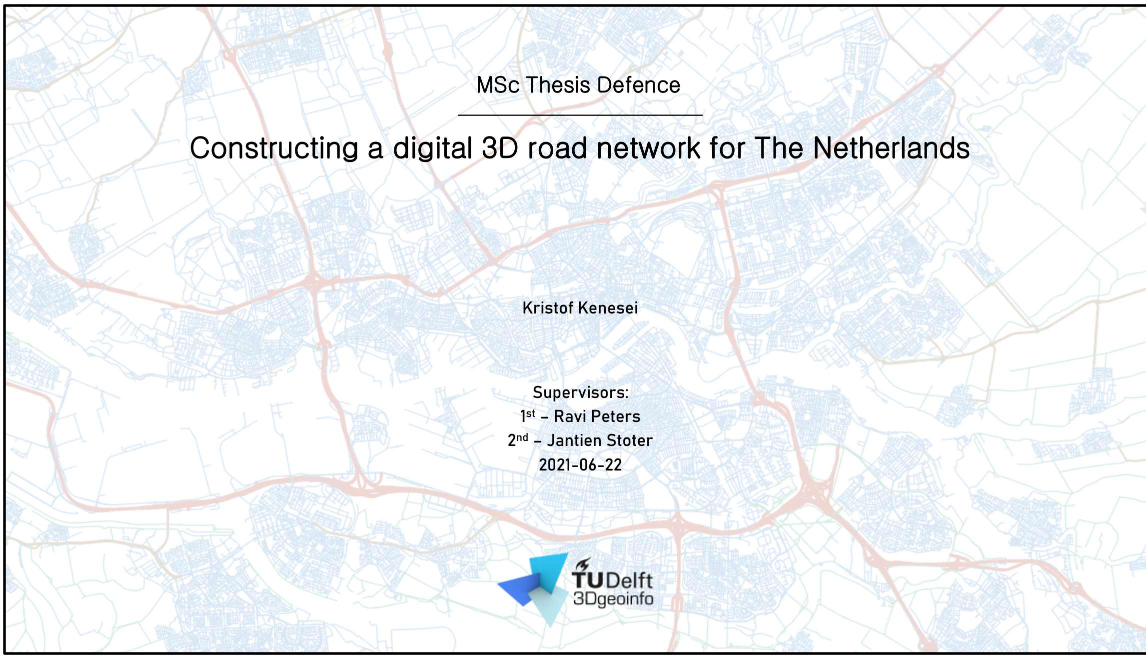
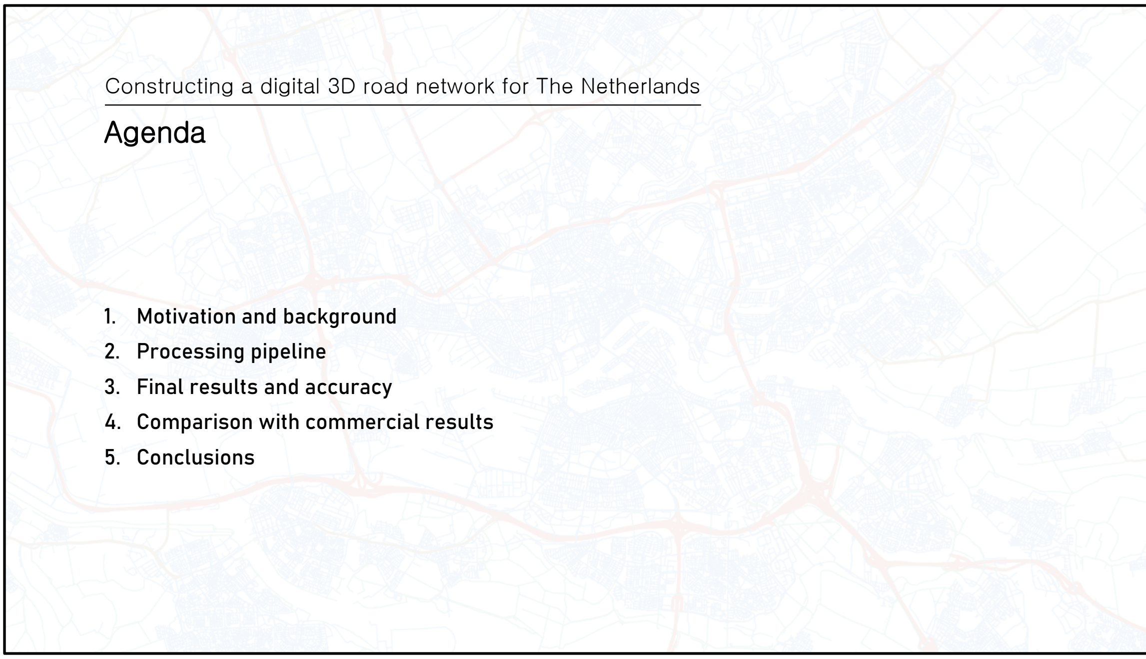


Welcome to my thesis defence. The title of my graduation project is *"Constructing a digital 3D road network for The Netherlands"*.



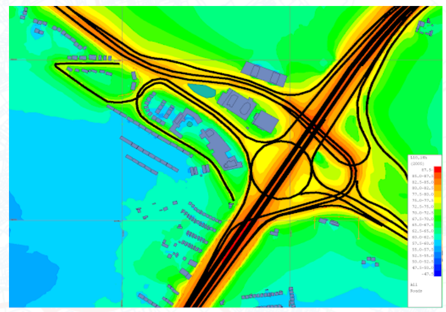
- Agenda of the presentation will be:
  - Motivation and some background
  - Details of the pipeline design and showcase of intermediate results
  - Final results and accuracy assessment
  - Outcome of comparison with the commercial results
  - Concluding remarks

- Full name is *Nationaal Wegenbestand*, which stands for national road product
- Open data geospatial product of The Netherlands
- Produced and maintained by NDW
    - They are the commercial “client” of this project
- Graph-like representation of all named roads in The Netherlands
    - *Knooppunten:* real-life intersections, i.e. graph nodes
    - *Wegwakken:* road centreline segments, i.e. graph edges

Motivation and background

Purpose of georeferencing

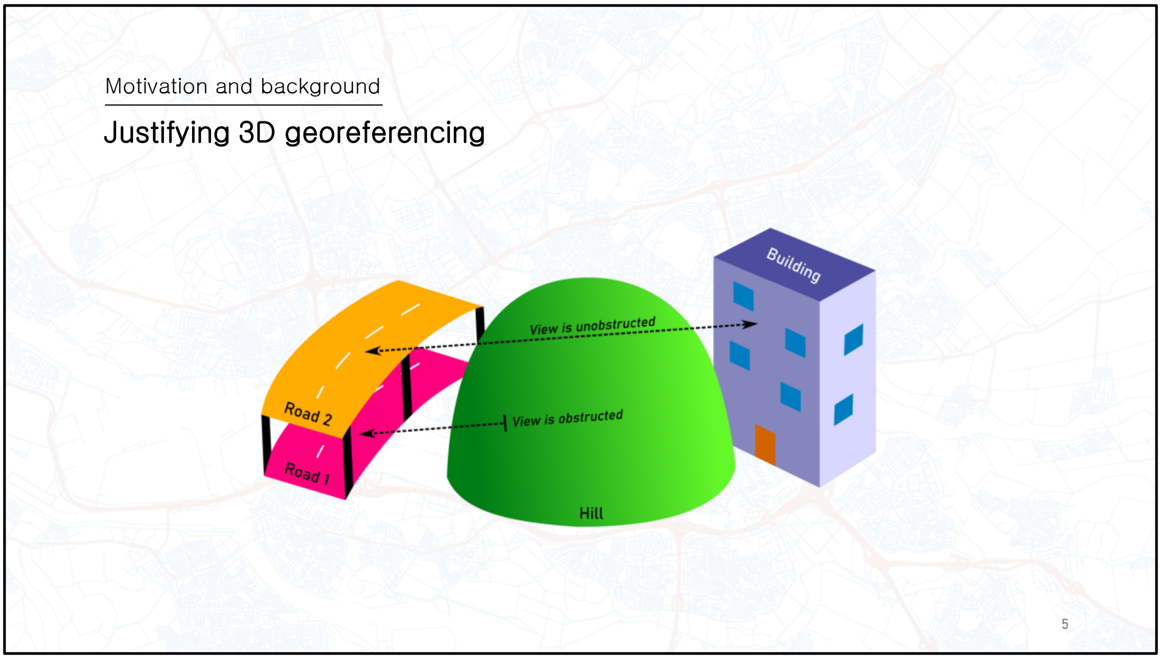
- In general:
  - We can relate the road to its physical surroundings – required in many applications
- In this specific case:
  - This project is relevant to noise modelling
  - → Need to know the geographical source of noise



http://www.noisemap.ltd.uk/wpress/features/roads/

4

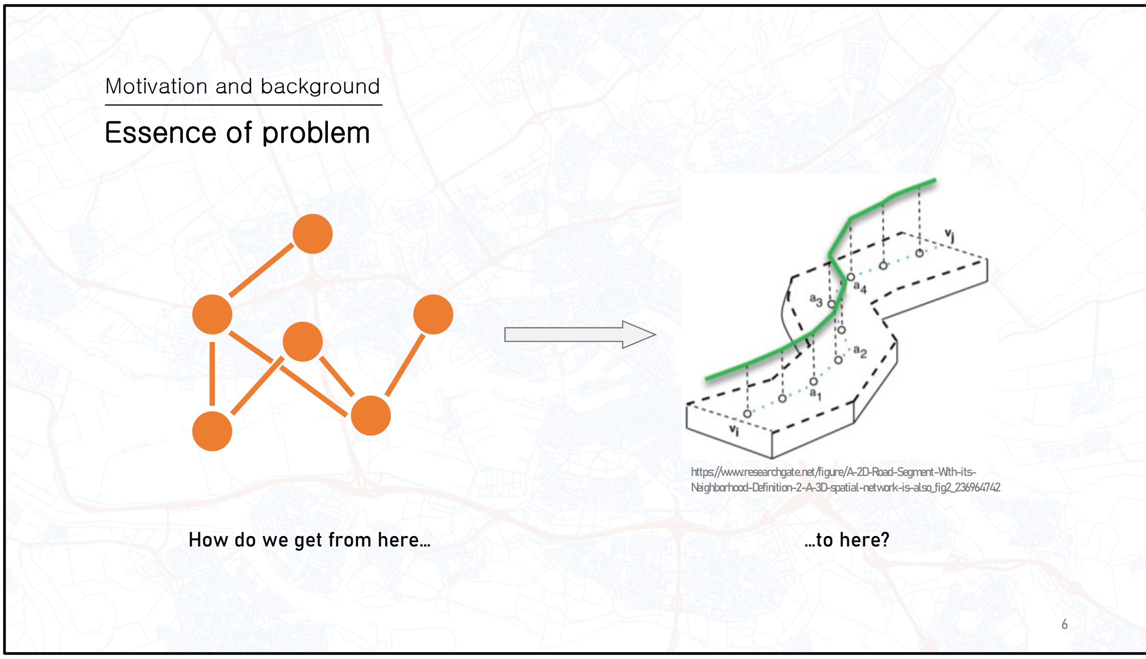
- Georeferencing allows us to relate roads to their surroundings
  - I.e. to correlate with external information, such as addresses in route planning
- Our scenario: NWB is a traffic noise source in a noise propagation model
  - Position relative to surrounding objects needs to be known
  - Otherwise the noise load on the objects cannot be determined



