

UNIVERSITEIT HASSELT

MASTERPROEF VOORGEDRAGEN TOT HET BEHALEN VAN DE
GRAAD VAN MASTER IN DE INFORMATICA

Forensisch betrouwbare 3D-reconstructie met Gaussian Splatting

Auteur:

Jamie Withofs

Promotor:

Prof. dr. Lode Jorissen

Co-promotor:

Prof. dr. Nick Michiels

Begeleider(s):

De heer Joren Michels

Externe assessor:

dr. Jeroen Put

Academiejaar 2025-2026



Dankwoord

Graag wens ik mijn promotor, Prof. dr. Lode Jorissen, te bedanken voor zijn waardevolle begeleiding, kritische blik en constructieve feedback doorheen het gehele proces. Zijn expertise en inzichten hebben in belangrijke mate bijgedragen aan de kwaliteit en richting van dit werk.

Mijn dank gaat eveneens uit naar mijn begeleider, de heer Joren Michels, voor zijn voortdurende ondersteuning, constructieve adviezen en bereidheid om te helpen bij zowel inhoudelijke als praktische aspecten van het onderzoek. Zijn betrokkenheid en hulp waren onmisbaar bij het realiseren van deze masterproef.

Daarnaast wens ik de heer Jeroen Nelis, Project Lead XR Labs en Innovation Manager bij de Federale Politie Limburg, te bedanken voor zijn externe begeleiding, de mogelijkheid om dit onderzoek in een professionele en innovatieve context uit te voeren, en zijn waardevolle input vanuit de praktijk.

Bijzondere dank gaat ook uit naar Commissaris Marleen Claesen van de Federale Gerechtelijke Politie Limburg, voor haar bereidheid om haar expertise en praktijkervaring te delen tijdens het interview. Haar inzichten in de forensische werkwijze op de plaats delict waren een waardevolle aanvulling op de theoretische en technische aspecten van dit onderzoek.

Tot slot gaat mijn dank uit naar iedereen die op directe of indirecte wijze heeft bijgedragen aan de realisatie van dit werk. ¹

¹Dit werk is tot stand gekomen met ondersteuning van MAXVR-INFRA, een schaalbare en flexibele infrastructuur die de overgang naar digitale en fysieke werkomgevingen faciliteert. Dit veerkrachtfondsproject wordt gesubsidieerd door de Vlaamse Overheid en de Europese Unie.

Samenvatting

Deze masterproef onderzoekt de inzetbaarheid van Gaussian Splatting als betrouwbare techniek voor forensische 3D-reconstructie van misdadaascènes. Gaussian Splatting is een recente neurale weergavemethode die vanuit multi-view videobeelden fotorealistische 3D-modellen genereert via ellipsvormige volumetrische primitieven, zogenaamde splats, die samen een continue en vloeiende representatie van de scène vormen. De techniek onderscheidt zich van klassieke mesh-based methoden door haar uitzonderlijke visuele kwaliteit en efficiënte renderingpipeline, maar brengt tegelijkertijd een fundamenteel forensisch probleem met zich mee: tijdens de trainingsoptimalisatie heeft het model de neiging om geometrische informatie bij te verzinnen om gaten in de initialisatiedata op te vullen. Wanneer de reconstructie als juridisch bewijsmateriaal dient, is dergelijke hallucinatie onaanvaardbaar, omdat elke weergegeven vorm of afstand herleidbaar moet blijven tot authentieke, meetbare brondata.

Om dit probleem aan te pakken werd een volledige onderzoeks- en ontwikkelingspipeline opgezet in samenwerking met de Federale Gerechtelijke Politie Limburg. De dataverzameling combineerde videobeelden en meervoudige LiDAR-scans van een gecontroleerde testscène. De videobeelden werden verwerkt via een COLMAP-gebaseerde Structure from Motion (SfM) pipeline die de relatieve camera-poses en een sparse puntenwolk reconstrueerde. De LiDAR-scans werden samengebracht tot één dense referentiepuntenwolk via iteratieve ICP-gebaseerde registratie met outlier filtering op basis van IQR-analyse, radiale filtering en asgeoriënteerde filtering, en vervolgens geëxporteerd als een geïntegreerd .e57-bestand.

De kern van de bijdrage is een alignment pipeline die de sparse COLMAP-puntenwolk uitlijnt met de dense LiDAR-data. De resulterende transformatiematrix wordt toegepast op alle camera-extrinsieken, zodat de LiDAR-geometrie als initialisatiebasis kan dienen voor de Gaussian Splatting-training zonder de architectuur van de trainingsumgeving te wijzigen. Een eigen visibility-based downsampling-stap geeft daarbij prioriteit aan de LiDAR-punten die zichtbaar zijn vanuit de gecalibreerde camera-posities, wat de trainingsefficiëntie verhoogt zonder de geometrische volledigheid te compromitteren.

De Gaussian Splatting-training werd vervolgens uitgebreid met constraints die de geometrische trouw aan de LiDAR-meetdata waarborgen. In de volledig geconstrained variant worden zowel de positie-optimalisatie als de densification van Gaussians uitgeschakeld, waardoor elke splat exact op zijn initiële LiDAR-positie blijft staan. Dit levert een forensisch maximaal betrouwbaar model op, ten koste van visuele volledigheid in zones met beperkte LiDAR-dekking. Een uitgebreide parameterstudie onderzocht vervolgens tussenliggende configuraties, waarbij de interactie tussen de positie learning rate, de opacity learning rate, de opacity regularization en het MCMC-herplaatsingsmechanisme systematisch in kaart werd gebracht. De resultaten tonen aan dat een werkelijk compromis slechts beperkt haalbaar is: een gedeeltelijke vrijstelling van de positie-optimalisatie levert voor structurele elementen zoals wanden en vloer een aanvaardbare visuele kwaliteit op, maar biedt onvoldoende garantie voor dunne objecten met beperkte LiDAR-punt dichtheid. Op basis van de studie werden twee aanbevolen configuraties geformuleerd: één voor gebruik als visueel illustratiemateriaal voor rechters en jury's, en één als volledig geconstrained forensisch bewijsmiddel.

Aanvullend werd een forensische toolchain ontwikkeld die de integriteit van de reconstructie

kwantitatief en visueel onderbouwt. De Lichtfeld Studio-module berekent per Gaussian een betrouwbaarheidsscore op basis van de afstand tot zijn initiële LiDAR-positie, en visualiseert deze als een groen-geel-rood kleuroverlay. Een hallucination mask maakt een binaire splitsing tussen originele en nieuw aangemaakte splats, met kleurcodering voor de mate van positionele drift. De juridische validatiemodule voert een kwantitatieve alignmentvalidatie uit op basis van RMSE en overlappercentage, en genereert een integriteitsgeborgd JSON-rapport met SHA-256-hashing over alle relevante velden, een gedocumenteerde chain of custody en een onafhankelijk verifieerbaar RFC 3161-tijdstempel. Tot slot werd een Electron-gebaseerde rechtbankviewer ontwikkeld die dit rapport omzet naar een formeel PDF-deskundigenverslag.

De evaluatie bevestigt dat Gaussian Splatting via de ontwikkelde constrained pipeline forensisch inzetbaar is, mits een duidelijk onderscheid wordt gehanteerd tussen het gebruik als illustratiemateriaal en het gebruik als geometrisch bewijsmiddel. In het laatste geval fungeert de Gaussian Splatting-reconstructie als visuele toelichting bij de LiDAR-puntenwolk, die als primaire meetreferentie bewaard en ingediend dient te worden. De ontwikkelde toolchain biedt de forensische praktijk een transparant en controleerbaar raamwerk om dit onderscheid procedureel te verankeren.

Inhoudsopgave

1	Introductie	7
1.1	Context en relevantie	7
1.2	Probleemstelling	7
1.3	Methodologische benadering	8
1.4	Doelstellingen van het onderzoek	8
2	Related Work	9
2.1	3D-reconstructietechnieken uit videobeelden	9
2.1.1	Overzicht van klassieke methoden	9
2.1.2	Gaussian Splatting	12
2.1.3	Puntenwolk-initialisatie in combinatie met Structure from Motion	17
2.2	Gaussian Splatting vs Mesh-based reconstructie	18
2.2.1	Verschillen	18
2.2.2	Forensische toepasbaarheid	19
2.3	3D-reconstructies in forensisch onderzoek	21
2.4	Kwaliteits- en betrouwbaarheidscriteria	22
3	Methode	24
3.1	Overzicht van de experimentele pipeline	24
3.2	Data-acquisitie	26
3.2.1	Video-opname van de scène	26
3.2.2	LiDAR-scan van de scène	27
3.2.3	Consistentie tussen video- en LiDAR-data	28
3.3	Initiële reconstructie met Structure from Motion	28
3.3.1	Video preprocessing	28
3.3.2	COLMAP-pipeline	29
3.4	Alignment en transformatie pipeline	30
3.4.1	Aligneren van sparse puntenwolk met LiDAR	30
3.4.2	Transformatie van camera-extrinsieken	34
3.4.3	Conversie van LiDAR naar COLMAP-compatibel formaat	35
3.5	Hybride COLMAP reconstructie	39
3.6	Gaussian Splatting training	40
3.6.1	Baseline implementatie	40
3.6.2	Constrainede implementatie	41
3.7	Lichtfeld Studio	42
3.7.1	Beschikbaarheid, initialisatie en scèneschaal	42
3.7.2	Berekening van de betrouwbaarheidsscore	42
3.7.3	Selectie, groepsanalyse en visuele kleurcodering	46
3.7.4	Export van gehallucineerde en originele splats	47
3.8	Juridische validator	47
3.8.1	Validatiemodule	48
3.8.2	Chain of custody	49

3.8.3	Vertrouwde tijdstempel	49
3.8.4	Integratie in de validatie pipeline	49
3.9	Validatierapport-viewer	50
3.9.1	Architectuur: Electron als platform	50
3.9.2	Stap 1: Uploaden van het JSON-rapport	50
3.9.3	Stap 2: Invullen van operatorgegevens	51
3.9.4	Stap 3: Optionele tijdstempelverificatie	51
3.9.5	Stap 4: PDF-generatie	52
4	Evaluatie	55
4.1	Experimentele opzet	55
4.1.1	Dataset	55
4.1.2	Trainingsopzet	56
4.1.3	Evaluatiemetrieken	57
4.2	Grensgevallen	58
4.2.1	Baseline: geen beperkingen	58
4.2.2	Volledig geconstrained: geen positie-optimalisatie en geen densification	62
4.3	Parameterstudie	64
4.3.1	Invloed van opacity-parameters	65
4.4	Aanbevolen configuratie	75
4.4.1	Gebruik als illustratiemateriaal	75
4.4.2	Gebruik als forensisch bewijsmateriaal	75
4.4.3	Algemene aanbevelingen voor forensisch gebruik	76
5	Conclusies	77
5.1	Samenvatting van het onderzoek	77
5.2	Beantwoording van de onderzoeksvragen	77
5.2.1	Kan Gaussian Splatting worden verankerd aan LiDAR-geometrie?	77
5.2.2	Is er een bruikbaar compromis tussen visuele kwaliteit en forensische betrouwbaarheid?	77
5.2.3	Welke configuratie is aanbevolen voor forensisch gebruik?	78
5.3	Bijdragen van dit onderzoek	78
5.4	Beperkingen	79
5.5	Toekomstig onderzoek en maatschappelijke relevantie	79
5.6	Slotbeschouwing	80

Hoofdstuk 1

Introductie

1.1 Context en relevantie

De reconstructie van driedimensionale scènes uit videobeelden speelt steeds een belangrijkere rol in operationele en juridische contexten. Binnen forensisch onderzoek, voornamelijk bij het analyseren en presenteren van misdadescènes, is het cruciaal dat ruimtelijke informatie zowel nauwkeurig als betrouwbaar weergegeven wordt. Een correcte 3D-weergave kan bijdragen aan het beter begrijpen van gebeurtenissen, het ondersteunen van onderzoeksbeslissingen en het verduidelijken van bewijsvoering tijdens rechtszittingen.

Traditioneel werken politiediensten en instellingen zoals Europol steeds vaker met puntenwolken, verkregen uit fotogrammetrie, LiDAR-scanning of videobronnen. Hoewel deze puntenwolken meestal een hoge ruimtelijke nauwkeurigheid bieden, zijn ze minder intuïtief interpreteerbaar voor niet-technische gebruikers en missen ze vaak visuele samenhang. Vooral in juridische omgevingen, waar duidelijkheid en transparantie belangrijk zijn, vormt dit een grote beperking. [K. 20]

Deze spanning tussen technische nauwkeurigheid en juridische communiceerbaarheid is niet alleen een theoretisch probleem. In de praktijk worden 3D-reconstructies voornamelijk ingezet bij ongewone sterfgevallen, denk hierbij aan schietpartijen of verdachte ongevallen. Telkens opnieuw staan forensische experts voor de uitdaging om complexe ruimtelijke data begrijpelijk te maken voor rechters, jury's en advocaten die geen technische achtergrond hebben.

1.2 Probleemstelling

Recente vooruitgang in neurale representaties, en in het bijzonder Gaussian Splatting, biedt een manier om vanuit multi-view afbeeldingen een fotorealistisch en vloeiend 3D-model te genereren. Gaussian Splatting geeft een punt weer als een ellipsvormige vorm met specifieke parameters voor de positie, grootte, oriëntatie, kleur en dekking. Deze splats overlappen elkaar om een vloeiende, continue weergave van de 3D-scène te creëren zonder dat er een traditionele rendering-engine nodig is.

Echter, een belangrijke uitdaging bij het gebruik van neurale representaties in forensische toepassingen is dat deze modellen tijdens training de neiging kunnen hebben om informatie bij te verzinnen om gaten in de data op te vullen. Deze vorm van bijverzonnen informatie kan de visuele kwaliteit verhogen en de visuele samenhang optimaliseren, maar vormt een aanzienlijk risico wanneer de reconstructie als bewijsmateriaal moet dienen. In dergelijke contexten is absolute betrouwbaarheid vereist, elke pixel of vorm moet kunnen worden herleid tot authentieke data.

Daarom is het belangrijk om een methode te ontwikkelen die Gaussian Splatting toepast binnen de grenzen van wat daadwerkelijk waarneembaar is, zodat de gegenereerde 3D-modellen er niet

alleen realistisch uitzien, maar ook juridisch betrouwbaar en verdedigbaar blijven.

1.3 Methodologische benadering

De aanpak combineert computationele en empirische evaluatie. Eerst wordt een implementatie van Gaussian Splatting uitgebreid met constraints die de reconstructie beperken tot de puntenwolk. Vervolgens wordt deze methode getest op datasets die zelf opgenomen zijn, inclusief videomateriaal en 3D-scans.

De resultaten worden beoordeeld aan de hand van zowel kwantitatieve metrische nauwkeurigheid als kwalitatieve visuele evaluatie. Ten slotte wordt de praktische bruikbaarheid onderzocht in samenwerking met politiediensten, met het oog op mogelijke implementatie bij reconstructies van misdadsscènes.

1.4 Doelstellingen van het onderzoek

Deze masterproef heeft als hoofddoel het ontwikkelen en evalueren van een methode om neurale 3D-representaties betrouwbaar inzetbaar te maken binnen forensische reconstructies. Dit betekent dat de reconstructietechniek niet alleen visueel overtuigend moet zijn, maar ook juridisch verantwoord, waarbij elke gereconstrueerde informatie herleidbaar moet blijven tot meetbare en authentieke brondata.

Binnen dit brede doel worden de volgende specifieke doelstellingen geformuleerd:

- Het onderzoeken van bestaande Gaussian Splatting-technieken en hun beperkingen binnen nauwkeurighedsgevoelige toepassingen.
- Het ontwikkelen van een constrained training pipeline waarin de reconstructie trouw blijft aan de geometrie van de puntenwolk.
- Het evalueren van de nauwkeurigheid en betrouwbaarheid van de gegenereerde reconstructies in vergelijking met de oorspronkelijke meetdata.
- Het analyseren van de toepasbaarheid van deze methode binnen forensische en juridische contexten.
- Het vergelijken van neurale reconstructies met traditionele mesh-based methodes.

Hoofdstuk 2

Related Work

Dit hoofdstuk geeft een uitgebreid overzicht van de fundamentele concepten en huidige ontwikkelingen die relevant zijn voor dit onderzoek. We zullen de theoretische en praktische aspecten onderzoeken van de belangrijkste technologieën en methodologieën die aan de basis liggen van onze studie. Dit onderdeel start met een bespreking van verschillende 3D-reconstructietechnieken, hier zal ook dieper ingegaan worden op het onderwerp Gaussian Splatting en wordt besproken wat het effect is van een dense puntenwolk in combinatie met Structure From Motion te gebruiken. Het volgende gedeelte, 2.2, bespreekt in welke aspecten Gaussian Splatting en Mesh-based reconstructies van elkaar verschillen en bespreekt een eigen vergelijking op visueel vlak. Om beter inzicht te krijgen in de huidige toestand van 3D-reconstructies in forensisch onderzoek zal sectie 2.3 nagaan hoe 3D-modellen momenteel gebruikt worden in forensische analyse en training, en welke beperkingen deze op dit moment hebben. Tot slot worden kort de kwaliteits- en betrouwbaarheidscriteria besproken voor het gebruik van 3D-modellen binnen gerechtelijke toepassingen.

2.1 3D-reconstructietechnieken uit videobeelden

3D-reconstructies uit videobeelden zijn een belangrijk onderzoeksgebied binnen visual computing en spelen een groeiende rol in forensische toepassingen, waar betrouwbare ruimtelijke analyses van een omgeving cruciaal zijn. Dit hoofdstuk bespreekt de belangrijkste benaderingen die relevant zijn voor het reconstrueren van 3D-informatie uit videobeelden. Eerst volgt een overzicht van klassieke methoden, met nadruk op Structure from Motion en Multi-View Stereo. Daarna wordt Gaussian Splatting behandeld, de technische werking hiervan, toepassingen in realistische 3D-scan-omgevingen en de beperkingen voor forensische inzetbaarheid. Tot slot bespreken we de vergelijking met puntenwolk-initiatie, waarbij wordt ingegaan op het voordeel van het gebruik van een vooraf gereconstrueerde puntenwolk ten opzichte van klassieke SfM-pipelines (zoals COLMAP) als initialisatie op een input video.

2.1.1 Overzicht van klassieke methoden

Structure from Motion

Structure from Motion (SfM) is een techniek waarbij de 3D-structuur van een omgeving wordt gereconstrueerd aan de hand van een set 2D-images die van verschillende standpunten genomen zijn. Naast enkel de structuur worden ook de posities en oriëntaties van de camera's die de beelden hebben gemaakt geschat. In de praktijk wordt dit onder andere gebruikt binnen robotica en Augmented Reality, maar ook binnen archeologie vindt SfM toepassingen. Volgens [Joh16] bestaat de SfM-pipeline uit verschillende stappen: Image Acquisition, Feature Detection, Feature Matching, Camera Pose Estimation, Triangulation, Bundle Adjustment en Texturing And Model Generation.

In de Image Acquisition stap worden meerdere overlappende beelden van dezelfde omgeving vanuit verschillende standpunten verzameld. Dit wordt meestal gedaan door de verschillende frames uit een video te halen. Hoe meer overlap en variatie in standpunten, hoe beter de reconstructie.

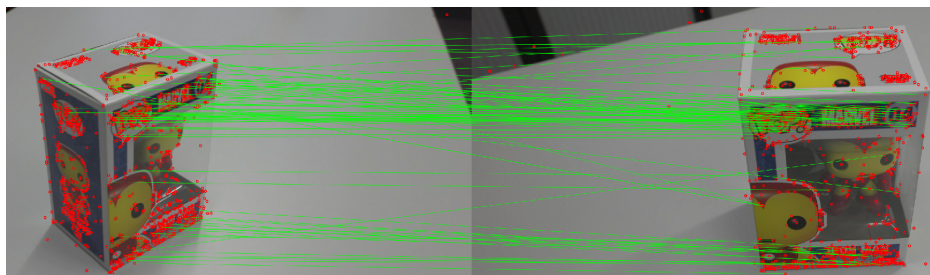
Omdat het erg moeilijk is om rechtstreeks uit deze beelden automatisch de overeenkomst tussen elke pixel te vinden, worden enkel lokale beeldkenmerken gebruikt. Dit gebeurt in twee opeenvolgende stappen: Feature Detection en Feature Matching.

Bij Feature Detection worden in elke afbeelding afzonderlijk keypoints geïdentificeerd: locaties in het beeld die zich onderscheiden door een hoge mate van herhaalbaarheid en onderscheidbaarheid. Typische voorbeelden zijn hoekpunten, blob-structuren of regio's met sterke intensiteitsgradiënten. Voor elk gedetecteerd keypoint wordt vervolgens een compacte lokale beschrijving berekend, een zogenaamde feature descriptor, die de visuele omgeving van dat punt op een schaal- en rotatie-invariante manier codeert. Een veelgebruikte methode hiervoor is SIFT (Scale-Invariant Feature Transform) [Ton12], die keypoints detecteert in een schaalruimterepresentatie van het beeld en ze beschrijft aan de hand van histogrammen van lokale gradiëntoriëntaties. In figuur 2.1 is een voorbeeld te vinden van het resultaat van de feature detection stap.

Bij Feature Matching worden de descriptorren van keypoints uit verschillende beelden vergeleken om corresponderende punten te vinden. Voor elk keypoint in een bronafbeelding wordt gezocht naar het meest gelijkende keypoint in de doelfbeelding, typisch op basis van de Euclidische afstand tussen de bijbehorende descriptorren. Het resultaat is een verzameling betrouwbare punt-naar-puntcorrespondenties over meerdere beelden, die als invoer dienen voor de verdere reconstructie. In figuur 2.2 is een voorbeeld te vinden van het resultaat van de feature matching stap.

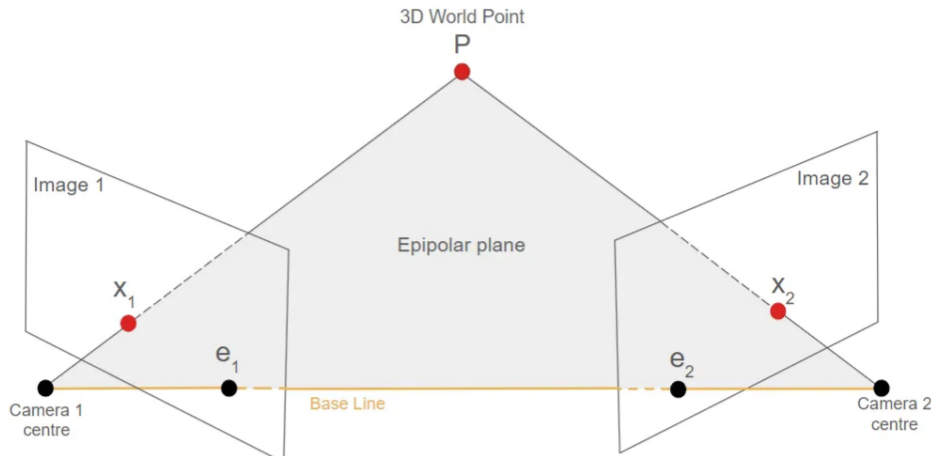


Figuur 2.1: Voorbeeld feature detection resultaat



Figuur 2.2: Voorbeeld feature matching resultaat

De volgende stap is de 3D reconstructie zelf. Allereerst worden de relatieve camerapose geschat. Deze stap bepaalt de onderlinge positie en oriëntatie van de camera's op basis van beeldcorrespondenties. De geometrische relatie tussen twee camera's, een 3D-punt en de projecties van dat punt in de respectieve beeldvlakken wordt beschreven door de epipolaire geometrie. Deze geometrie wordt weergegeven in Figuur 2.3. Deze geometrie legt vast hoe een punt in het ene beeld zich moet verhouden tot een epipolaire lijn in het andere beeld, onafhankelijk van de expliciete 3D-structuur van de scène.



Figuur 2.3: De algemene opzet van epipolaire geometrie ([Lob23])

Algebraïsch wordt deze relatie vastgelegd door de Fundamentele Matrix in het ongekalibreerde geval, of door de Essentiële Matrix wanneer de intrinsieke camerakalibratie bekend is. Deze matrices worden geschat uit de gematchte beeldpunten en coderen de epipolaire constraints tussen beeldparen. Door de ontbinding van de fundamentele of essentiële matrix kunnen een set van compatibele projectiematrices worden afgeleid, waaruit de relatieve rotatie en translatie tussen de camera's worden bepaald.

Zodra de cameraposities en -oriëntaties zijn geschat, kunnen de 3D-coördinaten van scènepunten worden gereconstrueerd uit hun overeenkomstige projecties in twee of meer beelden. Dit is de Triangulation stap. Het basisprincipe van triangulatie stelt dat een 3D-punt zich idealiter bevindt op het snijpunt van de teruggeprojecteerde stralen die vertrekken vanuit de cameracentra en door de gemeten beeldpunten lopen. In de praktijk zorgen meetruis, kalibratiefouten en onnauwkeurigheden in de geschatte camerapose ervoor dat deze stralen elkaar doorgaans niet exact snijden. Daarom wordt het 3D-punt geschat door een positie te vinden die een geschikte foutmaatstaf minimaliseert. Concreet worden de 3D-coördinaten van overeenkomstige kenmerken berekend door hun beeldposities te combineren met de bekende projectiematrices van de camera's. De resulterende 3D-punten dienen vervolgens als input voor verdere verfijning van zowel de structuur als de cameraparameters. Dit kan bijvoorbeeld via Bundle Adjustment.

Bundle adjustment is een iteratief niet-lineair optimalisatieproces waarbij zowel de cameraparameters als de 3D-puntposities gelijktijdig worden geoptimaliseerd. Het doel is het minimaliseren van een kostenfunctie die typisch wordt gedefinieerd als een som van de gekwadrateerde herprojectiefouten, dat wil zeggen het verschil tussen de waargenomen beeldpunten en de projecties van de geschatte 3D-punten in alle beschikbare weergaven. Naast het verfijnen van de geometrische consistentie van de reconstructie maakt bundle adjustment het ook mogelijk om inconsistente of foutieve 3D-punten te detecteren. Punten met uitzonderlijk grote herprojectiefouten kunnen als uitschieters worden geïdentificeerd en uit de reconstructie worden verwijderd, wat bijdraagt aan een robuustere en nauwkeurigere 3D-representatie van de scène.

De laatste stap bestaat uit het genereren van een 3D-mesh op basis van de gereconstrueerde puntenwolk en het toepassen van texturen afkomstig uit de originele beelden, wat resulteert in

een realistische visualisatie van de scène.

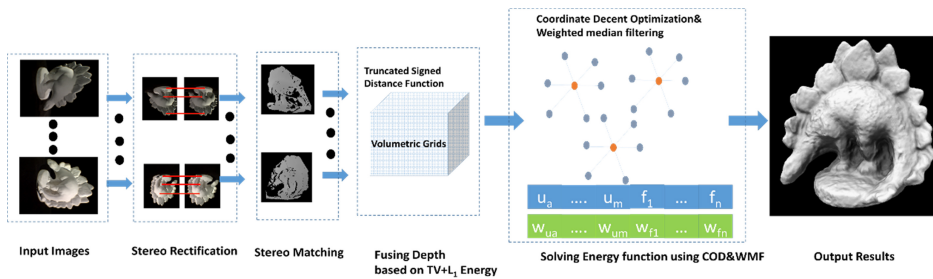
Multi-View Stereo

Multi-view stereo (MVS) wordt gebruikt om een gedetailleerde 3D-representatie van een scène te reconstrueren op basis van meerdere gekalibreerde en overlappende beelden die vanuit verschillende gezichtspunten zijn opgenomen. MVS richt zich op het berekenen van dichte diepte-informatie door voor (bijna) elke pixel overeenkomstige beeldpunten in andere beelden te vinden, wat resulteert in dichte puntenwolken. ([Fan26])

Het belangrijkste verschil met SfM is dus dat SfM voornamelijk wordt gebruikt voor camera- en poseschetsing en een ruwe, schaarse 3D-structuur, terwijl MVS net die informatie gebruikt om vervolgens een veel verfijndere en gedetailleerdere 3D-reconstructie van de scène te genereren. In veel reconstructiepijplijnen vormt SfM dan ook een voorbereidende stap voor MVS.

De MVS-pipeline begint dus met het verzamelen van meerdere overlappende, gekalibreerde beelden van dezelfde scène, waarbij de cameraposities vaak worden geschat met Structure from Motion. Vervolgens worden de beelden epipolair uitgelijnd, zodat corresponderende punten tussen de beelden eenvoudiger kunnen worden gevonden. In moderne MVS-methoden gebeurt dit soms via technieken zoals cost-volume representaties ([Nan21]). Voor elke pixel in een beeld wordt vervolgens gezocht naar corresponderende pixels in de andere beelden, wat resulteert in dieptekaarten per view. Dit kan gebeuren met traditionele matchingtechnieken of met behulp van Deep Learning based modellen die de kwaliteit van de correspondenties verbeteren. (??)

Aangezien de relatieve positie en oriëntatie van de camera's bekend zijn, kunnen punten correct worden getransformeerd tussen verschillende views. Met deze informatie worden de corresponderende punten vervolgens getrianguleerd, wat leidt tot een ruwe 3D-puntenwolk. Moderne MVS-algoritmen fuseren vaak de dieptekaarten van meerdere beelden om een dichte en nauwkeurige 3D-reconstructie te verkrijgen. Een voorbeeld van hoe zo'n pipeline eruitziet, is terug te vinden in Figuur 2.4. Hiervoor is de depth-map-fusion-based MVS methode CoD-Fusion als voorbeeld genomen ([LWMX16]).

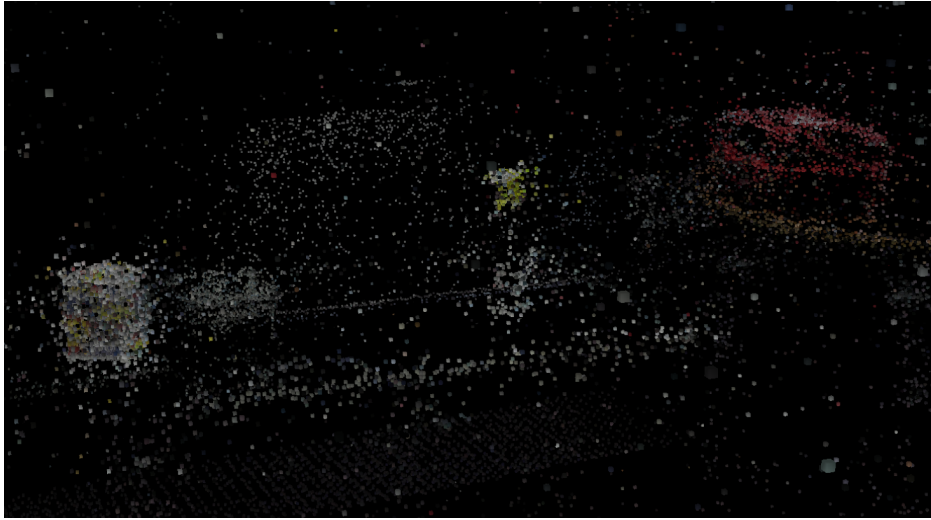


Figuur 2.4: Overzicht van de CoD-Fusion pipeline (??)

Ten slotte wordt de ruwe puntenwolk verfijnd door middel van alignment, meshing en texturering, waardoor een gedetailleerd en visueel consistent 3D-model ontstaat dat geschikt is voor verdere analyse of visualisatie.

2.1.2 Gaussian Splatting

Gaussian Splatting is een nieuwe 3D-renderingstechniek waarmee zeer gedetailleerde en fotorealistische 3D-modellen worden gemaakt door scènes weer te geven als miljoenen kleine, zachte, gekleurde "splats" (Gaussians), in plaats van traditionele polygonen of driehoeken met harde randen. 3DGS is in de kern een rasterization techniek. De techniek werkt door 3D-werelden op te bouwen uit deze splats die bij het renderen vloeiend in elkaar overgaan. Elke Gaussian bevat eenvoudige gegevens over zijn kleur, positie en transparantie. Wanneer uw computer



Figuur 2.5: Voorbeeld COLMAP-puntenwolk

de scène weergeeft, projecteert hij deze vlekken en mengt hij ze, waardoor vloeiende randen, natuurlijke lichteffecten en realistische oppervlakken ontstaan die fijne details weergeven. ([3Ds25], [Pix25], [KKLD23])

Gaussian Splatting werd in 2023 geïntroduceerd en biedt verschillende belangrijke voordelen ten opzichte van oudere methoden. In vergelijking met traditionele, op polygonen gebaseerde grafische afbeeldingen levert het natuurlijker ogende resultaten op, met zachtere randen en betere lichteffecten. Een uitgebreidere vergelijking met Mesh-based reconstructies wordt besproken in sectie 2.2. Verder maakt Gaussian Splatting, in tegenstelling tot Neural Radiance Fields (NeRF's) die enorme rekenkracht vereisen en erg traag zijn, realtime weergave mogelijk met hoge framesnelheden tussen 60 en 90 frames per seconde. ([3Ds25], [Pix25])

Technische werking van Gaussian Splatting

Zoals beschreven in [KKLD23] en [Ebe23] is de eerste stap het schatten van een puntenwolk via Structure-from-Motion (SfM), zoals uitgelegd in 2.1.1. Een veelgebruikte implementatie hiervoor is COLMAP (?). Een voorbeeld van hoe zo een puntenwolk eruit ziet is te zien in Figuur 2.5. Elk punt in deze puntenwolk dient als initialisatie voor één Gaussian. Belangrijk is dat SfM slechts positie $\boldsymbol{\mu} \in \mathbb{R}^3$ en kleur oplevert. Alle andere Gaussian-parameters worden geïntialiseerd op standaardwaarden en vervolgens geoptimaliseerd.

Elke Gaussian wordt gedefinieerd als een anisotrope 3D-Gaussiaanse functie gecentreerd op $\boldsymbol{\mu}$:

$$G(\mathbf{x}) = e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})} \quad (2.1)$$

waarbij $\boldsymbol{\Sigma} \in \mathbb{R}^{3 \times 3}$ de covariantiematrix is. Deze matrix bepaalt de vorm en oriëntatie van de Gaussian in 3D-ruimte: een grote eigenwaarde in een bepaalde richting geeft aan dat de Gaussian in die richting breed uitgespreid is. Wiskundig gezien beschrijft $\boldsymbol{\Sigma}$ een ellipsoïde.

