

Aarhus School of Architecture // Design School Kolding // Royal Danish Academy

Visualisering af punktskyer i landskabsarkitekturen og den fysiske planlægning

Odgaard, Martin

Publication date:
2023

Document Version:
Også kaldet Forlagets PDF

Document License:
CC BY

[Link to publication](#)

Citation for pulished version (APA):
Odgaard, M. (2023). *Visualisering af punktskyer i landskabsarkitekturen og den fysiske planlægning: Working Paper #1 - fra dataforsyningen.dk til arbejdsrum.*

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Visualisering af punktskyer i landskabsarkitekturen og den fysiske planlægning

Working Paper #1 - fra dataforsyningen.dk til arbejdsrum

Version 1.0

Af Martin Odgaard

Arkitekt, Ph.d., Lektor, Arkitektskolen Aarhus

Indledning og problemstilling

Dette arbejdsrapport redegør for potentialer og mulige fremgangsmåder for, på en enkel måde at benytte punktsky-data i by- og landskabsanalyse. Papiret skal ses som en øjenåbner og praktisk anvisning, hvilket måske vil gøre, at dele af den tekniske beskrivelse af fremgangsmetoder, med tiden vil blive uaktuelle.

Punktskyer kan især i forhold til visualisering have sin fordel i forhold til en mesh- eller trianguleret model, idet man i langt højere grad kan opleve transparens den rumlige kompleksitet der kan opstå i- og ved beplantning. Hvor en mesh-model kan insinuere at en overflade er 'lukket', er det omvendt med punktskyerne – her er udgangspunktet at det kun er punkterne der er synlige, og disse punkter kan man så skalere og bearbejde til der opnås en rumligt definerende virkning, eller fastholdes på en måde der i højere grad repræsenterer en beplantnings transparens. Resultatet kan ses som en abstraktion, en repræsentation, med væsentligt anderledes visuel virkning end gængse 3D-mesh baserede modeller.¹

Den afledte pointe er, at triangulerede udgaver af overflademodeller (DSM) måske nok virker fint og visuelt repræsentativt for opfattelsen af volumener, men til at visualisere transparens virkning for skov- og beplantningspartier er det uegnet. Punktskyer kan importeres i AutoCAD-pakken, Rhino og andre CAD-løsninger, og der er således mulighed for at kombinere de to repræsentationsteknikker. I dette arbejdsrapport fokuseres der på tre overordnede tilgange til 'input' – i dette tilfælde fortolket som forskellige typer af omgivelser. Hertil kommer en kort gennemgang af den nødvendige bearbejdning der kan være nødvendig for de forskellige 'input'-data og hvordan de efterfølgende kan visualiseres.

En del af formålet med forsøgene relateret til dette arbejdsrapport er også at afsøge grænserne for automatisk fotogrammetri. Andre fremgangsmåder som f.eks. laser-baseret

¹ Se f.eks. skov-punktskyer (SLU, 2016): <https://www.youtube.com/watch?v=A4CGtticF6A>, og billedkunstneren Alex le Guillous (2019) animationer Forest (2019) <https://alexleguillou.com/Forest> og Woods Walk (2020) <https://alexleguillou.com/Woods-Walk>

SLAM² kan bringes i anvendelse i en skovkontekst pga. metodens høje rumlige opløsning, men et åbent spørgsmål er stadig, om det er muligt også at give et tilstrækkeligt indtryk af den rumlige kontekst med automatisk fotogrammetri i de lavere skalatrin.

En laser-baseret SLAM-tilgang kræver herudover noget forholdsvis dyrt udstyr, hvilket jeg også ønsker at undgå for netop at anvise den bredest muligt tilgængelige metode. RGB-D baseret SLAM kan typisk gennemføres med væsentligt billigere udstyr som f.eks. Kinect, men rækkevidderne på disse typer sensorer er pt. ikke tilstrækkelige (~ max 6 m) til at gøre metoden anvendelig til mit formål hér³.

Udstyr

Til databearbejdning og visualisering har jeg anvendt en forholdsvis almindelig 64-bit Intel x86-baseret Windows 10 bærbar PC med 6 processor-kerner og 32 GB RAM. Computeren har et dedikeret grafikkort af mærket NVidia, så dette har også bidraget til beregning af data via de såkaldte CUDA-kerner⁴ samt til visualiseringen. Til bearbejdning og visualisering af data, har jeg prioriteret at benytte frit tilgængelige data og open source-software for at understrege metodens tilgængelighed. Jeg har benyttet såvel de frie data fra Styrelsen for Dataforsyning og Effektivisering (SDFE)⁵ som selvskabte data.