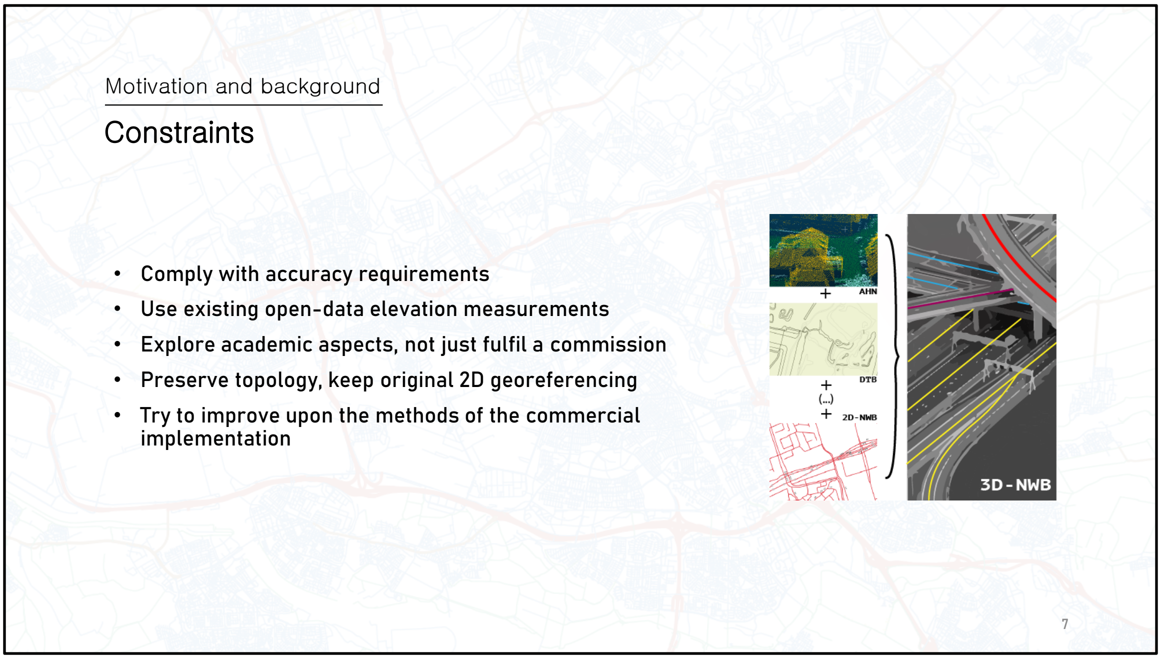
- Traffic noise modelling traditionally done in 2D, but these are not very realistic
- A 3D noise propagation model should use a 3D road network

- Consider the example:
    - This is a best case scenario, we already have a terrain model and 3D buildings
    - If we model the road to always lie on the terrain, then the hill (part of the terrain) will block the noise
    - If we consider that it is in fact an elevated road, some of the noise will propagate to the building

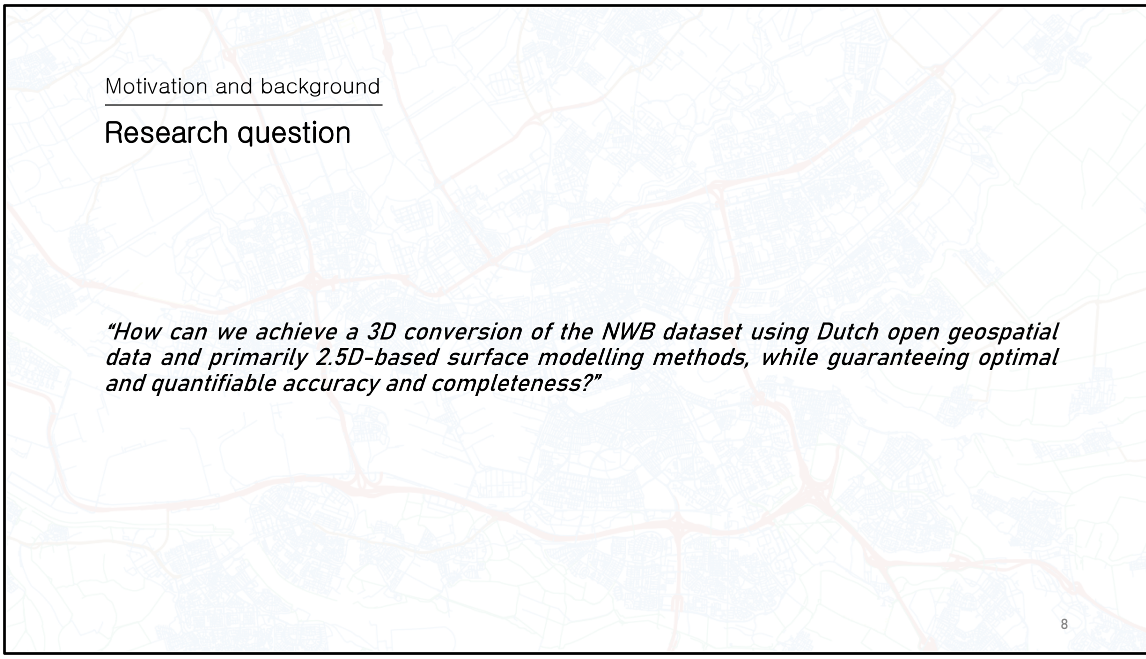
- New noise legislations in The Netherlands prescribe a 3D model



- So how do we get from a 2D road network (graph) to a 3D road network?
- In other words, how do we enrich NWB with elevation data?



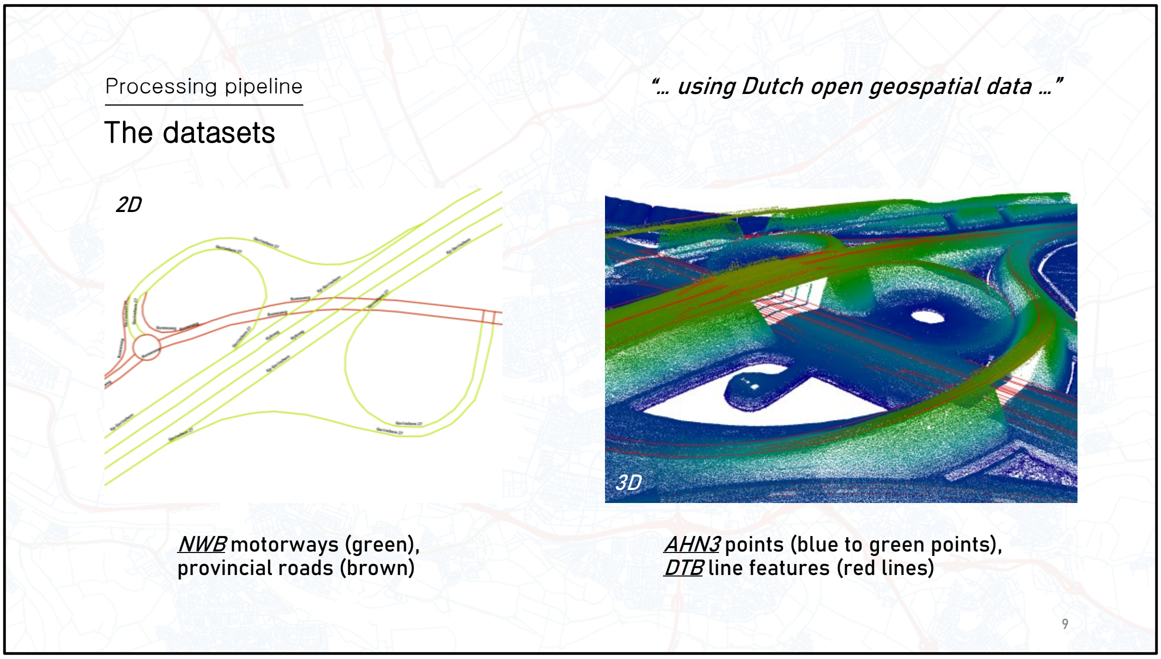
- In addition to the base problem, there are a few constraints
  - Noise regulations prescribe a 20 cm minimum accuracy at 2 STDs
  - We wish to use open data datasets only
  - Explore academic aspects, not only complete a commercial commission
  - Preserve topology, keep the 2D georeferencing
  - Compare with, and try to improve upon, the commercial implementation
    - Was developed in parallel with academic project



- In the end we came up with the research question:

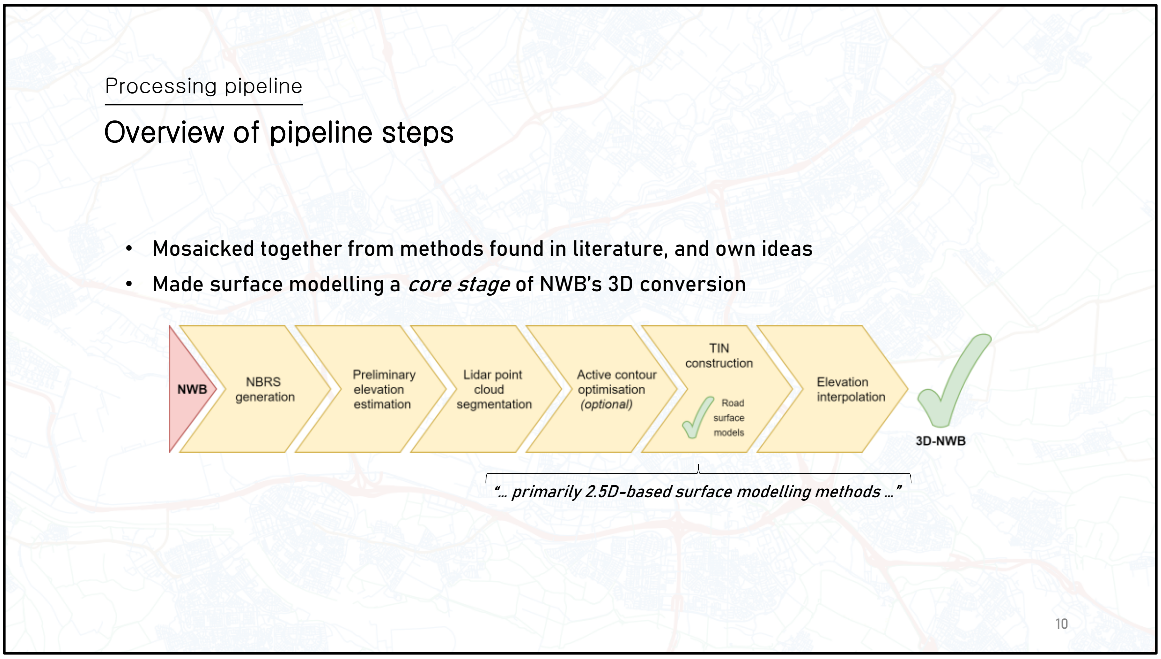
"How can we achieve a 3D conversion of the NWB dataset using Dutch open geospatial data and primarily 2.5D-based surface modelling methods, while guaranteeing optimal and quantifiable accuracy and completeness?"

