# Proces

Om de LAZ data van het hoogtemodel te gebruiken in combinatie met de BGT, worden twee verschillende soorten LAZ data ingeladen.

1. AHN

Het AHN4 wordt uiteindelijk gebruikt om als eerst te combineren met de BGT data. De transformatie technieken volgens de workflow van Nico worden hierop toegepast. (Voor nu wordt AHN3 gebruikt omdat AHN4 nog niet volledig beschikbaar is)

1. Lidar data van het volledig maaiveld

Dit omvat alle data die de Lidar metingen heeft ingewonnen. Dus ook auto’s, bomen etc. Dit zal daarna toegevoegd worden aan het AHN en ook getransformeerd worden volgens de workflow van Nico.

De site data.amsterdam.nl/data laat de verschillen tussen deze twee databronnen zien. Laidar maakt het namelijk mogelijk om data van elkaar te onderscheiden en het verschil tussen maaiveld en objecten boven de grond te herkennen. In data.amsterdam kunnen twee verschillende data aangevinkt worden:

1. Terreinmodel (DTM AHN)
   1. Staat dus gelijk aan AHN, waarbij objecten als bomen en auto’s weggelaten zijn
2. Oppervlaktemodel (DSM AHN)

**Activate environment containing pdal installation**

conda activate pdalenv

**create loading json file**

[

      {

*"type"* : "readers.las",

*"filename"* : "<path\_to\_lazData>\<name\_lazData>",

*"spatialreference"*:"EPSG:28992"

       },

       {

*"type"*:"filters.chipper",

*"capacity"*:400

    },

    {

*"type"*:"writers.pgpointcloud",

*"connection"*:"host='localhost' dbname='<dbname>' user='postgres' password='<password>'",

*"table"*:"paasheuvel",

*"compression"*:"dimensional",

*"srid"*:"28992"

    }

**create pipeline for loading laz data**

pdal pipeline load\_paasheuvel\_laz\_data.json

Nu is de LAZ data overgeschreven in de Postgres database, volgens de settings van de json file

**Mesh to point**

Binnen QGIS kan de LAZ data ingeladen worden uit Postgres. De LAZ data is omgezet naar meshes tijdens de transformatie. Binnen QGIS kunnen de punten uit de meshes gehaald worden met de SQL code:

SELECT

PcPoint(

 PC\_Explode(pa))

FROM paasheuvel

WHERE id<10;

Echter krijg je dan een PCPoint dtype:

"{""pcid"":1,""pt"":[486,2,3,0,0,1,-17,6,110,0,0,0,2.70047e+008,117227,481054,-1.24]}"

**Method 2: ahn3 inladen**

Ook kan het ahn3 model van <https://app.pdok.nl/ahn3-downloadpage/> ingeladen worden. Het verkregen bestand is dan een .TIF file. Aan de hand van de Python Package rasterio kan een tif file omgezet worden in punten.

import numpy as np

import rasterio

with rasterio.open("<path\_to\_TIFfile>\M\_25DZ2.TIF") as src:

    r= src.read()

om tif file naar points om te zetten en op te slaan zijn twee packages nodig. Geopandas en shapely. Voor geopandas moeten de dependencies: GDAL, Fiona en een windows binary voor packages. (zoals aangeraden is op <https://gis.stackexchange.com/questions/330840/error-installing-geopandas>). Op deze twee sites kunnen Fiona en de binary wheels gedownload worden.

1. <https://www.lfd.uci.edu/~gohlke/pythonlibs/#fiona>
2. <https://www.lfd.uci.edu/~gohlke/pythonlibs/>

vervolgens kan de “pip install <wheel path>” uitgevoerd worden.

Daarna kan de pip install geopandas wel uitgevoerd worden.

**TIF to csv to postgres**

Van <https://geotiles.nl/> een TIF file downloaden. (DTM, 0.5m).

Dan TIF omzetten naar CSV binnen de OSgeo4w shell. Hiervoor zijn GDAL/pandas/np dependencies.

gdal2xyz -allbands -skipnodata "<path\_to\_TIFfile>\M\_25DN2.TIF" "<path\_to\_outputFile>\output.txt"

Het outputbestand output.txt moet nog herschreven worden naar output.csv, zodat het ingelezen kan worden door ogr2.

Het inladen van het csv bestand in Postgres gaat via ogr2ogr, via dezelfde command dat de BGT data heeft ingeladen. Dit duurt lang (zon 20 min)

ogr2ogr -f PostgreSQL "PG:host=localhost port=5432 user=postgres password=<password> dbname=paasheuvel\_csvFromTif" "<path\_to\_outputFile>\output.csv"

**Cropping data**

voor het croppen van de data is een SQL query nodig. In Qgis kan de BGT data van de Paasheuvel ingeladen worden. Daarop kan je makkelijk de coordinaten vinden van de hoeken door erover een te hoveren. Vervolgens moet ern een SQL query geschreven worden dat de data dropped dat buiten deze coordinaten vallen.

DROP TABLE FROM output

WHERE field1 <117226.8 --min x

AND field1 >117610.6 --max x

AND field2 >481053 --min y

AND field2 <481439; --max y