Jeg har anvendt Meshroom til selv at generere punktskyer, CloudCompare til datakonvertering af punktskyer og Unreal Engine 5.0.2 til visualisering af data. Denne landskabsanalyse vælger jeg hér, skal foregå i spil-udviklingsværktøjet 'Unreal Engine' idet denne kan visualisere punktskyer på en måde så de både lyssættes, kaster selvskygge, og så punkterne opfattes klarere rumligt ved at de gives en perspektivisk effekt. Hvert punkt udtegnes som en 'kugle' der formindskes/forstørres perspektivisk.

Herudover er det nødvendigt at oprette sig som bruger hos download.dataforsyningen.dk for at kunne hente punktsky-data for større landskaber. For de såkaldte 'frie data', som punktskyerne er en del af, er der ingen abonnementsomkostninger.

² <https://geoslam.com/what-is-slam/> - senest tilgået 30/5-2022

³ Nyeste udgave af Kinect – Kinect Azure, har en maksimal rækkevidde på ca. 5,5 m - <https://docs.microsoft.com/en-us/azure/Kinect-dk/hardware-specification#depth-camera-supported-operating-modes> – tilgået 25-05-2022

⁴ <https://blogs.nvidia.com/blog/2012/09/10/what-is-cuda-2/> - tilgået 3/5-2022.

⁵ <https://dataforsyningen.dk/> - tilgået 3/5-2022

Introduktion til 3 skalatrin med 2 tilgange – case DK

For at forklare bredden i anvendelsen, vælger jeg hér at dele analysesubjekterne op i tre undergrupper - skalatrin, der hver især korresponderer med forskellige tilgange og potentialer.

1. En storskala-landskabsskala der primært beskæftiger sig med omgivelser som set oven- og udefra. Her er fokus på større beplantningspartier, topografi og forholdet bebyggelse-landskab. Skalaen på punktskyerne er her i størrelsesordenen nogle få km².
2. En mindre skala der behandler skovpartier, indre rum i beplantning (f.eks. skovrum), og andre serier af landskabsrum – typisk omkranset af beplantning, men kan også være beplantning som set ovenfra. Skalen på punktskyerne er her i størrelsesordenen nogle få hektar.
3. Forholdsvist veldefinerede uderum – typisk i bymæssig kontekst og egentlige interiørrum. Omgivelserne af præget af forholdsvist veldefinerede flader, hvor beplantning er sparsom eller ensidig i forhold til en modstående bebyggelse. Størrelsesordenen er hér typisk 100-1.000 m².

Til den store skala, har jeg valgt at benytte SDFE's DHM-punktsky, idet SDFE siden 2007 har publiceret et landsdækkende sæt med punktskyer. Punktskymodellen danner bl.a. grundlag for det landsdækkende sæt af højdekurver og de danske overflade- og terrænmodeller, men kan altså også hentes som rå data. Punktskyerne hentes via dataforsyningen.dk i pakker á 10 x 10 km. Inden for hver pakke er der 100 filer i .laz-format – der hver dækker 1 x 1 km. Hvert punkt er placeret i det relevante koordinatsystem⁶ og rummer RGB-data således at det er let at orientere sig og visualisere med udgangspunkter i en given konteksts naturlige farver. Punkterne rummer herudover også en individuel klassifikation, så man kan farvelægge med udgangspunkter i disse klasser, eller tænde/slukke de enkelte klasser som lag.⁷