- 2.5D-based surface modelling and scientific accuracy assessment are mentioned specifically because these are the main added academic aspects



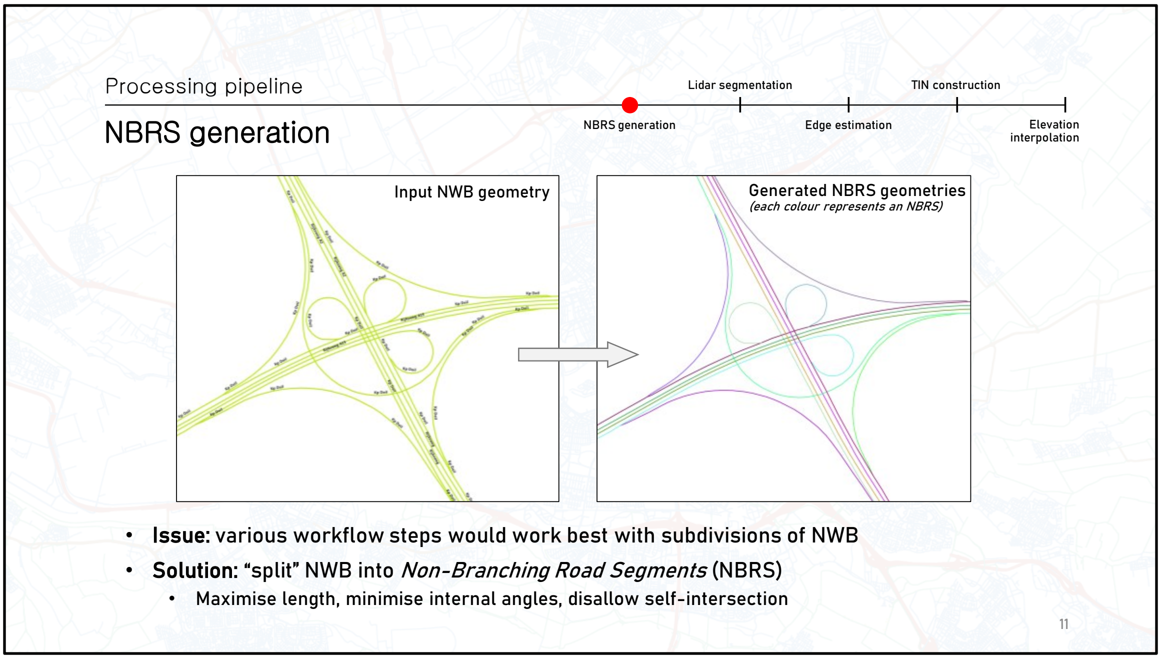
- The datasets are:
    - NWB – needs to be converted to 3D
    - We are only interested in motorways and provincial roads
- AHN3 – the Dutch national remote sensing dataset
    - Cannot penetrate objects that are opaque to light
    - Difficult to sense 3D relationships due to this
    - Feature extraction is necessary, which is never perfectly accurate
- DTB – a road management dataset of The Netherlands
    - We use the road marking line models from this dataset
    - This can provide coverage in AHN3 gaps
    - Coverage is limited, and it only exists for motorways

- Highlighted the relevant part of the main research question



- This flowchart shows our planned processing pipeline
- Result of a preparation process that spanned about two months
    - Performed the necessary literature review
    - Consulted with NDW and their commercial developer
    - Mosaicked together the procedure based on the above, and my own ideas
    - Made the generation of road surface TIN models a core stage
        - Producing surface models is for research purposes only

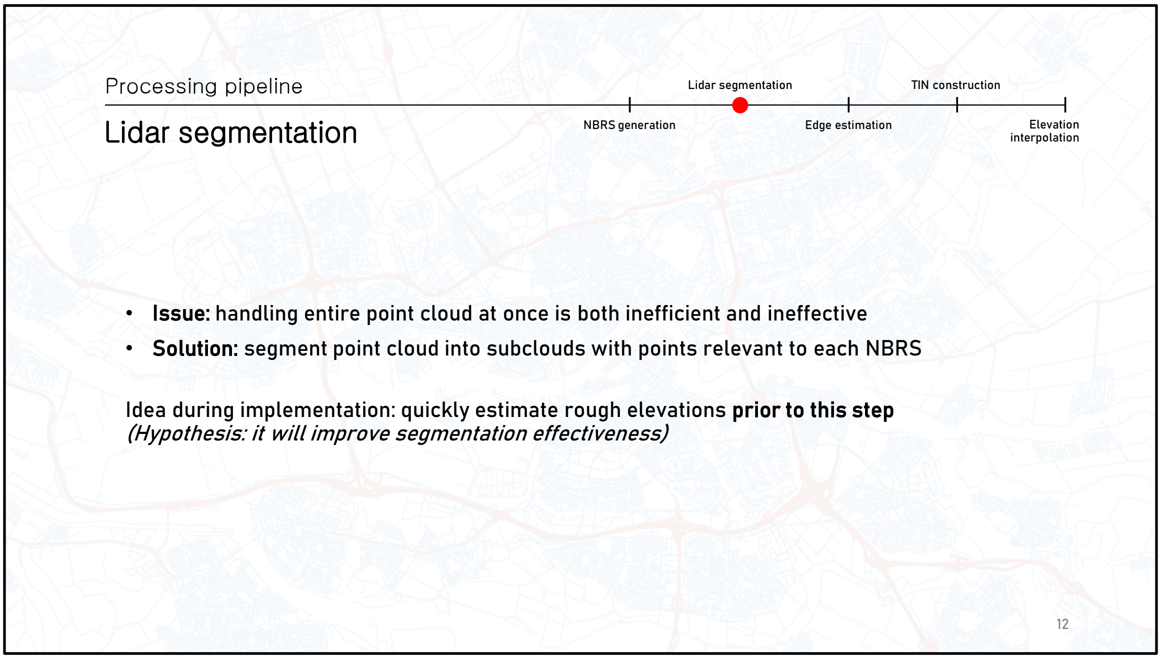
- Highlighted the relevant part of the main research question



- First step of the pipeline: **NBRS generation**
- Subdivide NWB into well-behaved segments, a prerequisite for various subsequent steps
  - I call these Non-Branching Road Segments or NBRSs for short

- How exactly do we generate NBRSs?
  - NBRS are assembled from connected *wegvakken* (the building blocks of NWB)
  - The algorithms try to maximise length and minimise internal angles
  - Self-intersections are not allowed – 2.5D modelling is made possible

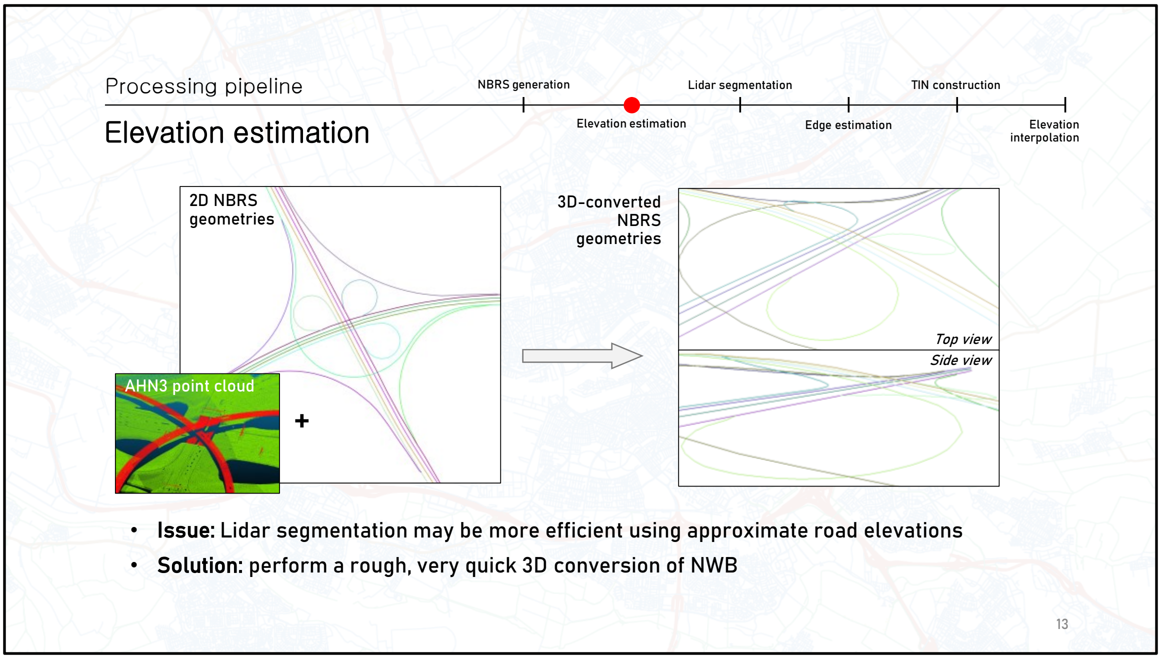
- Implemented two algorithms
  - "Geometric" algorithm relies only on NWB geometry
  - "Semantic" algorithm uses attribute data too, but is thus specialised to NWB



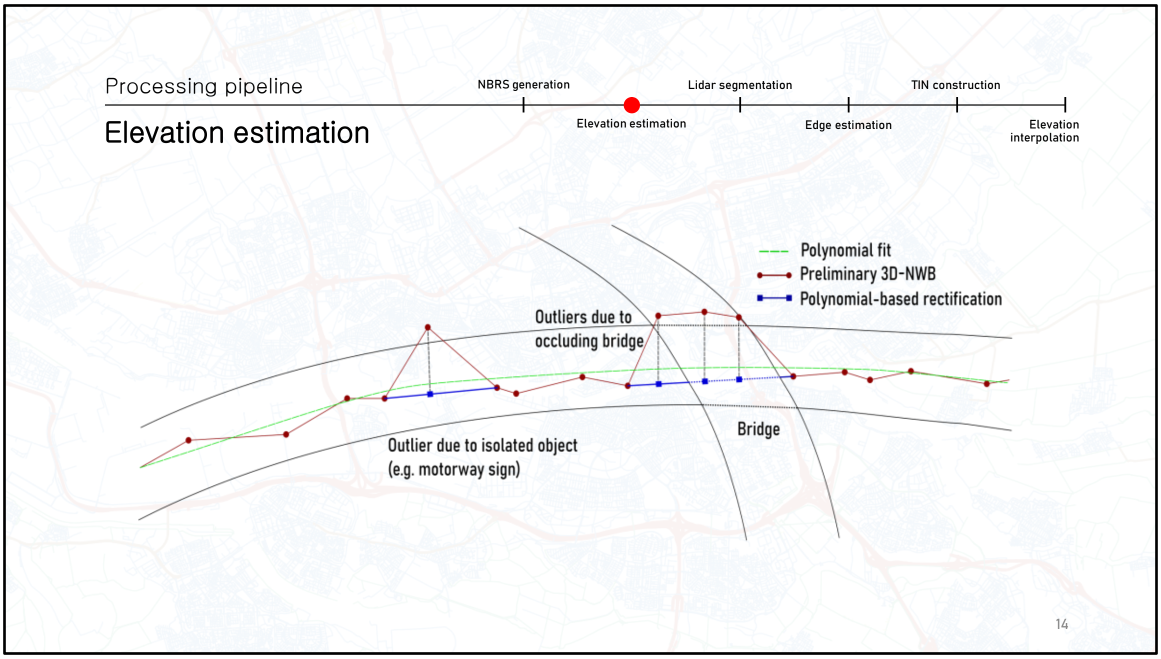
- What is Lidar segmentation?
  - In our case, it means the selection and grouping of Lidar points…
  - …that are relevant to the underlying road surface of each NBRS

