

NATIONAAL NOKLIJNENBESTAND

Theorie, Uitvoering & Resultaten



Keuris' Earth Observation

L. Keuris

29 november 2024

Versie 1.0

Colofon

ORGANISATIE

Keuris Earth Observation

DATUM

29 november 2024

VERSIE

Versie 1.0

PROJECTNAAM

Nationaal Noklijnenbestand

AUTEUR

Lars Keuris

E-MAIL

keuris.earth.observation@proton.me

WEBSITE

earth.observation.keuris.nl

BESTANDSNAAM

KEO_HWH_Nationaal_Noklijnenbestand_v10.pdf

© Keuris Earth Observation, Nederland

Afbeelding voorblad: *Google Maps* (<https://www.google.nl/maps>, bewerkt)

Inhoudsopgave

1. Introductie	5
2. Theorie	6
2.1. Conventies	6
2.2. Definitie noklijn	6
2.3. Dataset	6
2.4. Lessen uit eerdere pogingen	6
2.5. Methode: DakGemak	7
2.5.1. Directe schatting	7
2.5.2. Indirecte schatting	8
2.6. Validatie	8
3. Uitvoering	10
3.1. Rekenstappen	10
3.1.1. Directe schatting	10
3.1.2. Indirecte schatting	12
3.2. Implementatie	13
3.3. Parameters	14
3.4. Voorbeeldberekeningen	14
3.5. Uitvoer	15
3.5.1. Directe schatting	15
3.5.2. Indirecte schatting	15
3.5.3. Nationaal noklijnenbestand	15
4. Resultaten	16
4.1. Verwerkingstijd	16
4.2. Aantallen	16
4.3. Validatie	16
4.4. Vergelijking datasets	17
4.5. Voorbeelden	20
5. Discussie	24
5.1. Beperkingen	24
5.2. Rastergebruik	24
5.3. Tegelranden	24
5.4. Validatie	25
5.5. Symmetriemethode	25
5.6. Nokpunt	25
5.7. Race condities	26
6. Conclusie	27
7. Bijlagen	28
7.1. Installatie	28
7.2. Codevoorbeeld	29

7.3.	Verschilberekening	30
7.4.	Attributentabel uitvoer	31
7.4.1.	Noklijnen	31
7.4.1.1.	Opmerkingen bij kolommen	31
7.4.2.	Dakvlakken	32
7.4.2.1.	Opmerkingen bij kolommen	32
7.4.3.	Vlakke Daken	32
7.4.3.1.	Opmerkingen bij kolommen	33
7.5.	Validatie histogrammen	34
7.6.	Uitleg figuren	36
7.7.	Controlegetallen	37

1. Introductie

Dit document beschrijft de bijdrage aan de ontwikkeling en implementatie van software voor Het Waterschapshuis om daken en noklijnen uit puntenwolken te schatten. Ten tweede worden de resultaten van die software, namelijk een nationaal noklijnenbestand, uitgelicht en vergeleken met referentiedata. Het voorgaande is gedurende een periode van ongeveer twee maanden (10-09-2024 t/m 20-11-2024) opgezet door Keuris Earth Observation in opdracht van Het Waterschapshuis.

Dit project is onderdeel van de uitwerking van de Integrale Hoogtevoorziening Nederland (IHN) en volgt in grote lijnen de adviezen van:

- Het Waterschapshuis met *Over de keuze voor een algoritme* (gepubliceerd op 5 oktober 2023), geschreven door Daan Hulskemper
- Het Waterschapshuis met *Inventarisatie van AHN puntenwolken en zadeldakvergelijkingen* (gepubliceerd op 27 november 2023), geschreven door Daan Hulskemper
- 3DGI met *Noklijnen in het IHN en synergieën met de 3DBAG* (gepubliceerd op 29 augustus 2024), geschreven door Ravi Peters
- De Universiteit van Twente met *Quality of Ridgelines: Ridgeline Extraction from Gable Roofs in Airborne Point Clouds* (gepubliceerd op 1 september 2024), geschreven door Sander Oude Elberink en George Vosselman

Daarnaast zijn er uitwerkingen van vlak- en noklijnextractiemethoden gemaakt in 2023 door Daan Hulskemper, Jan-Willem van Zwieten en Adriaan van Natijne vanuit Het Waterschapshuis. Deze uitwerkingen staan aan de basis van de softwareontwikkeling voor de productie van het nationale noklijnenbestand binnen dit project.

Het nationale noklijnenbestand is een collectie van digitale vectorbestanden met noklijnbeschrijvingen die zijn afgeleid uit puntenwolldata. Het Waterschapshuis beschikt over puntenwolldata uit twee bronnen, namelijk 1) het Actueel Hoogtebestand Nederland¹ (AHN) en 2) Beeldmateriaal² (BM). Het Waterschapshuis heeft de benodigde reken capaciteit en data beschikbaar gesteld voor de uitvoering van de ontwikkelde software (DakGemak), de begeleiding en ondersteuning hierbij werd verzorgd door Adriaan van Natijne.

Dit document is opgedeeld in vier hoofddelen, namelijk 1) Theorie, 2) Uitvoering, 3) Resultaten en 4) Discussie. Als laatste volgen de Conclusie en de Bijlagen.

¹ AHN (geraadpleegd op 18 november 2024): <https://www.ahn.nl/>

² Beeldmateriaal (geraadpleegd op 18 november 2024): <https://www.beeldmateriaal.nl/>

2. Theorie

2.1. Conventies

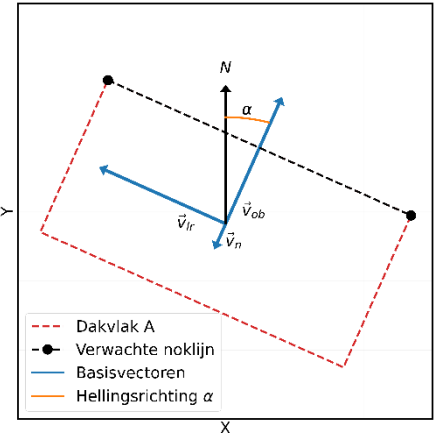
In de beschrijvingen in dit rapport worden afstanden altijd in meters en hoeken altijd in graden gebruikt, tenzij anders vermeld. Daarnaast is het coördinaatsysteem EPSG:7415 aangenomen, dat Rijksdriehoekskoördinaten (RD) als horizontale assen (X,Y) en Normaal Amsterdams Peil (NAP) als verticale as (Z) gebruikt.

2.2. Definitie noklijn

De noklijn in het noklijnenbestand is een **virtuele** 3D lijn die gelijk is aan de snijlijn van twee eindige dakvlakken. De twee vlakken hellen elk met minimaal 20 tot maximaal 70 graden t.o.v. het horizontale (X, Y) vlak. Vlakken met een andere helling zijn uitgesloten. Geldige vlakken hellen naar elkaar toe **omhoog** waarbij de hellingsrichting **exact** 180 graden verschilt. Als gevolg daarvan is een noklijn **altijd horizontaal** en kan dus worden beschreven door twee eindpunten. Dat kan dan met twee X- en Y-coördinaten en één unieke Z-coördinaat. Deze twee eindpunten worden afgeleid van de uiteinden van de twee eindige vlakken.

Een noklijn wordt berekend uit twee dakvlakken. Een gebouw kan meerdere dakvlakken en dus ook meerdere noklijnen hebben. Een dakvlak kan daarbij ook bijdragen aan meerdere noklijnen. Voor elk gebouw uit de Basisregistratie Adressen en Gebouwen (BAG) wordt de meest representatieve noklijn afgeleid. Een BAG-registratienummer heeft dus altijd maar **één noklijn**.

Uit het voorgaande volgt ook dat dakvlakken worden direct afgeleid en noklijnen indirect.



Figuur 1 Bovenaanzicht van een dakvlak. De hellingsrichting van een dakvlak wordt aangegeven door hoek α tussen de positieve Y-as en $\vec{v}_{ob}$. De positieve Y-as is bij benadering gelijk aan het noorden. De hoek tussen $\vec{v}_{ob}$ en het horizontale vlak is de hellingshoek (zie Figuur 7). $\vec{v}_{ob}$ en $\vec{v}_{lr}$ liggen allebei in het dakvlak. $\vec{v}_n$ staat loodrecht op het dakvlak.

2.3. Dataset

Er zijn op het moment van schrijven puntenwolken van 3 AHN datasets (AHN2, AHN3, AHN4) en 3 BM datasets (BM18, BM19, BM20). AHN4 is de meest recente AHN dataset en heeft samen met AHN3 de beste combinatie punt dichtheid en nauwkeurigheid.

2.4. Lessen uit eerdere pogingen

Aan de basis van de noklijnextractiemethoden ligt het schatten van één of twee vlakken tegelijk. Relevant onderscheid kan tussen de verschillende vlakschattingsmethoden worden gemaakt met:

- Deterministisch of niet-deterministisch. Zo is *Random Sample Consensus* (RANSAC) niet-deterministisch en een kleinste-kwadratenvereffening (LSE) deterministisch. Met andere woorden; RANSAC vindt het best passende vlak met een vooraf ingestelde waarschijnlijkheid. Hoe hoger de vereiste waarschijnlijkheid, hoe intensiever de algehele berekening is. Een deterministische methode vindt daarentegen altijd hetzelfde vlak.
- Directe of iteratieve schatters. Iteratieve schatters, zoals een *Region Growing* methode, beginnen met één of meerdere (willekeurige) startpunten in een puntenwolk en maken een vlakschatting met nabijgelegen punten. Mocht deze volgens vooraf ingestelde criteria passend zijn, dan worden er stapsgewijs meer passende punten betrokken totdat het vlak groot genoeg is en/of geen passende punten meer te vinden zijn. Vanwege de iteraties wordt deze methode snel intensief. Directe schatters maken een vlakschatting zonder iteraties op een vooraf aangewezen selectie. Deze selectie moet dan op een andere manier worden afgeleid.

Het Waterschapshuis heeft ervaring opgedaan met 3 methodes voor het schatten van noklijnen, namelijk LSE, RANSAC en Region Growing gebundeld in een Python module: PyDakje³. Het werd daaruit duidelijk dat Region Growing 1) de ruis onderdrukt door het gebruik van groepen punten ineen, wat degelijke vlakschattingen bevordert en dat 2) coherente daken een gevolg zijn van het zoeken naar nabijgelegen punten. Met andere woorden; de Region Growing methode beantwoordt impliciet de vraag: Welke punten horen samen bij eenzelfde dakvlak? Een reken technisch nadeel is echter dat er significant veel meer vlakschattingen nodig zijn dan de andere twee methodes. Dat

³ GitLab (beperkte toegang): <https://gitlab.com/hetwaterschapshuis/ahn-en-beeldmateriaal/ihtn>

is nadelig omdat nabijgelegen punten zoeken en de betrokken vlakschattingen relatief veel tijd kosten. De andere twee methodes presteren slechter omdat het weinig rekening houdt met welke punten tot eenzelfde dakvlak behoren. In dat opzicht is er niets mis met de vlakschatting zelf, maar juist met de punten die geselecteerd zijn.

Daaruit volgt dat de puntselectie vóór de vlakschatting minstens zo belangrijk is als de vlakschatting zelf. Bij het uitwerken van een verbeterde versie wordt daarom rekening gehouden met de volgende vraag:

Kunnen we impliciet of expliciet op een efficiënte manier punten bij elkaar zoeken die tot eenzelfde dakvlak behoren? De vlakschatting hoeft dan in het beste geval maar één keer te worden gedaan. Daarnaast willen we dakvlakken zo goed mogelijk samenvoegen. De invloed van onderbrekingen op het veronderstelde dakvlak door bijvoorbeeld dakkapellen en zonnepanelen moet hierbij zo klein mogelijk worden gehouden. Hierbij is exacte herhaalbaarheid wenselijk (deterministische methode).

2.5. Methode: DakGemak

In dit deel wordt een aangepaste methode geïntroduceerd, genaamd DakGemak. Hierbij horen idealiter ook de twee sterke punten van Region Growing, namelijk:

1. Benader meerdere punten tegelijk om ruis te onderdrukken bij schattingen.
2. Zoek in de nabije omgeving voor soortgelijke punten om een zo goed mogelijke representatie van een dak te vinden.

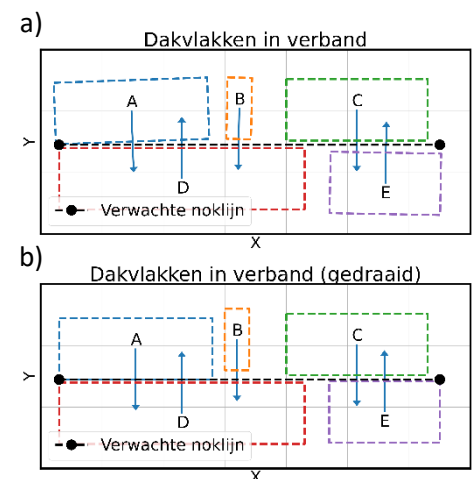
In DakGemak wordt hieraan gehoor gegeven door het gebruik van rasters. Door de puntenwolk te verdelen over cellen gaat detail verloren, maar verliest de ruis zijn aandeel. Ten tweede is het reken technisch efficiënter om in een raster te zoeken naar andere cellen die mogelijk bij eenzelfde dakvlak horen dan om dat op puntenniveau te doen. Dat betekent dat er sneller en makkelijker verder gezocht kan worden. De implementatie is echter complexer dan Region Growing door een extra stap; er wordt namelijk eerst gewerkt op rasterniveau en daarna op puntniveau. Verder is DakGemak opgedeeld in twee programma's, namelijk één voor een directe schatting (`dakgemak_direct.py`) en één voor een indirecte schatting (`dakgemak_indirect.py`). Bij de directe schatting wordt er geen informatie over de locatie van dakvlakken en noklijnen meegegeven, bij de indirecte schatting wel. Dat betekent dat de indirecte schatting simpeler en sneller is, omdat de zoektocht naar de punten voor een geschikt paar dakvlakken grotendeels wordt overgeslagen.

2.5.1. Directe schatting

Het doel is om de dakvlakken én noklijnen in een puntenwolk te detecteren, isoleren en op te slaan. De directe schatting werkt het beste op een dataset met zoveel (voor een goede representatie) en nauwkeurig mogelijke punten. Hierom kiezen we AHN4. De BAG wordt niet gebruikt bij de voorselectie van punten. Alleen punten met een classificatiewaarde gelijk aan 6 (gebouw) worden gebruikt.

De AHN4 dataset wordt in een raster opgedeeld met cellen van 0.5 bij 0.5 meter in X- en Y-richting. Daaruit wordt 1) het aantal 2) de mediaan en 3) de standaardafwijking van de Z-waardes van elke cel berekend. Met de mediaan van elke cel en diens naburige cellen wordt de helling en de hellingsrichting berekend. Vervolgens worden alle bij elkaar gelegen cellen met soortgelijke helling en hellingsrichting gegroepeerd tot een vlak. Als gevolg daarvan worden automatisch alle sterk variërende of te kleine vlakken uitgesloten.

Elke vlak bestaat uit cellen die tot eenzelfde dakvlak behoren. Voor elk vlak wordt een orthonormale basis geschat met de punten uit de puntenwolk die in deze cellen liggen. De geschatte orthonormale basis staat loodrecht op de veronderstelde dakranden. Voor alle vlakken wordt gezocht naar een ander vlak dat een tegenovergestelde hellingsrichting heeft. Dit gebeurt iteratief zodat vlakken met tegenovergestelde hellingsrichting die in verband liggen ook worden meegenomen. Als resultaat zijn er voor elk dak twee sets van één of meerdere vlakken, één voor elke zijde van het dak (zie het dak in Figuur 2a met vlak A, B, C en vlak D, E).



Figuur 2 Boven- en onderaanzicht van 1 dakgroep met 2 sets vlakken (A, B en C tegenover D en E) van een gebouw, waarbij de dakvlakken in verband liggen. De vectoren geven $\vec{v}_{ob}$ en dus de hellingsrichting weer. In het onderste subfiguur zijn de dakvlakken gedraaid zodat de hellingsrichting 180 graden verschilt aan beide kanten van de veronderstelde noklijn.

