

DELFT UNIVERSITY OF TECHNOLOGY
FACULTY OF ARCHITECTURE

SYNTHESIS PROJECT
GEO1101

Integrate and store 3D roads and terrain for navigation purposes

Final Report

Authors:

Dimitris Mouzakidis (5849993)

Puti Nabila Riyadi (5771633)

Georgios Konstantinos (Akis) Nestoras (5863686)

Longxiang Xu (5722918)

Na Liu (5827108)

Stein Köbben (4886488)

4th November 2023

Abstract

Modern navigation heavily relies on Global Navigation Satellite Systems (GNSS) and digitized road network databases, but faces limitations in GNSS-denied areas and complex 2D road networks. This project addresses these challenges by developing a methodology to create and store a comprehensive 3D road and terrain dataset for enhanced navigation. In collaboration with TomTom, a company that aims to fulfill software requirements, making significant advancements in geolocation technology and societal contributions. The main research question of the project is: "How can we create a 3D map of roads using information about the center of the road and elevation data?". The approach to answer this question involves extracting 2D road polygons from centerline data based on width of the roads, the direction and the amount of lanes of them. These 2D polygons undergo enrichment with elevation data, with techniques like filtering, segmentation, and primitive extraction ensuring alignment with the digital terrain model. The methodology encompasses data acquisition, creation of polygons using the centerlines dataset, 2D-to-3D polygon conversion, elevation integration and data storage in CityJSON format.

Preface

The Geomatics Synthesis Project (GEO1101) is a 10-week collaborative course designed for students in the fifth semester of the Master of Science program in Geomatics for the Faculty of Architecture and the Built Environment at Delft University of Technology. This initiative enables students to apply the knowledge they've acquired during their first year of master's studies. Moreover, it provides an opportunity for students to engage with real stakeholders and clients, making it a valuable exercise in project management. The primary aim of the final report is to document the entire process, including problem identification, formulation of research questions, data collection, data analysis, data modeling, and data dissemination. The purpose of this report is to offer a broad summary to the report's readers. Creating a 3D representation of road networks is necessary to provide a more realistic view of road systems, aiding in navigation, safety, urban planning, and decision-making. It also allows for a detailed analysis of critical elements like intersections and structures, enhancing our understanding of road infrastructure and its interactions with the surrounding environment.

Acknowledgements

We are sincerely grateful to Stelios Vitalis, representing TomTom, for his invaluable support, guidance, and extensive knowledge. He has been a gracious host and a driving force behind the project's success. We also appreciate Dr. Ken Arroyo Ohori for his valuable insights and dedication, which enriched our project a lot. We also extend our thanks to Jantien Stoter for her contributions to the project.

Concluding, as a student group we experienced this project as a tremendously valuable and positive experience. Despite the challenges, decision-making points, and various tasks along the way, we are proud to have reached highly promising end results. We would like to express our deep appreciation to TomTom for their collaboration and support, as their involvement has played a crucial role in advancing the importance and impact of our work. We look forward to the continued relevance and value our contributions bring to the field.

Contents

1	Introduction	7
2	Problem Definition	9
2.1	Research Question	9
2.2	Project Requirements	9
3	Methodology	10
3.1	Extracting Polygons from Centerlines	10
3.1.1	Assigning Width to Centerlines	10
3.1.2	Converting centerlines into polygons	10
3.1.3	Junctions and Road Connectors	10
3.1.4	Cleaning up Polygons	11
3.1.5	Marking Overlapping Polygons	11
3.1.6	Storing the Polygon Data	12
3.2	Assigning Elevation to polygons	14
3.2.1	Inferring Polygon Elevation	14
3.2.2	Assigning Vertices Elevation	15
3.3	A Triangulation Method for Polygon and Vertices elevation assignment	15
3.3.1	Terrain	16
3.3.2	Output	16
4	Results	18
4.1	Results	18
4.1.1	Stage 1 Results	18
4.1.2	Stage 2 Results	21
4.2	Evaluation of Results	27
4.2.1	Evaluation of Polygons	27
4.2.2	Evaluation of Elevation	29
4.3	Evaluation of the Methods	30
4.3.1	Evaluation of Stage 1 Methods	30
4.3.2	Evaluation of Stage 2 Methods	31
5	Conclusion	32
5.1	Requirements	32
6	Further Development	34
6.1	Alternative Approach in Managing Polygon Topology	34
6.2	Obtain the Road Width from Other Data	34
6.3	Enrich the Attributes of the Junctions	34
6.4	Scale up to Large Dataset	34
6.5	Different Treatment of Some of the Overlapping Polygons	35
6.6	Lightweight Terrain Integration	35
6.7	Using Distance-based Weighting Instead of "Topology" Steps for Assigning Polygon Elevation	35
6.8	Triangulation Refinement for Elevation Assignment	35
A	Project Organisation	36
A.1	Team Members	36
A.2	Responsibilities of Team Members	37
A.3	Client	37
A.4	Responsibilities of Client	37
A.5	Supervisors	37
A.6	List of needs	37
A.7	Communication plan	38
A.8	Planning	38
A.9	Risk Overview	39