- Why do we need Lidar segmentation?
  - In short: to improve performance and make subsequent steps more effective

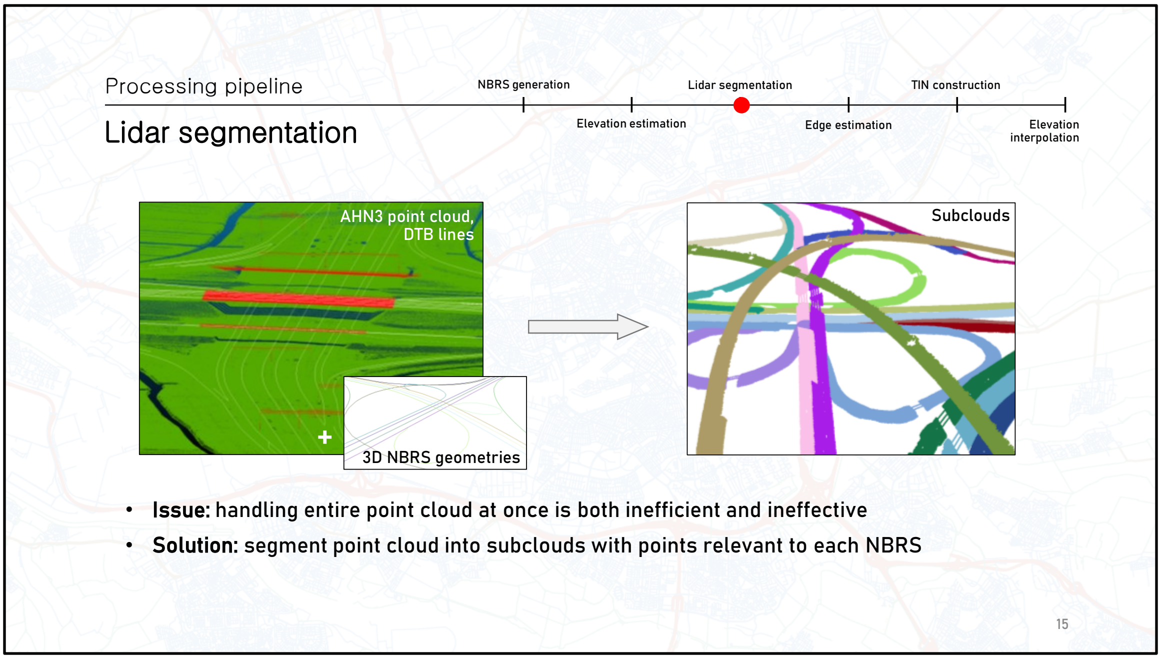
- Starting from a 2D road network limits effectiveness → try to first create a rough 3D conversion of NWB



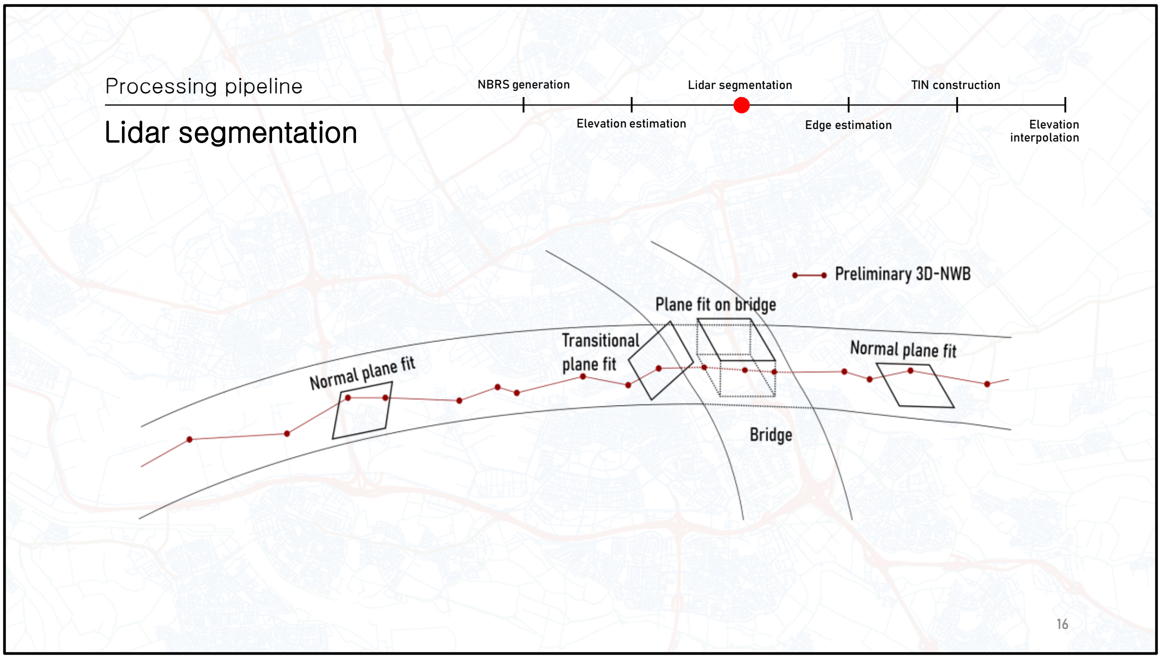
- Added a step to the pipeline: **elevation estimation**
- The goal is to extract elevations for NWB from AHN3
    - As quickly as possible
    - And keeping the algorithm relatively simple
- Really just to provide a starting point for later steps



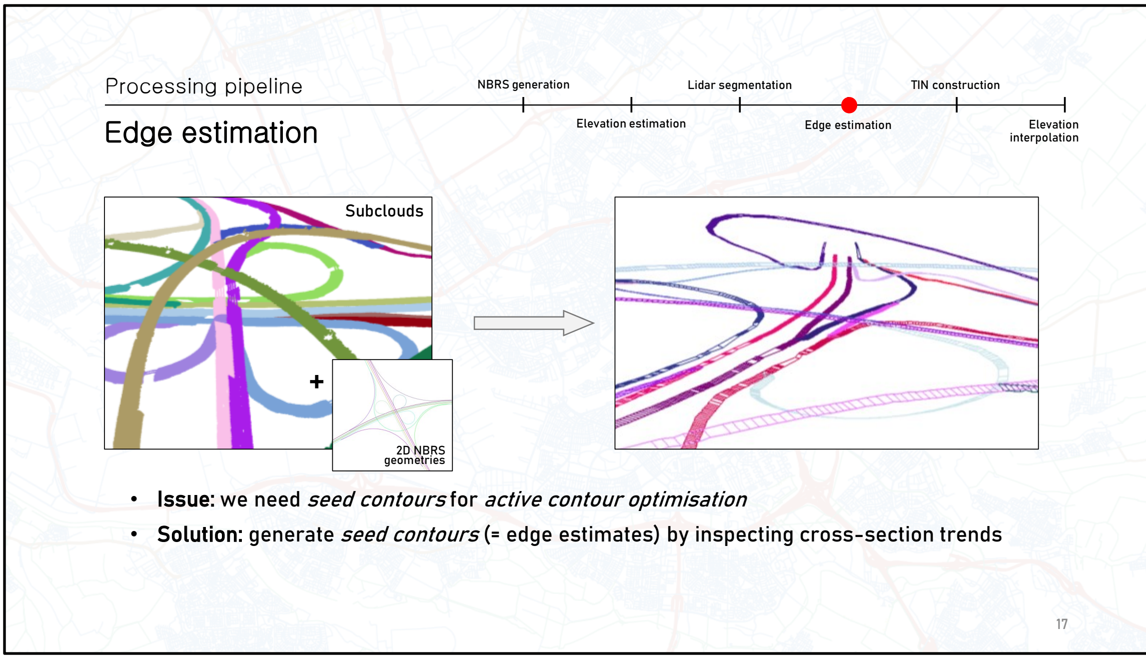
- The implemented workflow is simple:
  1. Densify vertices where necessary
     - It means the addition of vertices where line segments are very long
     - Allows us to better sample the elevation data
  2. At each NWB vertex, query AHN3 for nearby points and derive an elevation value
     - Red vertices/lines in figure
  3. Fit polynomials on the 2D profiles represented by each NBRS to identify outliers
     - Illustrated by green line
  4. Interpolate new values for the outliers via linear interpolation
     - Blue vertices and lines
- The results are surprisingly accurate wherever the elevation profiles of the roads are relatively simple



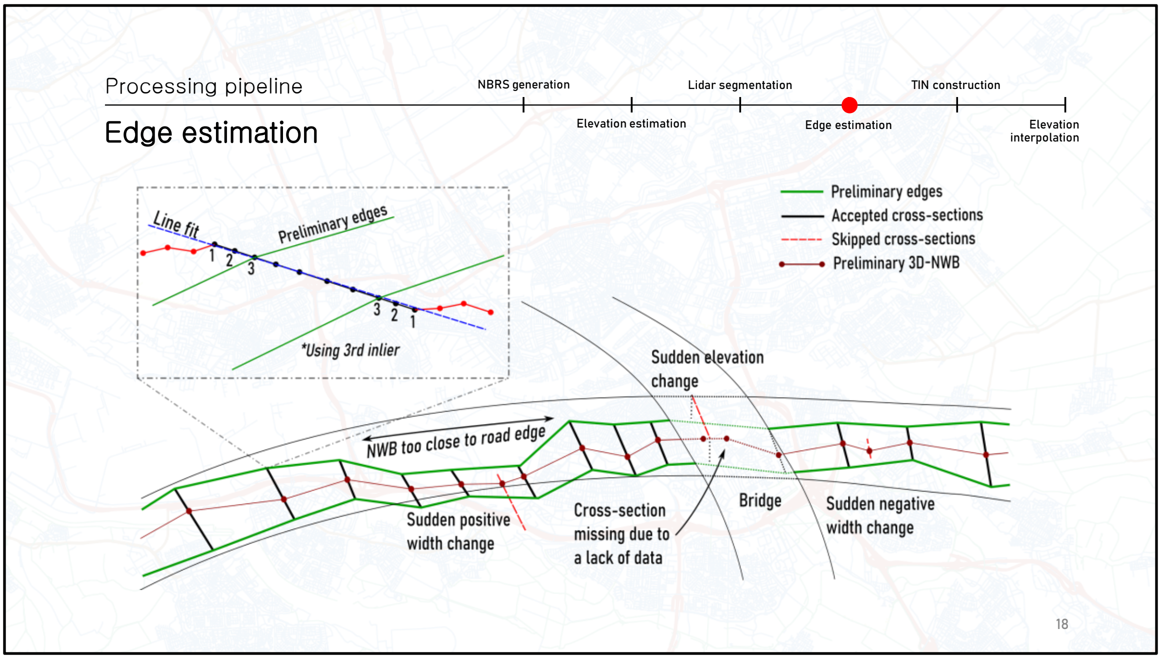
- Next step in pipeline: **Lidar segmentation**
- The preliminary elevations proved to indeed be useful
  - Thanks to them, less irrelevant Lidar points are examined in complex areas
- DTB is used in conjunction with AHN3
  - It is used both as an elevation source, and as reference (like the 3D centrelines)