Het is mogelijk dat de vlakken aan een zijde enkel parallel en niet co-planair zijn. Niet-co-planaire vlakken binnen een set worden opgesplitst en vormen weer een nieuwe set. Naderhand zijn de vlakken aan elke zijde van het dak co-planair en vormen daarmee vormen de twee sets een unieke noklijn: een dakgroep. Een vlak kan in meerdere sets (dakgroepen) voorkomen en daardoor meerdere noklijnen beschrijven.

Voor elke dakgroep worden voor de twee sets vlakken elk één vlakschatting gemaakt met alle punten uit de puntenwolk binnen een set. Mochten de twee daken die nu een noklijn beschrijven aan de hand van hun snijlijn niet exact een tegenovergestelde hellingsrichting hebben, dan worden beide daken tot zo gedraaid dat de hellingsrichting aan beide zijden met 180 graden verschilt (zie Figuur 2). De snijlijn (noklijn) tussen de twee vlakken is nu horizontaal. Voor elke noklijn wordt vervolgens een passend BAG pand gezocht. Als er meerdere noklijnen binnen het bereik van een BAG pand liggen, dan wordt de noklijn met de beste relatieve bezetting behouden.

2.5.2. Indirecte schatting

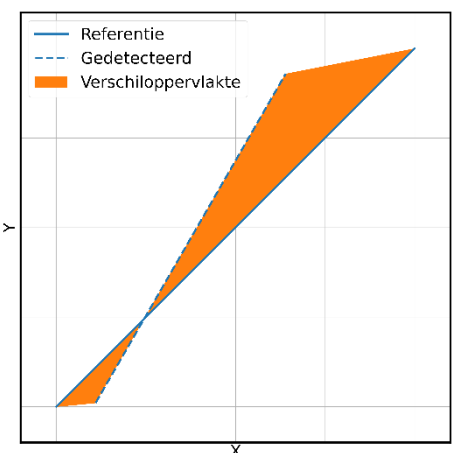
Het doel is om de noklijnen in een puntenwolk te detecteren, isoleren en op te slaan waarbij de 3D locatie van de bijbehorende dakvlakken al bekend is. De indirecte schatting werkt het beste op een dataset met zoveel (voor een goede representatie) en nauwkeurig mogelijke punten. Dankzij het feit dat de locatie van de dakvlakken al bekend is werkt deze methode beter op datasets met een lagere punt dichtheid. De BAG wordt niet gebruikt bij de voorselectie van punten. Er is geen puntselectie op basis van classificatie.

De twee 3D dakvlakken die horen bij elke noklijn worden gebruikt om direct punten uit de puntenwolk te selecteren. Punten die te ver van een van beide dakvlakken af liggen worden uitgesloten, waarna apart twee nieuwe dakvlakken worden geschat met de overgebleven punten. Mochten de twee daken die nu een noklijn beschrijven aan de hand van hun snijlijn niet exact een tegenovergestelde hellingsrichting hebben, dan worden beide daken tot zo gedraaid dat de hellingsrichting aan beide zijden met 180 graden verschilt (zie Figuur 2). De snijlijn (noklijn) tussen de twee vlakken is nu horizontaal. Voor elke noklijn wordt vervolgens een passend BAG pand gezocht. Mochten er meerdere noklijnen op eenzelfde BAG pand liggen, dan wordt de noklijn met de beste relatieve bezetting behouden.

2.6. Validatie

Geodelta, 3DGI en TerraSolid hebben elk op hun eigen manier de noklijnen uit enkele 5x6,25 km² tegels (alle AHN en BM puntenwolken) berekend⁴. De tegels 06CZ2, 09BN2, 25GN1, 37FN2, 46AZ2 voor Geodelta en 3DGI en 06CZ2, 09BN2 voor TerraSolid. De noklijnen die berekend zijn met DakGemak worden vergeleken met de noklijnen van Geodelta, 3DGI en TerraSolid waar deze beschikbaar zijn.

Voor elk BAG pand waar er een referentie noklijn en een noklijn uit DakGemak beschikbaar is wordt een verschil berekend. Het verschil tussen twee noklijnen wordt bepaald door het gebied **tussen** de referentienoklijn en de noklijn van DakGemak (zie Figuur 16). De oppervlakte van het gebied wordt vervolgens gedeeld door de gemiddelde lengte van de twee noklijnen, wat de gemiddelde afstand tussen twee noklijnen weergeeft. Figuur 3 laat dit alleen zien voor het verschil in het horizontale vlak (X/Y). Hetzelfde kan worden gedaan in het verticale vlak (XY/Z). Het totaalverschil wordt berekend door het verschil in het horizontale vlak en het verschil in het verticale vlak kwadratisch op te tellen (en vervolgens wortel te trekken). De verschillen zijn altijd positief door het gebruik van oppervlaktes. De mediaan, de mediaan van absolute afwijkingen (MAD), het gemiddelde en de standaardafwijking van deze gemiddelde afstanden worden gebruikt om verschillen in de positie van de noklijn te kwantificeren.



Figuur 3 Het verschil tussen twee noklijnen wordt bepaald door de oppervlakte dat zich tussen de twee noklijnen bevindt.

Co-lineaire noklijnen hebben een verschil van nul, oriëntatieverschillen tussen noklijnen hebben een altijd verschil groter dan nul en parallelle noklijnen hebben een verschil gelijk aan de (altijd gelijke) afstand tussen de twee lijnen. Daarnaast is het verschil symmetrisch; het verschil tussen noklijn A en noklijn B is hetzelfde als het verschil tussen noklijn B en noklijn A.

Doordat gebouwen meerdere noklijnen hebben en er geen afspraken zijn over welke noklijn het beste is, is een vergelijking van **alle** noklijnen niet eenduidig. Dit is geïllustreerd in Figuur 4.

⁴ Deze referentiedataset is in beheer bij het Waterschapshuis.



Figuur 4 Het BAG-registratienummer is weergegeven als label met witte achtergrond in het midden van de bijbehorende groene veelhoek. De gestippelde lijnen zijn de noklijnen met BAG- registratienummer. Er zijn 3 BAG panden in dit gebied, namelijk 1) de silo met een kegelvormig dak, 2) de 2 aaneengesloten stalgebouwen en 3) een stalgebouw. Links: DakGemak met één noklijn per BAG pand. Midden: Noklijnen van TerraSolid. Rechts: Noklijnen van 3DGI. Het product van Geodelta had in dit gebied geen noklijnen.

Vergelijkingen tussen noklijnen van verschillende producenten waarbij de positie verder dan 0,25 meter verschilt worden gezien als een andere noklijn en worden daarmee uitgesloten. Dit is geen perfecte oplossing, maar toereikend voor de controle in dit document. De resultaten hiervan zijn te vinden in de Validatie.

Een codevoorbeeld voor de verschilberekening is gegeven in 7.3 in de Bijlagen.

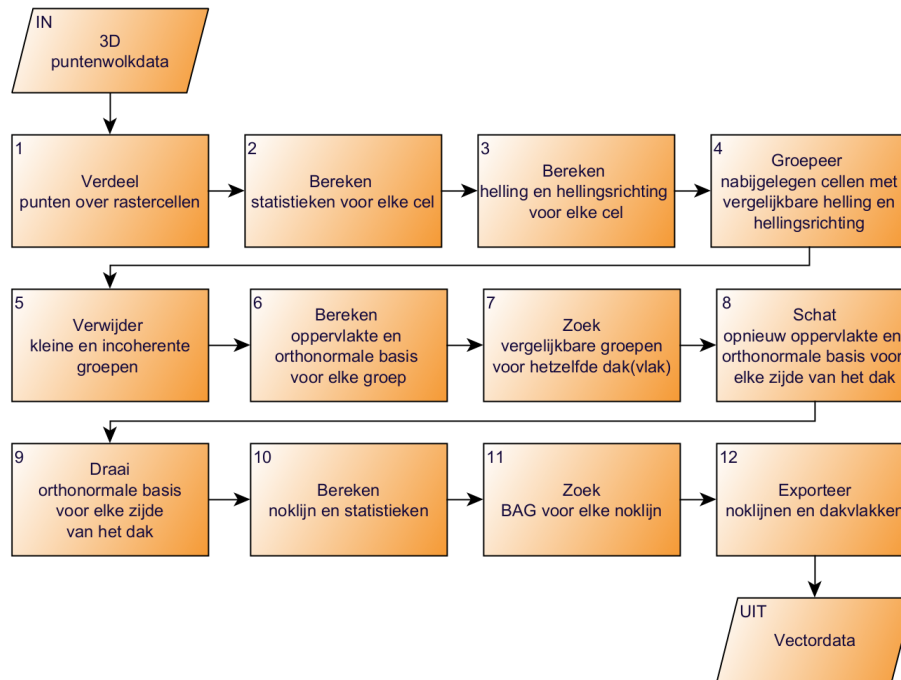
3. Uitvoering

In de uitvoering zijn details belangrijk. Deze sectie gaat daarom dieper in op de verwerkingsstappen met de implementatie in Python in het achterhoofd.

3.1. Rekenstappen

De rekenstappen die plaatsvinden in de directe schattingsmethode (`dakgemak_direct.py`) en indirecte schattingsmethode (`dakgemak_indirect.py`) worden hieronder doorgenomen in de context van de uitvoering en implementatie.

3.1.1. Directe schatting



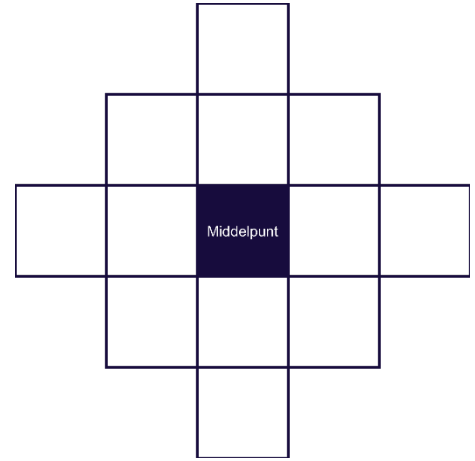
Figuur 5 Stappen in gebruik bij de **directe** schattingsmethode (`dakgemak_direct.py`) in *DakGemak*.

De directe schattingsmethode zoals in tekst beschreven in 2.5.1 in de Theorie is nogmaals beknopt weergegeven in Figuur 5. In de onderstaande tekst worden relevante details van de rekenstappen samen met de softwareparameters (met hoofdletters aangegeven) weergegeven. Deze zijn al ingesteld en hoeven niet te worden aangepast.

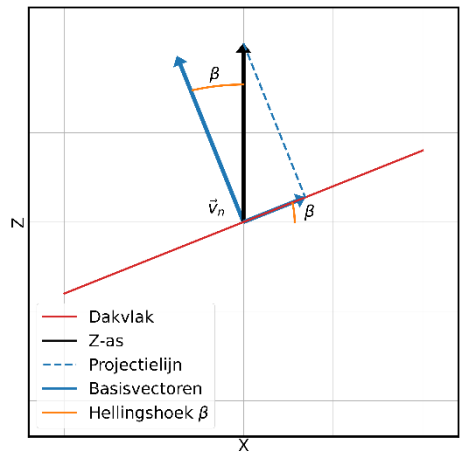
De software werkt met grote (5 km x 6,25 km) en kleine (1 km x 1,25 km) .LAZ bestanden. Alleen puntenwolldata met classificatiewaarde 6 (bebouwing, `BUILDING_CLASS_LAZ`) wordt gebruikt. Het opgegeven .LAZ bestand wordt in delen met een vaste grootte ingelezen (`N_CHUNK`) en moet een minimumaantal punten bevatten (`N_POINTS_LAZ_MIN`).

1. Het rasterbereik wordt automatisch gedefinieerd op basis van de puntenwolldata, maar heeft een aanpasbare celgrootte. Geschikte resultaten werden behaald met een celgrootte van 0,5 m (`BINSIZE_DEFAULT`). De geschikte celgrootte is afhankelijk van de puntdichtheid, oneffenheden op het dak en de grootte van het dak zelf.
2. Het aantal punten en de mediaan en standaardafwijking van de Z-coördinaat wordt berekend voor elke cel. Cellen met te weinig punten (`CELL_N_MIN`) of cellen met een te hoge standaardafwijking (2 keer `AHN_ZSTD`) worden uitgesloten.

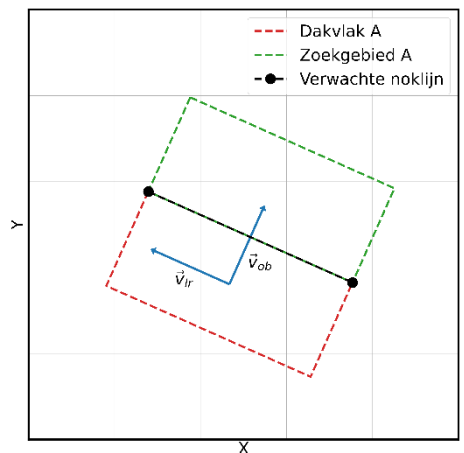
3. Voor elke cel zijn de naaste cellen (onder, boven, links en rechts) nodig om de helling en de hellingsrichting te berekenen. Daarom is het raster altijd iets groter dan het bereik van de puntenwolk (N_BUFFER).
4. Cellen horen bij eenzelfde vlak mits ze 1) bij elkaar in de buurt liggen en niet te veel verschillen in 2) helling ($CLUSTER_SLOPE_DEG_MARGIN$) en 3) hellingsrichting ($CLUSTER_ASPECT_DEG_MARGIN$). Voor alle geldige cellen wordt er in de nabijgelegen cellen gekeken met behulp van het stencil in Figuur 6. De celgrootte en het stencil hangen samen.
5. Kleine groepen ($CLUSTER_N_SMALL$) worden uitgesloten als het merendeel van de cellen ($CLUSTER_F_DISCARD$) in een groep direct naast vlakke daken liggen of de gemiddelde helling niet tussen grenswaarden ligt ($SLOPE_DEG_MIN$, $SLOPE_DEG_MAX$). Groepen die een te grote standaardafwijking voor de helling of de hellingsrichting hebben worden ook uitgesloten ($CLUSTER_STD_SLOPE_MAX$, $CLUSTER_STD_CIRC_ASPECT_MAX$).
6. Voor elke groep worden alle punten uit de puntenwolk die in de geselecteerde cellen liggen bij elkaar genomen en gebruikt voor de schatting van een orthonormale basis. Alle punten die te ver van het vlak afliggen worden uitgesloten ($CLUSTER_NORMAL_N_MAD$). Vervolgens wordt nogmaals een orthonormale basis geschat met de resterende punten. Voor het schatten van de orthonormale basis worden de covariantiematrix en daarna de eigenwaarden -en vectoren berekend. De normaalvector van het vlak wordt positief in de z-as gesteld en de z-as wordt vervolgens geprojecteerd op het vlak dat loodrecht op de normaalvector staat (zie Figuur 7). De laatste as is het kruisproduct tussen de geprojecteerde z-as en de normaalvector. De oppervlakte van het dakvlak wordt berekend door alle punten te projecteren op elk van de twee orthonormale vectoren die in het dakvlak liggen. Hieruit worden de minima en maxima genomen en de twee keer twee combinaties vormen daarmee de hoekpunten van het dakvlak.
7. Groepen die tot hetzelfde dak (met twee dakvlakken) behoren worden gevonden door het dakvlak te verlengen aan de hoogste zijde zoals is weergegeven in Figuur 8. De zoekactie vindt dan plaats tussen twee veelhoeken waarvoor de *Separating Axis Theorem*⁵ (SAT) wordt gebruikt. Een dakvlak wordt gekoppeld aan een ander dakvlak als de hellingshoek tussen beide niet te veel verschilt ($180 \pm CLUSTER_ASPECT_DEG_MARGIN$). Dakvlakken die parallel aan elkaar liggen maar niet co-planair zijn worden gezien als aparte groepen ($CLUSTER_NORMAL_DISTANCE$). Doordat de hellingsrichting niet precies 180 graden verschilt kan het zijn dat dakvlak A in de zoekruimte van dakvlak B ligt, maar niet andersom. Ook kan het voorkomen dat dakvlakken in verband liggen (zie Figuur 2). Hierdoor moeten verbindingen tussen dakvlakken iteratief worden gelegd en is een efficiënte implementatie hiervan complex. Daar staat tegenover dat 1) het impliciet bekend is welke dakvlakken aan elke kant horen, 2) welke dakvlakken bij elk gebouw horen, 3) een dakvlak meerdere noklijnen kan beschrijven en 4) verbanden tussen dakvlakken over relatief grote afstanden kunnen worden gelegd.
8. Op dit punt zijn de dakvlakken opgedeeld sets van twee groepen (met dakvlakken met tegenovergestelde hellingsrichtingen in een aparte groep). Voor alle dakvlakken aan elke zijde worden alle betrokken punten gebruikt voor de schatting van een orthonormale basis en de oppervlakte. Een set wordt uitgesloten als de hellingsrichting tussen de twee dakvlakken te veel verschilt ($180 \pm ROOF_ASPECT_DEG_DIFFERENCE$ graden).