List of Figures

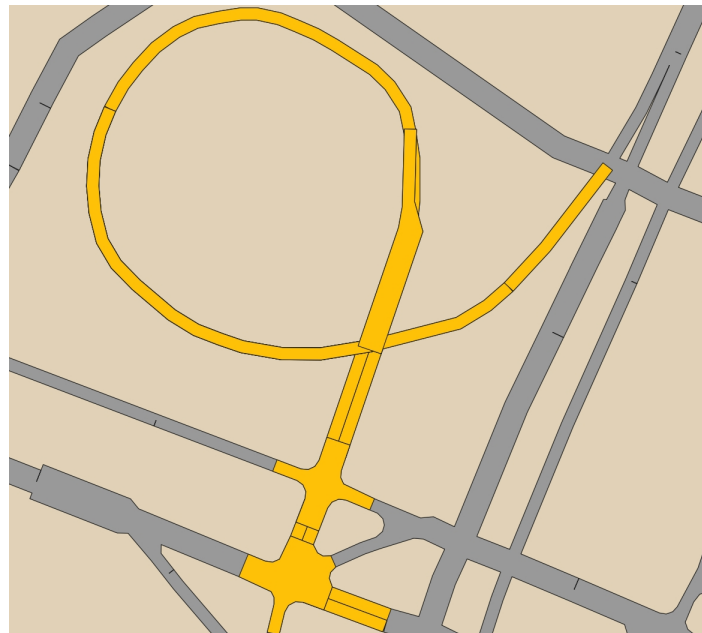
1	An example comparison of Road Overpass with 2D and 3D maps	7
2	Before and after performing clip operation to polygons near junction	11
3	Overlapping polygons	12
4	Pipeline of stage 1	13
5	Example of elevation interpolation of a polygon	15
6	An example illustrating the result of different strategies for assigning vertices elevation	16
7	Example of non-planar road polygons	16
8	Pipeline of stage 2	17
9	The result of the test case (Aerial image from Kadaster [1])	18
10	Visualisation of part of the road network. On the left side a junction is appeared and on the right a road connector is shown with green color.	19
12	Junctions	20
14	The Overpasses test case (Aerial image from Kadaster [1])	22
15	The Residential area test case (Aerial image from Kadaster [1])	23
16	The Hills test case (Aerial image from Kadaster [1])	24
17	The Overpasses test case results	25
18	The Residential area test case results	26
19	The Hills test case results	27
20	Gap in road turn	27
21	Dangling polygon	28
22	Road Connector is missing	28
24	Overpasses test case on Google Earth (a) [2] and results (b) in the same location	29
25	Hills test case on Google Earth (a) [2] and results (b) in the same location	29
26	Gantt diagram of the project planning	39

Abbreviations

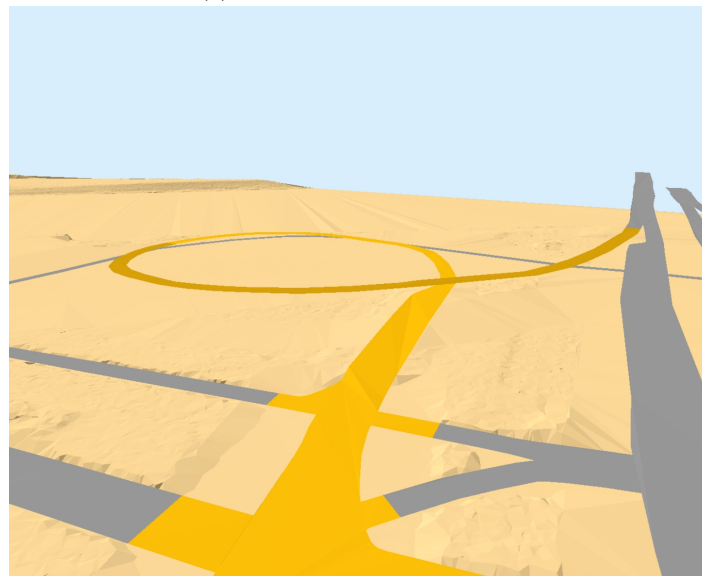
AHN	Actueel Hoogtebestand Nederland
OSM	Open Street Map
2D	2-dimensional
3D	3-dimensional
PID	Project Identification Document
TIN	Triangulated Irregular Network
DTM	Digital Terrain Model
GNSS	Global Navigation Satellite Systems
LiDAR	Light Detection and Ranging