Een voor de hand liggend idee zou zijn $\boldsymbol{\Sigma}$ direct te optimaliseren. Dit is echter problematisch: covariantiematrices zijn enkel fysisch zinvol wanneer ze positief semi-definiet zijn, en gradiëntstappen kunnen deze eigenschap gemakkelijk schenden [KKLD23]. Daarom wordt $\boldsymbol{\Sigma}$ ontbonden in een schaalmatrix S en een rotatiematrix R :

$$\boldsymbol{\Sigma} = R S S^\top R^\top \quad (2.2)$$

$S = \text{diag}(s_x, s_y, s_z)$ wordt opgeslagen als een 3D-schaalvector $\mathbf{s} \in \mathbb{R}^3$, en R als een eenheidskwaternion $\mathbf{q} \in \mathbb{R}^4$. Omdat $S S^\top$ altijd positief semi-definiet is en R orthogonaal, is $\boldsymbol{\Sigma}$ gegarandeerd geldig na elke optimalisatiestap.

Naast positie, schaal en rotatie heeft elke Gaussian nog twee extra parameters:

- **opacity** $\alpha \in [0, 1]$: hoe doorschijnend of ondoorzichtig de Gaussian is.
- **Kleur**: gerepresenteerd via sferische harmonischen (*Spherical Harmonics*, SH) van graad ℓ . In tegenstelling tot een vaste kleurwaarde per punt laten sferische harmonischen toe om de waargenomen kleur te laten variëren in functie van de kijkrichting. Dit maakt het mogelijk om view-afhankelijke effecten te modelleren, zoals spiegelingen, glansen en subtiele kleurveranderingen die optreden wanneer een oppervlak vanuit een andere hoek wordt bekeken. Wiskundig gezien worden de kleurcomponenten (r, g, b) per Gaussian uitgedrukt als een lineaire combinatie van sferische harmonische basisfuncties $Y_\ell^m(\mathbf{d})$, geëvalueerd in de genormaliseerde kijkrichting \mathbf{d} :

$$c(\mathbf{d}) = \sum_{\ell=0}^L \sum_{m=-\ell}^{\ell} k_\ell^m Y_\ell^m(\mathbf{d}) \quad (2.3)$$

waarbij k_ℓ^m de leerbare coëfficiënten zijn en L de maximale graad bepaalt. Een hogere graad L laat complexere view-afhankelijke kleurverlopingen toe, maar verhoogt ook het aantal te optimaliseren parameters per Gaussian. In de praktijk wordt vaak graad $\ell = 3$ gebruikt, wat overeenkomt met 16 coëfficiënten per kleurkanaal.

Projectie naar 2D: Differentiable Gaussian Rasterization. Om een beeld te renderen, moet elke 3D-Gaussian worden geprojecteerd naar het 2D-cameraplane. Gegeven de viewing-transformatiematrix W (de world-to-camera transformatie) wordt de geprojecteerde 2D-covariantiematrix Σ' berekend als [KKLD23]:

$$\Sigma' = JW\Sigma W^\top J^\top \quad (2.4)$$

waarbij J de Jacobiaan is van de affine benadering van de projectieve transformatie. Door de derde rij en kolom van Σ' weg te laten, bekomt men een 2×2 covariantiematrix die de geprojecteerde 2D-splat beschrijft.

Alpha-blending en het beeldvormingsmodel. De uiteindelijke pixelkleur C wordt berekend door de bijdragen van alle Gaussians die die pixel overlappen te mengen, gesorteerd van voor naar achter op diepte. Dit α -blending model is:

$$C = \sum_{i \in \mathcal{N}} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \quad (2.5)$$

waarbij c_i de kleur is van de i -de Gaussian en α_i bepaald wordt door de geëvalueerde 2D-Gaussian vermenigvuldigd met de geleerde opacity van die Gaussian. Dit beeldvormingsmodel is equivalent aan volumetrische ray-marching in NeRF [KKLD23], maar laat zich veel efficiënter implementeren via rasterisatie.

De rasterisatie verloopt tile-based: het scherm wordt opgedeeld in tegels van 16×16 pixels. Per tegel worden enkel de relevante Gaussians verwerkt, wat een efficiënte GPU-parallellisatie mogelijk maakt.

Optimalisatie via Stochastic Gradient Descent. Het optimalisatieproces minimaliseert een lossfunctie die de fotometrische fout meet tussen het gerenderde beeld \hat{I} en het grondwaarheidbeeld I . De gebruikte lossfunctie combineert een \mathcal{L}_1 -term met een D-SSIM-term [KKLD23]:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{\text{D-SSIM}} \quad (2.6)$$

met $\lambda = 0.2$ in de originele implementatie. D-SSIM (Differentiable Structural Similarity Index) bestraft structurele verschillen in het beeld, wat fijnere details preserveert dan een louter

pixelgewijze fout. In tegenstelling tot \mathcal{L}_1 dat elke pixel onafhankelijk evalueert, berekent D-SSIM de gelijkenis over lokale beeldvensters aan de hand van drie perceptuele componenten: luminantie, contrast en structuur. Hierdoor is de verliesterm gevoelig voor textuurpatronen en randen die pixelgewijze metrieken kunnen missen wanneer fouten ruimtelijk verspreid zijn.

De te optimaliseren parameters per Gaussian zijn: positie $\boldsymbol{\mu}$, schaalvector \mathbf{s} , kwaternion \mathbf{q} , opacity o en SH-coëfficiënten k_ℓ^m . Al deze parameters worden simultaan bijgewerkt via Stochastic Gradient Descent met Adam-optimizer. Cruciaal hiervoor is dat zowel de rasterisatie (vergelijking 2.5) als de projectie (vergelijking 2.4) differentieerbaar zijn, zodat gradients door het volledige renderingproces kunnen terugvloeien naar alle Gaussian-parameters.

Adaptieve densificatie en pruning. Een centraal onderdeel van het trainingsproces is het adaptief beheren van het aantal Gaussians. Na elke N iteraties wordt een densificatie- en pruningsstap uitgevoerd, gestuurd door twee drempelwaarden: τ_{pos} voor de positiegradient en τ_{size} voor de grootte van een Gaussian [KKLD23]:

- **Densificatie:** Gaussians met een te grote positiegradient $\|\nabla_{\boldsymbol{\mu}}\mathcal{L}\|$ worden gesplitst of gekloneerd. De drempel hiervoor is $\tau_{\text{pos}} = 0.0002$. Het onderscheid tussen klonen en splitsen hangt af van de grootte van de Gaussian:
 - *Klonen:* als de Gaussian klein is ($\max(\mathbf{s}) < \tau_{\text{size}}$) wordt een identieke kopie aangeemaakt en beiden verschoven in de richting van de positiegradient. Dit dicht gaten op in gebieden met te weinig dekking (*under-reconstruction*).
 - *Splitsen:* als de Gaussian groot is ($\max(\mathbf{s}) \geq \tau_{\text{size}}$) wordt ze vervangen door twee kleinere Gaussians, gesampled via de probabiliteitsdichtheid van de oorspronkelijke Gaussian, met schalingsfactor $1/1.6$. Dit verbetert de representatie van fijne details in gebieden met *over-reconstruction*.
- **Pruning:** Gaussians worden verwijderd als hun opacity o onder een drempelwaarde ϵ_α zakt. Bovendien worden periodiek alle opacities gereset naar nul, zodat overbodige Gaussians de kans krijgen om door het optimalisatieproces vanzelf te verdwijnen.

Dit densificatiemechanisme stelt het systeem in staat om te starten vanuit een schaarse puntenwolk en gradueel een gedetailleerde representatie op te bouwen van 1 tot 5 miljoen Gaussians, afhankelijk van de complexiteit van de scène. Een schematisch overzicht van hoe een Gaussian Splatting training werkt is terug te vinden in Figuur 2.6.

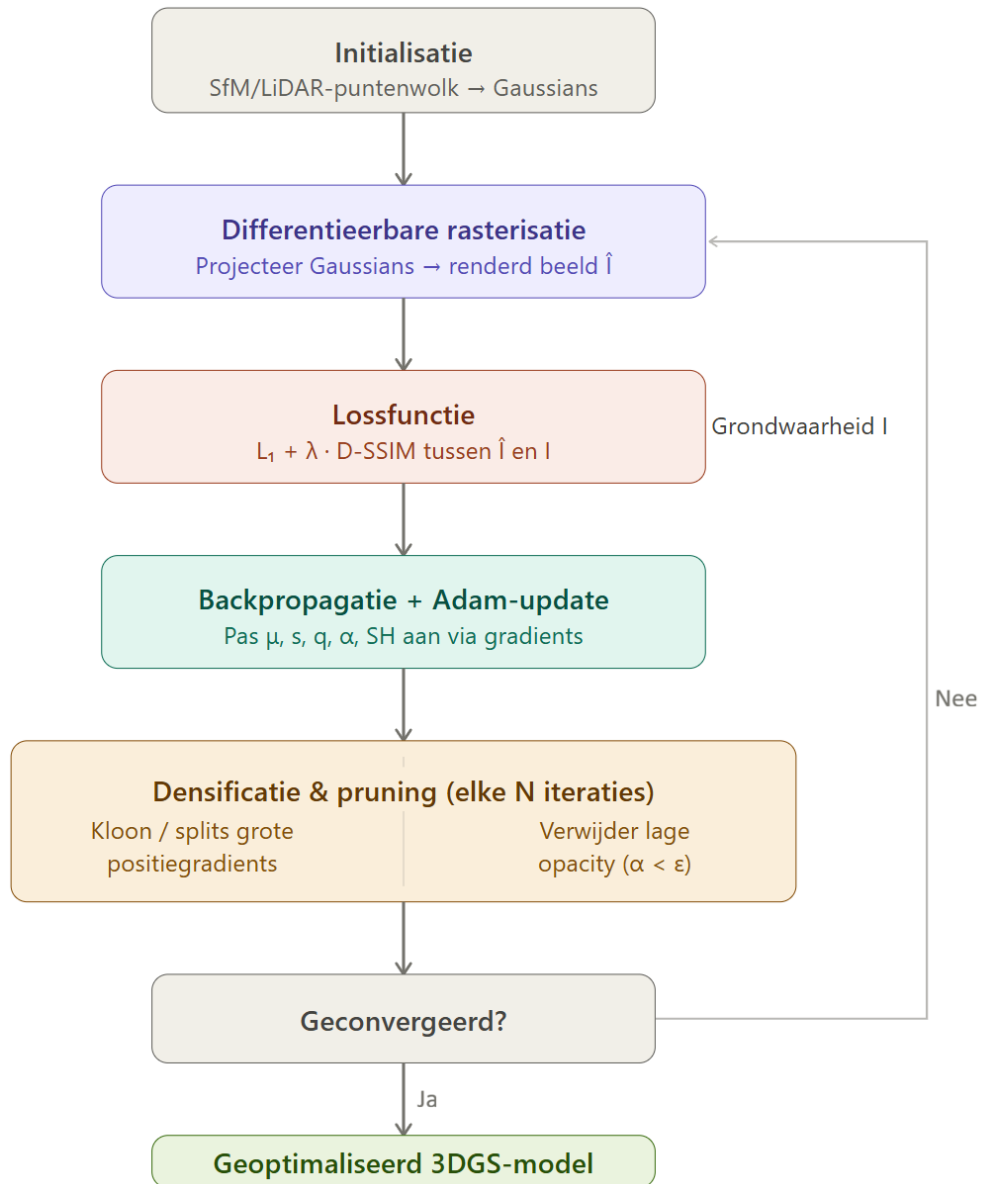
Toepassingen in realistische 3D-scan-omgevingen

De techniek heeft praktische toepassingen op meerdere gebieden, waarbij de balans tussen visuele kwaliteit en technische precisie een belangrijke rol speelt in de bruikbaarheid voor verschillende gebruikersgroepen.

Voor fotografen en mobiele gebruikers maakt 3D-gaussian splatting een snelle conversie van video's of scans naar 3D-modellen via apps mogelijk. Voor AR/VR- en game-ontwikkelaars biedt het vloeiendere, comfortabelere immersieve ervaringen. Kunstenaars en creators profiteren van de realistischere beeldkwaliteit met onder andere gladde randen, zachte verlichting en reflecties die met driehoeken of traditionele fotogrammetrie moeilijker te realiseren zijn. ([3Ds25])

Een andere toepassing hiervan vinden we bij interactieve vastgoedervaringen. Potentiële kopers maken een virtuele rondleiding door 3D-omgevingen die de realiteit nabootsen. Met Gaussian Splatting creëren ontwikkelaars levensechte omgevingen op basis van foto's, volledig met realistisch ruimtegevoel, uitzichten en belichting. Dit maakt het ideaal voor online bezichtigingen en presale-ervaringen. ([Gro25])

Voor professioneel landmeetkundig en technisch werk kent Gaussian Splatting momenteel echter nog beperkingen. Het kan nog niet de centimeter- of millimeterprecisie garanderen die fotogrammetrie en laserscanning bieden. Daarentegen is Gaussian Splatting wel al geïntegreerd



Figuur 2.6: Gaussian Splatting Training Schema

in tools zoals V-Ray wat het mogelijk maakt om dronebeelden makkelijker te combineren met digitale modellen, waardoor bijvoorbeeld architecten nieuwe gebouwen in hun werkelijke omgeving kunnen presenteren. ([Gro25])

Een laatste toepassing die we bespreken is om Gaussian Splatting in te zetten voor creatieve reclame. Hierbij plaatsen marketeers hun producten in levensechte omgevingen. Een boot kan bijvoorbeeld worden getoond tegen prachtige, werkelijkheidsgetrouwe achtergronden. Dit creëert krachtige visuele verhalen die zorgen voor meer emotionele betrokkenheid bij klanten. ([Gro25])

Beperkingen voor forensische inzetbaarheid

Wanneer beelden motion blur of andere suboptimale effecten bevatten, kunnen er artefacten, zogenaamde 'floaters', ontstaan. Deze visuele artefacten verminderen de helderheid en uniformiteit van het model en kunnen post-processing of analysetaken moeilijker maken, vooral in forensische contexten waar interpreteerbaarheid cruciaal is ([Dee25]). Een studie ([RWW⁺25]) uitgevoerd door onder andere de politieacademie van Nederland en Technologies for criminal Investigations (Saxion) onderzoekt hoe verschillende filmtechnieken de kwaliteit van 3D-reconstructies kunnen verbeteren, met bijzondere aandacht voor het gebruik ervan bij onderzoek van indoor crime scenes. Hieruit werd geconcludeerd dat zaken zoals de camera-oriëntatie (landscape vs portait), de loopsnelheid van de camera-operator, scan-paden en segmentatie van de omgeving een belangrijke invloed hebben op de kwaliteit van 3D-reconstructies.

Hoewel Gaussian Splatting visueel indrukwekkend is en vaak sneller is dan alternatieven, voldoet het nog niet aan de geometrische nauwkeurigheidseisen voor forensische documentatie waar metingen cruciaal zijn. De techniek blijft voornamelijk een visualisatie-tool en is nog niet geschikt voor nauwkeurige forensische toepassingen waar millimeter-precisie vereist is.

Een andere belangrijke beperking van 3D Gaussian Splatting voor forensische inzetbaarheid is het feit dat ontbrekende of onvoldoende vastgelegde delen van een scène impliciet door het algoritme worden ingevuld. Wanneer bepaalde oppervlakken of objectdelen niet of nauwelijks zijn vastgelegd, reconstrueert het model deze gebieden op basis van de geoptimaliseerde Gaussian-parameters uit aangrenzende, wel waargenomen regio's, geïnterpoleerd via het renderingproces. Dit leidt tot visueel overtuigende, maar potentieel onjuiste representaties, waarbij niet altijd duidelijk te onderscheiden is welke delen gebaseerd zijn op directe observaties en welke het resultaat zijn van modelextrapolatie tijdens het optimalisatieproces. Dit onderscheid is in een forensische context problematisch omdat een reconstructie die visueel overtuigend oogt maar gedeeltelijk op modelinterpolatie berust de interpreteerbaarheid van bewijsmateriaal kan bemoeilijken. ([Gye25])

2.1.3 Puntenwolk-initialisatie in combinatie met Structure from Motion

Structure from Motion, zoals geïmplementeerd in COLMAP, blijft een essentieel onderdeel van de Gaussian Splatting pipeline voor het schatten van cameraposities en -oriëntaties. Zonder accurate cameraparameters is het onmogelijk om Gaussians correct in 3D-ruimte te positioneren. SfM wordt echter beschreven als een "*slow and computationally expensive*" stap die een belangrijke bottleneck vormt voor praktische toepassingen ([FRY⁺24]). Het gebruik van een bestaande puntenwolk vervangt SfM dus niet, maar biedt een rijkere en meer accurate geometrische initialisatie dan de sparse puntenwolk die SfM zelf produceert.

De puntenwolk die COLMAP genereert is doorgaans schaars en onvolledig, wat de Gaussian-optimalisatie bemoeilijkt in gebieden met weinig textuur, dynamische content of sterke view-afhankelijke effecten zoals reflecties ([FRY⁺24]). Een externe puntenwolk, bijvoorbeeld afkomstig van LiDAR, biedt een metrisch nauwkeurigere en completere geometrische prior voor oppervlakken die wel zichtbaar zijn in het beeldmateriaal, maar waarbij de SfM-puntenwolk schaars blijft door gebrek aan textuur, sterke reflecties of onvoldoende beeldoverlap. In deze gebieden

zorgt de LiDAR-initialisatie voor een stabielere startpositie van de Gaussians, waardoor het optimalisatieproces sneller convergeert en minder afhankelijk is van densificatie vanuit een lege ruimte.

De combinatie van LiDAR-geometrie en SfM-cameraparameters wordt beschouwd als een complementaire aanpak die de voordelen van beide technieken benut. SfM levert de nauwkeurige cameraposities die noodzakelijk zijn voor Gaussian Splatting, terwijl de externe puntenwolk een geometrisch rijkere en robuustere initialisatie biedt dan de schaarse COLMAP-output. Dit is met name relevant voor uitdagende scenario's zoals wateroppervlakken met dynamische content of sterk begroeide terreinen, waar SfM-puntenwolken onvolledig of onbetrouwbaar zijn ([FRY⁺24, LZY21]).

SfM blijft een onmisbare stap in de Gaussian Splatting pipeline voor de schatting van cameraparameters. Wat wel vervangen of aangevuld wordt, is de *initialisatie* van de Gaussians: in plaats van de sparse SfM-puntenwolk als startpunt te gebruiken, biedt een externe puntenwolk een completere en nauwkeurigere geometrische basis. Voor toepassingen waar volledigheid, metrische nauwkeurigheid en robuustheid in uitdagende omgevingen prioriteit hebben, vormt deze hybride aanpak een aanzienlijke verbetering ten opzichte van het exclusief vertrouwen op COLMAP-uitvoer als initialisatie. In hoofdstuk 4 wordt empirisch onderzocht in welke mate deze hybride initialisatie de reconstructiekwaliteit beïnvloedt.

2.2 Gaussian Splatting vs Mesh-based reconstructie

Gaussian splatting en 3D meshing zijn twee fundamenteel verschillende technieken voor de representatie en visualisatie van driedimensionale ruimtelijke data. Hoewel beide methoden worden ingezet om complexe 3D-omgevingen te visualiseren, verschillen ze sterk in structuur, toepassingsgebied en analytische mogelijkheden, wat vooral duidelijk wordt in domeinen zoals Geographic Information Systems (GIS) en Building Information Modeling (BIM). ([Car25b])

2.2.1 Verschillen

Een essentieel verschil tussen Gaussian splatting en 3D meshing is de aanwezigheid van expliciete geometrie. Gaussian splatting representeert 3D-data als een verzameling punten met bijbehorende Gaussian distributions, wat resulteert in een continue, volumetrische visuele weergave zonder duidelijk afgebakende oppervlakken. Hierdoor ontbreekt een expliciete definitie van vlakken, randen en topologie. 3D meshes daarentegen bestaan uit vertices, edges en faces en vormen een expliciet gedefinieerde oppervlaktegeometrie. Deze gestructureerde representatie is cruciaal in GIS- en BIM-contexten, waar objecten zoals gebouwen, wegen en terrein nauwkeurig moeten worden gemodelleerd. In figuur 2.7 is een visuele vergelijking tussen een Mesh-based reconstructie en een Gaussian Splatting reconstructie terug te vinden.

Een ander verschil zit in de mogelijkheden voor ruimtelijke analyse. GIS-toepassingen vereisen nauwkeurige geometrische modellen om analyses uit te voeren zoals zichtlijnanalyse, overstroomingsmodellering en hellingsberekeningen. Door het ontbreken van duidelijke oppervlakken zijn Gaussian splats moeilijk toepasbaar in klassieke GIS-algoritmen. De techniek is primair gericht op visuele representatie en minder geschikt voor analytische bewerkingen. 3D meshes ondersteunen daarentegen topologische en geometrische operaties zoals intersecties, buffering en terreinmodellering. Hierdoor zijn ze beter geschikt voor ruimtelijke analyse en besluitvorming in stedelijke planning en risicobeoordeling.

Een ander belangrijk verschil is de mate van ondersteuning binnen bestaande software-ecosystemen. GIS- en BIM-tools zoals ArcGIS, QGIS, Revit en AutoCAD zijn ontworpen rond polygonale modellen en ondersteunen standaard bestandsformaten zoals OBJ, FBX en STL. Gaussian splatting kent daarentegen nog beperkte ondersteuning binnen deze platformen. Integratie vereist vaak maatwerkoplossingen of conversiestappen, wat de toepasbaarheid binnen bestaande workflows bemoeilijkt.



Figuur 2.7: Visueel verschil tussen Mesh-based reconstructie (links) en Gaussian Splatting (rechts) ([Pix25])

In BIM-toepassingen is het frequent aanpassen en herontwerpen van modellen essentieel. 3D meshes laten gedetailleerde bewerkingen toe, zoals het aanpassen van geometrie, topografie of infrastructuurelementen. Gaussian splatting is hoofdzakelijk een renderingtechniek. Doordat de data bestaat uit losse Gaussiaanse primitieve elementen zonder expliciete randen of vlakken, is precieze geometrische bewerking moeilijk of zelfs onmogelijk. Dit beperkt de bruikbaarheid voor ontwerp- en modelleringstaken. ([Ziy26])

Tot slot biedt Gaussian splatting voordelen op het vlak van real-time rendering en visuele kwaliteit, vooral bij grote puntendatasets. De techniek kan fotorealistische beelden genereren met relatief lage rekentijd voor visualisatie. Bij grootschalige GIS-toepassingen kan dit echter problematisch worden door het hoge GPU- en geheugenverbruik. 3D meshes kunnen daarentegen worden geoptimaliseerd via LOD-technieken (Level of Detail), waardoor ze efficiënter schaalbaar zijn voor stads- en terreinmodellen. Recentelijk zijn er echter ook LOD-technieken ontwikkeld specifiek voor 3D Gaussian Splatting, die vergelijkbare schaalbaarheidsvoordelen bieden ([Ber24]).

Tabel 2.1 geeft nog eens een samenvattend overzicht tussen Mesh-based reconstructies en Gaussian Splatting.

2.2.2 Forensische toepasbaarheid

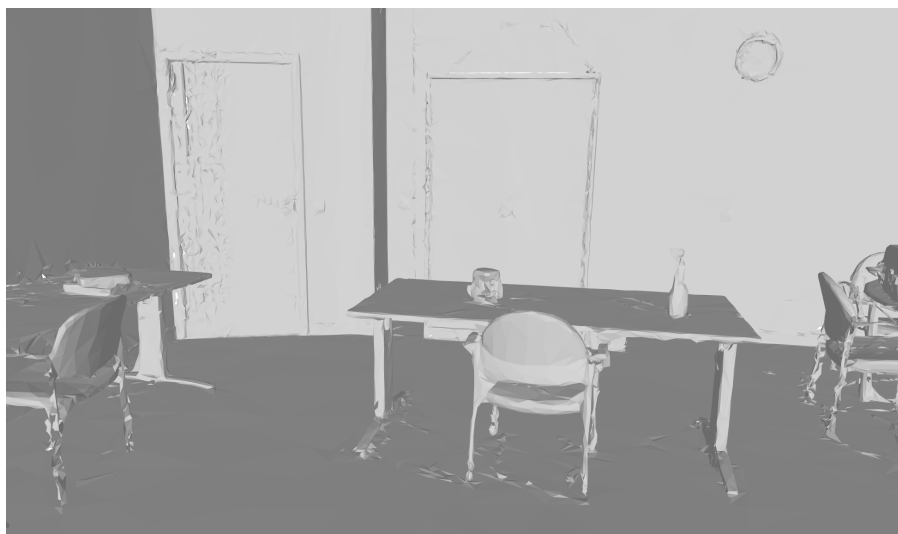
Wanneer mesh-based reconstructie en 3D Gaussian Splatting worden vergeleken vanuit een forensisch perspectief, komen zowel complementariteiten als fundamentele spanningen aan het licht. Elk van beide technieken biedt specifieke voor- en nadelen die bepalend zijn voor hun inzetbaarheid als forensisch bewijsmiddel.

Het voornaamste forensische voordeel van mesh-based reconstructie is de geometrische precisie en meetbaarheid. Laserscanning en fotogrammetrie produceren gestructureerde puntwolken die worden omgezet in meshes met expliciet gedefinieerde hoekpunten en vlakken, waardoor afstanden, hoeken en ruimtelijke relaties tussen bewijsstukken direct en consistent kunnen worden berekend. In Figuur 2.8 is een mesh-based reconstructie te zien van de dataset beschreven in Sectie 3.2. In Figuur 2.9 is een meting van de deur uit deze scène waarop rechtsboven te zien is dat de grootte ervan overeenkomt met wat er in het echt gemeten is. De expliciete geometrie van meshes maakt het ook mogelijk om bewijsstukken digitaal te annoteren en te meten zonder fysieke interventie op de scène.

Ondanks dit voordeel kent mesh-based reconstructie ook beperkingen die forensisch relevant

Feature	Gaussian Splatting	Mesh Modeling
Kernstructuur van gegevens	Discrete punten en Gaussian kernels	Hoekpunten en driehoekige vlakken
Visueel realisme	Blinkt uit in het weergeven van ingewikkelde details zoals draden, gebladerte en lichtreflecties.	Vertrouwt op texturen en belichtingstechnieken, maar heeft moeite met complexe, niet-solide details.
Geometrische precisie en metingen	Randen zijn wazig door puntgebaseerde weergave, wat kan leiden tot inconsistenties in metingen tussen verschillende weergaven.	Duidelijke hoekpunten en randen maken nauwkeurige 3D-coördinatenmetingen mogelijk met consistente resultaten. Goed voor forensische doeleinden
Bewerking en interactie	Nog steeds een uitdaging. Bewerking is afhankelijk van complexe parameteraanpassingen, omdat intuïtieve hoekpunt- en vlaktools ontbreken.	Gebruiksvriendelijk. Directe manipulatie van hoekpunten, randen en vlakken maakt bewerken en aanpassen eenvoudig.
Hardware en performance	Zeer efficiënte weergave, maar sterk afhankelijk van high-end GPU's voor optimale resultaten. Bij grootschalige scènes kan het geheugenverbruik problematisch worden, hoewel hiërarchische LOD-methoden voor 3DGS dit deels adresseren.	Lagere hardware-eisen, hoewel de weergavekwaliteit meer afhangt van modeloptimalisatie dan van GPU-vermogen.
Industrie	Opkomende technologie met beperkte huidige softwareondersteuning.	Industriestandaard. Formaten zoals OBJ en FBX worden breed ondersteund op verschillende platforms.

Tabel 2.1: Samenvattende tabel vergelijking Gaussian Splatting en Mesh Modeling



Figuur 2.8: Mesh-based reconstructie van de dataset beschreven in Sectie 3.2



Figuur 2.9: Meting deur uit mesh-based reconstructie

zijn. Ten eerste presteren meshes minder goed bij textuurarme, glanzende of transparante oppervlakken zoals glas, metaal of natte vloeren. Materiaaltypen die frequent voorkomen op een plaats delict. Verder bieden meshes een beperktere visuele overtuigingskracht bij complexe, niet-solide details zoals bloedspattingspatronen of fijn sporenmateriaal. 3D-gaussian splatting reproduceert dergelijke details fotorealistischer, wat relevant is wanneer reconstructies worden gepresenteerd aan een jury zonder technische achtergrond.

Samenvattend is mesh-based reconstructie superieur op het vlak van geometrische precisie, metrische reproduceerbaarheid en juridische traceerbaarheid, eigenschappen die in een forensische context van primair belang zijn. 3D-gaussian splatting biedt daarentegen voordelen op het vlak van visueel realisme, verwerkingssnelheid en het vastleggen van complexe scènedetails. De kritische beperking van 3D-gaussian splatting voor forensisch gebruik is het ontbreken van een gevalideerde methodologie om de geometrische nauwkeurigheid aantoonbaar en reproduceerbaar te garanderen.

2.3 3D-reconstructies in forensisch onderzoek

De inzet van 3D-reconstructietechnieken binnen forensisch onderzoek kent een groeiende maar nog geen gestandaardiseerde praktijk. Zaak per zaak wordt nagegaan of de inzet nuttig kan zijn voor het verder onderzoek. Hoewel de technologische mogelijkheden snel evolueren, loopt de operationele implementatie bij politiediensten daar vaak op achter. De finaliteit voor het nemen van een 3D-scan werd nog niet bepaald. Dit blijkt ook uit een interview dat werd afgenomen met Commissaris Marleen Claesen, Afdelingshoofd CSI en Adjunct-afdelingshoofd Criminaliteit Tegen Personen bij de Federale Gerechtelijke Politie Limburg. Zij bevestigt dat 3D-scanning bij haar dienst wel al sinds 2016 wordt ingezet, maar dat de technologie zo snel evolueert, dus ook constant in testfasen verkeert.

De meest gehanteerde technologie is LiDAR-scanning, in Limburg uitgevoerd door een gespecialiseerde privédeskundige van de firma 3ES. Concreet wordt onder meer gebruik gemaakt van Leica-hardware. Bij de dienst bloedspoorpatroonanalyse van de politie, gevestigd te Brussel, die

nationaal steun levert aan alle CSI-diensten maakt men gebruik van de LiDAR-sensor via een iPad Pro en bijhorende software.

Een belangrijk organisatorisch gegeven is het principe van spoorbewust werken: de 3D-scan wordt sinds een jaar pas op het einde van het sporenonderzoek uitgevoerd, nadat alle fysieke sporen zijn geregistreerd en veiliggesteld. Dit heeft als gevolg dat de scan een snapshot is van de scène na interventie, niet van de scène in haar originele toestand. Een experimentele toepassing die in het verleden ter sprake gekomen was om dit op te lossen, is het achteraf digitaal ‘inplakken’ van een lichaam in een bestaande LiDAR-scan. Dit blijkt technisch uitdagend: een correct geïntegreerde plaatsing van een lichaamsobject in een puntenwolk of 3D-model vereist een hoge geometrische consistentie tussen beide databronnen, wat niet vanzelfsprekend is wanneer ze op verschillende tijdstippen of met verschillende sensoren zijn opgenomen.

Naast de gevestigde LiDAR-workflow worden momenteel ook alternatieve systemen geëvalueerd. Zo werd een xgrids PortalCam getest als snellere en laagdrempeligere opnamemethode. De PortalCam laat toe om met beperkte expertise een volledige scène vast te leggen, wat aansluit bij de operationele wens dat iedereen binnen de dienst met de technologie moet kunnen werken. Een belangrijk voorbehoud is echter dat dergelijke systemen niet wetenschappelijk metrisch zijn: de geometrische nauwkeurigheid is onvoldoende om betrouwbare afstandsmetingen te garanderen. Het is noodzakelijk dat systemen toegankelijk en snel zijn, maar ook metrische garanties bieden. Nauwkeurige systemen bevatten hogere technische drempels en zijn daardoor minder operationeel schaalbaar.

Binnen de dienst wordt 3D-reconstructie steeds ingezet bij moord/doodslag op bepaalde verdachte overlijdens waarbij de ruimtelijke context een belangrijk onderzoकेlement kan zijn, bv. een val van een trap waarbij de verwondingen van het slachtoffer niet overeenkomen met een dergelijke val. Er is echter nog geen formeel beleid over in welke types zaken een 3D-scan verplicht of aanbevolen is, de inzet wordt per zaak beoordeeld op nut en meerwaarde. Als potentiële toepassingen worden ook vermeld: het visueel ondersteunen van een assisenproces, het bijpraten van nieuwe onderzoekers die later op een zaak worden gezet en het gebruik in politieopleidingen.

Een centraal onderscheid in de forensische praktijk is dat tussen metrisch bruikbaar bewijsmateriaal en louter illustratief materiaal. Commissaris Claesen geeft aan dat 3D-scans tot nu toe nog niet visueel als bewijsmateriaal zijn gebruikt in een rechtszaak, hoewel ze wel retroactief zijn gebruikt om meetvragen te beantwoorden die tijdens het onderzoek niet onmiddellijk werden gesteld (bijvoorbeeld de hoogte van een nog niet eerder gemeten object). De technologie wordt momenteel eerder gezien als een aanvulling op klassieke fotografie, niet als een vervanging.

2.4 Kwaliteits- en betrouwbaarheidscriteria

Hoewel de toepassing van het gebruik van 3D-reconstructies in forensisch onderzoek nog volop in ontwikkeling is, bestaan er op dit moment wel kwaliteits- of betrouwbaarheidscriteria specifiek voor deze technologie. Dit werd ook bevestigd door Commissaris Claesen tijdens het interview: de deskundigen die ingezet worden tijdens een onderzoek zijn beëdigd en hebben een juridische opleiding gevolgd. Enkel op deze manier kan een persoon/firma op de deskundigenlijst van de federale overheidsdienst opgenomen worden als forensisch 3D-deskundige. Het gebruikte materiaal dient ten alle tijden geijkt te zijn en zij dienen in het bezit te zijn van de nodige attesten.

Het meest concrete kwaliteitsonderscheid dat in de praktijk wordt gemaakt, betreft de metrische nauwkeurigheid van het gebruikte systeem. Commissaris Claesen geeft aan dat LiDAR-gebaseerde systemen als wetenschappelijk en metrisch betrouwbaar worden beschouwd door de aanwezigheid van de ijkingsattesten, terwijl het geteste systeem, de XGRIDS PortalCam dat voorlopig niet is. Dit systeem is niet voldoende technologie-gestuurd om de metrische nauwkeurigheid te garanderen.