Figuur 6 Stencil voor het zoeken naar gelijksoortige cellen.



Figuur 7 De projectie voor de schatting van de orthonormale basis van een dakvlak.



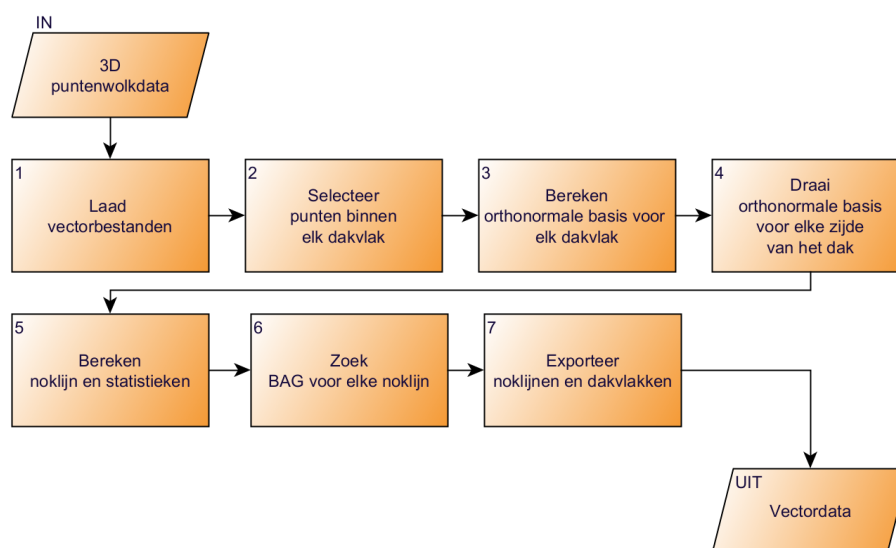
Figuur 8 Zoekruimte voor elk dakvlak. $\vec{v}_{ob}$ is de vector loodrecht op de noklijn binnen het dakvlak. $\vec{v}_{tr}$ is de vector parallel aan de noklijn binnen het dakvlak, loodrecht op $\vec{v}_{ob}$.

⁵ Wikipedia (geraadpleegd op 18 november 2024): https://en.wikipedia.org/wiki/Hyperplane_separation_theorem

9. De hellingshoeken zijn vaak nooit helemaal 180 graden verschillend en dus worden de vlakken zodanig gedraaid dat de hellingshoek exact 180 graden is. De draaiing van elk vlak wordt bepaald door het gewogen gemiddelde met $1/\sigma^2$ als gewicht. Hierbij is σ^2 is de variantie van de normaalafstanden van de punten tot het dakvlak.
10. De noklijn wordt berekend aan de hand van het kruisproduct van de twee normaalvectoren (één van elk dakvlak). De eerste schatting van de middenpositie van de noklijn wordt beschreven door drie vergelijkingen (X, Y en Z) en twee onbekenden (coëfficiënten voor de twee vectoren) vanaf de middelpunten van de twee dakvlakken. Deze wordt daarom opgelost met een kleinste kwadraten vereffening. Alle punten van beide dakvlakken worden vervolgens geprojecteerd op de noklijn, zodat deze langs één lijn komen te liggen. De lengte van de noklijn wordt dan bepaald door de uiteinden van de punten op de noklijn als begin en eindpunt van de noklijn te nemen. Vervolgens wordt het midden van de noklijn opnieuw bepaald, deze is namelijk halverwege de twee uiteinden. Verder wordt het minimum, maximum en de standaardafwijking van de geprojecteerde afstand (normaalafstand) tot het dakvlak voor beide dakvlakken wordt opgeslagen, samen met de gemiddelde puntpositie en andere relevante parameters. Kleine dakvlakken (`ROOF_AREA_2D_MIN`) zijn uitgesloten en dakvlakken met weinig punten (`ROOF_N_SMALL`) worden ook uitgesloten als deze 1) sterk verschillen in puntenaantallen (`ROOF_N_BALANCE`) of 2) de geprojecteerde afstanden tot het vlak een hoge standaardafwijking (`ROOF_STD_MAX`) hebben. De grenswaarde voor een te hoge standaardafwijking wordt gehalveerd voor korte noklijnen (`RIDGE_SHORT`).
11. De noklijnen worden één voor één gekoppeld aan een BAG object door te zoeken naar de BAG objecten die overlappen met de dakvlakken van elke noklijn. Dat betekent dat een BAG object meerdere noklijnen kan hebben en de oplossing daarvoor is niet triviaal. Daarom de volgende procedure bij meerdere noklijnkandidaten voor een BAG object. Alleen de noklijnen worden behouden waarvan de dakvlakken de BAG veelhoek het compleetst overlappen. Deze overlap kan relatief worden uitgedrukt [0-1] en wordt naar beneden afgerond op tienden, waarbij 1,0 ook naar 0,9 wordt afgerond. Bij gelijke relatieve overlap wordt de noklijn gekozen waarbij de maximale standaardafwijking (naar boven afgerond per 0,05 m) van de twee normaalafstanden het laagst is. Bij gelijke maximale standaardafwijking wordt de langste noklijn gekozen.
12. De noklijnen en de vlakken worden geëxporteerd naar vectorbestanden.

Terwijl de bovenstaande procedure wordt gevolgd, worden ook de vlakke daken geschat. De vlakke daken volgen de stappen 1 t/m 6 en 10 t/m 12 (Figuur 5) voor vlakke daken in plaats van noklijnen. De vlakke daken worden beschreven door een veelhoek en een gemiddelde hoogte (Z-as). Deze veelhoek wordt geschat als convexe veelhoek met *Andrew's Monotone Chain algorithm*⁶, terwijl de fysieke dakranden zeker niet altijd een convexe veelhoek beschrijven. Hierdoor is het beschreven vlakke dak altijd gelijk aan of groter dan het fysieke vlakke dak. In stap 11 worden 1) de veelhoeken die de vlakke daken beschrijven afgesneden op de BAG gebieden en 2) de schuine daken verwijderd. Dat betekent dat deze veelhoeken nu wel concaaf kunnen zijn, met mogelijk gaten erin.

3.1.2. Indirecte schatting



Figuur 9 Stappen in gebruik bij de *indirecte* schattingsmethode (`dakgemak_indirect.py`) in DakGemak.

⁶ Wikibooks (geraadpleegd op 18 november 2024): https://en.wikibooks.org/wiki/Algorithm_Implementation/Geometry/Convex_hull/Monotone_chain

De indirecte schattingsmethode zoals in tekst beschreven in 2.5.2 in de Theorie is nogmaals beknopt weergegeven in Figuur 9. In de onderstaande tekst worden relevante details van de rekenstappen samen met de softwareparameters (met hoofdletters aangegeven) weergegeven. Deze zijn al ingesteld en hoeven niet te worden aangepast.

De software werkt met grote (5 km x 6,25 km) en kleine (1 km x 1,25 km) .LAZ bestanden. Alleen puntenwolldata dat in het XY-bereik (*bounding box*) van minstens een van de elementen in de vectorbestanden ligt wordt geladen. De classificatie wordt niet gebruikt. Het opgegeven .LAZ bestand wordt in delen met een vaste grootte ingelezen (`N_CHUNK`) en moet een minimumaantal punten bevatten (`N_POINTS_LAZ_MIN`).

Twee vector bestanden worden ingeladen voor de schatting van nieuwe noklijnen, namelijk 1) `_direct_ridges.gpkg` en 2) `_direct_ridge_roofs.gpkg` (andere extensies mogelijk) (zie Uitvoer).

1. Alle punten binnen het X/Y bereik (*bounding box*) van elk paar dakvlakken (één aan elke zijde van de noklijn) wordt geselecteerd en geprojecteerd op de twee hoofdassen van het dakvlak. Alle punten die zich niet binnen de twee dakvlakken vinden kunnen zo worden uitgesloten in het horizontale bereik. Alle punten die te ver van het dakvlak afliggen (normaalafstand) worden ook uitgesloten (`ROOF_NORMAL_D_MAX`, `CLUSTER_NORMAL_N_MAX` en voor platte daken `SURFACE_Z_N_MAX`, `SURFACE_DZ_MAX`). Een paar dakvlakken wordt in het geheel uitgesloten als een van beide daken te weinig punten heeft (`IND_POINTS_N_MIN`).
2. De orthonormale basis wordt voor elk paar dakvlakken geschat. Voor het schatten van de orthonormale basis worden de covariantiematrix en daarna de eigenwaarden -en vectoren berekend. De normaalvector van het vlak wordt positief in de z-as gesteld en de z-as wordt vervolgens geprojecteerd op het vlak dat loodrecht op de normaalvector staat (zie Figuur 7). De laatste as is het kruisproduct tussen de geprojecteerde z-as en de normaalvector. Een paar dakvlakken wordt in het geheel uitgesloten als een van beide dakvlakken te vlak of te steil is (`SLOPE_DEG_MIN_IND`, `SLOPE_DEG_MAX_IND`). De oppervlakte van het dakvlak wordt berekend door alle punten te projecteren op elk van de twee orthonormale vectoren die in het dakvlak liggen. Hieruit worden de minima en maxima genomen en de twee keer twee combinaties vormen daarmee de hoekpunten van het dakvlak.
3. Zie stap 9 in Directe schatting.
4. Zie stap 10 in Directe schatting.
5. Zie stap 11 in Directe schatting.
6. Zie stap 12 in Directe schatting.

Terwijl de bovenstaande procedure wordt gevolgd, worden ook de vlakke daken geschat. De vlakke daken volgen de stappen 1 t/m 3 en 5 t/m 7 (Figuur 9) voor vlakke daken in plaats van noklijnen. Alleen vlakke daken met een BAG-registratie worden hier gebruikt. De invoer is daarom een `_surfaces_bag.gpkg` bestand (zie Uitvoer) Bij de puntselectie voor vlakke daken wordt het complete X/Y/Z bereik (*bounding box*) gebruikt (inclusief gaten).

3.2. Implementatie

De methode is geïmplementeerd in [Python](#) 3.12.3. Er wordt gebruik gemaakt van een aantal modules, waaronder [Numpy](#), [Numba](#), [Shapely](#), [GeoPandas](#) en [Matplotlib](#). De complete lijst met modules met corresponderende moduleversies is apart bijgevoegd in een `environment.yml` bestand. De voertaal in DakGemak is Engels.

De code is langer dan men misschien gewend is. Dat komt door het gebruik van Numba, waarbij de uitvoering van routines kan worden versneld. Soms gaat dat automatisch en soms is daar extra maatwerk voor nodig. Een voorbeeld is de `compute_binned_statistic_2d()` functie. Deze functie is een versnelde aanpassing van `scipy.stats.binned_statistic_2d()`. Zie 7.2 in de Bijlagen voor meer informatie.

De noklijnen en vlakke daken is berekend op (sub)tegels⁷ om zo heel Nederland in behapbare stukken te kunnen verwerken. Deze zijn naderhand weer samengevoegd met een GDAL/OGR programma (`ogr_merge.py`)⁸. De (noklijn)duplicaten die ontstaan voor BAG panden worden opgelost door de hoogste standaardafwijking van beide daken per noklijn te nemen, die naar boven af te ronden per 0,05 m en daarvan de laagste standaardafwijking te kiezen tussen conflicterende noklijnen. Mochten er meerdere een gelijke afgeronde standaardafwijking hebben, dan

⁷ Fwrite (geraadpleegd op 20 november 2024): <https://weblog.fwrite.org/kaartbladen/>

⁸ GDAL (geraadpleegd op 20 november 2024): <https://gdal.org/en/latest/>

wordt de langste noklijn gekozen. Er vindt **geen reductie plaats** om duplicaten bij de andere bestanden (vlakke daken en dakvlakken) te verwijderen.

Voor de validatie wordt het op zichzelf staande programma `dakgemak_difference.py` gebruikt.

3.3. Parameters

De gebruiker kan de **verwerkingsparameters** die genoemd zijn in Rekenstappen aanpassen in `dakgemak_aux.py`. In Rekenstappen zijn deze verwerkingsparameters in hun context geplaatst en toegelicht. **Gebruikersparameters** zoals invoer, uitvoer, uitvoertype en parameters omtrent rapportering zijn beschikbaar in het programma via de opdrachtprompt (CMD of terminal) door het programma (`dakgemak_direct.py` of `dakgemak_indirect.py`) op te roepen met de hulptoptie (`-h` of `--help`).

```
python dakgemak_direct.py -h
python dakgemak_indirect.py -h
```

3.4. Voorbeeldberekeningen

DakGemak is een Python programma dat via de opdrachtprompt (of terminal) kan worden gebruikt, mits alle benodigdheden zijn geïnstalleerd (zie 7.1 in de Bijlagen). Een gebruiksvoorbeeld van een directe schatting:

```
python dakgemak_direct.py -i C:\path\to\input\AHN4_31DZ2_05.LAZ -d C:\path\to\output\results -
-export_points --export_figures
```

De resultaten worden allemaal geëxporteerd naar de map `C:\path\to\output\results` en zijn verdeeld over de volgende vectorbestanden:

```
AHN4_31DZ2_05_20241111T111111_direct_ridges.gpkg, AHN4_31DZ2_05_20241111T111111_direct_ridges_bag.gpkg,
AHN4_31DZ2_05_20241111T111111_direct_surfaces.gpkg, AHN4_31DZ2_05_20241111T111111_direct_surfaces_bag.gpkg,
AHN4_31DZ2_05_20241111T111111_direct_ridge_roofs.gpkg, AHN4_31DZ2_05_20241111T111111_direct_points_roofs.gpkg,
AHN4_31DZ2_05_20241111T111111_direct_points_surfaces.gpkg
```

De laatste twee vectorbestanden worden alleen geëxporteerd als `--export_points` wordt meegegeven. De volgende figuren worden ook geëxporteerd dankzij `--export_figures` (zie Tabel 8 in Bijlagen voor uitleg):

```
AHN4_31DZ2_05_20241111T111111_direct_median_z.png, AHN4_31DZ2_05_20241111T111111_direct_count.png,
AHN4_31DZ2_05_20241111T111111_direct_slope.png, AHN4_31DZ2_05_20241111T111111_direct_aspect.png,
AHN4_31DZ2_05_20241111T111111_direct_svalid.png, AHN4_31DZ2_05_20241111T111111_direct_fvalid.png,
AHN4_31DZ2_05_20241111T111111_direct_label_scluster.png, AHN4_31DZ2_05_20241111T111111_direct_label_fcluster.png,
AHN4_31DZ2_05_20241111T111111_direct_label_scluster_clean.png, AHN4_31DZ2_05_20241111T111111_direct_label_fcluster_clean.png,
AHN4_31DZ2_05_20241111T111111_direct_splanes.png, AHN4_31DZ2_05_20241111T111111_direct_sdiscarded.png,
AHN4_31DZ2_05_20241111T111111_direct_splanes_final.png,
```

Een gebruiksvoorbeeld van een indirecte schatting:

```
python dakgemak_indirect.py -i C:\path\to\input\AHN4_31DZ2_05.LAZ --gfile_ridges
C:\path\to\input\AHN4_31DZ2_05_20241116T132132_direct_ridges.gpkg --gfile_ridge_roofs
C:\path\to\input\AHN4_31DZ2_05_20241116T132132_direct_ridge_roofs.gpkg -d
C:\path\to\output\results --export_figures
```

De resultaten worden allemaal geëxporteerd naar de map `C:\path\to\output\results` en zijn verdeeld over de volgende vectorbestanden:

```
AHN4_31DZ2_05_20241111T111111_indirect_ridges.gpkg, AHN4_31DZ2_05_20241111T111111_indirect_ridges_bag.gpkg,
AHN4_31DZ2_05_20241111T111111_indirect_surfaces_bag.gpkg
```

Het volgend figuur wordt ook geëxporteerd dankzij `--export_figures` (zie Tabel 9 in Bijlagen voor uitleg):

```
AHN4_31DZ2_05_20241111T111111_direct_median_z.png
```

Voor een vergelijking tussen twee methodes kan het programma `dakgemak_difference.py` als volgt worden gebruikt:

```
python dakgemak_difference.py -a
C:\path\to\input\AHN4_31DZ2_05_20241116T132132_direct_ridges_bag.gpkg -b
C:\path\to\input\AHN4_31DZ2_05_20241116T132132_indirect_ridges_bag.gpkg -d
C:\path\to\output\results
```

3.5. Uitvoer

Er zijn twee programma's, namelijk voor de directe schatting (`dakgemak_direct.py`) en voor de indirecte schatting (`dakgemak_indirect.py`), en voor elk een eigen uitvoer.