1 Introduction

3D road data underpins immersive navigation. In modern context, navigation typically involves positioning and route directing for users to move from one place to another. Enhanced by the Global Navigation Satellite Systems (GNSS) and digitized road network databases, real-time navigation for users is now achievable. However, there are certain limitations to this experience. First, the existing positioning methods heavily rely on GNSS signals. This means current system cannot operate in GNSS-signal denied places such as tunnels or urban canyons. Additionally, the intricate modern road network challenges 2D navigation. For instance, in regions with merging, diverging, or layered roads, users might find it puzzling to identify which layer or path they are on. Thus, there's a pressing need for a 3D map to ensure a more seamless navigation experience. By visualizing 3D roads and terrains, users can swiftly determine their location and direction by comparing it to the digital representation of their surroundings. Figure 1 depicts an example of navigating road overpasses. The comparison between 2D and 3D representations clearly highlights the superiority of the 3D map.



(a) Road Overpass in 2D map



(b) Road Overpass in 3D map

Figure 1: An example comparison of Road Overpass with 2D and 3D maps

This project aims to develop a method for creating and storing a 3D road and terrain dataset for navigation. The starting point was standard 2D road centerline data from which 2D polygons were extracted based on road width and lane count. These 2D polygons were then enriched with elevation data. Techniques like filtering, segmentation, and primitive

extraction were employed to align the 2D road polygon with the Digital Terrain Model. Finally, the 3D road dataset was combined with terrain data to create a comprehensive 3D dataset optimized for display and navigation guidance. Furthermore, efficient storage solutions for this 3D dataset were explored.

The project was carried out with TomTom, a world-leading geolocation technology company which aims at "building the smartest, most useful map on the planet to help people find their way" [3]. The project's aim was to advance the company closer to its vision while also making a societal contribution as Geomatics students.

The scope of the project is made more specific in section 2 by stating the research questions and the requirements. section 3 will then explain how the project was carried out and will provide an overview of the software. The results of this software are shown and discussed in section 4. The most important conclusions of the project are discussed in section 5 and section 6 will explain how the research can be developed further.

2 Problem Definition

2.1 Research Question

The main research question that was answered during this project is

”How can a 3D model of road networks be developed from a center-line dataset and elevation data?”

This project seeks to find the methodologies, tools and best practices required to create a 3D representation of road networks, highlighting critical elements such as road intersections, overpasses, and structures around the main roads. At the same time, it is also important to explore effective methods of storing the 3D representation of the road network. To be able to answer the main research question, the following series of sub-questions need to be answered.

1. How to construct 2D polygons from the center-lines?
2. How to handle the incompatibility of input data that come in different formats?
3. How to store elevation data to the road network?
4. What methods are used to model the roads, road structures, and overpasses in 3D representation?
5. How to handle the topology of a road network in a 3D dataset?
6. How to store the final 3D road network with all the required attributes?

These research questions is addressed using the methodology that will be explained in section 3.

2.2 Project Requirements

To answer these research questions a software product has been made. It is important to make sure the team, the supervisors, and the client have the same expectations of the project. To this end, the requirements for the software product are listed here using the MoSCoW method[4]. This method divides the requirements into four categories. ‘Must’ are requirements that have to be met, otherwise, the project has failed. ‘Should’ are requirements that are important for the project and make the quality of the result a lot better. ‘Could’ are requirements that would be nice to have, if the team has the time to do it, but these are not essential for the project. ‘Will not’ are things that are not included in the project.

Must

- Extract a 2D polygon representation of the roads based on the center-lines dataset.
- Integrate terrain elevation to the road polygons.
- Model and store the terrain around the roads with a 3D representation.
- Handle topology in the 3D dataset correctly.
- Store the 3D representation of road network effectively to use for navigation purposes.
- Store the directions and the lanes of the road.

Should

- Create a 3D representation of overpasses like intersections and overpass bridges.
- Compared the accuracy of resulting 2D polygon with the polygon data provided by Tomtom.

Could

- Model the objects around the main roads (sidewalks, buildings) with a 3D representation.

Will not

- Develop a 3D visualization application that is dedicated to 3D navigation purposes.

3 Methodology

The methodology for this project is divided into two stages which are the construction of 2D road polygons, followed by the development of 3D polygons.

3.1 Extracting Polygons from Centerlines

Establishing a precise and informative 2D road network prior to elevating the roads to their respective heights can significantly streamline the process for creating 3D roads. A well-constructed 2D road network serves as a foundational blueprint, reducing complexities and ensuring a smoother transition to the third dimension for the roads. The task of constructing a 2D road network, as in Figure 4, can be divided into several distinct phases; assigning width to the centerlines, converting the centerlines into road polygons, processing 'special' polygons including junctions and road connectors separately, identifying overlapping polygons, and storing the result.