Het gewenste nauwkeurighedsniveau hangt natuurlijk af van het beoogde gebruik: voor louter visuele doeleinden, bijvoorbeeld voor een reconstructie ter ondersteuning van een assisenproces is centimeterprecisie minder kritisch, terwijl meetvragen over posities of afstanden een hogere geometrische nauwkeurigheid vereisen. De finaliteit van het gebruik van de scan is dus allesbepalend of een systeem kan voldoen of niet.

Een tweede informeel richtsnoer wordt afgeleid uit de bestaande praktijk rond forensische fotografie. Voor digitale foto's zijn bij de dienst strikte protocollen van kracht: het originele bestand wordt ongewijzigd bewaard, metadata mag niet worden aangepast, en elke bewerking wordt gedocumenteerd zodat de integriteit van het bewijsmateriaal aantoonbaar blijft. Commissaris Claesen geeft aan dat voor 3D-scandata gelijkaardige principes gelden. Bij navraag bij de deskundige blijkt dit ook het geval te zijn. De scandata van de puntenwolk blijft behouden en kan niet aangepast worden. Alle aanpassingen die gebeuren bij het postprocessen worden ook gedocumenteerd en blijven achterhaalbaar. Hierdoor blijft het originele bestand ook ongewijzigd.

Een praktisch criterium dat bij de evaluatie van nieuwe systemen al impliciet wordt meegenomen, is de vraag waar de data terechtkomt. Bij het beoordelen van nieuwe hardware of software wordt nagegaan of opgenomen data lokaal wordt bewaard of naar externe servers wordt verstuurd en of de leverancier afkomstig is uit een geopolitiek gevoelig land. Hoewel dit geen formeel vastgelegd criterium is, getuigt het van een groeiend bewustzijn rond gegevensbeveiliging en ketenintegriteit, ook buiten de eigenlijke reconstructietechnologie.

Tot slot is het veelzeggend dat er op het moment van het interview nog geen ervaring bestond met 3D-reconstructies die in een rechtszaak door de verdediging formeel werden aangevochten. Dit hangt samen met het feit dat dergelijke reconstructies vooralsnog zelden of nooit als zelfstandig bewijsmateriaal worden ingediend. Hoewel er wel degelijk kwaliteitscriteria bestaan, zoals de vereiste van ijkingsattesten, de beëdiging van deskundigen en de documentatie van postprocessingstappen, zijn deze criteria eerder praktisch en impliciet van aard dan formeel vastgelegd in een specifiek kader voor forensische 3D-reconstructie. Zolang reconstructies niet als zelfstandig bewijsmateriaal worden ingediend en juridisch worden getoetst, blijft de druk om zulke criteria verder te formaliseren beperkt voelbaar.

Hoofdstuk 3

Methode

3.1 Overzicht van de experimentele pipeline

Dit onderzoek volgt een experimentele computationele methode om 3D-reconstructies op basis van Gaussian Splatting geometrisch te koppelen aan meetdata afkomstig van een LiDAR-scan. Het doel van de voorgestelde methode is het combineren van visuele informatie uit videobeelden met metrisch betrouwbare geometrie, zodat de resulterende reconstructie geschikt is voor forensische toepassingen.

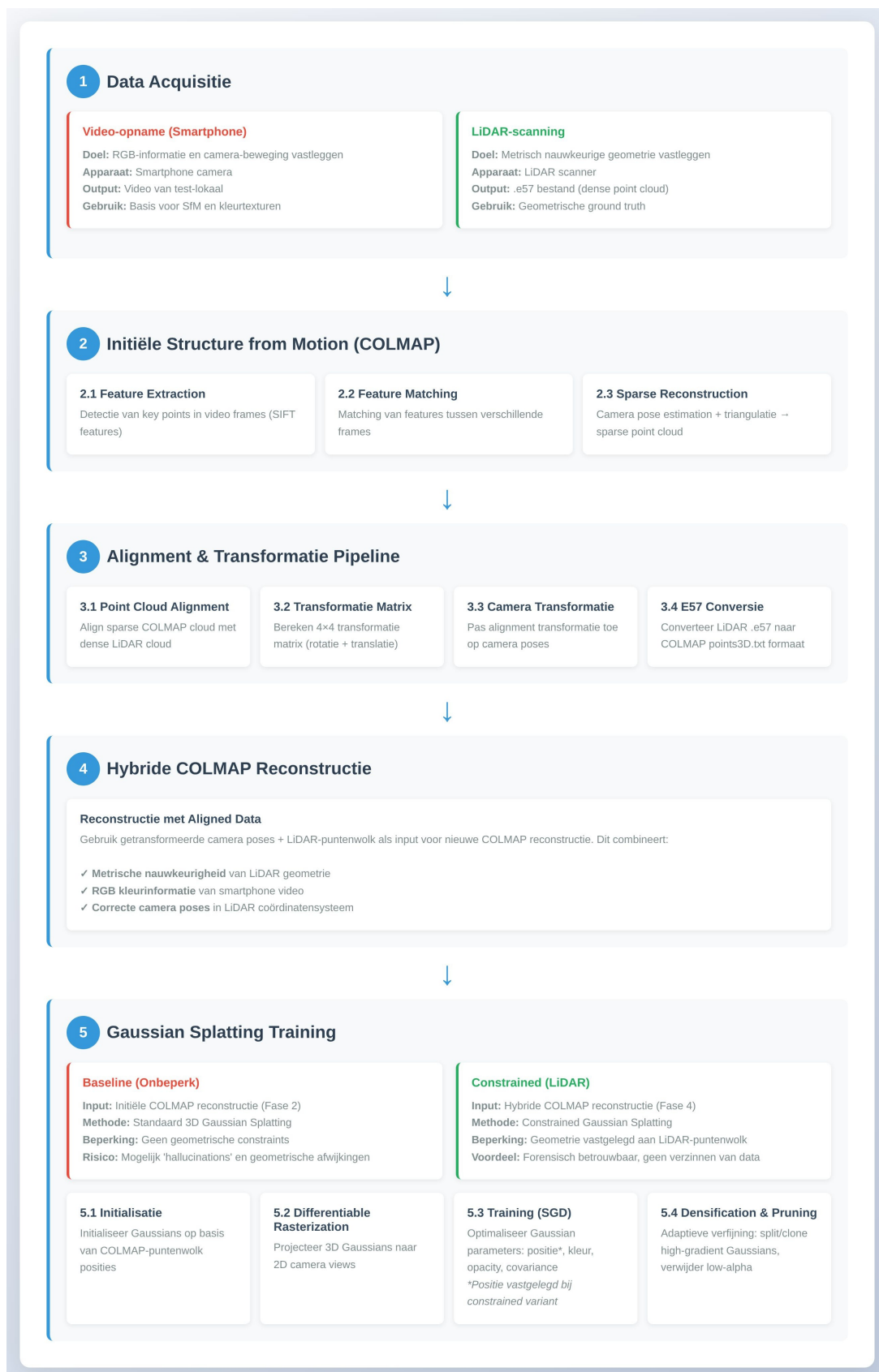
De methodologie bestaat uit vijf hoofdfasen, zoals weergegeven in Figuur 3.1.

Fase 1: Data-acquisitie. Voor elke scène worden twee datasets verzameld. Met een smartphone wordt een video opgenomen die RGB-informatie en voldoende camerabeweging voor Structure from Motion-reconstructie levert. Daarnaast wordt een LiDAR-scan gemaakt die de metrisch correcte geometrie van de scène vastlegt in een dichte puntenwolk (.e57-formaat).

De LiDAR-captatie omvat echter meer dan een enkelvoudige meting. Omdat een volledige scène doorgaans niet vanuit één scanpositie kan worden vastgelegd, worden meerdere scans vanuit verschillende standpunten opgenomen. Deze afzonderlijke scans worden vervolgens verwerkt via een dedicated pipeline: eerst worden de scans grof uitgelijnd op basis van gedeelde referentiepunten of targets, waarna een verfijningsstap via het ICP-algoritme zorgt voor een nauwkeurige registratie van de afzonderlijke puntenwolken tot één coherente, metrisch consistente scène. Het resultaat wordt geëxporteerd als een gefuseerde puntenwolk in .e57-formaat.

Fase 2: Initiële Structure from Motion reconstructie. De smartphone-video wordt verwerkt met COLMAP. Dit levert een sparse 3D-reconstructie op bestaande uit geschatte camera poses (rotaties en translaties), camera intrinsieken en een sparse puntenwolk. Hoewel deze reconstructie geometrisch minder nauwkeurig is dan de LiDAR-scan, bevat deze waardevolle informatie over de camera-trajecten.

Fase 3: Alignment en transformatie pipeline. De kern van de methode is de alignment tussen beide datasets. De sparse COLMAP-puntenwolk wordt gealigneerd met de dichte LiDAR-puntenwolk, wat resulteert in een 4×4 transformatiematrix. Deze transformatie wordt vervolgens toegepast op alle COLMAP camera poses, zodat de camera's correct gepositioneerd zijn ten opzichte van de LiDAR-geometrie. Daarna wordt de LiDAR-puntenwolk geconverteerd naar COLMAP's points3D.txt formaat, waarbij eerst ook een voxel downsampling wordt toegepast om de dataset hanteerbaar te houden voor Gaussian Splatting training. Hier wordt in sectie 3.4 dieper op ingegaan.



Figuur 3.1: Pipeline van de voorgestelde methode

Een cruciale technische uitdaging is het correct combineren van de alignment transformatie met de initiële centering transformatie. Hoe deze wordt opgebouwd wordt verder besproken in Sectie 3.4.1.

Fase 4: Hybride COLMAP reconstructie. Met de getransformeerde camera poses en de LiDAR-geometrie in COLMAP-formaat wordt een nieuwe reconstructie gemaakt. Deze combineert de metrische nauwkeurigheid van de LiDAR-puntenwolk met de camera trajecten uit de video-analyse. Het resultaat is een COLMAP sparse folder die dezelfde structuur heeft als een standaard SfM-reconstructie, maar waarbij de geometrie vastligt op LiDAR-kwaliteit.

Fase 5: Gaussian Splatting training. In deze methode worden er twee parallelle Gaussian Splatting trainingen gedaan. De baseline variant gebruikt de oorspronkelijke COLMAP-reconstructie uit Fase 2 en volgt de standaard training procedure waarbij zowel de positie, kleur, opacity als covariantie van elke Gaussian geoptimaliseerd worden. De constrained variant daarentegen start met de hybride reconstructie uit fase 4 en beperkt de optimalisatie: de positie van elke Gaussian blijft gefixeerd op de LiDAR-geometrie, terwijl alleen kleur, opacity en covariantie worden getraind. Hierdoor kan het model wel een visueel overtuigende weergave genereren, maar kan het geen geometrie bijverzinzen die niet in de originele LiDAR-data aanwezig was.

3.2 Data-acquisitie

Om de voorgestelde methode te testen, werd dezelfde binnenruimte op twee manieren vastgelegd: met een RGB-camera en met een LiDAR-scanner. Beide datasets werden onafhankelijk van elkaar verzameld, maar met als doel het reconstrueren van dezelfde scène. Hieronder wordt toegelicht hoe beide datasets werden verzameld.

3.2.1 Video-opname van de scène

Doel en rationale

De video-opname heeft meerdere functies. In de eerste plaats levert ze RGB-kleurinformatie die nodig is voor 3D-gaussian splatting. Daarnaast bevat de video informatie over de camerabeweging, die met Structure from Motion-algoritmen zoals COLMAP kan worden gebruikt om de cameraposes te bepalen.

Opname-apparatuur

De visuele data werd initieel verkregen door middel van een handheld smartphonecamera. Voor dit onderzoek werd gebruik gemaakt van een Samsung Galaxy a71.

Met deze camera werd een continue video-opname gemaakt van het lokaal, waarbij de ruimte vanuit verschillende invalshoeken en op verschillende afstanden werd vastgelegd. Tijdens de opname werd erop gelet dat alle relevante delen van de scène meerdere malen in beeld kwamen, zodat er voldoende overlap tussen opeenvolgende frames aanwezig was voor een stabiele reconstructie.

In een later stadium werd overgeschakeld naar een Canon EOS R5 C. De intrinsieke camera-eigenschappen zijn relevant voor de SfM-reconstructie:

- Resolutie: 8192x4320 pixels
- Framerate: 30 frames/seconde
- Lens: 35,40MP Canon RF mount
- Bestandsformaat: .mp4 met H.264 (HEVC) codec

De voornaamste reden hiervoor was de aanwezige motion blur in de smartphone-opnames: doordat de Samsung Galaxy A71 bij handheld gebruik beperkte controle biedt over sluitertijd en een kleinere beeldsensor heeft, resulteerden snelle camerabewegingen in onscherpe frames die de kwaliteit van de SfM-reconstructie negatief beïnvloedden. De Canon EOS R5 C bood in dit opzicht een merkbare verbetering, wat zich vertaalde in scherpere invoerframes en stabielere initiële reconstructies in COLMAP.

Output

De video-opname werd opgeslagen als MP4-bestand en werd nadien opgesplitst in individuele beeldframes, die gebruikt werden als input voor COLMAP. Deze frames vormen de basis voor het schatten van cameratransformaties en het opbouwen van een sparse 3D-puntenwolk. De video wordt in deze context enkel gebruikt als bron van visuele informatie.

3.2.2 LiDAR-scan van de scène

Doel en rationale

De LiDAR-scan levert de geometrische ground truth. LiDAR meet afstanden door de tijdsduur te meten tussen het uitzenden en ontvangen van laserpulsen. Dit resulteert in een metrisch nauwkeurige puntenwolk waarbij elk punt een exacte 3D-coördinaat heeft in het coördinatensysteem van de scanner. Deze nauwkeurigheid is essentieel voor forensische toepassingen waar afstandsmetingen juridische implicaties kunnen hebben.

Scan-apparatuur

Voor de LiDAR-scan werd gebruik gemaakt van de Leica RTC360 3D Laser Scanner. De technische specificaties van deze scanner zijn:

- Bereik: 0,5 - 130 m
- Nauwkeurigheid:
 - Hoeknauwkeurigheid: 18"
 - Bereiknauwkeurigheid: 1,0 mm + 10 ppm
 - 3D-puntnauwkeurigheid: 1,9 mm @ 10 m 2,9 mm @ 20 m 5,3 mm @ 40 m
- Punt acquisitie snelheid: Tot 2.000.000 punten per seconde
- Field of View: 360° horizontaal en 300° verticaal
- Bestandsformaat: .e57

De LiDAR-scan werd uitgevoerd in dezelfde ruimte als de video-opname, waarbij de scanner zodanig werd geïmponeerd dat een zo volledig mogelijke dekking van de scène werd bekomen. Dit werd gedaan door negen scans in de ruimte uit te voeren waarna ze allemaal werden gemerged tot één grote puntenwolk. De resulterende puntenwolk bevat zowel structurele elementen als objecten die aanwezig waren in de ruimte op het moment van de scan.

Output

De scan werd opgeslagen in .e57 formaat. Dit formaat is een industriestandaard voor uitwisseling van 3D-scans. In dit onderzoek wordt deze LiDAR-puntenwolk gebruikt als geometrische referentie voor de verdere verwerking. Alle latere reconstructies en visualisaties worden geometrisch vastgelegd aan deze dataset. Een voorbeeld output is terug te vinden in Figuur 3.2.



Figuur 3.2: LiDAR-scan output bibliotheekdataset beschreven in Sectie 4.1.1

3.2.3 Consistentie tussen video- en LiDAR-data

Hoewel de video-opname en de LiDAR-scan met verschillende sensoren en op verschillende momenten zijn gemaakt, tonen beide datasets dezelfde fysieke ruimte. We gaan ervan uit dat de algemene structuur van de scène niet is veranderd tussen beide opnames, waardoor een eenvoudige rigid-bodytransformatie voldoende is om beide datasets in hetzelfde coördinatenstelsel samen te brengen.

Omdat we de datasets uitsluitend combineren op basis van ruimtelijke overeenkomsten, was er geen expliciete tijdsynchronisatie tussen de video- en LiDAR-data nodig. Eventuele bewegende elementen of kleine verschillen tussen beide opnames worden in dit onderzoek als verwaarloosbaar beschouwd.

3.3 Initiële reconstructie met Structure from Motion

Na de data-acquisitie wordt de video-opname verwerkt met COLMAP om een initiële 3D-reconstructie te verkrijgen. Deze reconstructie is belangrijk voor twee doeleinden: ten eerste levert het de cameraposities die nodig zijn voor de latere alignment met de LiDAR-data en ten tweede dient het als baseline voor vergelijking met de constrained Gaussian Splatting methode.

3.3.1 Video preprocessing

De opgenomen video werd opgesplitst in afzonderlijke beeldframes die gebruikt werden als input voor COLMAP. Dit werd gedaan met behulp van ffmpeg: `ffmpeg -i input.mp4 -qscale:v 1 -qmin 1 -vf fps=1 input/%04d.jpg`. Door gebruik te maken van een video-opname kon er veel overlap tussen opeenvolgende beelden worden gegarandeerd, wat belangrijk is voor goede feature-detection en -matching.

Er werd geen aanvullende beeldbewerking toegepast, zoals geometrische correctie of artificiële verscherping, om de oorspronkelijke visuele informatie zoveel mogelijk te behouden. De resulterende beeldset werd rechtstreeks ingevoerd in COLMAP voor verdere verwerking.

3.3.2 COLMAP-pipeline

Feature extraction

De eerste stap in de COLMAP-pipeline is het detecteren van kenmerkende punten in elke frame. Deze features zijn beeldlocaties zoals hoeken, randen of texturen die goed te herkennen zijn in meerdere views. Belangrijke parameterkeuzes hiervoor waren:

- Het PINHOLE cameramodel gebruiken in plaats van SIMPLE_RADIAL omdat de beelden al undistorted zijn [Joh25]
- De optie `Shared for all images` aanvinken om aan te geven dat elke image met dezelfde camera is gemaakt en dus ook dezelfde intrinsieken hebben.

Het resultaat van deze stap is een database met voor elk frame de gedetecteerde features. Hieruit kwamen voor de bibliotheekdataset, beschreven in Sectie 4.1.1, 1 050 493 features uit 1209 images.

Feature matching

Na feature extraction volgt feature matching, waarbij COLMAP bepaalt welke features in verschillende frames overeenkomen met hetzelfde 3D-punt in de scène. De exhaustive matcher werd gebruikt om maximale nauwkeurigheid te garanderen. Deze matcher vergelijkt elk frame met elk ander frame. Hoewel dit computationeel duur is, zorgt het er wel voor dat ook niet-sequentiële matches worden gevonden, wat belangrijk is als de camera terugkwam op eerder bezochte locaties.

Sparse reconstruction

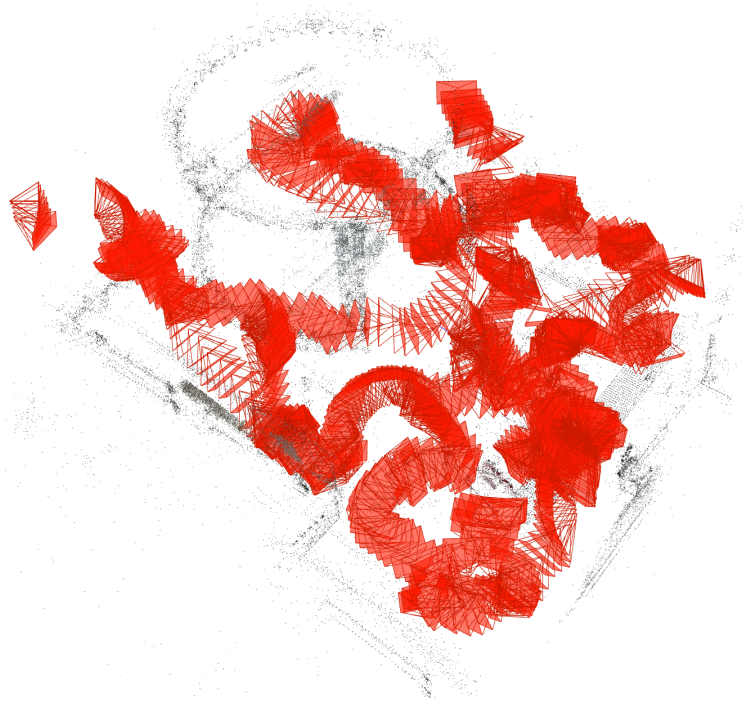
Met de feature matches wordt vervolgens de 3D-reconstructie uitgevoerd. Dit proces schat tegelijkertijd ook de verschillende camera poses. Naast deze camera poses genereert COLMAP een sparse puntenwolk, opgebouwd uit de gereconstrueerde 3D-posities van gematchte beeldfeatures. Deze puntenwolk biedt een eerste geometrische indicatie van de structuur van de scène, maar is beperkt in dichtheid en metrische nauwkeurigheid. Een voorbeeld van een sparse reconstructie is te zien in Figuur 3.3. Hierop is de sparse puntenwolk zichtbaar samen met de geschatte cameraposities in het rood.

In dit onderzoek wordt deze sparse puntenwolk niet gebruikt als finale geometrische representatie. In plaats daarvan dient deze als hulpmiddel voor het bepalen van de ruimtelijke relatie tussen de COLMAP reconstructie en de LiDAR-puntenwolk. De sparse puntenwolk vormt dus de basis voor de aligneringsstap die in Sectie 3.4 wordt beschreven.

Het sparse reconstruction proces resulteerde in:

- Geregistreerde beelden: 1001 van de 1209 frames (82,8%)
- Sparse puntenwolk: 87539 punten

Frames die niet konden worden geregistreerd bevatten waarschijnlijk onvoldoende kenmerkende features of hadden te weinig overlap met de rest van de scène. Dit is acceptabel zolang de belangrijkste delen van de scène gedekt zijn.



Figuur 3.3: Colmap reconstructie bibliotheekdataset beschreven in Sectie 4.1.1

3.4 Alignment en transformatie pipeline

Deze sectie beschrijft de kern van de methode: het aligneren van de sparse SfM-reconstructie met de dense LiDAR-puntenwolk en het transformeren van alle data naar een gemeenschappelijk, metrisch correct coördinatensysteem. Deze stap is cruciaal omdat het de forensische betrouwbaarheid van de LiDAR-scan behoudt. Hierdoor kunnen er betrouwbare afstanden uit gemeten worden en geometrisch correcte projecties en analyses op uitgevoerd worden.

De alignment pipeline bestaat uit drie hoofdcomponenten. Ten eerste wordt de sparse COLMAP-puntenwolk gealigneerd met de dense LiDAR-scan om de transformatie te bepalen. Vervolgens worden de camera poses getransformeerd naar het LiDAR-coördinatensysteem met behoud van hun intrinsieke eigenschappen. Ten slotte wordt de LiDAR-data geconverteerd naar een COLMAP-compatibel formaat waarbij uitgebreide filtering wordt toegepast om de dataset hanteerbaar en robuust te maken. Deze stappen werden geïmplementeerd in een custom Python pipeline met gebruik van Open3D ([ZPK18]) voor puntenwolk manipulatie, NumPy ([HMvdW⁺20]) voor matrix operaties, en PyE57 ([Car25a]) voor E57 bestandsverwerking.

3.4.1 Aligneren van sparse puntenwolk met LiDAR

Voordat alignment kan plaatsvinden, moet de COLMAP-puntenwolk worden gefilterd voor outliers. SfM-reconstructies bevatten vaak foutieve 3D-punten door incorrecte feature matches, vooral in gebieden met repetitieve patronen of weinig textuur. Deze outliers kunnen de alignment sterk verstoren. De filtering strategie combineert twee complementaire methoden die samen een robuuste outlier detectie realiseren.

De eerste methode gebruikt radiale outlier detectie waarbij punten die ver van het centrum van de puntenwolk liggen worden verwijderd. Deze methode berekent de Interquartile Range (IQR) van de afstanden tot het centrum. Concreet wordt het 25e percentiel (Q1) en het 75e percentiel (Q3) van alle afstanden berekend, waarna de IQR wordt gedefinieerd als het verschil tussen deze twee waarden. Punten worden vervolgens als outliers beschouwd wanneer hun afstand tot het

centrum buiten het interval $[Q1 - 3.0 \times IQR, Q3 + 3.0 \times IQR]$ valt. Deze threshold zorgt ervoor dat alleen extreme outliers worden verwijderd.

De tweede methode voert axis-aligned outlier detectie uit, waarbij voor elke dimensie (X, Y, Z) afzonderlijk extreme waarden worden geïdentificeerd. Voor elke as wordt opnieuw een IQR berekend, maar nu met een striktere threshold van $1.5 \times IQR$ in plaats van $3.0 \times IQR$. Dit helpt vooral tegen punten die in één specifieke richting uitsteken, bijvoorbeeld door fouten in verticale triangulatie. Door beide filters te combineren ontstaat een robuuste outlier filtering die valide punten behoudt terwijl extreme afwijkingen worden verwijderd. In de praktijk werden 15,7% van de punten als outliers geïdentificeerd en verwijderd.

Voor een stabiele alignment worden beide puntenwolken eerst gecentreerd. In een eerste stap wordt de sparse cloud vertaald zodat haar bounding box center samenvalt met dat van de dense cloud. Deze centering translatie wordt apart opgeslagen omdat ze later gecombineerd moet worden met de alignment transformatie voor de correcte camera transformaties. Intern in de alignmentfunctie wordt daarna een tweede centering uitgevoerd waarbij zowel source als target elk afzonderlijk op hun eigen zwaartepunt gecentreerd worden. Wanneer beide puntenwolken rond de oorsprong gecentreerd zijn, moet het algoritme enkel nog een rotatie en kleine verschuiving zoeken, zonder dat grote coördinatenwaarden de berekeningen bemoeilijken. Dit maakt het optimalisatieprobleem eenvoudiger en numeriek stabiel, wat snellere en nauwkeurigere convergentie oplevert.

Omdat de COLMAP-reconstructie een arbitraire schaal heeft die sterk kan afwijken van de metrische schaal van de LiDAR-scan, wordt voor de eigenlijke registratie een expliciete schaal-schatting uitgevoerd. De schaalfactor wordt bepaald als de verhouding tussen de bounding box-diagonalen van de gedownsamplende target en source puntenwolk. Deze schaalfactor wordt vervolgens toegepast op alle coördinaten van de source cloud, zodat de twee wolken globaal dezelfde grootte hebben voordat de rotatie-alignment begint.

Binnen de alignmentfunctie worden ook op beide puntenwolken statistical outlier removal-filters toegepast (30 burens, drempel 2.0 standaarddeviaties), als aanvulling op de eerder beschreven IQR-filtering op de originele COLMAP-punten. Deze extra stap verwijdert eventuele resterende outliers op de intern gedownsamplende versies van de puntenwolken.

Voor de eigenlijke alignment wordt een feature-based variant van ICP, Iterative Closest Point ([Juy23]), gebruikt. Standaard ICP is punt-gebaseerd en kan vastlopen in lokale minima, vooral wanneer de initiële poses sterk verschillen. Door gebruik te maken van FPFH, Fast Point Feature Histograms ([Rad09]), wordt de alignment robuuster omdat deze descriptor de geometrische eigenschappen van lokale oppervlakken beschrijven in plaats van alleen punt-posities met elkaar te vergelijken. Het alignment proces verloopt in drie opeenvolgende fasen: feature extraction, global registration met RANSAC, en refinement met een tweefasige point-to-plane ICP-strategie.

In de eerste fase worden voor beide puntenwolken FPFH features berekend. FPFH is een lokale geometrische descriptor die de oriëntatie en kromming van het oppervlak rondom elk punt beschrijft. Om deze features te kunnen berekenen, moeten eerst de oppervlakte-normalen worden geschat voor elk punt. Dit gebeurt door een lokale buurt rondom elk punt te analyseren en het vlak te fitten dat deze punten het beste benadert. De normaal is dan de vector loodrecht op dit vlak. Voor de normaal-schatting wordt een `voxel_size` parameter gebruikt die de schaal bepaalt waarop features worden berekend. Een grotere waarde maakt de features robuuster tegen ruis maar minder discriminatief voor details.

Na de normaal-schatting worden de FPFH features berekend met een grotere search radius dan gebruikt werd voor de normalen. Dit zorgt ervoor dat de features informatie bevatten over een grotere lokale omgeving. De resulterende FPFH descriptor zijn 33-dimensionale vectoren die de geometrische eigenschappen van het lokale oppervlak coderen op een manier die invariant is voor rotaties en translaties.

In de tweede fase wordt een grove alignment uitgevoerd via RANSAC, Random Sample Consensus ([Mar81]). RANSAC is een robuuste schattingsmethode die goed werkt in de aanwezigheid

van outliers. Het algoritme werkt iteratief: in elke iteratie wordt een willekeurige subset van punt-correspondences geselecteerd, een transformatie geschat op basis van deze subset, en vervolgens geëvalueerd hoeveel van alle correspondences consistent zijn met deze transformatie. De transformatie met de meeste inliers wordt als beste beschouwd. Voor het matchen van punten worden de FPFH-descriptoren gebruikt zonder mutual filtering, waarbij een match alleen geldig is wanneer punt A in cloud 1 het meest lijkt op punt B in cloud 2 en punt B het meest lijkt op punt A. Het weglaten van deze filter kan het aantal false matches verhogen, maar vergroot tegelijk de kans dat correcte correspondences worden meegenomen. Er worden twee validatiechecks toegepast: een edge length check die controleert of de relatieve afstanden tussen punten behouden blijven door de transformatie en een distance check die nagaat of gematchte punten na transformatie daadwerkelijk dicht bij elkaar liggen.

In de derde fase wordt de RANSAC-transformatie verder verfijnd met een tweefasige ICP-strategie. In de eerste fase werd een relatief grote corresponderende afstand gehanteerd, gelijk aan vijf keer de voxelgrootte van de puntenwolken. Dit stelde het algoritme in staat om de source puntenwolk globaal op de target te positioneren, zelfs als er nog aanzienlijke afwijkingen aanwezig zijn. Als initiële transformatie werd de identiteitsmatrix gebruikt, en de schatting van de rigide transformatie gebeurde via de point-to-plane methode, waarbij punten uit de source naar de oppervlakken van de target werden geprojecteerd. Deze point-to-plane variant van ICP minimaliseert niet alleen de punt-tot-punt afstand maar ook de afstand langs de oppervlakte-normaal, wat resulteert in betere alignment van vlakke oppervlakken zoals muren en vloeren. De iteraties werden beperkt tot maximaal 100, waarna de verkregen transformatie op de source werd toegepast om een ruwe uitlijning te verkrijgen. De kwaliteit van deze coarse stap werd geëvalueerd aan de hand van de fitness, die de fractie van punten in de source meet die binnen de gespecificeerde afstand van de target liggen.

Op basis van deze ruwe uitlijning werd vervolgens de fine ICP uitgevoerd, met een corresponderende afstand van tweemaal de voxelgrootte. Dit stelde het algoritme in staat om subtiele lokale correcties door te voeren en de puntenwolken nauwkeurig uit te lijnen. Om de stabiliteit van de uitlijning te waarborgen, werden aanvullende stopcriteria ingesteld op basis van minimale verbeteringen in fitness en root mean square error (RMSE), terwijl het maximale aantal iteraties werd verhoogd tot 500. Ook in deze fase werd de point-to-plane methode gebruikt. Na afronding van de fine ICP wordt de volledige finale transformatiematrix opgebouwd door alle afzonderlijke stappen te combineren via matrixvermenigvuldiging:

$$\mathbf{T}_{\text{final}} = \mathbf{T}_{\text{back}} \cdot \mathbf{T}_{\text{ICP, fine}} \cdot \mathbf{T}_{\text{ICP, coarse}} \cdot \mathbf{T}_{\text{RANSAC}} \cdot \mathbf{T}_{\text{scale}} \cdot \mathbf{T}_{\text{origin}} \quad (3.1)$$

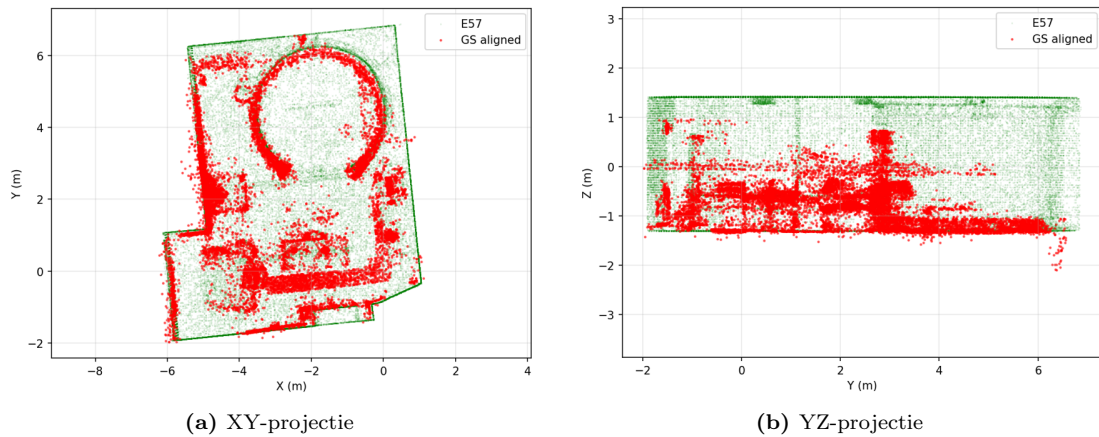
waarbij $\mathbf{T}_{\text{origin}}$ de oorspronkelijke centrering van de source-puntenwolk voorstelt, $\mathbf{T}_{\text{scale}}$ de schaalcorrectie op basis van de bounding box-diagonaalverhouding, $\mathbf{T}_{\text{RANSAC}}$ de grove RANSAC alignment, $\mathbf{T}_{\text{ICP, coarse}}$ en $\mathbf{T}_{\text{ICP, fine}}$ respectievelijk de coarse en fine ICP-verfijning en \mathbf{T}_{back} de terugvertaling naar het centrum van de target-puntenwolk.

