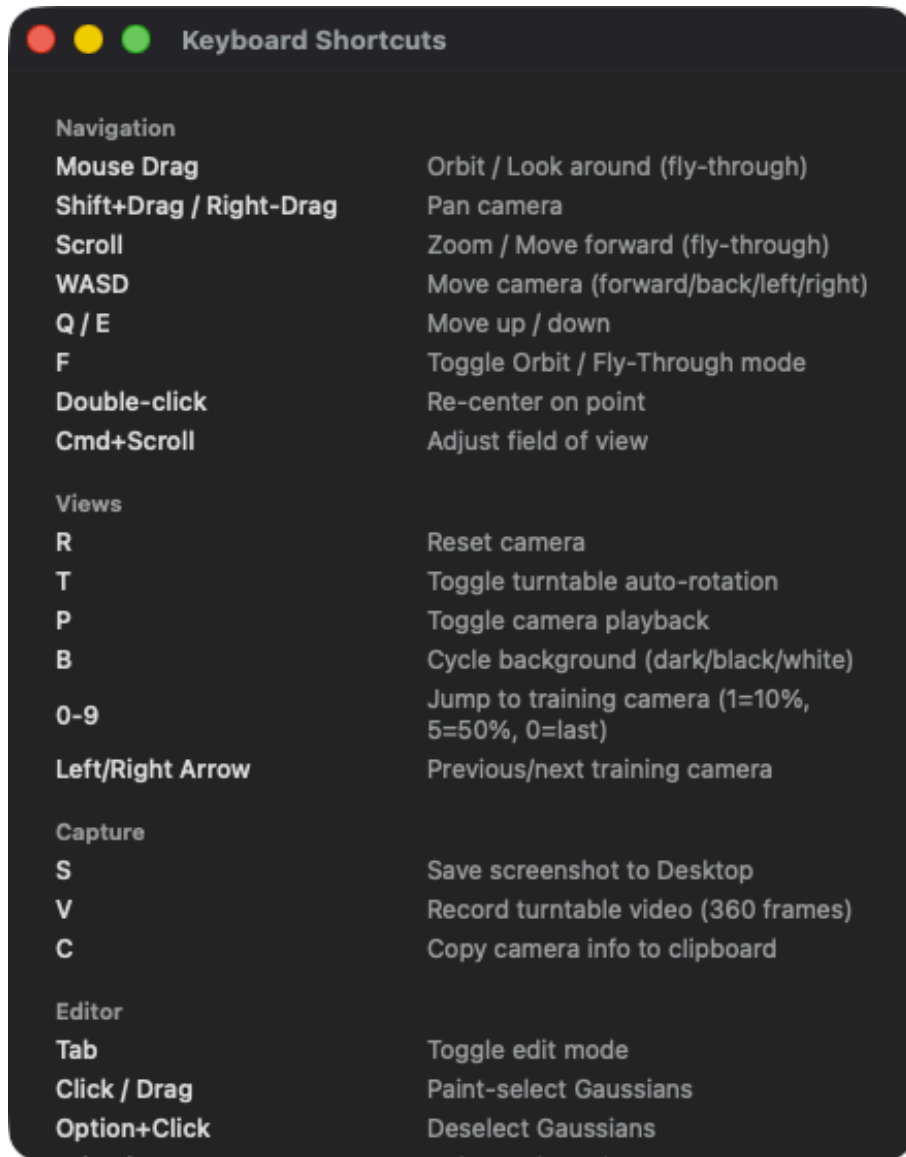


Scrollbar, vertikalt stablende innholdsområde med lazy rendering fordi lengre guides lett kan ha over 200 markdown-blokker — en ikke-lazy variant ville instansiere dem alle samtidig. Hver blokk får en egen ID, enten heading-ankeret (for hoppbare H1–H3) eller en indeks-plassholder. Maksimumsbredde er 720 pt, padding 32 horisontalt / 24 vertikalt, slik at lange linjer beholder et godt lesbart layout. Tabeller rendres celle-vis med horisontale stacks og separator-linjer; inline-kode av den innebygde markdown-engine. Egentlige kodeblokker behandles aktuelt som paragraph — en kjent begrensning ved hjelpvinduet.

KORT FORKLART

Selve hjelpeteksten. Scrollbar, godt lesbar bredde, klar typografi.

Keyboard Shortcuts (W5–W6)



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

HVA BILDET VISER Statisk referanseliste i fem seksjoner. **Navigation:** Mouse Drag (orbit/fly), Shift+Drag/Right-Drag (pan), scroll (zoom), WASD (fly-through- bevegelse), Q/ E (opp/ned), F (toggle orbit/fly), dobbeltklikk (re-center), Cmd+scroll (FoV-justering). **Views:** R (reset camera), T (auto-rotation), P (camera playback), B (background-cycle), 0–9 (hopp til training-cam 1=10 %/5=50 %/0=last), venstre/høyre pil (prev/next cam). **Capture:** S (skjerm bilde til desktop), V (turntable-video), C (copy camera info). **Editor:** Tab (edit-mode), klikk/dra (paint-select), Option+klikk (deselect), X / Delete (slett seleksjon), Cmd-Z (angre siste sletting), [/] (pensel mindre/større), Esc (opphev seleksjon). **Training:** Start, Pause/Resume, Cancel, Continue +5K/+10K/+20K via meny-hurtigtastene i M9–M14.

Hva det er: En enkel statisk oversikt over alle hurtigtaster — Navigation, Views, Capture, Editor, Training. Innhold er hardkodet i, ingen markdown-loading.

NÅR ÅPNE Når du leter etter den raskeste veien til å gjøre noe i viewporten. WASD-fly-through, R til camera-reset, B til background-cycling — alle står her.

W5 ScrollView (innholdsområde)

 HVOR

Help → Keyboard Shortcuts (⌘/).

 TEKNISK

Et enkelt scroll-område med en vertikal liste i. Padding 20 hele veien rundt, ingen sidefelt-navigasjonstre (listen er kort nok). Innhold er gruppert i fem seksjoner (Navigation, Views, Capture, Editor, Training). Per tastekombinasjon en linje med oversettbar tekst i begge kolonner. Venstre kolonne (tastekode) fastgjort til 180 pt bredde, slik at beskrivelsene til høyre forblir vertikalt aligned. Ingen interaksjon ut over scrolling — klikk på en linje utløser ingenting, hurtigtastene er ekte tastatur-modifiers i menyen og på viewporten.

KORT FORKLART

Tabell over alle shortcut-taster. Statisk cheat sheet for rask oppslag.

W6 VStack (shortcut-seksjoner)

 HVOR

Innenfor ScrollView.

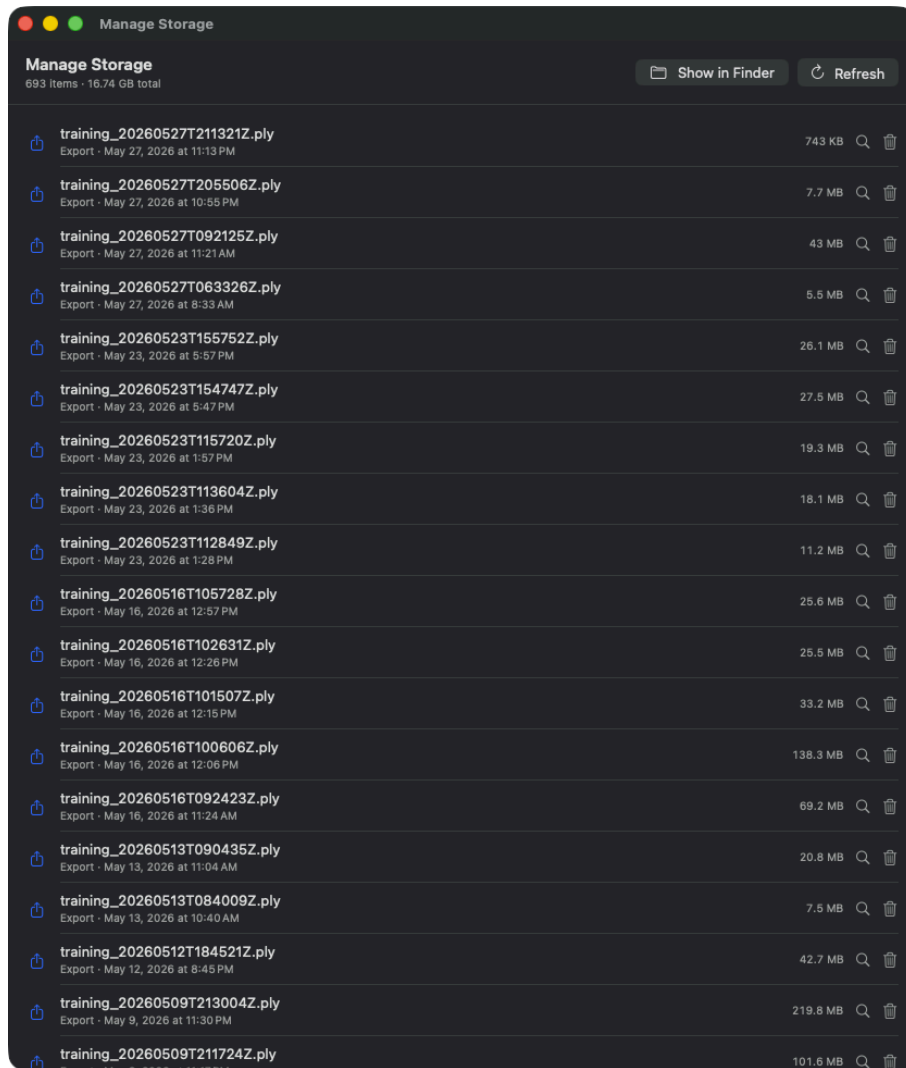
 TEKNISK

Venstrejustert stablede seksjoner med 16 pt avstand. Innenfor de fem seksjonene hver heading + linjefølge. Headings bruker en sekundær subheadline-stil — bevisst ikke title-format fordi seksjonene ikke trenger å være navigerbare. Innhold er bevisst flatt (ingen disclosure, ingen search, ingen filter), slik at komponenten kjører uendret på enhver macOS-versjon, og filen forblir lesbar.

KORT FORKLART

Grupperingen av taster etter funksjon (Navigation, Views, Editor og så videre).

Manage Storage (W7–W12)



Figur 19: Manage Storage-vindu — header viser „693 items · 16.74 GB total«, tabell med export-PLY-filer sortert etter dato, hver med format-pille + filnavn + størrelse + dato

HVA BILDET VISER Tabellvisning av alle filer RadianceKit forvalter. Header teller 693 items, 16.74 GB samlet størrelse. Toolbar øverst: „Show in Finder« + „Refresh«. Hver linje: PLY-ikon, filnavn (f.eks. training_20260527T211321Z.ply), eksportdato, størrelse (varierer 7 KB til 218 MB), forstørrelses-ikon (reveal) og papirkurv-ikon (move to trash). Filer er sortert etter dato, nyeste øverst. I dette demo-opptaket dominerer PLY-eksporter fordi det er arbeidet mye med `--benchmark`.

Hva det er: En disk-usage-oversikt for alt RadianceKit legger under `~/Documents/RadianceKit/` — logs, exports, scenes, capture-bundles (fra iOS-companion), imports (staging-kopier av input-bildene). Per oppføring en størrelse i bytes og to knapper: „vis i Finder« og „flytt til papirkurv«. Det er INGEN automatisk opprydding — appen sletter ingenting selv; du bestemmer per oppføring.

NÅR ÅPNE Når disken blir full. Spesielt logs samles opp (en JSONL per treningsforsøk, pluss `_qualityMetrics.json`); eksportene også (PLY 100 % rådata, en per eksport).

Også nyttig etter en krasj der imports-staging-mappen har gamle kopier av input-bilderne liggende (se „disk-pressure incident« i `dev_v549f-needle-reduction.md`).

W7 Knappen „Show in Finder«

 HVOR

Header øverst til høyre i storage-browser-vinduet.

 TEKNISK

Åpner hele RadianceKit-mappen (`~/Documents/RadianceKit/`) i Finder, slik at du kan se mappestrukturen direkte og også manipulere den med Finder selv. Handlingen åpner et nytt Finder-vindu og bytter ikke til app-sandbox-containeren — `~/Documents/RadianceKit/` er det regulært app-tilgjengelige Documents-domenet, ingen sandboxed-container-sti.

 KORT FORKLART

Åpner mappen i Finder slik at du selv kan håndtere filene.

W8 Knappen „Refresh«

 HVOR

Header, ved siden av Finder-knappen.

 TEKNISK

Utløser en bakgrunnsscanning som kjører på en brukerinitiert asynkron task, slik at scanningen av store mappetrær ikke blokkerer UI-en. Selve gjennomgangen går hver kjent undermappe (Logs, Exports, Scenes, Captures, Imports) gjennom og genererer en storage-oppføring per direkte barn. Per oppføring bestemmes den rekursive størrelsen — helst det faktiske diskforbruket (inkludert APFS-hardlinks-sharing) med fallback på den logiske filstørrelsen.

 KORT FORKLART

Leser listen på nytt hvis du i mellomtiden har slettet eller lagt til noe i Finder.

W9 List (storage-oppføringer)

Hovedinnhold under headeren.



Liste med dette layoutet per linje: kategori- spesifikt SF-symbol-ikon (dokument for logs, upload-pil for exports, terning for scenes, tray for imports), navn + undertittel (kind-label + formatert modifikasjonsdato), bytes-teller til høyre (høyrejustert, monospaced), reveal-knapp (forstørrelses-symbol), trash-knapp (papirkurv). Sortering: primært etter kind (scenes først, så exports, logs, captures, imports, other), sekundært etter modifikasjonsdato fallende (nyeste øverst). Hvis scanningen fortsatt kjører, viser stedet i stedet en „Scanning...«- fremgang. Hvis ingenting ble funnet, en empty-state-visning med tray-ikon.

 KORT FORKLART

Liste over alle RadianceKit-dataene dine, sortert etter type og aktualitet. Per oppføring ser du størrelsen og kan slette direkte.

W10 Row-knapp „Reveal in Finder«

Per linje, forstørrelses-symbol til høyre.



Åpner Finder og velger den spesifikke oppføringen (fil eller mappe). Forskjell fra W7: W7 åpner rotmappen; W10 markerer nøyaktig denne ene oppføringen. Praktisk workflow: identifiser en stor oppføring, klikk på forstørrelsen, kopier den så f.eks. til et eksternt volume.

 KORT FORKLART

Hopper i Finder direkte til denne oppføringen, slik at du raskt finner den.

W11 Row-knapp „Move to Trash«

Per linje, papirkurv-symbol til høyre for forstørrelsen.



Utløser bekreftelsesdialogboksen (W12). Først etter bekreftelse kjører macOS-standardoperasjonen „flytt til papirkurv« (altså reversibel, ingen direkte sletting). Etter vellykket trash fjernes oppføringen fra listen, og total-byte-telleren oppdateres. Ved feil slås det inn en modal feildialog.

 KORT FORKLART

Flytter oppføringen til papirkurven. Dialog spør først.

W12 ConfirmationDialog (slettebekreftelse)

Utløses av W11, vises som macOS-sheet.

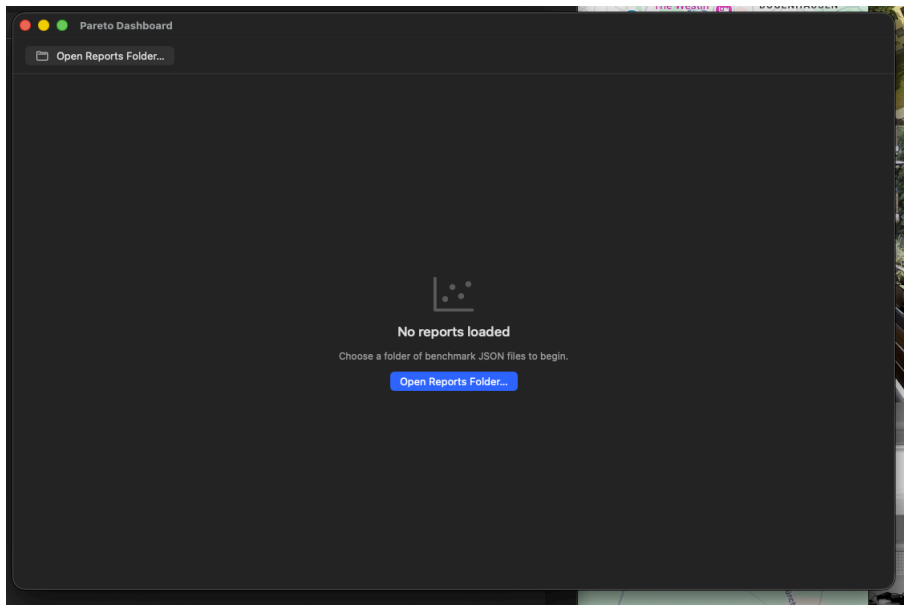


Standard bekreftelsesdialog med dynamisk tittel „Delete <navn>?« og en meldingslinje som eksplisitt påminner om at oppføringen lander i papirkurven og kan gjenopprettes derfra (inntil papirkurven tømmes). To knapper: „Move to Trash« som destruktiv handling (vist rødt) og „Cancel« med automatisk Esc-binding. Dialogen er non-modal i den forstand at den bare blokkerer dette vinduet, ikke hele appen — det er macOS-standard for reversible slettinger.

 KORT FORKLART

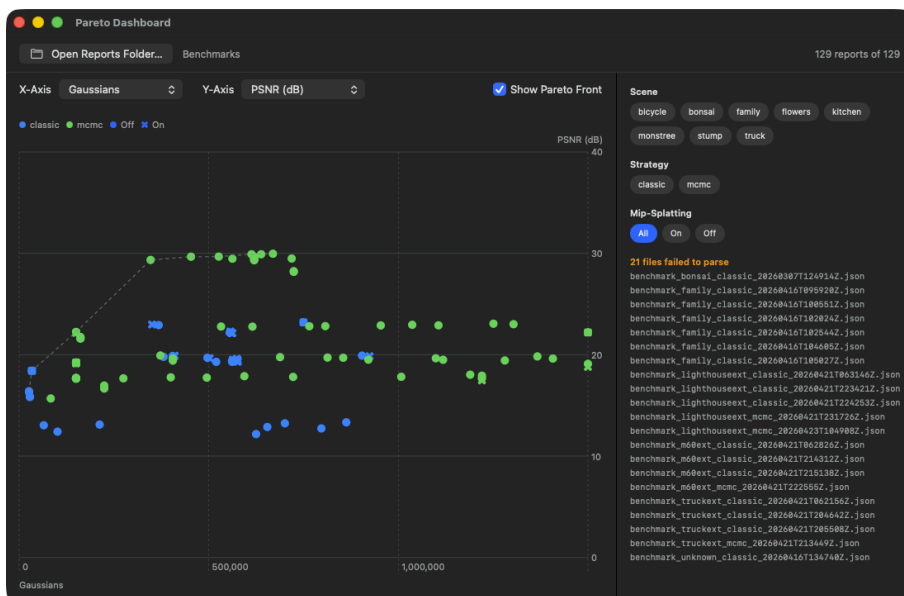
Sikkerhetsspørsmål før sletting. „Move to Trash« er reversibelt — så lenge papirkurven ikke er tømt.

Pareto Dashboard (W13–W22)



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

Tom tilstand (etter første åpning) — empty-state med call-to-action „Open Reports Folder...«. Datapunktene dukker opp så snart treningsreports er lastet inn, se neste shot.



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

HVA BILDET VISER Header-toolbar viser „129 reports of 129“ (alle reports i den valgte mappen ble parsed med suksess — 21 ytterligere filer kunne ikke parses pga. eldre format, se hint-listen til høyre). Akser: X-Axis-velger på Gaussians , Y-Axis-velger på PSNR (dB) . Scatter-plot: grønne punkter = Classic-strategi, blå punkter = MCMC. Den stiplede Pareto-front- linjen løper langs de best oppnådde PSNR-verdiene og platåer omkring PSNR≈30 dB fra ca. 500K gaussians. Filter-chips til høyre: 7 scener (bicycle,


bonsai, family, flowers, kitchen, stump, truck), 2 strategier (classic, mcmc), 3 Mip-splattung-alternativer (All, On, Off). For øyeblikket er alle filtre åpne, derav den tette punktklyngen.

Hva det er: Et multi-run-sammenligningsverktøy. Du har i fortiden trent flere scener eller samme scene med forskjellige presets — hver av disse treningskjøringene produserer (hvis du har gitt med `--benchmark` eller kalt via benchmark-funksjonen) en JSON-report-fil som bl.a. inneholder final-PSNR, SSIM, LPIPS, gaussian-count og wall-clock-tid. Dashbordet laster inn en hel mappe med slike reports samtidig og plotter dem som 2D-scatter med valgbare akser. I tillegg tegnes Pareto-fronten (mengden av ikke-dominerte punkter) som stiplet linje.

NÅR ÅPNE Etter at du har opprettet minst tre eller fire treningsreports. Med færre punkter er frontier-linjen ikke sigende. Typisk use-case: du har prøvd å rekonstruere en outdoor-scene og har spilt P3 Balanced (Classic), P4 Quality (Classic), P7 MCMC Quality og P9 Outdoor (tuned) gjennom etter hverandre — nå vil du vite hvilken konfigurasjon som leverer beste PSNR per sekund treningstid, eller hvilken som krever færrest gaussians for gitt PSNR.

SLIK TOLKES DET Begge akser kan fritt velges (X-akse:,, psnr , ssim , lpips , ...; Y-akse likeså). Pareto-front-logikken i ParetoFront2D.indices vet for hver metrikk om „mindre = bedre« (f.eks. LPIPS, Loss, Time) eller „større = bedre« (PSNR, SSIM) — linjen løper altså avhengig av akse-valget fra nederst til venstre til øverst til høyre eller fra øverst til venstre til nederst til høyre, alltid langs den beste oppnådde kombinasjonen. Et punkt er Pareto-optimalt hvis INGEN andre punkter i BEGGE dimensjoner er minst like gode (altså ingenting annet dominerer det). Pareto-optimale punkter ligger på linjen, andre punkter til høyre/over (avhengig av akseorientering). Punkter PÅ linjen er de ekte kandidatene for „beste preset«; punkter LANGT fra linjen er bortkastet treningstid.

FILTER-CHIPS Du kan begrense valget til en bestemt scene (hvis du f.eks. bare vil sammenligne outdoor-runs), til en bestemt strategi (Classic eller MCMC), eller Mip-splattung på/av (relevant etter fase Q1.5 hvor Mip forblir 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). Sett X-aksen til trainingTime, Y-aksen til PSNR. Run B ligger øverst til høyre, Run C enda lengre mot øverst til høyre, Run A nederst til venstre. Pareto-fronten forbinder A og C — begge ikke-dominerte. Run B er „lost« (C er bedre i både time OG PSNR). Innsikt: for „truck« er MCMC-defaulten ikke verdt det; enten rask+ok (A) eller lang+veldig god (C). Lagre konfigurasjonen fra C som egen preset (Inspector → I1 Save Preset).

Neste handling: Lagre beste konfigurasjon som preset. Konkret: kikk på Pareto-punktene (hover viser PSNR/SSIM/LPIPS/Gs/ Time i toltippet), bestem hvilken time-vs-quality-tradeoff som passer best, åpne den tilhørende reporten (filnavnet inneholder run-timestampet), kopier dens treningskonfigurasjon i en ny run eller lagre den etter neste treningssesjon som preset via Inspector.

W13 Knappen „Open Reports Folder...«

Toolbar øverst til venstre.



Åpner en mappevelger med oppfordringen „Select a folder containing benchmark .json reports«. Etter bekreftelse kjører en bakgrunnstask som parser alle .json -filer i mappen sekvensielt. Feilaktige reports (defekt JSON, feil skjema) samles og vises nederst i sidefeltet som „N file failed to parse« — ingen krasj. Hvis det kommer et annet klikk mens en første load fortsatt kjører, avbrytes den tidligere tasken slik at det ikke skrives to resultater inn i state samtidig.

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

 KORT FORKLART

Velger mappen der benchmark-reportene dine ligger. Standardstien er `~/Documents/RadianceKit/Reports/`. Laster så alle JSON-er på en gang.

W14 Velger „X-Axis«

Over charten, til venstre.



Menyvelger med alle tilgjengelige metrikkakser i dashboard-modulet (PSNR, SSIM, LPIPS, gaussian-count, treningstid og så videre). Default er gaussian-count. Ved bytte nullstilles det hoverede punktet fordi en hittil highlightet posisjon i det gamle aksekoordinatsystemet ikke lenger gir mening etter aksebytte. Velgeren er begrenset til innholdsbredden slik at den ikke spenner over hele bredden.

 KORT FORKLART

Hvilken metrikk som skal stå på den horisontale aksene. Som regel „treningstid« eller „gaussian-antall« fordi det er „kostnadene« du vil sammenligne.

W15 Velger „Y-Axis«

Over charten, ved siden av X-Axis.



Identisk med W14, bortsett fra at default er PSNR. Aksevalget lagres uavhengig, slik at brukeren også kan velge tøysekombinasjoner (X=PSNR, Y=PSNR — ville kaste alle punkter ned på en diagonal). Slike kombinasjoner fanges imidlertid ikke; bevisst beslutning fordi en sammenligning „SSIM vs PSNR« sikkert kan være interessant for å se hvor konsistent metrikkene oppfører seg.

KORT FORKLART

Hva som står på den vertikale aksene. Vanligvis „PSNR« eller „SSIM« som kvalitetsmål.

W16 Toggle „Show Pareto Front«

Til høyre for akse-velgerne.



Standard macOS-toggle. Når aktiv tegnes utover punktskyen i Pareto-charten en linje med den beregnede 2D-Pareto-fronten. Stil: stiplet (stregmønster 4–4), grå halvtransparent, linjetykkelse 1,5 pt. Pareto-beregningen kjører på hovedtråden — ved det typiske antallet reports ($\leq \sim 50$) er det problemfritt raskt. Når toggle er av, utelates linjen slik at bare de nakne punktene står.

KORT FORKLART

Viser linjen som går gjennom de „hittil beste« punktene. Hvis linjen er i veien (f.eks. fordi du bare vil sammenligne de enkelte trades), så slå den av.

W17 Chips „Scene«-filter

Høyre sidefelt i dashboard-vinduet.



Filter-chips for hver scene som dukker opp i de innlastede reportene. Eget flow-layout som automatisk ompakker chips i flere linjer så snart bredden er utnyttet. Aktive chips får accent-bakgrunn, inaktive en nøytral standard-material-bakgrunn. Multivalg er mulig (set-semantikk); hvis ingen chip er valgt, gjelder alle scener som „sluppet inn« — dvs. set-logikken er „tomt valg = alt«, ikke „tomt valg = ingenting«.

 KORT FORKLART

Klikk på et scenenavn filtrerer punktene til bare denne scenen. Multivalg mulig. Tom = alle scener.

W18 Chips „Strategy«-filter

Under scene-filteret i sidefeltet.



Nøyaktig som W17, men for treningsstrategier — typisk de to verdiene „classic« og „mcmc«, avledet fra strategy- feltet i benchmark-report-JSON-ene. Hjelpsom hvis du har blandet reports fra begge strategier og bare vil se den ene typen (f.eks. „vis bare MCMC-runs fordi jeg har utelukket Classic«).

 KORT FORKLART

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

W19 Chips „Mip-Splatting«-filter

Under strategy-filteret i sidefeltet.



Treverdis-filter (i stedet for set som W17/W18): „All« / „On« / „Off«. Bakgrunn: Mip-splatting ble evaluert i fase Q1.5 som eksperimentell multi-skalering, og det endelige verdikttet var „ingen pen win gjennomgående; beholdt som opt-in flag«. Når du gjør Mip-on/off-sammenligninger, vil du ofte kunne skille veldig skarpt. Derfor det dedikerte ternære filteret med tilstandene „la alt igjennom«, „bare Mip på«, „bare Mip av«. Sidefelt-seksjonen slås på bare hvis det er minst én Mip-report OG minst én ikke-Mip-report i datamengden (ellers gir filtreringen ingen mening).

 KORT FORKLART

Hvis du vil sammenligne Mip-splatting på/av, treverdig filter her. Ellers ignorer.

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

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



Minimalistisk knapp-wrapper. Innhold: label-tekst med caption-skriftgrad og padding 10 horisontalt / 5 vertikalt. Bakgrunn betinget: når aktiv → app-accentfarge med hvit tekst; ellers nøytral standard-material-bakgrunn med sort tekst. Formen er en capsule (pilleformet). Plain-buttonstyle slik at capsule-materialet ikke dekkes over av en system-border.

 KORT FORKLART

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

W21 Chart (Pareto-scatter)

Midten av dashbordet.



Swift-Charts-diagram med to lag: 1. et punkt per report — posisjon fra de valgte X- og Y-metrikker, farge etter strategy, symbol etter Mip-status. Symbolstørrelse normalt 80, highlightet 200 (når ID svarer til den aktuelt hoverede reporten). 2. en linje for Pareto-fronten, bare hvis toggle er på.

Chart-overlay: et transparent rektangel registrerer musebevegelse; per frame bestemmes den euklidisk nærmeste punktposisjonen i plot-frame, og den hoverede reporten oppdateres hvis distansen er under 24 px (ellers nullstilt). Slik får du tooltippet uten klikk — hover holder.

 KORT FORKLART

Selve scatter-diagrammet. Hvert punkt er en treningskjøring. Hover for detalj-tooltip.

W22 Tooltip (hover-detalj)

Under charten, slått inn ved hover.

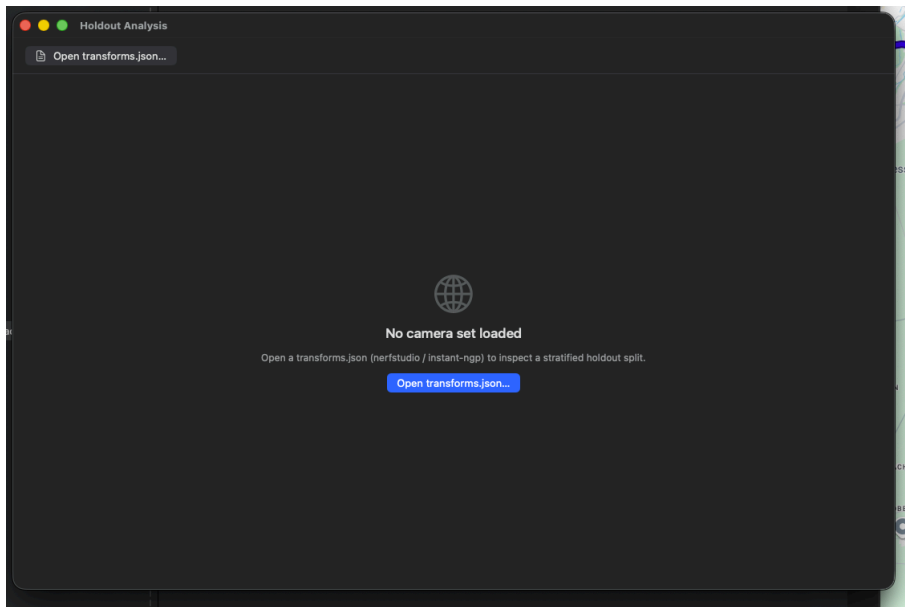


Horisontalt stack: scenenavn (headline), strategy-tag (caption), separatorlinje, så PSNR/SSIM/LPIPS/Gs/Time-metrikker i hver en liten vertikal gruppe (label + monospaced verdi). Hvis Mip var aktivert, dessuten en „Mip«-capsule-tag i accentfarge. Bakgrunn halvtransparent blur, avrundet rektangel med 8 pt radius. Viser bare når musen faktisk er over et punkt. Forsvinner automatisk ved forlating.