The constructing process for conventional roads and special roads are separated based to their level of complexity. Junctions are constructed where there are more than two roads connected in the same node. Meanwhile, road connectors are being developed in order to replicate the smooth transition between two roads with differing number of lanes. Due to the construction of junctions and connectors in some areas, overlapping polygons are generated. Consequently, it becomes necessary to initiate a cleaning procedure. The polygons that remain after this cleanup operation serve as indicators for tunnels and bridges, which are crucial for determining their respective elevations. This is the reason why these polygons are specifically marked. Finally, in order to ensure that the outcome can be effectively used for elevation assignment, it should be stored as an array within a specifically designed class.

3.1.1 Assigning Width to Centerlines

The aim of the output product is the first item to consider while creating polygons from road centerlines. The geometry of the centerline itself is not sufficient enough to create a 2D representation of a road that will be used for navigation purposes. One of the main parameters that needs to be computed is the width of the road, which can be associated with the number of lanes in a particular road.

Road Hierarchy The width of the road varies according to the purpose. The Netherlands classifies roads into three categories: access roads, distributor roads, and through-roads. These roads are designed following optimal road design guidelines, allowing for a reasonable assumption of road width based on their respective categories.

Lanes of the Roads To develop a comprehensive 2D road map, accounting for the number of lanes in each road is vital. This information obtained from the road catalog, ensuring an accurate representation of the road network.

3.1.2 Converting centerlines into polygons

A buffer is created extending from the centerline to a distance matching the road width. By integrating this with the lane width and lane count data, multiple polygons can be derived from a single centerline, accurately representing the various lanes of a road.

To ensure smooth road generation and minimize discontinuity, segments belonging to the same road are consolidated into a multilinestring for unified buffering. Following the buffering process, the resulting polygons are segmented based on the initial nodes present in the multilinestring, maintaining accuracy and continuity in the representation of the road network. The nodes that segment a road into sections adhere to specific rules, cutting the road where there are sharp curves to maintain a reasonable level of road detail and continuity, ensuring a smooth representation of the road network.

3.1.3 Junctions and Road Connectors

Junctions have been established using the nodes at the intersections of roads. A line is generated for each road that connects to a node. These lines originate from the node and are extended along the centerline of each road. Consequently, if three roads intersect at a given node, multiple distinct lines are created. The lines are extended to a specific distance based on the classification of the road. Then, these lines are merged resulting to a new multilinestring. This multilinestring is buffered based on the width of the roads that it originally belongs to. Finally, junctions in the road network may have sharp angles or irregular geometries that can make the map or visualization look cluttered or less aesthetically pleasing. To improve the visual representation, the sharp angles in the junction polygons are modified or smoothed.

In road networks, number of lanes can be changed without the presence of junctions or interchanges. To provide smoother visual representation of such changes in consecutive road polygons, the function *Road_Connector* was introduced in the code. This function combines the consecutive road segments that have different number of lanes which may exist and its distance is visualised according to the type of the road so a seamless lane change representation is ensured. The function initially identifies the junction IDs solely connected to two roads to exclude the junctions. It then proceeds to compare the number of lanes on these two roads. In cases where these two roads feature differing lane counts, the function generates a line segment along these roads from the node to which they are both connected. Subsequently, it produces a buffer based on the width of the road. The connectors further refine this process by creating a positive buffer and subsequently

a negative buffer at the same distance to ensure a smoother transition.

The newly created junctions and road connectors initially lack any associated attributes. However, several critical attributes must be added to these special polygons, including 'geometry,' 'ID,' 'METERS,' 'LANES,' 'T_JNCTID,' 'F_JNCTID'. The 'geometry' attribute is designated to the geometry specific to the respective junction or connector. As for the 'ID,' these polygons are assigned unique IDs, where the word 'junction' or 'connector' is followed by an incrementing number as new special polygons are generated. Additionally, the 'lanes' attribute is set to 0 for junctions to prevent ambiguity, particularly when multiple roads converge, each potentially having different lane configurations. The value of the 'meters' attribute is adjusted with the actual length of the polygon. Lastly, the information about the 'to junction ID' and 'from junction ID' are filled using the node ID that is being used to create the junctions and connectors in the first place.

3.1.4 Cleaning up Polygons

Clip and Merge The construction of junctions and road connectors in some places creates some unwanted overlaps with the regular roads. These regular roads need to be cut using the junction polygons in order to have overlap-free roads. This operation is being done using the difference operation of the *overlay()* method from GeoPandas[5] library, which is used to find the geometric difference between two GeoDataFrames. The outcome represents the portion of the regular road polygons that do not intersect with the junction polygons. As a result of the presence of overpasses, bridges, and tunnels, it's not feasible to apply the clip operation to all regular road polygons. Using the clip operation without applying any condition would lead to unintended cuts in cases where a road passes beneath or over a junction. Consequently, the clip operation can only be performed where the junctions and the regular roads to be clipped share the same elevation level. The elevation information is extracted from the centerline dataset using the 'T_ELEV' and 'F_ELEV' columns. Afterwards, the outcome of the initial clip operation undergoes a second round of clipping, following the same procedure. However, this time, the roads are clipped using the road connector polygons. Figure 2 compares the roads before and after clipping process. The result from these two clip operations is ready to be compiled with the junctions and road connectors polygons.