Na de alignment wordt automatisch een rotatie correctie uitgevoerd. De alignment transformatie bevat naast schaal en translatie ook een rotatie-component die de oriëntatie van de COLMAP-puntenwolk ten opzichte van de LiDAR-scan aanpast. Voor de Gaussian Splatting training is een specifieke oriëntatie strikt genomen niet vereist, maar in de praktijk is het wenselijk dat de gealignde puntenwolk dezelfde opstaande oriëntatie behoudt als de originele COLMAP-reconstructie. Een gestandaardiseerde oriëntatie vergemakkelijkt namelijk de navigatie in de gereconstrueerde scène, zowel tijdens inspectie als bij forensische analyse. Daarom wordt de rotatie-component uit de finale transformatie geëxtraheerd via polaire decompositie, waarbij de 3×3 bovenlinkerdeelmatrix wordt gesplitst in een pure rotatiematrix en een symmetrische schaalmatrix. De inverse rotatie (R^T) wordt vervolgens toegepast op de gealignde puntenwolk, gecentreerd rond haar zwaartepunt, zodat de oriëntatie gecorrigeerd wordt terwijl schaal en positie volledig behouden blijven. Deze rotatie correctie wordt opgeslagen als een 4×4 matrix en later ook toegepast op de camera poses en de LiDAR-puntenwolk.

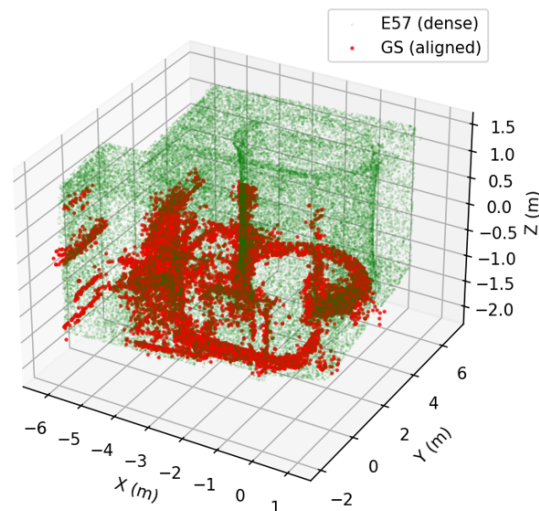
Voor wetenschappelijke reproduceerbaarheid wordt een vaste random seed gebruikt voor alle stochastische operaties in de pipeline. Dit garandeert dat de RANSAC-stap en andere willekeurige processen deterministisch verlopen, wat essentieel is voor reproduceerbaarheid van de resultaten.

Het resultaat van de alignment is een 4×4 homogene transformatiematrix die de relatie tussen de twee coördinatensystemen beschrijft. Deze matrix bestaat uit een 3×3 rotatiematrix die de oriëntatie transformeert en een 3×1 translatievector die de positie transformeert. De onderste rij van de matrix is altijd $[0, 0, 0, 1]$ voor homogene coördinaten. Deze transformatiematrix wordt opgeslagen voor gebruik in de volgende stappen van de pipeline.

Na alignment wordt een visuele controle uitgevoerd met matplotlib 3D plots ([Hun07]). Deze visualisatie toont de originele COLMAP-puntenwolk, de dense LiDAR-puntenwolk (gesubsampled voor visualisatie), en de gealigneerde COLMAP-puntenwolk in verschillende kleuren. Dit maakt het mogelijk om visueel te verifiëren dat de alignment correct is. Specifieke aandachtspunten om te controleren zijn overeenkomstige muren, objecten zoals tafels en stoelen die op dezelfde positie staan en of er geen verschuivingen of rotaties zichtbaar zijn in de gealigneerde puntenwolk. Voorbeelden van de visuele vergelijkingen voor de bibliotheekdataset, beschreven in Sectie 4.1.1, zijn terug te vinden in Figuur 3.4 en Figuur 3.5.



Figuur 3.4: Projectie POV alignment output bibliotheekdataset beschreven in Sectie 4.1.1



Figuur 3.5: Globale alignment output bibliotheekdataset beschreven in Sectie 4.1.1

3.4.2 Transformatie van camera-extrinsieken

Nu de transformatie van sparse naar dense puntenwolk bekend is, moet deze worden toegepast op de camera posities uit COLMAP. Dit is technisch complex omdat camera posities een specifieke wiskundige structuur hebben die behouden moet blijven tijdens de transformatie.

In COLMAP wordt elke camera pose beschreven door een quaternion en een translatievector. De quaternion (QW, QX, QY, QZ) representeert de rotatie van de camera, terwijl de translatievector (TX, TY, TZ) de positie beschrijft. Een belangrijk detail is dat in COLMAP's conventie deze parameters de transformatie van wereld-coördinaten naar camera-coördinaten beschrijven. Dit betekent dat het camera center C in wereldcoördinaten wordt gegeven door $C = -R^T * t$, waarbij R de 3×3 rotatiematrix is afgeleid uit de quaternion en t de translatievector is. ([Sch25])

Een klein detail maar belangrijke vaststelling is dat de alignment transformatie mogelijk een schaalfactor bevat. Dit komt omdat COLMAP's reconstructie een arbitraire schaal heeft, de afstanden zijn relatief correct maar de absolute schaal is onbekend. De schaal wordt uit de transformatiematrix gehaald door de determinant van de rotatie-component te berekenen en hier de kubieke wortel van te nemen. Deze schaalfactor geeft aan hoeveel groter of kleiner de LiDAR-wereld is ten opzichte van de COLMAP-wereld. Een schaal van bijvoorbeeld 47,3 betekent dat één COLMAP-eenheid overeenkomt met 47,3 meter in het LiDAR-systeem. Na extractie van de schaal wordt de rotatie-component van de transformatiematrix gedeeld door deze schaalfactor om een pure rotatiematrix te verkrijgen zonder schaling.

De volledige transformatie voor cameras moet beide transformaties combineren: de initiële centering en de alignment transformatie. Dit gebeurt door matrixvermenigvuldiging. Eerst wordt een 4×4 identiteitsmatrix gemaakt en de centering translatievector wordt in de vierde kolom geplaatst. Vervolgens wordt deze centering matrix rechts-vermenigvuldigd met de alignment matrix, wat resulteert in de volledige transformatie. De volgorde is hier belangrijk: eerst moet de sparse cloud worden gecentreerd zoals gedaan voor alignment, dan moet de alignment transformatie worden toegepast.

Voor elke camera in het COLMAP images.txt bestand wordt vervolgens de transformatie toegepast. Allereerst worden de COLMAP parameters geparsed: de quaternion en translatievector worden uit het tekstbestand gelezen. De quaternion wordt geconverteerd naar een 3×3 rotatiematrix met behulp van standaard quaternion-naar-matrix conversieformules. Vervolgens wordt het camera center berekend met $C = -R^T * t$.

Het camera center wordt nu getransformeerd naar het LiDAR-coördinatensysteem. Eerst wordt het center geschaald met de eerder gevonden schaalfactor, wat het center converteert van COLMAP-eenheden naar meters. Daarna wordt het rotatie-component van de transformatie toegepast en het translatie-component toegevoegd, wat resulteert in het nieuwe camera center in LiDAR-coördinaten.

De camera-oriëntatie moet ook worden getransformeerd. Dit gebeurt door de originele rotatiematrix rechts-te-vermenigvuldigen met de getransponeerde van de transformatie-rotatie. Op die manier wordt de rotatie correct toegepast, zodat de camera in het nieuwe coördinatenstelsel dezelfde richting behoudt ten opzichte van de scène.

Na transformatie is het mogelijk dat de nieuwe rotatiematrix niet meer perfect orthogonaal is door numerieke fouten. Een rotatiematrix moet orthonormaal zijn, wat betekent dat de kolommen orthogonaal unit vectors moeten zijn, en de determinant moet precies 1 zijn. Indien de determinant significant afwijkt van 1.0, wordt SVD-based orthogonalisatie toegepast. Singular Value Decomposition ontbindt de matrix in drie componenten, en door de singuliere waarden te vervangen door éénen wordt een perfecte rotatiematrix verkregen. Als de determinant negatief is na SVD, wat een reflectie zou impliceren, wordt het teken van de laatste rij omgekeerd om te garanderen dat de determinant 1 is.

Vervolgens wordt de nieuwe translatie berekend voor COLMAP's conventie met $t_{new} = -R_{new} * C_{new}$. De nieuwe rotatiematrix wordt geconverteerd terug naar een quaternion

met behulp van standaard matrix-naar-quaternion conversieformules. Een belangrijk detail is dat scipy's quaternion representatie de volgorde (x, y, z, w) gebruikt terwijl COLMAP (w, x, y, z) verwacht, dus de volgorde moet worden aangepast bij het schrijven naar het output bestand.

De nieuwe quaternion moet ook genormaliseerd worden om te garanderen dat deze een unit quaternion is. De norm wordt berekend als de wortel van de som van de gekwadeerde componenten, en als deze fel afwijkt van 1.0 worden alle componenten gedeeld door de norm.

Indien een rotatie correctie matrix beschikbaar is, wordt deze ook toegepast op elke camera. Het camera center wordt geroteerd rond het centroid van de gealignde puntenwolk met de correctiematrix R_{corr} : $C_{new} = R_{corr}(C-c)+c$, waarbij c het rotatiepunt is. De camera-oriëntatie wordt aangepast door de rotatiematrix rechts te vermenigvuldigen met de getransponeerde van R_{corr} , zodat de camera in de goede richting blijft kijken ten opzichte van de gecorrigeerde scène. Op die manier zijn zowel de puntenwolk als de camera poses na alle stappen consistent in hetzelfde goed georiënteerde coördinatensysteem.

Het resultaat is een nieuwe images.txt met getransformeerde camera posities. Elke camera blijft geassocieerd met dezelfde image ID, camera ID (de intrinsics blijven ongewijzigd), bestandsnaam, en 2D-3D correspondences. Alleen de extrinsics, de rotatie en translatie, zijn getransformeerd naar het LiDAR-coördinatensysteem. In de praktijk werden 1001 camera's succesvol getransformeerd.

De camera intrinsics blijven ongewijzigd omdat deze eigenschappen zijn van de camera zelf, niet van de wereldpositie. Het bestand cameras.txt wordt daarom letterlijk gekopieerd naar de output directory zonder modificaties. Dit is correct omdat parameters zoals focal length en principal point alleen afhangen van de lens en sensor, niet van waar de camera zich in de wereld bevindt.

Een overzicht van deze transformatiepipeline is terug te vinden in Figuur 3.6.

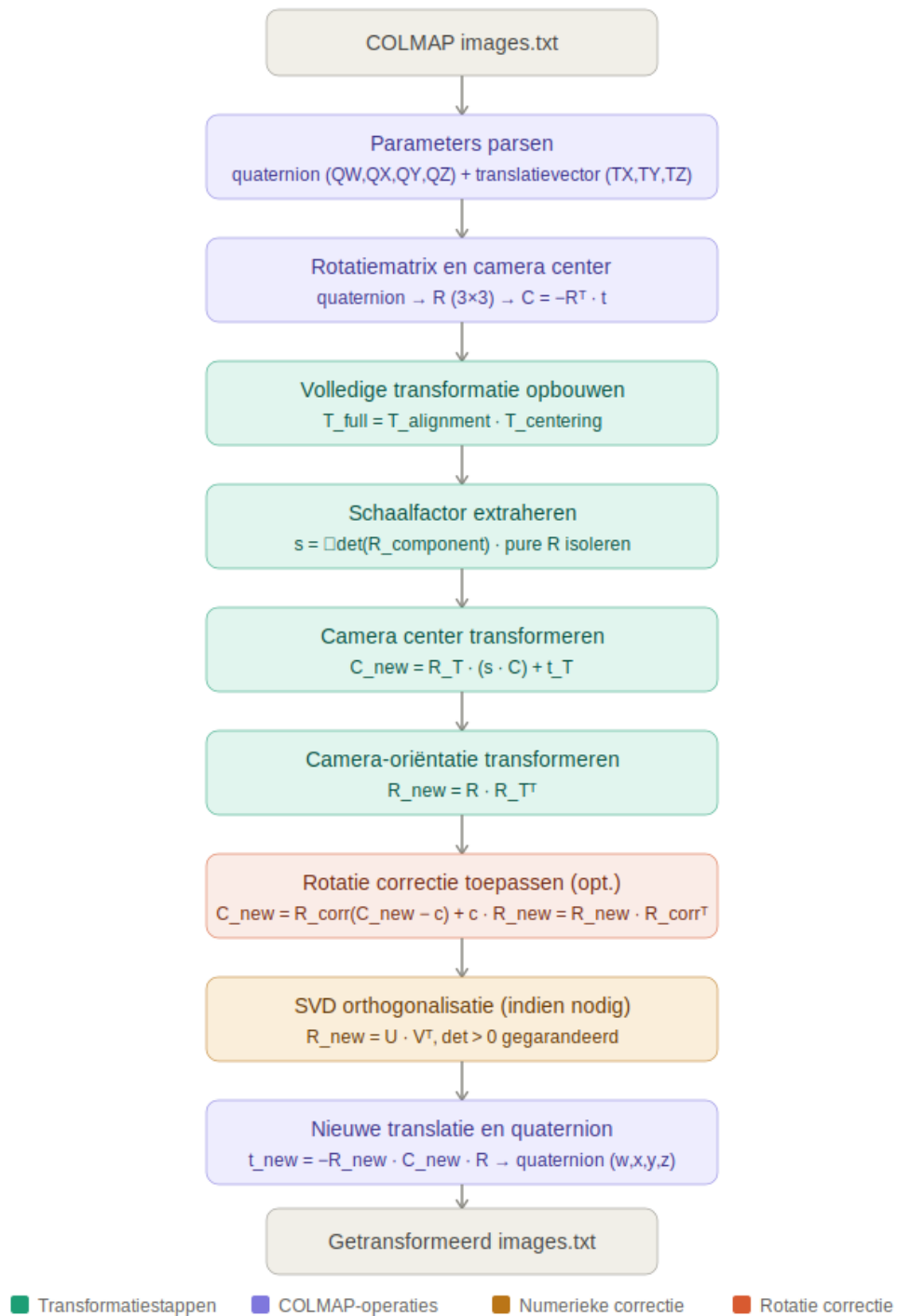
3.4.3 Conversie van LiDAR naar COLMAP-compatible formaat

De laatste stap is het converteren van de E57 LiDAR-scan naar COLMAP's points3D.txt formaat. Dit is niet-triviaal omdat E57 bestanden miljoenen tot honderden miljoenen punten kunnen bevatten, wat te groot is voor efficiënte verwerking door COLMAP en Gaussian Splatting. De LiDAR scans bevatten daarnaast ook vaak ruis en outliers die moeten worden gefilterd. De conversie pipeline omvat daarom meerdere stappen die de dataset reduceren tot een hanteerbare grootte terwijl de geometrische kwaliteit behouden blijft en rekening wordt gehouden met de zichtbaarheid vanuit de cameraposities.

E57 is een gestandaardiseerd formaat volgens ASTM E2807 voor uitwisseling van 3D imaging data ([AST22]). Een E57 bestand kan meerdere scans bevatten van verschillende scannerposities, elk met Cartesiaanse coördinaten per punt, optioneel RGB kleuren, en optioneel intensiteitswaarden die de reflectie-sterkte weergeven. Voor dit onderzoek werden 9 scans gebruikt die samen 566 087 485 punten bevatten voor de gehele ruimte.

Het E57 bestand wordt geladen met de pye57 library die specifiek is ontworpen voor het lezen en schrijven van dit formaat. Voor elke scan worden de XYZ-coördinaten geëxtraheerd uit de cartesianX, cartesianY en cartesianZ velden. Indien beschikbaar worden ook de RGB kleuren geëxtraheerd uit de colorRed, colorGreen en colorBlue velden. De kleurwaarden worden genormaliseerd door te delen door 255 om waarden tussen 0 en 1 te verkrijgen zoals COLMAP verwacht. Tijdens het laden worden ook ongeldig punten gefilterd door te controleren op NaN (Not a Number) en Inf (oneindig) waarden, die kunnen ontstaan door meetfouten of reflecties.

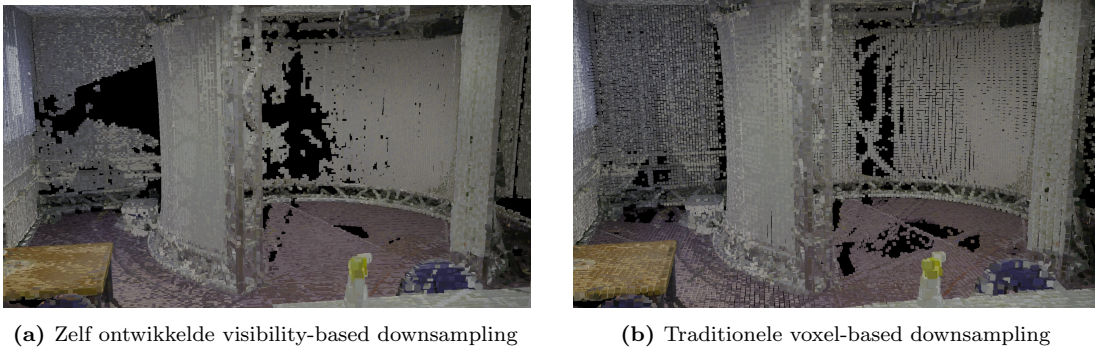
De eigenlijke conversie verloopt via een driefasige pipeline die de puntenwolk reduceert op basis van zichtbaarheid vanuit de gecalibreerde cameraposities. Dit is specifiek voor de juridische context ontworpen omdat speurders intuïtief meer filmen op plaatsen die zij belangrijk vinden



Figuur 3.6: Alle stappen voor de transformatie van de camera-extrinsieken

in een scène. Deze manier van reduceren zorgt er dus automatisch voor dat er meer punten bewaard worden op plekken die het belangrijkst zijn voor het onderzoek.

In Figuur 3.7 wordt de vergelijking getoond tussen de traditionele voxel-based downsampling (rechts) en de zelf ontwikkelde visibility-based downsampling (links) voor de bibliotheekdataset beschreven in Sectie 4.1.1. Een opvallend visueel verschil is dat de voxel-based methode achterin de scène meer punten bewaart, waardoor de grond in het midden meer gaten vertoont. De visibility-based methode daarentegen vult de grond dichter op en laat punten die zich verder op de achtergrond bevinden weg.



Figuur 3.7: Visibility based downsampling vs voxel based downsampling

De eerste twee passes verwerken de geometrie streaming per scan, zodat ook zeer grote E57 bestanden op standaard hardware verwerkt kunnen worden zonder alle data tegelijkertijd in het geheugen te laden. Een globaal overzicht van de volledige visibility-based downsampling pipeline is terug te vinden in Figuur 3.8.

Pass 1: BVH-mesh bouwen

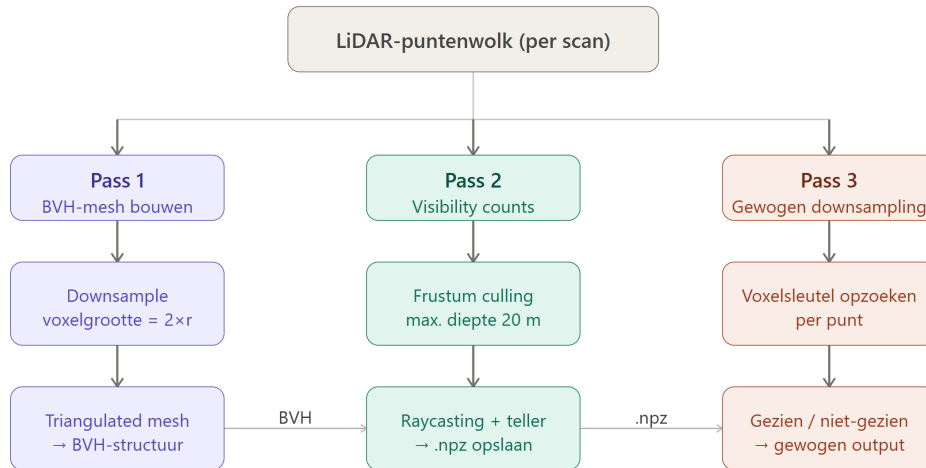
In de eerste pass wordt een BVH-mesh (Bounding Volume Hierarchy mesh) opgebouwd uit de volledige puntenwolk. Deze mesh dient als oclusiemodel voor de zichtbaarheidsberekeningen in de tweede pass. Per scan worden de punten ingeladen, gedownsampled met een grove voxelgrootte gelijk aan tweemaal de oclusiestraal, en toegevoegd aan een gecombineerde puntenwolk. Op basis van deze gecombineerde puntenwolk wordt via Open3D een triangulated mesh geconstrueerd en geladen in een BVH-structuur die efficiënte raycasting mogelijk maakt.

Pass 2: Streaming visibility counts per voxel

In de tweede pass wordt voor elke camerapose bepaald welke voxels van de puntenwolk zichtbaar zijn. Per scan worden de punten ingeladen en gedownsampled naar een grovere resolutie om de raycasting te versnellen. Vervolgens wordt voor elke camera in twee fasen de zichtbaarheid bepaald. In de eerste fase wordt frustum culling toegepast: punten die buiten het zichtfrustum van de camera vallen of verder dan de maximale diepte van 20 meter liggen, worden onmiddellijk uitgesloten. In de tweede fase wordt voor elk kandidaatpunt een oclusiecheck uitgevoerd via raycasting op de BVH-mesh. Een punt wordt als zichtbaar beschouwd wanneer de ray vanaf het cameracentrum naar het punt niet eerder de mesh snijdt dan het punt zelf, binnen een bepaalde tolerantie. De zichtbare punten worden omgezet naar voxelsleutels en de teller van het bijbehorende voxel wordt verhoogd. Zo accumuleert elke voxel het totale aantal camera's waaruit hij zichtbaar is. De resulterende tellerarray wordt opgeslagen als een gecompriemd NPZ-bestand voor gebruik in de derde pass.

Pass 3: Visibility-gewogen voxel downsampling

In de derde pass wordt de uiteindelijke downsampling uitgevoerd, waarbij de visibilitycounts uit Pass 2 als wegingsfactor dienen. Per scan worden de punten inclusief kleurinformatie ingeladen. Voor elk punt wordt via de voxelsleutel de bijbehorende visibilityteller opgezocht, waarna een onderscheid wordt gemaakt tussen *geziene* punten (zichtbaar vanuit minstens één camera) en *niet-geziene* punten.



Figuur 3.8: Globaal overzicht van de visibility-based downsampling pipeline

Geziena punten worden gedownsampld op een fijn voxelgrid met de doelvoxelgrootte van 0.03 meter. Wanneer meerdere punten in dezelfde voxel vallen, wordt het punt met de hoogste visibilityteller geselecteerd in plaats van het gemiddelde te nemen. Dit zorgt ervoor dat punten die vanuit veel camera's zichtbaar zijn de voorkeur krijgen boven minder goed geobserveerde punten in hetzelfde voxel, wat de kwaliteit van de kleurinformatie ten goede komt.

Niet-geziene punten ondergaan een grovere downsampling op een voxelgrid dat een factor 6 groter is dan de doelvoxelgrootte. Hiermee worden gebieden die nooit door een camera werden gezien sterk gereduceerd maar niet volledig verwijderd: ze dragen bij aan de geometrische volledigheid van de puntenwolk maar nemen verhoudingsgewijs veel minder ruimte in. Na verwerking van alle scans worden de resultaten van geziene en niet-geziene punten samengevoegd tot één gecombineerde puntenwolk.

Na de drie passes worden twee complementaire outlier removal methoden toegepast. Hier wordt opnieuw outlier removal toegepast omdat de e57-puntenwolk na het inladen tijdens de alignment niet wordt opgeslagen. Bij de daaropvolgende conversie naar COLMAP-compatibel formaat wordt de puntenwolk immers opnieuw ingeladen vanaf het originele bestand, inclusief de oorspronkelijke outliers. De eerste is statistical outlier removal, een methode die voor elk punt de gemiddelde afstand tot zijn dichtstbijzijnde burens berekent. Over alle punten wordt dan het globale gemiddelde en standaarddeviatie van deze afstanden berekend. Punten waarvan de gemiddelde afstand tot hun burens meer dan een bepaald aantal standaarddeviaties boven het globale gemiddelde ligt worden als outliers beschouwd en verwijderd. Voor dit onderzoek werden 30 burens gebruikt voor de afstandsberkening en een threshold van 1.5 standaarddeviaties. Deze methode is effectief voor het verwijderen van geïsoleerde punten die significant afwijken van hun lokale omgeving, wat typisch wordt veroorzaakt door reflecties, multi-path errors, of stof in de lucht tijdens scanning.

De tweede methode is radius outlier removal, die vereist dat elk punt minimaal een bepaald aantal burens heeft binnen een bepaalde radius. Punten met minder burens worden als outliers beschouwd en verwijderd. Deze methode is complementair aan statistical outlier removal omdat het kleine clusters van punten kan verwijderen die wel intern consistent zijn maar niet verbonden

zijn met de rest van de puntenwolk. Voor dit onderzoek werd een radius van 0.06 meter gebruikt met een minimum van 10 burens. Dit betekent dat elk punt dat overblijft minstens 10 andere punten heeft binnen een straal van 6 centimeter, wat zorgt voor een goed verbonden en dense puntenwolk.

De combinatie van de visibility-gewogen downsampling en beide outlier removal methoden resulteerde in een finale puntenwolk van 1 148 098 punten, een reductie waarbij maar ongeveer 0,20% van de punten behouden wordt ten opzichte van de originele scan. Deze reductie is aanzienlijk maar behoudt alle belangrijke geometrische structuren van de ruimte, met nadruk op de gebieden die daadwerkelijk door de camera's werden waargenomen.

De gefilterde puntenwolk wordt vervolgens geschreven naar COLMAP's `points3D.txt` formaat. Elk punt wordt geschreven als één regel met de volgende informatie: een uniek point ID (sequentieel vanaf 1), de XYZ-coördinaten met zes decimalen, de RGB-kleurwaarden (de corresponderende puntkleur uit de e57 of een neutrale grijswaarde) als integers tussen 0 en 255, een error waarde, en optioneel een track lijst. Voor LiDAR-punten wordt een vaste error waarde van 2.0 gebruikt, wat matige betrouwbaarheid aangeeft. Dit verschilt van sparse SfM-punten die typisch een error tussen 0.5 en 1.0 hebben op basis van hun reprojection error. De hogere error voor LiDAR-punten geeft aan dat ze minder direct geobserveerd zijn in de video frames en mogelijk minder betrouwbaar zijn voor kleurtoekenning.

Een belangrijk verschil met de sparse COLMAP-punten is dat LiDAR-punten geen track informatie hebben. Een track is een lijst van image ID's en 2D pixel-locaties die aangeven in welke beelden een 3D-punt werd geobserveerd. LiDAR-punten zijn direct gemeten door de scanner en niet getrianguleerd uit video frames, dus ze hebben geen natuurlijke track. Het track-veld wordt daarom leeg gelaten in het output bestand.

3.5 Hybride COLMAP reconstructie

Na voltooiing van alle stappen bevat de output directory drie bestanden die samen een geldige COLMAP sparse reconstructie vormen. Het `cameras.txt` bestand bevat de ongewijzigde camera intrinsics. Het `images.txt` bestand bevat de getransformeerde camera poses die correct aligned zijn met het LiDAR-coördinatensysteem. Het `points3D.txt` bestand bevat de dense, gefilterde LiDAR-puntenwolk. Deze drie bestanden kunnen direct gebruikt worden als input voor Gaussian Splatting training zonder verdere preprocessing.

Het resultaat is een hybride reconstructie die de sterktes van beide meettechnieken combineert. De camera poses zijn afkomstig van SfM en bieden nauwkeurige visuele localisatie van elk videoframe in de scène. De puntenwolk is afkomstig van de LiDAR-scanner en biedt een geometrisch accurate, metrisch correcte representatie van de ruimtelijke structuur. Door beide samen te voegen in één gemeenschappelijk coördinatensysteem wordt de zwakte van elke techniek afzonderlijk gecompenseerd: de willekeurige schaal en beperkte dichtheid van de sparse SfM-reconstructie worden vervangen door de absolute metrische nauwkeurigheid en hoge densiteit van de LiDAR-scan, terwijl de visuele informatie van SfM de textuurbinding tijdens Gaussian Splatting training mogelijk maakt.

Een cruciaal aspect van deze hybride aanpak is dat de geometrische basis van de reconstructie forensisch betrouwbaar blijft. Omdat de puntenwolk rechtstreeks afkomstig is van de gekalibreerde LiDAR-scanner en niet getrianguleerd wordt uit videoframes, zijn de 3D-posities van punten onafhankelijk van de beeldkwaliteit, belichtingsomstandigheden of bewegingsartefacten in de video. Afstandsmetingen uitgevoerd op basis van de LiDAR-puntenwolk zijn daardoor metrisch correct en reproduceerbaar, wat een vereiste is voor forensische toepassingen.

De hybride reconstructie wijkt op één belangrijk punt af van een standaard COLMAP sparse reconstructie: de punten in `points3D.txt` hebben geen track-informatie die hen koppelt aan specifieke beeldpixels. In een gewone SfM-reconstructie worden 3D-punten trianguleerd vanuit meerdere beelden en is voor elk punt bekend in welke beelden het zichtbaar is en op welke

pixellocatie. Voor LiDAR-punten ontbreekt deze informatie omdat ze direct gemeten zijn. Dit heeft als gevolg dat de initialisatie van Gaussian Splatting anders verloopt dan bij een puur SfM-gebaseerde aanpak: de 3D-gaussians worden geïnitieerd op basis van de LiDAR-puntposities, en de kleur van de Gaussians worden geïnitieerd op de corresponderende puntkleur. Ontbreekt deze kleurinformatie, dan worden alle punten geïnitieerd met een neutrale grijswaarde (RGB 128, 128, 128). In de praktijk heeft dit voor beide gevallen weinig invloed op de kwaliteit van het uiteindelijke resultaat, omdat Gaussian Splatting sowieso de kleur optimaliseert op basis van de beelddata en niet uitsluitend vertrouwt op de initiële kleurwaarden van de puntenwolk.

De kwaliteit van de hybride reconstructie is afhankelijk van de nauwkeurigheid van de alignment stap. Een misalignment tussen de SfM camera poses en de LiDAR-puntenwolk vertaalt zich direct in een geometrische inconsistentie: de cameras kijken naar een licht verkeerde positie in de puntenwolk. Dit kan leiden tot wazige of verdubbelde structuren in het gerenderde resultaat. De visuele controle beschreven in Sectie 3.4.1 is daarom belangrijk om de kwaliteit van de alignment te verifiëren voordat de data wordt doorgegeven aan het trainingsproces. Kwantitatief wordt de alignmentkwaliteit beoordeeld aan de hand van onder andere de fitness-score en RMSE van de fine ICP-stap, maar visuele inspectie blijft noodzakelijk om systematische fouten te detecteren die niet worden weerspiegeld in deze metriek. Voor de bibliotheekdataset beschreven in Sectie 4.1.1 zijn de kwantitatieve resultaten het volgende:

- RANSAC globale registratie Fitness: 0.9548 (ongeveer 95% van de punten vonden een match)
- ICP Coarse Fitness: 0.9807 (verbetering ten opzichte van RANSAC)
- ICP Coarse RMSE: 0.0595 (gemiddelde afwijking van ongeveer 6 cm)
- ICP Fine Fitness: 0.8952 (gedaald ten opzichte van ICP Coarse, maar normaal bij fine ICP met strengere criteria)
- ICP Fine RMSE: 0.0416 (gemiddelde afwijking verbeterd naar ongeveer 4 cm)

3.6 Gaussian Splatting training

Deze sectie beschrijft de training van twee parallele Gaussian Splatting modellen: een baseline variant die geen geometrische beperkingen heeft, en een constrained variant waarbij de geometrie gefixeerd blijft op de LiDAR-metingen. Beide modellen worden getraind met LichtFeld Studio [Stu25], een open-source, high-performance implementatie van Gaussian Splatting.

3.6.1 Baseline implementatie

Invoerdata en initialisatie

LichtFeld Studio verwacht invoer in COLMAP-formaat. Hier is dit de sparse COLMAP-reconstructie die in Sectie 3.3 is opgebouwd. Dit omvat de camera poses in `images.txt`, de camera intrinsics in `cameras.txt`, en de sparse SfM-puntenwolk in `points3D.txt`. Deze bestanden kunnen dus zonder verdere preprocessing aan Lichtfeld Studio worden doorgegeven.

De initiële posities van de 3D Gaussians worden afgeleid uit de sparse SfM-puntenwolk. Elk punt wordt omgezet naar een initiële Gaussian met een positie gelijk aan de puntcoördinaat en een initiële kleur afgeleid uit de RGB-waarden van het punt. Schaal, rotatie en opacity worden geïnitieerd op standaardwaarden. Doordat de sparse SfM-puntenwolk slechts een beperkte dichtheid heeft, starten de Gaussians op een beperkte geometrische basis. Het model zal gedurende de training via densification nieuwe Gaussians aanmaken om lege gebieden op te vullen.

Een belangrijk gevolg van de sparse initialisatie is dat de Gaussians geen absolute metrische verankering hebben. De SfM-puntenwolk heeft een arbitraire schaal en de posities zijn niet rechtstreeks herleidbaar tot werkelijke afstanden in de scène. Dit is aanvaardbaar voor de

baseline omdat het doel enkel het maximaliseren van de visuele kwaliteit als referentie voor de vergelijking is.

Trainingsproces

Het trainingsproces verloopt iteratief: in elke stap wordt een camera view geselecteerd, wordt het huidige model gerenderd vanuit die view, en worden de parameters van de Gaussians bijgewerkt op basis van het verschil tussen het gerenderde beeld en het overeenkomstige videoframe. LichtFeld Studio minimaliseert een gecombineerde verliesfunctie die een L1-fotometrische verliesterm en een D-SSIM-term combineert ([KKLD23]):

$$\mathcal{L} = (1 - \lambda) \cdot \mathcal{L}_1 + \lambda \cdot \mathcal{L}_{\text{D-SSIM}} \quad (3.2)$$

waarbij λ de relatieve bijdrage van de structurele gelijkheid bepaalt. Tijdens training worden alle parameters van elke Gaussian vrij geoptimaliseerd: de positie, de schaal en oriëntatie van de ellipsoïde, de opacity en de kleur via spherical harmonics coëfficiënten. Gaussians zijn dus vrij om te bewegen ten opzichte van hun initiële positie wanneer dit de verliesfunctie verlaagt, ook al leidt dit tot geometrische afwijkingen ten opzichte van de werkelijke scène.