 KORT FORKLART

Detalj kortet nederst når du fører musen over et punkt. Viser alle kvalitetsmetrikker og run-konfigurasjonen på en gang.

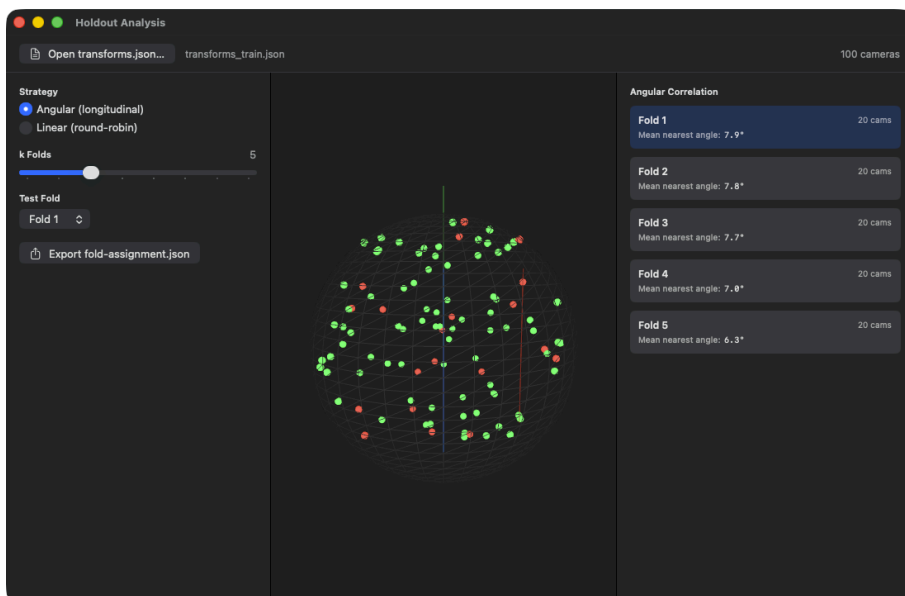
Holdout Analysis (W23–W29)



Figur 22: Holdout Analysis — tom tilstand før innlasting av en transforms.json

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

Tom tilstand (etter første åpning) — kameramarkørene dukker opp så snart en transforms.json er lastet inn, se neste shot.



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

HVA BILDET VISER Header viser innlastet fil (transforms_train.json) og cam-count („100 cameras«). Venstre sidefelt: strategy-velger med to alternativer — Angular (longitudinal) aktiv (justerer folds etter lengde-/breddesektorer på sfæren slik at hver test-fold er geometrisk tett) vs Linear

(`round-robin`) (rekkefølge-basert, alle k -te frames som test-sett). k -Folds-slider står på 5, test-fold-velger på fold 1. Export-knapp genererer en `fold-assignment.json` for Nerfstudio/Instant-NGP. Midt-panel: 3D-globus-projeksjon av alle 100 kameraer — grønne punkter = train, røde punkter = aktuell test-fold (fold 1 med 20 kameraer). Høyre sidefelt (Angular Correlation): per 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 verdi betyr at kameraene innen denne folden ligger tett sammen, altså at holdout-splittet er romlig koherent.



Hva det er: En 3D-visualizer for kameraarrangementet ditt med cross-validation-logikk. Du laster inn en `transforms.json` (standardformatet fra Nerfstudio / Instant-NGP for kameraposer), appen leser alle kameraer, projiserer blikkretningene deres på en enhets-kule og viser dem som små kulemarkører på en virtuell globus. Deretter deler den kameraene inn i k folds (etter valgt strategi: angular eller linear), markerer grønt for treningsdelen og rødt for testdelen (holdout) og beregner per fold en angular-correlation-score som forteller deg hvor langt test-folden ligger fra trenings-folden i blikkvinkel-rommet.

NÅR ÅPNE Når du vil gjøre holdout-evaluering — altså: hvor godt generaliserer modellen din til usette blikkvinkler? Standard i treningen er „every-8th view som holdout« (Mip-NeRF360-konvensjon), men det er en veldig lineær oppdeling. Hvis bildene dine f.eks. er tidsmessig klyngete (først én side av objektet, så den andre), er „every-8th« ikke representativ — en tilfeldig sekvensposisjon lander i testen, men alle naboene er i treningen, det er for lett. Med „angular« stratifiserer man i stedet over blikkvinkel-rommet: hver fold inneholder kameraer fra alle deler av orbiten, slik at testen virkelig prøver generaliseringshull.

SLIK TOLKES DET Angular vs Linear:

- Angular (standard): deler kameraene etter longitudinalvinkel (ϕ -koordinat rundt Y-aksen) i k like sektorer. Fold 0 er kameraer med $\phi \in [0^\circ, 360/k^\circ)$, fold 1 de neste, og så videre. Fordel: hver fold dekker en delregion av orbiten; test-folden er romlig kompakt, men bredt fordelt over verdens-datasettet. God for klassiske orbit-opptak.
- Linear (round-robin): fold-indeks = $(\text{image_index} \bmod k)$. Det er den enkle „every- k -th«-oppdelingen. Fungerer hvis bilderekkefølgen ingen spatial bias har (f.eks. tilfeldig sorterte droneopptak). Fungerer dårlig når bildene klynger seg tidsmessig.

I 3D-globusen ser du straks: grønne punkter (trening) og røde punkter (test). Hvis de røde punktene alle klynger seg i et hjørne, er holdout dårlig (ingen god generaliserings-test). Hvis de ligger jevnt mellom de grønne, er den god. Angular-correlation-scoren per fold (høyre sidefelt, i grader) forteller dessuten: mindre verdi = testen er tett på treningen (hvert test-kamera har et nært treningskamera, lett test); større verdi = testen er langt fra treningen (hardere generalisering).

 Du har tatt opp truck-scenen din med 251 bilder, eksporterer via meny punkt M33 (Export SfM transforms.json) en nerfstudio-fil. Åpne holdout-vinduet ()H, last inn JSON-en via „Open transforms.json...«, kikk på globusen. $k=5$ (default) gir deg 5 folds. Klikk på „Fold 3« — se om de røde markørene er noenlunde jevne. Hvis ja: „Export fold-assignment.json«, legg den eksporterte filen i reports-mappen, og ved neste training-run med `--benchmark` (eller tilsvarende Inspector-innstillinger) brukes nøyaktig denne fold-oppdelingen som test-holdout — i stedet for standarden „every-8th«.

W23 Knappen „Open transforms.json...«

Toolbar øverst til venstre.



Åpner en filvelger, begrenset til JSON-filer. Etter bekreftelse laster holdout-modulet filen. Lasteren parser både nerfstudio-formatet (kamera-intrinsics pluss liste over frames med bildestier og transform-matrise) og instant-ngp-formatet (samme oppbygning). Per frame trekkes blikkretningen ut fra transform-matrisen (kameraets z-akse i lokalbasis) og lagres. Hvis parsing feiler, vises en feilmelding i status-området.

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

 KORT FORKLART

Laster kamera-pose-JSON-en din. Standard er Nerfstudio- og Instant-NGP-eksporter. RadiancKit selv kan eksportere transforms.json via meny → Export → SfM.

W24 Velger „Strategy« (angular/linear)

Venstre sidefelt, øverst.



Radiovelger med to alternativer: Angular og Linear. Strategy-bytte utløser automatisk en gjenberegning av folds. Blikkretningene er en liste over 3D-enhetsvektorer på sfæren; angular-strategien projiserer dem på longitudinalvinkelen ϕ og sorterer, linear-strategien gjør bare en modulo-oppdeling over frame-index.

 KORT FORKLART

Angular for jevne orbit-opptak (standard, trygt), Linear bare hvis bildene dine ikke klynger romlig.

W25 Slider „k Folds«

Venstre sidefelt, i midten.



Slider fra 3 til 10, skrittstørrelse 1. Ved endring utløses fold-beregningen automatisk, slik at folds-listen, train/test-indices og per-fold-scoren beregnes umiddelbart. Den valgte verdien vises som monospaced-digit-tekst til høyre for labelen.

Tommelfingerregel: $k=5$ er standard (gir deg 20 % test per fold, noe som er vanlig for cross-validation). $k=10$ hvis du har mye data og trenger flere folds for statistisk sigende verdi. $k=3$ hvis du har lite data.

 KORT FORKLART

Hvor mange folds oppdelingen har. 5 er standard og passer nesten alltid.

W26 Velger „Test Fold«

Venstre sidefelt, under k-sliden.



Menyvelger. Alternativer er dynamisk $0..<k$, label „Fold 1“ til „Fold N« (altså 1-indexed i UI-en, 0-indexed internt). Hvis den tidligere valgte indeksen er $\geq k$ (f.eks. fordi du har redusert k fra 10 til 5), nullstilles den automatisk til 0. Den valgte test-folden vises rødt i globusen, alle andre grønt.

 KORT FORKLART

Hvilken fold som akkurat nå er test-folden. Du kan klikke deg gjennom og se hvordan hver enkelt fold ser ut i globusen.

W27 Knappen „Export fold-assignment.json«

HVOR

Venstre sidefelt, nederst.



TEKNISK

Åpner en lagre-dialog med default-filnavn `fold-assignment.json`. Etter bekreftelse koder holdout-modulet den aktuelle oppdelingen til et JSON-skjema (per-frame fold-tildeling pluss strategy-meta-blokk). Denne filen kan så gis med til neste trening med `--benchmark`, slik at samme holdout brukes for den endelige metrikk-evalueringen. Skrivefeil vises som feiltekst; suksess i grønn tekst som „Saved to (filename)«.

KORT FORKLART

Lagrer den aktuelle train/test-oppdelingen som JSON. Denne filen kan du så gi direkte med ved treningen, slik at samme testsett brukes igjen.

W28 SCNView (3D camera globe)

HVOR

Midt-panel i holdout-vinduet.



TEKNISK

SceneKit-globus-view. Scenen består av: en wireframe-kule (radius 1.0, 36 segmenter, mørkegrå), tre fargede aksestumper (rød/grønn/blå for X/Y/Z, hver 1.2 lang) og per kamera en liten markørkule (radius 0.03) ved den tilsvarende blikkretningsposisjonen på enhetskulen (litt utenfor, slik at den ikke forsvinner i wireframe-kulen). Markørene gjenoppbygges IKKE ved hver fold-ændring — gjenoppbygging er bare nødvendig når frame-listen endres (altså når en ny JSON lastes inn). I stedet kjører en in-place-oppdatering av materialfargene per update: rødt for test-index, grønt for trening, lysgrå hvis hverken eller. Slik forblir slider-tikker performante også ved $N > 1000$ kameraer.

Kamerastyringen er aktivert — du kan rotere globusen med musen, zoome, panorere. Belysning sørger for at markørene ikke ser flate ut. Bakgrunn er mørkegrå.

KORT FORKLART

3D-globusen med kameraposisjonene. Grønn = trening, rød = test, lysgrå = ikke tildelt (forekommer ikke, alle kameraer hører til et sted). Med musen kan du rotere globusen og zoome.

W29 FoldCard (tap to select fold)

HVOR

Høyre sidefelt, „Angular Correlation«-seksjon.



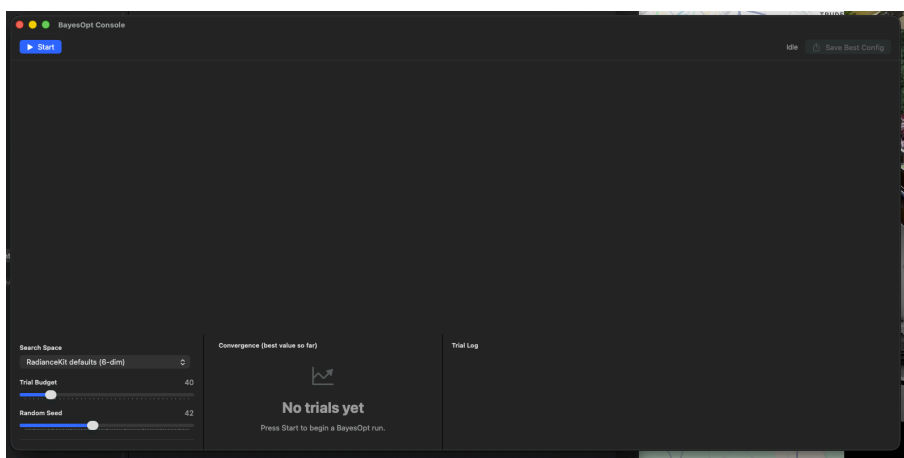
TEKNISK

Per fold en kortvisning — avrundet rektangel med 6 pt radius, padding 10, vertikalt layout med to linjer (øverst „Fold N« + kameraantall, nederst „Mean nearest angle:» + verdi i grader). Bakgrunnsfarge betinget: aktiv fold = accentfarge halvtransparent, inaktive = nøytralt standard-material. Tap velger folden, og globusen farger om live.

„Mean nearest angle«-scoren er den gjennomsnittlige minste vinkelen per test-kamera til det nærmeste treningskameraet (i radian internt beregnet, vist i grader i UI-en).

KORT FORKLART

Per fold et lite kort til høyre med antall kameraer og gjennomsnittsavstanden til det nærmeste treningskameraet. Klikk på det velger denne folden som test.

BayesOpt Console (W30–W39)

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

Tom tilstand med search-space-velger (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 (etter første åpning) — convergence-chart og trial-tabell fylles så snart en run er startet, se neste shot.



Figur 25: BayesOpt-konsoll etter 40 trials — convergence-chart stiger bratt inntil trial 15, Best Value 0.9943, trial log med init/bo/restart-tags

HVA BILDET VISER Status øverst til høyre „Finished — best 0.9943 after 40 trials«. Venstre sidefelt: search-space- velger på RadianceKit defaults (6-dim), trial-budget 40, random seed 42. Parameter-liste viser de seks hyperparametrene som skal tunes, med verdiområdene deres: 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, oransje punkter = restart-trials (#22 og #31). Beste-verdi-linjen stiger bratt til trial ~7, deretter bare marginal forbedring til trial 15, fra da av flatt platå ved 0.99+. Høyre sidefelt: trial-log #1-#34 med score + tag (init/bo/restart). Save- Best-Config-knappen øverst til høyre skriver `bayesopt-best.json`.

Hva det er: En Bayes-optimerings-konsoll for hyperparameter-søking. Bayes-Opt er en automatisk metode som prøver å finne det optimale punktet for en ukjent funksjon med så få eksperimenter som mulig — typisk: „hvilken kombinasjon av mcmcMaxGaussians, capMultiplier, ssimWeight og gradThreshold gir den beste PSNR for min sceneklasse?« I stedet for et grid på $6^4 = 1296$ trials prøver Bayes-Opt ca. 40-100 informerte trials og kommer dermed nær optimum.

Viktig: Den aktuelt leverte versjonen i appen kjører ikke optimeringen mot ekte trenings-runs (det ville ta dager), men mot et syntetisk demo-objektiv — et multi-modalt landskap med hill-climbing-karakter pluss litt noise. Det er bevisst slik: vinduet skal vise deg optimizerens atferd (konvergensforløp, sample-punkter, best-so-far) og la deg forstå search-space- definisjonene. For ekte treningsdrevne BayesOpt-kjøringer (som utført i fase Q7 for scene-class-presets) brukes et separat offline CLI-workflow; vinduet er live-UI-varianten.

NÅR ÅPNE Tre anvendelsestilfeller: 1. Du vil forstå hvordan BayesOpt arbeider — så start en demo-run og iaktta convergence-charten. 2. Du planlegger en ny sceneklasse (f.eks. „akvarier« eller „antikke møbler») som de innebygde 10 presetene ikke passer perfekt til. Definer mentalt et søkerom, sjekk det her med „Bowl demo« eller „Densify»-preset, eksporter så best-config som JSON og bruk det som startpunkt for en ekte trenings-run.

3. Du vil inspisere de default-search-spacene som er definert i

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

SLIK TOLKES DET

- **Convergence-chart** (midt-kolonnen): Y = beste hittil oppnådde objective-funksjon-verdi. X = trial-index. I starten bratt stigende (BayesOpt prøver de initial-samples tilfeldig, noen av dem er heldige), deretter tiltagende flatt fordi nær-optimum-regionen er uttømt. Hvis linjen forblir flat i 20+ trials, kan du stoppe kjøringen — flere trials gir ingenting mer. De enkelte punktene i charten er de individuelle trial-verdiene (altså ikke „best so far«), farget etter fase: grå = initial sample, blå = bayesopt acquisition, oransje = restart.
- **Trial-tabell** (høyre kolonne): #1, #2, #3, ... hver med verdi og fase-tag. Det hittil beste trial er markert med en gul stjerne. Ut fra tabellen kan du identifisere best-trial og se parameter-verdiene dens senere ved eksport.
- **Search-space-inspector** (venstre sidefelt): viser for det valgte presetet alle parameter-navn og søkeområdene deres `[lo, hi]`. Hvis du står ved presetet „RadianceKit defaults (6-dim)«, ser du f.eks. „densifyGradThreshold [5e-7, 5e-6]« — altså log-uniform mellom disse to verdiene.

📍 Velg preset „RadianceKit defaults (6-dim)«, trial-budget 40, seed 42. Klikk „Start«. Iaktta: de første 8 trials er grå (initial samples, LHS-Latin-Hypercube), de følgende blå (BayesOpt-ervertet). Convergence-charten blir bratt til trial ~15, deretter flater den ut. Ved trial ~30–40 stabiliseres den beste verdien. Klikk „Save Best Config« — en `bayesopt-best.json` lagres med preset-navn, trial-index, verdi og de dekodete parameter-verdiene. Denne JSON-en kan du så manuelt overta i preset-definisjonen din.

W30 Knappen „Start«

📍 HVOR

Toolbar til venstre, i idle/finished-state.

⚙️ TEKNISK

Nullstiller trial-listen, skifter til running-state, genererer en ny run-ID (for stale-detection ved flere start-klikk) og oppretter en frisk pause-gate. Deretter starter en bakgrunns-task som kjører optimeringen som asynkron stream. Initial-samples-størrelsen følger av $\min(8, \text{budget} / 4 + 1)$ — altså typisk 8 Latin-Hypercube-samples ved budsjett ≥ 28 , færre ved lite budsjett. Trial-updates mottas inkrementelt og legges til listen. Stale-run-beskyttelse: hvis et annet start-klikk i mellomtiden setter run-ID på nytt, kastes updates fra den gamle run bort.

Primary-action-stil for den prominente knapp-looken.

🗨️ KORT FORKLART

Starter en frisk optimeringskjøring med det aktuelle søkerommet, budsjettet og seedet.

W31 Knappen „Pause«

Toolbar til venstre, i running-state.



Setter pause-gate aktiv og bytter til paused-state. Den egentlige effekten: runneren venter i et 50-ms-polling-loop før den evaluerer neste objective-funksjon. Det betyr at et pågående trial føres til ende (det er syntetisk og varer bare mikrosekunder), men ingen ytterligere trial startes. Så snart resume kjører, fortsetter den der den slapp.

KORT FORKLART

Holder kjøringen tilbake. Aktuell beregning løper til ende, så pauser den.

W32 Knappen „Stop«

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



Avbryter runner-tasken, nuller referansen, løser pause-gate (hvis fortsatt paused), og bytter til finish-state (hvis trials finnes) eller idle-state (hvis ingen). De allerede beregnede trialene forblir synlige i listen — stop sletter dem ikke. Destruktiv knapp-rolle viser knappen rødt fordi den avbryter kjøringen.

KORT FORKLART

Avbryter kjøringen endelig. Trials forblir synlige, du kan fortsatt eksportere best-config.

W33 Knappen „Resume«

Toolbar til venstre, i paused-state.



Løser pause-gate og bytter tilbake til running-state. Runner-tasken kjører allerede (den venter jo i polling-loopet); så snart loopet merker at pausen er opphevet, kjører den videre og starter neste trial.

KORT FORKLART

Fortsetter en paused kjøring.

W34 Knappen „Save Best Config«

HVOR

Toolbar til høyre, alltid synlig (men deaktivert hvis ingen bestTrial finnes).



TEKNISK

Åpner en lagre-dialog med default-filnavn `bayesopt-best.json`, begrenset til JSON. Etter bekreftelse bygges en payload-dictionary: preset-navn, trial-index, verdi (objective-score), parametere (dictionary av dekodete parameter-navn → verdier). Dekodingen projiserer de normaliserte søkerom-koordinatene i $[0,1]^d$ tilbake til det opprinnelige verdiområdet (med log-uniform/linear/integer-skalaer tilsvarende). JSON-output er pretty-printed med sorterte keys. Ved skrivefeil ignoreres det (i den aktuelle demo-versjonen) stille — ingen error-UI, fordi det er en demo-sti.

Knappen forblir grå så lenge ingen trial er kjørt.

KORT FORKLART

Lagrer parameter-verdiene for det hittil beste trial som JSON. Du kan så manuelt overta disse verdiene i preset-konfigurasjonen din.

W35 Velger „Search Space«-preset

HVOR

Venstre sidefelt, øverst.



TEKNISK

Menyvelger med fire preset-alternativer:

- „RadianceKit defaults (6-dim)« — det fulle standardsøkerommet med alle Q7-hyperparametere.
- „Mip subset (2-dim)« — bare `mipSmoothing3DScale` [0.05, 0.5] log-uniform og `mipFilter2DVariance` [0.1, 0.6] linear. Nyttig hvis du vil tune Mip-splatting for en sceneklasse.
- „densify-until + ssim-weight + grad-thresh« — tre densify-relevante parametere (`densifyGradThreshold` log-uniform, `ssimWeight` linear, `densifyUntilIter` integer).
- „Bowl demo (1-dim)« — pedagogisk single-parameter-søkerom for „slik virker BayesOpt«-demoner.

Mens en kjøring er aktiv, kan søkerommet ikke byttes (ville forvirre optimizeren).

KORT FORKLART

Hvilket hyperparameter-søkerom BayesOpt gjennom søker. Standard er „RadianceKit defaults«. For målrettede Mip-tuning-forsøk „Mip subset«. For å forstå hvordan BayesOpt arbeider, „Bowl demo«.

W36 Slider „Trial Budget«

Venstre sidefelt, under search-space-velgeren.



Slider fra 10 til 200, skrittstørrelse 5. Default 40. Det betyr: BayesOpt kan maksimalt gjøre N trials. Av disse er de første par initial samples (Latin-Hypercube), resten er ekte BayesOpt-trials. Tommelfingerregler i praksis: et søkerom med d dimensjoner krever 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 kjøringen er slideren deaktivert.

 KORT FORKLART

Hvor mange optimeringsforsøk maksimalt. Flere forsøk = bedre løsning, men koster mer tid. 40 er god default for demo-objective.

W37 Slider „Random Seed«

Venstre sidefelt, under budsjettslideren.



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

til noise-komponenten i demo-objective. Reproduserbarhet: samme seed + samme søkerom + samme budsjett gir nøyaktig identisk trial-sekvens. Nyttig for „får alle kolleger samme kjøring hvis de bygger demoen etter?«. Under kjøringen deaktivert.

 KORT FORKLART

Styrer den tilfeldige generatoren. Samme seed = samme kjøring — for å reproducere.

W38 Chart (convergence)

HVOR

Midt-kolonnen i vinduet.



TEKNISK

Swift-Charts-diagram med to lag: 1. en linje for „best-value-so-far« per trial — en monotont stigende eller konstant kurve i accentfarge. 2. et punkt per trial med den individuelle objective-verdien, farget etter fase. Symbolstørrelse

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

En liten legende viser fase-fargene 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 FORKLART

Forløpscharten. Den ubrutte linjen er „beste hittil funne løsning«; punktene er de enkelte forsøkene. Hvis linjen forblir flat lenge, har BayesOpt funnet optimum.

W39 Table (trial log)

HVOR

Høyre kolonne i vinduet.



TEKNISK

Scroll-område med lazy stablede trial-linjer. Per linje et horisontalt stack: trial-nummer (3-sifret monospaced, til venstre), verdi (monospaced, høyrejustert, 70 pt bred), fase-tag (capsule, fylt med fase-farge ved 25 % opacity), valgfritt en gul stjerne hvis dette trial er det aktuelt beste. En auto-scroll-mekanisme hopper automatisk til enden så snart et nytt trial kommer, slik at du kan medlese live-forløpet i bunnen av skjermen uten selv å scrolle.

KORT FORKLART

Tabellen over alle forsøk. Verdi, fase, stjerne for det beste. Scroller automatisk med, nye trials dukker opp nederst.

Hovedvindu: loss-forløp og gaussian-count (I39–I41, krysshenvising)

Tre av Inspector-visningene i hovedvinduet fortjener en egen forklaring fordi de ses konstant under en kjørende trening, og det er viktige tommelfingerregler for når forløpet ser sunt ut. Visningene er i Inspector under „Loss Chart«-seksjonen (se kapittel 2 — Inspector) og supplerer holdout-analysen fra hjelpvinduet ovenfor.

Når er loss-kurven sunn? En sunn loss-kurve viser tre faser: (1) **Warmup** — i de første 200–500 iterasjonene faller loss-en bratt fra høyt (typisk 0.15–0.25 for L1+SSIM-kombinert avhengig av scenen) til ca. halvparten. Hvis loss-en IKKE faller i denne fasen, er input som regel feil (bilder defekte, SfM-posisjoner dårlige, antall initial-gaussians for lite). (2) **Densification** — mellom ~500 og densify-iteration (klassisk 15K, MCMC til 20K eller 25K) faller loss-en videre, ofte med små sprang nedover når densify-operasjonene setter inn nye gaussians og optimizere utnytter dem. Gaussian-count stiger i denne fasen. (3) **Refinement** — deretter går loss-en ut i en flatere hale. Typiske sluttverdier: 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).

Hva betyr et platå? Et platå (loss-kurven løper horisontalt over flere tusen iterasjoner) har to tolkninger: (a) modellen er konvertert, ytterligere trening gir ingenting mer — det er den gode situasjonen. (b) Modellen er stuck (lokalt minimum, dårlig gradient-informasjon, et cap ved buffer-grensen) — den dårlige situasjonen. Begge ser identiske ut i charten. Skille: kikk på gaussian-count. Hvis den også er flat OG tett på MCMC-cap-et (f.eks. 150K av 150K ved `.fullMCMC`), er du ved grensen — enten hev cap-et eller aksepter platået. Hvis gaussian-count fortsatt vokser, men loss-en ikke faller, henger den fast.

Når avbryte vs videretrene? Tommelfingerregel: 10K iterasjoner uten forbedring av min-loss → avbryt, ytterligere iterasjoner er bortkastet. Før: kan du via Cmd+T (training-menyen → Continue Training → +5K iterations) fortsatt henge en forlengelse på hvis du ser grensemessig forbedring. Advarsel: ved MCMC er platået ofte ekte — cap-et er den naturlige grensen.

Gaussian-count-platå er IKKE et „ferdig«-signal. Det betyr bare at MCMC har nådd cap-et, eller at Classic densification er uttømt. Selve „ferdig«-spørsmålet stilles først av holdout-analysen — PSNR/SSIM/LPIPS på et uavhengig test-sett, evaluert i holdout-vinduet (W23–W29) eller via `--benchmark` -flagg.

PSNR/holdout er sannheten, loss kun proxy. Loss-en er en relativ metrikk: den faller mens modellen din tilpasser seg trenings-views. En lav loss betyr imidlertid ikke automatisk god modell — hvis modellen har lært treningsbildene utenat (overfitting), ville loss-en være liten, men PSNR på usette views (holdout) ville være dårlig. Derfor: for den endelige kvalitetsbedømmelsen kikk alltid på holdout-metrikker, ikke på end-loss alene.

Tommelfingerregel-boks

- User Guide og Keyboard Shortcuts er statisk hjelp — ved stikkordsspørsmål raskt, for dybde bruk denne manualen.
- Manage Storage åpnes så snart disken faller under 10 % fri plass. Logs og imports-staging er de vanlige synderne.
- Pareto Dashboard er først meningsfullt etter minst tre eller fire treningsreports. X-akse = kostnader (time / Gs), Y-akse = kvalitet (PSNR / SSIM). Pareto-fronten viser de effektive kombinasjonene.

- Holdout Analysis brukes før du offentliggjør PSNR-benchmarks med andre — det sikrer deg at test-settet ditt virkelig er representativt.
- BayesOpt Console er primært et lærings- og inspeksjonsverktøy for søkerom-definisjoner. For ekte treningsdrevet hyperparameter-tuning bruk offline CLI-workflowen.
- Loss-platå og gaussian-count-platå må tolkes adskilt. Cap-grense er ikke et „ferdig«-signal. Ekte kvalitet måler bare holdout-PSNR.
- 10K iterasjoner uten min-loss-forbedring → stopp trening.