3.5.1. Directe schatting

De volgende vectorbestanden worden geëxporteerd aan het einde van het draaien van `dakgemak_direct.py`:

1. `${bestandsnaam}_${yyyymmddTHHMMSS}_direct_ridges.${ext}`
Noklijnen zonder BAG-registratienummer.
2. `${bestandsnaam}_${yyyymmddTHHMMSS}_direct_ridges_bag.${ext}`
Noklijnen met BAG-registatienummer. Een opgewerkte versie van 1. door het opknippen en selecteren van de best passende noklijn per BAG-pand.
3. `${bestandsnaam}_${yyyymmddTHHMMSS}_direct_ridges_planes.${ext}`
Dakvlakken die gelabeld zijn identificatienummers. Elke noklijn in 1. en 2. identificatienummers heeft 2 identificatienummers, namelijk één voor elk dakvlak.
4. `${bestandsnaam}_${yyyymmddTHHMMSS}_direct_surfaces.${ext}`
Vlakke daken zonder BAG-registratie.
5. `${bestandsnaam}_${yyyymmddTHHMMSS}_direct_surfaces_bag.${ext}`
Vlakke daken met BAG-registratie.

, waarbij `${bestandsnaam}` de bestandsnaam van het opgegeven .LAZ bestand is. `${yyyymmddTHHMMSS}` is de UTC-tijd van de machine vlak na de start van het programma (de letters volgen de 1989 C code standaard en laten zich gemakkelijk raden en sorteren)⁹. `${ext}` is de gekozen extensie voor het vector bestand. De opties hiervoor zijn Geopackage (`gpkg`), Shapefile (`shp`), SQLite (`sqlite`) of GeoJSON (`geojson`). Geopackage wordt aangeraden en is ingesteld als standaard. Optioneel kunnen de volgende twee vectorbestanden worden geëxporteerd:

6. `${bestandsnaam}_${yyyymmddTHHMMSS}_direct_points_roofs.${ext}`
3D punten uit de puntenwolk die gebruikt zijn voor de schatting van daken die gebruikt zijn voor een noklijnschatting met als attribuut het gereduceerde identificatienummer van dakvlak.
7. `${bestandsnaam}_${yyyymmddTHHMMSS}_direct_points_surfaces.${ext}`
3D punten uit de puntenwolk die gebruikt zijn voor de schatting van vlakke daken met als attribuut het gereduceerde identificatienummer van het dakvlak.

De attributentabel voor deze bestanden zijn te vinden in 7.4 in de Bijlagen.

3.5.2. Indirecte schatting

De vectorbestanden die worden geëxporteerd aan het einde van het draaien van `dakgemak_indirect.py` zijn gelijk aan bestand 1, 2 en 5 in Directe schatting. Het label `_direct` verandert hierbij wel naar `_indirect`.

3.5.3. Nationaal noklijnenbestand

Een voorbeeld van een van de nationale noklijnenbestanden is als volgt:

1. `AHN04_NL_direct_ridges_bag_v10.gpkg`

en volgt de sorteringen uit de vorige twee secties.

⁹ Python (geraadpleegd op 18 november 2024): <https://docs.python.org/3/library/datetime.html#strptime-and-strftime-format-codes>

4. Resultaten

4.1. Verwerkingstijd

De directie schattingsmethode op heel AHN4 (tegels van 1 km x 1,25 km) nam ongeveer 340 CPU uren (wat ongeveer gelijk is aan twee weken) in beslag. De indirecte schattingsmethode op dezelfde tegelgrootte nam tot ongeveer de helft van dat aantal CPU uren in beslag.

4.2. Aantallen

De detectieaantallen zijn weergegeven in Tabel 1. Het verschil tussen de kolom *Aantal noklijnen* en kolom *Aantal BAG noklijnen* wordt bepaald door 1) de eis dat elk BAG pand maar één noklijn mag hebben (minder noklijnen) en 2) het opknippen van noklijnen (meer noklijnen).

Tabel 1 Het aantal gevonden noklijnen en vlakke daken voor de puntenwolk datasets van heel Nederland met DakGemak en het eerdere PyDakje. Er kunnen meerdere platte daken voor één BAG pand zijn. In tegenstelling tot het aantal noklijnen per BAG pand, dat is er maximaal één.

Dataset	Schattingsmethode	Aantal noklijnen	Aantal BAG noklijnen	Aantal platte daken	Aantal platte BAG daken	Aantal BAG noklijnen (PyDakje)	Aantal platte BAG daken (PyDakje)
AHN4	Direct	5.940.570	5.792.987	6.245.869	7.486.994	4.797.109	5.344.309
AHN4	Indirect	5.649.882	5.637.527	-	7.484.486	4.797.109	5.344.309
AHN3	Indirect	4.198.884	4.565.981	-	6.951.285	4.300.011	
AHN2	Indirect	4.058.280	4.535.791	-	6.540.872	4.227.778	
BM20	Indirect	4.494.842	5.089.608	-	6.996.973	1.166.841	5.422.588
BM19	Indirect	4.345.707	4.911.897	-	6.880.898	1.136.688	5.334.151
BM18	Indirect	4.268.957	4.897.098	-	6.818.886	1.079.611	5.051.542

4.3. Validatie

De absolute verschillen in de positie tussen de noklijnen van DakGemak (**directe schatting**) en de noklijnen van Geodelta, 3DGI en TerraSolid zijn samengevat in Tabel 2. De mediaan en MAD van de verschillen in het horizontale en verticale vlak zijn rond de 1 cm of kleiner. De mediaan van het totaalverschil is ongeveer 1 tot 3 cm. De MAD is in alle gevallen ongeveer even groot of kleiner dan de mediaan. Het gemiddelde verschil is vooral groter dan het mediaanverschil in de verticale richting voor 3DGI en TerraSolid. Dit gaat in beide gevallen gepaard met een hoge standaardafwijking, wat (deels) kan komen door buitenliggers.

Tabel 2 Samengevatte absolute verschillen in meters tussen **direct** geschatte noklijnen van DakGemak en noklijnen van 1) Geodelta, 2) 3DGI en 3) TerraSolid voor AHN4 data. Het aantal vergelijkingen tussen de noklijnen van DakGemak en de referentie dataset is weergegeven onder **Aantal**. De percentages onder het aantal geven weer hoeveel dat aantal was van ten opzichte van alle noklijnen beschikbaar voor de gebruikte tegels. Zo zijn er 24.355 vergelijkingen tussen DakGemak en Geodelta gemaakt. Dit was 64,0% van alle beschikbare DakGemak noklijnen en 97,4% van alle beschikbare Geodelta noklijnen.

Geodelta	Mediaan	MAD	Gemiddelde	Standaardafwijking	Aantal
Horizontaal (X/Y)	0,0195	0,0105	0,0261	0,0251	24355 (64,0% / 97,4%)
Verticaal (Z)	0,0182	0,0095	0,0245	0,0241	
Totaal (X/Y/Z)	0,0314	0,0126	0,0390	0,0312	
3DGI	Mediaan	MAD	Gemiddelde	Standaardafwijking	Aantal
Horizontaal (X/Y)	0,0105	0,0064	0,0216	0,0318	34720 (75,8% / 91,3%)
Verticaal (Z)	0,0111	0,0071	0,0190	0,0243	
Totaal (X/Y/Z)	0,0179	0,0096	0,0314	0,0380	
TerraSolid	Mediaan	MAD	Gemiddelde	Standaardafwijking	Aantal
Horizontaal (X/Y)	0,0060	0,0040	0,0127	0,0215	3753 (9,9% / 88,9%)
Verticaal (Z)	0,0076	0,0044	0,0131	0,0185	
Totaal (X/Y/Z)	0,0112	0,0056	0,0196	0,0274	

De data waaruit Tabel 2 is afgeleid is in histogrammen weergegeven in Figuur 18 in de Bijlagen.

Voor de noklijnen uit de **indirecte schatting** van DakGemak is dezelfde vergelijking mogelijk (zie Tabel 3). De resultaten van de indirecte schatting verschillen niet significant met de resultaten van de directe schatting.

Tabel 3 Samengevatte absolute verschillen in meters tussen indirect geschatte noklijnen van DakGemak en noklijnen van 1) Geodelta, 2) 3DGI, 3) TerraSolid en 4) direct geschatte noklijnen van DakGemak voor AHN4 data. Het aantal vergelijkingen tussen de noklijnen van DakGemak en de referentie dataset is weergegeven onder Aantal. De percentages onder het aantal geven weer hoeveel dat aantal was van ten opzichte van alle noklijnen beschikbaar voor de gebruikte tegels. Zo zijn er 24.286 vergelijkingen tussen DakGemak en Geodelta gemaakt. Dit was 67,4% van alle beschikbare DakGemak noklijnen en 97,2% van alle beschikbare Geodelta noklijnen.

Geodelta	Mediaan	MAD	Gemiddelde	Standaardafwijking	Aantal
Horizontaal (X/Y)	0,0193	0,0104	0,0255	0,0239	24.286 (67,4% / 97,2%)
Verticaal (Z)	0,0184	0,0092	0,0237	0,0213	
Totaal (X/Y/Z)	0,0311	0,0121	0,0379	0,0287	
3DGI	Mediaan	MAD	Gemiddelde	Standaardafwijking	Aantal
Horizontaal (X/Y)	0,0095	0,0056	0,0177	0,0255	33.247 (72,6% / 92,6%)
Verticaal (Z)	0,0097	0,0062	0,0173	0,0234	
Totaal (X/Y/Z)	0,0159	0,0082	0,0269	0,0330	
TerraSolid	Mediaan	MAD	Gemiddelde	Standaardafwijking	Aantal
Horizontaal (X/Y)	0,0052	0,0033	0,0112	0,0205	3.621 (10,1% / 85,8%)
Verticaal (Z)	0,0067	0,0039	0,0119	0,0180	
Totaal (X/Y/Z)	0,0097	0,0049	0,0174	0,0266	
DakGemak (Direct)	Mediaan	MAD	Gemiddelde	Standaardafwijking	Aantal
Horizontaal (X/Y)	0,0026	0,0018	0,0060	0,0115	35.894 (94,4% / 100%)
Verticaal (Z)	0,0028	0,0019	0,0064	0,0137	
Totaal (X/Y/Z)	0,0045	0,0026	0,0096	0,0175	

Om de consistentie tussen de directe en indirecte schatting te kwantificeren zijn de noklijnen van beide schattingen vergeleken voor AHN4. Dit is weergegeven in de laatste drie rijen van Tabel 3. Het verschil is ongelijk aan nul, omdat de punten die zijn gebruikt voor de schatting van de dakvlakken in de directe schatting **niet** direct overeenkomen met de punten die zijn gebruikt voor de schatting van de dakvlakken in de indirecte schatting. Dit komt door een andere selectiemethode langs de Z-as (de selectie in het horizontale vlak is gelijk). Desalniettemin komen deze verschillende selecties tot op grote hoogte (zie Tabel 1 en Tabel 3) overeen.

4.4. Vergelijking datasets

De noklijnen zijn geschat met de indirecte methode voor de verschillende datasets. Daarbij is de vergelijking tussen de indirecte en directe methode nogmaals op nationaal niveau gemaakt. De resultaten hiervan zijn samengevat in Tabel 4. De histogrammen zijn door hun grote aantal 16 maal 3 histogrammen als losse .png bestanden bijgevoegd.

Tabel 4 Samengevatte absolute verschillen in meters tussen de schattingen uit verschillende datasets. Het aantal vergelijkingen tussen de noklijnen tussen twee datasets is weergegeven onder Aantal. De percentages onder het aantal geven weer hoeveel dat aantal was van ten opzichte van alle noklijnen beschikbaar. Zo zijn er 5.636.438 vergelijkingen tussen AHN4 indirect en AHN4 direct gemaakt. Dit was 100% van alle beschikbare noklijnen afgeleid met AHN4 indirect en 97,3% van alle beschikbare afgeleid met AHN4 direct.

AHN4 Indirect AHN4 Direct	Mediaan	MAD	Gemiddelde	Standaardafwijking	Aantal
Horizontaal (X/Y)	0,0027	0,0018	0,0060	0,0060	5.636.438 (100% / 97,3%)
Verticaal (Z)	0,0024	0,0016	0,0052	0,0051	
Totaal (X/Y/Z)	0,0042	0,0023	0,0085	0,0085	
AHN4 Indirect AHN3 Indirect	Mediaan	MAD	Gemiddelde	Standaardafwijking	Aantal
Horizontaal (X/Y)	0,0161	0,0096	0,0204	0,0193	4.558.608 (80,9% / 99,9%)
Verticaal (Z)	0,0099	0,0062	0,0136	0,0149	
Totaal (X/Y/Z)	0,0224	0,0079	0,0273	0,0213	
AHN4 Indirect AHN2 Indirect	Mediaan	MAD	Gemiddelde	Standaardafwijking	Aantal
Horizontaal (X/Y)	0,0161	0,0099	0,0209	0,0202	4.527.598 (80,3% / 99,8%)
Verticaal (Z)	0,0119	0,0074	0,0159	0,0167	
Totaal (X/Y/Z)	0,0241	0,0083	0,0294	0,0227	
AHN4 Indirect BM18 Indirect	Mediaan	MAD	Gemiddelde	Standaardafwijking	Aantal
Horizontaal (X/Y)	0,0150	0,0098	0,0231	0,0254	4.886.666 (86,7% / 99,8%)
Verticaal (Z)	0,0202	0,0127	0,0277	0,0269	