Voor de baseline worden de standaard trainingsparameters van LichtFeld Studio gebruikt:

- Aantal trainingsiteraties: 30 000
- Optimalisatiestrategie: mcmc (zie Sectie 4.1.2)
- Maximum aantal Gaussians: 1 150 000

Rol van de baseline

De baseline dient als referentiepunt voor de evaluatie in Sectie 4. Doordat er geen bijkomende beperkingen worden opgelegd en de initialisatie alleen op de sparse SfM-puntenwolk is gebaseerd, is de baseline vrij om de visuele kwaliteit te maximaliseren. Dit kan echter ten koste gaan van de metrische nauwkeurigheid. Het model kan Gaussians verplaatsen om reflecties of belichting beter te modelleren, ook al komt de geometrie dan niet meer overeen met de werkelijke afmetingen van de scène.

3.6.2 Constrainede implementatie

Invoerdata en initialisatie

De constrained variant bouwt verder op de baseline maar gebruikt de hybride COLMAP-reconstructie uit Sectie 3.4 als invoer. Dit omvat de getransformeerde camera poses, de ongewijzigde camera intrinsics, en de dense gefilterde LiDAR-puntenwolk in `points3D.txt`. Dezelfde COLMAP-datastructuur wordt gebruikt als bij de baseline, zodat het enige verschil in invoer de puntenwolk is.

Dit heeft een directe impact op de initialisatie. Waar de baseline start vanuit een sparse puntenwolk met arbitraire schaal, start de constrainede variant vanuit een dense, metrisch correcte geometrie afkomstig van de LiDAR-scanner. De Gaussians zijn daardoor van bij de start correct gepositioneerd in de werkelijke ruimte.

Parameterkeuzes en motivatie

De trainingsparameters worden restrictiever ingesteld om de geometrische vrijheid van de Gaussians te beperken. Het doel is een model dat enerzijds visueel voldoende kwaliteit behoudt, maar anderzijds minder sterk afwijkt van de initiële LiDAR-geometrie dan de baseline. De uiteindelijke parameterkeuzes werden empirisch bepaald via een reeks experimenten. De resulterende configuratie vormt een afweging tussen enerzijds geometrische trouw aan de LiDAR-meting en anderzijds behoud van voldoende fotometrische kwaliteit. De parameters waarmee hoofdzakelijk geëxperimenteerd worden zijn positie-optimalisatie en opacity-parameters. In Sectie 4.3

wordt dieper ingegaan op de experimenten en welke invloed het aanpassen van deze parameters heeft.

Verwacht effect op het trainingsproces

Door de geometrische vrijheid te beperken zal het trainingsproces minder agressief Gaussians verplaatsen of densificeren in gebieden waar de LiDAR-geometrie reeds een goede dekking biedt. Dit heeft als verwacht gevolg dat de gerenderde beelden iets minder fotometrisch optimaal zijn dan bij de baseline, maar dat de onderliggende geometrie dichter aanleunt bij de werkelijke ruimtelijke structuur van de scène. In Sectie 4 wordt geëvalueerd of dit verwachte gedrag effectief optreedt en in welke mate de geometrische nauwkeurigheid verbetert ten opzichte van de baseline.

3.7 Lichtfeld Studio

In het kader van dit onderzoek werden verschillende uitbreidingen aangebracht aan LichtFeld Studio om forensische bruikbaarheid te ondersteunen. Een eigen ontwikkelde centrale toevoeging is de *Forensic Reliability*-module: een component die per Gaussian Splat kwantificeert in hoeverre de geometrie nog herleidbaar is tot de oorspronkelijke initialisatiepuntenwolk. De module maakt het mogelijk om visueel te inspecteren welke splats geometrisch verankerd zijn en welke mogelijk het gevolg zijn van vrije optimalisatie tijdens het trainingsproces.

3.7.1 Beschikbaarheid, initialisatie en scèneschaal

De forensische betrouwbaarheidsmodule is niet altijd beschikbaar: ze vereist dat bij het laden van een .ply-bestand de bijhorende checkpoint wordt gevonden. LichtFeld Studio zoekt hiervoor naar een bestand van de vorm `checkpoint-<iter>.resume`, zowel in dezelfde map als in een submap `checkpoints`. Uit die checkpoint worden het datasetpad en de bijhorende opties gelezen. Vervolgens wordt de dataset opnieuw geladen en wordt de initiële puntenwolk, dit is de puntenwolk waarmee de training gestart werd, eruit geëxtraheerd. Op basis van die initiële puntenwolk wordt de `ForensicReliability`-module geïnitieerd. Als een van deze stappen mislukt, bijvoorbeeld doordat geen checkpoint aanwezig is of de dataset niet meer bereikbaar is, is de forensische betrouwbaarheidsanalyse niet beschikbaar voor dat model.

Bij initialisatie worden alle posities uit de initiële puntenwolk ingelezen en opgeslagen als een lijst van driedimensionale coördinaten. Tegelijkertijd wordt een scènespecifieke referentieafstand berekend, aangeduid als σ_{pos} . Deze waarde bepaalt de gevoeligheid van de positiegebaseerde betrouwbaarheidsterm. De berekening verloopt als volgt: eerst wordt de axis-aligned bounding box bepaald over alle initiële punten, waarna de lengte van de diagonaal d wordt berekend. De referentieafstand wordt vervolgens vastgelegd als:

$$\sigma_{\text{pos}} = \text{clamp}(0,02 \cdot d, 0,01, 1,0) \quad (3.3)$$

Hierdoor is de referentieafstand grofweg 2% van de scenediagonaal, wat zorgt voor een schaalbestendige maatstaf. De waarde van 2% is gekozen als een schaalbestendige benadering die aansluit bij gangbare drempelwaarden in puntenwolkverwerking, waarbij waarden in het bereik van 1–5% van de scenediagonaal doorgaans robuust gedrag vertonen. De clamp voorkomt dat de gevoeligheid extreem hoog of laag wordt bij zeer kleine of zeer grote scènes.

3.7.2 Berekening van de betrouwbaarheidsscore

Voor elke Gaussian Splat wordt de betrouwbaarheidsscore berekend op basis van twee componenten: een positiegebaseerde term en een schaalgebaseerde term. Een belangrijk aspect van de implementatie is dat de afstand *index-gebaseerd* wordt bepaald: voor een Gaussian met

index i wordt de Euclidische afstand genomen tot het initiële punt met diezelfde index i in de oorspronkelijke puntenwolk:

$$\delta = \|\mathbf{p}_i - \mathbf{q}_i\| \quad (3.4)$$

Deze keuze is bewust: tijdens de initialisatie van 3DGS wordt elke Gaussian i rechtstreeks geplaatst op punt \mathbf{q}_i uit de puntenwolk, waardoor er een één-op-één correspondentie bestaat tussen Gaussian-index en initialisatiepunt. Een nearest-neighbor zoekopdracht zou die semantische koppeling doorbreken: omdat de initialisatiepuntenwolk zeer dense is, zou elke splat altijd een goede nearest-neighbor vinden en een Gaussian die ver van haar oorsprong is gedreven zou als betrouwbaar worden beoordeeld puur omdat een andere naburige Gaussian toevallig wel op haar plek bleef. De index-gebaseerde meting is daardoor forensisch eerlijker: zij registreert hoeveel elke individuele Gaussian ten opzichte van haar eigen beginpositie is verplaatst, zodat afwijkingen door vrije optimalisatie zichtbaar worden in de score. De methode vereist wel dat de Gaussian-index overeenkomt met de index in de initiële puntenwolk. Gaussians die buiten dit bereik vallen, bijvoorbeeld extra splats ontstaan door densificatie tijdens het trainingsproces, hebben geen corresponderend initieel punt en worden bijgevolg direct als laag betrouwbaar beschouwd. Dit wordt gedaan door de betrouwbaarheidsscore r , beschreven in Formule 3.8, direct op een waarde te zetten binnen het bereik van de laagbetrouwbaarheids categorie. De categorieën worden gedefinieerd in Tabel 3.1.

Vervolgens wordt de effectieve schaal van de Gaussian, s , geclampt op minimaal 10^{-6} . De schaal s kan tijdens de optimalisatie naar arbitrair kleine waarden convergeren. Zonder deze ondergrens zou de berekening van de genormaliseerde afstand een deling door nul veroorzaken. De drempel 10^{-6} is laag genoeg om geen materiële invloed te hebben op Gaussians met realistische schaalwaarden. Hieruit volgt de genormaliseerde afstand:

$$\tilde{\delta} = \frac{\delta}{s} \quad (3.5)$$

De absolute afstand δ en de genormaliseerde afstand $\tilde{\delta}$ meten fundamenteel verschillende soorten afwijking. De absolute afstand drukt verplaatsing uit ten opzichte van de scèneschaal, de genormaliseerde afstand ten opzichte van de eigen grootte van de Gaussian. Zonder $\tilde{\delta}$ zouden een grote Gaussian ($s = 2$ m) en een kleine Gaussian ($s = 0,01$ m) die beide 0,5 m bewegen een gelijke absolute afwijking krijgen, terwijl de proportionele drift fundamenteel verschilt: $\tilde{\delta} = 0,25$ respectievelijk $\tilde{\delta} = 50$. Deze normalisatie is essentieel omdat de schaal s aangeeft hoe groot de Gaussian is en dus hoe precies de positie wordt voorgesteld. Een kleine Gaussian staat voor een klein, gedetailleerd gebied en betekent dat de positie heel nauwkeurig is. Een grote Gaussian beschrijft een groter gebied en is minder precies. Daardoor heeft dezelfde verschuiving een groter effect bij een kleine Gaussian dan bij een grote. Door de afstand te normaliseren kijk je naar een relatieve fout in plaats van een absolute. Dit past bij het idee dat de onzekerheid in de positie meegroeit met de schaal ($\sigma \propto s$). Zonder deze aanpassing zouden kleine structuren te weinig ‘‘straf’’ krijgen en grote structuren te veel invloed hebben, wat een vertekend en misleidend resultaat geeft.

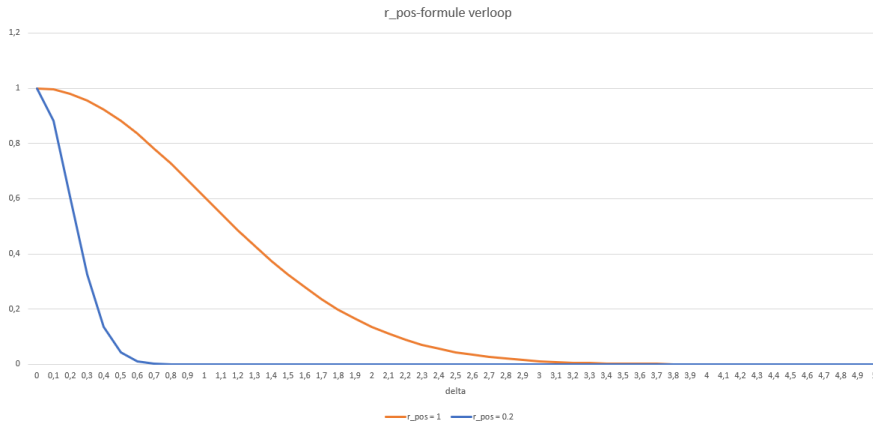
Op basis van beide afstandswaarden worden twee Gaussian-vormige betrouwbaarheidstermen berekend:

$$r_{\text{pos}} = \exp\left(-\frac{1}{2} \left(\frac{\delta}{\sigma_{\text{pos}}}\right)^2\right) \quad (3.6)$$

$$r_{\text{scale}} = \exp\left(-\frac{1}{2} \tilde{\delta}^2\right) \quad (3.7)$$

Een opvallend verschil tussen beide termen is dat r_{pos} een expliciete scènereferentie σ_{pos} vereist, terwijl r_{scale} die niet nodig heeft. Bij r_{pos} wordt een absolute verplaatsing δ gemeten: zonder σ_{pos} zou niet duidelijk zijn of die verplaatsing groot of klein is ten opzichte van de scène. Bij r_{scale} is de scènecontext echter al impliciet aanwezig via de eigen grootte s van de Gaussian. In een grote scène zijn Gaussians gemiddeld groter, waardoor $\tilde{\delta} = \delta/s$ klein blijft voor een vergelijkbare absolute verplaatsing. In een kleine scène zijn Gaussians kleiner en wordt $\tilde{\delta}$ snel groot. De Gaussian fungeert daarmee als haar eigen lokale referentie-grootte, waardoor één globale scènemaatstaf overbodig is. De standaarddeviatie in r_{scale} is impliciet gelijk aan één schaal-eenheid, wat de term per definitie scènagnostisch maakt.

De exponentiële kwadratische vorm is om drie redenen gekozen. Ten eerste levert zij altijd een uitvoer in $[0, 1]$: Figuur 3.9 toont het verloop van de positiegebaseerde betrouwbaarheidsterm r_{pos} als functie van de absolute verplaatsing δ , voor twee waarden van de referentieafstand σ_{pos} . Bij $\delta = 0$ is de betrouwbaarheidsscore maximaal ($r_{\text{pos}} = 1$) en daalt zij asymptotisch naar 0 naarmate δ toeneemt. Beide curves volgen dezelfde exponentiële kwadratische vorm, maar de waarde van σ_{pos} bepaalt hoe snel de daling verloopt. Bij $\sigma_{\text{pos}} = 1$ (oranje curve) blijft de score tot grote verplaatsingen relatief hoog, bij $\sigma_{\text{pos}} = 0,2$ (blauwe curve) daalt de score al merkbaar rond $\delta \approx 0,2$. Dit illustreert de schaalafhankelijkheid van de term: in een grotere scène, met een corresponderende grotere σ_{pos} , worden dezelfde absolute verplaatsingen als minder afwijkend beschouwd dan in een kleinere scène. Ten tweede kent de vorm een zachte afstraffing dichtbij de oorsprong en een sterke afstraffing op grotere afstand, wat het verwachte gedrag van een geoptimaliseerde Gaussian weerspiegelt: kleine afwijkingen zijn inherent aan het optimalisatieproces en mogen de score nauwelijks verlagen, terwijl substantiële verplaatsingen duidelijk zichtbaar moeten zijn. Ten derde is de Gaussian-vorm differentieerbaar en statistisch interpreteerbaar als de maximale-entropieverdeling voor een gegeven variantie, wat de parameterkeuze voor σ_{pos} te motiveren maakt als een veronderstelling over de verwachte spreiding van Gaussian-posities.



Figuur 3.9: Grafiek van het verloop van de Formule 3.6

De term r_{pos} meet de absolute geometrische afstand ten opzichte van de scèneschaal, terwijl r_{scale} de afstand uitdrukt relatief ten opzichte van de eigen grootte van de Gaussian. Voor r_{scale} is de standaarddeviatie impliciet gelijk aan één schaal-eenheid, zodat een verplaatsing gelijk aan de eigen grootte van de Gaussian de score tot $e^{-1/2} \approx 0,61$ brengt, een bewuste keuze die zulke proportionele drift in de categorie “matig” plaatst.

De eindbetrouwbaarheidsscore wordt berekend als een gewogen combinatie van beide termen, waarbij het gewicht w instelbaar is via de gebruikersinterface (`position_weight`, geclampt naar $[0.5, 1]$):

$$r = w \cdot r_{\text{pos}} + (1 - w) \cdot r_{\text{scale}} \quad (3.8)$$

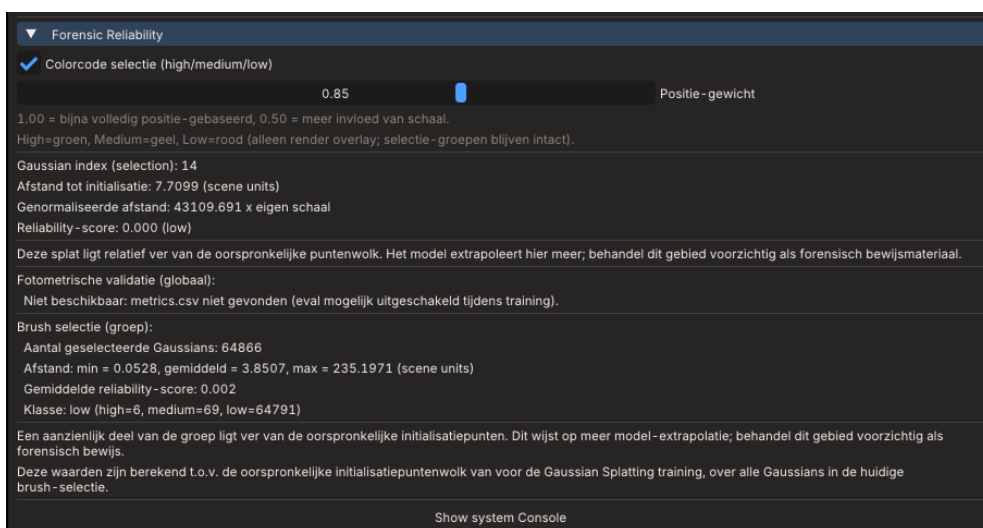
De lineaire convexe combinatie garandeert dat $r \in [0, 1]$ voor alle $w \in [0.5, 1]$, zodat de drempelwaarden in Tabel 3.1 direct toepasbaar blijven. De instelbare w erkent dat de relatieve importantie van absolute versus proportionele afwijking scèneafhankelijk is: bij een scène met vaste referentiepunten is absolute positie doorslaggevend ($w \rightarrow 1$). Bij objecten zonder vaste schaal is proportionele drift informativer ($w \rightarrow 0$). Door w als gebruikersparameter te exposeren blijft de methodologie generiek toepasbaar zonder dat de onderliggende formule verandert.

Op basis van de eindwaarde wordt elke splat ingedeeld in één van drie betrouwbaarheidscategorieën:

Categorie	Drempelwaarde	Interpretatie
Hoog	$r \geq 0,8$	De splat ligt geometrisch dicht bij de oorspronkelijke puntenwolk.
Matig	$0,5 \leq r < 0,8$	De splat ligt op matige geometrische afstand van de initialisatie.
Laag	$r < 0,5$	De splat ligt relatief ver van de oorspronkelijke puntenwolk.

Tabel 3.1: Classificatieschema voor forensische betrouwbaarheidscategorieën

Er is een belangrijk technisch detail rond de verklarende tekst per splat. Die tekst wordt intern vastgelegd op basis van de standaardweging ($0,85 \cdot r_{\text{pos}} + 0,15 \cdot r_{\text{scale}}$). Wanneer de gebruiker nadien het gewicht w aanpast via de interface, wordt de betrouwbaarheidsscore en het label herberekend, maar de verklarende tekst niet overschreven. Rond de drempelwaarden 0,5 en 0,8 kan daardoor incidenteel een kleine discrepantie ontstaan tussen het weergegeven label en de bijbehorende toelichting. We zouden dit kunnen oplossen door zowel de score, het label en de verklarende tekst opnieuw te bepalen op basis van de nieuwe weging wanneer de gebruiker w aanpast. Dit elimineert de discrepantie volledig, maar vereist dat de verklarende tekst ook gekoppeld wordt aan de gewichtsparemetrie. Een voorbeeld van zo een verklarende tekst is terug te vinden in Figuur 3.10. Deze tekst hoort bij de selectie die terug te vinden is in Figuur 3.11.



Figuur 3.10: Betrouwbaarheidsselectietool: verklarende tekst voor selectie te zien in Figuur 3.11

3.7.3 Selectie, groepsanalyse en visuele kleurcodering

In het Forensic Reliability-paneel van de interface kan een individuele Gaussian worden geïnspecteerd. De selectielogica werkt als volgt: als er een actieve selectiemask aanwezig is, wordt de eerste geselecteerde index als representatieve Gaussian gebruikt.

Voor de geselecteerde Gaussian worden de positie en de effectieve schaal opgehaald. De schaal wordt afgeleid uit de drie logaritmische schaalcomponenten (s_x, s_y, s_z) als het geometrisch gemiddelde in log-ruimte:

$$s = \exp\left(\frac{s_x + s_y + s_z}{3}\right) \quad (3.9)$$

Met deze positie en schaal wordt vervolgens de betrouwbaarheidsscore berekend.

Wanneer meerdere splats tegelijk geselecteerd zijn, worden twee analyses parallel uitgevoerd. Enerzijds wordt voor de eerste geselecteerde index de individuele detailinformatie getoond. Anderzijds wordt over alle geselecteerde indices geïtereerd om groepsstatistieken te berekenen: de minimale, gemiddelde en maximale afstand, de gemiddelde betrouwbaarheidsscore en het aantal splats per categorie.

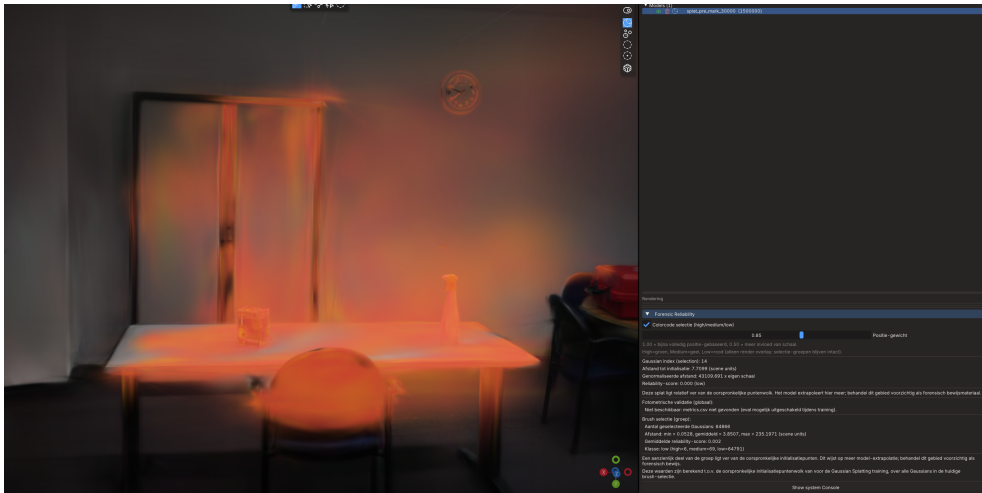
Het groepslabel wordt toegekend als *hoog* wanneer de gemiddelde betrouwbaarheid $\geq 0,8$ en de fractie laag geclassificeerde splats $< 10\%$ bedraagt. De combinatie van beide voorwaarden is bewust: het gemiddelde alleen volstaat niet, omdat een hoog gemiddelde kan samengaan met een kleine maar forensisch relevante deelverzameling sterk afwijkende splats. De 10%-grens fungeert daarom als een uitzonderingsregel die ook bij een verder goed scorende groep toch ingrijpt wanneer te veel individuele splats onbetrouwbaar zijn.

Het label *laag* wordt toegekend wanneer de gemiddelde betrouwbaarheid $< 0,5$ of de fractie laag geclassificeerde splats $> 40\%$ is. De disjunctieve formulering weerspiegelt dat een groep op twee manieren globaal onbetrouwbaar kan zijn: ofwel scoort de meerderheid zwak, ofwel is een substantieel aandeel van de splats individueel laag geclassificeerd. De drempel van 40% is gekozen als een heuristisch omslagpunt waarbij de onbetrouwbare splats dominant worden binnen de groep. In alle overige gevallen wordt het label *matig* toegekend.

De gehanteerde drempelwaarden zijn heuristisch bepaald op basis van de individuele categoriegrenzen en intern consistent, maar niet empirisch gevalideerd. Verfijning op basis van uitgebreidere experimentele data wordt aanbevolen als richting voor toekomstig onderzoek.

Als de optie `forensic_colorize_selection` is ingeschakeld, bouwt de RenderingManager een afgeleide renderoverlay op basis van de selectiemask. Per geselecteerde Gaussian wordt de betrouwbaarheidsscore berekend en vertaald naar een vaste render-groepsidentificator: 127 voor hoog, 126 voor matig en 125 voor laag. Deze identificatoren worden vervolgens gekleurd weergegeven: groen voor hoog, geel voor matig en rood voor laag. Dit resulteert erin dat binnen eenzelfde selectie tegelijk groen, geel en rood gekleurde splats zichtbaar kunnen zijn, afhankelijk van hun individuele betrouwbaarheidsscore. De onderliggende selectiedata wordt hierbij niet gewijzigd. De kleurcodering is enkel een visuele overlay tijdens de rendering. Een voorbeeld van hoe de overlay eruitziet is terug te vinden in Figuur 3.11.

De keuze voor drie discrete kleuren in plaats van een continue kleurgradiënt is bewust gemaakt vanuit forensische leesbaarheid. In een forensische context is een onmiddellijk interpreteerbare visuele classificatie belangrijker dan maximale informatiedichtheid. Een vloeiende gradiënt legt de interpretatie bij de waarnemer: waar precies de overgang ligt tussen betrouwbaar en onbetrouwbaar wordt dan een visueel oordeel in plaats van een methodologische keuze. De drie kleuren sluiten bovendien direct aan bij de categorieën in Tabel 3.1, waardoor de visuele output consistent blijft met de onderliggende classificatie. Een continue kleurgradiënt op basis van de exacte betrouwbaarheidsscore r zou een rijkere weergave bieden en is technisch haalbaar, maar wordt hier niet toegepast omdat de meerwaarde voor forensische interpretatie beperkt is zolang de categoriegrenzen 0,5 en 0,8 de operationele beslissingsbasis vormen.



Figuur 3.11: Betrouwbaarheidsselectietool: renderoverlay op basis van drie kleuren

3.7.4 Export van gehallucineerde en originele splats

Naast de forensic reliability analyse wordt aan het einde van het trainingsproces ook een binaire splitsing van het model weggeschreven. Op de finale iteratie exporteert de trainer twee aparte PLY-bestanden: `splat_hallucinated.ply` en `splat_original.ply`. Deze bestanden worden niet per iteratie herhaald, maar enkel aangemaakt wanneer `iter_num == optimization.iterations`.

De splitsing is gebaseerd op een booleaanse hallucination mask die de actieve trainingsstrategie bijhoudt. Concreet roept de trainer `strategy->get_hallucinated_mask()` aan en verdeelt het model op basis van die mask: splats waarvoor de mask `true` is worden in `splat_hallucinated.ply` opgeslagen, de overige in `splat_original.ply`. De kern van de mask is de interne `is_new`-vlag per splat, die bijgehouden wordt door zowel de MCMC- als de ADC-strategie.

Bij MCMC wordt `is_new` op `true` gezet voor splats die tijdens training nieuw zijn aangemaakt via groei (`add_new_gs`) of die een eerder afgestorven slot opnieuw invullen via relocatie (`relocate_gs`). Bij ADC gelden dezelfde gevallen, maar wordt relocatie als soft-delete afgehandeld: slots worden als vrij gemarkeerd in plaats van fysiek verwijderd, en enkel actieve (niet-vrije) slots tellen mee in de finale mask.

Vlak voor de eigenlijke export past de trainer ook een visuele markering toe via `mark_hallucinated_splats()`: gehallucineerde splats krijgen een rode tint, terwijl originele splats die tijdens training significant zijn verplaatst een blauwe tint krijgen (drempel: 10^{-3}). Deze kleuring dient uitsluitend voor visuele inspectie en heeft geen invloed op de binaire splitsing zelf. Een splat die blauw gemarkeerd is, verplaatst maar niet nieuw, blijft dus origineel in de exportbestanden. Merk ook op dat de export de markering niet bevat: vlak voor de markering wordt eerst nog een aparte snapshot weggeschreven als `splat_pre_mark-<iter>.ply`, zodat de ruwe eindtoestand zonder tinting bewaard blijft.

Als de hallucination mask ongeldig is of de shape niet overeenkomt met het model, worden de gesplitste bestanden niet aangemaakt. Als een van de twee deelsets leeg is, wordt die specifieke PLY eveneens niet weggeschreven.

3.8 Juridische validator

Om de forensische bruikbaarheid van de reconstructiepipeline te ondersteunen, werd een juridische validatielaag ontwikkeld die bestaat uit drie onderling verbonden modules, deze worden in de volgende secties verder besproken. Samen zorgen deze modules ervoor dat elk gegenereerd

validatierapport herleidbaar, integer en tijdelijk verankerd is, wat essentiële vereisten zijn voor gebruik als bewijsmateriaal in een gerechtelijke context.

3.8.1 Validatiemodule

De kernmodule `validate_alignment.py` voert een kwantitatieve beoordeling uit van de alignmentkwaliteit tussen de sparse COLMAP-puntenwolk en de dense LiDAR-referentie. Het script accepteert als invoer een originele sparse puntenwolk (als `.ply` of COLMAP `points3D.txt`), een gealignde versie van dezelfde puntenwolk, en de dense LiDAR-referentie als `.ply`. Als uitvoer genereert het een uitgebreid JSON-rapport.

De validatie verloopt in een aantal opeenvolgende stappen. Eerst worden alle puntenwolken geladen en gefilterd met dezelfde outlier-filteringstrategie als in de alignmentpipeline zelf: een combinatie van radiale IQR-filtering (drempel $3,0 \times \text{IQR}$) en as-georiënteerde filtering (drempel $1,5 \times \text{IQR}$) per dimensie. Omdat de validatietool als zelfstandige module is geïmplementeerd en geen toegang heeft tot de reeds ingeladen puntenwolken in geheugen, worden de bronbestanden opnieuw ingelezen vanuit de originele `.e57`-bestanden. Dit garandeert dat de validatie exact dezelfde punten beoordeelt als de pipeline gebruikt heeft.

Vervolgens worden per-puntafstanden berekend via een k-d-boom: voor elk punt in de sparse of gealignde cloud wordt de kortste afstand naar het dichtstbijzijnde punt in de dense referentie bepaald. Op basis van deze afstandsverdelingen worden statistieken berekend zowel voor als na de alignment, waaronder het gemiddelde, de mediaan, de RMSE, het 95e percentiel en het overlappercentage. Dit laatste geeft aan welk deel van de punten binnen een instelbare drempel (standaard 10 cm) van de referentie ligt.

Optioneel kan een inlier-ratio worden opgegeven die overeenkomt met de ICP-fitness uit de alignmentstap. In dat geval worden de statistieken ook berekend over uitsluitend de beste $X\%$ punten, wat een realistischer beeld geeft van de nauwkeurigheid in de overlappende gebieden.

Op basis van de RMSE en het overlappercentage kent de module een gelaagd kwaliteitsoordeel toe:

Oordeel	RMSE-criterium	Overlap-criterium
Uitstekend	< 2 cm	> 80%
Goed	2–5 cm	60–80%
Matig	5–10 cm	40–60%
Slecht	> 10 cm	< 40%

Tabel 3.2: Kwaliteitsoordelen voor forensische alignmentvalidatie

De grens van twee cm *uitstekend* sluit aan bij de typische meetnauwkeurigheid van professionele terrestrische laserscans, waarbij een RMSE onder twee cm betekent dat de resterende afwijking vergelijkbaar is met de intrinsieke meetonzekerheid van het instrument zelf. Een alignment op dit niveau introduceert bijgevolg geen materiële extra fout bovenop de sensoronzekerheid. De grens van vijf cm voor *goed* vertegenwoordigt een drempel waarbij de geometrische afwijking nog steeds klein is ten opzichte van de relevante scèneafmetingen bij een typische forensische scène, maar waarbij de alignment niet langer als instrumenteel nauwkeurig beschouwd kan worden. Boven tien cm zijn afwijkingen groot genoeg om de forensische interpretatie van de reconstructie te beïnvloeden, bijvoorbeeld bij het bepalen van relatieve posities van sporenmateriaal.

De overlapperpercentages volgen een analoge redenering. Een overlap van meer dan 80% garandeert dat het overgrote deel van de scène door beide scans wordt gedekt, waardoor de alignment op een representatieve puntenverzameling is gebaseerd. Onder 40% is de gedeelde puntenverzameling te beperkt om een betrouwbare alignment te ondersteunen, ongeacht de RMSE-waarde.

Het eindoordeel is het minimum van het RMSE-oordeel en het overlappercentage-oordeel. Het rapport bevat ook een interpretatieparagraaf in leesbare tekst, met concrete aanbevelingen en een expliciete vermelding van de beperkingen van de validatie, zoals het feit dat punten buiten het bereik van de E57-scan niet geverifieerd kunnen worden.

Het rapport krijgt een ingebed SHA-256-hash over alle velden behalve het hashveld zelf, zodat elke latere wijziging detecteerbaar is.

3.8.2 Chain of custody

De module `chain_of_custody.py` implementeert een digitale chain of custody. Elk validatierapport bevat een `chain_of_custody`-sectie die de volgende informatie vastlegt:

- De identiteit en functie van de operator die het rapport heeft gegenereerd, inclusief naam, functietitel en organisatie.
- Een zaakverwijzing waarmee het rapport gekoppeld kan worden aan een specifiek dossier.
- De versie van het gebruikte validatiesysteem.
- De uitvoeringstijdstempel in UTC en lokale tijd.
- SHA-256-hashes en bestandsgroottes van alle invoerbestanden (sparse puntenwolk, gealignde puntenwolk en dense referentie). Hierdoor is achteraf controleerbaar of precies dezelfde bestanden zijn gebruikt als bij de validatie.
- Een overzicht van de runtime-omgeving, waaronder de versies van Python en de gebruikte bibliotheken (`numpy`, `scipy`, `open3d`).
- Een formele verklaring in de eerste persoon, gegenereerd op basis van de operatorgegevens, die stelt dat het rapport een getrouwe weergave is van de uitgevoerde technische validatie.

Na het aanmaken van dit object wordt via `attach_report_hash()` een SHA-256-hash berekend over het volledige rapport, geserialiseerd als JSON met gesorteerde sleutels. Deze hash wordt als apart veld aan het rapport toegevoegd. Elke partij die het rapport later ontvangt, kan deze hash herberekenen en vergelijken om te bevestigen dat het rapport niet is gewijzigd na generatie.

3.8.3 Vertrouwde tijdstempel

Na het wegschrijven van het JSON-rapport wordt via `timestamp.py` een RFC 3161-tijdstempel aangevraagd bij een externe Timestamp Authority (TSA). RFC 3161 is een IETF-standaard voor vertrouwde tijdstempels, waarbij een onafhankelijke derde partij cryptografisch bevestigt dat een bepaald document op een specifiek tijdstip heeft bestaan [CPDR01]. Dit is relevant in forensische en juridische contexten omdat het manipulatie na het feit moeilijker maakt: de hash van het rapport is op een aantoonbaar tijdstip vastgelegd door een onafhankelijke instantie.