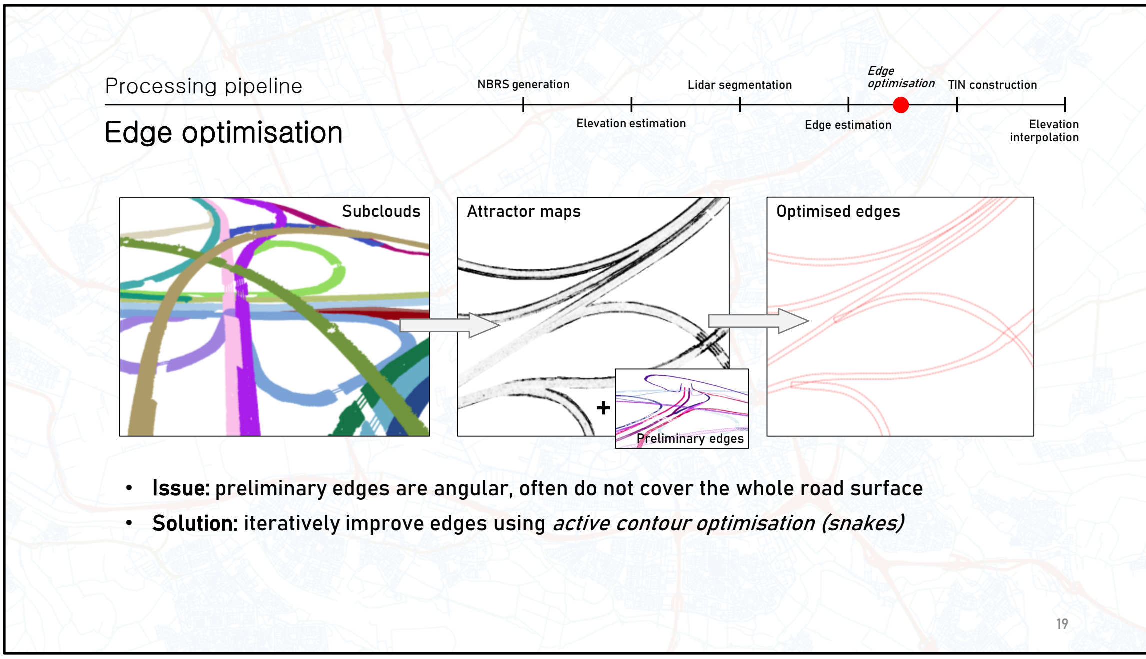
- Workflow:
    1. Fit planes on point cloud points close to each NBRS vertex
        - 3D squares and red vertices/lines in illustration, respectively
    2. For each vertex, pre-select close-by points that conform with local plane fit
        - Not shown here
    3. Progressing along NBRS vertices, detect shifts in the planes
        - Such as the transitional plane illustrated here
    4. If DTB can be used to disambiguate, make use of it
        - Such as underneath the bridge here
        - Also use its (densified) vertices where reliable
    5. Accept planes (and conformant points) which agree with the global trend of the NBRS
        - → Assemble the subclouds
- Detecting transitional plane fits was key, and also the most challenging task



- Next step in pipeline: **edge approximation**
- We are looking for 3D lines that *approximately* delineate the smooth, traffic-occupied parts of roads.
    - These were originally going to be optimised in the next pipeline step
- Key part of original plan to accurately classify road surface reflections

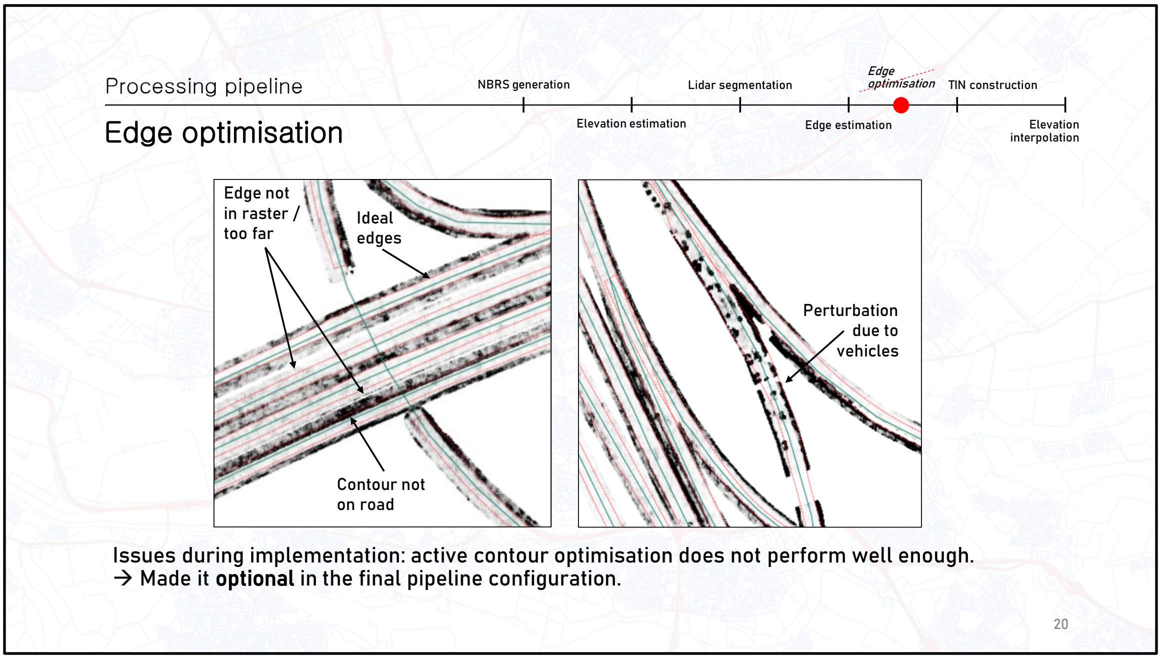


- Workflow:
  1. Construct cross-sections on NBRS vertices and densify them
     - Black vertices and lines in inset
  2. Compute elevations for them from nearby part of relevant subcloud
     - Not shown here
  3. Select suitable edge points on both sides of NBRS from inliers based on line fit
     - Dashed blue line
  4. Discard non-conformant cross-sections
     - Dashed red lines
  5. Assemble preliminary edges of NBRS from these discrete edge points
     - Green lines
- Dealing with NWB getting close to road edges was the most challenging aspect
  - Shown in illustration
  - This is due to NWB's coarse/inaccurate 2D georeferencing

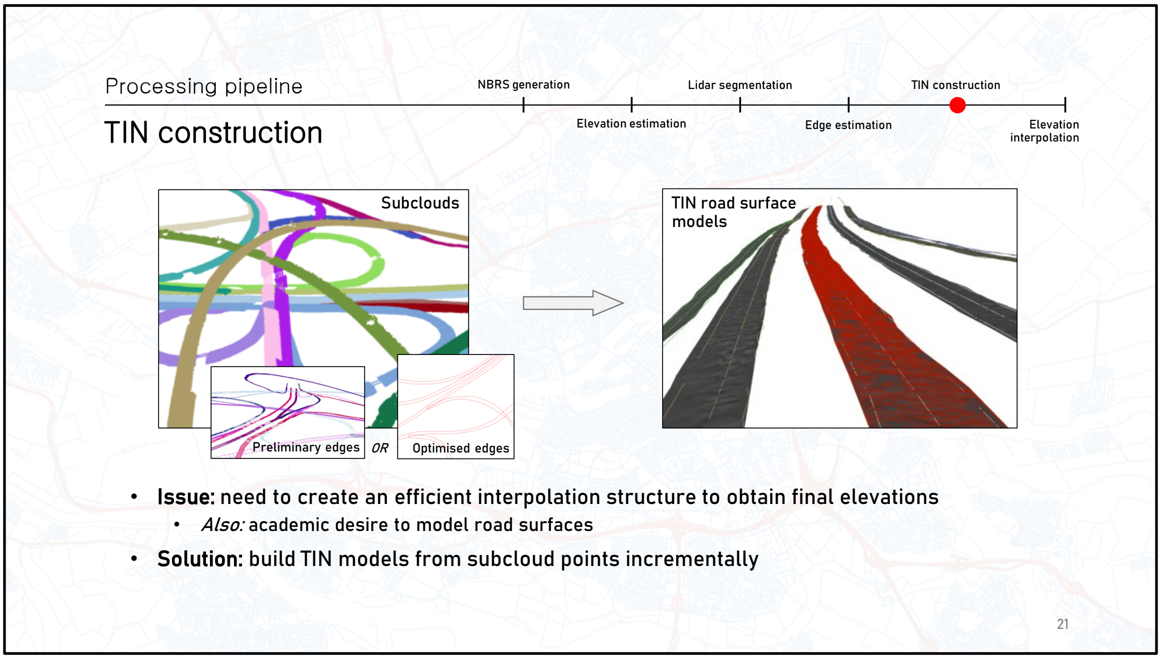


- Next step in pipeline: **edge optimisation**
- This was a core part of the original pipeline design
    - The original intention was to generate rough preliminary edges…
    - …then optimise them using active contour optimisation…
    - …to thus be able to select road surface reflections with near-perfect accuracy