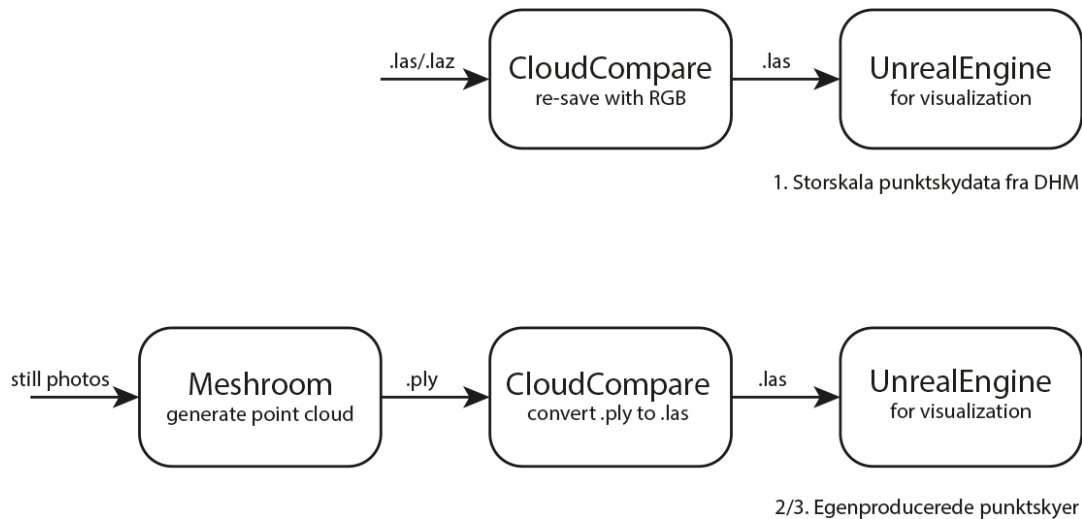
Til at illustrere de to mindre skalatrin, har jeg valgt en metode med egenproducerede punktskyer. Jeg har lavet forsøg med punktskyer på baggrund af droneoverflyvning med video (som eksempel på niveau 2-kontekst) og interiør- og haverum (som eksempler på niveau 3-kontekster) hvor jeg har brugt kombinationer af video-stills og almindelige fotos. I forbindelse med genereringen af punktskyen, får det en RGB-farvekode som DHM-punktskyen, men ikke en type-klassificering som den der findes i DHM-datasættet.

⁶ Hvilket i dette tilfælde er UTM32-ETRS89 (EPSG:25832)

⁷ SDFE (2020), s. 13ff.

Forberedelse af data

Alt efter hvilken af de tre tilgange der benyttes, er der forskellige dataflows. Formålet er, i sidste ende, at generere en visuel repræsentation der kan danne baggrund for en landskabsanalyse. Denne landskabsanalyse vælger jeg hér, skal foregå i spil-udviklingsværktøjet 'Unreal Engine', så i dette afsnit, skal der redegøres for, hvordan vi når hen til 'Unreal Engine'. For overskuelighedens skyld – se figur 1 med de to forskellige workflows.



Figur 1 - Forskellige dataflows til forskellige formål

I den store skala, har vi at gøre med data der allerede er genereret På kortforsyningen.dk downloades et 10x10 km udsnit af overflademodellen DHM-DSM som punktsky i LAZ-format. LAZ er en komprimeret udgave af det mere velkendte LAS-format. I forhold til at bruge disse data i Unreal Engine (UE), er der dog det problem, at RGB-værdierne af en ukendt grund går tabt i importen. For at komme dette problem til livs, kan de konverteres vha. programmet Cloud Compare (CC). CC er et databearbejdningsprogram for punktskyer, og ved først at åbne det originale datasæt i CC, for derefter at gemme det i .las version 1.3/1.4 bliver RGB-værdierne lagret på en måde, så de igen bliver synlige i Unreal Engine. I Unreal Engine skal man aktivere point cloud plugin'et. På nuværende tidspunkt er det en beta-version (ver. 0.8) og der er således en del fejl stadigvæk. F.eks. skal man typisk aktivere plugin'et og genstarte Unreal Engine for hver gang man starter på et nyt projekt. Når plugin'et er aktiveret, skal man først tilføje .las-filen til UE's 'content drawer' og derefter tilføje det til arbejdsrummet – f.eks. vha. drag-and-drop.

På de to næste niveauer, opererer jeg i dette eksempel med egenproducerede punktskyer, som bliver genereret med udgangspunkt i video-stills eller billeder – og altså ikke med laseropmåling. Min foreløbige erfaring er, at der skal flere hundrede billeder med stort overlap til, for at kunne generere en sammenhængende punktsky-model. Hvis man blot selv går rundt og manuelt tager billeder, kan det være svært at opnå det nødvendige overlap. Jeg vil derimod anbefale at man konverterer f.eks. 4K video til stills. Til at generere punktskyen,