De module bouwt handmatig een DER-gecodeerd `TimeStampRequest` (TSQ) op overeenkomstig met RFC 3161, met daarin de SHA-256-hash van het rapport en een willekeurige nonce om replay-aanvallen te voorkomen. Dit verzoek wordt verzonden naar een lijst van publiek beschikbare TSA-servers, waaronder FreeTSA, DigiCert, Sectigo en GlobalSign. Bij succes slaat de module het ontvangen `TimeStampReply` (TSR) op als een binair `.tsr`-bestand naast het rapport. Bij falen van alle servers wordt een gedetailleerde foutstatus in het rapport opgenomen, inclusief instructies voor handmatige tijdstempeling via `openssl` of de FreeTSA-webinterface.

3.8.4 Integratie in de validatie pipeline

De drie modules zijn geïntegreerd in de hoofdvalidatiefunctie `validate_alignment()`. Het eindrapport is daarmee zowel inhoudelijk als procedureel controleerbaar: de meetwaarden kunnen worden geverifieerd door dezelfde berekeningen opnieuw uit te voeren op de ghashe

invoerbestanden, de operatoridentiteit en het tijdstip zijn vastgelegd in de chain of custody, en de RFC 3161-tijdstempel bevestigt het moment van aanmaak onafhankelijk van de operator.

3.9 Validatierapport-viewer

Om het JSON-validatierapport te vertalen naar een voor de rechtbank bruikbaar document, werd een grafische desktop-applicatie ontwikkeld op basis van Electron. De applicatie combineert een web-based gebruikersinterface (`index.html`, `styles.css`, `script.js`) met een Node.js-backend (`main.js`) en een beveiligde communicatielaag (`preload.js`). Het eindresultaat is een enkelvoudig PDF-bestand dat voldoet aan de structurele eisen van een technisch deskundigenverslag in de Belgische gerechtelijke procedure.

3.9.1 Architectuur: Electron als platform

Electron maakt het mogelijk om een webapplicatie als native desktopapplicatie te verpakken, waarbij de weergave-laag in een Chromium-venster draait en de systeemoperaties in een afzonderlijk Node.js-proces worden uitgevoerd. Dit is relevant voor de rechtbank-viewer om twee redenen. Ten eerste vereist de verificatie van de RFC 3161-tijdstempel de uitvoering van een extern commando (`openssl ts -verify`), wat vanuit een gewone browseromgeving niet mogelijk is. Ten tweede moeten tijdelijke bestanden op het bestandssysteem worden aangemaakt en verwijderd tijdens de verificatie, wat eveneens vereist dat de applicatie native systeemtoegang heeft.

De communicatie tussen de weergave-laag en de Node.js-backend verloopt via een gecontroleerd Inter-Process Communication (IPC) kanaal. Het `preload.js`-bestand definieert welke functies vanuit de browserlaag naar de Node.js-laag mogen worden aangeroepen. In de huidige implementatie is dit beperkt tot één functie: `electronAPI.verifyTSR()`, die de tijdstempelverificatie aanstuurt. Hierdoor heeft de weergave-laag geen directe toegang tot het bestandssysteem of de procesuitvoering, wat een beveiligingsrisico zou vormen.

3.9.2 Stap 1: Uploaden van het JSON-rapport

De applicatie opent met een uploadscherm waarop het JSON-validatierapport kan worden geladen, ofwel via een klikbare uploadzone ofwel via drag-and-drop. Het bestand wordt uitgelezen als `ArrayBuffer` om de originele bytereeks te bewaren. Dit is essentieel: voor de latere tijdstempelverificatie moet het rapport byte-voor-byte identiek zijn aan het bestand waarvoor de TSA-server oorspronkelijk een tijdstempel heeft afgegeven. De `ArrayBuffer` wordt bewaard in een globale variabele en wordt niet via JSON opnieuw geserialiseerd, zodat geen enkel verschil in witruimte of tekencodering de verificatie kan verstoren. In Figuur 3.12 is het uploadscherm te zien.



Figuur 3.12: Uploaden van het JSON-rapport

Na het inlezen wordt de ArrayBuffer gedecodeerd als UTF-8 tekst en geparsed als JSON. De velden uit de `chain_of_custody`-sectie worden automatisch ingevuld in het operatorformulier: naam, functie, organisatie en zaakverwijzing worden rechtstreeks overgenomen uit het rapport, zodat de identiteitsgegevens consistent blijven met de vastgelegde chain of custody.

3.9.3 Stap 2: Invullen van operatorgegevens

Na het uploaden verschijnt het rapport scherm met een formulier voor operatorgegevens. De velden naam, functie, organisatie en dossiernummer zijn verplicht en worden gevalideerd voor de PDF-generatie. Optioneel kunnen ook de bevoegde rechtbank en bijkomende opmerkingen worden ingevuld. Hoewel de kerngegevens worden vooringevuld vanuit het JSON-rapport, kunnen ze hier worden gecorrigeerd of aangevuld, bijvoorbeeld met de naam van de specifieke rechtbank waarvoor het verslag is bestemd. Dit scherm is terug te vinden in Figuur 3.13.

Tegelijkertijd worden alle secties van het rapport visueel weergegeven: het kwaliteitsoordeel met kleurgecodeerde badges (groen voor goed of uitstekend, geel voor matig, rood voor slecht), de statistieken voor en na de alignment in tabelvorm, de vergelijking van de verbetering, de interpretatie en aanbevelingen, en de SHA-256-hashes van de invoerbestanden en het rapport zelf.

The screenshot shows the 'Forensisch Alignment Validatierapport' interface. At the top, there is a header with the logo and text: 'Forensisch Alignment Validatierapport' and 'Rechtbank-klaar documentatiesysteem — Belgische gerechtelijke procedure'. Below the header, there are four buttons: 'PDF Genereren voor Rechtbank', '.TSR UPLOADEN', '✓ Controleer Timestamp', and '↶ Ander rapport laden'. A green status bar indicates 'GESLAAGD — Alignment voldoet aan forensische eisen'. The main form is titled 'Operatorgegevens & Dossierinformatie' and contains several input fields: 'NAAM OPERATOR *' (filled with 'Jamic Withofs'), 'FUNCTIE/TITEL *' (filled with 'Student'), 'ORGANISATIE *' (filled with 'UHasselt'), 'DOSSIERNUMMER *' (filled with 'TEST-BIB123456789'), 'DATUM RAPPORT' (filled with '20/05/2026'), and 'RECHTBANK / INSTANTIE'. There is also a text area for 'OPMERKINGEN / CONTEXT'.

Figuur 3.13: Invullen van operatorgegevens

3.9.4 Stap 3: Optionele tijdstempelverificatie

Voordat de PDF wordt gegenereerd, kan optioneel het bijhorende `.tsr`-bestand worden geüpload voor verificatie. Na het uploaden wordt de knop "Controleer Timestamp" actief. Bij het klikken worden zowel de ArrayBuffer van het rapport als de bytes van het TSR-bestand als Base64-gecodeerde strings via het IPC-kanaal doorgegeven aan de Node.js-backend.

De backend schrijft beide bestanden tijdelijk weg naar de systeemtempmap, voert het commando `openssl ts -verify -data <rapport> -in <tsr> -CAfile cacert.pem` uit, leest de uitvoer, verwijdert de tijdelijke bestanden en retourneert het resultaat naar de weergave-laag. De verificatie controleert of de SHA-256-hash die is ingebed in het TSR-bestand overeenkomt met de hash van het rapport zoals het nu op schijf staat. Een geslaagde verificatie bevestigt dus zowel de integriteit van het rapport als het tijdstip waarop het bestond.

Het verificatieresultaat, inclusief de gebruikte OpenSSL-uitvoer en de exacte SHA-256-hash van het geverifieerde bestand, wordt weergegeven in een aparte sectie en wordt ook opgenomen in de gegenereerde PDF.

3.9.5 Stap 4: PDF-generatie

De PDF-generatie wordt volledig in de browser uitgevoerd via de bibliotheek jsPDF [Par23]. Na validatie van het formulier wordt een PDF-document opgebouwd dat de structuur van een formeel technisch deskundigenverslag volgt. Het document bestaat uit de volgende onderdelen:

1. Een voorblad met identificatiegegevens (dossiernummer, datum, operator, rechtbank) en een kleurgecodeerd statusblok dat het algehele kwaliteitsoordeel weergeeft. Een voorbeeld hiervan is te zien in Figuur 3.14.
2. Een inhoudsopgave met puntleiders voor alle negen secties.
3. Een samenvatting met het gedetailleerde kwaliteitsoordeel per criterium in tabelvorm.
4. De statistieken na alignment, inclusief optionele inlier-analyse.
5. De statistieken voor alignment als referentie.
6. De vergelijkingstabel voor/na met een indicatie of elke metriek is verbeterd.
7. De technische interpretatie, aanbevelingen en beperkingen uit het JSON-rapport.
8. De bestandsidentificatie met SHA-256-hashes van alle invoerbestanden.
9. De rapport-integriteitshash met toelichting over verificatiemethode.
10. De RFC 3161-tijdstempelverificatie, indien uitgevoerd.
11. Een formele verklaring en handtekeningpagina met twee ondertekeningsvelden voor de deskundige en de organisatie.

Elke pagina bevat een lopende koptekst met dossiernummer en een voettekst met paginanummer en de aanduiding "Vertrouwelijk — Rechtbank gebruik". Alle tekst wordt gesanitiseerd naar Latin-1 om compatibiliteitsproblemen met jsPDF te vermijden: geaccentueerde letters worden vervangen door hun basisvorm en niet-afdrukbare tekens worden verwijderd. Om een concreet idee te geven van hoe zo een PDF-pagina er ongeveer uitziet is in Figuur 3.15 een voorbeeld te terug te vinden.

Het gegenereerde rapport heeft uitsluitend betrekking op de alignmentkwaliteit tussen de sparse COLMAP-puntenwolk en de dense LiDAR-referentie. Informatie over de daaropvolgende stappen in de pipeline, zoals de detectie van gehallusineerde splats of de individuele betrouwbaarheidsscores uit de Forensic Reliability Module, is niet opgenomen in dit rapport. Deze aspecten worden afzonderlijk beoordeeld via de interactieve visualisatieomgeving en vallen buiten het bereik van de juridische validator.

FORENSISCH TECHNISCH DESKUNDIGENVERSLAG

Puntenwolk Alignment — Kwaliteitsvalidatie

IDENTIFICATIEGEGEVENS

Dossiernummer:	123456789
Datum rapport:	16/06/2026
Opgesteld door:	Jamie Withofs
Functie / Hoedanigheid:	Student
Organisatie:	Uhasselt

RESULTAAT: GESLAAGD

De alignment voldoet aan de forensische kwaliteitseisen.

Dit verslag is opgesteld conform de eisen voor technisch deskundigenadvies in Belgische gerechtelijke procedures. De hierin vervatte technische analyses en bevindingen zijn gebaseerd op objectieve meetgegevens en zijn uitsluitend bestemd voor gebruik door de bevoegde gerechtelijke instanties.

Gegenereerd op: 16/6/2026, 09:50:21

Forensisch Alignment Validator v1.0

Vertrouwelijk — Rechtbank gebruik

Belgische gerechtelijke procedure

Pagina 1

Figuur 3.14: Voorbeeld van de hoofding uit het Forensisch technisch deskundigenverslag

1. SAMENVATTING & KWALITEITSOORDEEL

RESULTAAT: GESLAAGD

De alignment voldoet aan de forensische kwaliteitseisen.

GEDETAILLEERD KWALITEITSOORDEEL

CRITERIUM	BEOORDELING	KWALIFICATIE
Algemeel oordeel	GOED	Voldoet
RMSE-beoordeling	GOED	Voldoet
Overlapbeoordeling	UITSTEKEND	Voldoet
Forensische bruikbaarheid	Geschikt voor forensisch gebruik	

TOELICHTING BIJ HET OORDEEL

- RMSE berekend over beste 89.5% van punten (inlier filtering)
- RMSE 4.2 cm - acceptabel voor forensisch gebruik

2. STATISTIEKEN NA ALIGNMENT

VOLLEDIGE PUNTENWOLK

Metriek	Waarde
Gemiddelde afstand	0.0532 m
RMSE	0.1016 m (10.16 cm)
Mediaan afstand	0.0279 m
P95-afstand	0.1639 m
Maximale afstand	1.0765 m
Overlap	87.09%
Aantal punten	70.501

INLIERS (BESTE 89,5% VAN DE PUNTEN)

Metriek	Waarde (inliers)
Gemiddelde afstand	0.0331 m
RMSE	0.0416 m (4.16 cm)
Mediaan afstand	0.0246 m
P95-afstand	0.0898 m

Figuur 3.15: Voorbeeld van een pagina uit het Forensisch technisch deskundigenverslag

Hoofdstuk 4

Evaluatie

Dit hoofdstuk evalueert de voorgestelde constrained Gaussian Splatting-methode aan de hand van een reeks experimenten. Het doel is om na te gaan in welke mate de verschillende trainingsparameters, voornamelijk positie-optimalisatie, densification en opacity, een invloed hebben op zowel de visuele kwaliteit als de geometrische nauwkeurigheid van het gereconstrueerde model.

Als referentiepunt dienen twee grensgevallen: een volledig onbeperkte baseline die de maximale visuele kwaliteit nastreeft en een volledig geconstrained variant waarbij positie-optimalisatie en densification zijn uitgeschakeld. Tussen deze twee uitersten worden aanvullende trainingen uitgevoerd waarbij de parameters systematisch worden gevarieerd. Op die manier wordt onderzocht welke combinatie de beste balans biedt tussen visuele getrouwheid en forensische betrouwbaarheid.

4.1 Experimentele opzet

4.1.1 Dataset

De experimenten worden uitgevoerd op een dataset die opgenomen werd in de bibliotheek van het Digital Future Lab van Universiteit Hasselt. Deze ruimte werd gekozen als testscène omdat ze gedurende een lange periode onveranderlijk bleef en verder ook representatief is voor de soort binnenomgevingen die in forensisch onderzoek voorkomen: een middelgrote, gestructureerde ruimte met een verscheidenheid aan oppervlaktetypes, meubels en wisselende lichtomstandigheden.

De geometrische basis van de dataset is een LiDAR-puntenwolk, verkregen met de Leica RTC360 zoals beschreven in Sectie 3.2. Na de visibility-gewogen downsampling en outlier removal beschreven in Sectie 3.4.3 bedraagt de uiteindelijke puntenwolk 1 148 098 punten. Dit vormt tegelijk de initialisatie voor alle constrained experimenten: elke Gaussian start op een positie die rechtstreeks herleidbaar is tot een LiDAR-meting. Om die correspondentie tussen Gaussian-index en initialisatiepunt te bewaren, zoals vereist door de forensische betrouwbaarheidsmodule (Sectie 3.7.2), wordt de puntenwolk niet willekeurig geherscht voor of tijdens het trainingsproces.

Het visuele materiaal bestaat uit 1001 cameraframes, geëxtraheerd uit de videoopname en verwerkt via COLMAP om de bijhorende camera poses en intrinsics te schatten. Het hoge aantal frames zorgt ervoor dat de meeste oppervlakken van de scène vanuit meerdere invalshoeken worden gedekt, wat de fotometrische optimalisatie stabiliseert. Tegelijkertijd impliceert dit dat het trainingsproces per iteratie kiest uit een grote cameraverzameling, wat de kans verkleint dat bepaalde delen van de scène structureel onderbelicht worden tijdens training.

4.1.2 Trainingsopzet

Alle experimenten worden uitgevoerd met LichtFeld Studio [Stu25], de Gaussian Splatting-implementatie die ook in de rest van de pipeline wordt gebruikt. Zo worden verschillen in resultaat uitsluitend verklaard door de gekozen parameterinstellingen en niet door implementatieverschillen tussen tools.

Optimalisatiestrategie: MCMC

Als optimalisatiestrategie wordt MCMC (Markov Chain Monte Carlo densification) gebruikt [Sha25]. In tegenstelling tot de klassieke ADC-strategie (Adaptive Density Control), waarbij Gaussians worden gesplitst of gekloond op basis van gradiëntdrempels, behandelt MCMC densification als een stochastisch herplaatsingsproces: Gaussians met een lage opacity worden geherpositioneerd naar gebieden met een grotere fotometrische fout, in plaats van nieuwe splats bij te maken op vaste iteratiemomenten.

Deze keuze is relevant in de context van dit onderzoek om twee redenen. Ten eerste werkt MCMC met een vast maximum aantal Gaussians gedurende de volledige training, waardoor het model niet onbeperkt kan groeien. Dit maakt het geheugenverbruik voorspelbaar en de vergelijking tussen trainingen eerlijker, omdat alle modellen met dezelfde capaciteitslimiet opereren. Ten tweede is MCMC minder afhankelijk van abrupte densification-gebeurtenissen op vaste iteratiemomenten, wat de training stabielier maakt bij een dense initialisatiepuntenwolk zoals de LiDAR-input: bij ADC kunnen zulke gebeurtenissen tot numerieke instabiliteit leiden wanneer het aantal initiële punten al hoog is. Het grootste visuele verschil bij een vergelijkings experiment is dat bij ADC meer floaters aanwezig zijn dan bij MCMC. Figuur 4.1 geeft een visuele vergelijking weer.



(a) Reconstructie met ADC-optimalisatiestrategie



(b) Reconstructie met MCMC-optimalisatiestrategie

Figuur 4.1: Vrije reconstructie vergelijking tussen MCMC- en ADC-optimalisatiestrategie.

Gedeelde trainingsparameters

Om de vergelijkbaarheid tussen de experimenten te garanderen, worden de parameters beschreven in Tabel 4.1 in alle trainingen constant gehouden.

De keuze voor een maximum van 1 150 000 Gaussians is bewust. Omdat de constrained experimenten starten met 1 148 098 LiDAR-punten als initialisatie, biedt een limiet net daarboven MCMC de ruimte om een beperkt aantal nieuwe splats aan te maken wanneer de fotometrische fout dit vereist, zonder dat het model substantieel verder groeit dan de geometrische basis.

Parameter	Waarde	Motivatie
Aantal iteraties	30 000	Standaardwaarde in de literatuur, biedt voldoende convergentietijd zonder excessieve trainingstijd.
Optimalisatiestrategie	MCMC	Zie Sectie 4.1.2.
Maximum aantal Gaussians	1 150 000	Iets boven het initiële puntenaantal van 1 148 098, zodat MCMC beperkte groei toelaat maar het model niet ongelimiteerd kan uitdijen.
Spherical harmonics degree	3	Degree 3 biedt 16 coëfficiënten per kleurkanaal en is daarmee in staat om view-afhankelijke effecten zoals spiegelreflecties en kleurgradiënten adequaat te modelleren, zonder de geheugen- en rekenkost van degree 4.
Learning rate SH-coëfficiënten	0,0025	Lage learning rate voor de kleurcoëfficiënten voorkomt dat het model te snel convergeert naar view-afhankelijke kleuren die de geometrie maskeren. Een trage kleurupdate dwingt het model om eerst een goede geometrische basis te vinden.
Learning rate schaal	0,0050	Bepaalt hoe snel de grootte van elke Gaussian zich aanpast. De waarde ligt dubbel zo hoog als die van rotatie en SH, wat reflecteert dat schaalcorrecties grotere stappen mogen zetten om lege gebieden efficiënt op te vullen, zonder de oriëntatie van splats te destabiliseren.
Learning rate rotatie	0,0010	Laagste learning rate van de drie, omdat abrupte rotatiewijzigingen de coherentie tussen aangrenzende splats snel kunnen verstoren. Een conservatieve update stabiliseert de oppervlaktoriëntatie gedurende de volledige training.

Tabel 4.1: Gedeelde trainingsparameters voor alle experimenten

4.1.3 Evaluatiemetrieken

Om de visuele kwaliteit van de getrainde modellen te kwantificeren, worden drie complementaire metrieken berekend: PSNR (Peak Signal-to-Noise Ratio) ([Fer16]), SSIM (Structural Similarity Index) ([Jim20]) en LPIPS (Learned Perceptual Image Patch Similarity) ([Ric18]).

De evaluatie verloopt in twee stappen. Eerst worden de getrainde Gaussians gerenderd naar af-

beeldingen vanuit de originele camerastandpunten via het `render.py`-script van de 3D Gaussian Splatting-implementatie [KKLD23]. Dit genereert voor elke camerapose zowel een gesynthetiseerde render als de bijhorende ground truth afbeelding. Vervolgens worden de metrieken berekend via `metrics.py`, dat gebruikmaakt van de PyTorch Image Quality (PIQ) bibliotheek.

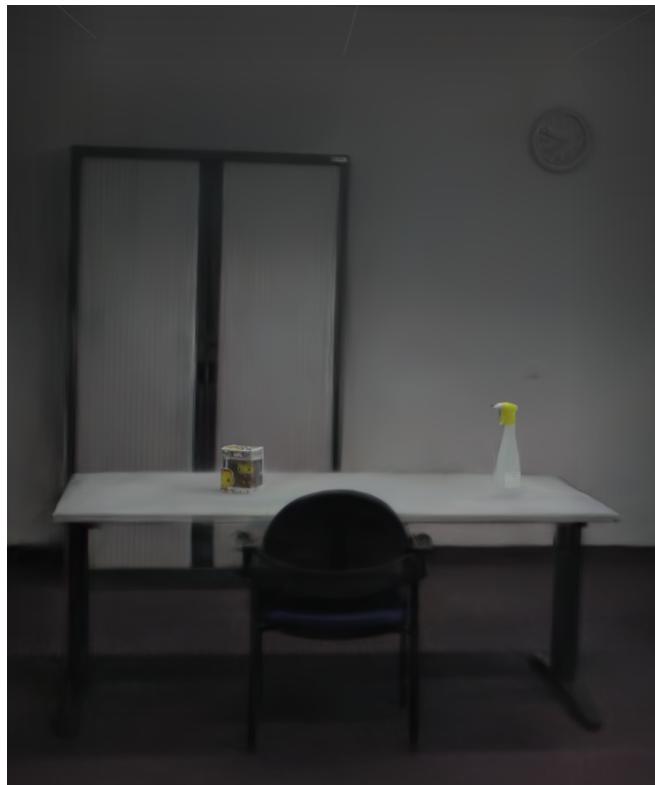
Omdat de dataset geen vooraf gedefinieerde train/test-splitsing bevat, worden de metrieken berekend op de volledige trainingsset van 1001 cameraframes. Dit impliceert dat de gerapporteerde waarden een bovengrens vormen op de werkelijke generalisatieprestatie: het model heeft alle geëvalueerde viewpoints tijdens training gezien. Voor de vergelijking tussen experimenten onderling blijft deze aanpak consistent en dus valide.

4.2 Grensgevallen

Voordat de individuele parameterexperimenten worden besproken, worden eerst de twee grensgevallen geanalyseerd die als ankerpunten dienen voor de vergelijking: een volledig onbeperkte baseline en een volledig geconstrained variant. Deze twee extremen bakenen het spectrum af waarbinnen alle overige experimenten zich bevinden. Beide gevallen worden geïnitieerd op de LiDAR-puntenwolk.

4.2.1 Baseline: geen beperkingen

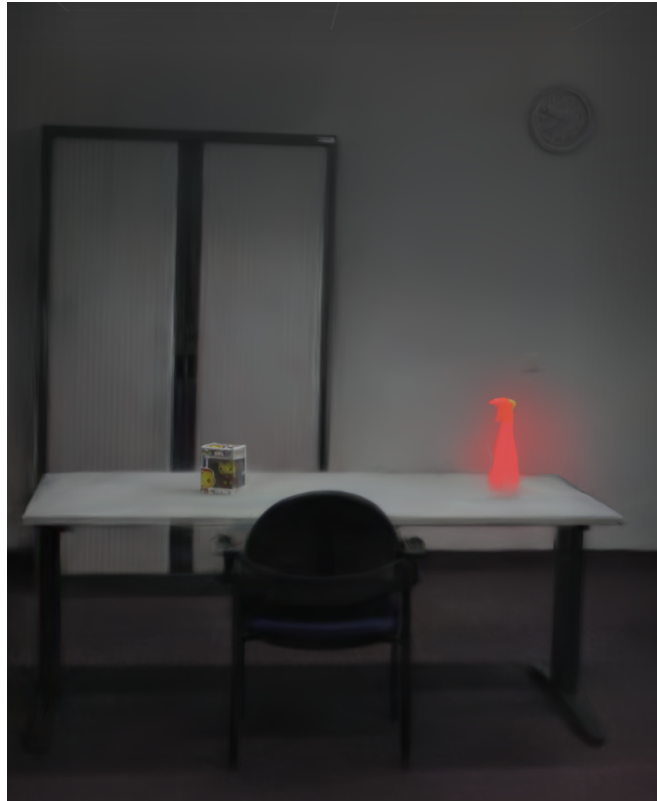
De baseline wordt getraind zonder enige beperking op positie-optimalisatie of densification. De Gaussians zijn volledig vrij om te bewegen ten opzichte van hun initiële LiDAR-positie en het model kan nieuwe splats aanmaken waar de fotometrische fout dat vereist. Dit resulteert in het visueel meest kwalitatieve model van alle experimenten. De kwantitatieve evaluatie bevestigt dit: de baseline behaalt een PSNR van 33.47 dB, een SSIM van 0,949 en een LPIPS van 0,252.



Figuur 4.2: Gerenderd beeld van de baseline (vrije optimalisatie) vanuit een representatief standpunt in de bibliotheek van het Digital Future Lab.

Figuur 4.2 toont het gerenderde beeld van de baseline vanuit een representatief standpunt. De scène is herkenbaar en de objecten op tafel, de Funko Pop-figuur en de spuitfles, zijn duidelijk zichtbaar. De globale structuur van de ruimte, inclusief het kastframe op de achtergrond en de klok op de muur, wordt correct weergegeven. Wel is enige vervaging zichtbaar aan de randen van het kastframe, wat wijst op Gaussians die zich over meerdere dieptelagen hebben uitgespreid om de overgang tussen object en achtergrond te overbruggen.

De forensische betrouwbaarheidsanalyse onthult echter een fundamenteel probleem.



Figuur 4.3: Forensische betrouwbaarheidsoverlay van de baseline. Rood duidt op Gaussians die ver van hun initiële LiDAR-positie zijn afgedreven (onbetrouwbaar), groen duidt op geometrisch verankerde splats (betrouwbaar).

Figuur 4.3 toont dat de spuitfles, die in de referentieopname wit is met een geel dopje, door de forensic overlay volledig in rood wordt weergegeven: de Gaussians die dit object representeren zijn ver van hun initiële LiDAR-positie afgedreven en worden als onbetrouwbaar beoordeeld. Dit is een direct gevolg van de vrije positie-optimalisatie: omdat de LiDAR-puntenwolk de fijne geometrie van dunne objecten zoals de spuitfles slechts beperkt kan vastleggen, hebben de bijhorende Gaussians tijdens het trainingsproces significante verplaatsingen ondergaan om de fotometrische fout te minimaliseren. Het model reproduceert het object visueel correct, maar de onderliggende geometrie is forensisch niet herleidbaar tot de originele meting. Dit illustreert het kernprobleem dat deze thesis probeert aan te pakken: een hoge visuele kwaliteit biedt geen garantie voor geometrische betrouwbaarheid.

Vergelijking met de sparse SfM-reconstructie

Ter referentie toont Figuur 4.4 een vergelijking van de reconstructie op basis van de sparse COLMAP-puntenwolk en die van de LiDAR-puntenwolk. Kwantitatief presteert de SfM-reconstructie met een PSNR van 34.11 dB, SSIM van 0,953 en LPIPS van 0,249 marginaal beter

dan de LiDAR-gebaseerde baseline.



(a) Reconstructie op basis van de LiDAR-puntenwolk. (b) Reconstructie op basis van de SfM-puntenwolk.

Figuur 4.4: Vrije reconstructie vergelijking tussen LiDAR- en COLMAP-puntenwolk.

Hoewel de scène herkenbaar is, verschilt deze reconstructie op twee fundamentele punten van de LiDAR-gebaseerde baseline. Ten eerste heeft de SfM-reconstructie een arbitraire schaal: de puntenwolk is niet metrisch verankerd en de absolute afmetingen van objecten zijn niet herleidbaar tot werkelijke afstanden. Ten tweede wijkt de oriëntatie van het model af van de LiDAR-gebaseerde reconstructie, omdat SfM de coördinaatassen vrij kiest op basis van de camerabeweging.



Figuur 4.5: Overlay van de sparse SfM-reconstructie (voorgond: tafel met stoelen) en de LiDAR-gebaseerde baseline (achtergrond). Het schaal- en positieverschil tussen beide reconstructies is duidelijk zichtbaar: de modellen overlappen niet correct, wat de noodzaak van een metrische LiDAR-initialisatie onderstreept.

Figuur 4.5 toont beide reconstructies over elkaar geprojecteerd. De tafel met stoelen op de voorgond is afkomstig van de SfM-reconstructie, de achtergrond behoort toe aan de LiDAR-

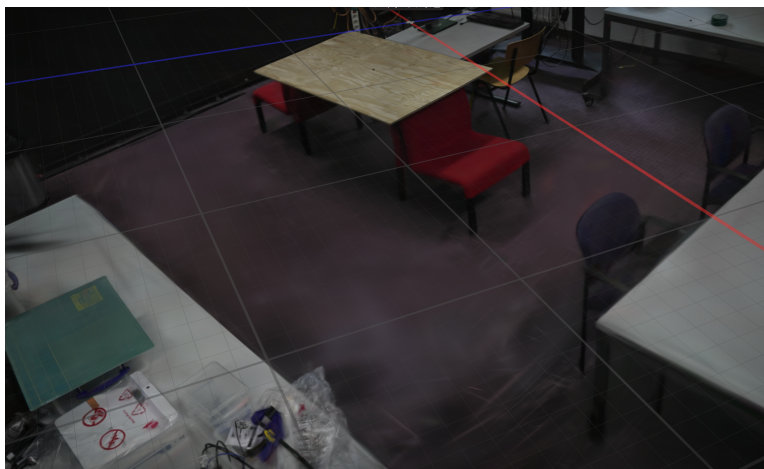
gebaseerde baseline. Het is duidelijk dat beide modellen niet correct op elkaar aansluiten: er is een zichtbare schaalafwijking en de ruimtelijke positionering verschilt. Dit toont aan dat een Gaussian Splatting-model getraind op een sparse SfM-puntenwolk, ook al levert het een visueel aanvaardbaar resultaat, niet zomaar forensisch bruikbaar is: zonder metrische verankering zijn afstanden en afmetingen in het model niet interpreteerbaar als absolute ruimtelijke maten. De LiDAR-initialisatie, zoals toegepast in alle experimenten in dit hoofdstuk, is dan ook een noodzakelijke voorwaarde voor forensische toepasbaarheid.

Tweede scène: validatie van de e57-pipeline

Om de generaliseerbaarheid van de ontwikkelde e57-pipeline te valideren, werd een tweede scène opgenomen en getraind volgens dezelfde werkwijze. Ook hier werd de LiDAR-puntenwolk ingeladen via het e57-formaat, omgezet naar de vereiste COLMAP-structuur, en gebruikt als initialisatie voor het Gaussian Splatting-trainingsproces. Het doel van deze tweede scène is niet enkel het bekomen van een visueel resultaat, maar het bevestigen dat de pipeline herhaalbaar en robuust is over verschillende opnames heen. De kwantitatieve resultaten van de tweede scène, PSNR 32.00 dB, SSIM 0,929 en LPIPS 0,258, liggen iets lager dan die van de hoofdcène, wat verklaarbaar is door verschillen in scènecomplexiteit en opnameomstandigheden. De resultaten blijven echter ruim binnen het bereik van kwalitatief goede reconstructies, wat de robuustheid van de pipeline over verschillende opnames bevestigt.



Figuur 4.6: Reconstructie met de e57-initialisatie

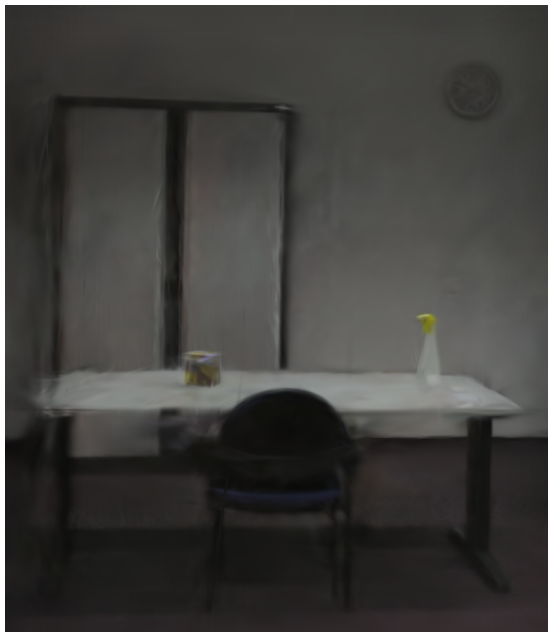


Figuur 4.7: Reconstructie met de COLMAP-initialisatie

Als we bij deze scène de verschillende initialisaties vergelijken, zien we bij de reconstructie met de e57-initialisatie een significante verbetering in de grond ten opzichte van de reconstructie met de COLMAP-initialisatie. Dit is weergegeven in Figuur 4.6 en Figuur 4.7.