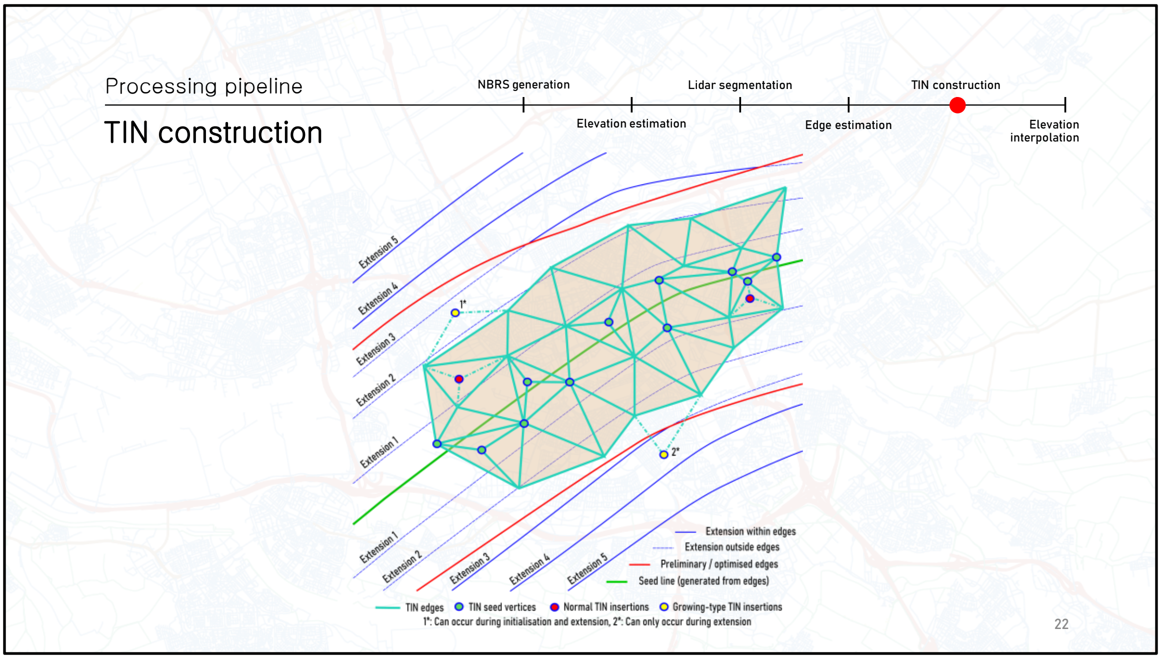
- Workflow:
    1. Construct attractor maps for each NBRS
        - They are rasters that describe how smooth the Lidar-defined surface is locally
    2. Run active contour optimisation for each NBRS, inputting the attractor maps and preliminary edges



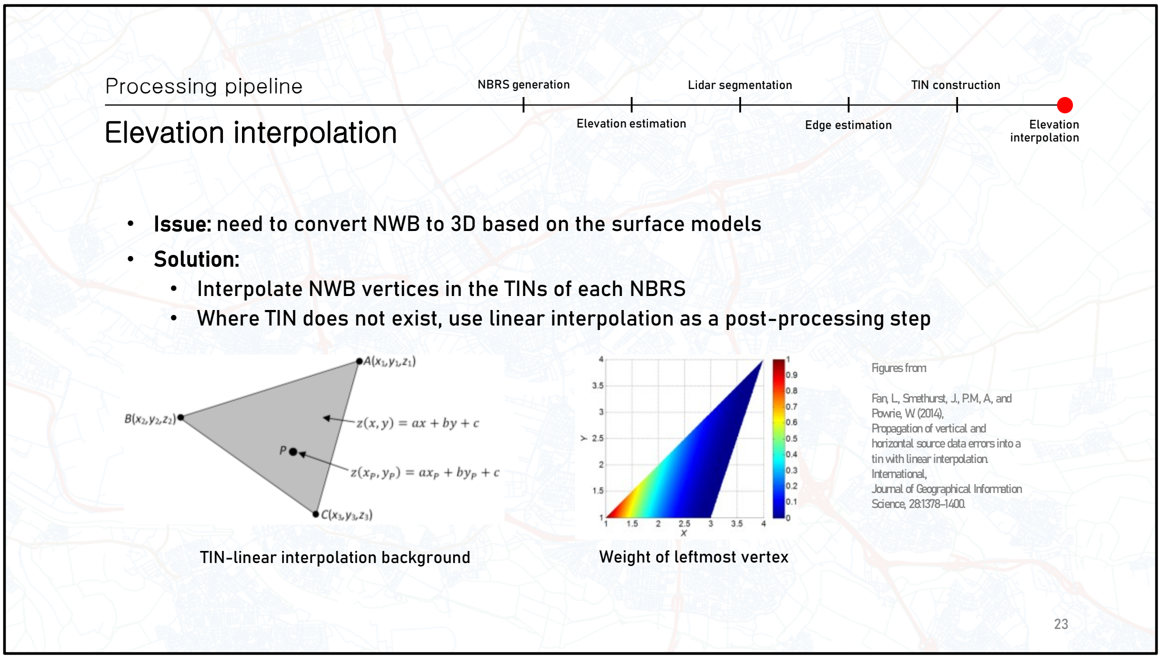
- This did not pan out as well as I was hoping
- Using conventional active contour optimisation, some edges are drawn too far inwards or outwards
- Imperfect attractor maps result in disruptive artefacts
- Difficult to find a parametrisation that works in all possible scenarios
- I only found implementations of basic active contour optimisation, not advanced ones
- I eventually decided to phase out edge optimisation (made it optional, it can still be used)
  - Research timeframe did not allow me to spend more time on this
  - Required significant modifications to the edge estimation and TIN construction steps



- Next step in pipeline: **TIN construction**
- Need to create an interpolation structure from where elevations for NWB can be extracted effectively
    - Also want to model road surfaces – why not do both in a single step?
- Original intention was to just construct a CDT using the optimised edges and the Lidar points between them
- After phasing out edge optimisation, it was necessary to do this differently
    - The TINs are now constructed incrementally, via a surface growing process



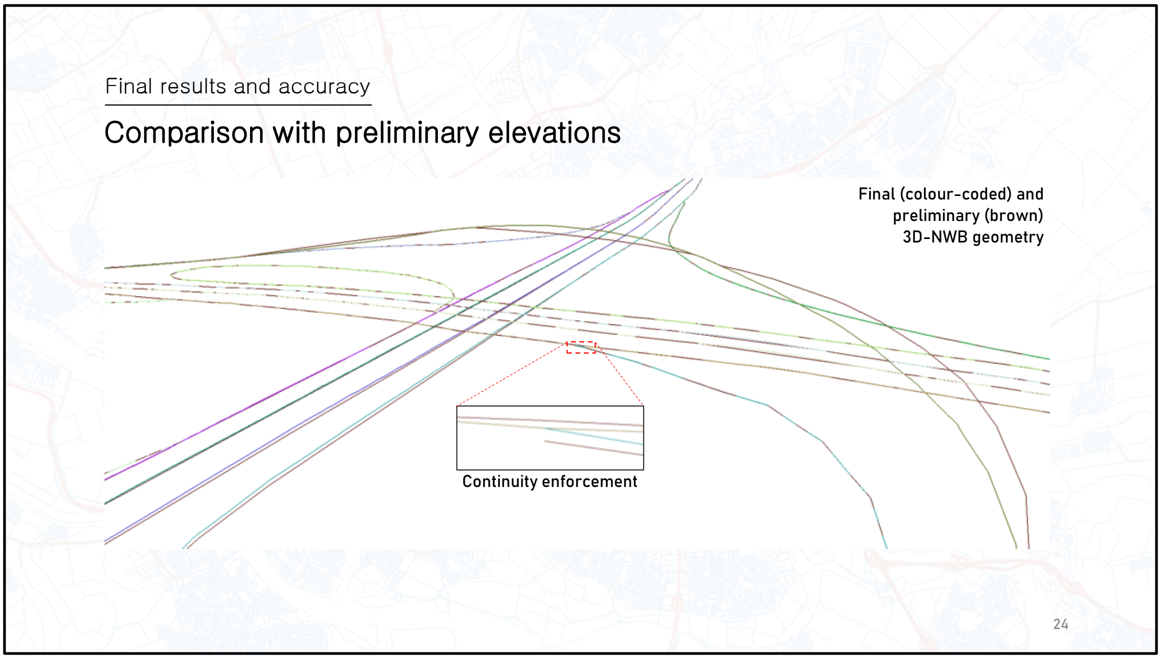
- Workflow:
  1. Seed the TIN using points close to the skeleton of road edges
     - Green line and green points, edges shown as red lines
  2. Grow the TIN *within the edges* using an algorithm inspired by TIN-based ground filtering workflows
     - Normal conditional insertions are shown as red points, insertions take place if point conforms with local triangle
     - Growing-type insertions are shown as yellow points, these inspect the local planar trend in a larger area
  3. Extend the TIN beyond the edges using targeted conditional insertions between extension boundaries (optional)
     - For instance, in extension 3, Lidar points between the blue lines labelled “Extension 3” and “Extension 4” would be considered
     - Can help fill areas missed by the previous step, such as hard shoulders



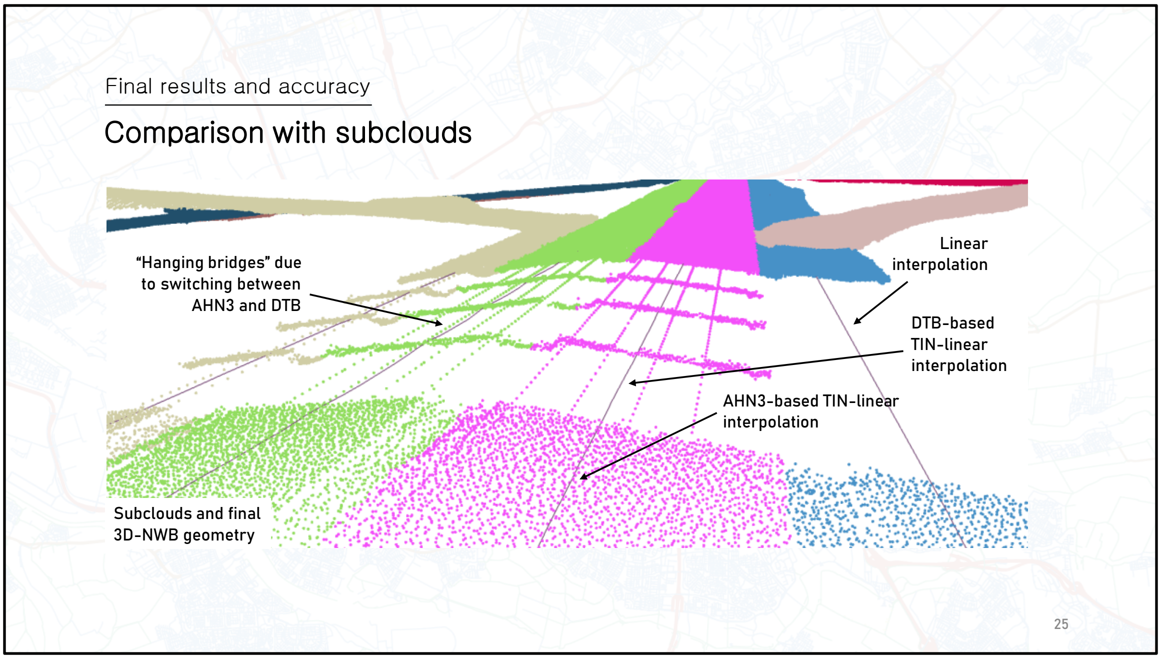
- Next step in pipeline: **elevation interpolation**

- Workflow:
  1. Interpolate NBRS vertex elevations in corresponding TINs
  2. Take care to ensure continuity in intersections across NBRS
  3. Interpolate missing values linearly where the TINs do not exist locally
     - As a post-processing step