KAPITTEL

Kapittel 6 — Trenings-konfigurasjon

```
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 eksportert 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)

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

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

Dette kapittel 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 tvil gjelder den dér gemte doc-comment og `initializer-default`'en som source-of-truth.

MERK · UI VS. PRESET/CLI

Kun 12 av 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** (kapittel 7) og kan kun overskrives direkte via **CLI-flag** (se kapittel 5). Denne adskillelse er tilsigtet: defaults forblir stabile og produktionsafprøvede, power-users har likevel en flugtluge. Hvis et felt særligt interesserer dig: kig først i kapittel 2 (Inspector) og kapittel 5 (CLI), om du kan nå det uden JSON-pillerier.

Innholdsfortegnelse:

1. Iterasjon (T1–T2)
2. Learning rates (T3–T10)
3. Densification — Classic (T11–T16)
4. Loss (T17–T20)
5. SH-grad-progresjon (T21)
6. Performance (T22–T25)
7. Diagnose og punktsky-forberedelse (T26–T30)
8. Regularisering (T31–T37)
9. Forfining (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. Statistiske presets (TP1–TP9)
18. Metode:
19. Hvilket felt til hva? (cheat-sheet)
20. Farlige felter

Iterasjon (T1–T2)

T1 maxiterations

DETLJER

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 treningsiterasjoner, som backend gjennomløper. En iterasjon består av et forward-render av et enkelt trenings-kamera, et backward-pass 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 iterasjoner (sessioner 1–32, V546-tests), ved MCMC ved 60 000–200 000 (V534). En drastisk forhøjelse ud over de i presets gemte verdier giver sjældent ekstra kvalitet — Adam-momentum mætter, og uden LR-decay-ende stagnerer loss'en. Omvendt fører underskridelse av ~5 000 til ufuldstændigt konvergerede geometrier (density-control har for lidt tid til klon/split).

KORT FORKLART

Hvor længe appen regner. Flere iterasjoner = bedre resultat, men på et tidspunkt ikke lenger mærkbart bedre, til gengæld meget længere. Presets er valgt sådan, at du uden at tænke har en god verdi: 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. Inntil da kloner, splittes og prunes gaussians etter reglerne parametriseret i T11–T16 (Classic) eller T67–T70 (MCMC); deretter forblir gaussian-antallet konstant, og kun positioner, rotationer, skaler, opaciteter og SH-koeffisienter optimeres (refinement-fase). I 3DGS-originalpapiret ligger verdien ved 50 % av T1, i Radiances Kits `.full`-preset ved kun ~14 % (5 000 av 35 000) — følge av V310/V338-eksperimentet, der viste, at videre densifisering etter 5 000 iterasjoner snarere forverrer resultatet (flere floaters, mere lagerbehov, ingen kvalitetsgevinst). MCMC derimot lader relocation køre til 80 % av T1 (V504b), fordi MCMC ikke produserer skadelige floaters. Hvis T2 velges for lille (< 1 000), oppstår der for få gaussians; for stor ved Classic (> 50 % av T1) fører til overgrowth og RGB-saturation-outliers (se outdoor-overtraining-findings).

KORT FORKLART

Hvor længe appen må generere nye gaussians. Deretter forfines kun det, der allerede er. Ved klassisk trening med 35 000 iterasjoner er 5 000 her den riktige verdi — alt derover gjør scenen mere plumret. Ved MCMC er det 80 % av samlede iterasjoner (altså 160 000 ved 200 000-kørsel). Hvis du ændrer Quality-preset, lad heller dette felt være.

Learning rates (T3–T10)

T3 positionLearningRate

DETALJER

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

TEKNISK

Adam-læringsrate for XYZ-posisjonen av hver gaussian ved treningsstart (iterasjon 0). Følger en cosine-annealing-kurve og synker i treningsens forløb til T4 `positionLearningRateFinal`. Default 0.00016 stammer fra 3DGS-originalpapiret (Kerbl et al. 2023) og kan i RadiancKit ikke skaleres med bildeopløsningen — posisjonen beveger seg i verdenskoordinatsystemet, ikke i pixel-rummet. En markant forhøyelse (> 0.0005) bevirker, at gaussians springer over lange distanser, og loss'en blir ustabil; verdier markant under (< 0.00005) fører til, at feil-initialiserte punktskyer aldrig finner deres plass. V414 testede en fordobling av init-værdien \rightarrow 16.8 % dårligere L1-loss; V544a-tuning bekræftede paper-default'en som optimal. Merk: ved `.fullMCMC` lader vi denne verdi bevidst stå ved default — MCMC har brug for konstante læringsrater til sin relocation-logik, så tuning her giver intet.

KORT FORKLART

Hvor raskt splat-punktene må bevege seg i rummet. Standardværdien er meget veljusteret og kræver i virkeligheden ingen ændring. Kun hvis splats i bildet „flagrer« eller et helt hjørne mangler, fordi intet vil bevæge sig dertil, ville læringsraten være et punkt at dreje på — men så passer typisk noe andet allerede ikke (kamera-posisjoner, initialpunktsky).

T4 positionLearningRateFinal DETALJER

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

 TEKNISK

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

 KORT FORKLART

Hvor sakte position-læringsraten blir i slutningen av træningen. Vi har bevidst sat den mindre aggressivt end originalpapiret — splats kan stadig wackle lidt til slutningen, det gjør dem skarpe. Hvis du drejer på det: høyere = 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 — 2×) **Range:** 0.0001 – 0.05
Defined in:

 TEKNISK

Adam-læringsrate for DC-andelen (degree 0, altså konstant albedo) av spherical-harmonic-farven. SH-DC svarer til den retnings-uafhængige grundtone av en gaussian, så at si „basisfarven«. V176- og V188-eksperimenter fant 2× høyere end paper-default'en optimal — raskere farve-konvergens, særlig fordi SH-DC ved kort trening (5 000 iterasjoner) ellers ikke kommer i form. I modsætning til de geometriske LR'er har SH-DC ingen decay; læringsraten forblir konstant over alle iterasjoner (eller følger kun den valgfri extended-phase-decay fra `T51`). V416 testede en firedobling til 0.01 → 6.4 % dårligere loss ved $\beta_2=0.99$ -Adam.

 KORT FORKLART

Hvor raskt grundfarven for hver splat tilpasser sig. Værdien endrer man næsten aldrig selv — presets har den rigtige verdi. Høyere ville gå raskere, men kan føre til ustabile farver.

T6 shRestLearningRate DETALJER

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

 TEKNISK

Adam-læringsrate for SH-koeffisienterne av høyere orden (degree 1, 2, 3 — altså de view-direction-afhængige farveandele, der sørger for høylys, refleksjoner og blød skygning). 20× mindre end `T5` ifølge paper-konvention, fordi disse koeffisienter 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 overmætte bildet. Frigives i to trin — inntil første markering i `T21` `shDegreeUpgradeIterations` er kun degree 0 aktiv (altså kun `T5`), deretter 1, senere 2, til sidst 3. Lave verdier her er særlig viktige på scener med meget diffus belysning; ved meget blanke overflader (billa, vand) giver det ingen mening at dreje på det — SH-repræsentasjonen selv er begrenset.

 KORT FORKLART

Hvor raskt de retnings-afhængige farve-effekter (refleksjoner, glans) lærer. Som standard meget lille, fordi ellers blanker alt. Værdien lader man bedre stå — den, der vil få bedre høylys, er bedre tjent med MCMC og længe-re treningstid end med denne LR.

T7 `opacityLearningRate` DETALJER

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

 TEKNISK

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

 KORT FORKLART

Hvor raskt splats blir gjennom-sigtige eller uigennem-sigtige. Vikt-igt for opryddningen — splats, der ikke bidrager med noe, skal forsvinne raskt, så der ikke op-står en slør. Standardværdien passer, kun professionelle ænd-erer 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 deaktivert, og opacity-LR'en forblir 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 igjen av tilfældige gradient-udsving. Empirisk bekræftes det ikke — logit-clamping-logikken fanger det likevel. Feltet forblir tilgjengelig for fremtidige eksperimenter; også meget lange MC-MC-kørsler ($> 500K$ iterasjoner) kunne drage nytte av det.

 KORT FORKLART

Om opacity-læringsraten skal bli mindre mod slutningen. Stan-dard: nej. Vi har prøvet det, var dårligere, lader det deaktivert. Bli 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 av hver gaussian i log-space (RadianceKit gemmer $\log(\text{scale})$, så skalaerne forblir positive). Paper-default 0.005, i RadianceKit fordoblet til 0.01 for bedre skala-konvergens ved de optimerede læringsrate-konfigurasjoner. V423-eksperiment: 0.005 med $\text{beta2}=0.99$ -Adam \rightarrow 18.7 % dårligere loss og synlig for få gaussians (density-control kunne ikke klone, fordi skala- updates var for sakte). Skala kontrollerer udstrækningen av hver gaussian — for rask læring fører til „needle«-gaussians (ekstremt lange tynde splats, se T34 `scaleRatioPruneThreshold`), for sakte læring lader splats forbli for kompakte, og density-control skal splitte for ofte.

KORT FORKLART

Hvor raskt formen for splats tilpasser sig. Standard er god. Hvis du skruer det opp, får du gjerne „nåle«-splats — ekstreme lange tynde dråber, der får bildet 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 av hver gaussian. Quaternion'en normaliseres igjen i hvert optimizer-skridt etter Adam-update'en (L_2 -norm = 1) — ellers ville kovariansmatrisen bli degenereret. RadianceKit fordobler paper-default'en i quality-presets, fordi rotation har mindre absolute gradient-magnituder end skala / position (på enhedssfæren forblir hvert skridt kort), og uten 2x ville rotationen i 35 000-iterations-vinduet være markant under- konvergeret. V188 dokumenterer. På NeRF-Blender-scener (Lego, Chair) virker rotation særlig — objekternes kanter retter sig først korrekt ud etter 5 000–10 000 iterasjoner.

KORT FORKLART

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

Densification — Classic (T11–T16)

T11 densifyGradThreshold

DETALJER

Default: 0.000002 (initializer, kalibreret for 0.5x oppløsning), 0.000001 (`.full` , kalibreret for 1.0x), 0.000004 (`.quickTest` , kalibreret for 0.25x), 2e-7 (`.fullClassicPaper`) **Range:** 1e-8 – 1e-3 (opløsningsafhængig) **Defined in:**

TEKNISK

Tærskel for L2-normen av den skærmmrums-projicerede gradient `dMean2D` , over hvilken en gaussian markeres til klon eller split. Den absolutte verdi afhænger direkte av trenings-oppløsningen — `dMean2D` skalerer ca. som $1/\text{opløsning}^2$ (flere pixel = mindre per-pixel-gradienter). Derfor kræver hvert T22 `trainingRenderScale`-trin en kalibreret tærskel: 0.25x \rightarrow 4e-6, 0.5x \rightarrow 2e-6, 1.0x \rightarrow 5e-8 ... 1.1e-6 (`.full`). Paper-default 0.0002 er NDC-normaliseret og ikke direkte sammenlignelig i RadianceKits world-space-pipeline. Med V440 indkoblet 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 forblir fra. Q5 (T77–T79) leverer en alternativ adaptiv logik via rolling median. **Dette felt er ikke ufarligt** — halvering genererer 2–4x flere gaussians (minne-pres, OOM-risiko); fordobling kan under-densificere scenen.

KORT FORKLART

Hvor følsom appen er, når den skal beslutte, om et splat er for lidt repræsenteret og skal mangfoldiggøres. Lav verdi = mere følsom = flere splats. Højere = færre splats. Det er en av de farligste verdier overhovedet: for lav, og Mac'en fyldes med millioner av 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 iterasjon, fra hvilken densification blir aktiv. Før det sker kun „nøgen« læring på den initiale SfM-punktsky, uten at nye gaussians genereres. Default 500 stammer fra 3DGS-papiret og giver initialiseringen tid til at stabilisere sig — hvis der allerede densificeres fra iterasjon 0, klones forkert positionerede SfM-punkter mange gange, før de overhovedet finner deres rigtige plass. V349 testede 1000 → let dårligere loss; default'en er optimal.

KORT FORKLART

Hvornår appen første gang begynner 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 mangfoldiggjør.

T13 densifyInterval

DETLJER

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

TEKNISK

Hvor mange iterasjoner der ligger mellom to densification-skridt. Ved paper-default 100 — hver 100 iterasjoner evalueres listen av densify-kandidater, kloner/ splittes, og samtidig fjernes listen av prune-kandidater ($\text{sigmoid}(\text{opacity}) < T_{14} \text{pruneOpacityThreshold}$). V112-tests fant 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 etter 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-spesifikke logik.

KORT FORKLART

Hvor ofte appen kigger etter nye splats. 100 = ofte, 200 = mellom. Høyere betyder: hvert splat har lengere tid til at indrette sig, før det mangfoldiggjøres igjen. Det er godt. At senke til 50 kan optage GPU'en vedvarende uten, at det blir nævneværdigt bedre.

T14 pruneOpacityThreshold

DETLJER

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 neste densification-step. Virker sammen med T7 `opacityLearningRate` og `logit-clamp`-logikken i optimiseren. V393 sænkedde default'en fra 0.005 til 0.001 i `.full` — følge: splats, som kun spiller en rolle under eksotiske kigge-vinkler, bevares lenger og bidrager til SH-detalle. V394 testede 0.0001 → let dårligere (for lidt pruning, minne spildt). Viktigt: `density-control` skal ALTID prune, selv hvis buffer-kapasiteten allerede er full av andre tiltag (se „Density Control Must Always Prune« i CLAUDE.md) — ellers akkumulerer døde gaussians, og count fryser fast.

KORT FORKLART

Hvornår et splat regnes som „gennemsiktig nok« til at bli slettet. 0.005 er paper-standarden, vi har i Quality 0.001 — altså vi gir splats en lenger chance. Det gjør blødt lys og svage skygger bedre at fremstille. At sette høyere (over 0.01) lader splat-antallet raskt synke — kan være meningsfullt ved minneknappe, men koster detalje.

T15 opacityResetInterval

DETLJER

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

TEKNISK

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

KORT FORKLART

Hver hvor mange iterasjoner appen nulstiller synligheten av alle splats til „næsten usynlig« — en slags reset-knap for opacity. Ved os deaktivert (verdi så høy, at det aldri skjer), fordi andre mekanismer gjør det overflødig. Kun ved paper-tro eksperimenter at slå til.

T16 maxScreenSize DETALJER

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

 TEKNISK

Maksimal skærmsstørrelse (i projiserte pixel), som en gaussian må nå, før den tvunget splittes. Verdien er satt til 0 (V48 testet og revertert) — RadianceKits density-control bruker 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 av det. Aktivering (verdi > 0, f.eks. 20) ville tvinge splats, der er blitt meget store på skærmen, til at dele seg — relevant ved store, glatte vægoverflader, hvor et enkelt riesensplat tilbyr for lidt detalje.

 KORT FORKLART

Begrænsning på, hvor stort et enkelt splat må bli 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 udtrykkelig eksperimenteres med det.

Loss (T17–T20)

T17 ssimWeight

DETLJER

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

TEKNISK

Vægt av D-SSIM-andelen i den kombinerte loss-funksjon $loss = (1 - \lambda) * L1 + \lambda * D-SSIM$, hvor $\lambda =$ T17. 3DGS-paper-default 0.2 er optimal for Classic-densification — V383 testede 0.3 → 28.9 % dårligere, V373b bekreftede 0.2 som sweet spot. For MCMC ble det i V521b/V534 uafhængigt fastslået: 0.05 er optimal, fordi MCMC via sin stokastiske eksplorering har brug for en stærkere L1-signal-andel — højere SSIM-vægte ville udvandre relocation-beslutningerne. SSIM er markant dyrere at beregne end L1 (lokale 11×11-vinduer over hele bildet); RadianceKit bruger en MPS-accelerert implementering, der blir under 1 ms pr. 1080p-bilde. Q7-BayesOpt-sweeps fandt scene-specifikke optima mellom 0.05 (`.outdoorPreset` : 0.082) og 0.171 (`.indoorPreset`).

KORT FORKLART

Hvor viktig appen ud over „hver pixel stemmer« også finner „strukturer er ens«. 0.2 er standarden og leverer et godt bilde. Lavere = pixel-nøjtligere, men kan få blødere overgange. Højere = mere strukturelt, men detaljer blir 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-verdi for refinement-fasen etter 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 etter densification — når der ikke lenger opstår nye gaussians — en stærkere SSIM-andel ville maksimere den strukturelle skarphed. 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 forblir tilgængeligt for fremtidige eksperimenter med perceptual loss (T60) eller edge-loss (T19), hvor en refinement-specifik loss-komposition kunne være meningsfuld.

 **KORT FORKLART**

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

T19 **edgeLossWeight** **DETALJER**

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

 **TEKNISK**

V437-eksperimental-loss: vægt av 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 hjørnestein av billedkvalitet, og en eksplisitt term burde opmuntre gaussians til bedre at træffe kanterne. Test-resultater: vægt 0.1 → 11 % dårligere loss, 0.01 → kvalitets-neutral, men 10 % saktere. Sobel-passet koster et yderligere MPS-forward på ground-truth og render. Derfor permanent deaktivert. 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 FORKLART**

Eksperimentel tilføjelse, der gjør kanter ekstra viktige. 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 bilde 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 — noe som oppfattes som „floater«. Uden sky-mask ville loss'en i dette område aldrig bli nul, fordi himlen i bildet varierer let, og appen evigt forsøker 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 FORKLART

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-grad-progresjon (T21)

T21 shDegreeUpgraderIterations

DETLJER

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 verdi i [0, maxIterations] , monotont stigende **Defined in:**

TEKNISK

Iterasjoner, 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), etter første markering DC + 3 degree-1-koeffisienter, etter anden markering + 5 degree-2-koeffisienter, etter tredje markering alle 15 koeffisienter. 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 høyere 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 iterasjoner og kun bruker optimerer-kapasitet. Q6 (T80–T81) tilbyder en alternativ curriculum-logik, der dynamisk overskriver denne liste.

KORT FORKLART

På hvilke punkter i træningen appen lærer, at farver kan se forskjellige ud fra forskjellige synsvinkler (højlys, refleksjoner). Først sent — så formen først stemmer, så farven. Værdierne i presets er sat sådan, at det fungerer godt. Lad være med at ændre noe, 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-oppløsning ved trening relativt til trenings-billedernes oprindelige oppløsning. Ved 0.5 nedskales hvert bilde til 50 % bredde × 50 % høyde (altså 25 % av pixlerne), og gaussian-renderingen sker i denne mindre oppløsning. Reduserer både minne- og regneindsats kvadratisk. Viktig: `T11 densifyGradThreshold` skal passe til den valgte oppløsning — gradient-magnituder skaleres med $1/\text{oppløsning}^2$, derfor har `.quickTest` (0.25×) en meget høyere tærskel ($4e-6$) end `.full` (1.0×, $1.1e-6$). RadianceKit advarer ved meget store bilder og tilpasser automatisk — 3-MP-mål-oppløsning. Ved ekstreme 4K-input-bilder ville 0.5 eller endda 0.25 være meningsfullt, ellers kører selv Mac'en kun i CPU-compaction.

KORT FORKLART

Hvor store bildene er ved treningen. 1.0 = original, 0.5 = halv størrelse. Halv størrelse = fire ganger raskere, men de fineste detaljer mangler. Presets velger den riktige verdi; ved ekstremt store input-bilder (over 12 megapixel) skifter appen automatisk ned.

T23 resolutionWarmupScale

DETALJER

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

TEKNISK

V133-optimering: tren densification-fasen (iter 0 til `T2`) i en lavere oppløsning end refinement-fasen. V308 har deaktivert den igjen for `.full`, fordi tids-gevinsten ved `T22 = 1.0` og cosine-annealing var marginal, og kvaliteten led minimalt. Forbliver i feltkataloget, fordi den kunne bli meningsfull igjen ved 4K-input og lange trenings-kørsler — Q6 curriculum (T80) har taget en lignende logikk opp, men der er den koblet til LR-schedulen. Hvis aktivert og `T80 curriculumResolutionRamp` likeledes true, vinder Q6 og overskriver denne verdi.

KORT FORKLART

Spesial-feature: i første treningshalvdel læres med mindre bilder, 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 av rasteriserings-tiles i pixel. Gaussian-splattung-renderingen er tile-basert: bildet opdeles i 16x16-pixel-fliser, hver flise samler de for den relevante gaussians, sorterer dem etter dybde og blander dem ind. 16 er standard, som praktisk talt alle 3DGS-implementeringer bruker, og er hardkodet i RadianceKits Metal-kerneler; en ændring av denne verdi ville kræve re-kompilering av shaderne og er ikke effektiv i den aktuelle stand. Forbliver som felt, hvis en fremtidig engine-versjon understøtter tile-size dynamisk.

 KORT FORKLART

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 mellom treningsiterasjoner i millisekunder. 0 = fuld hastighed (standard). Højere verdier gjør Mac'en mere „brugelig« under træningen ved, at GPU/CPU regelmæssigt får pusterum — andre apps' brugervenlighed stiger, men træningstiden også lineært med forsinkelsen. Typiske verdier: 1–2 ms („let« throttling, +5 % træningstid, Mac føles mere responsive), 5 ms („mellom«, +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 FORKLART

Hvor mange millisekunder pause appen tager mellom trenings-skridt. 0 = ingen pause, hurtigst muligt. Højere verdier gjør 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 verdi.

Diagnose og punktsky-forberedelse (T26–T30)

T26 depthDistortionWeight

DETALJER

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

TEKNISK

V366-eksperimental: vekt av en depth-distortion-regulariserings-loss. Straffer gaussians, der langs en render-stråle ganske vist er dybt stablede, men konseptuelt hører til samme overflate — det oppmuntrer konsentrerte dybdefordelinger og reduserer 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 tilgjengelig for multi-view-datasæt med særligt rene poser (synthetic, Mip-NeRF 360 med ground truth).

KORT FORKLART

Eksperimentel feature til at unngå flere splats bagved hinanden på samme sted. Ikke aktivert, fordi testene ikke har givet noe.

T27 singleViewOverfit

DETALJER

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

TEKNISK

Diagnose-flag: hvis true, brukes i hver treningsiterasjon tvunget kamera-index 0 i stedet for en tilfeldig 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 etter 10 000 iterasjoner), er der en fundamental bug i forward/backward-passet. Denne kontakt ble brukt intensivt under utviklingen av Metal-shaderne og differentiable-rasterizer-kernels — V42–V47-fasen. I dag kun tilgjengelig som sanity-check, hvis noen har modificeret backend-kode og vil lave en regression test. Via CLI med `--single-view`.

KORT FORKLART

Test-tilstand for utviklere. De kan dermed tjekke, om appen overhovedet kan lære fra ÉT bilde. Irrelevant for almindelige brukere, lad alltid 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 bilder av en 1 500-bilde-droneflyvning?« Ikke eksponeret i UI.

 KORT FORKLART

Diagnose-felt for udviklere — brug kun de første N bilder, ignorer resten. Almindelig bruger har ikke brug for det, verdi 0 = alle bilder. Flere bilder = 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 av initiale SfM-punkter, som træningen starter med. Tætte COLMAP- rekonstruktioner kan producere > 60 000 punkter, noe som 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 treningsmotor) bringer initial-densiteten på niveau med, hva reference-3DGS bruger, og reducerer overlap dramatisk. Sættes automatisk ved meget tætte SfM'er; via CLI med `--max-points N`.

 KORT FORKLART

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 trenger 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 treningsmotor 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 10x 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 av årsagerne til fase-3.9-funnel-vs-training-anti-korrelationen: bedre SfM kan downstream føre til dårligere trening, fordi for mange initial-punkter dræbes. Feltet ligger som CLI-override (`--camera-cluster-outlier-multiplier`) til A.3-sweeps; ikke eksponert i UI. Verdier under 5 er som regel for restriktive, over 20 virkningsløse.

 **KORT FORKLART**

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 fjerntliggende punkter (bjerge i det fjerne) i bildet ser ud som svævende klumper. At sætte lavere kun i nødtilfælde — du mister dybde-detajle.

Regularisering (T31–T37)

T31 coarseToFineBlurRadius

DE TALJER

Default: 0 (= deaktivert) **Range:** 0 eller 1 – 10 **Defined in:**

TEKNISK

V369-eksperimental: box-blur-radius, der i starten av densification-fasen anvendes på ground-truth-bildet og lineært reduseres til slutningen av densification (T2) til 0. Hypotese: coarse-to-fine-trening – 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 basert på bilde-domæne-gradienten, og blur reduserer nettopp de signaler, der er viktige for „her skal kloner«. Forbliver i feltkataloget til fremtidige tests med annet density-control-skema.

KORT FORKLART

Eksperimentel „først-grov-så-detaljert«-tilstand. Har intet givet, forblir fra.

T32 scaleRegWeight

DE TALJER

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

TEKNISK

V370-eksperimental: L1-regularisering på verdensrumlig skala. Straffer gaussians, der blir for store – hindrer „mega-splats«, der dækker hele vægoverflader 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, noe som igjen betyder mere gradient-arbejde. Disabled, men dokumenteret til Mip-splatting-eksperimenter (T74): i denne kontekst kunne en skala-undergrænse være meningsfull.

KORT FORKLART

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

T33 anisotropyRegWeight DETALJER

Default: 0.0 (= deaktivert) **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 oppfattes som floaters. Tests: 0.01 → 69 % dårligere, 0.001 → 15 % dårligere. Grunden: regulariseringen tvinger splats mod „runde« former, noe som på en flad overflate (væg, bord, gulv) er nettopp 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 ble revertert.