Figure 2: Before and after performing clip operation to polygons near junction

After the polygons are all stored in the same GeoDataFrame, additional cleanup process is implemented to remove unwanted polygons in the end. Some of these are the polygons with areas smaller than a predefined threshold, and they typically result from the clip operations. This situation arises when the junctions do not align perfectly with the regular roads due to the junction smoothing process. Other undesirable polygons are those with duplicate IDs as a result of the clip operation. The road polygons may be divided up into several pieces as the result of clip operation, and these fragments become a newly created polygons with the same ID. The area of each polygon is being considered when dropping duplicate polygons with the same ID. The polygon that needs to be kept is the biggest one, which is assumed to represent the road more compared to the residue pieces produced by clipping.

Finding Adjacent Polygons Identifying the polygons that are in immediate adjacency is crucial for determining the elevation of the vertices. The determination of the originating and terminating polygons is based on the "F_JNCTID" and "T_JNCTID," representing the indices denoting the commencement and conclusion of the centerline and respective polygons. Within our dataset, we record all adjacent polygons in the attributes "from", "to", and "adjacent", which contain the "fid" of each polygon that shares adjacency with the current row's polygon. In instances where polygons serve as junctions, multiple "from" and "to" polygons are possible, and their indices are stored as a list for reference.

3.1.5 Marking Overlapping Polygons

After the cleaning up process is done, the remaining polygons that overlap should indicate a real overlapping road in the real world. This means we can identify which roads are considered bridges, tunnels and overpasses. The initial version of this function looped through the entire polygon dataframe every time it assessed a polygon, leading to a significantly slow runtime. To improve efficiency, the function identifies possible matching polygons that consist of a set of nearby polygons

and then applies the remaining overlapping operations only to these selected polygons. These potentially intersecting polygons are found by checking the bounding box of the evaluated polygon against the other polygons in the dataframe.

Then, the FID of the overlapping polygons are stored in a dictionary, with the FID of the current polygon assessed being the dictionary key, and the list of overlapping polygon's FIDs are stored in the dictionary values. The level of polygons that are overlapping is also needed for assigning elevation. Thus, the function further evaluates the level of the road using attributes stored in 'F_ELEV' and 'T_ELEV' columns of the centerline. The "to" and "from" elevation values range between -2 and 1, and this information will be stored in a new 'overlaps' column in the polygon dataframe. To determine the value for this column, we'll consider the maximum elevation level from both the "to" and "from" values for each overlapping polygon. If the maximum level is -2, the "overlaps" value will be 1; if it's -1, the value will be 2; for 0, it's 3, and for 1, it's 4. Figure 3 shows an example of four road polygons overlapping with each other, that happens in two different elevations.

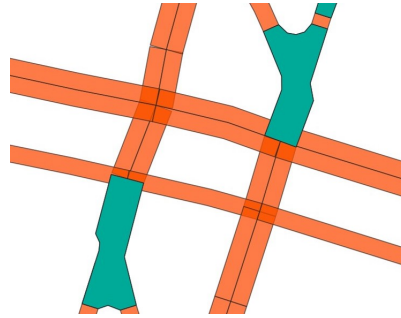


Figure 3: Overlapping polygons

3.1.6 Storing the Polygon Data

The 2D road data is stored in a GeoPandas dataframe, functioning both as a database to retain geographic features and allowing for basic spatial computations when needed, striking a balance between efficient data storage and spatial analysis capabilities. To enhance the flexibility of elevation assignment in the following step, the results are also stored in a Numpy array within a class. Each polygon's position is represented by its vertices, stored as geographical coordinates within a vertices array. Additionally, the vertices will be indexed, allowing for efficient storage of polygons using only vertex indexes, and vertices can be stored separately. Furthermore, the information about which polygons are labeled as overlapping is also stored in the same way.