har jeg benyttet programmet Meshroom, som kan generere både punktskyer og 3D-modeller på baggrund af netop stillbilleder. I udgangspunktet benyttes en kun let modificeret udgave af 'default' indstillingerne Meshroom. Der skal redigeres en 'node' og tilføjes en 'node' for at kunne eksportere en fil i .ply format, som er det vi skal bruge i næste led i vores workflow. Metoden er beskrevet i Alicevision (2021) under afsnittet '*Export Dense Pointcloud from the Meshing node*'. Efter tilføjelse af billeder via drag-and-drop og ændringerne i Meshrooms nodes, køres processen, og vi ender ud med en farvelagt punktsky af vores ønskede kontekst. Den genererede .ply-fil åbnes i CloudCompare. Hvis ikke punktskyen ligger vandret, kan man manuelt sætte et nul-plan ved hjælp af værktøjet 'Pick three points to make a cloud or mesh 'level' i værktøjspanelet der typisk findes i venstre side. Værktøjet findes også i 'tools' -> 'level'. Herefter gemmes i .las 1.3/1.4-format, så kan åbnet i UE på samme måde som med DHM-punktsky-sættet.

Visualisering i UE

Til selve visualiseringen har jeg valgt programmet Unreal Editor. Et program der typisk benyttes til udvikling af computerspil. Grundet til det er, at det har en forholdsvis unik kombination af at være gratis tilgængeligt og tilbyder en visualiserings-motor for punktskyer om end denne pt. er i beta-version. Visualiseringen af punktskyerne i UE tildeles som standard en perspektivisk formindskning, ligesom de rent visuelt renderes som om de var bestående af kugler. Dette har flere fordele. For det første kan man justere punktstørrelsen, hvor aggressivt UE skal søge at udfylde huller mellem punkterne samt sætte en dagslysmode inklusiv skyggesætning. Hvor det umiddelbart kan virke besynderligt at lyssætte en punktsky, er effekten overraskende effektiv – især når punkterne gennem justering af de to førstnævnte variable gives en tilstrækkelig tæthed. Ønsker man derimod af lægge vægt på eksempelvis bebyggelse i kombination med en beplantnings transparens, kan man gøre punkternes radier mindre for at kunne ane bebyggelse igennem beplantning, og tilsvarende fjerne udfyldning af huller.

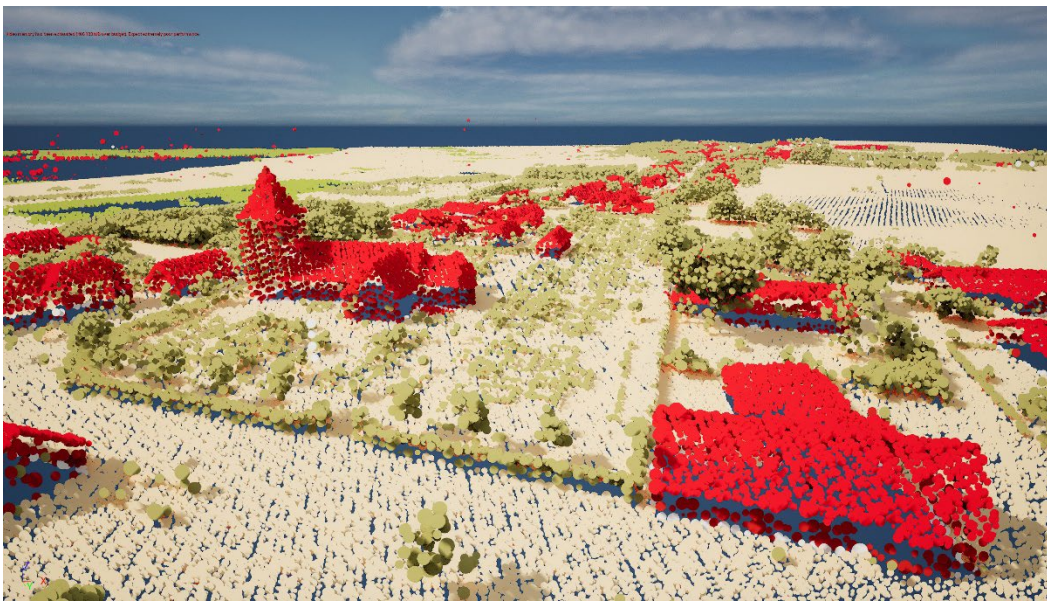
Selve farvesætningen af punkterne følger to spor for DHM-punktskyen – RGB-værdier såvel som klassifikation, mens det for de selvgenererede punktskyer primært er RGB-værdierne der vil være relevante.