 KORT FORKLART

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

T34 scaleRatioPruneThreshold DETALJER

Default: 0.0 (= deaktivert) **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 — bildet ble hullet. Derfor fra som default, CLI-flaget (`--scale-ratio-prune N`) forblir tilgjengelig til målrettede eksperimenter. Anbefalede verdier, hvis man likevel vil teste: 30 (meget konservativ, fjerner kun ekstreme outliers), 10 (aggressiv, koster detalje).

 KORT FORKLART

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

T35 `opacityRegWeight` DETALJER

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

 TEKNISK

V446-eksperiment: 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 deaktivert. Disabled. Advarsel: i 1.4.3-beta dukkede en bug op, der havde præcis dette felt med en default-verdi-ændring (initializer = 0.01), noe som førte til mass-extinction av gaussian-count (460 K → 5 på en iterasjon). Siden 1.4.4 fast på 0.0 som default.

 KORT FORKLART

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

T36 `opacityDecayFactor` DETALJER

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

 TEKNISK

V546-implementering av HTGS-skemaet (Hierarchical Time-Gating, Eurographics 2025): hver T37 `opacityDecayInterval` iterasjoner multipliseres sigmoid-opaciteten av hver gaussian med denne faktor. 0.9995×100 anvendelser giver ~95 %-tilbake 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 (inntil T2), deretter kører treningen videre uten decay, så de i refinement etablerte opaciteter forblir stabile. Bruges ikke ved MCMC (MCMC har egne mekanismer via T67 `mcmcRelocationInterval` 1. T68 `mcmcDeadOpacityThreshold`).

 KORT FORKLART

„Blid falmen« av alle splats over treningstiden. Gjør inaktive splats raskere transparente, så de fjernes ved opryddning. Var den viktigste 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 mellom to anvendelser sker nok gradient-updates til, at opacity stiger igjen. Korte intervaller (<20) gjør decay for aggressiv. Kun aktiv i densification-fasen.

 KORT FORKLART

Hvor ofte „falmen« anvendes. 50 = hver 50 iterasjoner et lille falme-skridt. Passer.

Forfining (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 utføres. 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 % saktere (fordi unfused-stien er dyrere end fused-stien). Reverteret for `.full`, men bevidst brukt til MCMC — `.fullMCMC` kører med, men V544a-tests viste, at quality-gap-pet til Classic skrumper til 5 % (i stedet for 11 %). I initializer-default 1, i aktuelle preset 1, forblir CLI-flag (`--accum-steps N`).

 KORT FORKLART

Hvor mange bilder appen ser, før den tilpasser splats. 1 = hvert bilde enkeltvis. Høyere = se flere bilder samtidig og anvend så en middelværdi. Giver intet i standardtilfældet; ved MCMC kan 2 hjelpe lidt.

T39 testViewIndices DETALJER

Default: `[]` (= tom, alle views brukes til trening)

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

 TEKNISK

V546-feature: sæt av camera-indices, som IKKE brukes til trening, 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 innstilling av 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 verdier mellom forskjellige datasæt.

 KORT FORKLART

Hvilke bilder der „udelades« ved træningen for senere at brukes til kvalitetsmåling. Du setter ikke det selv; `--benchmark` -flaget gjør det automatisk (hvert otteende bilde er test). Hvis du setter egne indekser: farligt, kan forfalske benchmarket.

T40 refinementPruneInterval DETALJER

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

 TEKNISK

V425-feature: hver N iterasjoner i refinement- fasen (etter `T2`) kjøres et yderligere prune-pas, der fjerner gaussians med `sigmoid(opacity) < T41 refinementPruneOpacityThreshold`. **Mening:** under densification er der regelmæssige density-control-kald, deretter ikke lenger — gaussians, hvis opacity fortsætter med at synke, forblir dog i bufferen. V425 testede og reverterede: den ekstra pruning korrelerte med V426 (two-phase densification, ligeledes avsluttet i 0-gaussians-cascade failure). Disabled. CLI-flag tilgjengelig til eksperimenter; hvis aktivert, er 1 000 eller 2 000 fornuftige verdier.

 KORT FORKLART

Yderligere opryddning under refinement-fasen. Giver intet, forblir 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. Etter densification har de fleste gaussians nået en markant høyere 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 fortsatt. Kun relevant, hvis T40 > 0.

 **KORT FORKLART**

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

T42 midTrainingCompactificationIterations **DETALJER**

Default: [] (= deaktivert) **Range:** [Int], verdier i (densifyUntilIteration, maxIterations) **Defined in:**

 **TEKNISK**

V549-feature: eksplisitte iterations-punkter under refinement-fasen, hvor et compactification-pas kjøres (fjerner sigmoid(opacity) < 0.01 + outlier-skala-gaussians, samme logik som T56 postTrainingCompactification). Mening: lange refinement-faser kan vise confetti-/floater-akkumulering, hvis SH så overfitter på view-spesifikke artefakter. Typisk konfigurasjon hvis aktivert: [10000, 20000, 30000] for 40K Classic. **MEN:** V549-A/B-tests på Family-dataset viste i alle konfigurasjoner 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" tilgjengelig, hvis man foretrekker den visuelle floater-tradeoff (mindre confetti i viewporten) over loss-regressjonen.

 **KORT FORKLART**

Imellem-opryddninger under treningen. 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 bilde.

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

V549b-feature: etter trening fjernes alle gaussians, der ligger uten for foreningen av alle treningskamera-frusta. Sådanne gaussians ble aldri begrenset av loss-signalet og er alltid floaters. Særlig effektivt for scener, hvor novel-viewet ligger bag eller ved siden av kamerastien (f.eks. bagsiden av en lineær dronflyvning) — floaters dér blir aldrig synlige i treningsfasen, men meget vel ved senere bevægelse i 3D-vieweren. V549b A/B på dronflyvninger positive resultater, derfor tilgjengelig 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 FORKLART**

Specialfilter til dronflyvninger eller lineære optagelser: etter træningen slettes splats, der ikke ble „set« i noe 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, noe som ville beskære vaklende 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 likevel kunne bli synlige. Verdier > 1.2 gjør cullen praktisk talt virkningsløs, fordi det udvidede frustum omfatter meget mere rum.

 **KORT FORKLART**

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 treningsstart genereres en kugleformet punktsky (Fibonacci-sphere med T46 sample-points), placeret i en radius av T47 $\text{skyDomeRadiusMultiplier} \times \text{scene_extent}$ omkring scenens midtpunkt og initialiseret med farverne fra de sky-maskerede pixler i alle treningskameraer (se T20 skyMaskingEnabled). Disse sky-dome-gaussians indsættes i begyndelsen av gaussian-bufferen og „fryses« under træningen (position/skala/rotation-gradient = 0, kun SH og opacity forblir 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 FORKLART

Slår en kunstig „himmelkuppel« til omkring scenen. Gjør 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 verdier → tettere sky-dome (bedre ved store oppløsninger og meget synlig himmel), men mere lagerbehov. 5 000 er sweet spot for 4K-renderinger; ved lavere oppløsninger rækker 2 000–3 000. Punkterne initialiseres etter cosine-distance til hver treningskamera-view-vektor med de tilsvarende sky-maskerede pixler — sample-points, hvis view-cone ikke ser noe kamera, forblir med lav opacitets-startverdi bagved, men ændres ikke i træningen (frosset).

KORT FORKLART

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

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

 **KORT FORKLART**

Hvor langt væk den kunstige himmelkuppel skal være. 30 = ret langt. Ved store landskaber er 50–60 bedre (Outdoor-preset gjør 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 begynnelsen av bufferen, hvis position/skala/rotation-gradienten settes til nul i optimiseren — de forblir rumlig stive over hele treningen. Density-control må ikke klone, splitte eller prune dem. Bruges til sky-dome-injection (se T45): hvis sky-dome er tændt, settes dette felt automatisk til T46 skyDomeSampleCount. Manuel innstilling er mulig (f.eks. for at fryse en forhånds-placeret punktsky fra en LiDAR-scanning), men ikke direkte tilgjengelig i UI. Viktigt: de første N gaussians i bufferen er alltid de frosne — rækkefølgen i bufferen avgjør, ikke et eksplisitt index.

 **KORT FORKLART**

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

Adam + LR-schedule (T49–T55)

T49 adamResetIteration

DE TALJER

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

TEKNISK

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

KORT FORKLART

Reset-knap for det interne Adam-optimizers „minne«. Har skadet i tests, forblir fra.

T50 positionLRScheduleEndIteration

DE TALJER

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: iterasjon, hvor cosine-annealing- kurven for position-LR når sit minimum. Hvis 0, er det identisk med `T1 maxIterations`. Hvis > 0, kører schedulen til denne verdi og forblir deretter konstant ved `T4`

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

KORT FORKLART

Hvornår appen holder op med yderligere at senke position-læringsraten. Hvis lavere end maksimum-iterasjon, kører deretter med konstant mini-rate — det forfiner meget sakte, men meget stabilt. Indbygget i Quality-pre-set, du trenger ikke dreje.

T51 extendedPhaseLRDecay DETALJER

Default: 0.0 (= deaktivert, 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: etter T50 er nået, og position-LR er allerede ved T4. Hvis 0.1, cosine-decayes skala/rotation/opacity/SH for deres del fra 1.0 (= deres standard-LR) til 0.1× av deres standard. Hvis 0.0 (default), forblir de konstante. V457 testede fuldt decay (0.0 = decay-til-nul) mod intet-decay og fant: avg 0.0400 (2 runs) = samme loss som V438 uten decay. Adfærd renere med decay, men ikke målbart bedre. Derfor disabled. Forbliver i CLI som `--nonpos-lr-scale F`.

 KORT FORKLART

I den sene refinement-fase også gjøre farve- og form-læringsrater mindre. Gjør treningen „stabilere«, men empirisk ikke bedre. Fra.

T52 adaptiveDensifyThreshold DETALJER

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

 TEKNISK

V440-eksperiment: hvis true, beregner appen i hvert densification-skritt p98 av den aktuelle gradient-fordeling og bruker det som dynamisk tærskel (klampt til minst 0.5× av den konfigurerte verdi fra T11, så det ikke skejser for langt ud). Hypotese: automatisk tilpasning til aktuell scene-fase ville gjøre 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 iterasjoner er ekstremt højt, og så overskrider næsten intet tærsklen). Den faste tærskel er allerede godt kalibrert, dynamisk tilpasning skader mere end den gavner. Q5 (T77) tilbyr en alternativ adaptiv logik via rolling median, der omgår problemet.

 KORT FORKLART

Adaptiv version av densify-tærsklen. I tests katastrofal (splat-antallet styrtdykkede til 63K). Fra. Q5 har en bedre variant av det.

T53 mergeAfterDensification **DETALJER**

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

 **TEKNISK**

V438-feature: ved slutningen av densification-fasen (iter `T2`) utføres et engangs-merge-pass, der sammenfatter nærtliggende gaussians med lignende skala og farve. Reduserer gaussian-antallet med typisk 5–15 % uten synlig kvalitetstab. Mening: etter intensiv kloning oppstår klynger av kvasi-identiske gaussians, der ikke bidrager med noe nytt — merging frigjør optimizer-kapasitet til andre områder. Standard i Classic-quality-presets. Bruges ikke ved MCMC, fordi MCMC via sin relocation-logik slet ikke lader sådanne klynger oppstå.

 **KORT FORKLART**

I slutningen av splat-mangfoldiggørelses-fasen sammenfattes kloner, der er næsten identiske. Reduserer datamengde uten synlig effekt. Som standard tændt i Quality- preset.

T54 densifyPhase2FromIteration **DETALJER**

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

 **TEKNISK**

V426-eksperimental: muliggjør en anden densification-fase, der etter refinement-pausen starter ved denne iterasjon og kører til `T55` . Hypotese: etter en refinement-fase har gradient-akkumulatorerne stabilere magnitudes og kan mere præcist si, hvilke områder der stadig har bruk 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 tilgjengelig til eksperimenter.

 **KORT FORKLART**

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

T55 densifyPhase2Untilliteration DETALJER

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

 TEKNISK

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

 KORT FORKLART

Slutning av 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: etter treningslutt fjernes gaussians med sigmoid(opacity) < 0.01 hårdt (de bidrager praktisk talt ikke lenger til bildet). Reduserer gaussian-count med typisk 58 % og eksport-filstørrelse med 55 % uten synlig kvalitetstab. Som standard aktiv i production-presets — slutresultatet skal kunne leveres så kompakt som mulig. I .quickTest fra, fordi en diagnose-kørsel likevel ikke eksporteres. I modsætning til T42 midTrainingCompactificationIterations (V549) finner compactificationen først sted ved slutningen — refinement kan inntil da bruke alle gaussians.

 KORT FORKLART

Opryddning etter træningen: næsten usynlige splats fjernes. Gjør eksport-filen ca. halvt så stor uten 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 interpolasjon i 3D-viewer-output. Hvis treningsoppløsning < viewport-størrelse (f.eks. trening på 0.5x, viewport-visning i full oppløsning), kan MetalFX levere et markant skarpere bilde. Ændrer sig live i viewporten, ingen genoptræning nødvendig. Udelukker T58 `mpsLanczosScaling` — MetalFX har forrang. Anbefaling: tænd, hvis bildet i vieweren virker „udvasket« sammenlignet med den forventede detalje.

 KORT FORKLART

Apple-ML-basert bilde-skarping i 3D-vieweren. Hjælper, hvis du har trent i lavere oppløsning og viser resultatet i full skærm. Live-toggle, prøv.

T58 mpsLanczosScaling DETALJER

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

 TEKNISK

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

 KORT FORKLART

Klassisk skarpmetode til 3D-vieweren (Lanczos). MetalFX (T57) er ML-basert 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 iterasjoner et nyt render i vieweren — godt nok til at iagttage fremskridt uden at sinke træningen. 0 = vieweren opdateres slet ikke (baggrunds- trening, 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 FORKLART

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

T60 perceptualLossWeight DETALJER

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

 TEKNISK

V444-future-feature: vægt av 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 viktig end „ser realistisk ud«. Implementering stadig udestående (kode-stub finnes, men forward-pass ikke implementert). Default 0.0. Forbliver i feltkataloget til fremtidig aktivering; CLI-flag `--percep-weight F` reservert.

 KORT FORKLART

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 mellom 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. Forsigtighet 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 iterasjon, fordi MCMC-OpacityReg på 0.01 dræber Classic-opacities). Derfor er MCMC-init-defaults bevidst „blødgjorte« (alle reg-verdier 0.0).

KORT FORKLART

Hvilken algoritme der bruges til at mangfoldiggøre splats. Classic = oprindelig metode (raskt, mange splats). MCMC = nyere metode (saktere, 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

DE TALJER

Default: 150 000 (initializer + `.fullMCMC` + `.mcmcBalanced`), 100 000 (`.mcmcPreview`), 1 500 000 (`.fullMCMCMip` — Mip-splatting-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 av 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 forblir 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 mellom 670 K (Indoor) og 1.25 M (Outdoor). Ved verdi 0 bruger engine den fulde buffer-kapacitet som cap.

KORT FORKLART

Maksimum antal splats ved MC-MC. 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 minne; ned er snarere en nødbremse.

T63 mcmcNoiseScale

DE TALJER

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

TEKNISK

Multiplikator for det gaussiske støj, der i hver MC-MC-iterasjon adderes til positionen av hver gaussian (SGLD- logik). Højere = mere eksploration (gaussians vandrer mere, finner potentielt bedre pladser), lavere = mere udnyttelse (gaussians forblir, 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 av decay-området er støj reelt 0, og gaussians konvergerer.

KORT FORKLART

Hvor meget tilfeldig „wackeln« appen tillader splats, så de selv finner den bedste plass. Standardværdien er optimalt testet. Hvis du skruer det op, blir splats urolige.

T64 **mcmcOpacityRegWeight** **DETALJER**

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

 **TEKNISK**

MCMC-spesifik L1-penalty på opacity. Paper-default 0.01 (trykker ubrugte gaussians mod nul, gjør dem tilgjengelige for relocation). V464b viste dog: uten reg er det målbart bedre i RadianceKit (session 28 bekreftet). Grund: pruning-kriteriet definert med T68 `mcmcDeadOpacityThreshold` rækker alene — en yderligere L1-penalty tvinger også verdifulde, lav-opacity- gaussians til at dø. Derfor default 0.

Advarsel: i 1.4.3-beta-build var initializer-default'en fejlagtigt 0.01, noe som resulterede i mass-extinction-bug'en (se T61-forklaring); siden 1.4.4 fastsat på 0.0.

 **KORT FORKLART**

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

T65 **mcmcScaleRegWeight** **DETALJER**

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

 **TEKNISK**

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

 **KORT FORKLART**

Som T64, men for splat-størrelse. Fra.

T66 mcmcRelocationInterval DETALJER

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

 TEKNISK

Iterations-interval, hvor MCMC reloacerer døde gaussians (sigmoid(opacity) < T68 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 fant 155 som scene- specifikt optimum for outdoor — de let længere intervaller giver Adam mere tid til at integrere nyplacerede gaussians, før neste reloc-event sætter dem under pres.

 KORT FORKLART

Hver hvor mange iterasjoner MC-MC flytter de døde splats et annet sted hen. 100 er standard. Du trenger ikke dreje selv — Outdoor-preset har allerede den optimale verdi.

T67 mcmcWarmupIterations DETALJER

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

 TEKNISK

Antal initial-iterasjoner, hvor ingen MCMC- relocation endnu sker. Først etter denne warmup begynner reloc- logikken. Mening: i de første iterasjoner er opacity-værdierne endnu ikke afbalanserte — hvis der startedes direkte med reloc, ville gaussians bli placeret de forkerte steder og straks skulle flyttes igjen, noe som ødelegger Adam-momentum. Paper-default 500. RadianceKit overtager denne verdi, fordi V464b viste, at den er robust.

 KORT FORKLART

Hvor mange iterasjoner MCMC først „lander«, før det begynner 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 fant 0.01 som optimal (0.005 marginal, 0.02 dårligere). Højere = mere aggressiv reloc (flere gaussians flyttes), lavere = mere forsiktig. 0.01 svarer ca. til „0.5 % visuel synlighed«. P10 Indoor bruker via Q7-BayesOpt 0.0142 som optimum.

 KORT FORKLART

Fra hvilken gennemsigtighet et splat regnes som „dødt«, så MC-MC flytter det et annet sted hen. 0.01 er optimal i vores tests. Du trenger ikke dreje selv.

T69 `mcmcNoiseDecayEnd` DETALJER

Default: 0 (initializer = „ingen decay«), 160 000 (`.fullMCMC` = 80 % av 200K), 96 000 (`.mcmcBalanced` = 80 % av 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 fant 80 % av maxIterations optimal — giver MCMC nok eksplorations-tid, men lader de siste 20 % gå til konvergens uden støj. 0 = konstant støj over alle iterasjoner (sjældent meningsfuldt, MCMC kan så ikke konvergere).

 KORT FORKLART

Hvornår det tilfældige „wackeln« av splats holder op. I MCMC-presets ved 80 % av samlede iterasjoner — 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 av MCMC-populations-targetet pr. relocation-step. Logikken: ved hver reloc-event øges mål-populations-størrelsen med $(1 + \text{growthRate})$, inntil T62 `mcmcMaxGaussians` (eller den per T72/T73 skalerede variant) er nået. V512/V522 fant 0.05 som optimal — høyere verdier fører til for rask vækst (gaussians indsættes, før Adam-momentum kan integrere dem), lavere til under-densificerede scener til sidst.

 **KORT FORKLART**

Hvor raskt splat-antallet ved MC-MC 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 støy end lavt-opake. $K = 100$ er skarp, sprich overgangen fra „full-noise« til „ingen-noise« sker meget raskt omkring opacity 0.5. V484–V487 fant $K = 100$ optimal — mindre verdier (10–50) lader også højt-opake gaussians wackle med (ødelægger konvergerede gaussians), større (> 500) gjør overgangen kunstigt hård, og døde gaussians flyttes slet ikke mere.

 **KORT FORKLART**

Specialparameter, der bestemmer, hvor skarpt MCMC adskiller mellom „gennemsiktig 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 (= deaktivert) 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-kapasitet). 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 fant scene-spesifikke optima: outdoor profiterer av høy multiplier (5.32 → ~830 K cap ved 156 K bicycle-init), indoor nøjes med 1.76 (vægge mætter raskere). Full oppløsning av cap'et se -metoden.

 KORT FORKLART

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

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, brukes udelukkende `T62 mcmcMaxGaussians` som cap (tilbake til 1.4.4-utførelse). Som standard tændt, fordi mass-extinction-problemerne ved store scener fra 1.4.3 ellers kommer tilbake. Manuelt deaktivere kun, hvis du eksplisitt vil sette et hårdt cap — f.eks. for at trene en 150 K-variant, hvis slutstørrelse kan planlægges.

 KORT FORKLART

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

Mip-splatting (Q1.5) (T74–T76)

Status: Q1.5 ble 2026-05-25 etter 14 autonome iterasjoner

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

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

T74 useMipSplatting

DETLJER

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 -kompensasjon, der begrænser per-gaussian-frekvensen til Nyquist-grænsen av den tætteste træningskamera-sampling-rate. Teoretisk mål: eliminering av aliasing ved rendering i off-training-skalaer ($0.5\times$ eller $2\times$ av træningsopløsningen). Aktiveret i preprocess- og backward- projection-shaderne, funksjonelt korrekt verifisert i Q1.5-D- test. Men: det originale acceptance-gate ($\Delta@1\times \geq +0.3$ dB OG $\text{avg}(\Delta@0.5\times, \Delta@2\times) \geq +1.5$ dB) ble ikke nået på noen av de 11 pair-scenes. Maksimalt observeret: family 750K classic $\Delta@2\times = +0.270$ dB. Outdoor-scener (Truck, Flowers) viste endda forværring $1\times$ og $0.5\times$. Hypotese: 3D-smoothing konkurrerer med MCMC-relocation ved high-Gs. Feltet forblir til fremtidig multi-scale-re-eval med korrekt Mip-NeRF-360-metodologi (se O3-backlog i benchmark-stien).

KORT FORKLART

Aliasing-filter fra et 2024-paper. Teoretisk fedt, praktisk har det intet givet i vores tests og noen gange endda skadet. Forbliver tilgjengelig 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 = skarpe-re, 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 FORKLART

. 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 verdi fuldstændigt og skriver det hardkodede 0.3 — så baselinen ikke kan regredere (Codex-round-1-S3-1-garanti). Hvis, bruges den her satte verdi. Forbliver i feltkataloget til Mip-sweeps.

 KORT FORKLART

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

Adaptive densification (Q5) (T77–T79)

T77 adaptiveDensification

DETLJER

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 `median(seneste N avgGrad-samples) × T79 adaptiveDensifyMultiplier`. $N = T78 \text{ adaptiveWindow}$. Strengere end V440 p98 (den katastrofale 63 K-pruning-fælde), median + 2× sidder omtrent ved p70–p80 av 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 ble 2026-05-25 godkendt som opt-in og aktiveres via CLI `--adaptive-densify`. Q6 er her „carrier« av kvalitetsgevinsten, Q5 bidrager snarere til stabilitet.

KORT FORKLART

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

T78 adaptiveWindow

DETLJER

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

TEKNISK

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

KORT FORKLART

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 av den typiske gradient-fordeling. Lavere = mere aggressiv vækst (flere kloner), høyere = strengere (færre kloner). Q5-tests i range 1.5–3.0 — 2.0 bedste default. Kun relevant hvis.

 KORT FORKLART

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: treningsoppløsning starter ved 0.5× 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 `resolutionWarmupScale`, hvis aktivert. Q6 er godkendt som „carrier av kvalitetsgevinsten« i Q5+Q6-bundtet (se T77) — den trinvis oppløsningsforøgelse giver appen tid til at finne grov geometri på den lavere oppløsning, før den går over til det fine detalje-arbejde. Via CLI: `--curriculum-resolution`.

 KORT FORKLART

„Først grov, så fin« for treningsoppløsningen. Halv oppløsning i første halvdel, så full oppløsning. 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-trappingene jævnt over treningstiden i stedet for at front- load dem. Hypotese: stabil geometri etableres før color-detail- eksplosion, noe som positionerer view-direction-afhængige glans- effekter mere præcist. Q5+Q6 sammen PASS 1/3 scener, Q6 som carrier av gevinsten (Q5 alene FAIL). Via CLI:

`--curriculum-sh .` **KORT FORKLART**

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