4.2.2 Volledig geconstrained: geen positie-optimalisatie en geen densification

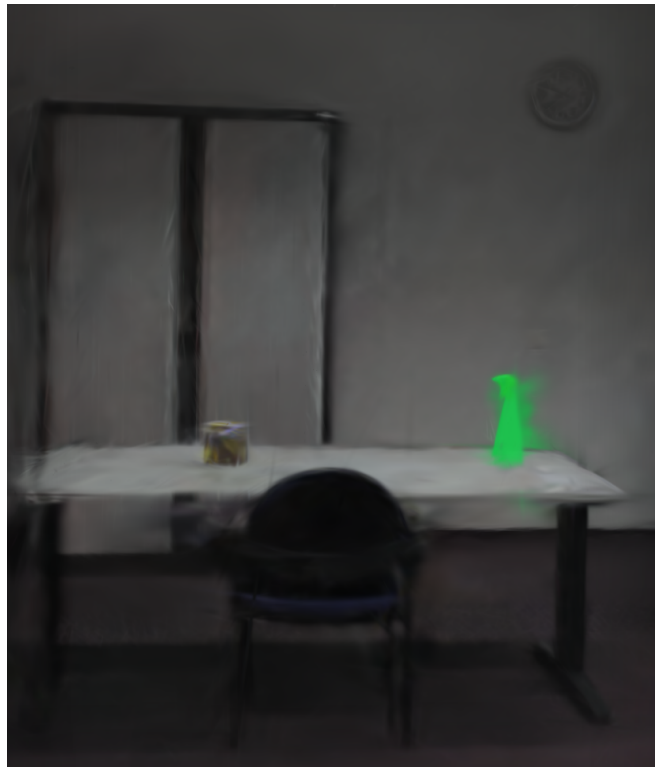
De volledig geconstrained variant schakelt zowel positie-optimalisatie als densification volledig uit. De Gaussians blijven gedurende de volledige training gefixeerd op hun initiële LiDAR-positie en er worden geen nieuwe splats aangemaakt. Enkel de kleur (via spherical harmonics), de schaal en de rotatie van elke Gaussian worden nog geoptimaliseerd.



Figuur 4.8: Gerenderd beeld van de volledig geconstrained variant (geen positie-optimalisatie, geen densification) vanuit hetzelfde standpunt als Figuur 4.2.

Figuur 4.8 toont het gerenderde beeld van dit model. De visuele kwaliteit is merkbaar lager dan bij de baseline: de contouren van het kastframe zijn waziger en de algemene scherpte van de scène is verminderd. De kwantitatieve evaluatie bevestigt dit: de baseline behaalt een PSNR van 30.86 dB, een SSIM van 0,930 en een LPIPS van 0,296. Dit is te verwachten: zonder positievrijheid kunnen de Gaussians zich niet hergroeperen om gebieden met onvoldoende dekking op te vullen, en zonder densification blijft het model beperkt tot de initiële 1 148 098 splats. De fotometrische verliesfunctie convergeert daardoor naar een suboptimaal lokaal minimum.

De forensische betrouwbaarheidsanalyse toont een sterk contrasterend beeld ten opzichte van de baseline.



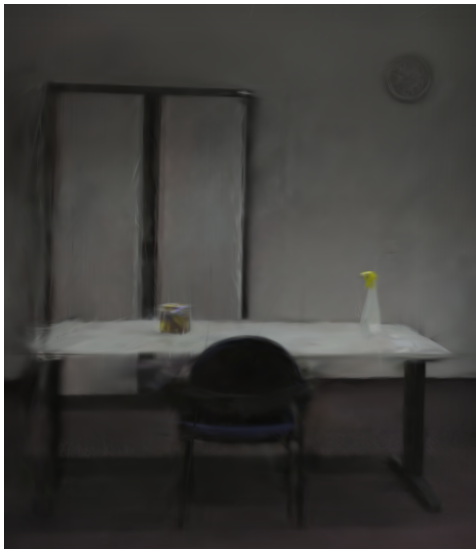
Figuur 4.9: Forensische betrouwbaarheidsoverlay van de volledig geconstrained variant. De spuitfles wordt in groen weergegeven, wat aangeeft dat de bijhorende Gaussians forensisch betrouwbaar zijn gebleven.

Figuur 4.9 toont dat de spuitfles nu in groen wordt weergegeven: de bijhorende Gaussians bevinden zich nog steeds in de buurt van hun initiële positie en worden als forensisch betrouwbaar beschouwd. Dit is het directe gevolg van het uitschakelen van positie-optimalisatie: de Gaussians zijn letterlijk niet in staat geweest om van hun startpositie weg te bewegen, waardoor de geometrische verankering aan de LiDAR-meting gegarandeerd is.

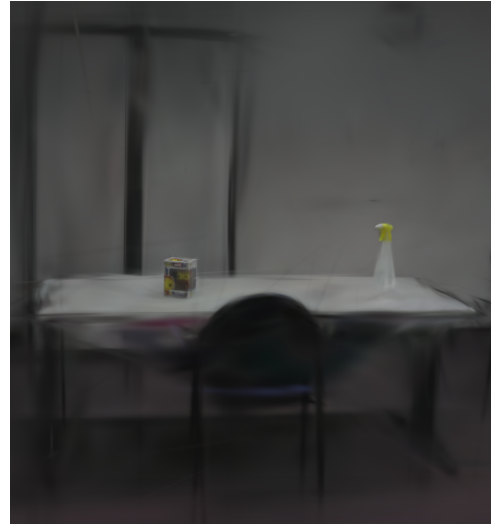
Dit grensgeval bevestigt dat de forensische betrouwbaarheidsmodule correct functioneert: het verschil in overlay tussen baseline en geconstrained variant is consistent met de verwachte impact van de trainingsparameters. De centrale onderzoeksvraag, die in de volgende secties wordt beantwoord, is of er een tussenpositie bestaat die voldoende visuele kwaliteit combineert met een aanvaardbare mate van forensische betrouwbaarheid.

Vergelijking met de sparse SfM-reconstructie

Ter referentie toont Figuur 4.10 een vergelijking van de reconstructie op basis van de sparse COLMAP-puntenwolk en die van de LiDAR-puntenwolk. Kwantitatief presteert de SfM-reconstructie met een PSNR van 16.11 dB, SSIM van 0,699 en LPIPS van 0,391 slechter dan de LiDAR-gebaseerde reconstructie. Ten opzichte van de sparse reconstructie uit sectie 4.2.1 presteert deze constrainede reconstructie zowel visueel als kwantitatief ook veel slechter.



(a) Reconstructie op basis van de LiDAR-puntenwolk.



(b) Reconstructie op basis van de sparse SfM-puntenwolk.

Figuur 4.10: Constrainede reconstructie vergelijking tussen LiDAR- en COLMAP-puntenwolk.

4.3 Parameterstudie

Op basis van de twee grensgevallen uit Sectie 4.2 wordt in deze sectie onderzocht welke parameters de grootste invloed hebben op de balans tussen visuele kwaliteit en forensische betrouwbaarheid. Twee groepen van parameters worden systematisch gevarieerd: de **positie-optimalisatie** en de **opacity**-parameters.

De keuze voor deze twee groepen is gemotiveerd door hun directe impact op de geometrische vrijheid van de Gaussians. Positie-optimalisatie bepaalt in welke mate individuele splats van hun initiële LiDAR-positie mogen afwijken: een hogere learning rate geeft het model meer vrijheid om fotometrische fouten te minimaliseren, maar vergroot tegelijkertijd het risico op geometrische drift. De opacity-parameters bepalen hoe snel Gaussians naar lage opacity-waarden convergeren en bijgevolg hoe frequent ze door het MCMC-mechanisme worden herplaatst: een agressieve opacity-regulering drijft splats sneller naar herplaatsing, terwijl een te lakse regulering Gaussians langer op hun huidige positie laat staan en de positie-optimalisatie meer tijd geeft om geometrische drift te accumuleren.

Densification wordt in de experimenten niet als afzonderlijke variabele onderzocht. De reden hiervoor is dat het maximum aantal Gaussians bewust iets boven het aantal initiële LiDAR-punten is ingesteld, namelijk 1 150 000 tegenover 1 148 098 initiële punten. Door dit smalle marge kan densification slechts een verwaarloosbaar aantal nieuwe splats aanmaken. Het ingeschakeld houden van densification heeft in deze configuratie echter wel een positief effect: het MCMC-herplaatsingsmechanisme kan de initiële punten verfijnen en lokaal herpositioneren waar de fotometrische fout dat vereist, zonder dat het model substantieel uitbreidt voorbij de LiDAR-geometrie. Densification wordt daarom in alle experimenten ingeschakeld gelaten, maar de effectieve impact ervan is door de capaciteitslimiet inherent beperkt.

Een bijkomende reden om densification niet als experimentele variabele te behandelen is dat het verlagen ervan geen merkbare invloed heeft op de forensische betrouwbaarheidsoverlay. Zoals vastgesteld in de grensgevallenanalyse en bevestigd door een voorlopige experiment waarbij refinement vroeg werd stopgezet (na 10,000 iteraties), wordt de betrouwbaarheidsscore bij de standaard parameterinstellingen primair bepaald door de opacity-parameters: het is het MCMC-herplaatsingsmechanisme, aangestuurd door de opacity learning rate en opacity regularization,

dat beslist welke Gaussians van hun initiële positie afdrijven. Of er al dan niet nieuwe splats worden aangemaakt via densification heeft daarop een ondergeschikte invloed.

De overige parameters worden in alle experimenten op de standaardwaarden gehouden zoals samengevat in tabel 4.1.

4.3.1 Invloed van opacity-parameters

De opacity-parameters bepalen hoe agressief het model omgaat met transparante Gaussians. Twee parameters worden in deze subsectie gevarieerd: de `Opacity learning rate`, die bepaalt hoe snel de transparantie van een Gaussian zich aanpast tijdens de optimalisatie, en de `Opacity Regularization`, die een extra verliesterm toevoegt die lage-opacity Gaussians bestraft. In de MCMC-strategie worden laag-opacity Gaussians niet verwijderd maar herplaatst naar gebieden met een hogere fotometrische fout. De opacity learning rate bepaalt dus indirect hoe snel splats naar een lage opacity convergeren en bijgevolg hoe snel ze in aanmerking komen voor herplaatsing. Een lagere opacity learning rate vertraagt dit proces: Gaussians convergeren trager naar lage opacity-waarden en worden minder frequent herplaatst, waardoor ze langer op hun huidige positie blijven. In de baseline bedragen de opacity learning rate en opacity regularization respectievelijk 0,0250 en 0,0100.

De aanleiding voor het onderzoeken van deze parameters was, zoals eerder aangehaald, een voorlopig experiment waarbij de refinement vroeg werd stopgezet na 10,000 iteraties. In dat experiment was het aantal nieuw aangemaakte splats ten opzichte van de initiële puntenwolk minimaal, wat betekende dat densification nauwelijks had bijgedragen. Toch toonde de hallucination mask (kleurcodering van splats of deze al dan niet zijn bijgemaakt) in dat model vrijwel de volledige scène in rood: bijna alle Gaussians werden als geometrisch onbetrouwbaar geclassificeerd ondanks de beperkte densification-activiteit. Dit is visueel te zien in Figuur 4.11 Dit wees erop dat niet densification maar het MCMC-herplaatsingsmechanisme zelf de voornaamste bron van geometrische drift is: ook zonder nieuwe splats aan te maken herplaatst MCMC bestaande Gaussians continu op basis van hun opacity, waardoor de initiële LiDAR-verankering geleidelijk verloren gaat. Opacity-regulering werd daarmee geïdentificeerd als een kritische parameter voor forensische betrouwbaarheid.



Figuur 4.11: Hallucination mask van het experiment met vroege refinement-stop.

Experiment 1: verlaagde opacity learning rate en opacity regularization (0,0010 / 0,0010)

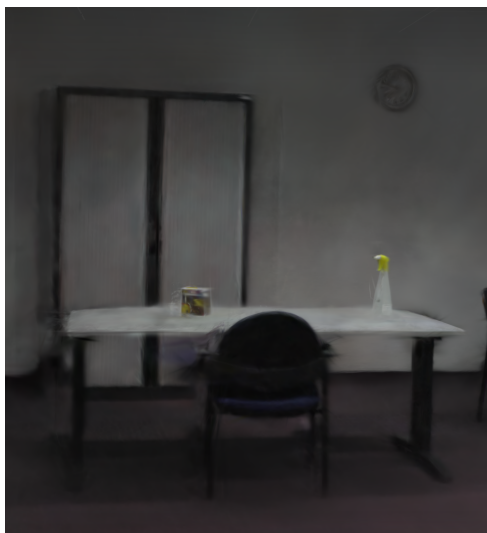
In dit eerste experiment worden zowel de opacity learning rate als de opacity regularization sterk verlaagd van 0,0250 naar 0,0010 respectievelijk van 0,0100 naar 0,0010. Door de tragere opacity-update convergeren Gaussians minder snel naar lage opacity-waarden, waardoor ze minder frequent worden herplaatst door het MCMC-mechanisme. Tegelijkertijd vermindert de lagere regularisatiedruk de neiging van het model om transparante splats actief te bestraffen.

Het gecombineerde effect is dat Gaussians langer op hun huidige positie blijven, ook wanneer ze fotometrisch weinig bijdragen.

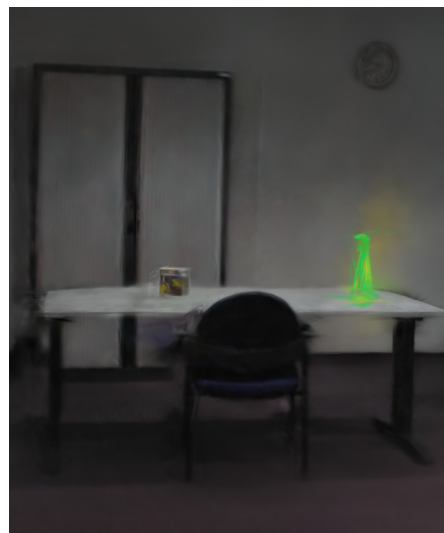


Figuur 4.12: Hallucination mask van Experiment 1 (opacity 0,0010 / 0,0010).

Figuur 4.12 toont de hallucination mask van dit experiment. Het beeld wordt gedomineerd door een uitgebreide rode en blauwe kleuring over vrijwel de volledige scène, wat aangeeft dat een groot deel van de Gaussians ofwel nieuw is aangemaakt buiten de LiDAR-geometrie (rood) ofwel sterk van hun initiële positie is afgedreven (blauw). Dit is opmerkelijk omdat enkel de opacity-parameters zijn aangepast en de positie learning rate en densification nog op hun standaardwaarden staan. De verklaring ligt in de interactie tussen opacity en het MCMC-herplaatsingsmechanisme: doordat Gaussians trager naar lage opacity convergeren, worden ze minder frequent herplaatst. Dit geeft de positie-optimalisatie meer tijd om individuele splats weg te drijven van hun initiële LiDAR-positie voordat het MCMC-mechanisme hen zou herpositioneren, wat de geometrische drift over de volledige scène versterkt.



(a) Gerenderd beeld van Experiment 1.



(b) Betrouwbaarheidsoverlay van Experiment 1.

Figuur 4.13: Render en forensic overlay van Experiment 1.

Figuur 4.13a toont het gerenderde beeld. Visueel is de scène herkenbaar: de tafel, stoelen, het kastframe en de objecten zijn aanwezig en de globale lay-out van de ruimte is correct. In vergelijking met de vrije baseline is de render iets waziger, wat consistent is met de verminderde

herplaatsingsactiviteit: Gaussians die fotometrisch weinig bijdragen blijven langer actief en introduceren een lichte nevelvorming over de scène. De kwantitatieve evaluatie bevestigt dit ook. De reconstructie scoort op PSNR 31.910 dB, op SSIM 0,937 en op LPIPS 0,271. Dit is zoals verwacht slechter dan de vrije baseline reconstructie, maar beter dan de constrainede baseline reconstructie.

Figuur 4.13b toont de forensische betrouwbaarheidsoverlay. De spuitfles wordt grotendeels groen weergegeven en is daarmee betrouwbaarder dan in de baseline, maar vertoont ook gele oppervlakken, wat wijst op Gaussians die matig betrouwbaar zijn. De verklaring hiervoor is dat de splats die de fijne geometrie van de spuitfles modelleren, zoals te zien in Figuur 4.12, wel enige drift hebben ondergaan.

Dit experiment illustreert dat het verlagen van de opacity-parameters een dubbelzinnig effect heeft: lokaal kan de forensische betrouwbaarheid van individuele objecten verbeteren doordat Gaussians minder frequent worden herplaatst, maar globaal neemt de geometrische integriteit af omdat diezelfde verminderde herplaatsing de positie-optimalisatie meer ruimte geeft om splats weg te drijven van de LiDAR-geometrie.

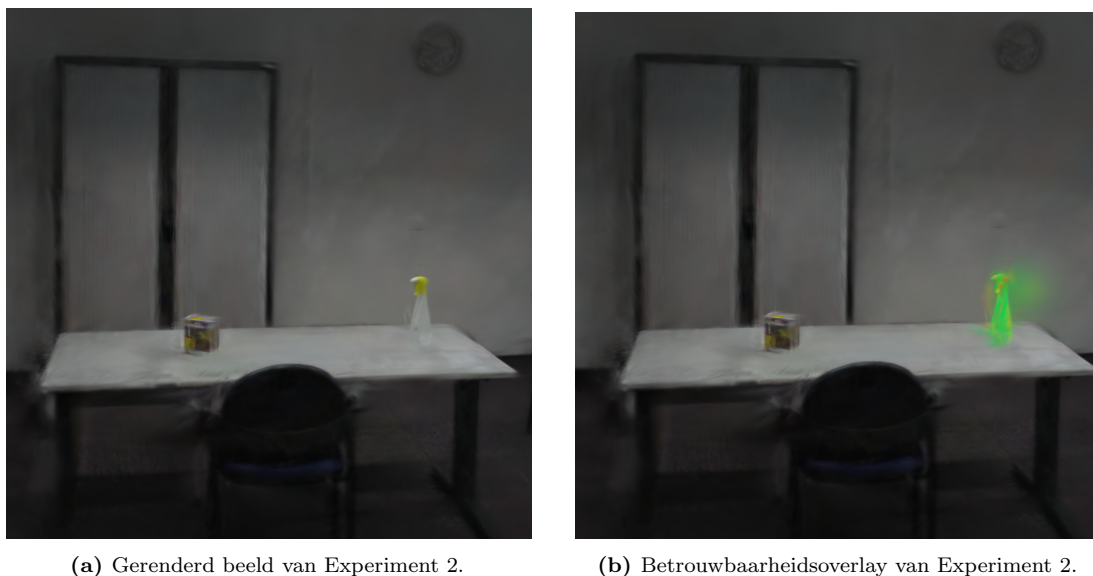
Experiment 2: verlaagde positie learning rate (0,000008) met verlaagde opacity-parameters (0,0010 / 0,0010)

In dit experiment wordt bovenop de verlaagde opacity-parameters uit Experiment 1 ook de positie learning rate verlaagd van 0,000016 naar 0,000008. Door de tragere positie-update kunnen Gaussians zich per iteratie minder ver verplaatsen van hun initiële LiDAR-positie, wat de geometrische drift in principe zou moeten beperken.



Figuur 4.14: Hallucination mask van Experiment 2 (positie learning rate 0,000008, opacity 0,0010 / 0,0010).

Figuur 4.14 toont de hallucination mask. Net als bij Experiment 1 wordt de scène gedomineerd door rode en blauwe tinten, wat aangeeft dat de geometrische drift niet substantieel is afgenomen ondanks de lagere positie learning rate. Het blauwe aandeel, dat verplaatste splats aanduidt, lijkt wel een beetje te zijn afgenomen.



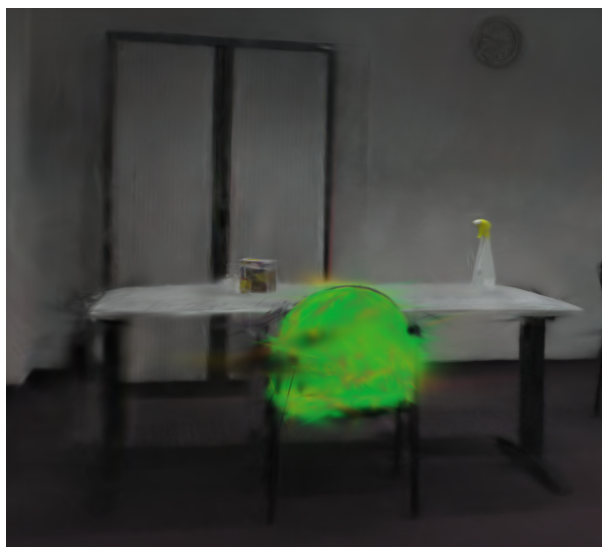
(a) Gerenderd beeld van Experiment 2.

(b) Betrouwbaarheidsoverlay van Experiment 2.

Figuur 4.15: Render en forensic overlay van Experiment 2.

Figuur 4.15a toont het gerenderde beeld. De visuele kwaliteit is vergelijkbaar met Experiment 1: de scène is herkenbaar, de objecten op tafel zijn duidelijk zichtbaar en de globale structuur van de ruimte is correct weergegeven. Als we kijken naar de kwantitatieve evaluatie bekomen we dezelfde conclusie, de PSNR-score is 31.753 dB, de SSIM-score is 0,936 en de LPIPS-score is 0,277. Dit ligt zeer dicht bij de scores van experiment 1.

Figuur 4.15b toont de forensische overlay. De spuitfles wordt zo goed als volledig in groen weergegeven, wat een verbetering is ten opzichte van zowel de baseline als Experiment 1 waar wel nog veel gele zones zichtbaar waren. Dit suggereert dat de combinatie van een lagere positie learning rate en verlaagde opacity-parameters de lokale geometrische verankering van dunne objecten zoals de spuitfles verder verbetert. In Figuur 4.16 kunnen we zien dat in de rest van de scène over het algemeen geen uitgesproken rode of oranje zones in de overlay zitten, wat consistent is met een model dat globaal dichter bij de initiële LiDAR-geometrie is gebleven, ondanks dat dit gebied in de hallucination mask toch aanzienlijk wat blauw aanwezig was.

**Figuur 4.16:** Betrouwbaarheidsoverlay van stoel Experiment 2.

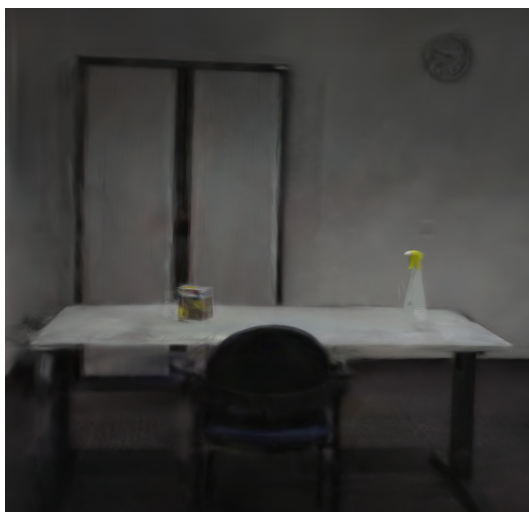
Experiment 3: verlaagde positie learning rate (0,000008), verlaagde opacity (0,0010) en gedeeltelijk herstelde opacity regularization (0,005)

In dit experiment wordt de opacity regularization verhoogd van 0,0010 naar 0,005, dichterbij de standaardwaarde van 0,0100, terwijl de positie learning rate (0,000008) en opacity learning rate (0,0010) ongewijzigd blijven ten opzichte van Experiment 2. Het idee achter deze aanpassing is dat een hogere regularisatiedruk op transparante splats het MCMC-herplaatsingsmechanisme actiever maakt, wat de visuele kwaliteit ten goede zou kunnen komen doordat Gaussians frequenter worden herpositioneerd naar gebieden met hogere fotometrische fout.

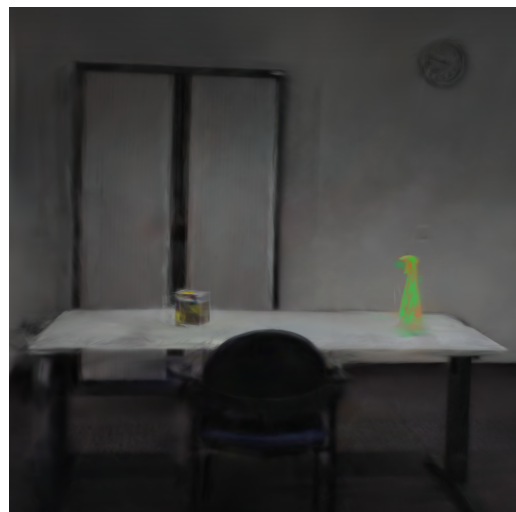


Figuur 4.17: Hallucination mask van Experiment 3 (positie learning rate 0,000008, opacity 0,0010, opacity regularization 0,005).

Figuur 4.17 toont de hallucination mask. In vergelijking met Experiment 2 bevat de mask opnieuw meer rood, wat aangeeft dat het verhogen van de opacity regularization het verwachte effect heeft: Gaussians convergeren sneller naar lage opacity-waarden en worden frequenter herplaatst door het MCMC-mechanisme. Die verhoogde herplaatsingsactiviteit leidt ertoe dat meer splats hun initiële LiDAR-positie verlaten en als geometrisch onbetrouwbaar worden geclassificeerd. Het patroon bevestigt daarmee de eerder vastgestelde relatie: een hogere opacity regularization activeert het herplaatsingsmechanisme, maar gaat gepaard met een grotere geometrische drift.



(a) Gerenderd beeld van Experiment 3.



(b) Betrouwbaarheidsoverlay van Experiment 3.

Figuur 4.18: Render en forensic overlay van Experiment 3.

Figuur 4.18a toont het gerenderde beeld. De visuele kwaliteit blijft vergelijkbaar met Experiment 2: de scène is herkenbaar en de objecten op tafel zijn zichtbaar, maar de render is opnieuw donkerder dan de baseline. De PSNR-score, SSIM-score en LPIPS-score zijn respectievelijk: 31.707 dB, 0,936 en 0,277. Dit is net iets slechter dan experiment 2, maar wel beter dan de constrainede baseline. De beoogde visuele verbetering door de hogere regularisatiedruk is dus niet eenduidig zichtbaar, wat suggereert dat de opacity learning rate van 0,0010 de voornaamste bottleneck blijft: zolang de opacity-waarden traag evolueren, heeft de regularisatieterm beperkte impact op de uiteindelijke renderingkwaliteit.

Figuur 4.18b toont de forensische overlay. De spuitfles wordt opnieuw in groen en rood weergegeven, een lichte achteruitgang ten opzichte van Experiment 2 waar de spuitfles zo goed als volledig groen was. Dit is consistent met de verhoogde herplaatsingsactiviteit die ook in de hallucination mask zichtbaar is: door de hogere opacity regularization worden splats frequenter herplaatst, wat de geometrische verankering van fijne objecten zoals de spuitfles gedeeltelijk tenietdoet.

Experiment 3 illustreert dat het gedeeltelijk herstellen van de opacity regularization naar de standaardwaarde niet leidt tot de beoogde combinatie van betere visuele kwaliteit en behoud van forensische betrouwbaarheid. De twee doelstellingen staan in directe spanning met elkaar: een hogere regularisatiedruk activeert het MCMC-herplaatsingsmechanisme en verbetert potentieel de renderingkwaliteit, maar ondermijnt tegelijkertijd de geometrische verankering aan de LiDAR-initialisatie.

Experiment 4: verlaagde positie learning rate (0,000008), standaard opacity (0,0250) en verlaagde opacity regularization (0,001)

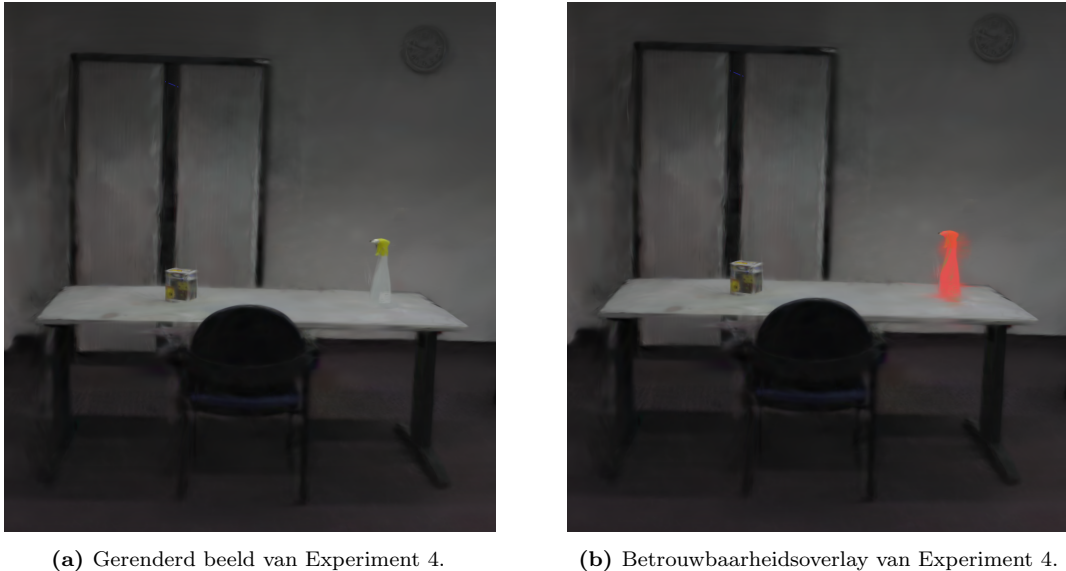
In dit experiment wordt de opacity learning rate teruggezet naar de standaardwaarde van 0,0250, terwijl de opacity regularization verlaagd blijft op 0,001 en de positie learning rate op 0,000008 staat. Het doel is te onderzoeken of de standaard opacity learning rate de visuele kwaliteit kan herstellen, terwijl de verlaagde regularization de herplaatsingsdruk beperkt houdt.



Figuur 4.19: Hallucination mask van Experiment 4 (positie learning rate 0,000008, opacity 0,0250, opacity regularization 0,001).

Figuur 4.19 toont de hallucination mask van experiment 4. Ten opzichte van de vorige experimenten is de mask opvallend anders van karakter: de rode dominantie is sterk toegenomen en het blauw is vrijwel afwezig. Dit patroon wijst erop dat het MCMC-herplaatsingsmechanisme zeer actief is geweest, maar dat de herplaatste Gaussians relatief dicht bij hun nieuwe positie

zijn gebleven in plaats van ver af te drijven. De massale rode kleuring duidt op een groot aantal splats dat zijn index-correspondentie met de initiële LiDAR-puntenwolk heeft verloren door herplaatsing, zonder dat er sprake is van grote absolute positieafwijkingen zoals blauw zou aanduiden.



Figuur 4.20: Render en forensic overlay van Experiment 4.

Figuur 4.20a toont het gerenderde beeld. De visuele kwaliteit is merkbaar verbeterd ten opzichte van de vorige experimenten: de render is helderder en de objecten op tafel zijn iets scherper weergegeven. Dit wordt ook bevestigd door de kwantitatieve evaluatie waar de PSNR-score 32.305 dB is, SSIM-score 0,939 is en de LPIPS-score 0,272 is. Dit experiment scoort ten opzichte van experiment 1 beter op PSNR en op SSIM, maar slechter op LPIPS. Dit bevestigt dat de standaard opacity learning rate een positieve invloed heeft op de renderingkwaliteit doordat het MCMC-herplaatsingsmechanisme actiever Gaussians herpositioneert naar gebieden met hoge fotometrische fout.

Figuur 4.20b toont echter een duidelijke forensische achteruitgang: de spuitfles wordt volledig in rood weergegeven, vergelijkbaar met de onbeperkte baseline uit Sectie 4.2.1. De hogere opacity learning rate heeft de herplaatsingsactiviteit dusdanig vergroot dat de Gaussians die de spuitfles modelleren ver van hun initiële LiDAR-positie zijn afgedreven. De verlaagde opacity regularization heeft deze drift niet kunnen compenseren: hoewel de regularisatiedruk lager is, overheerst de snellere opacity-convergentie het totaalgedrag van het model.

Experiment 4 maakt opnieuw duidelijk dat visuele kwaliteit en forensische betrouwbaarheid lastig te verenigen zijn. Een hogere opacity learning rate herstelt de renderingkwaliteit naar een niveau vergelijkbaar met de baseline, maar doet dit ten koste van de geometrische verankering. De combinatie van standaard opacity learning rate met verlaagde opacity regularization biedt geen werkbaar compromis: de twee parameters versterken elkaars negatieve forensische effect eerder dan dat ze elkaar compenseren.

Experiment 5: verlaagde positie learning rate (0,000008), gedeeltelijk verlaagde opacity (0,01) en verlaagde opacity regularization (0,001)

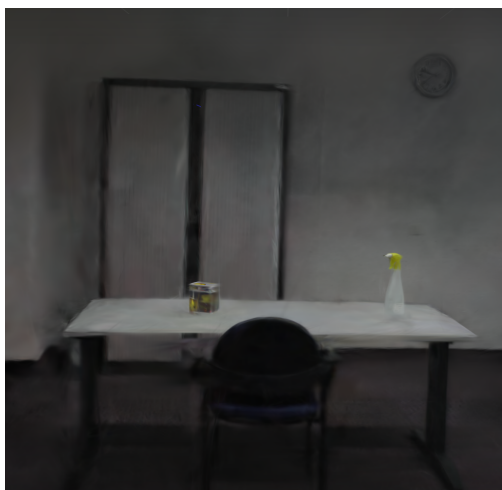
In dit experiment wordt de opacity learning rate ingesteld op 0,01, een tussenwaarde tussen de sterk verlaagde 0,0010 uit Experimenten 1 en 2 en de standaardwaarde van 0,0250 uit Experiment 4. De opacity regularization blijft verlaagd op 0,001 en de positie learning rate op 0,000008. Het doel is te onderzoeken of een gematigde opacity learning rate een betere balans oplevert tussen

de renderingkwaliteit van Experiment 4 en de forensische betrouwbaarheid van Experiment 2.

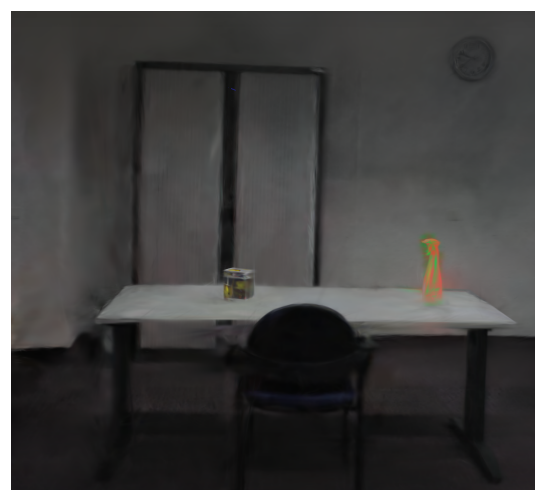


Figuur 4.21: Hallucination mask van Experiment 5 (positie learning rate 0,000008, opacity 0,01, opacity regularization 0,001).