Hvis jeg starter med DHM-punktskyen viser figur 2 netop en situation, hvor det er punkternes RGB-værdier der bliver visualiseret, og hvor punktstørrelsen er både forstørret og perspektivisk samt en vis begrænset grad af huludfyldning. Her benyttes punktskyen til at beskue en landsbys placering i et kystnært areal omkranset af marker.



Figur 2 - Visualisering af DHM-punktskyen vha. de indlejrede RGB-værdier, skalering af punkter, dagslysmode og skyggesætning.

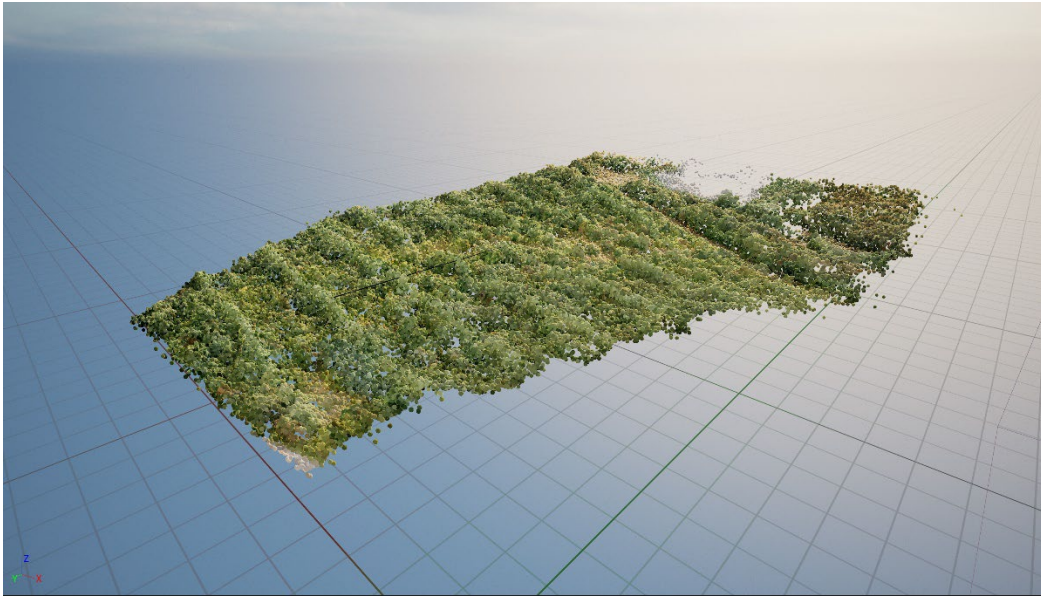
I det næste eksempel – figur 3, vises en situation hvor den DHM-punktskyens indlejrede klassifikation af punkter benyttes til at beskue og accentuere forholdet mellem bebyggelse, beplantning og terræn.



Figur 3 - Visualisering af DHM-punktskyen med bearbejdning af de indlejrede klassificeringer, skalering af punkter, dagslysmode og skyggesætning

Med hensyn til de autogenererede punktskyer, viser jeg her to eksempler. Det første er et resultat på en droneoverflyvning – genereret med kun én lineær overflyvning i én retning. På figur 4 vises resultatet af en ekstrapolering og visualisering af netop disse punktskyer. Selv med kun en enkelt overflyvning kan opnås en model, hvor både beplantningsstruktur og

landskabsrum fremgår. Med en dobbelt overflyvning, lavere overflyvningshøjde og øget opløsningsgrad i Meshroom, vil modellen kunne detaljeres yderligere.



Figur 4 - Visualisering af punktsky af mindre skovparti vha. RGB-værier, skalering af punkter, dagslysmode og skyggesætning. Genereret vha. automatiseret fotogrammetri i Meshroom

Det sidste eksempel jeg hér vil vise, er nede i have-rums skala (figur 5 og 6). Det er en punktsky genereret i en parcelhushave. Punktskyen viser forholdet mellem have og skovparti. Der er ikke et skovbryn, hvilket gør at modellen naturligvis opleves åben, men enkelte træer og træ kroner træder stadig frem på modellen.