Statiske presets (TP1–TP9)

Her kun de strukturelle forskjeller til initializer-default'en. Den fulde marketing-beskrivelse av de elleve UI-presets P1–P11 finner du i kapittel 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 % av maxIter) - 1.6e-6 \rightarrow 1.6e-5 (10 \times høyere, 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 FORKLART**

enhver vilkårlig initial vurdering av en nylig importert bilde-serie — 2–3 min ventetid, deretter rækker resultatet til et binært spørsmål „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 FORKLART**

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

TP3 `.fullClassicPaper` **DETALJER**

Q1.5-A-test-sibling av TP2, paper-tro Classic. Overrides i forhold til TP2: - 35 000 → 30 000 (paper-standard) - 5 000 → 15 000 (paper: 50 % av maxlter) - 1.6e-5 → 1.6e-6 (paper-default) -, tilbake 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 % av maxlter)

 **KORT FORKLART**

Q1.5-forskningseksperimenter, der har brug for paper-magnitude-gaussian-budgetter (1-2 M) til Mip-splatting- tests. Etter Q1.5-„closed no-win«-verdict forblir presetet tilgjengelig for advanced users, men er ikke production-anbefalt.

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 % av 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 % av 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 FORKLART**

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

TP5 `.fullMCMCMip` **DETALJER**

Q1.5-D-test-sibling av 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 FORKLART**

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 likevel closed-no-win — Mip-splatting-gevinst ikke reproducerbar over 11 pair-scenes. Preset forblir opt-in.

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 FORKLART**

Standardtilfælde med kortere ventetid. V149 identificeret som sweet spot.

TP7 `.mcmcPreview`

DETLJER

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 FORKLART

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

TP8 `.mcmcBalanced`

DETLJER

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 (mellom Preview 2.0 og Full 3.0)

KORT FORKLART

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

TP9 `.quickTest`

DETLJER

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

KORT FORKLART

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

Metode:

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

TEKNISK Eneste source-of-truth for spørsmålet „hvor mange gaussians må MC-MC 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 innstilling, 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 verdi til en lille og en gigantisk scene. Du trenger ikke selv kalde metoden — træningen bruger den internt.

Hvilket felt til hva? (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> $\leq 14\%$ av T1 |
| Reducere floaters i droneflyvninger | <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

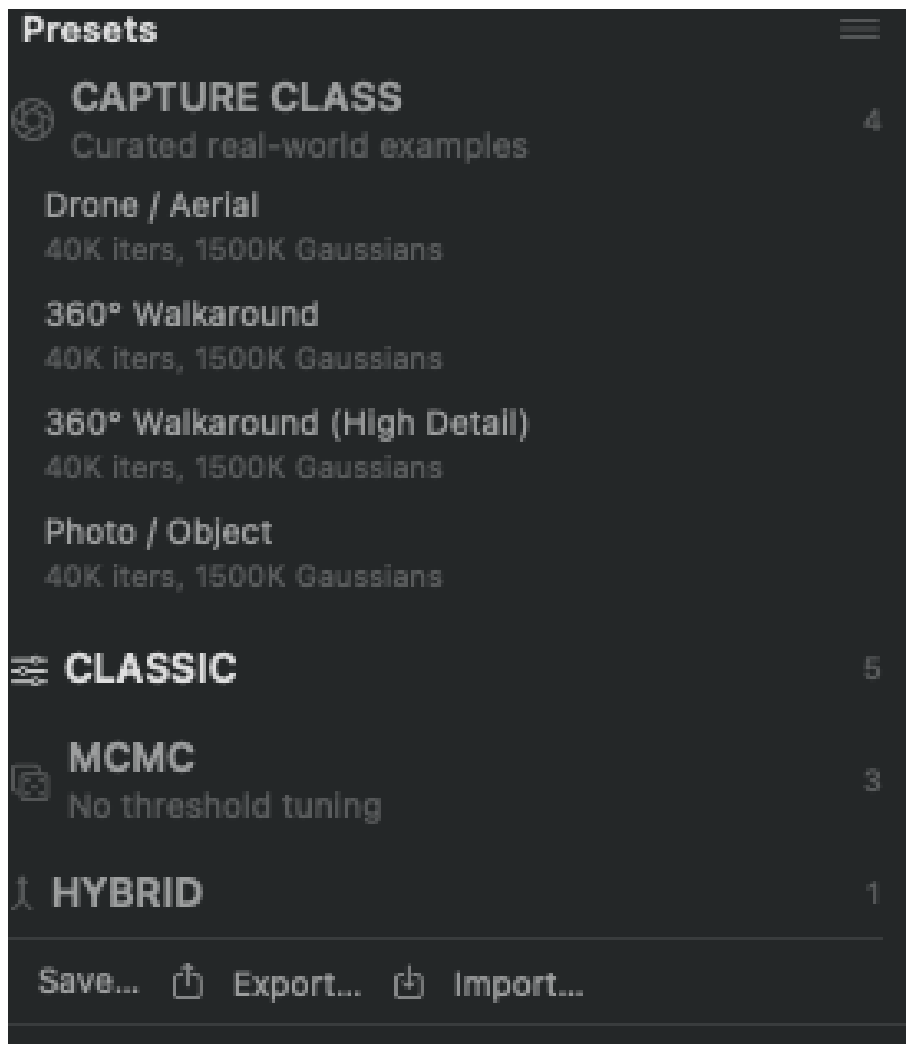
Disse felter kan ved feil-konfigurering føre til OOM, app-crash, mass-extinction av gaussians eller ubrugelige benchmark-data. Skal håndteres med forsigtighet:

- T11 `densifyGradThreshold` — en halvering kan generere 2–4× så mange gaussians, noe som raskt sprænger GPU-lageret. Merk 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 oppstår et resolved-cap på millioner av gaussians. På 36-GB-RAM-Mac'er OOM mulig. Outdoor-preset 5.32 fungerer kun, fordi Mip-NeRF-360-Bicycle har 156 K init-punkter → 830 K cap.
- T39 `testViewIndices` — manuel innstilling kan gjøre benchmarket ubrugelig (alle indekser > N → ingen holdouts). Lad `--benchmark -flaget` sette det.
- T64 `mcmcOpacityRegWeight` **og** T65 `mcmcScaleRegWeight` — i 1.4.3-beta sat til 0.01, noe som førte til mass-extinction (460 K → 5 gaussians på en iterasjon). Siden 1.4.4 fastsat på 0.0, men manuel forhøyelse kan reproducere problemet.
- T15 `opacityResetInterval` — hvis ikke 100 000+ (reelt fra) og treningen er kortere end 10 000 iterasjoner, ødelegger reset konvergensen. `.preview` har det derfor på 100 000 trods `maxIterations = 5 000`.
- T54/T55 `densifyPhase2*` — two-phase densification er i tests avsluttet i 0-gaussians-cascade. Lad begge stå på 0.
- T74 `useMipSplatting` — Q1.5 closed-no-win 2026-05-25, kan endda forverre PSNR på noen 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 av 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.

KAPITTEL

Kapittel 7 — Innebygde kvalitetsforhåndsinnstillinger



Figur 27: Forhåndsinnstillingsseksjon med alle fire grupper foldet ut — 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, merknad „No threshold tuning«), HYBRID (1 preset: Balanced (Hybrid))

HVA BILDET VISER Presets-seksjonen i Inspector, alle fire grupper foldet ut. CAPTURE CLASS med de fire kuraterede virkelighetspresetene (Drone / Aerial, 360° Walkaround, 360° Walkaround (High Detail), Photo / Object) — det er den primære gruppen og i Simple Mode den eneste synlige. CLASSIC med Quick (1K iters), Preview (5K iters,

aktivt valg med blå hake), Balanced (20K iters), Quality (35K iters) og Ultra Detail (35K iters). MCMC med undertittel „No threshold tuning« — MCMC trenger ikke densifisering-til-terstet: 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 forhåndsinnstilling er en forberedt konfigurasjon for treningen. RadianceKit leveres med tretten innebygde forhåndsinnstillinger i fire grupper: fire **Capture-Class**-presets (P9–P12) — kuraterede oppskrifter validert med øyet på ekte community-materiale for reelle opptakstyper (drone, 360°-rundgang, foto-objekt), og den primære aksens siden v1.6 —, fem Classic-presets (P1–P5: Quick/Preview/Balanced/Quality/Ultra Detail), tre MCMC-presets (P6–P8) og én Hybrid-preset (P13) som kombinerer Classic- og MCMC-strategiene. De tidligere „Scene-Class«-presetene (Render/3D, Outdoor, Indoor, i fase Q7 akademisk tunet mot Mip-NeRF-360- og NeRF-Blender-scener) ble trukket tilbake som synlig gruppe i v1.6 — Capture-Class, validert med øyet på ekte materiale, er nå den primære aksens; de Q7-tunede konfigurasjonene bevares bare internt videre. Du velger presetene i sidefeltet under **Presets** eller i Simple Mode ved import. **+** -knappene åpner dialoger for å opprette egne presets ved siden av — de tretten innebygde kan ikke slettes, men kan gjerne dupliseres.

I Expert-View vises presetene gruppert etter opptakstype og strategi (Capture Class / Classic / MCMC / Hybrid). Et klikk på en oppføring skriver den lagrede treningskonfigurasjonen inn i den aktuelle tilstanden. Det er ikke et øyeblikksbilde — hvis du deretter dreier på glidebrytere, endres tilstanden, men selve presetet forblir uendret; en farget merknad viser da „modified«.

Hvilken preset som er den rette når, avhenger mest av scenetypen og maskinvaren. De tre tabelloversiktene mot slutten av kapittelet oppsummerer dette.

| P1 — Quick



Inspector → Presets-seksjon → gruppe „Classic« → oppføring „Quick«. UUID-suffiks ...001 .



Diagnose-preset med 1 000 iterasjoner, klassisk (adaptiv) densification-strategi og en treningsoppløsningsskalering på 0.25× (input-bildet forminskes til 25 % før treningen). Skal ikke brukes til levering av en scene, men til raskt å finne ut om oppsettet (kameraposisjoner, punktsky, bildeserie) i det hele tatt viser meningsfull bevegelse i loss-verdiene. På en M3 Ultra typisk under 30 sekunder på 50–200 bilder. Den lille oppløsningen tilslører den ekte bildekvaliteten, men holder minneforbruket og render-arbeidet svært lavt. Velges også automatisk som default ved første oppstart hvis systemet har mindre enn 10 GB RAM.

KORT FORKLART

Rask funksjonstest. Bilder inn, vent et drøyt halvt minutt, kikk om scenens grove omriss dukker opp. Hvis bildet i vieweren ligner en utvasket klatt — så er det fint, det skal det. Hvis du derimot bare ser mørke punkter eller en fullstendig forvrengt form, er kameraposisjonene nok feil (se kapittel 9). For et resultat som kan vises, trenger du deretter minst P2 eller P3.

| P2 — Preview (Classic)



Inspector → Presets-seksjon → gruppe „Classic« → oppføring „Preview«. UUID-suffiks ...002 .



5 000 iterasjoner Classic-densification, 0.5× oppløsningsskalering, doble læringsrater i forhold til standard. Densification (klon + split) er aktiv over de første 2 500 iterasjonene, deretter bare pruning. Default-preset for systemer med ≥ 10 GB RAM. På en M3 Ultra typisk 90 sekunder til 3 minutter på en 200-bilders scene. Gir et brukbart inntrykk av geometrien og kamera-posen, men teksturer er tydelig mykt tegnet — 0.5× render-oppløsningen lar seg ikke uten videre omgå ved å trene på nytt med P3 eller P4, fordi læringsratene er kalibrert til den halve oppløsningen.

KORT FORKLART

Standarden for „bare ta en kikk«. Hvis du nettopp har importert nye bilder og vil se om scenen i det hele tatt kan rekonstrueres, er det det rette trinnet. Cirka 2–3 minutters venting, deretter kan du dreie i 3D-vieweren og vurdere om det er verdt å investere i flere treningsrunder. Først når forhåndsresultatet allerede ser bra ut, lønner Balanced eller Quality seg.

| P3 — Balanced (Classic)



Inspector → Presets-seksjon → gruppe „Classic« → oppføring „Balanced«. UUID-suffiks `...005` .



20 000 iterasjoner Classic-densification ved full bildeoppløsning. Densification kjører over de første 15 000 iterasjonene, fra iter 3 000 med et densify-intervall på 100. Empirisk det „sweet spot« fra de dokumenterte treningssesjonene: ved klassisk densification på Horse Full og Truck stabiliseres L1-loss mellom iter 18 000 og 22 000, en lengre trening gir ikke lenger signifikant forbedring under Quality (P4). På en M3 Ultra typisk 30–60 sekunder på 200 bilder, 5–8 minutter på 1 000+ bilder.

KORT FORKLART

Det „gode kompromisset«. De fleste scener ser allerede gode ut her, uten at du må vente en time. Hvis du vil vise sluttresultatet et sted (sosiale medier, nettside, en kundedemo), holder det ofte. Først når du vil zoome inn i splatmodellen eller trenger detaljer i overflateteksturen, lønner det seg å hoppe opp til P4 Quality eller P7 MCMC.

| P4 — Quality (Classic)



Inspector → Presets-seksjon → gruppe „Classic« → oppføring „Quality«. UUID-suffiks `...003` .



35 000 iterasjoner Classic-densification med V546-„Opacity Decay« (HTGS, Eurographics 2025): etter hver densify-syklus multipliseres opacity av alle eksisterende gaussians med en faktor < 1.0 , noe som pålitelig fjerner gaussians som har blitt inaktive, ved pruning og dermed oppnår 14 % bedre L1-tap enn den klassiske 35 000-kjøringen ved samme iter-antall. SSIM-tap er aktivert (`ssimWeight=0.05`). På en M3 Ultra typisk 2–4 minutter på 200 bilder. Leverer på NeRF-Blender (Lego, Chair, Drums) endelig L1 ≈ 0.023 — beste Classic-variant i de 560+ dokumenterte eksperimentene. Merk: krever ~3–5 GB GPU-minne; på 8-GB-systemer er P3 det trygge valget.

KORT FORKLART

Den beste klassiske varianten. Leverer skarp tekstur og fin geometri, særlig på objekt-opp-tak (en skulptur, en stol, en vase). Ved store outdoor-scener eller rom merker du derimot nesten ingen forskjell fra Balanced — der er det mer verdt å bytte til en MCMC-preset (P6–P8) eller en Capture-Class-preset (P9–P12) enn å hoppe fra P3 til P4. Vil du ha det absolutte maksimum i Classic-familien, tar du P5 Ultra Detail.

I P5 — Ultra Detail (Classic)



HVOR

Inspector → Presets-seksjon → gruppe „Classic« → oppføring „Ultra Detail«. UUID-suffiks `...008`.



TEKNISK

Rundt 35 000 iterasjoner Classic-densification — vinneren av held-out-kjøringen i kvalitetsmatrisen (2026-06-10). På alle tre testede Mip-NeRF-360-scener slår Ultra Detail det innebygde MCMC-„Quality«-presetet (P8) ved sammenlignbar wall-clock-tid med rundt +0.94 dB PSNR. Dermed er det det sterkeste Quality-presetet i Classic-gruppen og den skarpeste Classic-varianten RadianceKit leverer. På en M3 Ultra typisk i samme tidsramme som P4 Quality (2–5 minutter på 200 bilder), men krever litt mer GPU-minne; på 8-GB-systemer forblir P3 det trygge valget.

KORT FORKLART

Det skarpeste Classic-trinnet og held-out-vinneren i våre kvalitetstester: på ekte scener rundt en desibel bedre enn MCMC-„Quality«-varianten — ved omtrent samme ventetid. Vil du ha maksimal detaljtrofasthet med den bevarte klassiske densification og har nok GPU-minne, er dette førstevalget. Rekker ikke minnet, eller trenger du en så liten eksportfil som mulig, så bli ved P4 Quality eller en MCMC-preset.

I P6 — Preview (MCMC)



HVOR

Inspector → Presets-seksjon → gruppe „MCMC« → oppføring „Preview«. UUID-suffiks `...006`.



TEKNISK

60 000 iterasjoner MCMC-densification (3DGS-MCMC, NeurIPS 2024) ved et tak på 100 000 gaussians. MCMC erstatter den heuristiske klon/split-logikken med Markov-Chain-Monte-Carlo-relocation: døde gaussians plasseres på nytt via sigmoid-vektede sampling-dybder, noe som gir et kontrollert og reproduserbart antall gaussians. Taket setter et hardt lokk på maksimalt antall ved 100K — det sparer minne og render-tid, men koster detalje. På en M3 Ultra typisk 5–8 minutter på 200 bilder. Egnert som „MCMC-funksjonstest« — hjelper med å vurdere om et bytte fra Classic til MCMC ville være meningsfullt, før du investerer mer tid i P7 eller P8.

KORT FORKLART

Som P2 Preview, men med den nyere MCMC-metoden. Leverer ofte litt mer kompakte, jevnere fordelte splat-skyer enn Classic-varianten. For en første vurdering av en scene strekker de 5–8 minuttene som regel til. Hvis du liker forhåndsresultatet, er neste skritt P7 (Balanced) eller direkte P8 (Quality MCMC).

| P7 — Balanced (MCMC)



HVOR

Inspector → Presets-seksjon → gruppe „MCMC« → oppføring „Balanced«. UUID-suffiks ...007 .



TEKNISK

120 000 iterasjoner MCMC ved et tak på 150 000 gaussians. Det midterste MCMC-trinnet — nesten det endelige gaussian-antallet fra P8 Quality, men bare 60 % av iterasjonene. Empirisk ligger L1-tap i de dokumenterte treningssesjonene på 0.026–0.028 på Horse Full, mot P8 med 0.0246 — altså rundt 7 % høyere, til gjengjeld halve ventetiden. På en M3 Ultra typisk 8–15 minutter på 200 bilder. Bruker en metode som skalerer det effektive gaussian-taket til punkttettheten i input-SfM-punktskyen (se T75 i kapittel 6).

KORT FORKLART

MCMC med skikkelig detaljrikdom, men uten den lange fullkjøringen fra P8. For de fleste scener strekker det til, særlig hvis du vil presse en MCMC-kjøring inn i lunsjpausens tidsramme. Hvis minnet er knapt (f.eks. på M-processorer med bare 16 GB), så bli her — P8 krever mer GPU-minne.

| P8 — Quality (MCMC)



HVOR

Inspector → Presets-seksjon → gruppe „MCMC« → oppføring „Quality«. UUID-suffiks ...004 .

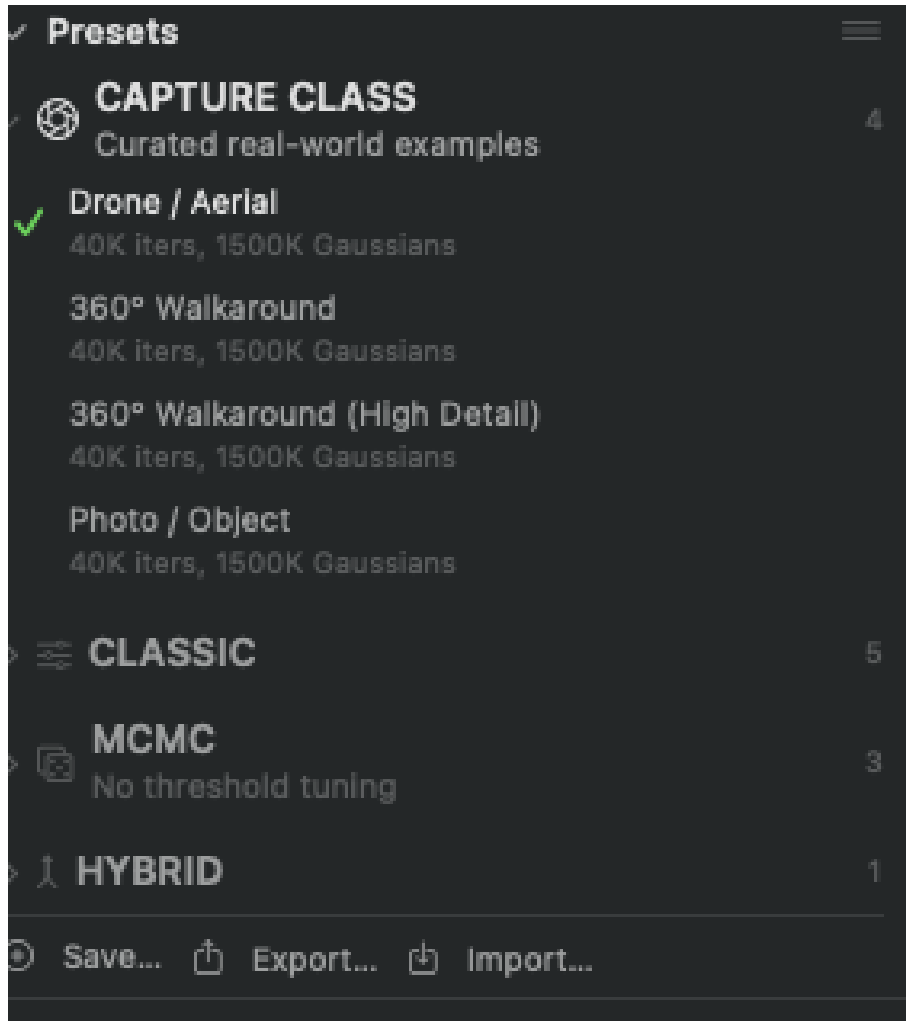


TEKNISK

200 000 iterasjoner MCMC ved et tak på 150 000 gaussians, SSIM-tap 0.05, MCMC-noise-decay over 80 % av iterasjonene. Best-single-run-L1 i de 560+ eksperimentene: 0.0238 på Horse Full, gjennomsnitt over 3 trials 0.0246 (mot P4 0.0230 på samme scene). MCMC leverer samtidig 71 % færre gaussians (150K vs ~524K) — avgjørende hvis du vil levere resultatet på nettet, fordi den mindre skyen gir markant mindre eksportfiler. Treningstid på en M3 Ultra typisk 20–35 minutter på 200 bilder; på 1 000+ bildersett nærmere 1–2 timer. Beste valg hvis maksimal bilde kvalitet ved minimal sluttstørrelse ønskes.

KORT FORKLART

Den beste MCMC-varianten. Leverer veldig rene, kompakte splat-skyer — ideelt når du senere vil bygge inn resultatet som web-3D-viewer eller sende det som fil (filen er mindre enn ved P4 Quality med sammenlignbar optisk kvalitet). Det krever imidlertid tålmodighet — på store opptak over en times venting. Planlegg det heller som en „nattkjøring«.



Figur 28: CAPTURE CLASS-gruppen foldet ut med alle fire kuraterte virkelighetspresets — 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 gruppen står helt øverst og er i Simple Mode den eneste synlige.

HVA BILDET VISER Inspector med CAPTURE CLASS-gruppen foldet ut — den primære preset-gruppen siden v1.6, i Simple Mode den eneste viste. Hver oppføring er en oppskrift validert med øyet på ekte community-materiale for en konkret opptakstype (drone, 360°-rundgang, foto-objekt), ikke en verdi optimert mot et akademisk testsett. Valg ved klikk skriver den lagrede treningskonfigurasjonen inn i den aktuelle tilstanden.

| P9 — Drone / Aerial



Inspector → Presets-seksjon → gruppe „Capture Class« → oppføring „Drone / Aerial«. UUID-suffiks `...010`.



Capture-Class-preset for luft- og drone-orbits av bygninger og landskap. MCMC-densifier, 40 000 iterasjoner, tak 1,5 mill. gaussians, SSIM-tap 0.5 pluss Edge-Aware-ledd 0.1. Avgjørende er anisotropi-straffen med vekt 0.003 ved en ratio-terskel på 6 — „spaghetti-killeren« mot de typisk nålformede artefaktene som drone-footage produserer. Validert på et ekte DJI-4K-droneflyging over Pensford-viadukten (sjekket med øyet, ikke bare metrisk).

KORT FORKLART

For opptak fra luften — droneflyginger rundt en bygning, over et landskap, langs en fasade. Den kraftige anisotropi-straffen rydder bort de nål- eller spaghettiformede artefaktene som drone-materiale gjerne produserer. Er materialet ditt tatt opp fra bakken, passer heller Photo / Object eller en Classic-preset.

| P10 — 360° Walkaround



Inspector → Presets-seksjon → gruppe „Capture Class« → oppføring „360° Walkaround«. UUID-suffiks `...011`.



Capture-Class-preset for 360°-walkaround-videoer. MCMC-densifier, 40 000 iterasjoner, tak 1,5 mill. gaussians, SSIM-tap 0.5 pluss Edge-Aware-ledd 0.1, mild anisotropi-straff (vekt 0.001 ved ratio-terskel 15). Person- og himmel-maske er aktive. Presetet forventer en 360°-equirect-video som internt reprojiseres til rundt 90° brede perspektiv-crops før treningen starter. Validert på 8K-360°-rundganger med selfie-stang (Monument-scene, sjekket med øyet).

KORT FORKLART

For 360°-rundgang-videoer — du går med et 360°-kamera eller en selfie-stang gjennom et rom eller rundt et objekt. RadianceKit deler selv kule-panoramaet opp i normale blikkvinkler og maskerer bort forbipasserende og himmel. For maksimal skarphet på samme materiale prøv i tillegg High-Detail-varianten (P11).

| P11 — 360° Walkaround (High Detail)



Inspector → Presets-seksjon → gruppe „Capture Class« → oppføring „360° Walkaround (High Detail)«. UUID-suffiks ...013 (Opt-in).



Opt-in-Capture-Class-preset for 360°-walkaround-videoer med maksimal detalje. Hybrid-densifier (klassisk abs-gradient-klon/split

1. MCMC-noise + relocation), 40 000 iterasjoner, tak 1,5 mill. gaussians,

anisotropi-straff 0.0015 ved ratio-terskel 15, SSIM-tap 0.2 og Edge-Aware-ledd 0 — den lockede „r50“-screen-split-oppskriften. På 360°-materiale slår det standard-MCMC-presetet „360° Walkaround« (P10) ved PSNR, LPIPS og synlig konfetti, og det med rundt en tredjedel av splat-antallet. Står bevisst opt-in ved siden av standard-360-presetet, til det er validert på flere scener.

KORT FORKLART

Det skarpere alternativet til standard-360-presetet (P10): mer detalje, mindre konfetti, klart mindre fil. Står bevisst ved siden av i stedet for å erstatte det — så langt bekreftet på en håndfull scener. Er 360°-rundgangen din rent tatt opp, prøv dette presetet først og sammenlign resultatet med P10.

| P12 — Photo / Object



Inspector → Presets-seksjon → gruppe „Capture Class« → oppføring „Photo / Object«. UUID-suffiks ...012 .



Capture-Class-preset for objekt-orbits fra skarpe enkeltfotos (ingen video). Hybrid-t1-densifier (med relocation), 40 000 iterasjoner, tak 1,5 mill. gaussians, SSIM-tap 0.5 pluss Edge-Aware-ledd 0.1, mild anisotropi-straff (vekt 0.001 ved ratio-terskel 15), Opacity-decay 0.9995 hver 50. iterasjon, **ingen** masking. Validert på 163 høyoppløselige 41-MP-fotos av et skjelett (sjekket med øyet). Få views (opptil rundt 600) holder seg da under Hybrid-collapse-terskelen.

KORT FORKLART

For objekt-opptak fra skarpe enkeltfotos — du omkranser en skulptur, en modell, et produkt med kameraet og tar fotos i stedet for video. Ingen masking, fordi skarpe fotos som regel har ren bakgrunn. For video-kilder tar du i stedet en 360°- eller Drone-preset.

| P13 — Balansert (Hybrid)

HVOR

Inspector → Presets-seksjon → gruppe „Hybrid« → oppføring „Balansert (Hybrid)«. UUID-suffiks ...009 .

TEKNISK

20 000 iterasjoner med Hybrid-densifiseringsstrategien: klassisk gradientdrevet clone/split plasserer kapasitet der tapet trenger den, MCMC-SGLD-støy fortsetter å utforske, og døde gaussians flyttes i stedet for å gå tapt i pruning. Opacity-decay (V546) erstatter opacity-resets; en anisotropi-straff (vekt 0.001, ratio-terskel 15) holder nålformede splats i sjakk. Gaussian-cap-en skalerer med scenen (150K basis, scene-aware x3.0). Validert på fem scener mot ren MCMC ved samme budsjett: i snitt +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 bilder.

KORT FORKLART

Et sterkt førstevalg for et endelig resultat: skarpere detaljer enn MCMC-presetene ved en omtrent like kompakt fil, på en brøkdel av P8-treningstiden. Har du bare tid til én kvalitetskjøring og ingen av Capture-klassene passer klart, start her. Classic-presetene forblir bedre for raske tester, og Capture-Class-presetene (P9–P12) er førstevalget når scenen din klart passer en av disse opp-takstypene.

Når hvilken preset?

| Scenario | Førstetest | Hovedkjøring |
|--|-----------------|--|
| Funksjonstest av nye bilder, < 30s | P1 Quick | — |
| Objekt-orbit fra skarpe enkeltfotos | P2 Preview | P12 Photo / Object |
| Enkeltobjekt-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, landskap | P6 Preview MCMC | P9 Drone / Aerial |
| Web-levering (liten, kompakt) | P2 | P8 Quality MCMC (minste fil ved full kvalitet) |
| Skarpe detaljer på kort tid, kompakt eksport | P2 eller P6 | P13 Balansert (Hybrid) |
| Maksimal detaljtrofasthet, Classic-strategi | P3 eller P6 | P5 Ultra Detail |
| Trykk, markedsføring, full detalje | P3 eller P6 | P4 Quality (Classic) eller P5 Ultra Detail |

Rask sammenligning

| Pre-set | Strategi | Iters | Maks-Gs | Render-skala | Typisk tid (200 bilder, M3 UI-tra) | Q-sweep |
|-----------------------------------|----------|---------|----------|--------------|------------------------------------|-------------------------|
| P1 Quick | Classic | 1 000 | ∞ | 0.25x | ~30 s | — |
| P2 Preview | Classic | 5 000 | ∞ | 0.5x | 2–3 min | — |
| P3 Balanced | Classic | 20 000 | ∞ | 1.0x | 30–60 s | — |
| P4 Quality | Classic | 35 000 | ∞ | 1.0x | 2–4 min | V546 HTGS |
| P5 UI-tra Detail | Classic | ~35 000 | ∞ | 1.0x | 2–5 min | Matrix $\Delta+0.94$ dB |
| P6 Preview MC-MC | MCMC | 60 000 | 100 K | 1.0x | 5–8 min | — |
| P7 Balanced MC-MC | MCMC | 120 000 | 150 K | 1.0x | 8–15 min | — |
| P8 Quality MC-MC | 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 | Øye / Via-dukt |
| P10 360° Walkaround | MCMC | 40 000 | 1.5 M | 1.0x | 10–25 min | Øye / Monument |
| P11 360° Walkaround (High Detail) | Hybrid | 40 000 | 1.5 M | 1.0x | 10–25 min | Øye (opt-in) |
| P12 Photo / Object | Hybrid | 40 000 | 1.5 M | 1.0x | 10–25 min | Øye / Skjelett |
| P13 Balanced (Hybrid) | Hybrid | 20 000 | 150 K | 1.0x | 5–10 min | Matrix $\Delta+0.45$ dB |

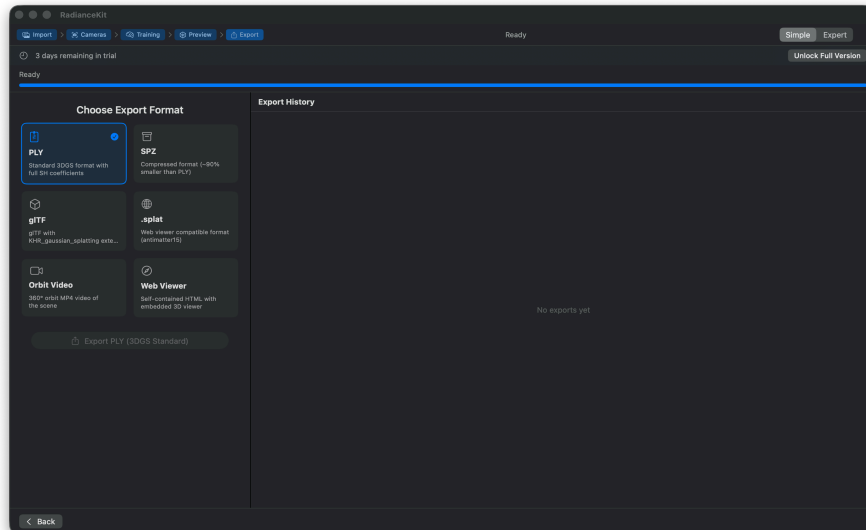
Egne forhåndsinnstillinger

Via knappen **Save...** i forhåndsinnstillingsseksjonen (I1 i kapittel 2) lagrer du den aktuelle treningskonfigurasjonen som en egen preset. Egne presets er ikke „Built-in« og kan gis nye navn, eksporteres (som JSON), deles via dra-og-slipp, dupliseres og slettes. De tretten innebygde presetene P1–P13 forblir upåvirket av slettknappen.

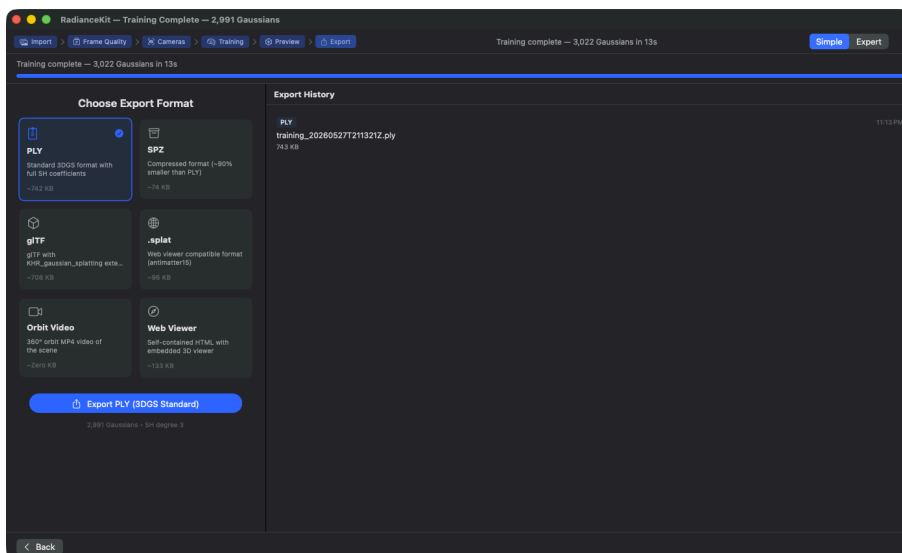
Tommelfingerregel: Hvis du endrer noe ved en preset som du vil bruke oftere — skydome på, høyere SSIM-vekt for en bestemt sceneklasse, avvikende iter-antall — så lagre varianten som en egen preset. Da vet du straks ved neste kjøring at det er en konfigurasjon som avviker fra standarden.

KAPITTEL

Kapittel 8 — Eksportformater



Figur 29: Eksportformatvalg i Simple Mode — seks formatkort



Figur 30: Eksportformat-grid live etter 5K-iter-trening på blomsterbukett — 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 høyre med allerede lagret PLY

Hva bildet viser (2 991 gaussians, SH degree 3, en syntetisk Blender-bukett som IP-clean testsett): Størrelsesangivelsene under hvert formatkort beregnes live ut fra aktuelt gaussian-antall og format-overhead — ikke hardkodet. Av 2 991 gaussians (SH

degree 3) oppstår 742 KB PLY, 74 KB SPZ (faktor ~10× mindre via kvantisering), 708 KB glTF (med KHR_gaussian_splating-extension, derfor nesten PLY-ekvivalent), 96 KB .splat (komprimert 24-byte-per-gaussian-format). Orbit Video viser „~Zero KB«, fordi størrelsen først er kjent etter MP4-encoding. Web Viewer (133 KB) samler en selvstendig HTML-fil med innebygd WebGL-viewer og komprimerte splat-data — større enn ren .splat på grunn av viewer-overhead. Export History til høyre lister allerede avsluttet PLY-eksport („training_20260527T211321Z.ply, 743 KB, 23:13») med format-pille og reveal-i-Finder-action.

En avsluttet trening leverer en gaussian-cloud — en samling på noen hundretusen til millioner av 3D-gauss-fordelinger, som til sammen rekonstruerer scenen. RadianceKit kjenner ti måter å skrive denne skyen til disk på. Seks av dem er rene 3D-dataformater (PLY, Compressed PLY, SPZ, SOG, glTF, .splat), én bundler skyen sammen med en ferdig HTML-viewer (Web Viewer), én renderer en MP4-fil ut av en orbit-kamerakjøring (Orbit Video), og to eksporterer ingen gaussian-innhold, men kun SfM-resultatet (kameraposisjoner og grov punktsky) for gjenbruk i andre trenings-pipelines (transforms.json + COLMAP-workspace).

Hvilket format som er det rette når, avhenger av målet. For arkivering av de fulle dataene uten kvalitetstap tar man PLY. For web-viewer på egen side holder som regel .splat eller den innebygde web-vieweren. Hvis filen skal være minimal, lønner SPZ eller SOG seg. For gjenbruk av SfM-resultatet i Nerfstudio, Postshot eller Brush er transforms.json og COLMAP-workspace de rette veiene.