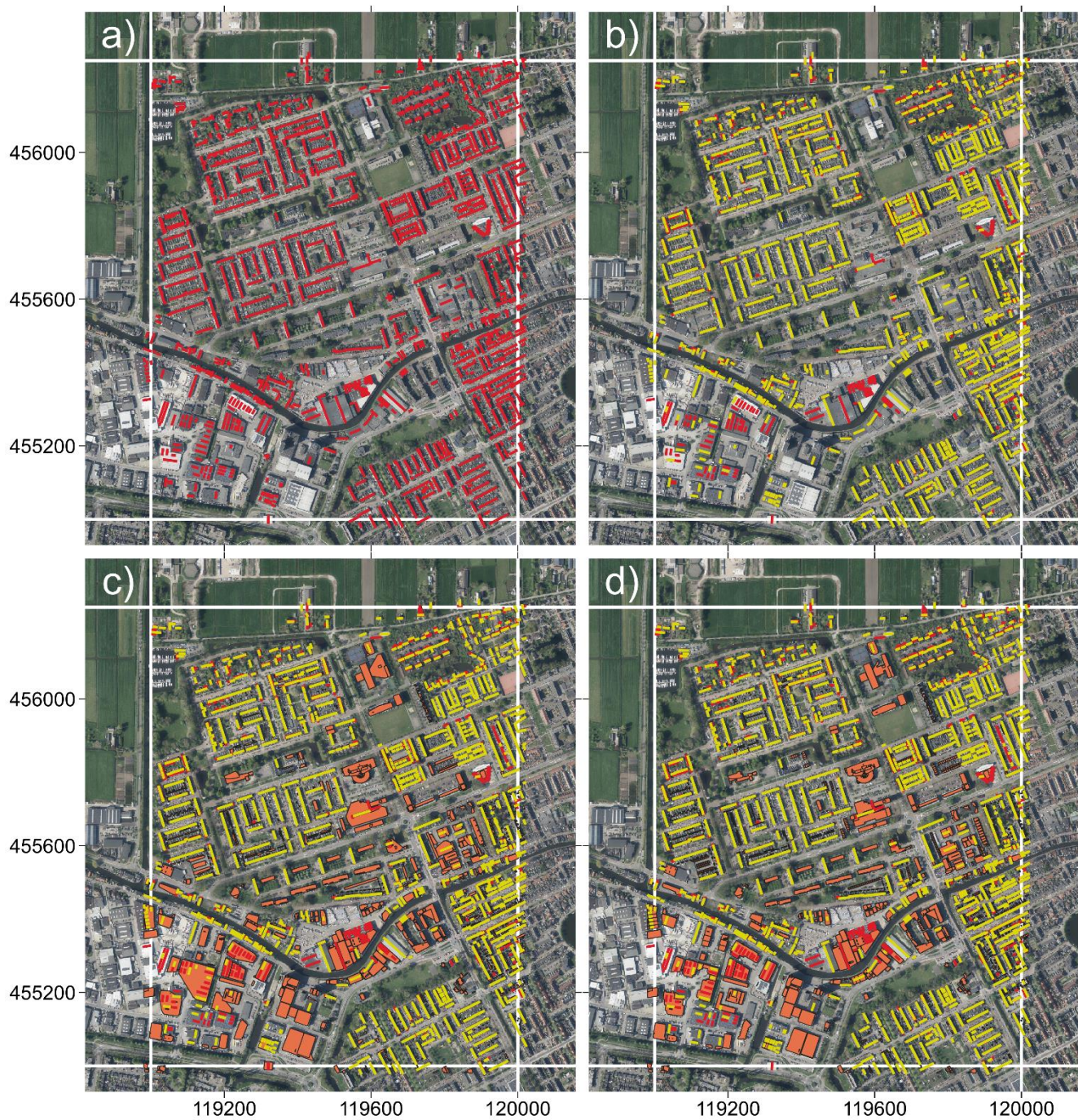
Totaal (X/Y/Z)	0,0299	0,0150	0,0394	0,0335	
AHN4 Indirect BM19 Indirect	Mediaan	MAD	Gemiddelde	Standaardafwijking	Aantal
Horizontaal (X/Y)	0,0150	0,0097	0,0223	0,0239	4.901.364 (86,9% / 99,8%)
Verticaal (Z)	0,0180	0,0111	0,0244	0,0237	
Totaal (X/Y/Z)	0,0280	0,0129	0,0363	0,0301	
AHN4 Indirect BM20 Indirect	Mediaan	MAD	Gemiddelde	Standaardafwijking	Aantal
Horizontaal (X/Y)	0,0159	0,0102	0,0238	0,0255	5.077.664 (90,1% / 99,8%)
Verticaal (Z)	0,0187	0,0119	0,0261	0,0262	
Totaal (X/Y/Z)	0,0295	0,0142	0,0388	0,0329	
AHN3 Indirect AHN2 Indirect	Mediaan	MAD	Gemiddelde	Standaardafwijking	Aantal
Horizontaal (X/Y)	0,0162	0,0101	0,0220	0,0217	3.951.683 (86,6% / 87,1%)
Verticaal (Z)	0,0117	0,0073	0,0160	0,0165	
Totaal (X/Y/Z)	0,0243	0,0111	0,0298	0,0244	
AHN3 Indirect BM18 Indirect	Mediaan	MAD	Gemiddelde	Standaardafwijking	Aantal
Horizontaal (X/Y)	0,0205	0,0126	0,0286	0,0281	4.103.823 (89,9% / 83,8%)
Verticaal (Z)	0,0223	0,0141	0,0305	0,0294	
Totaal (X/Y/Z)	0,0359	0,0166	0,0459	0,0361	
AHN3 Indirect BM19 Indirect	Mediaan	MAD	Gemiddelde	Standaardafwijking	Aantal
Horizontaal (X/Y)	0,0199	0,0122	0,0274	0,0268	4.090.189 (89,6% / 83,3%)
Verticaal (Z)	0,0193	0,0120	0,0264	0,0257	
Totaal (X/Y/Z)	0,0331	0,0150	0,0418	0,0329	
AHN3 Indirect BM20 Indirect	Mediaan	MAD	Gemiddelde	Standaardafwijking	Aantal
Horizontaal (X/Y)	0,0207	0,0127	0,0289	0,0283	4.194.441 (91,9% / 82,4%)
Verticaal (Z)	0,0203	0,0129	0,0285	0,0284	
Totaal (X/Y/Z)	0,0346	0,0160	0,0445	0,0357	
AHN2 Indirect BM18 Indirect	Mediaan	MAD	Gemiddelde	Standaardafwijking	Aantal
Horizontaal (X/Y)	0,0203	0,0125	0,0283	0,0278	4.097.709 (90,3% / 83,7%)
Verticaal (Z)	0,0221	0,0140	0,0306	0,0297	
Totaal (X/Y/Z)	0,0356	0,0164	0,0456	0,0361	
AHN2 Indirect BM19 Indirect	Mediaan	MAD	Gemiddelde	Standaardafwijking	Aantal
Horizontaal (X/Y)	0,0193	0,0118	0,0268	0,0265	4.071.626 (89,8% / 82,9%)
Verticaal (Z)	0,0195	0,0123	0,0269	0,0264	
Totaal (X/Y/Z)	0,0329	0,0150	0,0417	0,0332	
AHN2 Indirect BM20 Indirect	Mediaan	MAD	Gemiddelde	Standaardafwijking	Aantal
Horizontaal (X/Y)	0,0204	0,0125	0,0284	0,0280	4.180.449 (92,2% / 82,1%)
Verticaal (Z)	0,0206	0,0131	0,0289	0,0289	
Totaal (X/Y/Z)	0,0346	0,0161	0,0445	0,0359	
BM18 Indirect BM19 Indirect	Mediaan	MAD	Gemiddelde	Standaardafwijking	Aantal
Horizontaal (X/Y)	0,0185	0,0118	0,0274	0,0285	4.492.420 (91,8% / 91,5%)
Verticaal (Z)	0,0219	0,0141	0,0305	0,0297	
Totaal (X/Y/Z)	0,0343	0,0172	0,0448	0,0370	
BM18 Indirect BM20 Indirect	Mediaan	MAD	Gemiddelde	Standaardafwijking	Aantal
Horizontaal (X/Y)	0,0189	0,0120	0,0281	0,0292	4.618.295 (94,3% / 90,7%)
Verticaal (Z)	0,0214	0,0140	0,0302	0,0298	
Totaal (X/Y/Z)	0,0342	0,0174	0,0450	0,0376	
BM19 Indirect BM20 Indirect	Mediaan	MAD	Gemiddelde	Standaardafwijking	Aantal
Horizontaal (X/Y)	0,0176	0,0111	0,0261	0,0274	4.716.480 (96,0% / 92,7%)
Verticaal (Z)	0,0187	0,0123	0,0268	0,0273	
Totaal (X/Y/Z)	0,0310	0,0155	0,0409	0,0349	

Deze resultaten en de analyse ervan liggen buiten de projectbeschrijving van dit onderzoek. Hieronder toch een aantal observaties (gemaakt samen met de histogrammen, deze zijn apart meegeleverd):

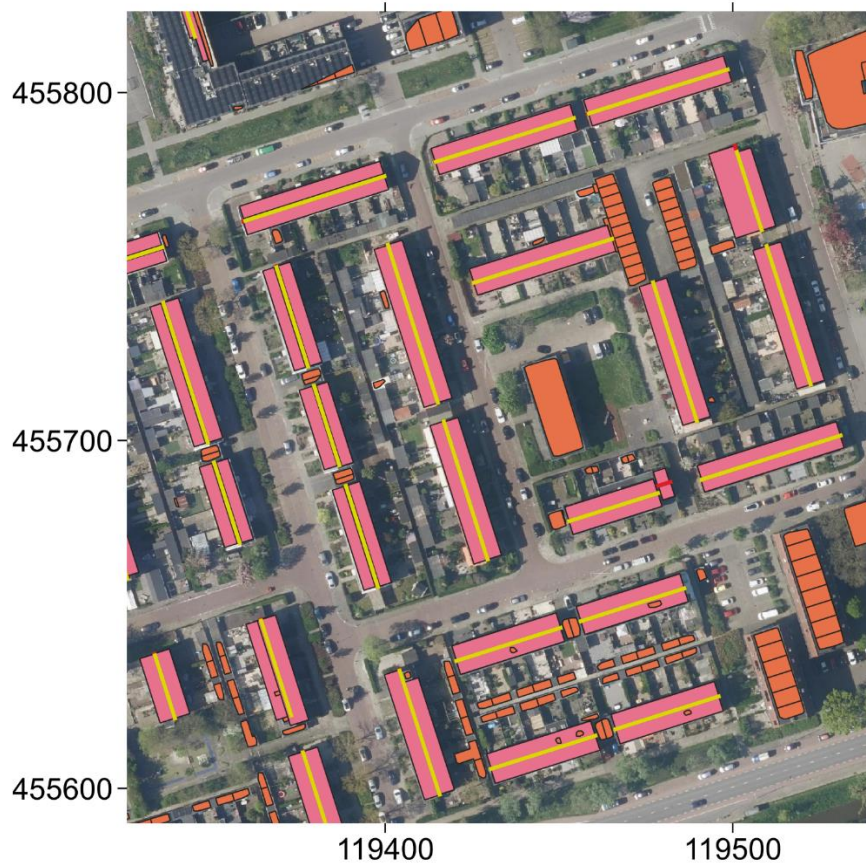
1. De gemiddelde verschillen in de posities van de noklijnen is voor alle bronnen kleiner dan 5 centimeter. De mediaan van de verschillen in de posities van de noklijnen is voor alle bronnen kleiner dan 4 centimeter.
2. De MAD is voor alle verschillen in de posities van de noklijnen kleiner dan de mediaan.
3. Alle vergelijkingen zijn in hoge mate representatief voor de complete bron (>80%).
4. De spreiding van de verschillen met alleen BM18, BM19, BM20 data is groter dan de vergelijkingen met alleen AHN2, AHN3, AHN4 data.

De verschillen uit Tabel 4 zijn ook samengevat in een vectorbestand (*Nationaal_Noklijnenverschil_v10.gpkg*), zodat de ruimtelijke spreiding van de verschillen ook kunnen worden bekeken (zie Figuur 14 voor een voorbeeld).

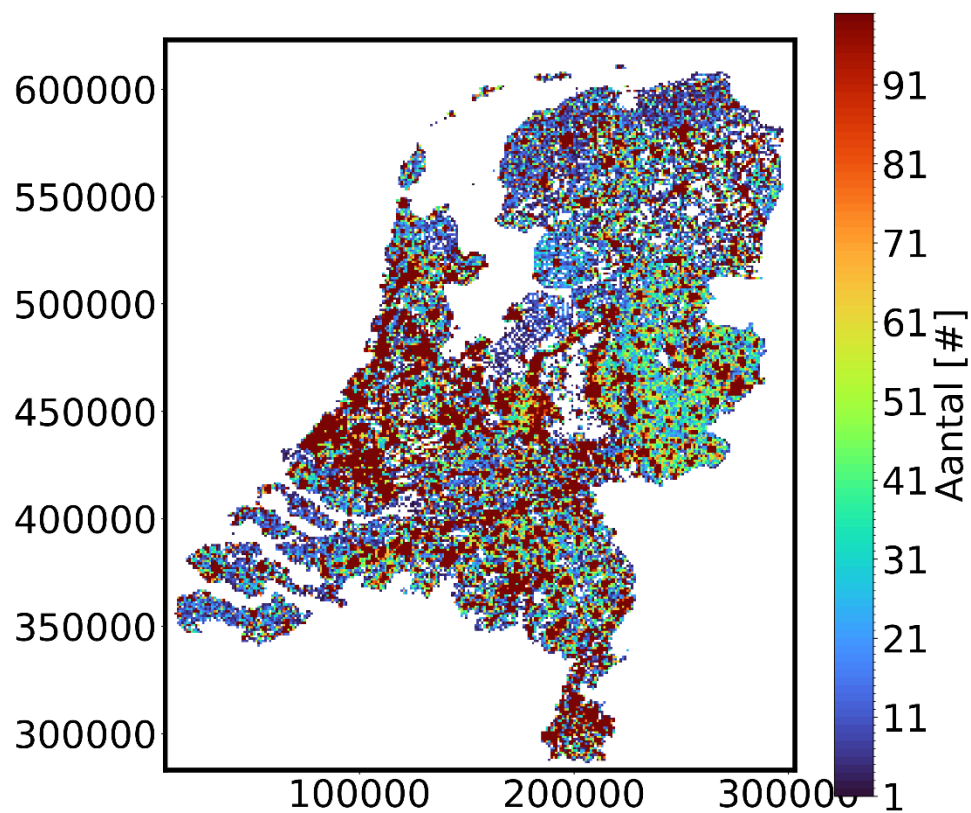
4.5. Voorbeelden



Figuur 10 Resultaten van DakGemak voor subtegel 31DZ2_05. Subfiguur a) noklijnen zonder BAG-registratie (in het rood). Subfiguur b) noklijnen zonder BAG-registratie (rood) en met BAG-registratie (geel) er overheen. Subfiguur c) is hetzelfde als b) met vlakke daken zonder BAG-registratie. Subfiguur d) is hetzelfde als b) met vlakke daken met BAG-registratie waarbij de vlakke daken zijn afgeknipt op de randen van de BAG percelen. De achtergrondfoto komt van Beeldmateriaal.



Figuur 11 Resultaten van DakGemak voor een deel van subtegel 31DZ2_05. Voorbeeld van dakvlakken samen met noklijnen (geel met BAG, rood zonder BAG) en vlakke daken (zonder noklijn). De achtergrondfoto komt van Beeldmateriaal.

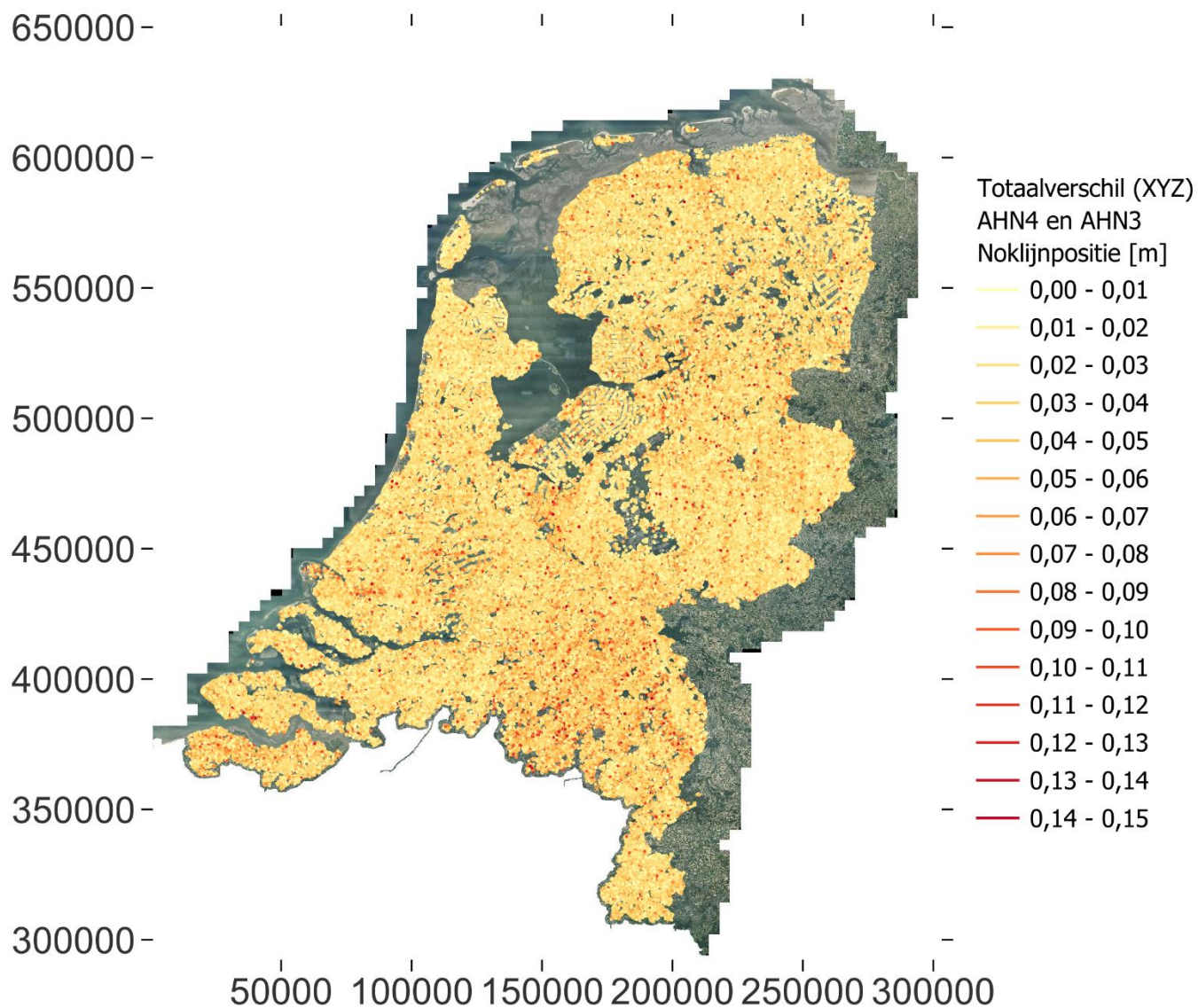


Figuur 12 Aantal BAG geïdentificeerde noklijnen op locatie in Nederland. De celgrootte is 1,25 km hoog en 1 km breed. Dit figuur is gemaakt met `imshow()` met `interpolation='none'`¹⁰.