Figur 5 - Visualisering af punktsky af haverum vha. RGB-værdier, skalering af punkter, dagslysmode og skyggesætning. Genereret vha. automatiseret fotogrammetri i Meshroom

I dette tilfælde, har det været nødvendigt med forholdsvis mange overlappende billeder, fordi jeg er så tæt på træer der opleves solitære. Det er muligt at sondre klart mellem bebyggelse, klart rumdefinerende beplantningselementer og beplantning der virker svagere rumskabende i haverummet. Man vil kunne forbedre modellen yderligere hvis man benyttede en drone til både høje og lave flyvninger og hvis billederne var taget med en lidt større afstand til skovens træer. Dette besværliggøres dog i dette tilfælde af bebyggelsen.



Figur 6 - Visualisering af punktsky af haverum vha. RGB-værdier, skalering af punkter, dagslysmodel og skyggesætning. Genererer vha. automatiseret fotogrammetri i Meshroom

Konklusion

Punktskyer kan benyttes som værktøj til landskabs- og byanalyse. Jeg har i denne artikel defineret tre forskellige skalatrin, og jeg vurderer at der på nuværende tidspunkt kan planlægges og gennemføres et tilfredsstillende dataflow fra punktsky til visualisering, hvor der kan gives et indtryk af landskabs- og bystrukturer der supplerer luftfotos og meshmodeller. Det er mit indtryk, at punktskyerne er meget anvendeligt til at give et hurtigt indtryk af et landskab, og bør vinde yderligere udbredelse, selvom det kan være krævende ift. computerudstyr. Dertil kommer potentialerne for at benytte punktskyer som kontekst for skitsering i CAD-programmer såsom Blender, Rhino og Revit.

Jeg har ud over de eksisterende DHM-datasæt afprøvet generering af punktskyer med gratis fotogrammetrisoftware med udgangspunkt i almindelige kameraer. Enten mobilkameraer af god kvalitet eller dronevideo. Dette har også været muligt at gennemføre med tilfredsstillende resultat. Dog er selve fotogrammetri-softwaren forholdsvis langsom. Betalingssoftware såsom Zephyr eller RealityCapture kan generere punktskyerne signifikant hurtigere. Især i mit 'haverums-eksperiment' blev det tidskrævende at få genereret en punktsky af tilstrækkelig opløsning.

Af fælles udfordringer for visualisering af punktskyerne, er sidste led i dataflowet – Unreal Editor (UE). UE kan i princippet kan det den skal ift. punktskyerne, nemlig skalering af punkter, huludfyldning, farvelægning på baggrund af såvel RGB som klassificering, og skyggesætning. Dog er punktsky-plugin'et stadig i beta-version og dermed også ustabil i brug. Dette besværliggør import af data, og man risikerer også at data bliver forvanskede, hvis man skifter til andre programmer mens datasættet er åbent. Jeg har forholdsvis stor tiltro til at dette på sigt bliver løst – bl.a. for firmaet bag UE netop har opkøbt udvikleren af et af de bedste betalings protogrammetri-programmer Reality Capture. Der pågår herudover også udvikling af visualisering af punktskyer i 3D CAD programmet Blender der ligeledes er gratis. På sigt er det også min forventning at Blender vil kunne tilbyde samme visualiseringsmuligheder som UE.

Som en kort afslutning, vil jeg anbefale alle interesserede i rumlige analyser i den store-skala at gøre sig bekendt med mulighederne for at benytte punktskyer. Der findes store eksisterende punktsky-datasæt på den store skala, det er forholdsvis enkelt at generere egne datasæt og der findes nok gratis software til at fylde hele dataflowet ud frem mod visualisering. Softwaren er måske stadig under udvikling, men det er mit indtryk at modningen går stærk.

Referencer

Dokumenter:

SDFE (2020). *DHM product specification v1.0.0*. København: SDFE.

https://dataforsyningen.dk/asset/PDF/produkt_dokumentation/dhm-prodspec-v1.0.0.pdf.

Webside senest tilgået 3/5-2022

Websider:

Alicevision (2021). *Export Pointcloud*. Website:

<https://github.com/alicevision/meshroom/wiki/Export-Pointcloud> - senest tilgået 16/5.2022

SDFE (2022). Dataforsyningen. <https://dataforsyningen.dk/> - senest tilgået 3/5-2022

Anvendt software:

Meshroom 2021.1.0

<https://alicevision.org/#meshroom> - senest tilgået 12/5-2022

Unreal Engine 5.0.2

<https://www.unrealengine.com/en-US/download> - senest tilgået 30/5-2022

Cloud Compare 2.12.0

<https://www.cloudcompare.org/> - senest tilgået 3/5-2022