Alle eksportfunksjoner ligger i menyen „Export« og i Simple Mode på det siste wizard-trinnet. De fleste formatene er fullt sandbox-konforme og virker i App Store-versjonen. Bare SOG krever en ekstern binary (`cwebp`), som ikke nødvendigvis er til stede i App Store-buildet — detaljer se E4.

| E1 — PLY (.ply)

HVOR

Menylinjen → Export → 3D Formats → Export PLY... (⌘E). Simple Mode: wizard-trinn Export → format-kort „PLY«. **Størrelse:** typisk 100 % (referanseverdi). **Kompatibel med:** SuperSplat, PolyCam, alle 3DGS-viewere.

TEKNISK

PLY er det kanoniske lagringsformatet for 3D Gaussian Splatting. RadianceKit skriver en binær Little-Endian-fil med det standardiserte 3DGS-property-oppsettet: per gaussian trekomponents posisjon, tre normaler alltid satt til null, tre DC-SH-koeffisienter (`f_dc_0..2`) for basis-RGB-fargen, deretter opp til 45 ytterligere SH-koeffisienter (`f_rest_0..44`) i den i Kerbl-2023-paperet definerte transponerte channel-major-rekkefølgen (først alle R-kanal-koeffisienter, så alle G, så alle B), etterfulgt av logit-opasitet (rå pre-sigmoid-verdier), tre log-space-skaler og en wxyz-quaternion-rotasjon. Den maksimalt eksporterte SH-graden klampes til minimum av brukerønske og faktisk lært grad; standard er 3 (45 rest-koeffisienter). Før skriving beregnes payload-størrelsen i 64-bit integer for å fange overflow ved ekstremt store skyer. Filen skrives atomisk, noe som ved store skyer kortvarig opptar dobbelt diskplass.

KORT FORKLART

Det er „originalfilen«. Største fil, høyeste kompatibilitet, ingen tap. Hvis du ikke vet hvilket format du skal velge, så ta PLY — det åpnes i nesten alle 3DGS-verktøy. Med 1 million gaussians blir det avhengig av SH-grad mellom 200 og 800 MB. Hvis filen blir for stor, så se på E2 (komprimert PLY) eller E3 (SPZ).

I E2 — Compressed PLY (.ply)

HVOR

Menylinjen → Export → 3D Formats → Export Compressed PLY... Simple Mode: formatkort „Compressed PLY«. **Størrelse:** ca. 10–20 % i forhold til PLY (5- til 10-ganger komprimering). **Kompatibel med:** SuperSplat, PlayCanvas-engine, webbaserte viewere.

TEKNISK

PlayCanvas-varianten av PLY-formatet med chunked kvantisering. Gaussians grupperes i 256-stykkers chunks. Per chunk legges min/max-bounds for posisjon, skala og farge separat i headeren; de enkelte gaussians refererer til verdiene sine relativt til disse boundsene og komprimeres til hver 32 bit: posisjon og skala med 11-10-11-bit-pakking, rotasjon som 2-10-10-10-bit „smallest-three«- quaternion, farge som 8-8-8-8-RGBA. Høyere SH-koeffisienter kvantiseres med bare 8 bit per komponent (`shCoeffCount * 3` uchar per gaussian). Selve formatet er fortsatt ASCII-header-PLY og derfor i prinsippet validerbart med PLY-verktøy, men vertex-properties er erklært som `uint`-felter. SH-grad er som standard 0 (ingen rest-koeffisienter) for å maksimere komprimeringen — høyere SH-grader kan velges eksplisitt.

KORT FORKLART

Den plassbesparende PLY-varianten. Identisk engine-kompatibilitet som vanlig PLY, men 5 til 10 ganger mindre. SuperSplat og PlayCanvas leser det nativt. For web-deployment nesten alltid bedre enn vanlig PLY. Kvalitetstapet via kvantisering er som regel ikke visuelt merkbart, så lenge scenen ikke inneholder ekstremt høyfrekvente detaljer.

| E3 — SPZ (.spz)

HVOR

Menylinjen → Export → 3D Formats → Export SPZ...
Simple Mode: formatkort „SPZ«. **Størrelse:** ca. 10 % i forhold til PLY (90 % mindre). **Kompatibel med:** Niantic Scaniverse, Niantic Spatial Fields, MetalS-platter.

TEKNISK

Niantics SPZ-v2-format. Posisjoner pakkes som 24-bit-fixed-point (det gir ca. 0,25 mm oppløsning), skaler som 8-bit-kvantisering i log-rom, rotasjoner som 8-bit-smallest-three (i v2 lagres bare xyz, w avledes i dekoderen av quaternion-normen), opasiteter som sigmoiderte 8-bit-verdier. DC-SH lagres med en SPZ-spesifikk pakke-formel ($dc_raw * 0.15 * 255 + 0.5 * 255$), høyere SH-bånd med 5 bit (bånd 1) henholdsvis 4 bit (bånd 2-3) per koeffisient. Hele den pakkebinære-bloben komprimeres deretter med standard-gzip (RFC 1952), noe som gir et gzipped-container-format med magic bytes `1f 8b`. RadianceKit kaller her `system-gzip`, fordi Apples innebygde `zlib-API` produserer proprietær Apple-framing som ikke ville være kompatibel med SPZ-leserne i Spatial Fields eller MetalS-platter. `system-gzip` kan fortsatt spawnes innenfor macOS-sandboxen.

KORT FORKLART

Den minste standardfilen. Hvis du kjenner Niantics Scaniverse — det er det formatet appen bruker. Veldig liten, veldig innlastingsvennlig for mobile apper. Direkte brukbar i Niantics egen sky-viewer (Spatial Fields). Cirka 90 % mindre enn en PLY med samme data, samtidig nesten ikke optisk skjelnbar for de fleste scener.

I E4 — SOG (.sog)

HVOR

Menylinjen → Export → 3D Formats → Export SOG...
Simple Mode: formatkort „SOG«. **Størrelse:** ca. 5–6 % i forhold til PLY (15- til 20-ganger komprimering — det minste alternativet). **Kompatibel med:** PlayCanvas-engine, SuperSplat-editor.

TEKNISK

„Spatially Ordered Gaussians« — et PlayCanvas-format som lagrer skyen GPU-ready i flere lossless-WebP-bilder. Først sorteres alle gaussians romlig via 3D-Morton-code (30-bit Z-order, 10 bit per akse), noe som gir bildene senere cache-lokalitet i rendereren. Så kvantiseres posisjoner med symmetrisk log-transformasjon (for bedre dynamikkomfang) til 16-bit-verdier og splittes i to RGBA-bilder (`means_l.webp` for de nederste 8 bit, `means_u.webp` for de øverste). Rotasjoner kodes som `smallest-three` med 3×8-bit pluss 2-bit-mode i et RGBA-bilde (mode lander i alpha som `252 + largest`). Skaler og DC-SH kvantiseres hver med en 256-oppførings-codebook (persentil-basert fordelt over alle verdier), indeksene lander i `scales.webp` og `sh0.webp` . De fem bildene pluss en `meta.json` med codebooks og bounds pakkes i en ZIP-fil (custom-encoder, fordi sandboxen blokkerer system- `zip`) og lagres med endelsen `.sog` .

Sandbox-advarsel: SOG er det eneste formatalternativet som krever en ekstern binary. WebP-encoder-trinnet kaller `cwebp` fra `/usr/local/bin/cwebp` eller `/opt/homebrew/bin/cwebp`. Hvis ingen `cwebp`-binary finnes, faller koden tilbake på rå PNG-encoding — men: **PNG-fallback fungerer ikke i SuperSplat**. I App Store-versjonen evalueres tilgjengeligheten ut fra build-varianten; i developer-varianten må `cwebp` være installert via Homebrew (`brew install webp`).

KORT FORKLART

Det aller minste 3DGS-formatet, markant mindre enn SPZ. Men: krever `cwebp`-verktøyet på Macen din, fordi RadianceKit selv ikke kan generere alle bildeformater. Installer det én gang med Homebrew, så kjører alt. I App Store-versjonen muligens ikke fullt funksjonell — hvis det kommer PNG i stedet for WebP ut ved eksport, kan du ikke åpne filen direkte i SuperSplat. Hvis du arbeider uten Homebrew, så velg heller SPZ (E3).

E5 — glTF (.glb)

HVOR

Menylinjen → Export → 3D Formats → Export glTF...
Simple Mode: formatkort „glTF«. **Størrelse:** sammenlignbar med PLY. **Kompatibel med:** glTF-viewere med `KHR_gaussian_splatting-extension` (Khronos-draft-standard).

TEKNISK

Skriver en selvstendig `.glb` -binærfil (ingen separat bin-fil-vedlegg) i samsvar med `KHR_gaussian_splatting-extension`-spesifikasjonen. Posisjoner lagres som vanlige glTF- `POSITION` -vertex-data (float3), alle andre attributter (rotasjon som float4, scale som float3, opacity som float, SH-koeffisienter som float3 × `shCoeffCount`) ligger i ytterligere vertex-attributter og refereres via extensionen. Viktig: glTF bruker høyrehendt Y-up-koordinatsystem, COLMAP/3DGS arbeider Y-down/Z-forward. Eksportøren anvender derfor en 180-graders rotasjon om X-aksen — posisjoner skrives om med `(x, -y, -z)`, quaternioner tilpasses til `(w, x, -y, -z)`. Det gir en geometrisk korrekt, hendet (ikke speilvendt) gjengivelse i glTF-viewere. JSON- og binærchunks paddes til 4-byte-alignment, som GLB-standardens krever.

KORT FORKLART

Det offisielle Khronos-standardformatet for 3D-data, i den ferske utvidelsen for Gaussian Splats. Fordel: glTF er utbredt i alle store 3D-engines (Babylon.js, Three.js, Unity, Unreal). Ulempe: utvidelsen er i 2026 fortsatt i draft-stadiet, mange viewere kan den ikke ennå. Meningsfullt særlig hvis du integrerer splat-data i en eksisterende glTF-pipeline eller skriver en viewer som allerede er glTF-i-stand.

E6 — Splat (.splat)

HVOR

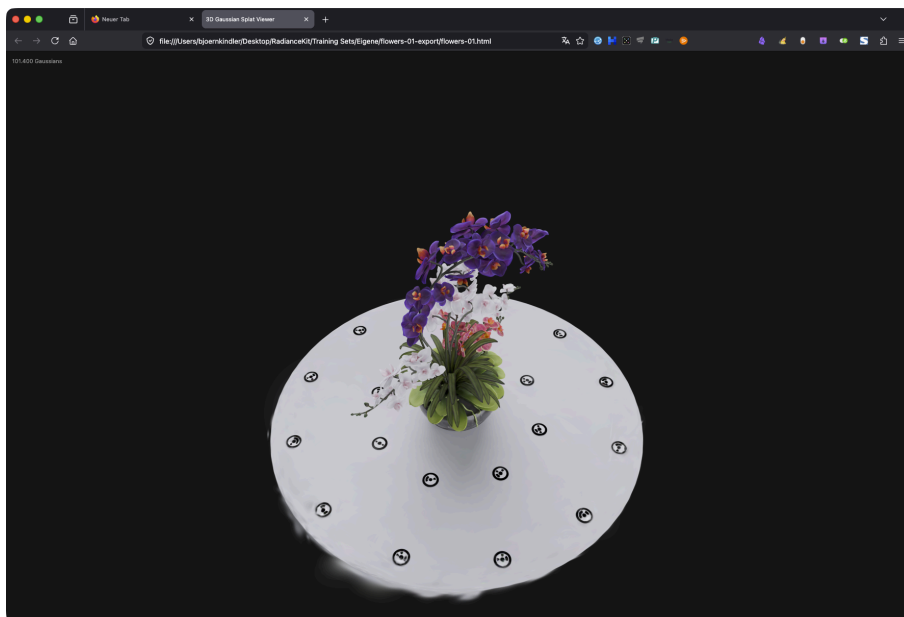
Menylinjen → Export → 3D Formats → Export .splat... Simple Mode: formatkort „.splat«.
Størrelse: nøyaktig 32 bytes per gaussian.
Kompatibel med: gsplat.js, webbaserte viewere (antimatter15-referanse), de fleste browser-3DGS-demoer.

TEKNISK

antimatter15- .splat -formatet — 32 bytes per gaussian, ingen header, ingen indireksjon. Layout per oppføring: 3 × float32 posisjon (verdens-koordinater), 3 × float32 scale (exp-transformert fra log-space i den interne bufferen), 4 × uint8 RGBA-farge (DC-SH-koeffisient skalert med `SH_C0 = 0.282...` og klampet til [0,255]), 4 × uint8 quaternion (w,x,y,z, normalisert og kodet som `128 + 128*q` i byte-området). Bare DC-SH lagres — høyere SH-bånd forkastes. Det gjør formatet ekstremt kompakt, men koster de view-avhengige fargeendringene som oppstår ved refleksjoner eller spekulære høylys. Skriverekkefølgen er nøyaktig skyens indeksrekkefølge (ingen romlig sortering), webviewere som `gsplat.js` rendrer ut fra det.

KORT FORKLART

Formatet å velge hvis du vil vise splaten i en egen web-viewer med `gsplat.js`. Veldig kompakt (32 bytes/gaussian), men ingen høyere SH-grad — altså ingen blanke refleksjoner eller fine fargeendringer avhengig av synsvinkel. For de fleste web-anvendelser ingen sak, fordi DC-farge fullstendig holder, og den manglende view-avhengigheten nesten ikke synes.



Figur 31: Web Viewer åpnet i Firefox — buketten-splat renderet med omliggende kamera-markør-kuler, browser-tab-bar synlig øverst, ingen CDN-/serveroppsett nødvendig. Selvsendig `bukett-01.html` åpnet direkte fra Finder ved dobbeltklikk i standardnettleseren — det innebygde WebGL2-programmet renderer gaussian-skyen umiddelbart uten nettverk eller server. De svarte markørene rundt buketten er treningskameraene, valgfritt kan slås på. Mus-dra roterer, scroll zoomer.

E7 — Web Viewer (.html)

HVOR

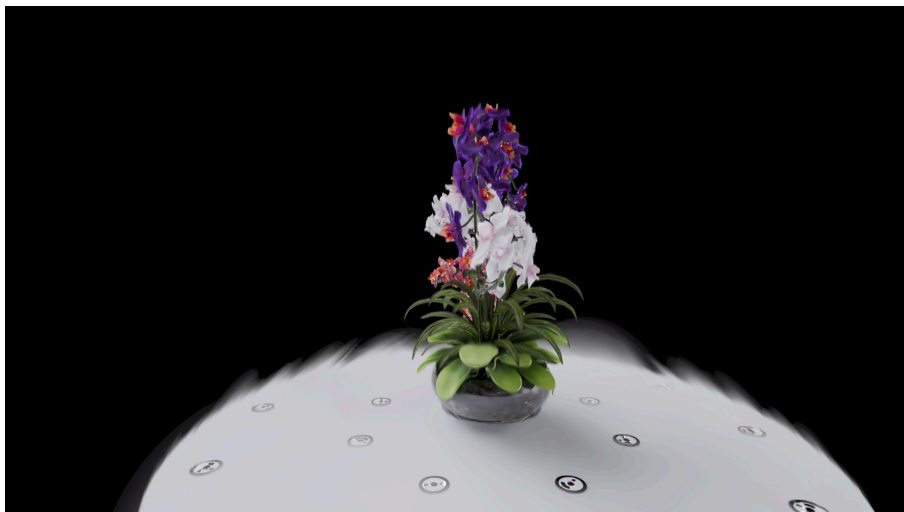
Menylinjen → Export → Media → Export Web Viewer... Simple Mode: formatkort „Web Viewer«. **Størrelse:** splat-data base64-kodet ($\approx 4/3$ overhead) + ca. 5 KB HTML/JS-shell. **Kompatibel med:** enhver moderne nettleser med WebGL2 (alle desktops, iOS 15+, Android 5+).

TEKNISK

Bundler gaussian-skyen sammen med en fullstendig inline skrevet WebGL2-renderer i en enkelt `.html`-fil. Det er ingen CDN-avhengigheter, ingen WASM, ingen annen fil. Skyen kodes internt først som `.splat`-binær (samme 32-byte-logikk som E6), deretter base64-innebygd, deretter dekodet i nettleseren med `atob`. Den innebygde rendereren gjør egen WebGL2-sortering, mus-orbit-styring og CPU-sortering per frame; hele JS-koden (shadere, matematikk, loop) kan ses i output-HTML-en. Akse-konvensjonen på lagring-til-renderer-grensen er nøyaktig den samme som i E5: posisjon $(x, -y, -z)$, quaternion $(w, x, -y, -z)$. Valgfritt kan et branding-overlay slås på (free-tier-kontakt). Da alt er inline, fungerer filen også direkte fra `file://`-protokollen — ingen lokal webserver nødvendig for test.

KORT FORKLART

En enkelt HTML-fil som du kan sende til noen per e-post eller bygge inn på en nettside. Dobbelklikk i Finder, og nettleseren viser scenen din med musedreining. Ingen opplasting til en sky nødvendig, ingen annen fil, ingen server. Ideelt for kundepresentasjoner, portefølje, e-postvedlegg. Ulempe: filen blir ca. en tredjedel større enn en ren `.splat` på grunn av base64-kodingen — for veldig store scener kan det derfor lønne seg med separat hosting av `.splat`-filen sammen med en standard-viewer.



Figur 32: Enkelt frame trukket ut fra `bukett-01.mp4` — buketten i profil-render, hvit plattform med kameramarkører synlig, svart bakgrunn (default-viewport-bakgrunn, kan endres i Settings). Kameraet kretser om scenen på en parametriske bane (elevasjon + avstand fast, yaw roterer), varighet typisk 6–10 sekunder ved 30 eller 60 fps. Frame-oppløsning skalerbar fra 480p til 8K via VideoPreset.

| E8 — Orbit Video (.mp4/.mov)

HVOR

Menylinjen → Viewport → Record Turntable Video
ELLER menylinjen → Export → Media → Export Orbit Video.... Simple Mode: formatkort „Orbit Video« med varighetsglider 3–30 s. **Størrelse:** avhenger av varighet, oppløsning, bitrate. **Kompatibel med:** alle plattformer (H.264 og HEVC er Apple-standard).

TEKNISK

Rendrer gaussian-skyen langs en parametrisk orbit-kamerakjøring og enkoder hvert frame via AVAs- setWriter til en MP4- eller MOV-fil. Orbit-konfigurasjonen styrer rotasjonshastighet (omdreininger), avstand, elevasjon, FOV, varighet og ease-in/out-faktor. Orbit-video-eksporten kjører gjennom Radian- ceKits EGEN ForwardPass med full SH-evaluering — piksel-identisk med in-app-viewporten (WYSIWYG). Per frame multipliseres verdens-tilpasningsmatrisen (beregnet av rendereren for å dreie de interne koordinatene inn i Y-up-orbit-verdenen) med kameraet, deretter brukes en kamera-konvensjons-flip (cam-Flip: orbit Y-up → COLMAP Y-down). Offscreen-render-targetet trekkes via IOSurface til en CVPixel-Buffer for encoderen. Encoderen støtter H.264 og HEVC, konfigurerbar bitrate og oppløsning fra 480p til 8K. Før første frame venter rendereren 200 ms, slik at den innledende splat-sorteringen er avsluttet. Denne eksporten er GPU-bound — ved 8K og millioner av gaussians ligger render-tiden per frame på flere sekunder, altså samlede render-tider på 10–30 minutter for 6 s video mulig.

KORT FORKLART

En ferdig MP4-fil med en dreining rundt scenen din. Perfekt for sosiale medier, markedsføring, presentasjoner. Du kan stille inn varighet (3–30 sekunder), dreieretning og hastighet. Filen kan bygges inn direkte på YouTube, Instagram, i PowerPoint og alle andre steder. Går noen ganger sakte, fordi appen må rendre hvert frame komplett — for en 8K-video kan du regne med fem til tretti minutter, avhengig av antall gaussians.

I E9 — SfM Transforms (transforms.json)

HVOR

Menylinjen → Export → Photogrammetry → Export SfM (transforms.json)... **Størrelse:** typisk 1–10 KB (kun posisjoner + intrinsics, ingen bilder, ingen gaussians). **Kompatibel med:** nerfstudio, Brush, gsplat, OpenSplat, Meshroom, alle moderne feed-forward 3DGS-trenere.

TEKNISK

Skriver nerfstudio- `transforms.json` -formatet med en liste over kameraposisjoner pluss delte intrinsics. Per kamera inverteres view-matrisen (RadianceKit-internt: world-to-camera i COLMAP-konvensjon), deretter speiles kameralokale Y- og Z-basisvektorer for å konvertere til nerfstudio-konvensjonen (OpenGL-stil, kamera ser langs `-Z`, `+Y` er opp). Den endelige 4×4-matrisen lander som row-major nested array av doubles i `transform_matrix` -feltet i hver frame. Intrinsics lagres på toppnivå (brennvidde `x/y`, hovedpunkt `x/y`, bildebredde/-høyde, `camera_model = "OPENCV"`, pluss `distortion-coefficients` `k1, k2, p1, p2`) — bortsett fra hvis eksportøren gjenkjenner flere forskjellige intrinsics-sett, da skrives de per frame. Bildestier skrives som `images/<filename>` relativt til JSON-filen; brukeren må opprette en `sibling-images/-`mappe med treningsfotoene.

KORT FORKLART

Denne JSON-filen beskriver for hvert foto hvor kameraet sto, og hvor det så hen. Filen alene er liten og ubrukelig — den brukes sammen med originalbildene i en mappe. Nerfstudio, Brush og noen andre trenere leser nøyaktig dette formatet, og du kan dermed overlevere RadianceKit-SfM-resultatene dine til et annet verktøy, uten at kamerarekonstruksjonen må beregnes på nytt der. Sparer timer på store scener.

I E10 — COLMAP Workspace (sparse/0/)

HVOR

Menylinjen → Export → Photogrammetry → Export SfM (COLMAP Workspace).... **Størrelse:** tre binære-filer til sammen typisk 4–8 MB — `points3D.bin` dominerer (én linje per 3D-punkt i sparse-skyen), `images.bin` og `cameras.bin` er hver markant under 100 KB. **Kompatibel med:** COLMAP selv, Nerfstudio, Postshot, Meshroom, alle verktøy som forventer et COLMAP-`sparse/`-katalog.

TEKNISK

Skriver standard-COLMAP-`sparse/0/`-layoutet med tre binære filer: `cameras.bin`, `images.bin`, `points3D.bin`. Formatreferanse er den offisielle COLMAP-dokumentasjonen. `cameras.bin` inneholder den dedupliserte intrinsics-listen (kameraer med identiske intrinsics + bildestørrelse samles til en enkelt oppføring); den benyttede kamera-modellen er `OPENCV` (modell 4), med `fx/fy/cx/cy` pluss de fire `distortion-coefficients` `k1/k2/p1/p2`. `images.bin` lister per bilde posisjonen som `wxyz`-quaternion pluss translation, etterfulgt av kamera-ID og filnavn; ingen 2D-3D-korrespondanser lagres. `points3D.bin` inneholder SfM-punktskyen med posisjon, farge (0-255 RGB) og default-verdier for reprojeksjon og track-length. Alt skrives i Little-Endian. Re-import i RadianceKit fungerer via File-menyen → „Import COLMAP/Metashape Workspace...« (se Q3 i SfM-backend-kapittelet).

KORT FORKLART

Det offisielle COLMAP-formatet. Hvis du vil fortsette treningen din i Postshot, Nerfstudio eller en annen COLMAP-i-stand programvare, er det veien. Tre små filer pluss originalbildene dine, og målprogrammet aksepterer det som om COLMAP selv hadde vært kildeprogrammet. Flere programmer forstår det enn `transforms.json`-formatet (E9), samtidig litt mindre hendig, fordi det er binært i stedet for tekstbasert.

Hvilket format når?

| Mål | Format |
|--|---|
| Web-viewer på egen side | E7 Web Viewer (.html) |
| Web-viewer med <code>gsp1at.js</code> | E6 Splat (.splat) |
| Pipeline-gjenbruk 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 påkrevet) |
| Markedsføring/social-video | E8 Orbit Video |

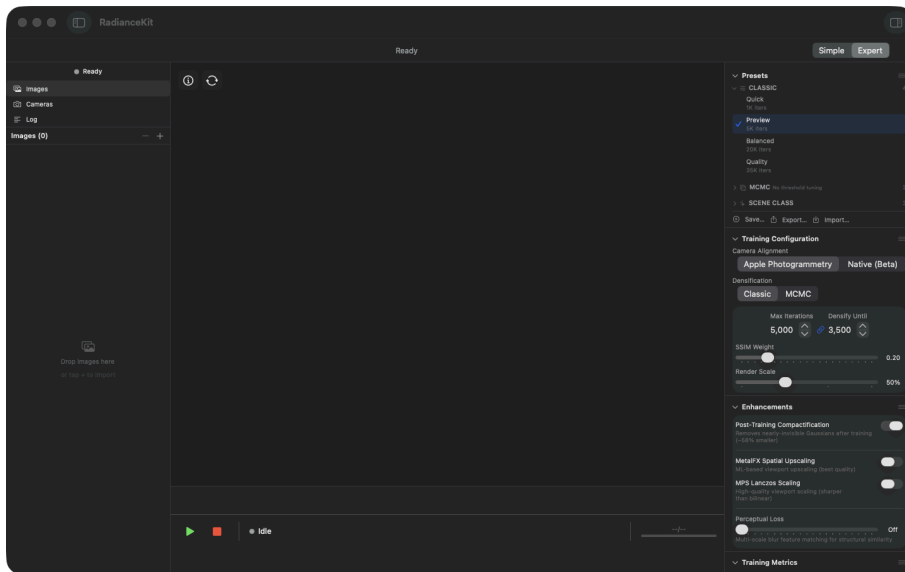
Rask sammenligning

| Format | Endelse | Sandbox | Størrelse (1M gauss) | Best til |
|----------------------|-------------|------------------|----------------------|-------------------------------|
| E1 PLY | .ply | ja | ~250 MB | Arkiv, høyeste 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 nettleseerfil |
| E8 Orbit Video | .mp4 / .mov | ja | variabel | Social/markedsføring |
| E9 SfM Transforms | .json | ja | ~5 KB | Pose-overlevering |
| E10 COLMAP Workspace | Katalog | ja | ~4–8 MB | Pose-overlevering binær |

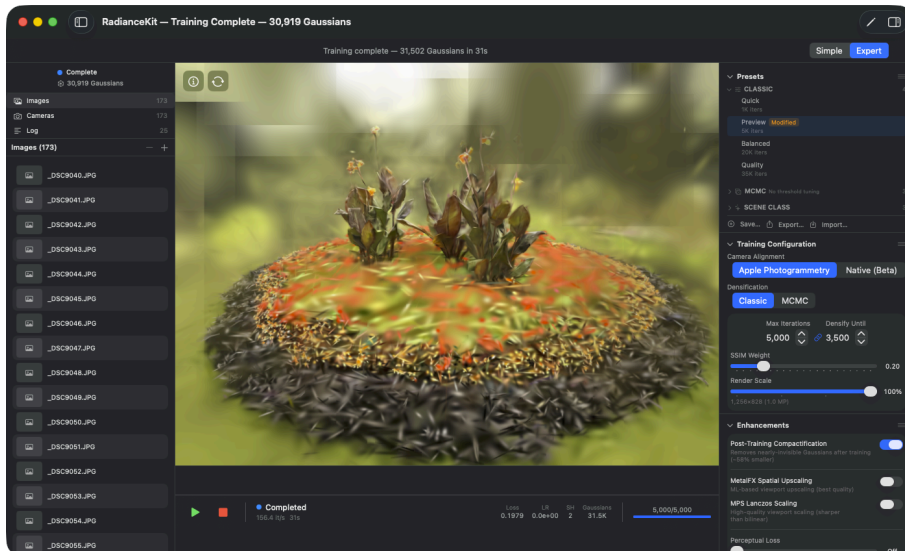
Størrelseskolonnen er grove pekepinner for 1 mio. gaussians ved SH-grad 3. Reelle verdier varierer avhengig av scenens komprimerbarhet; SH-grad 0 reduserer PLY/glTF med faktor 4.

KAPITTEL

Kapittel 9 — SfM-backends



Figur 33: Ekspertmodus med Camera Alignment-velger i Inspector (Apple Photogrammetry / Native (Beta))



Figur 34: Inspector med Native (Beta) aktiv — andre alternativ i Camera Alignment-velgeren valgt, alle andre treningskonfigurasjonsparametere uendret

HVA BILDET VISER Camera Alignment-velgeren i Inspector er en segmentert kontroll med to alternativer — Apple Photogrammetry (standard for App Store-builds, fullt sandbox-konform) og Native (Beta) (RadianceKits eget FAST+BRIEF+GLOMAP-pipeline-backend,

utviklet i fase 3.8/3.9, per 2026-05). Native (Beta) er orbit-only validert og raskere ved $\geq 1\,000$ frames enn Apple Photogrammetry, men oppfyller ennå ikke fase-3-§5-kvalitetsgaten ($\text{finalLoss} \leq 0.0115$) — derav Beta-merket. Eksterne SfM-resultater fra Metashape, COLMAP eller annen fotogrammetri-programvare kan dessuten importeres via File-menyen (Q3 COLMAP-tekstformat, Q6 Workspace-import) — velgeren skifter ikke, men de importerte posisjonene erstatter SfM-resultatet.

SfM står for **Structure from Motion**. Ut fra en mengde overlappende fotos rekonstruerer programvaren for hvert bilde kameraets posisjon og blikkretning i et felles 3D-koordinatsystem. Underveis genereres en grov 3D-punktsky som initialiserer treningen med Gaussian Splatting. SfM-resultatet er input til selve treningen og er avgjørende for den senere bildekvaliteten.

RadianceKit tilbyr fem SfM-veier: to backends innebygd i appen (Q1 Apple Photogrammetry, Q4/Q5 Native), to importstier fra eksterne verktøy (Q3 COLMAP-tekstformat, Q6 binær Workspace-import) samt Q2 COLMAP-binary, som bare er tilgjengelig i developer-builds utenfor App Store. Hvilken som er den rette, avhenger av scenetypen (orbit rundt et objekt, innendørs rom, droneflyging), og av om en ekstern programvare allerede leverer en rekonstruksjon.

I Q1 — Apple Photogrammetry

HVOR

Expert View → Inspector → Treningskonfigurasjon
→ Camera Alignment-velger, oppføringen „Apple Photogrammetry«.

TEKNISK

Innkapsler Apples innebygde fotogrammetri-framework, som opprinnelig ble utviklet til Object Capture. Apple trekker internt ut features med en proprietær pipeline (trinn er ikke offentlig dokumentert), verifiserer dem via multi-view-matching og løser bundle-adjustment på Apple Silicon Neural Engine + GPU. Backend er fullt App Store-konform (ingen ekstern binary, Sandbox=true, on-device), men leverer bare kameraposisjoner pluss en grov punktsky — ingen diagnose-metrikker som spor-lengde eller reprojeksjonsfeil. Skalerer ifølge Apples anbefaling opp til et par hundre bilder. Ved mer enn ~500 frames i lineære droneflyginger eller store outdoor-scener er det reproduserbart observert krasj eller stille frasortering av enkelte kameraer.

KORT FORKLART

Det er den enkleste veien. Bilder inn, app regner. Fungerer veldig bra ved klassiske objektskanninger — når du går rundt et møbel eller en skulptur og tar 50–200 fotos. Ved droneflyginger over landskaper eller ved virkelig mange bilder (over 500) blir Apples metode imidlertid gjerne ustabil. Til den slags scener kan du prøve ut Native-backend (Q4/Q5) eller regne kameraene i Metashape og laste dem inn via Workspace-importen (Q6).

POWER-USER

Q2 COLMAP-binary — starter det eksterne COLMAP-programmet som underprosess og er derfor **ikke tilgjengelig** i App Store-versjonen (sandbox). Virker bare i developer-builds utenfor App Store. For den kvaliteten COLMAP leverer, finnes det i App Store-versjonen Workspace-importen (Q3 eller Q6): kjør SfM i COLMAP eller Metashape utenfor appen, og last inn resultatet.

Q3 — COLMAP-tekstformat (Metashape / ETH3D) **HVOR**

Menyen „File → Import COLMAP / Metashape Workspace...« (Cmd+⇧+I) ELLER dra-og-slipp av en mappe med `sparse/0/cameras.txt`.

 **TEKNISK**

Leser den standardiserte COLMAP-teksteksporten — tre tekstfiler `cameras.txt`, `images.txt`, `points3D.txt` i undermappen `sparse/0/` — og konverterer til den interne SfM-resultat-modellen. Samme formatdefinisjon som COLMAP-binæreksporten, bare som ASCII i stedet for binær. Skriver ut av Agisoft Metashape, RealityCapture, PolyCam og ETH3D-benchmarken i nøyaktig dette oppsettet. Parseren deler kameramodell-deteksjon med binary-parseren (alle gjengse modeller: SIMPLE_PINHOLE, PINHOLE, OPENCV, OPENCV_FISHEYE, FULL_OPENCV). Robust mot kommentarlinjer og tomme linjer. Skalerer i tester opp til ~1 400 kameraer (ETH3D Tunnel) uten problemer.

 **KORT FORKLART**

Hvis du allerede har arbeidet med Metashape, RealityCapture eller en annen kommersiell foto-3D-programvare og har eksportert resultatet — kan du laste denne eksporten direkte inn i RadianceKit uten at appen selv må regne på nytt. Det sparer timer med venting. Last inn hele mappen via File-menyen eller dra den inn i vinduet.

I Q4 — Native SfM (inkrementell)

HVOR

Expert View → Inspector → Treningskonfigurasjon → Camera Alignment-velger, oppføringen „Native (Beta)«. Inkrementell er standardmodus for denne backenden — det finnes ingen separat mapper-velger i Inspector. Via CLI kan modus settes eksplisitt med `--native-sfm` eller `--sfm-mapper incremental`.

TEKNISK

Egen GPU-akselerert implementasjon av hele SfM-pipelinen: FAST+BRIEF-features EL-LER SuperPoint+LightGlue via CoreML (med `--coreml-features`), etterfulgt av Hamming-KNN-matching, RANSAC-fundamentalmatrise, track-building, initial-par-utvelgelse, two-view-bootstrap (F→E pluss DLT), grådig inkrementell mapper med PnP-registrering og multi-view-triangulering og til slutt bundle-adjustment via Schur-reduisert Levenberg-Marquardt med Huber-tap og analytiske jacobianer over Cholesky-løsning. Fullt App Store-konform: ingen ekstern binary, Sandbox=true. Med R2-collapse-detektoren som ble levert i fase 3.10: registrerer appen mindre enn 60 % av input-frames eller faller points-per-camera-raten under 13, byttes det automatisk til den globale mapperen (Q5). Empirisk ren på orbit-/turntable-scener; ved mer generelle bevegelser (droneflyging, innendørs rom med kompleks geometri) er suksessraten lavere — detektoren fanger imidlertid disse tilfellene. Skalerer pålitelig opp til ~200 kameraer, høyere med markant lengre kjøretid.

KORT FORKLART

Apples styrker (App Store-kompatibel, rask til orbits) med ytterligere diagnoseverdier. Fungerer spesielt godt når du, som ved en Object Capture, går rundt et motiv. Ved mer kompliserte opptak (droneflyging eller stue) gjenkjenner RadianceKit automatisk at det ikke lykkes, og hopper over til den globale metoden. Merket „Beta« fordi den fortsatt utprøves — standardanbefalingen er fortsatt Apple Photogrammetry til enkle objektskanninger og Workspace-importen (Q3 eller Q6) til krevende outdoor-sett.

I Q5 — Native SfM (global)

HVOR

Kalles automatisk når den inkrementelle mapperen (Q4) utløser collapse-detektoren (mindre enn 60 % av input-frames registrert eller points-per-camera-rate under 13). Manuelt tvangs-aktiverbar kun via CLI `--sfm-mapper global`. I Inspector er den globale metoden ikke tilgjengelig via en separat velger — appen bestemmer selv når den skifter.

TEKNISK

Global variant av den native pipelinen. Først feature-ekstraksjon + matching som i Q4, deretter relativ pose-estimering for alle verifiserte par, etterfulgt av rotation-averaging (synkroniserer alle kamerarotasjoner i verdens-koordinatsystemet) og translation-averaging (LSQR-basert på en matrix-fri sparse-formulering for å unngå integer-overflow ved store kameramengder). Skalerer i prinsippet til ~5 000 kameraer, i praksis kvalitetsforringet over noen hundre kameraer — fase-3.8-§5-akseptans-gate-målingen på K-1351 ga finalLoss 0.07 i stedet for de tilstrekte 0.0115. Håndteres som „fallback-tier«: kommer i spill når den inkrementelle mapperen degenererer, men kontrolleres ikke selv på nytt for kvalitet.