¹⁰ Matplotlib (geraadpleegd op 20 november 2024): https://matplotlib.org/stable/api/as_gen/matplotlib.pyplot.imshow.html



Figuur 13 Resultaten van DakGemak voor een deel van subtegel 30DZ2_07. Subfiguur a) directe schatting op AHN4 data. Subfiguur b) indirecte schatting op AHN4 data. Subfiguur c) indirecte schatting op AHN3 data. Subfiguur d) indirecte schatting op AHN2 data. Subfiguur e) indirecte schatting op AHN1 data. De achtergrondfoto komt van Beeldmateriaal.



Figuur 14 Verschil in noklijnpositie tussen AHN4 (DakGemak indirect) en AHN3 (DakGemak indirect). Alle noklijnen zijn weergegeven betrokken bij de verschilberekening zijn weergegeven. Uitgebreide analyses kunnen worden gemaakt met behulp van het Nationaal_Noklijnenverschil_v10.gpkg bestand. De achtergrondfoto komt van Beeldmateriaal.

5. Discussie

5.1. Beperkingen

1. Noklijnen zijn altijd horizontaal. Dit volgt de aanname van de AHN-bestekken.
2. Beide daken moeten tussen de 20 en 70 graden hellen. Platte daken moeten 5 graden of minder hellen. Dat betekent dat de dataset daken tussen circa 5 en 20 graden (en 70 en hoger) niet heeft nagelopen en gebruikt.
3. De lengte van een noklijn wordt beperkt door het gebruik van een raster. Het uiteinde kan namelijk worden beperkt door het niet meenemen een cel waarin de punten uit de puntenwolk het werkelijke uiteinde markeren (af rondingsfout tot $\sqrt{2}$ celgrootte). Het kan ook dat het uiteinde binnen een cel ligt die wel wordt meegenomen, dan is er geen beperking. Een cel wordt overwogen als de standaardafwijking van de hoogtes (Z-coördinaat) niet te hoog is. De praktijk leert dat het om een fractie van de celgrootte gaat, zie 5.2 voor meer informatie.
4. Daken waarbij de noklijn niet recht (lineair) is in het horizontale vlak, of hoeken hebben kleiner dan `CLUSTER_ASPECT_DEG_MARGIN` worden niet of slecht geschat (zie Figuur 15).
5. De validatie wordt berekend tussen noklijnen op BAG niveau, terwijl het veel voorkomt dat alle aaneengesloten rijtjeshuizen dezelfde noklijn delen. Daarbij is er een grote kans dat deze ook als één noklijn is geschat, waarna deze is opgeknipt voor elk BAG pand. Dat betekent dat er herhalingen zijn die tijdens de validatie met gelijke (on)afhankelijkheid als alle andere noklijnen zijn meegenomen. Aan de hand de histogrammen en het feit dat dit voor alle vergelijkingen op dezelfde manier is gedaan is te verwachten dat het compenseren voor deze weging hoogstwaarschijnlijk niet tot een andere conclusie zal leiden.



Figuur 15 Een luchtfoto van een noklijn die geleidelijk afbuigt en een noklijn die onder een kleine hoek afbuigt. Beide verschijningen zijn niet uitgewerkt in DakGemak. De noklijnen zijn (indirect) geschat op basis van AHN4 data uit tegel 25GN1. De achtergrondfoto komt van Beeldmateriaal.

5.2. Rastergebruik

Punt 3 in Beperkingen kan worden verholpen door een extra verwerkingstap toe te voegen na de eerste schatting van de orthonormale basis. Zodra de basis bekend is kan in het verlengde van het vlak ($\pm$ een bepaalde normaalafstand onder en boven het vlak), dus buiten de geselecteerde cellen, verder worden gezocht naar punten die bij het vlak horen. Op deze wijze kan een preciezere omschrijving van de dakgrootte worden gevonden en daarmee ook de lengte van de noklijn nauwkeuriger worden bepaald. Een tweede argument is om met meer punten nauwkeuriger de orthonormale basis van het dakvlak te bepalen en daarmee ook de positie van de noklijn. Uit ervaring blijkt echter dat er in de meeste gevallen nauwelijks punten worden toegevoegd (AHN4, <100). Naast het gegeven dat de meest coherente delen van het dak al gevonden zijn, namelijk door de selectie op rasterniveau, is er in de huidige opzet geen significante winst te behalen voor de nauwkeurigheid van de positie van het dak en de noklijn. De lengte van de noklijn telt echter nog wel als voordeel. De verloren lengte hangt af van (een fractie van) de celgrootte en in de huidige opzet komt dat gemiddeld neer op een verlies van enkele centimeters. De geschatte toegevoegde rekentijd en complexiteit die hiervoor nodig zijn, lijken niet de moeite waard.

5.3. Tegelranden

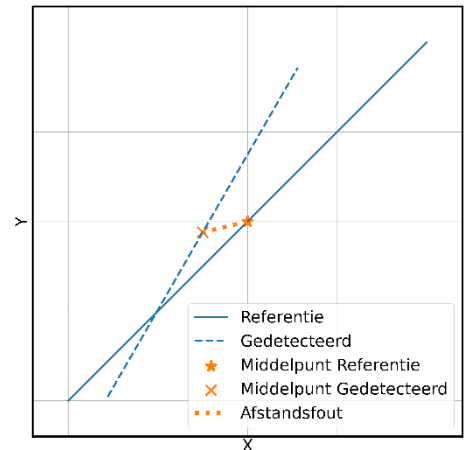
DakGemak houdt niet apart rekening met de randen van de invoer (sub)tegel(s). Dat betekent dat een noklijn of vlak dak dat op de rand van een tegel bevindt niet of slechts gedeeltelijk zal worden geschat. De tegels die zijn gebruikt (1,00 x 1,25 km²) voor de productie van een nationaal noklijnenbestand overlappen elkaar (20 m) aan alle zijden. Daarmee zullen dus voornamelijk grote gebouwen (of aaneengesloten rijtjeshuizen) onderbrekingen hebben. Daardoor is het mogelijk dat bij het samenvoegen van alle vectorbestanden meerdere noklijnen voor eenzelfde BAG-registratienummer ontstaan (zie Implementatie).

5.4. Validatie

Er zijn verschillende mogelijkheden voor de vergelijking van twee noklijnen. De volgende vier mogelijkheden worden hieronder beschouwd:

1. De afstand tussen de middelpunten van de twee noklijnen
2. De geprojecteerde afstand van het middelpunt van een van de twee noklijnen op de ander (of het gemiddelde van beiden).
3. De gemiddelde afstand tussen de uiteinden van de twee noklijnen
4. Het verschil in oriëntatie van de twee noklijnen

Co-lineaire noklijnen met verschillende lengtes wordt benadeeld door optie 1 en 3. Door het gebruik van een raster in DakGemak kan de lengte soms in beperkte mate worden bepaald (zie punt 3 in Beperkingen). De helft van het verschil in lengte aan een uiteinde wordt dan toegevoegd aan het horizontale verschil tussen twee noklijnen bij optie 1. Daarentegen is er geen horizontaal verschil bij optie 1 voor co-lineaire noklijnen die aan **beide** uiteinden met dezelfde lengte verlengd (of verkort) zijn. Het maakt dus ook uit **waar** het lengteverschil zit. Wanneer de referentie en gedetecteerde noklijn loodrecht op elkaar staan (en niet het midden van beide lijnen kruisen) is het verschil niet gedefinieerd bij optie 2. Optie 1, 2 en 3 beschouwen ook niet de hele noklijn, maar slechts één of enkele punten van de noklijn. Bij optie 3 levert eenzelfde draaiing andere verschillen op voor variërende lengtes van de noklijn. Deze bezwaren maken optie 1, 2 en 3 ongeschikt voor algemeen gebruik. Optie 4 is ongevoelig voor parallelle noklijnen en dus ook ongeschikt.



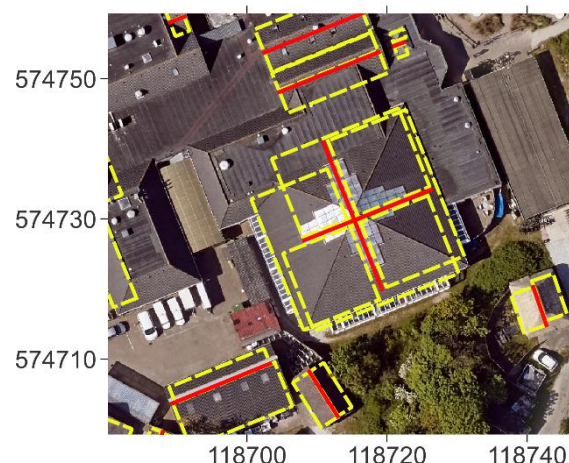
Figuur 16 2D weergave van een verschilberekening tussen twee noklijnen met optie 1. Het verschil wordt bepaald door de afstand tussen het middelpunt van de referentienoklijn en het middelpunt van de gedetecteerde noklijn.

5.5. Symmetriemethode

De symmetriemethode¹¹ was aan het begin van deze studie aangeduid als kanshebber voor een methode die dakvlakken en noklijnen nauwkeurig zouden kunnen schatten. De symmetriemethode werkt met (willekeurige) clusters punten aan beide zijden van een noklijn. DakGemak werkt daarentegen met zo groot mogelijke groepen punten om eerst vlakken te schatten en werkt daarmee op een hoger abstractieniveau. De implementatie en het testen van de symmetriemethode viel buiten de projectopdracht.

5.6. Nokpunt

Als vier daken bijeenkomen in een punt, dan worden de noklijnen dwars door de naastliggende daken getekend, die (vaak 90 graden) gedraaid liggen (zie Figuur 17).



Figuur 17 Een noklijn ligt soms op een onverwachte plek. De noklijnen zijn weergegeven met rode lijnen en dakvlakken zijn weergegeven met gele stippellijnen. De noklijnen liggen altijd tussen twee dakvlakken. De noklijnen zijn (indirect) geschat op basis van AHN4 data uit tegel 09BN2. De achtergrondfoto komt van Beeldmateriaal.

¹¹ S. Oude Elberink & G. Vosselman. Quality of Ridgelines: Ridgeline Extraction from Gable Roofs in Airborne Point Clouds (1 september 2024)

5.7. Race condities

DakGemak draait deels op meerdere processoren tegelijk. Hierdoor kunnen zogeheten race condities¹² optreden, waardoor berekende resultaten willekeurig onbetrouwbaar worden. Race condities laten zich vaak lastig vinden. Ter controle zijn daarom controlegetallen¹³ (*checksum* in het Engels) berekend tijdens het testen. Bij aanpassing van de code wordt aangeraden om de controlegetallen van de tussentijdse resultaten te berekenen. Tussentijdse resultaten kunnen worden geëxporteerd met `--export_intermediary`. Controlegetallen van Geopackages¹⁴ verschillen over het algemeen door het gebruik van interne een tijdsafdruk en zijn daarom niet geschikt voor dergelijke controles.

¹² Wikipedia (geraadpleegd op 18 november 2024): https://en.wikipedia.org/wiki/Race_condition

¹³ Wikipedia (geraadpleegd op 18 november 2024): <https://nl.wikipedia.org/wiki/Controlegetal>

¹⁴ Geopackage (geraadpleegd op 18 november 2024): <http://www.geopackage.org/guidance/extensions/metadata.html>

6. Conclusie

DakGemak berekent noklijnen uit puntenwolldata. Deze noklijnen verschillen 1 tot 3 cm in positie ten opzichte van de referentiedataset. Dit verschil is lager dan de positionele nauwkeurigheid van de metingen in de puntenwolldata. De aanname is wel dat gelijke posities ten opzichte van de referentiedatasets vertrouwen geven over de positionele nauwkeurigheid van de noklijnen. De **werkelijke** positie bestaat (nog) niet. Daarom wordt er in dit rapport expliciet van een **verschil** gesproken en niet van een **fout**.

DakGemak berekent een groot aantal noklijnen, groter dan voorheen was behaald binnen het Waterschapshuis (met PyDakje). De kwaliteit is echter niet (visueel) vergeleken.

7. Bijlagen

7.1. Installatie

De methode maakt gebruik van Python en enkele Python modules. Python modules kunnen worden geïnstalleerd met Miniconda. Op [deze website](#) kan Python en Miniconda ineen worden gedownload en vervolgens geïnstalleerd (op een Windows, macOS of Linux besturingssysteem). Na installatie kunnen de benodigde modules worden gedownload en geïnstalleerd in een afgezonderde omgeving met:

```
conda env create --name dakgemak --file=environments.yml
conda activate dakgemak
```

Of met:

```
conda create --name dakgemak python=3.12 pip
conda activate dakgemak
python -m pip install laspy[lazrs,laszip] ipykernel numpy pandas geopandas matplotlib numba scipy
python -m ipykernel install -user -name dakgemak
```

De omgeving is nu geïnstalleerd en geactiveerd.