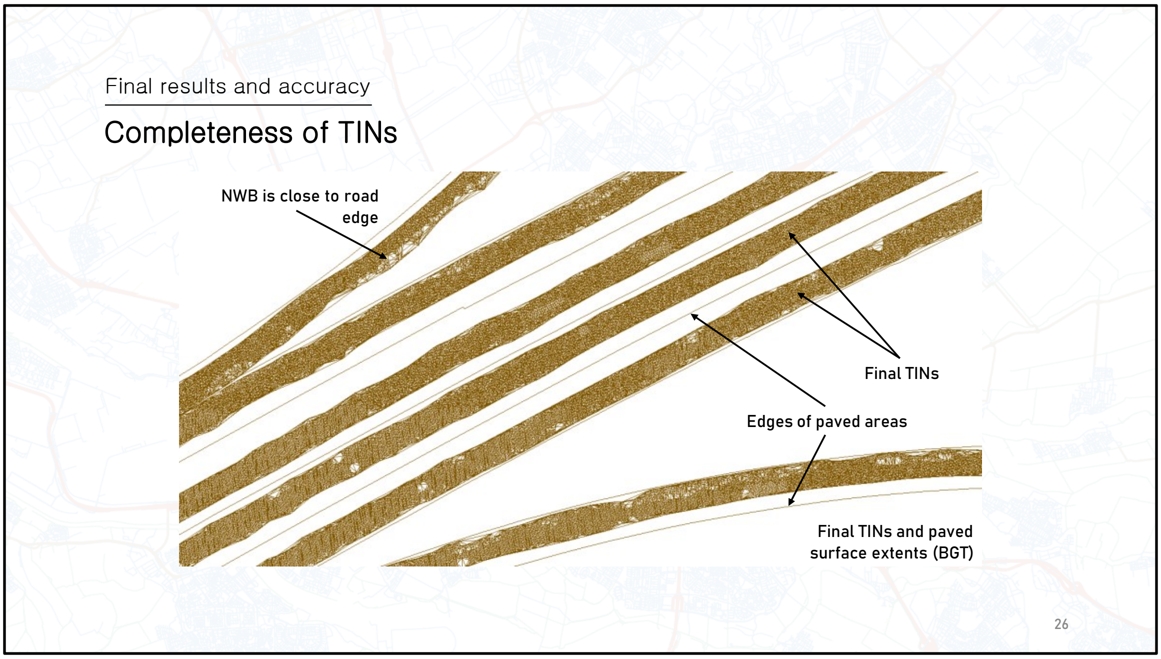
- Using the TIN-linear interpolator
  - The points are projected onto the triangle's surface, in which they are found
- The weight of each triangle vertex decreases away from it, as shown in the figure
- The output file is identical to the input in every aspect other than the added Z coordinates



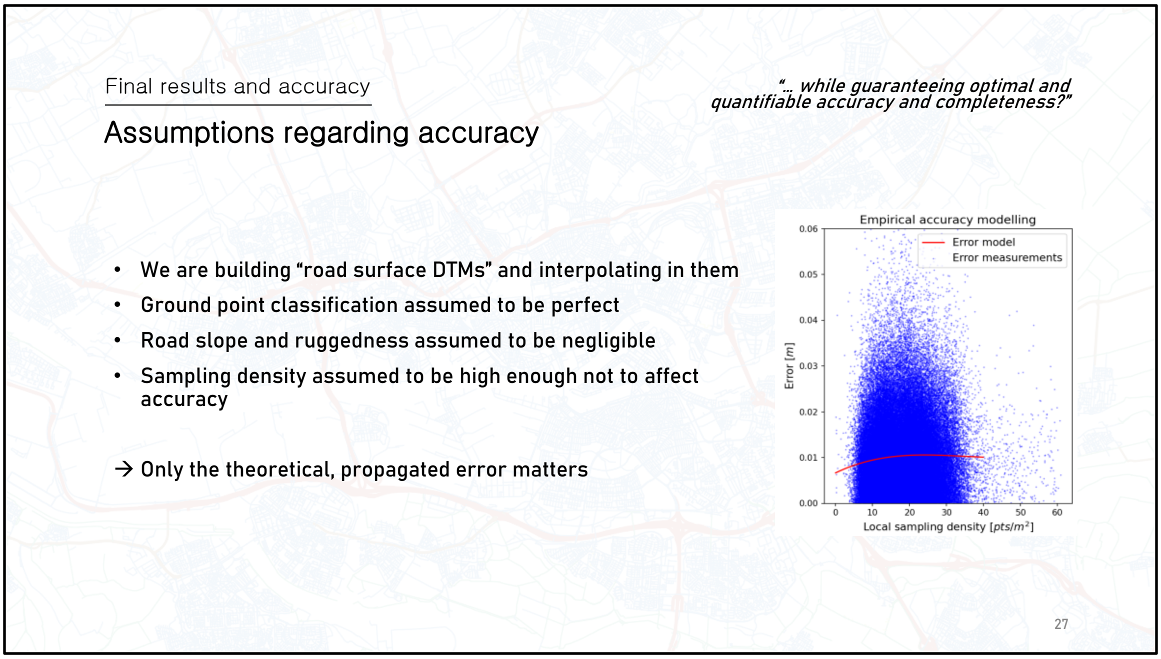
- The final elevations represent an improvement relative to the preliminary 3D geometries
    - The final 3D centrelines are more realistic where curvature is complex
    - Continuity is now being enforced
- Less smooth, because each elevation is based on exactly three Lidar points, rather than a patch



- Comparison of the final 3D-NWB geometries and the subclouds
- Switching between AHN3 and DTB can be spotted
    - Even where very little difference exists between two datasets
- Where AHN3 data is available, the final 3D-NWB geometry is fully defined by it
- Linear interpolation reasonable where no DTB data is available and the gaps are small



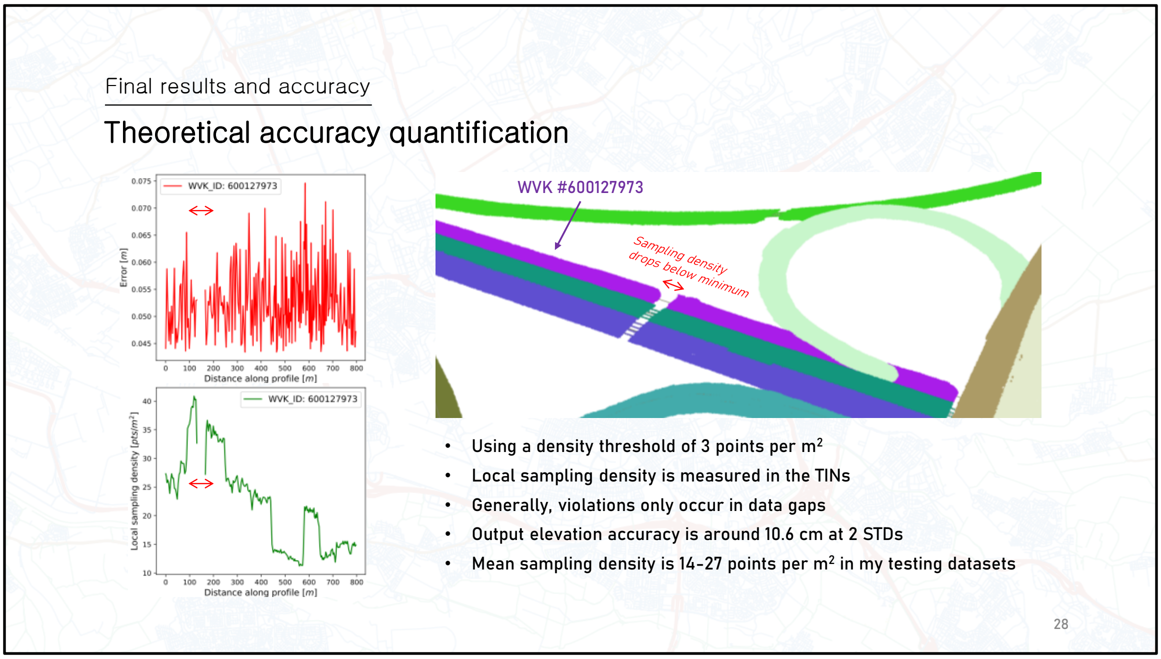
- Current pipeline does not allow the paved surfaces to be fully modelled
  - This was my original goal
- Approximately 75% of the paved surfaces are modelled
  - Here this is slightly lower, as motorways are very wide
  - For provincial roads it can be above 90%
- Traffic-occupied surfaces are fully covered under normal circumstances
- This is due to the combined limitations of the edge estimation and TIN construction algorithms
  - Could be overcome with further work
  - Is a result of phasing out edge optimisation (original assumption: edges will be near-perfect)



- Ultimate assumption is: we are effectively constructing DTMs from Lidar data
- Lidar data has been ground filtered nearly perfectly
- Road slope and ruggedness assumed to be negligible (smooth, relatively flat surfaces)
- Sampling density is high enough to describe the surfaces well
    - This is very much the case in general, as diagram shows

→ Output elevation accuracy can be determined via theoretical error propagation alone

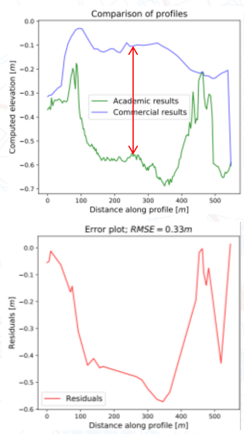
- Highlighted the relevant part of the main research question

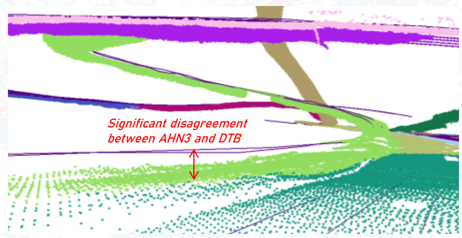


- Theoretical error in TIN-linear interpolation roughly constant
    - Because of the absence of steeply sloping triangles
    - Output elevation accuracy thus constant, and fully determined by input accuracy
- Sampling density in TINs remains high almost everywhere, where there is AHN3 coverage