KORT FORKLART

Plan-B-stien for den native engine. Kalles automatisk når den raskere inkrementelle stien svikter. Leverer et brukbart resultat, men er ved svært store eller vanskelige scener sjelden så presis som det du får fra Metashape eller en ekstern COLMAP-installasjon. Hvis Native blir din standardarbeidsflyt, kan det i slike tilfeller lønne seg å gå omveien via Workspace-importen (Q3 eller Q6).

I Q6 — Metashape / COLMAP-tekst-workspace-import (fase Q7)



HVOR

File-menyen → „Import COLMAP / Metashape Workspace...« (Cmd+⇧+I). Dra-og-slipp av en mappe med `sparse/0/cameras.{bin,txt}` og `images/`.



TEKNISK

Gjenkjenner automatisk om en mappe valgt via dra-og-slipp eller åpne-panelet svarer til ett av de tre COLMAP-workspace-layoutene (`sparse/0/`, `sparse/` eller roten), og om rekonstruksjonen finnes som binær (`cameras.bin`) eller tekst (`cameras.txt`). Den binære stien bruker COLMAP-binær-parseren, tekst-stien ETH3D-lasteren — begge produserer samme SfM-resultat-modell, og resten av pipelinen (importer bilder, start MCMC-trening) er agnostisk overfor kilden. Bildene åpnes via app-sandbox-bokmerkesystemet `security-scoped`, slik at importen også virker i App Store-versjonen. Spesifikt tiltenkt situasjonen „Metashape-eksport uten å regne rekonstruksjonen om«. Deteksjonen som nevnes i File-meny-oppføringen, advarer i app-loggen hvis den valgte mappen ikke er et gjenkjennelig workspace.

KORT FORKLART

Helt spesifikt funksjonen for Metashape-brukere. Hvis du har en lisens til Metashape eller RealityCapture og har laget kamerarekonstruksjonen der, kan du bare dra eksportmappen inn her og umiddelbart starte treningen. Sparer flere timers regnetid på store scener fordi RadiancKit da ikke selv må kjøre SfM-en.

Hvilken backend når?

| Scenario | Anbefalt backend |
|---|--|
| Objektskanning, 50–200 fotos | Q1 Apple Photogrammetry |
| Stor outdoor / drone / >500 bilder | Q6 Workspace-import (regn i Metashape eller COLMAP, last så inn) |
| Metashape/RealityCapture-eksport foreligger | Q6 Import (ingen SfM nødvendig) |
| ETH3D / akademisk COLMAP-tekstsett | Q3 COLMAP-tekst-import |
| Strengt App Store-konform + orbit-scene | Q4 Native inkrementell |
| Q4 feiler | Q5 Native global (automatisk) |
| ETH3D-benchmark-data | Q3 (autotest precomputed) |

Rask sammenligning

| Back-end | App Store | Sand-box | Ekstern binary | Best til | Maks ~kameraer |
|------------------------|-------------------------|----------|----------------|---------------------|----------------|
| Q1 Apple PG | ✓ | ✓ | — | Orbit-objekt | ~300 |
| Q2 COLMAP Binary | ✗ (kun developer-build) | — | colmap/glomap | Outdoor stor | ~5 000 |
| Q3 COLMAP-tekst-import | ✓ | ✓ | — | Bench-rigger | ~1 500 |
| Q4 Native inkrementell | ✓ | ✓ | — | Orbit-objekt | ~200 |
| Q5 Native global | ✓ | ✓ | — | Q4-fallback | ~1 351 |
| Q6 Workspace-import | ✓ | ✓ | — | Meta-shape-gjenbruk | pr. kilde |

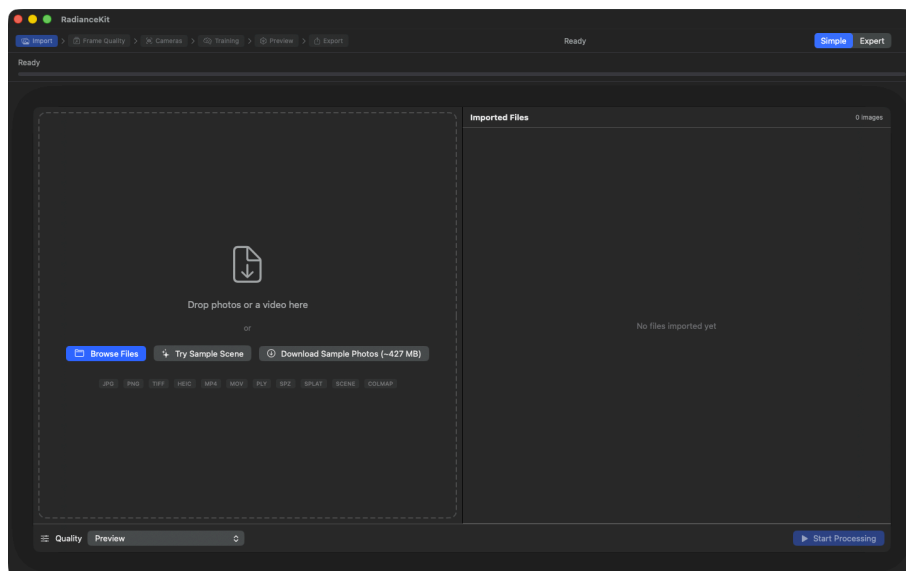
KAPITTEL

Kapittel 10 — Begynnermodus

Begynnermodus (engelsk Simple Mode, Cmd+1) er den veiledede arbeidsflyten for alle som for første gang rekonstruerer en 3D-Gaussian-Splatting-scene. I stedet for å vise et sidefelt fylt med Inspector-felter fører appen gjennom fire trinn: først importeres bilder eller en video, og en kvalitetspreset velges, så kjører behandlingen (SfM + trening), etterpå kan den ferdige scenen vurderes i en 3D-forhåndsvisning, og til slutt eksporteres til ønsket format. En smal fremgangsbjelke øverst i vinduet viser til enhver tid hvilket trinn du befinner deg på.

Sammenlignet med Expert Mode (Cmd+2), som viser alle betjeningsfelter samtidig, skjuler begynnermodus ubrukte alternativer, gir valideringsadvarsler ved for få eller dårlige bilder og tilbyr på hvert trinn bare de knappene som er meningsfulle i den aktuelle tilstanden. Du kan når som helst bytte mellom begynner- og Expert Mode (Cmd+1 / Cmd+2), hele tilstanden — importerte bilder, valgt preset, pågående trening, ferdig punktsky — bevares og er umiddelbart tilgjengelig i den andre modusen.

Z1 — Import (velg bilder & preset)



Figur 35: Simple Mode trinn 1 — tom drop-zone for bildeimport, crumb-trail øverst (Import → Frame Quality → Cameras → Training → Preview → Export), format-pills JPG/PNG/TIFF/HEIC/MP4/MOV/PLY/SPZ/SPLAT/SCENE/COLMAP

HVA BILDET VISER Crumb-trail (Import aktiv) viser den firetrinnsarbeidsflyten. Venstre drop-zone med tre CTA-er: „Browse Files« (NSOpenPanel), „Try Sample Scene« (bund-

led demo), „Download Sample Photos (~427 MB)« (Mip-NeRF360 flowers subset). Format-pills under lister alle aksepterte filtyper. Til høyre „Imported Files« med teller „0 images« og empty-state „No files imported yet«. Nederst quality-velger (default: Preview) og „Start Processing« (deaktivert så lenge ingen bilder er der).

Første trinn består i å gi appen bildemateriale. Via dra-og-slipp i det store, stiplede feltet i midten, via „Browse Files«-knappen eller via klikk på den medfølgende sample-scenen. Til høyre dukker det opp en liste over alle importerte bilder med oppløsning og filstørrelse; nederst i den svevende verktøylinjen velger du kvalitetspresetet og starter pipelinen med „Start Processing«. Valideringsadvarsler (rød ved < 3 eller < 10 bilder, oransje ved 10–19) viser om appen forventer en meningsfull rekonstruksjon eller ikke.

C-01 ProgressIndicator (trinn-visning)

HVOR

Øverst over arbeidsflyten, alltid synlig.

TEKNISK

Viser en horisontal fremgangsbjelke over hele pipelinen (frame-quality → SfM → trening) med stage-allokering: frame-quality opptar 0–5 % (fase 3.11, veldig kort), SfM opptar 0–30 % av bjelken, trening 30–100 %. Ved siden status-tekst og fase-navngitt prosent-visning („SfM 41 %«, „Training 12 500/20 000«), slik at brukere ikke leser det tilsynelatende tilbakeskrittet „41 % SfM → 25 % trening« som feil — bjelken viser hele pipeline-fremgangen, ikke sub-stage. ETA-beregning starter så snart det er målt nok treningstempo (typisk etter de første 100 iterasjonene). Samme visning brukes også i Expert Mode over Inspector.

KORT FORKLART

Den smale bjelken helt oppe er kartet ditt gjennom arbeidsflyten. Den forteller deg ikke bare hva appen gjør akkurat nå (justerer kameraer, trening kjører, ...), men også hvor langt den samlet allerede er. Oppdelingen er bevisst slik at kamera-beregningen opptar den første tredjedelen av bjelken, og selve treningen de to bakre tredjedelene — ellers ville fremgangen etter SfM virke som om den plutselig var tilbake på null. Du kan altså lene deg tilbake, et blick på bjelken holder for å se den grove etappen. Teksten ved siden forteller deg om du akkurat er i SfM-trinnet (f.eks. „SfM 41 %«) eller i treningen (f.eks. „Training 12 500/20 000«), slik at tallene ikke virker forvirrende. Hvis du ikke får vist ETA, er treningen bare ennå for ung — appen estimerer først så snart den har målt nok tempo.

C-03 DropZoneView (dra-og-slipp-område)

HVOR

Venstre side av import-trinnet, stort stiplet rektangel med symbol. Viser i begynnermodus med label „Drop photos or a video here«.

TEKNISK

Drop-område som lar symbolet kort hoppe og farger bakgrunnen så snart drag-items svever over feltet. Aksepterer JPG, PNG, TIFF, HEIC, MP4, MOV, PLY, SPZ, .splat, .radiance-scene-bundles og mapper. Drop-routing etter type: bilder samles og overgis sortert, videoer utløser frame-sampling-stien, splat-filer åpner direkte forhåndsvisningen, scene-bundles lastes inn. Mapper enumereres, og alle inneholdte bilder importerer. Security-scoped bookmarks for sandbox-konform tilgang opptas og frigis korrekt. Ustøttede endelser vises som advarselsbanner i 5 sekunder.

KORT FORKLART

Det store stiplede feltet er hovedbetjeningen i første trinn. Bare dra fotos eller en video dit, eller en hel mappe — appen tar alt den kjenner, og ignorerer resten. Når feltet blir blått og symbolet kort hopper, har appen gjenkjent draget. Slipp, og importen starter umiddelbart: bilder vandrer i listen til høyre, videoer utløser automatisk frame-sampling-trinnet, og allerede trente `.ply` / `.spz` / `.splat` -filer åpner direkte forhåndsvisningen. Hvis et format ikke passer (f.eks. PDF eller BMP), dukker det opp et kort hint øverst — appen sluker ikke ukjent materiale i stillhet.

C-05 Browse Files-knapp

HVOR

Inne i drop-zonen, prominent knapp.

TEKNISK

Knapp som åpner macOS-fildialogen med multivalg og filtypene JPG, PNG, TIFF, MP4, MOV, mapper samt appens eget scene-format. Resultat-URL-er er security-scoped og videreføres gjennom samme import-stier som dra-og-slipp. Hvis brukeren velger en mappe, enumereres den rekursivt for bilder.

KORT FORKLART

Hvis dra-og-slipp er ubekvem for deg, så klikk bare denne knappen og naviger i macOS-fildialogen til fotoene dine. Du kan velge flere filer samtidig (Cmd-klikk på de enkelte bildene) eller velge en hel mappe — appen gjennomfører da mappen rekursivt for alle støttede bildetyper. Det er spesielt praktisk hvis opptakene dine ligger innkapslet i undermapper (f.eks. „shoot-day1/«, „shoot-day2/») — ett klikk på hovedmappen holder. Funksjonelt gjør knappen nøyaktig det samme som dra-og-slipp; velg bare den veien som er mest bekvem for deg.

C-06 Try Sample Scene-knapp **HVOR**

Inne i drop-zonen, bare synlig hvis app-bundlen inneholder sample-scenen, og det ennå ikke er importert bilder/splats.

 **TEKNISK**

Dukker bare opp hvis (a) en `sample-scene.splat`, `.spz` eller `.ply` finnes i app-bundlen OG (b) ennå ingen bilder/videoer er importert, og ennå ingen punktsky finnes. Ved klikk lastes den ferdige punktskyen (foretrukket det minste formatet — `.splat` ~3 MB, `.spz` ~1.4 MB, fallback `.ply`) og setter etter 400 ms hardkodete kamera-verdier fra de originale metadata av blomster-scenen til et estetisk meningsfullt inngangsperspektiv.

 **KORT FORKLART**

Hvis du starter appen for første gang og bare vil se hva som kommer ut til slutt — klikk her. Det åpner en ferdig trent blomster-scene som du straks kan dreie og eksportere uten at appen må regne. Kameraet er forhåndsstilt til et estetisk meningsfullt inngangsperspektiv, slik at du straks ser noe pent. Perfekt for å prøve 3D-styringen og eksport-trinnet risikofritt før du går løs på egne opptak. Så snart du importerer egne bilder, forsvinner knappen automatisk — den vises bare så lenge prosjektet er helt tomt.

C-07 Download Sample Photos-knapp **HVOR**

Inne i drop-zonen, ved siden av „Try Sample Scene«; samme synlighetsbetingelser.

 **TEKNISK**

Utløser en nedlasting (repo github.com/bkindler/radiancekit-sample-photos) som laster inn ca. 427 MB med 960 fullopløsnings-frames og mater appen. Under nedlastingen er knappen deaktivert. Fremgangen vises i den øvre progress-baren som „Downloading X %« i en egen stage, fordi denne stagen beholder sin egen 0–100 %-skala og ikke overlapper det senere SfM-stage.


 **KORT FORKLART**

Akkurat som sample-scenen, bare med utgangsfotos i stedet for med det ferdige resultatet. Slik kan du selv kjøre hele pipelinen gjennom en gang og se hvor lang tid SfM og trening faktisk tar på Mac-en din. Nedlastingen er stor (omtrent en halv DVD = 427 MB), men skjer bare én gang — etterpå er fotos lokale, og du kan starte pipelinen vilkårlig mange ganger med forskjellige presets. Mens nedlastingen kjører, viser den øvre fremgangsbjelken den aktuelle nedlastingsstatusen i prosent, slik at du kan vurdere når det går i gang. Tips: ta helst et raskt WLAN eller kablet nett til dette — de 427 MB trekker ellers ut.

C-09 Quality Presets-velger **HVOR**

Svevende nedre verktøylinje av import-overlayet, til venstre for start-knappen.

 **TEKNISK**

Betjeningselement med label „Quality« grupperer de tilgjengelige presetene etter kategori (Classic / MCMC / Custom). Innebygde presets grupperes etter kategori; seksjonshodene er hardkodete. Custom-presets bare synlige hvis noen finnes. Locked-state: presets som ikke er i free-listen (Quick + Preview), får et „»-suffiks på navnet hvis brukeren ikke har kjøpt; ved valg springer velgeren tilbake på Preview og åpner automatisk purchase-sheetet. Ved valg anvendes presetet, noe som erstatter hele treningskonfigurasjonen.

 **KORT FORKLART**

Her velger du hvor presist og hvor lenge appen skal regne. „Quick« og „Preview« kan brukes uten kjøp og leverer et første resultat på få minutter — ideelt for å teste om bildene dine i det hele tatt er meningsfulle. „Balanced« og „Quality« krever fullversjonen og leverer markant rene modeller, men tar til gjengjeld timer i stedet for minutter. MCMC er en annen strategi som klarer seg med færre gauss-splats — bra hvis du senere vil eksportere modellen kompakt eller legge ut på nettet. Premium-presets kjenner du igjen på det lille låsesymbolet ved navnet; trykker du på en uten lisens, springer velgeren tilbake på Preview, og kjøps-sheetet åpnes automatisk. Tommelfingerregel: start alltid med Preview, se på resultatet, og bestem så om det er verdt å gjøre en lengre kjøring.

C-10 Start Processing-knapp **HVOR**

Svevende nedre verktøylinje av import-overlayet, til høyre for preset-velgeren.

 **TEKNISK**

Knapp som forblir grå så lenge hverken bilder eller en video er importert. Ved klikk startes pipelinen, og stage-maskinen bytter til rekkefølgen frame-quality → SfM → trening. Knappen selv har ingen ytterligere status; en kjørende behandling dukker opp i stedet som separat behandlingsskjerm.

 **KORT FORKLART**

„Sett i gang«-knappen. Så lenge den er grå, mangler det fortsatt input-bilder eller en video. Så snart du har dratt fotos inn, blir den aktiv, og du klikker den for å starte SfM og trening etter hverandre. Derfra overtar appen hele forløpet, og du lander automatisk på behandlingsskjermen (Z2). Du trenger ikke klikke videre — først etter trening bytter appen tilbake til forhåndsvisningen (Z3). Hvis du ombestemmer deg, kan du også deretter når som helst avbryte via Cancel.

C-11 Video Sampling-slides**HVOR**

Høyre bildeliste, bare synlig hvis en video (i stedet for bilder) er importert.

**TEKNISK**

Glidebryter 0.5 fps – 30 fps i 0.5-skritt. Ved endring oppdateres frame-tettheten, og dessuten beregnes antallet målframes (minst 10) ut fra tetthet og videolengde. Glidebryteren ligger utenfor bildelisten fordi liste-elementer ville blokkere muse-events fra glidebrytere. Under glidebryteren står de beregnede målframene („247 frames«) og videolengden („1m23s video«). Tooltip advarer: „Doubling the density doubles the number of frames and increases SfM time by ~100 %.«

**KORT FORKLART**

Hvis du har importert en video i stedet for fotos, bestemmer denne glidebryteren hvor mange enkeltbilder appen skal trekke ut av videoen. Flere bilder = bedre kvalitet, men lineært mer regnetid. For en 30-sekunders orbit-video er 5 fps (150 bilder) en god start; ved 1-minutts-opptak holder 3 fps ofte fullt ut. Under regulatoren viser appen live hvor mange frames som kommer ut ved den aktuelle innstillingen — så ser du straks om du treffer det fornuftige området på ca. 100–300 bilder. Hvis resultatet blir dårlig, så dra regulatoren til høyre og prøv igjen; dobling av frame-raten dobler imidlertid også grovt SfM-varigheten.

C-12 Clear All-knapp**HVOR**

Høyre bildeliste, nederst til høyre; bare synlig hvis bilder er importert.

**TEKNISK**

Rød knapp. Klikk åpner en bekreftelsesdialog med tittel „Clear all imported files?« og melding „N images will be removed.«. Bekreftelse tømmer alle importerte bilder/videoer, staging-mapper, punktskyen, treningsstatus, SfM-resultatet og alle caches; stagen hopper tilbake til import. Ved Cancel bevares alt. Dialogen er konfigurert som ikke-destruktiv default-sti (destruktiv knapp markert rød).

**KORT FORKLART**

Hvis du vil starte helt på nytt, så klikk her. Bekreftelsesspørsmålet dukker opp fordi slettingen kaster bort alle aktuelle importerte kameraer og treningsresultater — du kan ikke angre det. Meningsfullt hvis du vil bytte ut det valgte bildematerialet helt eller bli kvitt et gammelt prosjekt før du starter et nytt. Merk: å ta et enkelt bilde ut skjer via listen til høyre (se neste punkt), ikke via denne knappen. Filene dine på disken slettes ikke — appen glemmer bare referansene sine.

C-13 File List ForEach (enkeltbilde-fjern)



HVOR

Høyre bildeliste, hver oppføring.



TEKNISK

Liste over de importerte bildene med swipe-to-delete. Per bilde en linje med ikon, filnavn, oppløsning („1920 × 1080“) og filstørrelse (formatert KB/MB). Oppløsningen kommer fra en metadata-cache som asynkront fylles fra bilde-headerne, slik at brukergrensesnittet ikke blokkeres. Slett-handlingen tilbyr macOS-typisk swipe-delete (trackpad-swipe til venstre på en linje) samt tastatur-delete ved markert linje. Merk: den utvidede image-delete-stien med eksplisitt minus-knapp, backspace og Cmd-Z for å angre ble *bare i Expert Mode* lagt til i Project Navigator — i begynnermodus blir det ved swipe-delete.

KORT FORKLART

Listen til høyre viser hvert importerte bilde med oppløsning og filstørrelse — praktisk for med ett blick å se om du har blandet høy-oppløsningsmateriale med lav-oppløsningsmateriale. For å ta et enkelt bilde ut, swip det til venstre med to fingre på trackpaden — som i iOS Mail — eller velg det og trykk Delete. Appen sletter ikke selve filen; den tar den bare ut av det aktuelle prosjektet. Hvis du trenger en ekte minus-knapp eller Cmd-Z-angre, så bytt til Expert Mode (Cmd+2), der finnes det i Project Navigator. I begynnermodus blir det bevisst ved det enkle swipe-mønsteret.

C-15 Validation Warnings (3-trinns-tier)



HVOR

Under bildelisten, over Clear-All-knappen.



TEKNISK

Tre etter hverandre følgende terskler basert på antallet importerte bilder (bare aktiv hvis bilder finnes, og ingen video): - < 3 bilder: rødt banner (red octagon), tekst „At least 3 images are required. Camera alignment cannot be computed from fewer images.« - 3–9 bilder: 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 bilder: oransjefarget banner (warning triangle), tekst „Workable, but quality usually improves with 20+ images and good coverage around the scene.«

Fra 20 bilder forsvinner banneret. Terskelverdier er hardkodete og basert på empiriske 560+-trenings-eksperimenter.

KORT FORKLART

Appen kikker på hvor mange bilder du har importert og gir deg en farget vurdering. Rødt betyr: det blir med stor sannsynlighet ikke til noe — enten kan SfM ikke beregne kameraer, eller treningen overfitter på for lite materiale. Oransje betyr: kan kanskje lykkes, men regn ikke med toppkvalitet fordi algoritmen finner lite overlapp mellom bildene. Ingen banner betyr: gode forutsetninger, du har nok materiale. Hvis du vil ha virkelig rene modeller, så sikt etter minst 30–50 jevnt fordelte opptak rundt motivet ditt — gjerne markant flere ved utendørsscener eller store rom. Du kan starte tross advarsel, men ikke bli overrasket hvis SfM avbryter uten kommentar, eller modellen ser hullede ut.

C-16 COLMAP Workspace Detection **HVOR**

Ved drop av en mappe — ingen synlig knapp, men deteksjonslogikk.

 **TEKNISK**

Ved drop av en mappe sjekkes det om den inneholder et av de tre kanoniske workspace-layoutene: `sparse/0/cameras.bin`, `sparse/cameras.bin` eller direkte `cameras.bin` i roten. Hvis det er tilfellet, avbrytes standard-bilde-enumerasjonen, og i stedet åpnes en modal alert som spør brukeren om den eksisterende rekonstruksjonen skal brukes, eller bildene skal sendes gjennom Apple Photogrammetry på nytt. Samme sti også for tekst-format-workspaces (`cameras.txt`) og ETH3D-eksporter. Se kapittel 9 backend Q6 for detaljer. Virker i begynnermodus akkurat som i Expert Mode.

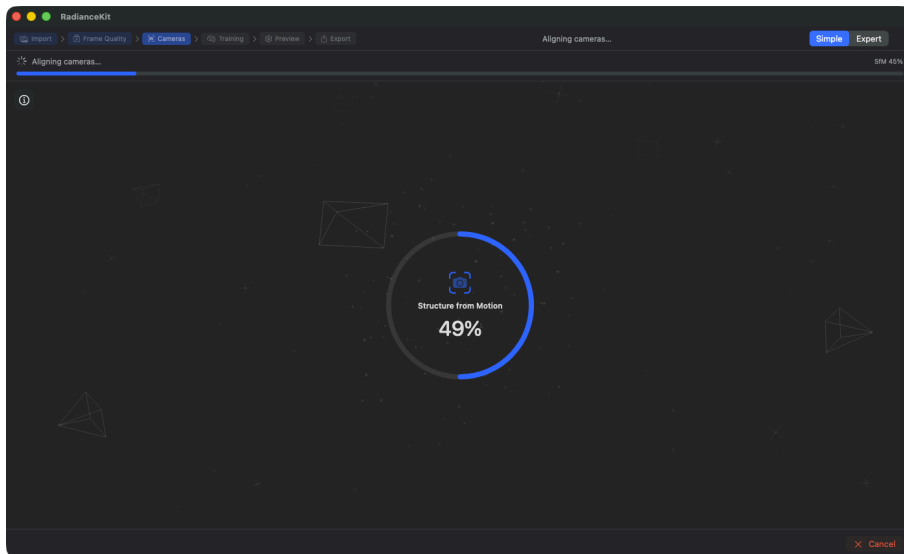
 **KORT FORKLART**

Hvis du allerede har arbeidet med Metashape, RealityCapture eller COLMAP og har latt kamera-beregningen kjøre der, kan du bare dra eksport-mappen inn her. RadianceKit gjenkjenner automatisk på innholdet at det er et COLMAP-workspace (det sjekker for `sparse/0/`, `cameras.bin` osv.) og spør deg om det skal overta den ferdige beregningen eller selv regne på nytt. Å overta sparer timer med venting på store scener fordi SfM hoppes helt over — treningen starter straks. Også tekst-format-workspaces (`cameras.txt`) og ETH3D-eksporter gjenkjennes. Denne funksjonen er tilgjengelig i begynnermodus som i Expert Mode; flere detaljer står i kapittel 9 under backend Q6.

Når til neste trinn?

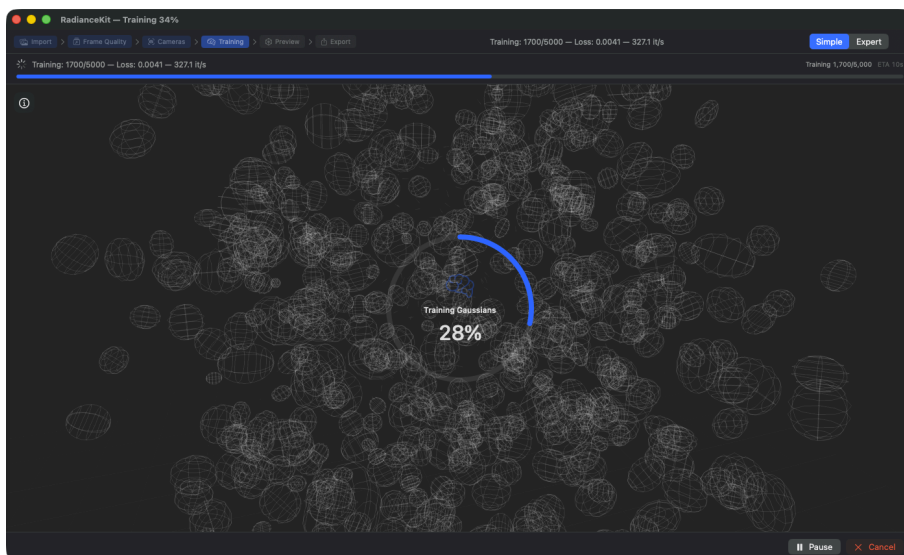
Du kan klikke Start Processing så snart (a) minst ett bilde eller en video er importert, og (b) validation-banneret er oransje eller forsvunnet. Ved rødt banner lar appen deg likevel starte, men du kan med stor sannsynlighet avbryte behandlingen igjen straks. Anbefales: minst 20 bilder, skarpe, med tydelig overlapp mellom påfølgende opptak, alle fra omtrent samme distanse til motivet. Velg før start en preset som passer tidsbudsjettet ditt — ved 30 bilder og Quick-preset er du gjennom på få minutter, ved Quality varer det heller 1–2 timer.

Z2 — Behandling (SfM + trening)



Figur 36: Z2 SfM-fase — stage-ikon „Structure from Motion« med 41 % i den store sirkelen, øvre statusbjelke ved „SfM 25 %«, Cancel- knapp nederst til høyre

SfM-fase (kameraer justeres): Stor fremgangssirkel viser sub-stage-fremgang (her 41 % av den kjørende Apple- Photogrammetry-sesjonen). Statustekst „Aligning cameras...« øverst til venstre. Crumb-trail markerer „Cameras« som aktivt trinn. Øvre statusbjelke viser pipeline-samlet-fremgang (25 %) — SfM opptar første halvdel av bjelken. Svevende wireframe- kameraer i bakgrunnen antyder at posisjoner estimeres.



Figur 37: Z2 treningsfase — 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

Treningsfase (gaussians optimeres): Sub-stage-ikon skifter til „Training Gaussians«, prosent teller iterasjoner ut fra valgt preset (her 400 / 5 000 for Preview-preset = 8 % av stagen). Live-metrikk-linjen viser loss-verdi (0.1642), iterasjoner per sekund (138.7 it/s) og ETA (33 s). Pipeline-samlet-fremgang klatrer fra 50 % til 100 % under denne

fasen. Pause-knappen (i stedet for Cancel-only i SfM-fase) tillater resume senere; Cancel kaster treningsresultatet og vender tilbake til Z1.

Så snart pipelinen kjører, fader appen import-overlayet ut og viser en fullskjerms behandlingsskjerm. I midten kjører en stor fremgangssirkel (220 × 220 piksel) med stage-ikon, status tekst og prosent-tall; i bakgrunnen visualiserer en diskret splat-animasjon symbolsk den kjørende beregningen. Øverst til venstre kan et info-panel slås inn som viser live-metrikker fra trening og SfM. Nederst er det pause/resume, Cancel og i feiltilfeller en retry-knapp.

C-18 SplatTrainingView (bakgrunnsanimasjon)

HVOR

Fullskjerms bakgrunn bak progress-sirkelen, fader ut ved avbrytelse eller feil.

TEKNISK

Dekorativ animasjon som avhengig av pipeline-fremgangen (0...1) rendrer et stigende antall små animerte splat-partikler. Kilden er en beregnet fremgangsverdi som mapper SfM-faser til 0–0.2 og trening til 0.2–1.0 (frame-quality til 0–0.05). Slik „bygger« splats seg synlig opp mens treningen kjører. Utelukkende dekorativt — visningen viser ingen ekte mellomresultater av den aktuelle treningen (det ville være live-preview i Expert Mode). Ved avbrytelse eller feil fader den ut, og bare statussirkelen forblir synlig.

KORT FORKLART

I bakgrunnen kjører en liten animasjon av dansende punkter slik at skjermen ikke virker så tom under beregningen. Det er ikke den ekte 3D-modellen din — den ser du først etter treningen i trinn Z3. Animasjonen har imidlertid samme tonalitet, slik at du ut fra det omtrentlige fortetningsnivået kan lese hvor langt treningen er kommet. I starten sees bare få punkter, mot slutten fylles bakgrunnen tydelig tettere — en søt visuell indikator i tillegg til prosent-visningen i sirkelen. Hvis animasjonen plager deg (f.eks. fordi du vil jobbe ved siden av i bakgrunnen), kan du bytte til Expert Mode hvor den faller bort.

C-19 Stor progress-sirkel

I midten av behandlingsskjermen, 220 × 220 piksel.



To over hverandre rendrede ringer: utvendig en dempet track-ring, innvendig en utfylt fremgangsring med accent- eller rød-stroke (rød ved feil). Inne i sirkelen et stage-ikon (hjerne for trening, kamera for SfM, film for video-frame-ekstraksjon, sparkles for frame-quality), stage-tittel og det live-animerte prosent-tallet i 32-punkts-rounded-font. Ikonet pulserer mykt så lenge behandlingen er aktiv. Visningen interpolerer på en 30-Hz-timer mykt mot den aktuelle ekte fremgangen — med konstant kryp (0.0003/frame) pluss proporsjonal andel (4 % av gapet) og et soft-ceiling som settes til 80 % av neste forventede milestone (for SfM fra en hardkodet milestone- tabell). Slik virker fremgangen flytende selv om de ekte SfM-updates bare ankommer hvert par sekunder.

 **KORT FORKLART**

Den store sirkelen i midten er hovedvisningen din mens appen regner. Den fylles mykt selv når de ekte beregningsopdatene bare kommer hvert par sekunder — det gir deg følelsen av at det skjer noe, i stedet for å stirre minutter på en fastfrossen prosent. Symbolet i midten skifter avhengig av om det akkurat nå ekstraheres frames (film-ikon), kameraer justeres (kamera-ikon) eller gaussians trenes (hjerne-ikon). Prosent-tallet refererer til det aktuelle deltrinnet — den samlede pipelinen ser du i den smale bjelken helt oppe. Ved en feil farges ringen rød i stedet for blå, og ikonet pulserer ikke lenger, slik at du straks merker at noe har gått galt.

C-22 Info-knapp (vis metrikker)

Øverst til venstre på behandlingsskjermen, 32 × 32 piksel.



Enkel knapp med material-bakgrunn. Slår info-panelet på eller av. Ikonet skifter mellom info-sirkel-outline og info-sirkel-fylt når aktiv. Myk fade-in-animasjon. Tooltip „Show detailed processing metrics«.