Na het gebruik van DakGemak kan de omgeving worden gedeactiveerd met:

```
conda deactivate
```

7.2. Codevoorbeeld

DakGemak maakt gebruik van Pythonmodule Numba om de uitvoering van de code te versnellen. Als voorbeeld is hieronder een implementatie te vinden van `compute_binned_statistic_2d()` als alternatief van `scipy.stats.binned_statistic_2d()` uit de Pythonmodule Scipy voor de berekening van het aantal punten, de mediaan en de standaardafwijking van de Z-waarden binnen een cel.

```
@nb.jit(nb.types.Tuple((nb.uint16[:],nb.float64[:],nb.float64[:]))(nb.float64[:,:], nb.float64[:], nb.float64[:]),
nopython=True, parallel=True, fastmath=True)
def compute_binned_statistic_2d(p: np.ndarray, binsx: np.ndarray, binsy: np.ndarray) -> tuple[np.ndarray, np.ndarray,
np.ndarray]:
    """Custom version of Scipy's binned statistic.

    Sorting is key.

    Source to Scipy: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.binned\_statistic\_2d.html

    :param p: 2D-Numpy array (3,m) point coordinates
    :param binsx: 1D-Numpy array (n1), encompassing the entire value range of p in the 2nd (1st in Python) dimension
    :param binsy: 1D-Numpy array (n0), encompassing the entire value range of p in the 1st (0th in Python) dimension
    :return: tuple of 3 1D-Numpy arrays ((n0-1)*(n1-1)) bin count, z-value median, z-value standard deviation
    """

    # Initialize
    h, w = binsy.size - 1, binsx.size - 1
    i0 = np.digitize(p[1,:], binsy, right=False) - 1
    i1 = np.digitize(p[0,:], binsx, right=False) - 1

    i01 = (i0*w+i1).astype(np.uint16)

    # Get number of points
    n_p = p.shape[1]

    # Allow for multithreading
    idxs = np.argsort(i01).astype(np.uint16)

    # Allocate
    n_ld = np.zeros((w*h,), dtype=np.uint16)
    med_ld = np.empty((w*h,), dtype=np.float64)
    std_ld = np.empty((w*h,), dtype=np.float64)

    # Iterate over all points
    for k in nb.prange(n_p):

        # Check if current value is the same as the previous one
        if ((k > 0) and (i01[idxs[k]] == i01[idxs[k-1]])):
            continue

        # Check if unique
        if (k >= n_p - 1) or (i01[idxs[k]] != i01[idxs[k+1]]):
            n_ld[i01[idxs[k]]] = 1
            med_ld[i01[idxs[k]]] = np.float64(p[2,k])
            std_ld[i01[idxs[k]]] = 0
            continue

        # Find end point
        for k_search in range(k+1, n_p):
            if (i01[idxs[k_search]] != i01[idxs[k]]):
                break

        # We could use k_search now as end point
        pz = p[2,idxs[k:k_search]].astype(np.float64)

        # Set values (thread-safe)
        n_ld[i01[idxs[k]]] = pz.size
        med_ld[i01[idxs[k]]] = np.median(pz)
        std_ld[i01[idxs[k]]] = np.std(pz)

    return n_ld, med_ld, std_ld
```

7.3. Verschilberekening

De verschillen tussen noklijnen zijn berekend met behulp van onderstaande Python code. De eerste twee regels code (`line_detection = ...` en `line_reference = ...`) moeten worden aangepast door de gebruiker. Een motivatie voor de onderstaande aanpak is gegeven in Validatie in de Theorie.

```
# Present detected and reference ridge line
line_detection = np.array([[p0x_det, p0y_det, p0z_det], [plx_det, ply_det, plz_det]])
line_reference = np.array([[p0x_ref, p0y_ref, p0z_ref], [plx_ref, ply_ref, plz_ref]])

# Combine coordinates
p = np.vstack((line_detection[0,:], line_detection[1,:], line_reference[0,:], line_reference[1,:]))

# If lines cross, we should take half the area
# Following https://math.stackexchange.com/questions/1342435/how-to-determine-if-2-line-segments-cross

# Compute the area of four triangles (X/Y plane)
pi3 = np.array([0,2,3])
area_1 = 0.5*(p[pi3[1],0] * p[pi3[2],1] + p[pi3[0],0] * p[pi3[1],1] + p[pi3[0],1] * p[pi3[2],0]
              - p[pi3[0],1] * p[pi3[1],0] - p[pi3[0],0] * p[pi3[2],1] - p[pi3[1],1] * p[pi3[2],0])
pi3 = np.array([1,2,3])
area_2 = 0.5*(p[pi3[1],0] * p[pi3[2],1] + p[pi3[0],0] * p[pi3[1],1] + p[pi3[0],1] * p[pi3[2],0]
              - p[pi3[0],1] * p[pi3[1],0] - p[pi3[0],0] * p[pi3[2],1] - p[pi3[1],1] * p[pi3[2],0])
pi3 = np.array([2,0,1])
area_3 = 0.5*(p[pi3[1],0] * p[pi3[2],1] + p[pi3[0],0] * p[pi3[1],1] + p[pi3[0],1] * p[pi3[2],0]
              - p[pi3[0],1] * p[pi3[1],0] - p[pi3[0],0] * p[pi3[2],1] - p[pi3[1],1] * p[pi3[2],0])
pi3 = np.array([3,0,1])
area_4 = 0.5*(p[pi3[1],0] * p[pi3[2],1] + p[pi3[0],0] * p[pi3[1],1] + p[pi3[0],1] * p[pi3[2],0]
              - p[pi3[0],1] * p[pi3[1],0] - p[pi3[0],0] * p[pi3[2],1] - p[pi3[1],1] * p[pi3[2],0])

# Intersect if and only if
# We have double the area (4 triangles), so times 1/2, in case of intersection, 1/4
if (((area_1 * area_2) <= 0) and ((area_3 * area_4) <= 0)):
    area = (np.abs(area_1) + np.abs(area_2) + np.abs(area_3) + np.abs(area_4))/4
else:
    area = (np.abs(area_1) + np.abs(area_2) + np.abs(area_3) + np.abs(area_4))/2

# Divide by the average length to acquire the average separation
xy_error = 2 * area / (np.linalg.norm(p[0] - p[1]) + np.linalg.norm(p[2] - p[3]))

# Now let's do the same for the Z-axis/coordinate
# Create a new coordinate sqrt(x^2+y^2) and use z
p[:,0] = np.hypot(p[:,0], p[:,1])
p[:,1] = p[:,2]

# Compute the area of four triangles (XY/Z plane)
pi3 = np.array([0,2,3])
area_1 = 0.5*(p[pi3[1],0] * p[pi3[2],1] + p[pi3[0],0] * p[pi3[1],1] + p[pi3[0],1] * p[pi3[2],0]
              - p[pi3[0],1] * p[pi3[1],0] - p[pi3[0],0] * p[pi3[2],1] - p[pi3[1],1] * p[pi3[2],0])
pi3 = np.array([1,2,3])
area_2 = 0.5*(p[pi3[1],0] * p[pi3[2],1] + p[pi3[0],0] * p[pi3[1],1] + p[pi3[0],1] * p[pi3[2],0]
              - p[pi3[0],1] * p[pi3[1],0] - p[pi3[0],0] * p[pi3[2],1] - p[pi3[1],1] * p[pi3[2],0])
pi3 = np.array([2,0,1])
area_3 = 0.5*(p[pi3[1],0] * p[pi3[2],1] + p[pi3[0],0] * p[pi3[1],1] + p[pi3[0],1] * p[pi3[2],0]
              - p[pi3[0],1] * p[pi3[1],0] - p[pi3[0],0] * p[pi3[2],1] - p[pi3[1],1] * p[pi3[2],0])
pi3 = np.array([3,0,1])
area_4 = 0.5*(p[pi3[1],0] * p[pi3[2],1] + p[pi3[0],0] * p[pi3[1],1] + p[pi3[0],1] * p[pi3[2],0]
              - p[pi3[0],1] * p[pi3[1],0] - p[pi3[0],0] * p[pi3[2],1] - p[pi3[1],1] * p[pi3[2],0])

# Intersect if and only if
# We have double the area, so times 1/2, in case of intersection, 1/2
if (((area_1 * area_2) <= 0) and ((area_3 * area_4) <= 0)):
    area = (np.abs(area_1) + np.abs(area_2) + np.abs(area_3) + np.abs(area_4))/4
else:
    area = (np.abs(area_1) + np.abs(area_2) + np.abs(area_3) + np.abs(area_4))/2

# Divide by the average length to acquire the average separation
z_error = 2 * area / (np.linalg.norm(p[0] - p[1]) + np.linalg.norm(p[2] - p[3]))

# Combine errors quadratically
error = np.hypot(xy_error, z_error)
```

7.4. Attributentabel uitvoer

Het (data)type in de attributentabel is het type **voordat** het attribuut wordt opgeslagen in een vectorbestand. Verschillende vectorbestandsformaten hanteren deze op verschillende manieren en verschillen dus mogelijk met wat in de onderstaande tabellen wordt weergegeven. Zo kent het Geopackage formaat (.gpkg) bijvoorbeeld geen *unsigned integer* (UINT)¹⁵.

7.4.1. Noklijnen

Het noklijnenbestand **zonder** BAG-registratie: $\{bestandsnaam\}_{yyyyymmddTHHMMSS}_{ridges.\{ext\}}$ (zie Uitvoer) is opgezet volgens Tabel 5. Het noklijnenbestand **met** BAG-registratie $\{bestandsnaam\}_{yyyyymmddTHHMMSS}_{ridges_bag.\{ext\}}$ (zie Uitvoer) is opgezet volgens Tabel 5 met kolomnaam **BAG** (UINT64) als **extra kolom** tussen kolom 2 en 3. Dat betekent dat alle kolomnummers boven de 2 in het noklijnen met BAG-registratie met één zijn verhoogd. Het nationale noklijnenbestand heeft nog een extra kolom voor de identificatie van het bronbestand waaruit de noklijnen komen, deze kolom is te vinden **na** het rijnummer.

Tabel 5 Attributentabel voor de vectorbestanden met noklijnen. INT staat voor integer, U voor unsigned en FLT voor float.

Kolom-nummer	Attribuut	Attribuutkolom	Eenheid (context) [bereik]	Type
1	Rijnummer	fid	[0, →)	INT64
2	Noklijn identificatienummer	ridge_id	[1, →)	UINT64
3	Middelpunt noklijn X-coördinaat	ridge_center_x	m (RD X-coördinaat)	FLT64t
4	Middelpunt noklijn Y-coördinaat	ridge_center_y	m (RD Y-coördinaat)	FLT64
5	Middelpunt noklijn Z-coördinaat	ridge_center_z	m (NAP)	FLT64
6	Lengte noklijn	ridge_length	m (0, →)	FLT64
7	Richting noklijn	ridge_direction	° (t.o.v. RD +Y-as) (-90,90]	FLT64
8	Hoek daken onder de noklijn	roofs_angle	° [40,140]	FLT64
9	Identificatienummer dak 1	roof1_id	[1, →)	UINT32
10	Identificatienummer dak 2	roof2_id	[1, →)	UINT32
11	Gereduceerd identificatienummer dak 1	roof1_rid	[1, →)	UINT32
12	Gereduceerd identificatienummer dak 2	roof2_rid	[1, →)	UINT32
13	Hoek normaal dak 1	roof1_angle_z	° (t.o.v. RD +z-as) [20,70]	FLT64
14	Hoek normaal dak 2	roof2_angle_z	° (t.o.v. RD +z-as) [20,70]	FLT64
15	Gemiddelde X-coördinaat dak 1	roof1_pcenter_x	m (RD X-coördinaat)	FLT64
16	Gemiddelde Y-coördinaat dak 1	roof1_pcenter_y	m (RD Y-coördinaat)	FLT64
17	Gemiddelde Z-coördinaat dak 1	roof1_pcenter_z	m (NAP)	FLT64
18	Gemiddelde X-coördinaat dak 2	roof2_pcenter_x	m (RD X-coördinaat)	FLT64
19	Gemiddelde Y-coördinaat dak 2	roof2_pcenter_y	m (RD Y-coördinaat)	FLT64
20	Gemiddelde Z-coördinaat dak 2	roof2_pcenter_z	m (NAP)	FLT64
21	Standaardafwijking normaalafstanden dak 1	roof1_std_d	m [0, →)	FLT64
22	Standaardafwijking normaalafstanden dak 2	roof2_std_d	m [0, →)	FLT64
23	Minimum normaalpositie dak 1	roof1_min_d	m	FLT64
24	Minimum normaalpositie dak 2	roof2_min_d	m	FLT64
25	Maximum normaalpositie dak 1	roof1_max_d	m	FLT64
26	Maximum normaalpositie dak 2	roof2_max_d	m	FLT64
27	Aantal punten dak 1	roof1_points_n	# [1, →)	UINT32
28	Aantal punten dak 2	roof2_points_n	# [1, →)	UINT32
29	Aantal vlakken dak 1	roof1_patches_n	# [1, →)	UINT16
30	Aantal vlakken dak 2	roof2_patches_n	# [1, →)	UINT16
31	Oppervlakte (2D) dak 1	roof1_area_2d	m ² (0, →)	FLT64
32	Oppervlakte (2D) dak 2	roof2_area_2d	m ² (0, →)	FLT64
33	Punt dichtheid (2D) dak 1	roof1_point_density_2d	#/m ² (0, →)	FLT64
34	Punt dichtheid (2D) dak 2	roof2_point_density_2d	#/m ² (0, →)	FLT64
35	Oppervlakte (3D) dak 1	roof1_area_3d	m ² (0, →)	FLT64
36	Oppervlakte (3D) dak 2	roof2_area_3d	m ² (0, →)	FLT64
37	Punt dichtheid (3D) dak 1	roof1_point_density_3d	#/m ² (0, →)	FLT64
38	Punt dichtheid (3D) dak 2	roof2_point_density_3d	#/m ² (0, →)	FLT64
39	Geometrie noklijn	geom		FLT64(2,3)

7.4.1.1. Opmerkingen bij kolommen

Kolom 2, 3, 4 : Het middelpunt van de noklijn is het gemiddelde van de twee eindcoördinaten in kolom 39.

Kolom 7 : De richting van de noklijn is altijd positief in de Y-as en daarom dus altijd tussen -90° en 90°

Kolom 8 : De hoek onder de noklijn is gelijk aan 180° minus de hoek van de normaal (of hellingshoek) van dak 1 minus de hoek van de normaal (of hellingshoek) van dak 2.

¹⁵ Geopackage (geraadpleegd op 18 november 2024): <https://www.geopackage.org/spec140/index.html>

Kolom 11, 12 : Veel dakvlakken lijken op elkaar omdat ze zijn hergebruikt en iets zijn gedraaid (zie punt 9 in Directe schatting). Een dakvlak kan immers bijdragen aan meerdere noklijnen. Alhoewel het verschil klein is, zijn door de draaiing de hergebruikte daken niet exact hetzelfde meer. Kolom 10 en 11 verwijzen naar het dakvlak met het laagste identificatienummer van dat dak (kolom 9 en 10) van het hergebruikte dakvlak. Daarmee zijn de identificatienummers in kolom 9 en 10 uniek en in 11 en 12 niet.

Kolom 15 – 20 : De coördinaten geven het zwaartepunt van het dakvlak weer. Dit is de gemiddelde 3D coördinaat van de punten gebruikt voor de schatting van het dakvlak.

Kolom 23 – 26 : De minimum- en maximumnormaalpositie geeft het bereik van de relatieve normaalafstanden van de punten in de puntenwolk tot het dakvlak. Hierbij ligt de minimumpositie **onder** het dakvlak en de maximumpositie **boven** het dakvlak.

Kolom 29, 30 : Voor het dak uit het voorbeeld in Figuur 2 zou dit 3 en 2 (of andersom) zijn.

Kolom 31 – 38 : In 3D is het dakoppervlakte groter (Z-afstand is dan ongelijk aan 0) en dus de puntendichtheid lager.

Kolom 39 : De FLT64(2,3) structuur is opgeslagen als een LINESTRING Z (zie [Shapely](#)) en wordt omgezet naar het geometrieformaat van de uitvoer (bijvoorbeeld Geopackage) aan het einde van het programma.

7.4.2. Dakvlakken

Het dakvlakken bestand is alleen gemaakt zonder BAG-registratie: `{bestandsnaam}_{yyyymmddTHHMMSS}_direct_ridge_roofs.{ext}` is opgezet volgens Tabel 6. Het nationale noklijnenbestand heeft nog een extra kolom voor de identificatie van het bronbestand waaruit de noklijnen komen, deze kolom is te vinden **na** het rijnummer.

Tabel 6 Attributentabel voor de vectorbestanden met dakvlakken die gebruikt zijn voor het schatten van een noklijn. INT staat voor integer, U voor unsigned en FLT voor float.

Kolom-nummer	Attribuut	Attribuutkolom	Eenheid (context) [bereik]	Type
1	Rijnummer	fid	[0, →)	INT64
2	Identificatienummer dak	roof_id	[1, →)	UINT32
3	Gereduceerd identificatienummer dak	roof_rid	[1, →)	UINT32
4	Hoek normaal dak	angle_z	° (t.o.v. RD +z-as) [20,70]	FLT64
5	Gemiddelde X-coördinaat dak	pcenter_x	m (RD X-coördinaat)	FLT64
6	Gemiddelde Y-coördinaat dak	pcenter_y	m (RD Y-coördinaat)	FLT64
7	Gemiddelde Z-coördinaat dak	pcenter_z	m (NAP)	FLT64
8	Standaardafwijking normaalafstanden dak	std_d	m [0, →)	FLT64
9	Minimum normaalpositie dak	min_d	m	FLT64
10	Maximum normaalpositie dak	max_d	m	FLT64
11	Aantal punten dak	points_n	# [1, →)	UINT16
12	Aantal vlakken dak	patches_n	# [1, →)	UINT16
13	Oppervlakte (2D) dak	area_2d	m ² (0, →)	FLT64
14	Puntichtheid (2D) dak	point_density_2d	#/m ² (0, →)	FLT64
15	Oppervlakte (3D) dak	area_3d	m ² (0, →)	FLT64
16	Puntichtheid (3D) dak	point_density_3d	#/m ² (0, →)	FLT64
17	Geometrie dakvlak	geom		FLT64 (4,3)

7.4.2.1. Opmerkingen bij kolommen

Kolom 39 : De FLT64(4,3) structuur is opgeslagen als een POLYGON Z (zie [Shapely](#)) en wordt omgezet naar het geometrieformaat van de uitvoer (bijvoorbeeld Geopackage) aan het einde van het programma.