Figuur 4.21 toont de hallucination mask. Het patroon is vergelijkbaar met dat van Experiment 1 en 2: een brede verspreiding van rood en blauw over de volledige scène, wat wijst op aanzienlijke geometrische drift en herplaatsingsactiviteit. Ten opzichte van Experiment 4, waar rood domineerde zonder noemenswaardig blauw, is hier opnieuw meer blauw aanwezig, wat suggereert dat de gematigde opacity learning rate het MCMC-mechanisme iets trager aanstuurt dan de standaardwaarde maar nog steeds voldoende actief om substantiële positieafwijkingen te accumuleren over de volledige trainingsloop.



(a) Gerenderd beeld van Experiment 5.



(b) Betrouwbaarheidsoverlay van Experiment 5.

Figuur 4.22: Render en forensic overlay van Experiment 5.

Figuur 4.22a toont het gerenderde beeld. De visuele kwaliteit ligt tussen Experiment 2 en Experiment 4 in: de scène is helderder dan bij de sterk verlaagde opacity-experimenten maar donkerder dan bij Experiment 4. Ook als we de kwantitatieve metrieken (PSNR-score is 31.818 dB, SSIM-score is 0,937, LPIPS-score is 0,274) vergelijken komen we tot dezelfde conclusie. De objecten op tafel zijn duidelijk herkenbaar en de globale structuur van de ruimte is correct weergegeven. Dit bevestigt dat de opacity learning rate een directe invloed heeft op de helderheid en scherpte van de render: een hogere waarde activeert het herplaatsingsmechanisme frequenter, wat de fotometrische kwaliteit ten goede komt.

Figuur 4.22b toont de forensische overlay. De spuitfles wordt weergegeven in een combinatie van groen en rood, met rode zones die de bovenzijde en het lichaam van de fles domineren. Dit is een achteruitgang ten opzichte van Experiment 2, waar de fles volledig groen was, maar een verbetering ten opzichte van Experiment 4, waar de fles volledig rood was. De gematigde opacity learning rate levert dus een tussenresultaat op, zowel visueel als forensisch, zonder dat één van beide doelstellingen volledig wordt bereikt.

Experiment 5 bevestigt dat een tussenwaarde voor de opacity learning rate geen werkelijk compromis oplevert: het model presteert op beide assen matig in plaats van op één as sterk. De resultaten van de experimenten tot nu toe wijzen er gezamenlijk op dat de opacity learning rate en opacity regularization te sterk gekoppeld zijn aan het MCMC-herplaatsingsmechanisme om onafhankelijk van de positie learning rate te worden geoptimaliseerd voor forensische betrouwbaarheid.

Experiment 6: verlaagde positie learning rate (0,000008), verlaagde opacity (0,005) en verlaagde opacity regularization (0,001)

In dit experiment wordt de opacity learning rate verder verlaagd naar 0,005, een tussenwaarde tussen de 0,0010 uit Experimenten 1 en 2 en de 0,01 uit Experiment 5. De opacity regularization blijft op 0,001 en de positie learning rate op 0,000008. Het doel is te onderzoeken of een verder verlaagde maar niet minimale opacity learning rate de geometrische drift beter kan beperken dan Experiment 5, zonder de renderingkwaliteit even sterk te verliezen als bij de sterk verlaagde waarden uit Experimenten 1 en 2.



Figuur 4.23: Hallucination mask van Experiment 6 (positie learning rate 0,000008, opacity 0,005, opacity regularization 0,001).

Figuur 4.23 toont de hallucination mask. In vergelijking met Experiment 5 is de blauwe component iets minder aanwezig, wat consistent is met de lagere opacity learning rate: Gaussians convergeren trager naar lage opacity-waarden en worden minder frequent herplaatst, waardoor de absolute positieafwijkingen ten opzichte van de initiële LiDAR-geometrie licht afnemen. De rode component blijft echter vergelijkbaar met Experiment 5, wat aangeeft dat het aantal nieuw aangemaakte splats buiten de LiDAR-geometrie nauwelijks verschilt. Dit suggereert dat de verlaging van de opacity learning rate van 0,01 naar 0,005 wel een beperkte invloed heeft op de mate van positiedrift, maar geen merkbare impact heeft op de aanmaak van nieuwe splats via het MCMC-mechanisme.



(a) Gerenderd beeld van Experiment 6.

(b) Betrouwbaarheidsoverlay van Experiment 6.

Figuur 4.24: Render en forensic overlay van Experiment 6.

Figuur 4.24a toont het gerenderde beeld. De visuele kwaliteit is merkbaar beter dan bij de experimenten met sterk verlaagde opacity: de scène is helderder, de spuitfles en de Funko Pop-figuur zijn duidelijk herkenbaar. Ten opzichte van Experiment 5 is de visuele kwaliteit vergelijkbaar, wat suggereert dat de verdere verlaging van de opacity learning rate van 0,01 naar 0,005 geen merkbare visuele achteruitgang veroorzaakt. Ook de kwantitatieve metriecken liggen met een PSNR-score van 31.828 dB, een SSIM-score van 0,937 en een LPIPS-score van 0,275 zeer dicht bij die van experiment 5.

Figuur 4.24b toont de forensische overlay. De spuitfles wordt weergegeven in een mix van groen en rood, maar met iets meer groene zones dan in Experiment 5. Dit is consistent met de licht verminderde positiedrift die ook in de hallucination mask zichtbaar was: de lagere opacity learning rate zorgt ervoor dat een iets groter deel van de Gaussians die de spuitfles modelleren dicht bij hun initiële LiDAR-positie is gebleven, wat zich vertaalt in een marginaal betere betrouwbaarheidsscore. Het verschil is echter beperkt en de rode zones blijven dominant aanwezig, wat aangeeft dat de verlaging van 0,01 naar 0,005 onvoldoende is om de forensische betrouwbaarheid van dunne objecten zoals de spuitfles structureel te verbeteren.

Experiment 6 bevestigt het patroon dat over de voorgaande experimenten is opgebouwd: een verlaging van de opacity learning rate tussen 0,001 en 0,025 levert geen werkelijk compromis op tussen visuele kwaliteit en forensische betrouwbaarheid. De twee parameters sturen het MCMC-herplaatsingsmechanisme in tegengestelde richtingen: een hogere waarde verbetert de renderingkwaliteit maar vergroot de geometrische drift, een lagere waarde beperkt de herplaatsing maar geeft de positie-optimalisatie meer ruimte om dezelfde drift te accumuleren via een langzamer maar even effectief pad.

4.4 Aanbevolen configuratie

Op basis van de experimenten in Sectie 4.3 en de grensgevallen uit Sectie 4.2 wordt in deze sectie een aanbeveling geformuleerd voor het gebruik van Gaussian Splatting als forensisch bewijsmateriaal of illustratiemateriaal in een gerechtelijke context. De aanbeveling onderscheidt twee scenario's: een reconstructie die primair dient als visueel illustratiemateriaal ter ondersteuning van getuigenissen en reconstructies en een reconstructie die forensisch strikt betrouwbaar moet zijn en waarbij de geometrie herleidbaar moet blijven tot de originele LiDAR-meting. Een overzicht van de bevindingen is terug te vinden in Tabel 4.2.

Experiment	PSNR	SSIM	LPIPS	Betrouwbaarheidsscore
Experiment 1	31.910 dB	0,937	0,271	Grotendeels hoog met matige zones
Experiment 2	31.753 dB	0,936	0,277	Bijna volledig hoog
Experiment 3	31.707 dB	0,936	0,277	Hoog met lage zones
Experiment 4	32.305 dB	0,939	0,272	Laag
Experiment 5	31.818 dB	0,937	0,274	Voornamelijk laag
Experiment 6	31.828 dB	0,937	0,275	Laag met hoge zones

Tabel 4.2: Samenvattende tabel kwantitatieve evaluatie en betrouwbaarheidsscores van de uitgevoerde experimenten.

4.4.1 Gebruik als illustratiemateriaal

Wanneer de Gaussian Splatting-reconstructie wordt ingezet als visueel ondersteuningsmateriaal, bijvoorbeeld om een rechter of jury een intuïtief beeld te geven van de scène, biedt Experiment 2 de beste balans. De combinatie van een verlaagde positie learning rate (0,000008) en sterk verlaagde opacity-parameters (0,0010 / 0,0010) levert een herkenbare, visueel aanvaardbare reconstructie op waarbij de grote meerderheid van de scène forensisch betrouwbaar blijft. De forensische overlay toont voor de typische elementen van een binnenruimte zoals wanden, vloer en meubels overwegend groene zones, wat wijst op een goede geometrische verankering aan de LiDAR-meting.

Een belangrijke beperking bij dit gebruik is dat fijne, dunne objecten zoals de spuitfles in de testscène ook in Experiment 2 lichte geometrische drift kunnen vertonen, zichtbaar als gele zones in de forensische overlay. Dit is inherent aan de beperkte LiDAR-dekking van dunne geometrieën: de initiële puntenwolk biedt voor zulke objecten een minder dense verankering, waardoor de positie-optimalisatie meer vrijheid heeft om de bijhorende Gaussians te verplaatsen. Bij gebruik als illustratiemateriaal dient de rechtbank hiervan op de hoogte te zijn: de reconstructie geeft een geometrisch correcte weergave van de ruimte als geheel, maar individuele kleine objecten kunnen in beperkte mate afwijken van hun werkelijke positie.

4.4.2 Gebruik als forensisch bewijsmateriaal

Wanneer de geometrische nauwkeurigheid van de reconstructie cruciaal is, bijvoorbeeld bij het bepalen van exacte afstanden, posities van objecten of de reconstructie van een incident, wordt aangeraden om de volledig geconstrained variant te gebruiken zoals beschreven in Sectie 4.2.2. Door positie-optimalisatie en densification volledig uit te schakelen blijft elke Gaussian gegarandeerd op zijn initiële LiDAR-positie, waardoor de geometrische trouw aan de meting maximaal is. De forensische overlay toont in dit geval vrijwel uitsluitend groene zones, wat de rechtbank de zekerheid biedt dat de reconstructie een directe visuele representatie is van de LiDAR-puntenwolk.

Een bijkomende overweging betreft de opacity-parameters, die zelfs bij uitgeschakelde positie-optimalisatie een indirecte invloed kunnen hebben op de geometrische stabiliteit in niet-geconstrained varianten. De opacity learning rate en opacity regularization bepalen samen hoe snel individuele Gaussians naar lage transparantiewaarden convergeren en bijgevolg hoe frequent ze door het MCMC-mechanisme worden herplaatst. Een te agressieve opacity-regulering drijft splats sneller weg van hun initiële LiDAR-positie, terwijl een te lakse regulering de positie-optimalisatie meer ruimte geeft om geometrische drift te accumuleren. Uit de parameterstudie in Sectie 4.3.1 blijkt dat geen enkele combinatie van verlaagde opacity-parameters een bevredigend evenwicht bereikt tussen visuele kwaliteit en forensische betrouwbaarheid: de twee doelstellingen staan in directe spanning met elkaar. Dit bevestigt dat de volledig geconstrained variant, waarbij de opacity-waarden wel nog worden geoptimaliseerd maar de positie van elke Gaussian gefixeerd blijft op de LiDAR-geometrie, de enige configuratie is die volledige geometrische garanties biedt voor forensisch gebruik.

De lagere visuele kwaliteit van dit model is in een forensische context aanvaardbaar: het doel is niet fotorealisme maar geometrische betrouwbaarheid. Wanneer een rechter of deskundige specifieke afstanden of posities wil verifiëren, kunnen deze rechtstreeks worden afgeleid uit de onderliggende LiDAR-puntenwolk, waarvan de Gaussian Splatting-reconstructie in dit geval een directe visuele representatie vormt.

4.4.3 Algemene aanbevelingen voor forensisch gebruik

Ongeacht de gekozen configuratie worden de volgende algemene aanbevelingen geformuleerd voor het gebruik van Gaussian Splatting als forensisch materiaal:

1. **Altijd de forensische betrouwbaarheidsoverlay rapporteren.** De overlay maakt zichtbaar welke delen van de reconstructie geometrisch verankerd zijn en welke mogelijk zijn afgedreven. Het gebruik van de reconstructie zonder deze context kan misleidend zijn voor een rechtbank die de visuele output niet kan onderscheiden van een fotorealistische maar geometrisch onbetrouwbare weergave.
2. **De hallucination mask als aanvullend bewijsstuk opnemen.** De hallucination mask geeft op splat-niveau aan welke Gaussians zijn herplaatst of nieuw zijn aangemaakt ten opzichte van de initiële LiDAR-puntenwolk. Dit biedt een kwantitatieve onderbouwing van de betrouwbaarheidsscore en maakt het mogelijk om bij betwisting aan te tonen welke delen van de reconstructie volledig herleidbaar zijn tot de originele meting.
3. **De LiDAR-puntenwolk als primaire referentie bewaren.** De Gaussian Splatting-reconstructie is een afgeleide representatie van de LiDAR-meting. De originele puntenwolk, met haar volledige metrische nauwkeurigheid, dient als primair bewijsstuk te worden bewaard en ingediend. De Gaussian Splatting-reconstructie fungeert dan als visuele toelichting bij de puntenwolk, niet als vervanging ervan.
4. **De trainingsparameters en pipelinebeschrijving documenteren.** Reproduceerbaarheid is een basisvereiste voor forensisch bewijsmateriaal. De volledige set trainingsparameters, de gebruikte software en de verwerkingsstappen van LiDAR-scan tot Gaussian Splatting-model dienen te worden gedocumenteerd zodat een onafhankelijke deskundige de reconstructie kan reproduceren en verifiëren.

Hoofdstuk 5

Conclusies

5.1 Samenvatting van het onderzoek

Deze masterproef onderzocht in hoeverre Gaussian Splatting inzetbaar is als betrouwbare techniek voor forensische 3D-reconstructie en hoe de geometrische trouw aan meetdata uit een LiDAR-scan kan worden gewaarborgd zonder daarbij de visuele kwaliteit van de reconstructie volledig op te offeren. De kern van het probleem is een tegenstrijdigheid: wat zorgt voor mooie beelden, bemoeilijkt tegelijk de nauwkeurige weergave van vormen en afstanden.

Het onderzoek combineerde een eigen data-acquisitie via video-opname en LiDAR-scanning, een custom alignment pipeline, een uitgebreide parameterstudie en de ontwikkeling van een volledig forensische toolchain, bestaande uit een forensische betrouwbaarheidsoverlay, een hallucination mask, een juridische validatiemodule en een Electron-gebaseerde rechtbankviewer.

5.2 Beantwoording van de onderzoeksvragen

5.2.1 Kan Gaussian Splatting worden verankerd aan LiDAR-geometrie?

Het antwoord is bevestigend. Door de initiële puntenwolk van COLMAP te vervangen door een dense LiDAR-puntenwolk, en door de positie-optimalisatie van de Gaussians te beperken, is het mogelijk om een reconstructie te produceren waarbij de geometrie herleidbaar blijft tot de originele LiDAR-meting. De volledig geconstrained variant, waarbij zowel positie-optimalisatie als densification worden uitgeschakeld, levert een model op dat in forensische zin maximaal betrouwbaar is: elke Gaussian bevindt zich exact op zijn initiële LiDAR-positie en de forensische overlay toont vrijwel uitsluitend groene zones.

Dit gaat uiteraard gepaard met een verlies aan visuele kwaliteit. Omdat het model geen nieuwe splats kan aanmaken en bestaande splats niet kan herpositioneren, kunnen gaten in de LiDAR-dekking niet worden opgevuld. De scène is desondanks herkenbaar en de globale structuur correct. In een forensische context, waarbij geometrische nauwkeurigheid prioriteit heeft boven fotorealisme, is deze trade-off aanvaardbaar.

5.2.2 Is er een bruikbaar compromis tussen visuele kwaliteit en forensische betrouwbaarheid?

De parameterstudie toonde aan dat een gedeeltelijk compromis mogelijk is, zij het met aanzienlijke beperkingen. De experimenten met verlaagde positie learning rate (0,000008) en sterk verlaagde opacity-parameters (0,0010 / 0,0010) leverden een reconstructie op waarbij de grote meerderheid van de scène geometrisch verankerd bleef, terwijl de visuele kwaliteit aanvaardbaar bleef voor gebruik als illustratiemateriaal.

Tegelijkertijd werd duidelijk dat dit compromis niet universeel geldt voor alle objecten in de scène. Dunne objecten met beperkte LiDAR-dekking, zoals de spuitfles in de testscène, vertonen zelfs bij sterk ingeperkte optimalisatie significante geometrische drift. Dit is geen fout in de implementatie maar een fundamenteel gevolg van de beperkte puntendichtheid waarmee LiDAR-scanners fijne geometrieën vastleggen: zonder voldoende initialisatiepunten heeft de positie-optimalisatie meer vrijheid om de bijhorende Gaussians te verplaatsen.

De experimenten wezen bovendien op een structurele koppeling tussen de opacity learning rate, de opacity regularization en het MCMC-herplaatsingsmechanisme. Het bleek niet mogelijk om via tussenwaarden een werkelijk compromis te bereiken: in de meeste gevallen presteerde het model op beide assen matig in plaats van op één as sterk. Dit wijst erop dat de forensisch relevante parameters niet onafhankelijk van elkaar kunnen worden geoptimaliseerd.

5.2.3 Welke configuratie is aanbevolen voor forensisch gebruik?

Op basis van de parameterstudie en de grensgevallen werden twee aanbevolen configuraties geformuleerd. Voor gebruik als visueel illustratiemateriaal, waarbij een rechter of jury een intuïtief beeld van de scène krijgt zonder dat exacte geometrische metingen nodig zijn, biedt de combinatie van een verlaagde positie learning rate en sterk verlaagde opacity-parameters de beste balans. De forensische overlay toont voor de meeste structurele elementen, zoals wanden, vloer en meubilair, overwegend groene zones.

Voor gebruik als forensisch bewijsmateriaal, waarbij exacte afstanden of objectposities gerechtelijk relevant zijn, is de volledig geconstrained variant de enige verantwoorde keuze. Elke andere configuratie laat enige mate van positiedrift toe, die voor specifieke objecten aanzienlijk kan zijn. In dit scenario fungeert de Gaussian Splatting-reconstructie als visuele toelichting bij de LiDAR-puntenwolk, die als primaire referentie bewaard en ingediend dient te worden.

5.3 Bijdragen van dit onderzoek

Eén van de centrale bijdragen van dit werk is de ontwikkeling van een robuuste alignment pipeline die de sparse COLMAP-puntenwolk uitlijnt met de dense LiDAR-data via een combinatie van IQR-gebaseerde outlier filtering, radiale en as-georiënteerde filtering, en ICP-gebaseerde alignment. De resulterende transformatiematrix wordt toegepast op alle cameraposes en de LiDAR-data wordt geconverteerd naar een COLMAP-compatibel formaat, waarbij een eigen ontwikkelde visibility-based downsampling prioriteit geeft aan punten die zichtbaar zijn vanuit de gecalibreerde cameraposes. Dit maakt het mogelijk om de Gaussian Splatting training te initialiseren met metrisch nauwkeurige geometrie, zonder de architectuur van de trainingsomgeving te wijzigen.

Een tweede centrale bijdrage is de ontwikkeling van de forensische analysetools. De betrouwbaarheidsoverlay visualiseert per Gaussian in hoeverre deze is afgedreven van zijn initiële LiDAR-positie, weergegeven als een groen-geel-rood kleurspectrum. De hallucination mask maakt een binaire splitsing tussen originele en nieuw aangemaakte splats, waarbij ook de mate van positionele drift zichtbaar wordt gemaakt via kleurcodering. Samen bieden deze tools een kwantitatieve en visuele onderbouwing van de forensische betrouwbaarheid van de reconstructie.

De derde bijdrage is de juridische validatielaag, bestaande uit een kwantitatieve alignment-validatie, een chain of custody en een RFC 3161-tijdstempelmodule. Het resultaat is een integriteitsgeborgd JSON-rapport met SHA-256-hashing over alle relevante velden, een gedocumenteerde operatoridentiteit en een onafhankelijk verifieerbaar aanmaaktijdstip. Dit rapport kan via de bijhorende Electron-applicatie worden omgezet naar een formeel PDF-deskundigenverslag dat voldoet aan de structurele eisen van de Belgische gerechtelijke procedure.

Het onderzoek toonde ook aan dat Gaussian Splatting en traditionele mesh-based reconstructie complementaire sterke punten hebben. Mesh-based methodes bieden een geometrisch nauw-

keurige en rechtstreeks meetbare representatie, maar missen de fotorealistische kwaliteit en de vloeiende weergave van Gaussian Splatting. Gaussian Splatting levert een visueel overtuigende reconstructie op, maar vereist extra forensische controlemechanismen om de geometrische integriteit te garanderen. De constrained pipeline uit dit onderzoek verkleint dit nadeel aanzienlijk, maar elimineert het niet volledig.

5.4 Beperkingen

De ontwikkelde pipeline kent een aantal concrete technische beperkingen die de toepasbaarheid in de praktijk beïnvloeden.

De alignment tussen de COLMAP-puntenwolk en de LiDAR-scan steunt op ICP, een iteratief algoritme dat convergeert naar een lokaal optimum. Wanneer de initiële uitlijning onvoldoende is kan ICP divergeren of vastlopen in een onjuiste oplossing. De pipeline beschikt niet over een automatisch mechanisme om een mislukte alignment te detecteren of te hervatten. Visuele inspectie van de uitvoer blijft noodzakelijk.

Een tweede beperking betreft de veronderstelling van een statische scène. De pipeline verwacht dat de LiDAR-scan en de videobeelden hetzelfde moment in de tijd vastleggen. In forensische praktijk worden deze opnames echter zelden simultaan uitgevoerd. Zoals Commissaris Claesen aangaf, heerst er op de plaats delict vaak tijdsdruk, waardoor fotodocumentatie en LiDAR-scanning als afzonderlijke, sequentiële stappen worden uitgevoerd. Hierdoor kunnen bewegende objecten of gewijzigde elementen in de scène tot inconsistenties leiden. De pipeline biedt geen mechanisme om tijdsgebonden verschillen te detecteren of te compenseren.

De visibility-berekening, die bepaalt welke LiDAR-punten zichtbaar zijn vanuit de cameraposities, maakt gebruik van het Ball-Pivoting Algorithm om een oppervlaktemesh te construeren voor occlusiecontrole. Dit algoritme presteert slecht op dunne objecten, scherpe geometrie of zones met lage punt dichtheid, waar het geen gesloten mesh kan vormen. In dergelijke gevallen kan de occlusiecheck onbetrouwbaar zijn, waardoor punten ten onrechte als zichtbaar of onzichtbaar worden geclassificeerd.

De pipeline ondersteunt uitsluitend het E57-bestandsformaat als invoer voor LiDAR-data. Andere gangbare formaten zoals LAS, LAZ of PTS worden niet rechtstreeks ingelezen. Daarnaast worden enkel de camera-modellen PINHOLE, SIMPLE_PINHOLE, SIMPLE_RADIAL en RADIAL ondersteund. Andere COLMAP-modellen worden via een fallback afgehandeld die mogelijk tot onjuiste intrinsieke parameters leidt.

5.5 Toekomstig onderzoek en maatschappelijke relevantie

Deze studie wordt uitgevoerd in samenwerking met politiediensten en Europol, die de ontwikkeling van betrouwbare en transparante 3D-reconstructietechnieken als een belangrijke innovatie zien binnen digitaal forensisch onderzoek. De resultaten van dit onderzoek kunnen bijdragen aan de verdere digitalisering van het forensische werkveld, waarbij 3D-reconstructies steeds vaker als bewijs of onderzoeksinstrument kan worden ingezet.

Een logische vervolgstap is de ontwikkeling van een acquisitiesysteem waarbij RGB-video en LiDAR-meting simultaan worden vastgelegd. Systemen zoals de Leica BLK ARC of geïntegreerde sensor-rigs die gebruikt worden in autonome voertuigen combineren beide sensoren op harde-ware niveau. Dit elimineert het tijdsverschil tussen de verschillende data-acquisities en de noodzaak van een expliciete alignment, waardoor de pipeline eenvoudiger, sneller en nauwkeuriger wordt.

De huidige forensische overlay vereist visuele inspectie door een deskundige. Een volgende stap is de ontwikkeling van een geautomatiseerde module die, op basis van de betrouwbaarheidscores en de hallucination mask, proactief waarschuwt wanneer specifieke objecten of zones

forensisch onbetrouwbaar zijn. Deze module zou ook een kwantitatief betrouwbaarheidsrapport per object kunnen genereren, wat de interpretatie voor niet-technische rechtbankdeelnemers vereenvoudigt.

De voorgestelde pipeline is momenteel een research prototype. In toekomstig onderzoek kan verder onderzocht worden hoe deze constrained Gaussian Splatting-technieken geïntegreerd kunnen worden in bestaande software van politiediensten, en hoe automatische validatie of foutdetectie kan helpen om de juridische betrouwbaarheid van 3D-modellen te verhogen. Dit vereist zowel softwarematige integratie als het ontwikkelen van gebruikersinterfaces die geschikt zijn voor niet-technische gebruikers.

De huidige methode gaat ervan uit dat de scène statisch is tijdens zowel de LiDAR-scan als de video-opname. In de praktijk kunnen forensisch relevante scènes echter bewegende elementen bevatten, zoals personen die aanwezig waren tijdens een incident. Het integreren van tijdsgebonden informatie, bijvoorbeeld door het combineren van Gaussian Splatting met videobeelden die beweging bevatten, zou de bruikbaarheid van de methode aanzienlijk uitbreiden.

Op langere termijn kan dit leiden tot een gestandaardiseerd systeem waarin neurale reconstructie wordt gecombineerd met een betrouwbare manier om de waarheid te waarborgen, een belangrijke stap richting de inzet van AI in gerechtelijke bewijsvoering.

5.6 Slotbeschouwing

Gaussian Splatting biedt een unieke combinatie van fotorealistische beeldkwaliteit en flexibele geometrische representatie die traditionele methodes niet kunnen evenaren. De uitdaging voor forensisch gebruik ligt niet in de visualisatietechniek zelf, maar in het verankeren van die visualisatie aan een meetbare, verifieerbare werkelijkheid. Dit onderzoek heeft aangetoond dat die verankering mogelijk is: door de training te initialiseren met LiDAR-geometrie, de optimalisatie te begrenzen en de resulterende reconstructie te voorzien van een gelaagde forensische analyselaag, kan Gaussian Splatting worden ingezet op een manier die zowel visueel overtuigend als juridisch verdedigbaar is.

De ontwikkelde pipeline is geen vervanging voor de LiDAR-puntenwolk als primair bewijsstuk. Ze is een aanvulling: een visuele toelichting die een rechter, jury of deskundige in staat stelt om ruimtelijke informatie intuïtief te interpreteren, zonder de metrische nauwkeurigheid van de onderliggende meting uit het oog te verliezen. In die rol, als transparante, traceerbare en forensisch verantwoorde visualisatietool, heeft Gaussian Splatting een reële toekomst binnen het digitale forensische werkveld.

De samenwerking met de Federale Politie Limburg en de bredere context van innovatie in forensische technologie bevestigen dat er een maatschappelijke nood is aan precies dit soort tools: systemen die de kloof overbruggen tussen technische meetdata en begrijpelijke visuele communicatie, zonder daarbij de wetenschappelijke en juridische integriteit te compromitteren. Dit onderzoek vormt een eerste, concrete stap in die richting.

Bibliografie

- [3Ds25] 3Dsurvey. What is gaussian splatting? <https://3dsurvey.si/what-is-gaussian-splatting/>, 2025.
- [AST22] ASTM. Standard specification for 3d imaging data exchange, version 1.0. <https://store.astm.org/e2807-11r19e01.html>, 2022.
- [Ber24] Bernhard Kerbl and Andreas Meuleman and Georgios Kopanas and Michael Wimmer and Alexandre Lanvin and George Drettakis. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. <https://repo-sam.inria.fr/fungraph/hierarchical-3d-gaussians>, 2024.
- [Car25a] David Caron. pye57: Python .e57 point cloud reader/writer, 2025.
- [Car25b] Rob Carroll. Gaussian splatting vs. 3d meshes: Limitations and trade-offs. <https://www.teraanalytics.xyz/blog/gaussian-splatting-vs.-3d-meshes>, 2025.
- [CPDR01] Adams C., Cain P., Pinkas D., and Zuccherato R. Internet x.509 public key infrastructure time-stamp protocol (tsp). <https://www.rfc-editor.org/rfc/rfc3161>, 2001.
- [Dee25] Marc Deetjen. 3d gaussian splatting. <https://poly.cam/tools/gaussian-splatting>, 2025.
- [Ebe23] Dylan Ebert. Introduction to 3d gaussian splatting. <https://huggingface.co/blog/gaussian-splatting>, 2023.
- [Fan26] Fangjinhua Wang and Qingtian Zhu and Di Chang and Quankai Gao and Junlin Han and Tong Zhang and Richard Hartley and Marc Pollefeys. Learning-based multi-view stereo: A survey. <https://arxiv.org/pdf/2408.15235>, 2026.
- [Fer16] Fernando A. Fardo and Victor H. Conforto and Francisco C. de Oliveira and Paulo S. Rodrigues. A formal evaluation of psnr as quality measurement parameter for image segmentation algorithms. <https://arxiv.org/pdf/1605.07116>, 2016.
- [FRY⁺24] Yalda Foroutan, Daniel Rebain, Kwang Moo Yi, Andrea Tagliasacchi, Simon Fraser University, University of British Columbia, University of Toronto, and Google DeepMind. Does gaussian splatting need sfm initialization? <https://arxiv.org/html/2404.12547v2>, 2024.
- [Gro25] Bonsai Media Group. Exploring gaussian splatting: Transforming photography into 3d reality. <https://bonsaimediagroup.com/blog/exploring-gaussian-splatting-transforming-photography-into-3d-reality>, 2025.
- [Gye25] Gyeongwan Lee and Seunghwan Hong and Junghun Suh. Prior-driven enhancements in 3d gaussian splatting: Normals and depths regularization. <https://isprs-archives.copernicus.org/articles/XLVIII-G-2025/891/2025/isprs-archives-XLVIII-G-2025-891-2025.pdf>, 2025.

- [HMvdW⁺20] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.
- [Hun07] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.
- [Jim20] Jim Nilsson and Tomas Akenine-Möller. Understanding ssim. <https://arxiv.org/pdf/2006.13846>, 2020.
- [Joh16] Johannes L. Schonberger and Jan-Michael Frahm¹. Structure-from-motion revisited. https://openaccess.thecvf.com/content_cvpr_2016/papers/Schonberger_Structure-From-Motion_Revisited_CVPR_2016_paper.pdf, 2016.
- [Joh25] Johannes L. Schoenberger. Colmap frequently asked questions. <https://colmap.github.io/faq.html>, 2025.
- [Juy23] Juyong Zhang and Yuxin Yao and Bailin Deng. Fast and robust iterative closest point. <https://arxiv.org/pdf/2007.07627>, 2023.
- [K. 20] K. Sheppard and S. J. Fieldhouse and J. P. Cassella. Experiences of evidence presentation in court: an insight into the practice of crime scene examiners in england, wales and australia. <https://link.springer.com/article/10.1186/s41935-020-00184-5>, 2020.
- [KKLD23] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. <https://arxiv.org/pdf/2308.04079>, 2023.
- [Lob23] Teresa Lobo. Understanding structure from motion algorithms. <https://medium.com/@loboateresa/understanding-structure-from-motion-algorithms-fc034875fd0c>, 2023.
- [LWMX16] Zhaoxin Li, Kuanquan Wang, Deyu Meng, and Chao Xu. Multi-view stereo via depth map fusion: A coordinate decent optimization method. <https://www.sciencedirect.com/science/article/pii/S0925231215016070>, 2016.
- [LZY21] Jianghua Liao, Jinxing Zhou, and Wentao Yang. Comparing lidar and sfm digital surface models for three land cover types. <https://www.degruyterbrill.com/document/doi/10.1515/geo-2020-0257/html>, 2021.
- [Mar81] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. <https://dl.acm.org/doi/10.1145/358669.358692>, 1981.
- [Nan21] Liangliang Nan. Course notes 5: Multi-view stereo. https://3d.bk.tudelft.nl/courses/geo1016/handouts/06-multi_view_stereo.pdf, 2021.
- [Par23] Parallax. jsPDF — client-side javascript pdf generation. <https://github.com/parallax/jsPDF>, 2023.
- [Pix25] Pix4D. Gaussian splatting at pix4d: a new era of 3d visualization. <https://www.pix4d.com/blog/pix4d-gaussian-splatting-3D-visualization>, 2025.
- [Rad09] Radu Bogdan Rusu and Nico Blodow and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. <https://www.researchgate.net/>

- publication/224557255_Fast_Point_Feature_Histograms_FPFH_for_3D_registration, 2009.
- [Ric18] Richard Zhang and Phillip Isola and Alexei A. Efros and Eli Shechtman and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. https://openaccess.thecvf.com/content_cvpr_2018/papers/Zhang_The_Unreasonable_Effectiveness_CVPR_2018_paper.pdf, 2018.
- [RWW⁺25] Dimitar Rangelov, Sierd Waanders, Kars Waanders, Maurice van Keulen, and Radoslav Miltchev. Impact of data capture methods on 3d reconstruction with gaussian splatting. <https://www.mdpi.com/2313-433X/11/2/65>, 2025.
- [Sch25] Johannes L. Schoenberger. Output format gaussian splatting. <https://colmap.github.io/format.html>, 2025.
- [Sha25] Shakiba Kheradmand and Daniel Rebain and Gopal Sharma and Weiwei Sun and Yang-Che Tseng and Hossam Isack and Abhishek Kar and Andrea Tagliasacchi and Kwang Moo Yi. 3d gaussian splatting as markov chain monte carlo. <https://arxiv.org/pdf/2404.09591>, 2025.
- [Stu25] LichtFeld Studio. Lichtfeld studio. <https://github.com/MrNeRF/LichtFeld-Studio>, 2025.
- [Ton12] Tony Lindeberg. Scale invariant feature transform. https://www.researchgate.net/publication/235355151_Scale_Invariant_Feature_Transform, 2012.
- [Ziy26] Ziyang Yan and Yihua Shao and Minwen Liao and Siyu Chen and Nan Wang and Muyuan Lin and Jenq-Neng Hwang and Hao Zhao and Fabio Remondino and Lei Li. 3dsceneeditor: Controllable 3d scene editing with gaussian splatting. <https://arxiv.org/pdf/2412.01583>, 2026.
- [ZPK18] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.