 **KORT FORKLART**

Som standard er skjermen bevisst ryddig — bare den store fremgangssirkelen, mer ser du ikke i første omgang. Hvis du som teknisk interessert bruker vil vite mer nøyaktig hva som skjer (hvilken iterasjon, hvor høy loss, hvor mange gaussians), så klikk på i-symbolet øverst til venstre. Et lite panel folder ut nederst og viser alle live-verdier. Enda et klikk fader det ut igjen. Innstillingen er ikke persistent — ved hvert nytt treningsforløp er panelet først skjult igjen, noe som bevisst er slik for ikke å skremme begynnere.

C-23 Info-panel (live-metrikker)**HVOR**

Nederst til venstre på behandlingsskjermen, bare synlig hvis `showProcessingInfo == true`.

**TEKNISK**

Tospaltet panel med ultra-thin-material-bakgrunn. Venstre kolonne: stage-spesifikke info-linjer — for SfM statustekst og prosent; for trening iterasjon, kombinert loss, L1-loss, D-SSIM-loss, gaussian-count (oransje farget), speed (it/s), elapsed-time, beregnet ETA, SH-degree og learning-rate. Høyre kolonne: statustekst, time-info-string, inline loss-chart (se C-28) og et discoverability-nudge (se C-32). Alle verdier leses fra treningsstatus som oppdateres ved hver treningstikk.

KORT FORKLART

Info-panelet viser alle live-verdier som i Expert Mode ville stå permanent i Inspector-sidefeltet: aktuell iterasjon, loss-verdi (mindre = bedre), antall gaussians, hastighet, estimert resttid, SH-degree og learning-rate. På høyre side kjører dessuten en liten loss-kurve med, som med ett blick forteller deg om treningen kjører i riktig retning. Hvis treningen virker treg, hjelper et blick her — en loss som ikke lenger faller, eller en ETA som ikke lenger synker, indikerer problemer. Hvis loss-en eksploderer (plutselig blir gigantisk) eller viser NaN, har treningen blitt ustabil, og en Cancel + Retry eller bytte til en annen preset er meningsfullt.

C-25 Pause/Resume-knapp**HVOR**

Nedre navigasjonslinje, bare synlig under treningsstagen (IKKE under SfM), og så lenge behandlingen kjører.

**TEKNISK**

Bordered knapp. Kaller avhengig av status pause eller resume. Label skifter mellom „Pause« (med pause-ikon) og „Resume« (play-ikon). Under SfM-trinnet vises knappen ikke fordi Apple Photogrammetry ikke kjenner pause-semantikk. Pause-tilstanden bevarer iterasjon, gaussian-status og optimizer-momentum fullstendig — resume fortsetter der det ble paused.

KORT FORKLART

Mens treningen kjører, kan du når som helst holde den tilbake og fortsette senere. Meningsfullt hvis du i mellomtiden vil gjøre noe annet på Macen som krever mye GPU — f.eks. video-redigering, spilltest eller en render-eksport fra en annen app. Klikk Pause, gjør tingen din, klikk Resume, treningen kjører videre nøyaktig der den var. Iterasjons-teller, gaussian-antall og optimizer-momentum bevares fullstendig, pause-state koster deg ingenting i kvalitet. Under SfM-fasen er pause ikke tilgjengelig — Apple Photogrammetry kjenner ikke til en pause-funksjon, der må du i nødstilfeller arbeide med Cancel.

C-26 Cancel-knapp

Nedre navigasjonslinje, synlig mens behandlingen kjører (SfM eller trening).



Rød bordered-knapp. Åpner en bekreftelsesdialog med tittel „Stop and discard progress?«, knapper „Discard Progress« (destruktiv) og „Keep Running« (Cancel). Ved bekreftelse settes cancel-flagget, treningstasken avsluttes, SfM-underprosessen om nødvendig avsluttes, og en summary-linje med avbryt-status skrives i JSONL-loggen. I motsetning til pause kastes treningsbuffer og status.

KORT FORKLART

Avbrytknappen. I motsetning til pause er det endelig — hvis du deretter vil starte på nytt, kjører behandlingen forfra, alle allerede trente iterasjoner er tapt. Meningsfullt hvis du har valgt feil preset, treningen kjører altfor sakte, eller appen åpenbart produserer skitt-resultater og du ikke vil vente. Før den faktiske avbrytelsen spør appen i en bekreftelsesdialog én gang til slik at du ikke ved et uhell mister timer med regnetid. Hvis du bare vil avbryte kort, så ta heller pause.

C-27 Retry-knapp

Nedre navigasjonslinje, synlig hvis pipelinen har feilet (SfM-status starter med „SfM failed«, eller treningen er i feiltilstand).



Accent-knapp. Starter hele pipelinen på nytt. Før start sjekkes om det fortsatt finnes importerte bilder/videoer. Tidligere feil-logger bevares i JSONL-mappen; en ny run skriver en ny log-fil med aktuell timestamp.

KORT FORKLART

Hvis SfM eller trening avbryter med en feilmelding, kan du prøve igjen her. Noen ganger hjelper det fordi mange trinn (RANSAC, densification) har tilfeldighetsdeler, og et annet forsøk kan lykkes der det første mislyktes. Hele pipelinen kjører da igjen forfra — SfM og trening, i en frisk JSONL-log-fil. Hvis også andre forsøk feiler, er input-bildene som regel problemet (for få, for lite overlapp, bevegelses-uskarphet, dårlig lys); gå da tilbake med Back og bytt ut materialet ditt. Tips: kikk samtidig i training-loggene (Help → Open Training Logs), der står det mer detaljert hvor det konkret gikk galt.

C-28 Inline Loss Chart **HVOR**

I info-panelet, høyre kolonne, bare synlig under trening med ikke-tom forløpshistorikk.

 **TEKNISK**

Kompakt tegne-område (40 piksel høyt), tegner loss-history som 1-piksel-linje i accent-farge. Data filtreres til finite-verdier (NaN-beskyttelse for ustabile treninger). Min/max beregnes over hele historikken — charten auto-zoomer altså på verdiområdet. Den siste loss-verdien står øverst til høyre over charten. Selve historikken bygges opp i app-tilstanden ved hvert treningstikk (typisk hver 100 iterasjoner).

 **KORT FORKLART**

En liten loss-kurve som med ett blikk viser deg om treningen „konvergerer« (linjen faller mot høyre), eller om den henger fast eller eksploderer (linjen flat eller stigende). Ved en sunn trening faller linjen bratt i starten og flater så — det er det forventede forløpet, lignende en halveringskurve. Charten zoomer automatisk på det aktuelle verdiområdet slik at også små forbedringer i slutten av treningen forblir synlige. Hvis linjen plutselig skyter oppover eller fryser, er det et godt signal om at noe går galt — enten er materialet problematisk, eller en annen preset ville være bedre egnet. Charten finner du i info-panelet som du folder ut øverst til venstre med i-symbolet.

C-32 Discoverability Nudge (Expert-Mode-hint) **HVOR**

I info-panelet, høyre kolonne nederst, bare synlig under trening OG i begynnermodus.

 **TEKNISK**

Liten linje med øyne-ikon og caption-tekst „Switch to Expert Mode (⌘2) for live splat preview«, i tilbakeholden tone og 10-punkts-skrift. Intet interaktivt element, bare hint. Reagerer ikke på klikk — brukeren må faktisk trykke Cmd+2 eller klikke menyen Mode → Expert Mode.

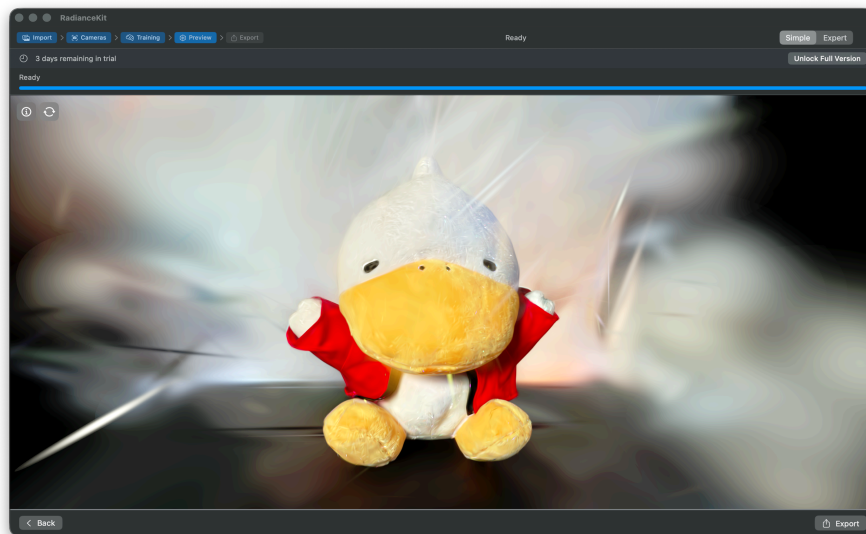
 **KORT FORKLART**

Et diskret hint om at i Expert Mode er den aktuelle mellomversjonen av 3D-modellen din live å se i viewporten under treningen. I begynnermodus er det bevisst skjult for å holde brukergrensesnittet rolig — men mange brukere vet ikke at funksjonen finnes, så vi peker mildt på det her. Trykk Cmd+2, og treningen kjører videre i bakgrunnen mens du kan se på hvordan modellen din bygger seg opp for øynene dine. Det er også et godt verktøy for å allerede etter få tusen iterasjoner vurdere om resultatet blir til noe, eller om du heller avbryter og starter på nytt. Cmd+1 bringer deg når som helst tilbake til begynner-visningen.

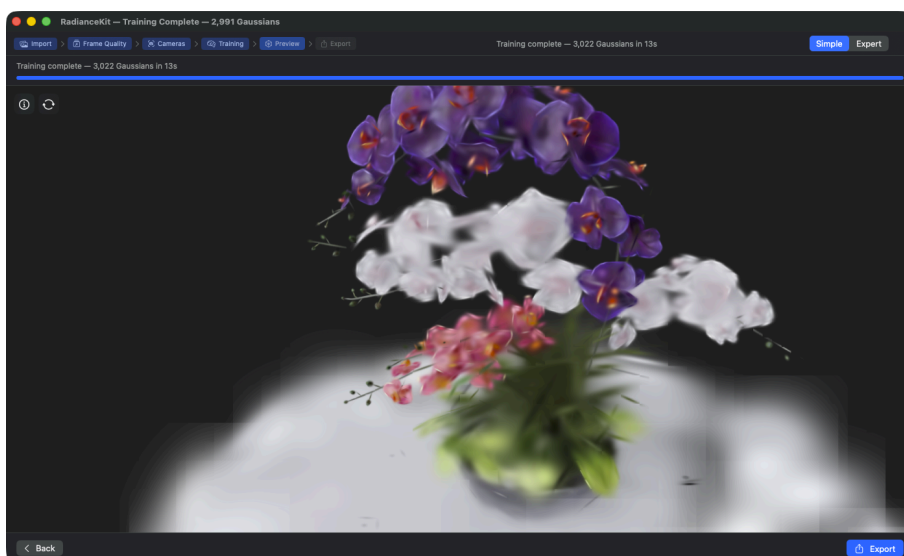
Når til neste trinn?

Appen bytter automatisk til Z3 (forhåndsvisning) så snart treningen er avsluttet med suksess — du trenger ikke klikke. Den nedre navigasjonslinjen bytter da fra Pause/Cancel til en Back-knapp (tilbake til import) og en Export-knapp (frem til eksport). I feiltilfeller (rød feilmelding, stage-ikon er X) dukker Retry opp i stedet, og du må bestemme om du starter igjen eller går tilbake til import med Back for å endre bildemateriale.

Z3 — Forhåndsvisning (drei 3D-modellen)



Figur 38: Simple Mode forhåndsvisnings-trinn med 3D-viewer



Figur 39: Z3 forhåndsvisning etter treningsavslutning — Blender-buketten rekonstruert, header viser „Training complete — 3,022 Gaussians in 13s«, Back- og Export-knapper nederst

HVA BILDET VISER Crumb-trail markerer „Preview« som aktivt trinn. Fullskjerms-3D-viewport rendrer den ferdig trente bukett-scenen (syntetisk Blender-testsett, 60-frame-subset fra 960 hemisfæriske cams). Header-statusbjelke: „Training complete — 3 022 Gaussians in 13 s» — gir endelig gaussian-antall og treningstid. Drag i viewporten roterer kameraet (yaw/pitch); scroll-wheel zoomer langs view-direction. „Back«-knapp (nederst til venstre) vender tilbake til Z2 for resume eller re-run; „Export«-knapp (nederst til høyre, primary) navigerer videre til Z4.

Etter avslutning av treningen lander appen automatisk i forhåndsvisningen. Her ser du den ferdige Gaussian-splattung- modellen din i en fullskjerms-metal-visning og kan dreie, zoome og panorere den med mus og trackpad. På oversiden av viewporten ligger en liten overlay med kamerastyring og info — auto- rotasjon, treningsstatistikk, reset-knapp. Før neste trinn (eksport) er det verdt å sjekke modellen fra forskjellige vinkler for å sikre at rekonstruksjonen er ren.

C-36 SplatViewportView (3D-hovedvisning)

HVOR

Fullskjermsbakgrunn av forhåndsvisnings-trinnet.

TEKNISK

Metal-basert 3D-viewport som rendrer den ferdige punktskyen. Rendereren er RadianceKits EGEN ForwardPass-rasterizer — den samme som allerede rendrer splattene under treningen — så det er ekte WYSIWYG (det som trenes, vises og eksporteres nøyaktig likt). Tile-basert rendering-pipeline med order-independent transparency. Hvis rendereren ikke kan initialiseres (f.eks. fordi Metal ikke er tilgjengelig på systemet), dukker det i stedet opp en sort bakgrunn med „Metal not available«-tekst. Visningen ignorerer safe-area slik at modellen rekker til vinduets kant.

KORT FORKLART

Hovedviewporten. Her ser du den ferdige 3D-modellen rekonstruert av fotoene dine, rendret på GPU-en i sanntid. Klikk og dra med venstre museknapp for å rotere. Scrollhjul eller trackpad-gestus med to fingre for å zoome. Høyre museknapp eller Cmd+dra for å panorere. Modellen består av titusenvis av semi-transparente 3D-ellipsoider („gaussians«), som rekonstruerer scenen din fotorealitisk — hver enkelt har en posisjon, retning, form og farge som treningen har lært. I det sjeldne tilfellet at Mac-en din ikke støtter Metal, ser du i stedet en sort bakgrunn med en hint-melding — RadianceKit krever tvingende en Metal-istandsatt GPU.

C-37 CameraControlsOverlay (styringsoverlay)**HVOR**

Over viewporten, svevende.

**TEKNISK**

Kompakt UI-overlay med knapper for auto-rotasjon (turntable), reset-camera, bakgrunnsvalg (Gray/Black/White), save-screenshot, toggle-info-panel. Binder til kamera-parametrene (distanse, azimut, elevasjon, target, FOV) og styrer auto-turntable. Under treningen (når brukeren i Expert Mode vil se viewporten kjøre med) viser overlayet dessuten en kompakt treningsstatuslinje.

KORT FORKLART

Den lille svevelinjen over modellen. Her starter du auto-rotasjonen (modellen dreier av seg selv, bra til skjermbilder og korte demoer), nullstiller kameraet via reset til startposisjonen (hvis du har vandret bort), bytter bakgrunn (grå for nøytral, sort for maksimal kontrast, hvit for lyse modeller) og lager skjermbilder direkte, som lagres under /Pictures. Praktisk hvis du vil vise en bestemt detalj fra en helt bestemt vinkel uten å måtte eksportere hele modellen. Auto-rotasjonen er også en god test for om modellen ser like god ut fra alle sider, eller om det er en „skitten side« som har oppstått av manglende opptak.

C-38 Export-knapp (navigasjonslinje)**HVOR**

Nedre navigasjonslinje i Z3.

**TEKNISK**

Accent-knapp med label „Export« og share-ikon. Klikk utløser bytte til Z4. Først sjekker den overordnede visningen om fullversjonen er låst opp — hvis ikke, vises i stedet for eksport-stagen låsevisningen (se U-06).

KORT FORKLART

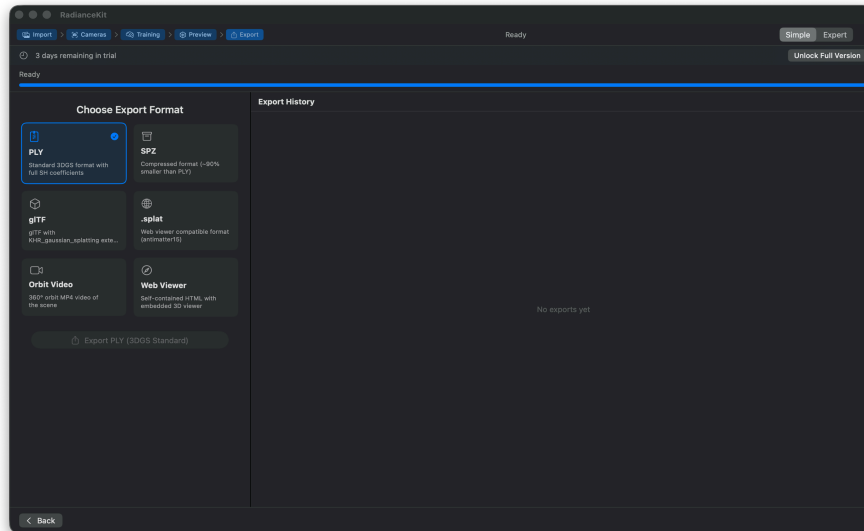
Hvis du er fornøyd med resultatet, så klikk Export, og du lander i siste trinn hvor du velger format og lagrer. Uten kjøpt fullversjon lander du i stedet på en skjerm-låsing med unlock-hint og kjøps-knapp — appen vil ikke påtvinge deg en fullversjon, men eksport er en av premium-funksjonene. Så snart du har avsluttet kjøpet, kjører appen direkte videre i låst-opp-tilstand, og du lander i den vante eksport-stagen. Hvis du ombestemmer deg, kommer du via Back-knappen tilbake til forhåndsvisningen og kan dreie videre på modellen.

Når til neste trinn?

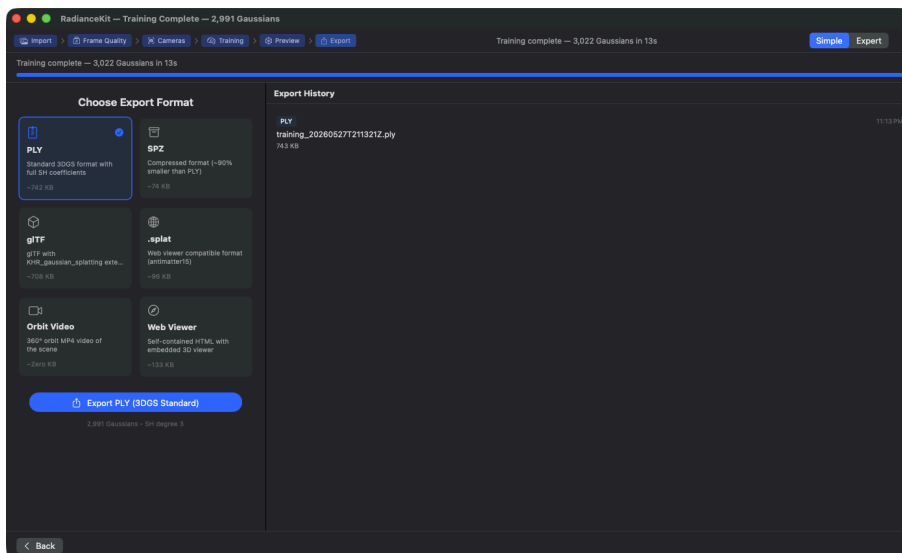
Før du eksporterer, dreie modellen en gang helt rundt og sjekk: er alle områder du har dekket i input-bildene dine til stede? Finnes det svevende „floaters« (fritt svevende

gauss-splat-skyer i luften)? Virker bakgrunnen/himmelen ren eller utvasket? Alvorlige problemer kan bare fikses ved nytrening — enten med flere bilder, annen preset eller i Expert Mode med floater- reduction-settings.

Z4 — Eksport (velg format & lagre)



Figur 40: Simple Mode eksport-trinn med format-kort



Figur 41: Z4 eksport-kort — 6 formater (PLY 742 KB valgt, SPZ 74 KB, glTF 708 KB, .splat 96 KB, Orbit Video, Web Viewer 133 KB), Export-History-sidfelt til høyre med allerede eksportert PLY

HVA BILDET VISER Crumb-trail markerer „Export« som aktivt trinn. Venstre kort-grid „Choose Export Format« med alle seks alternativer: PLY (standard-3DGS, 742 KB, med fulle SH- koeffisienter — her forhåndsvalgt med blå hake), SPZ (komprimert 3DGS-format, ~90 % mindre enn PLY, 74 KB), glTF (med KHR_gaussian_splatting-extension, 708

KB), .splat (web-viewer-kompatibel via antimatter15, 96 KB), Orbit Video (360°-MP4 av scenen, live-størrelsesberegning), Web Viewer (selvstendig HTML med innebygd 3D-viewer, 133 KB). Størrelsesangivelser beregnes live ut fra aktuelt gaussian-count og format-overhead. Til høyre „Export History« lister allerede avsluttede eksporter med format-pille, filnavn og timestamp — klikk avslører i Finder. Primary-CTA nederst til venstre: „Export PLY (3DGS Standard)» med gaussian-undertittel „2,991 Gaussians · SH degree 3”.

I siste trinn velger du blant 6 eksport-formater (PLY, SPZ, glTF, .splat, orbit-video, web-viewer) via 2-spaltet kort-grid, klikker Export og velger lagringssted i macOS-dialogen. Til høyre kjører en history over alle hittil værende eksporter — ved kort-valget vises straks under hvert kort den estimerte filstørrelsen, slik at du f.eks. foretrekker SPZ hvis du vil på nettet (lite), og PLY hvis du vil importere til en annen programvare (SuperSplat, Postshot, Blender via plugin) (stor og fullstendig).

C-39 2-Column Format Grid

HVOR

Venstre hovedside av eksport-trinnet.

TEKNISK

Kort-raster med to fleksible kolonner og 12 punkt avstand. Itererer over formatene som tilbys i begynnermodus — en filtrert delmengde av den fulle formatlisten som bare inneholder de 6 viktigste formatene: PLY, SPZ, glTF, .splat, orbit-video, web-viewer. Compressed-PLY og SOG tilbys KUN i Expert Mode.

KORT FORKLART

Et kort-raster med de 6 formatene som er relevante i begynnermodus: PLY (standardformat for andre 3D-verktøy), SPZ (komprimert variant for web), glTF (offisiell web3D-standard), .splat (for antimatter15-web-viewer), orbit-video (ferdig MP4 for å vise frem) og Web Viewer (selvstendig HTML-fil med innebygd 3D-avspiller). Dermed dekker du 90 % av anvendelsestilfellene. Hvis du trenger et av de mindre gjengse formatene (Compressed-PLY eller SOG for ekstrem komprimering), så bytt til Expert Mode, der er alle 8 formater tilgjengelige. Det kompakte valget her er bevisst slik at begynnere ikke overveldes av mangfoldigheten.

C-40 Format Card Button **HVOR**

Hvert kort i gridet.

 **TEKNISK**

Enkel knapp med kort-layout: ikon (f.eks. dokument-zipper for PLY, arkivboks for SPZ, video-ikon for orbit-video) øverst, format-navn som headline, beskrivelses- caption (2-linjes avkortet), under den estimerte filstørrelsen (live beregnet ut fra format, gaussian-count og SH-degree og formatert som KB/MB). Ved klikk velges formatet. Det valgte kortet får accent-bakgrunn, accent-border og et hake-ikon øverst til høyre. Tooltip er format-beskrivelsen.

 **KORT FORKLART**

Ett kort per format. Klikk ett, det fremheves med accent-farge og en hake, og export-knappen under tilpasser teksten sin („Export PLY«, „Export SPZ» osv.). Hvert kort viser et passende symbol, navnet, en tolinjet kortforklaring og den estimerte filstørrelsen ved ditt aktuelle treningsresultat. Størrelsen hjelper deg med å velge fornuftig — hvis du vil sende resultatet per e-post, så ta den minste varianten (som regel SPZ eller .splat); hvis du vil arbeide videre i en annen 3D-programvare, så ta den med best kompatibilitet (typisk PLY). Ved hover over et kort viser tooltipet en mer utførlig beskrivelse hvis du finner forskjellene mellom formatene uklare.

C-41 Video Duration-slider **HVOR**

Under format-gridet, bare synlig hvis et video-format er valgt (orbit-video eller social-video).

 **TEKNISK**

Glidebryter 3–30 sekunder i 1-sekunds-skritt, binder til videolengden i app-tilstanden. Maks-bredde 300 piksel. Viser bare hvis et video-format er valgt. Ved ikke-video-formater fjernes glidebryteren helt fra visningen — ingen død plass.

 **KORT FORKLART**

Hvis du velger en orbit-video som eksport, kan du her bestemme lengden. 3 sekunder = veldig rask dreining, 30 sekunder = sakte, rolig dreining om modellen din. For social-media-reels (Instagram, TikTok) er som regel 6–10 sekunder ideelt — lenge nok til å vise modellen, kort nok til at serne ikke hopper av. Ved presentasjoner eller portefølje-videoer kan du gjerne ta 15–20 sekunder. Slidern dukker bare opp hvis et video-format er valgt; ved fil-formater som PLY eller SPZ ville den være meningsløs og er skjult.

C-42 Export-knapp HVOR

Under format-griddet (og under duration-slideren hvis video valgt).

 TEKNISK

Stor accent-knapp. Label: „Export {format-navn}«, share-ikon. Ved klikk åpnes macOS-lagre-dialogen med format- passende endelse og default-filnavn „scene.{ext}«; ved bekreftelse skrives eksporten til den valgte URL-en. Deaktivert hvis intet treningsresultat finnes, eller en eksport allerede kjører.

 KORT FORKLART

Klikk, velg lagringssted i macOS-dialogen, ferdig — appen skriver filen i det valgte formatet på det valgte stedet. Default-navn er „scene.{endelse}« (f.eks. „scene.ply« eller „scene.spz«), den kan du endre vilkårlig i dialogen før du lagrer. Knappen er grå så lenge intet treningsresultat foreligger (skulle aldri skje her fordi du ellers slett ikke ville være i eksport-trinnet), eller en annen eksport allerede kjører. Så snart eksporten kjører, dukker det opp en fremgangsvisning under; appen forblir betjenbar slik at du kan allerede forberede neste eksport.

C-43 Export Progress Bar HVOR

Under export-knappen, bare synlig mens en eksport kjører.

 TEKNISK

Fremgangsvisning med maks-bredde 300 piksel, herunder caption „Exporting... N %«. Verdien går fra 0 til 1 og oppdateres under skrivingen — ved PLY i chunks på 10 000 gaussians, ved SPZ engang etter kvantisering, ved orbit-video i frame-intervaller.

 KORT FORKLART

Mens eksporten kjører, ser du her fremgangen som smal bjelke pluss prosent-visning. PLY er som regel ferdig i løpet av sekunder fordi filen rett og slett skrives bort binært. SPZ tar litt lengre fordi dataene underveis kvantiseres og komprimeres. Orbit-video er den mest tidkrevende eksporten — her rendres hver enkelt frame på nytt; avhengig av oppløsning og lengde kan det vare et minutt eller lengre. Under eksporten forblir appen betjenbar slik at du allerede kan forberede neste format eller klikke videre i viewporten.

C-44 Export Error Display **HVOR**

Under progress-baren, bare synlig hvis det oppsto en feil ved siste eksport.

 **TEKNISK**

Rød linje med warning-ikon og feiltekst. Rød 8 %-bakgrunns-opacity, avrundede hjørner. Maks-bredde 400 piksel. Hyppige feilårsaker: SOG forventer `cwebp` i system-PATH (ikke App Store-konform); skrivefeil ved full diskplass; sandbox-feil ved lagringsmål utenfor det tillatte området.

 **KORT FORKLART**

Hvis eksporten går galt, dukker det her opp i rødt en kort klar-tekst-beskrivelse av problemet. Som regel er årsaken innlysende — ingen plass på disken, ingen skriverettigheter til målappen eller et mål utenfor sandbox-tillatte områder. Spesielt ved SOG-formatet skjer det at `cwebp` mangler i systemet; i så fall kan SOG ikke brukes, og du må vike til SPZ. Hvis feilmeldingen er uklar, så kikk i log-mappen (Help → Open Training Logs), der står det mer utførlig hva som gikk galt. I tvil hjelper det å velge et annet lagringssted — f.eks. skrivebordet.

C-46 Export History List **HVOR**

Høyre side av eksport-trinnet.

 **TEKNISK**

Liste over eksport-historikken (lagret persistent som JSON i UserDefaults, pleid etter hver vellykket eksport). Hver linje viser format-badge (lite, accent-farget), timestamp (HH:mm), filnavn (1 linje avkortet) og formatert filstørrelse. Klikk på en linje åpner Finder med valgt fil. Empty-state: „No exports yet«.

 **KORT FORKLART**

En liste over eksportene dine hittil — format, klokkeslett, filnavn, størrelse, i kronologisk rekkefølge. Klikk på en linje, og filen fremheves i Finder uten at du selv må navigere gjennom mapper. Praktisk hvis du en time senere igjen trenger den siste eksporten og ikke lenger vet hvor du lagret den — historikken husker det. Hvis du aldri før har eksportert noe, står det her et vennlig hint „No exports yet«. Listen overlever omstarter av appen fordi den er lagret i UserDefaults.

C-48 History Context Menu (høyreklikk)



Høyreklikk på en history-linje.



Kontekstmeny på hver liste-oppføring med to handlinger: „Reveal in Finder« (åpner Finder med valgt fil som et vanlig klikk) og „Copy Path« (legger den fulle filstien som tekst i utklippstavlen). Sistnevnte er nyttig for dra-og-slipp i andre apper eller overlevering til kommandolinjen.

KORT FORKLART

Høyreklikk på en history-oppføring åpner en liten meny med to handlinger. „Reveal in Finder« gjør det samme som et vanlig klikk — åpner Finder med valgt fil slik at du straks ser den. „Copy Path« legger den komplette filstien i utklippstavlen slik at du kan lime den inn f.eks. i terminal-kommandoer, i andre apper eller i et notat. Spesielt praktisk hvis du vil gi eksporten videre til noen eller åpne den i et annet program som arbeider med sti-input. Funksjonelt en liten, men hjelpsom detalj som følger Mac-typiske betjeningsmønstre.

Når er arbeidsflyten avsluttet?

Etter en vellykket eksport har du 3D-modellen din som fil på disken, og historikken viser en ny oppføring. Det er ingen „Done«- knapp — du kan henge på vilkårlig mange eksporter i forskjellige formater uten å trene på nytt. Hvis du vil tilbake til forhåndsvisningen (f.eks. for igjen å sjekke et kamera-perspektiv), så bruk Back-knappen i den nedre navigasjonslinjen. Hvis du vil starte en helt ny scene, så gå via Back tilbake til Z1 og bruk der Clear All, eller File → New Project (Cmd+⇧+N).

Bytte til Expert Mode

Trykk når som helst Cmd+2 eller velg Mode → Expert Mode (M8). Hele tilstanden bevares: importerte bilder, valgt preset, kjørende eller ferdig trening, ferdig punktsky, eksport-historikk, til og med den aktuelle stagen. I Expert Mode vises i stedet for det firetrinns-stage hele Inspector-sidefeltet med alle ~150 betjeningsfelter. Spesielt: Project Navigator (se kapittel 2) tilbyr de utvidede bildeoperasjonene (minus-knapp, backspace-delete, Cmd-Z-undo, Quick-Look-forhåndsvisning), live-preview i viewporten under treningen samt alle loss-, MCMC-, densification- og Mip-splattung-parametre. Cmd+1 bytter tilbake til begynnermodus — også det taper ingen tilstand.

Ofte stilte spørsmål

Hvorfor forblir Start Processing-knappen min grå?

Du har ennå ikke importert bilder eller en video. Dra minst én fil inn i drop-zonen eller bruk „Browse Files«. Så snart bildelisten til høyre inneholder minst én oppføring, blir knappen aktiv. (Ved bare 1–2 bilder starter den riktignok, men SfM avbryter straks med feil — se det røde validation- banneret.)

Hvorfor er Export-knappen min låst?

I begynnermodus er det to trinn: (a) Hvis treningspipelinen ennå ikke er ferdig, og du ingen har, er knappen deaktivert — du må først avslutte Z2. (b) Hvis du ennå ikke har kjøpt fullversjonen (`PurchaseManager.hasAccess == false`), ser du i stedet for eksport-stagen en låsevisning med låseikon og „Unlock Full Version«-knapp som åpner purchase-sheetet. Quick- og Preview-presets tillater trening gratis, men eksport er premium.

Hvorfor kan jeg ikke velge en preset?

Du kan velge den — men hvis du trykker på en premium-preset (Balanced, Quality, MCMC-varianter) uten kjøpt fullversjon, hopper velgeren automatisk tilbake på Preview, og purchase-sheetet åpnes. Quick og Preview er de eneste gratis brukbare presetene.

Hvorfor er drop-zonen min tom og stiplet-grå selv om jeg drar bilder inn?

Sannsynligvis en UTI-type-mismatch. Appen aksepterer JPG, PNG, TIFF, HEIC, MP4, MOV pluss appens egne splat-formater. Andre bildeformater (BMP, GIF, WebP, RAW-formater) gjenkjennes IKKE. Hvis du er sikker på at bildetyperen din burde være der, så sjekk filnavn-endelsen — appen går primært etter extension, ikke etter fil-innhold.

Hvorfor varer SfM så lenge, selv om jeg bare har 30 bilder?

Apple Photogrammetry skalerer ikke lineært — ved visse bildekonstellasjoner (innendørs rom med komplekse teksturer, bevegelsesuskarphet, dårlig lys) tar det markant lengre enn bildecouant lar formode. Hvis SfM etter 10+ minutter ved 30 bilder fortsatt henger, så avbryt og prøv igjen med bedre materiale, eller bytt til Expert Mode og prøv COLMAP/Native- SfM (Cmd+2 → Inspector → Camera Alignment).

Hvor finner jeg training-loggene mine?

Help → Open Training Logs (Cmd+⇧+L). Det åpner `~/Documents/RadianceKit/Logs/`. Hver treningssesjon skriver en egen JSONL-fil med timestamp i filnavnet — første linje er konfigurasjonen, deretter følger en progress-linje hver 100 iterasjoner, siste linje er summary med final-loss og success-flag.



KOLOFON

*Satt 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