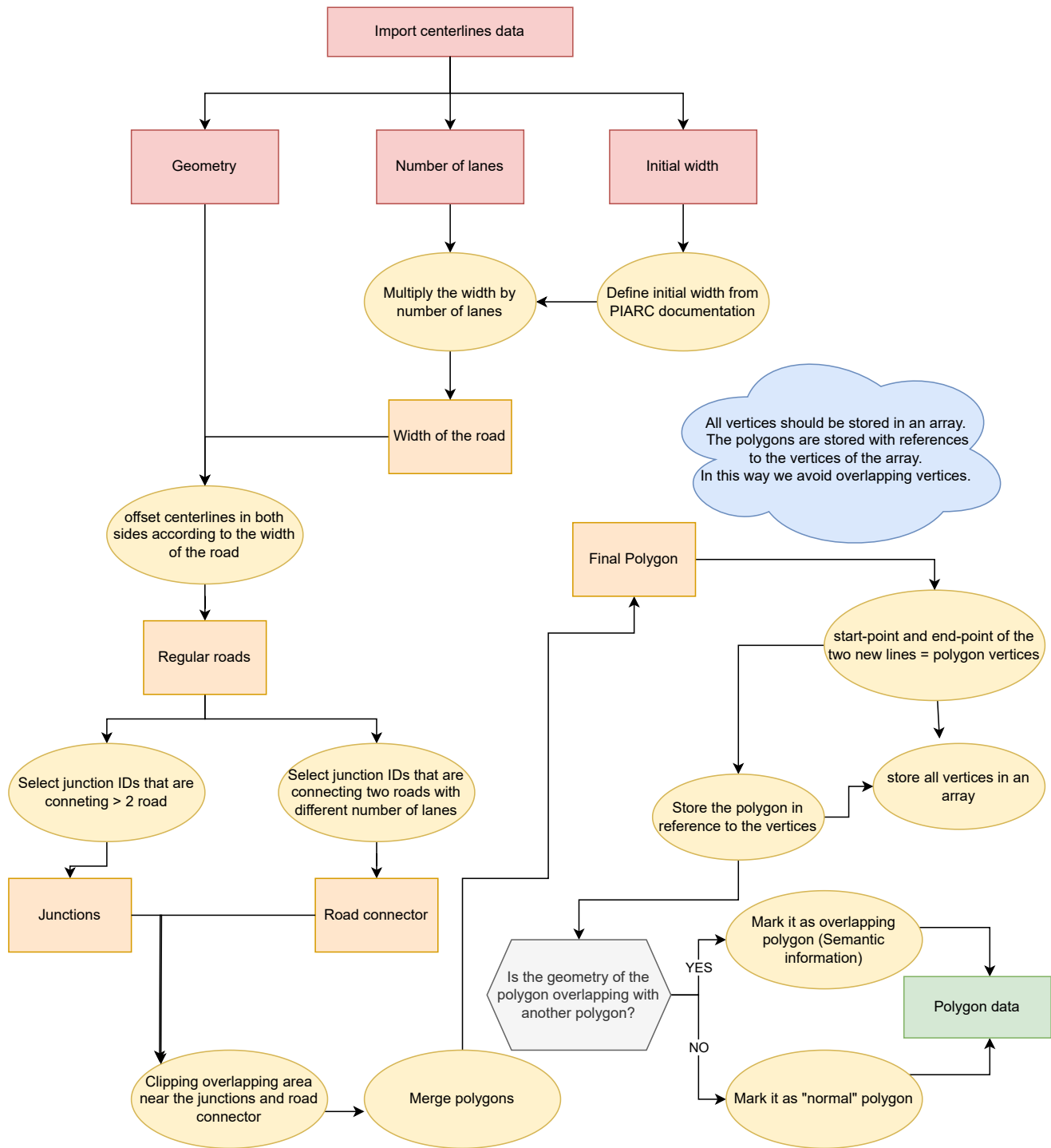


Figure 4: Pipeline of stage 1

3.2 Assigning Elevation to polygons

In the development of a 3D road dataset for navigational applications, it is crucial to assign elevation attributes to the vertices of 2D road polygons. In the proposed pipeline, Figure 8, we initially focus on determining the elevation attributes for the polygons themselves, prior to attributing these elevations to individual vertices based on adjacent polygons. Separating the elevation assignment process for vertices from that of polygons offers two primary advantages:

Resistance to Overlapping Errors Directly assigning elevations to vertices can introduce significant errors due to outliers or inaccurate measurements. This is particularly pertinent when the polygons, which are constructed based on road centerline data and assumptions about road width, do not perfectly align with the actual point cloud representation of the road. In contrast, the elevation of the entire polygon is computed using a fitting process that minimizes the influence of outliers, thereby enhancing numerical stability.

Smoothness Control In the context of developing a 3D road dataset for navigation, directly attributing elevation to vertices may produce an overly detailed, and consequently, a non-smooth topographical representation. Such granularity not only compromises the visual aesthetics of the map but also hampers the end-user’s navigation experience. Conversely, by deriving vertex elevations from polygon-based elevations, the resultant topography of road polygons is considerably smoothed. This method not only enhances the visual fidelity of the map but also significantly improves the overall navigational user experience.

Utilizing this two-stage strategy, our objective is to extract a precise and dependable 3D road dataset suitable for navigation. In terms of implementation, we propose an incremental approach to enhance algorithmic simplicity. Specifically, upon computing the elevations for two consecutive polygons, the elevation attributes of their shared vertices will be interpolated. Importantly, this procedure lends itself to parallelization, as the elevation assignment for vertices remains invariant regardless of the sequencing. This feature further augments the efficiency and scalability of our proposed methodology.