Comparison with commercial results

DTB-related differences



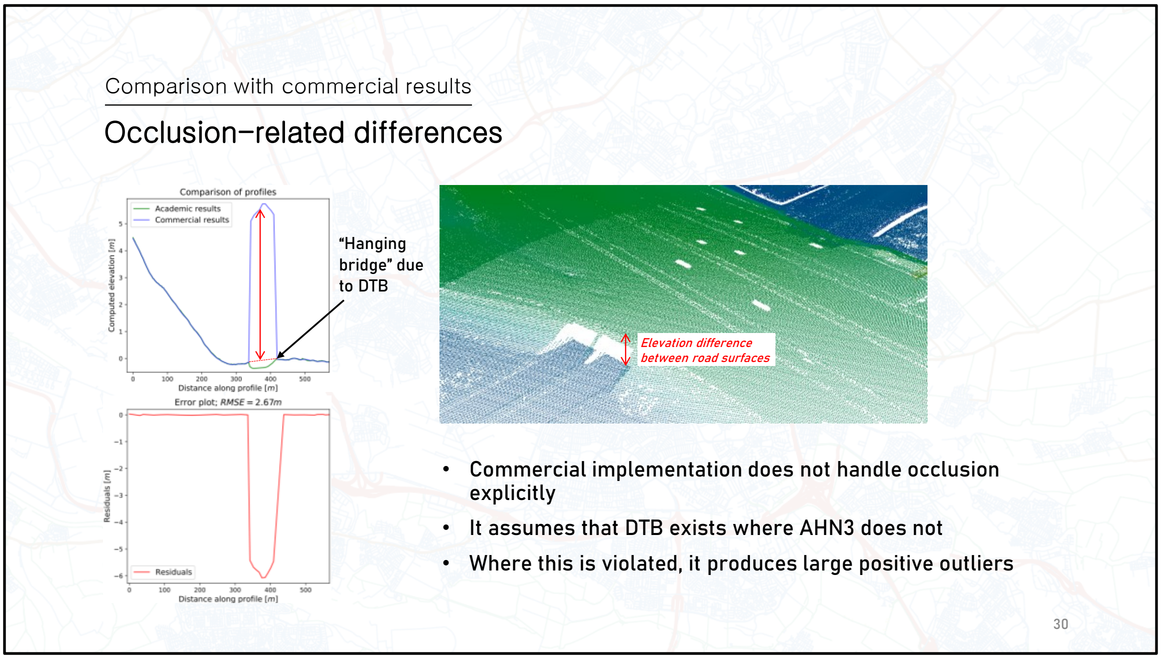


- Commercial implementation prioritises DTB unconditionally
- However, outdated elevation measurements are common in DTB
  - In part, can be explained with subsidence
- Differences in the range of 0.1 to 1 m are found in my testing datasets
- Observable in comparisons wherever commercial program uses DTB

29

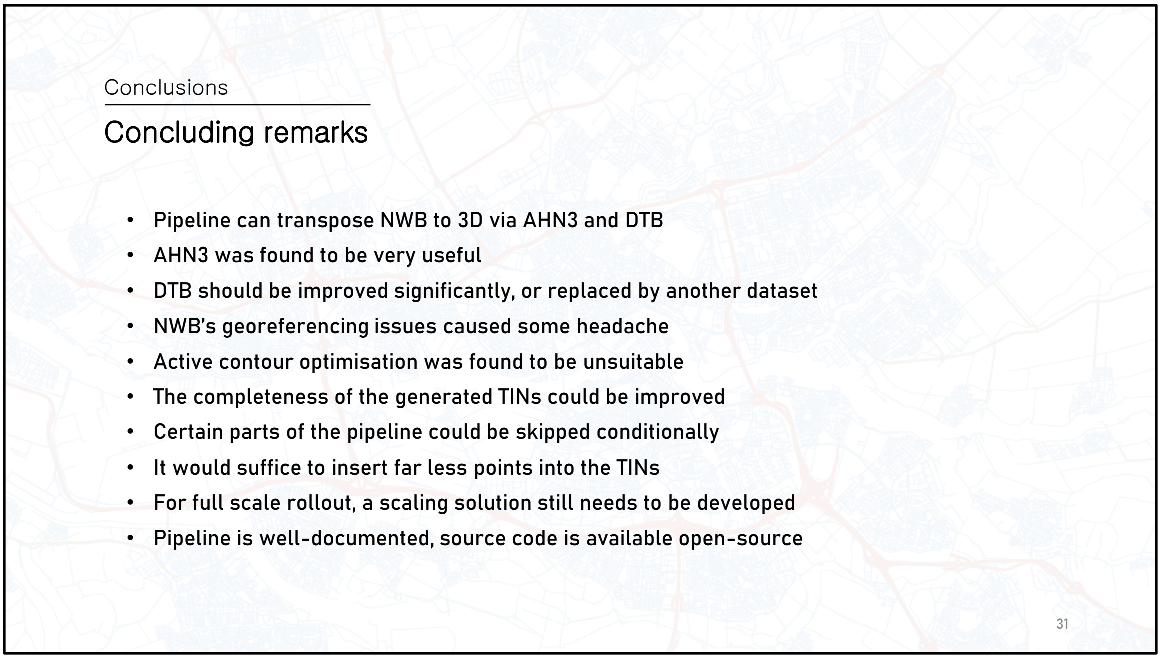
- Moving on to the comparison with the commercial results
- The primary source of disagreement is related to DTB
  - DTB contains lots of outdated elevation data
  - The commercial implementation always prioritises DTB in the case of motorways
  - These differences are carried over directly into the two sets of results
- Hypothesis is that this is ultimately due to subsidence
  - Other factors may also be partly responsible
- RMSE values between 1 to 15 cm where agreement is good
- RMSE values above 50 cm common where DTB is very outdated

- Commercial solution makes better use of DTB, because it is specialised to it

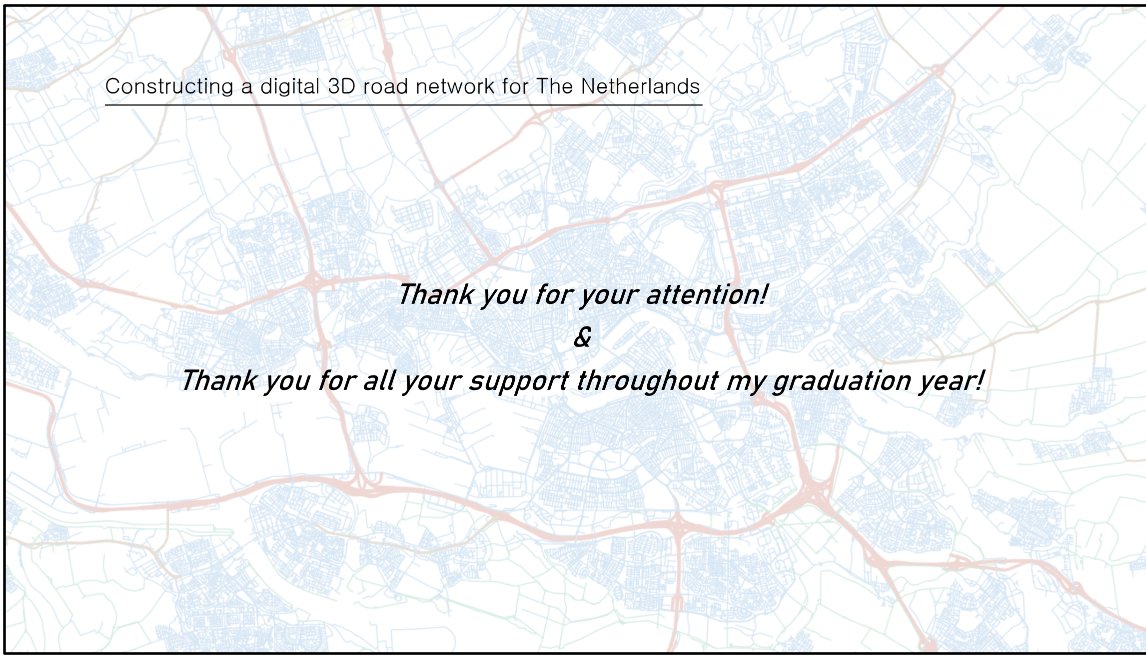


- Commercial implementation has no explicit means of dealing with occlusion
- Elevations are grossly overestimated where there is occlusion…
    - …wherever DTB is missing on motorways
    - …always in the case of provincial roads
- As shown, the profile simply jumps to the elevation of the bridge
    - This is because here the commercial implementation interpolates in infilled AHN3 DTM gaps
- The academic pipeline deals with these issues effectively, even where DTB is absent

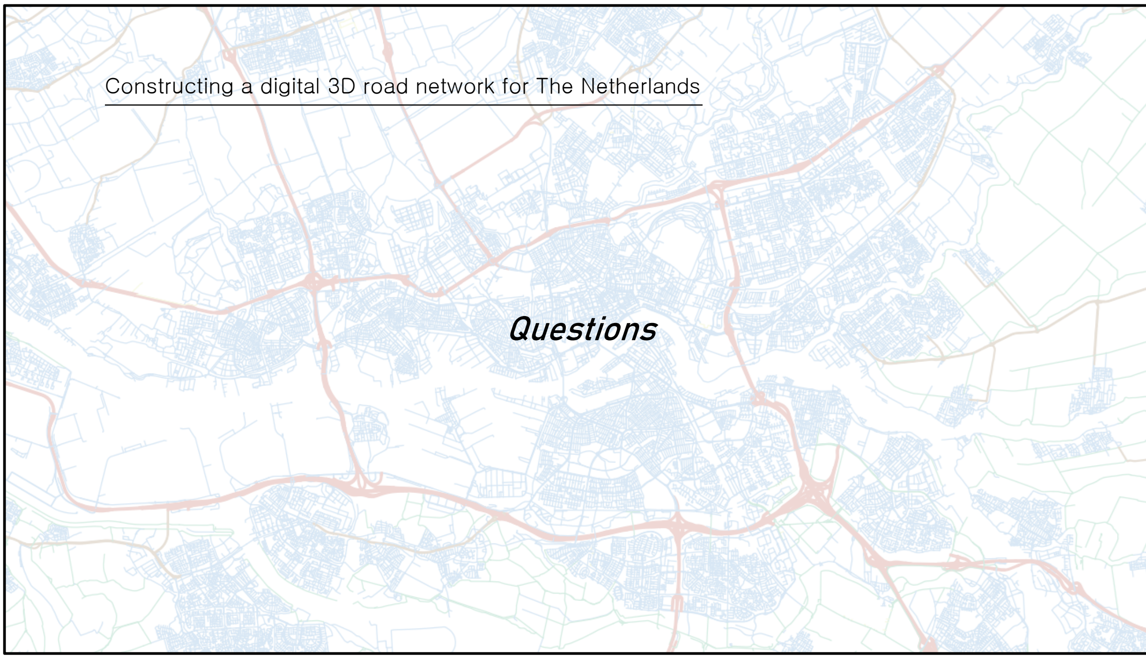
- Commercial solution works without a scaling solution



- Pipeline can transpose NWB to 3D via AHN3 and DTB
  - 5-10% interpolated linearly, another few percent via DTB
  - Rest can be regarded accurate, compliant with noise regulations (< 20 cm at 2 STDs)
- AHN3 was found to be suitable for this project, but not DTB
  - DTB's issues should be resolved, or better still, it could be replaced with an MLS dataset
- NWB's georeferencing issues caused some headache, it should be improved
  - NDW is already working on it, my thesis contains a comparison
- (Basic) active contour optimisation is not suitable to this application
  - And is not recommended for future work
- The completeness of the generated TINs could be improved via additional work
- Certain parts of the pipeline could be skipped conditionally, where they are not necessary
  - I.e. were preliminary elevations are accurate enough
  - This could improve runtimes considerably
- Less points should be inserted into the TINs
  - Somewhere slightly above the minimum sampling density would suffice
- For full scale rollout, a scaling solution still needs to be developed
- Pipeline is well-documented, source code of the implementation available open-source



Thank you for your attention, and I would like to thank my mentors and family for their help and support throughout my graduation year.



We can now move on to any questions you may have.