Zie 7.4.1.1 voor verdere opmerkingen.

7.4.3. Vlakke Daken

Het vlakke daken bestand zonder BAG-registratie: `{bestandsnaam}_{yyyymmddTHHMMSS}_surfaces.{ext}` (zie Uitvoer) en is opgezet volgens Tabel 7. Het vlakke daken bestand **met** BAG-registratie `{bestandsnaam}_{yyyymmddTHHMMSS}_surfaces_bag.{ext}` (zie Uitvoer) is opgezet volgens Tabel 7 met kolomnaam BAG (UINT64) als **extra kolom** tussen kolom 2 en 3. Dat betekent dat alle kolomnummers boven de 2 in het noklijnen met BAG-registratie met één zijn verhoogd. Ook worden er in dit geval het originele aantal punten, oppervlakte en puntichtheid weergegeven, omdat de veelhoeken zijn gereduceerd. Het nationale noklijnenbestand heeft nog een extra kolom voor de identificatie van het bronbestand waaruit de noklijnen komen, deze kolom is te vinden **na** het rijnummer.

Tabel 7 Attributentabel voor de vectorbestanden met vlakke daken.

Kolom-nummer	Attribuut	Attribuutkolom	Eenheid (context) [bereik]	Type
1	Rijnummer	fid	[0, →)	INT64
2	Vlak dak identificatienummer	surface_id	[1, →)	UINT64
3	Hoek normaal	angle_z	° (t.o.v. RD +z-as) [0,5]	FLT64
4	Gemiddelde Z-coördinaat	mean_z	m (NAP)	FLT64
5	Standaardafwijking Z-coördinaat	std_z	m (0, →)	FLT64
6	Mediaan Z-coördinaat	median_z	m (NAP)	FLT64
7	Mediaan van absolute afwijkingen Z-coördinaat	mad_z	m (0, →)	FLT64
8	Gemiddelde x-coördinaat dak 2	pcenter_x	m (RD X-coördinaat)	FLT64
9	Gemiddelde y-coördinaat dak 2	pcenter_y	m (RD Y-coördinaat)	FLT64
10	Aantal punten	points_n	# [1, →)	UINT32
11	Oppervlakte (2D)	area	m ² (0, →)	FLT64
12	Puntichtheid (2D)	point_density	#/m ² (0, →)	FLT64
13	Geometrie vlak dak	geom		FLT64(n,3)

7.4.3.1. Opmerkingen bij kolommen

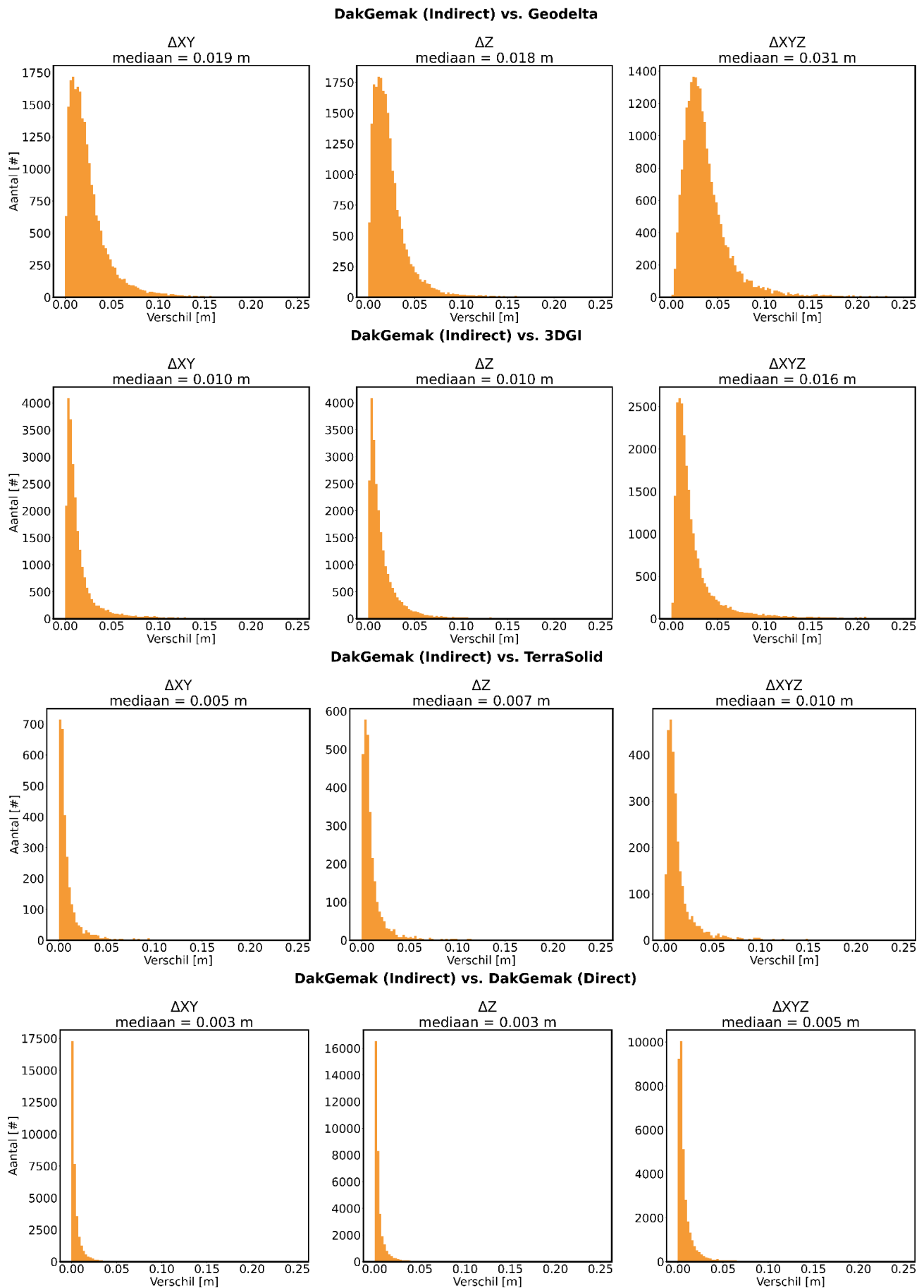
Kolom 10 : Het aantal punten correspondeert beter met de oppervlaktes die afgeknipt zijn op de BAG grenzen, dan de initiële schatting van het oppervlakte.

Kolom 10, 12 : Deze waarden horen bij alle vlakke daken met een gelijk identificatienummer (kolom 2).

Kolom 39 : De FLT64(n,3) structuur is opgeslagen als een POLYGON Z (zie [Shapely](#)) en wordt omgezet naar het geometrieformaat van de uitvoer (bijvoorbeeld Geopackage) aan het einde van het programma.

Zie 7.4.1.1 voor verdere opmerkingen.

7.5. Validatie histogrammen



Figuur 18 Verschillen in de noklijnpositie tussen de schattingen van indirecte methode van DakGemak en op rij 1) Geodelta (GD), op rij 2) 3DGI (3D), op rij 3) TerraSolid (TS) en op rij 4) de directe methode van DakGemak. De eerste kolom geeft het verschil in het horizontale vlak weer, de tweede kolom geeft het verschil langs de verticale as en de derde kolom geeft het totaalverschil weer.

De histogrammen voor de vergelijkingen tussen de verschillende datasets zijn apart meegeleverd vanwege het grote aantal (16 keer 3 histogrammen in totaal).

7.6. Uitleg figuren

Tabel 8 Uitvoerbestanden bij het exporteren van figuren met `dakgemak_direct.py`.

Voorbeeld (Suffix)	Uitleg
AHN4_31DZ2_05_20241111T111110_direct_median_z.png	Raster met mediaan van Z-waardes van punten in elke cel.
AHN4_31DZ2_05_20241111T111111_direct_count.png	Raster met puntenaantal in elke cel
AHN4_31DZ2_05_20241111T111111_direct_slope.png	Raster met hellingshoek berekend uit median_z
AHN4_31DZ2_05_20241111T111111_direct_aspect.png	Raster met hellingsrichtingshoek berekend uit median_z
AHN4_31DZ2_05_20241111T111111_direct_svalid.png	Geldige hellende cellen na stap 2 in Directe schatting
AHN4_31DZ2_05_20241111T111111_direct_fvalid.png	Geldige vlakke cellen na stap 2 in Directe schatting
AHN4_31DZ2_05_20241111T111111_direct_label_scluster.png	Hellende clusters na stap 4 in Directe schatting
AHN4_31DZ2_05_20241111T111111_direct_label_fcluster.png	Vlakke clusters na stap 4 in Directe schatting
AHN4_31DZ2_05_20241111T111111_direct_label_scluster_clean.png	Hellende clusters na stap 5 in Directe schatting
AHN4_31DZ2_05_20241111T111111_direct_label_fcluster_clean.png	Vlakke clusters na stap 5 in Directe schatting
AHN4_31DZ2_05_20241111T111111_direct_splanes.png	Dakvlakken met punten na stap 6 in Directe schatting
AHN4_31DZ2_05_20241111T111111_direct_sdiscarded.png	Punten die niet bij een hellend dakvlak horen na stap 6 in Directe schatting
AHN4_31DZ2_05_20241111T111111_direct_splanes_final.png	Dakvlakken met punten na stap 10 in Directe schatting

Tabel 9 Uitvoerbestanden bij het exporteren van figuren met `dakgemak_indirect.py`.

Voorbeeld (Suffix)	Uitleg
AHN4_31DZ2_05_20241111T111110_indirect_load_mask.png	Raster waarbij punten die in cellen liggen met een 1 wel worden geladen en met een 0 niet.

7.7. Controlegetallen

Tabel 10 Controlegetallen voor een integriteitscontrole van opgeleverde bestanden.

Bestand	MD5 controlegetal
dakgemak_direct.py	7b05140d69cf475283beea2ed27278be
dakgemak_indirect.py	3f7a14223a27313ba592eca0bae4f050
dakgemak_aux.py	dde2cdf1ccdac73a1492bae009ebc7
dakgemak_difference.py	1fc05eccce2157493c93e27ce6394072
dakgemak_environment.yml	19b983328d5a3c70e9530ef3326ba2fc
AHN04_NL_direct_ridges_v10.gpkg	b88a046af79caa64a6ece30f95f77ad7
AHN04_NL_direct_ridges_bag_v10.gpkg g	93948db895b8313222bcae884f195402
AHN04_NL_direct_ridge_roofs_v10.gpkg	98175ffd2472c87fc15b80b1fcc43c03
AHN04_NL_direct_surfaces_v10.gpkg	d9b3126b09e2d630040e3eda90614c79
AHN04_NL_direct_surfaces_bag_v10.gpkg	7f9bb39505d9fb11fed3b2ea7771d7b9
AHN04_NL_indirect_ridges_v10.gpkg	df0ad5c873d4427d17005758b993aeab
AHN04_NL_indirect_ridges_bag_v10.gpkg	244c485df7fe46f0e92be8b529202ca1
AHN04_NL_indirect_surfaces_bag_v10.gpkg	1b96ec20ef74577ed048cbd0b78ba5e9
AHN03_NL_indirect_ridges_v10.gpkg	da2f4f848d0a18b67f3b3c9dc29e3880
AHN03_NL_indirect_ridges_bag_v10.gpkg g	3c8a3540b6dc8dfa26614d0748aad8c9
AHN03_NL_indirect_surfaces_bag_v10.gpkg	992cba3a70032d701e82bf749088ac45
AHN02_NL_indirect_ridges_v10.gpkg	bfafe98080536fa0a059309603974193
AHN02_NL_indirect_ridges_bag_v10.gpkg g	0157e0bb1800639e8c21d27e8500822f
AHN02_NL_indirect_surfaces_bag_v10.gpkg	c9ba01379a425beab7cc17bd4dcd5db
BM20_NL_indirect_ridges_v10.gpkg	f70cb3d7631ba3f009feb946c3440fa1
BM20_NL_indirect_ridges_bag_v10.gpkg	7aebd8639d587be8c64f19ae6958ceb7
BM20_NL_indirect_surfaces_bag_v10.gpkg	cef183ea2e1b3adbf752d3d2e85984c
BM19_NL_indirect_ridges_v10.gpkg	be348e7e82c24052882e1b6af2ef3e6f
BM19_NL_indirect_ridges_bag_v10.gpkg g	6f348374cc63b12e0696a4bbc06d1379
BM19_NL_indirect_surfaces_bag_v10.gpkg	71052e80816b4bf68394541eb7fe25ee
BM18_NL_indirect_ridges_v10.gpkg	c2413c1c6a4093cf7af70da7617f08d0
BM18_NL_indirect_ridges_bag_v10.gpkg g	f13f31909e10527c40ad045311925b17
BM18_NL_indirect_surfaces_bag_v10.gpkg	a073d3735dfcaed439f4c3830326b5b5
Nationaal_Noklijnenverschil_v10.gpkg	4c3505941becf9e75785667d59777332
DakGemak_Indirect_AHN4_DakGemak_Direct_AHN4_20241125T132910_differences.png	3dfd07487f2140d51cc3d536a4edaf38
DakGemak_Indirect_AHN4_DakGemak_Indirect_AHN3_20241125T135102_differences.png	4a61e6e61e005f0616026e0ea0259cd3
DakGemak_Indirect_AHN4_DakGemak_Indirect_AHN2_20241125T141014_differences.png	df094b3e0e8a5f9352b90cb0717780a0
DakGemak_Indirect_AHN4_DakGemak_Indirect_BM18_20241125T142942_differences.png	eab279bbd873aeec60a812a6dda83c99
DakGemak_Indirect_AHN4_DakGemak_Indirect_BM19_20241125T145109_differences.png	7d6df62b1d728e720c9f6830bcc9b3d3
DakGemak_Indirect_AHN4_DakGemak_Indirect_BM20_20241125T151250_differences.png	1ca2c59e05589f90cdc95deb3393ca91
DakGemak_Indirect_AHN3_DakGemak_Indirect_AHN2_20241125T153512_differences.png	14b2d69fb1327617cf369002b680a917
DakGemak_Indirect_AHN3_DakGemak_Indirect_BM18_20241125T155308_differences.png	c77917e4da9d17c09a9a32aef57599db
DakGemak_Indirect_AHN3_DakGemak_Indirect_BM19_20241125T161122_differences.png	cd3858d41e0aa6596282ab40434764c3
DakGemak_Indirect_AHN3_DakGemak_Indirect_BM20_20241125T162915_differences.png	702d87dc254d519057d33c8f70ba5cbc
DakGemak_Indirect_AHN2_DakGemak_Indirect_BM18_20241125T164748_differences.png	f09f6a58250910afa5ec94b89708fd98
DakGemak_Indirect_AHN2_DakGemak_Indirect_BM19_20241125T170531_differences.png	0a6a72e1e9c2c781ed9e6100d7ef2ef2
DakGemak_Indirect_AHN2_DakGemak_Indirect_BM20_20241125T172331_differences.png	ea997e192531937cd0fc1dcca80bba8
DakGemak_Indirect_BM18_DakGemak_Indirect_BM19_20241125T174154_differences.png	380076b691c63aa46d3e74ceabfdc188
DakGemak_Indirect_BM18_DakGemak_Indirect_BM20_20241125T202020_differences.png	f71b4787188c162aa649a967df60d198
DakGemak_Indirect_BM19_DakGemak_Indirect_BM20_20241125T203902_differences.png	c086192bcfab43e352db3caf24d21e10