3.2.1 Inferring Polygon Elevation

The national Dutch AHN3¹ lidar dataset employs the LAS classification codes, such as 3, 4, and 5 for low, medium, and high vegetation respectively, 6 for buildings, 9 for water, and 1 for unclassified points which is also translated as vegetation as well as street furniture and cars. These classified points are filtered out because they are irrelevant to the road surfaces and only points related to the road network are of interest. This means only points with classification codes 2 and 26 are kept. These are ground points and civil structure points. A function, having as input the point cloud data and outputs the same data without the aforementioned irrelevant classified points is used. Thus, spatial indexing, which aids in finding which polygons contain a point, is handled more efficiently because of the absence of irrelevant points.

Another function is responsible for finding the relevant AHN3 points corresponding to each polygon based on that spatial indexing. This function uses the Open3D library [6]. This library handles the spatial indexing of the point cloud. The points are found by creating a 3D oriented bounding using the polygon and the minimum and maximum height in the point cloud. The point cloud is then cropped by this bounding box. The median of the points corresponding to a polygon is then assigned to that polygon as its elevation. However, this is only viable for polygons that are not overlapping with a road polygon that is above it. For those polygons the point cloud can not be used since the road there is blocked from the Lidar scan by another structure. For these polygons an interpolation method is used to find the elevation. This method first finds the closest polygons that do have an elevation based on the point cloud (elevated polygons). It does this by traversing through the polygons based on their topology: each polygon has as an attribute the adjacent polygons it is connected to. A weight is assigned to the elevated polygons based on how many steps away they are from the polygon in question. The weight is based on the number of steps between the polygon in question and the elevated polygon: $weight = 0.5^{steps-1}$. This means directly adjacent polygons have a weight of 1 and polygons that are 2 steps away have a weight of 0.5. To get the elevation of the polygon in question, the weighted average of the closest elevated polygons is taken. An example is shown in Figure 5. In this example the elevation of the polygon in question would be $(34 \cdot 1 + 30 \cdot 0.25)/1.25 = 33.2$.

¹Actueel Hoogtebestand Nederland (AHN) <https://www.ahn.nl>

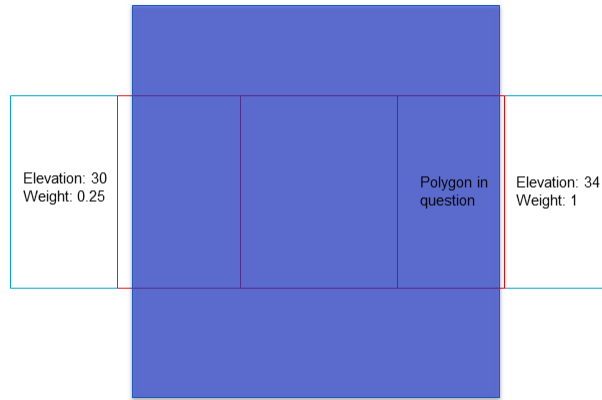


Figure 5: Example of elevation interpolation of a polygon

3.2.2 Assigning Vertices Elevation

The elevation of vertices is assigned based on their incident polygons. To make the road polygons connected in 3D space, we choose to assign the elevation of each vertex according to the elevation of the polygons directly incident to them. The connecting vertices will be assigned with the average elevation of the incident polygons. In this case, the road polygons can stay connected in 3D space when there are elevation changes. By adding elevation to polygon vertices, the road network dataset is converted to 3D representation.

3.3 A Triangulation Method for Polygon and Vertices elevation assignment

The vanilla method of using complete polygons described above works well for polygons where every corner is linked to another polygon, like the ones shown in Figure 5. It assumes that there's little or no height difference within a polygon. Yet, many polygons from the first stage don't meet this condition. The first stage produces polygons with several corners (often more than four) to show curved roads and with significant lengths. Most of these corners don't connect to other polygons, as seen in Figure 6a. Applying the basic approach, corners not linked to others get their height in two ways:

1. Assign the polygon elevation determined for the whole road polygon
2. Linear interpolate based on the elevations of vertices connected to other polygons

The first solution results in all disconnected corners having the same height, which fails to depict roads that slope up or down. Hence, road sections might appear flat as shown in Figure 6d, which in most cases does not align with the reality.

The second solution interpolates heights linearly from the edges. With this method, the road polygons could have a continuous elevation within each of them. However, it doesn't truly reflect variations in height as linear interpolation creates a constant slope. Therefore, this method fails to work for structures like arched bridges, where the start and end have the same height but the middle is different (see Figure 6e). In such cases, the elevation change in the middle are ignored and therefore unable to provide accurate reference to the user.

An example describing the degeneracy are illustrated in Figure 6. The example describes the case where the polygon elevation is determined as 1.5, the start and the end vertices elevations are 0, and the peak elevation is 3. It can be inferred that the triangulation will give the best approximate of the road elevation among other methods.

The limitations mentioned above raise the question of how to capture the elevation change within a large polygon and assign this variation to the vertices. Our solution is to use triangulation. By triangulating the road polygons with multiple vertices, we are able to subdivide the polygon to smaller regions (see Figure 6b). Then we apply the functions defined above to find the elevation of each triangles. The tessellation enables the inference of elevation variations and therefore we can obtain a 3D road dataset with more detailed and accurate representation. In addition, the smoothness and level of detail is adjustable by applying smoothing filter and triangle subdivision.

The method has potential drawbacks, including the planarity issue. Figure 7 illustrates that when triangulation adds detail, ridges may form on road surfaces if the diagonal vertices of a quadrilateral, formed by two adjacent triangles, have significantly different elevations. This issue however can be mitigated using the previously mentioned smoothing filter. Another limitation occurs with large triangles, where default triangulation fails to accurately represent true elevation changes in road segments. A possible fix for this flaw could be conforming triangulation[7], where steiner points are added on the polygon edges to make the triangles as equilateral as possible.

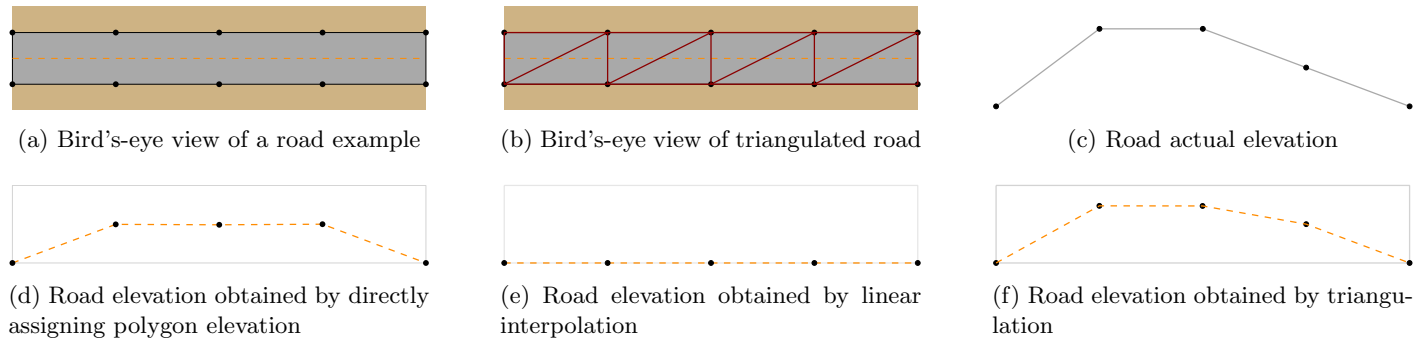


Figure 6: An example illustrating the result of different strategies for assigning vertices elevation

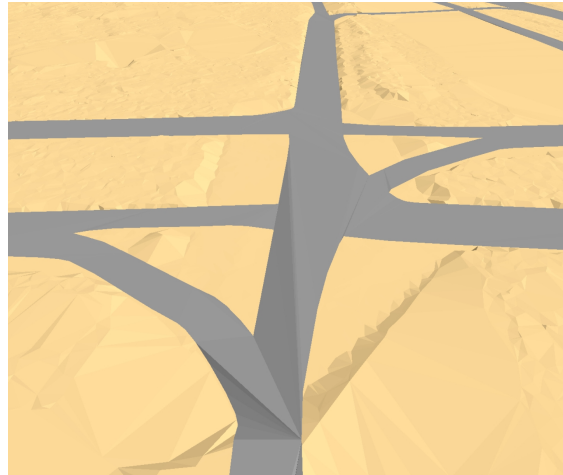


Figure 7: Example of non-planar road polygons

3.3.1 Terrain

The terrain around the roads is important for visualisation. To obtain a terrain model the same point cloud as for the roads is used. The terrain model should be made using a constrained Delaunay triangulation. The constraints should be the edges of the road polygons that are on the ground. This was not done during this project. Instead, a normal Delaunay triangulation is made, and the vertices of the road polygons are added into this. The terrain in the location of the roads is set to slightly below the elevation of road. The point cloud is thinned to keep ten percent of the points at random. This is done to increase the efficiency and remove redundant detail. This terrain model is then saved as a triangulated irregular network (TIN).

3.3.2 Output

The output of the algorithm is stored in a CityJSON file [8]. Each road polygon is stored as a city object with attributes. The network information of the polygons is stored by having as an attribute a list with adjacent polygons. Other semantics that are in the input dataset should also be stored in the output. This was not done during this project due issues with combining the two stages, which will be further discussed in section 4. The terrain is stored as a "TINRelief" object. The CityJSON file is written according to the CityJSON specifications ². [8]

²visited on 02-11-2023 <https://www.cityjson.org/specs/overview/>

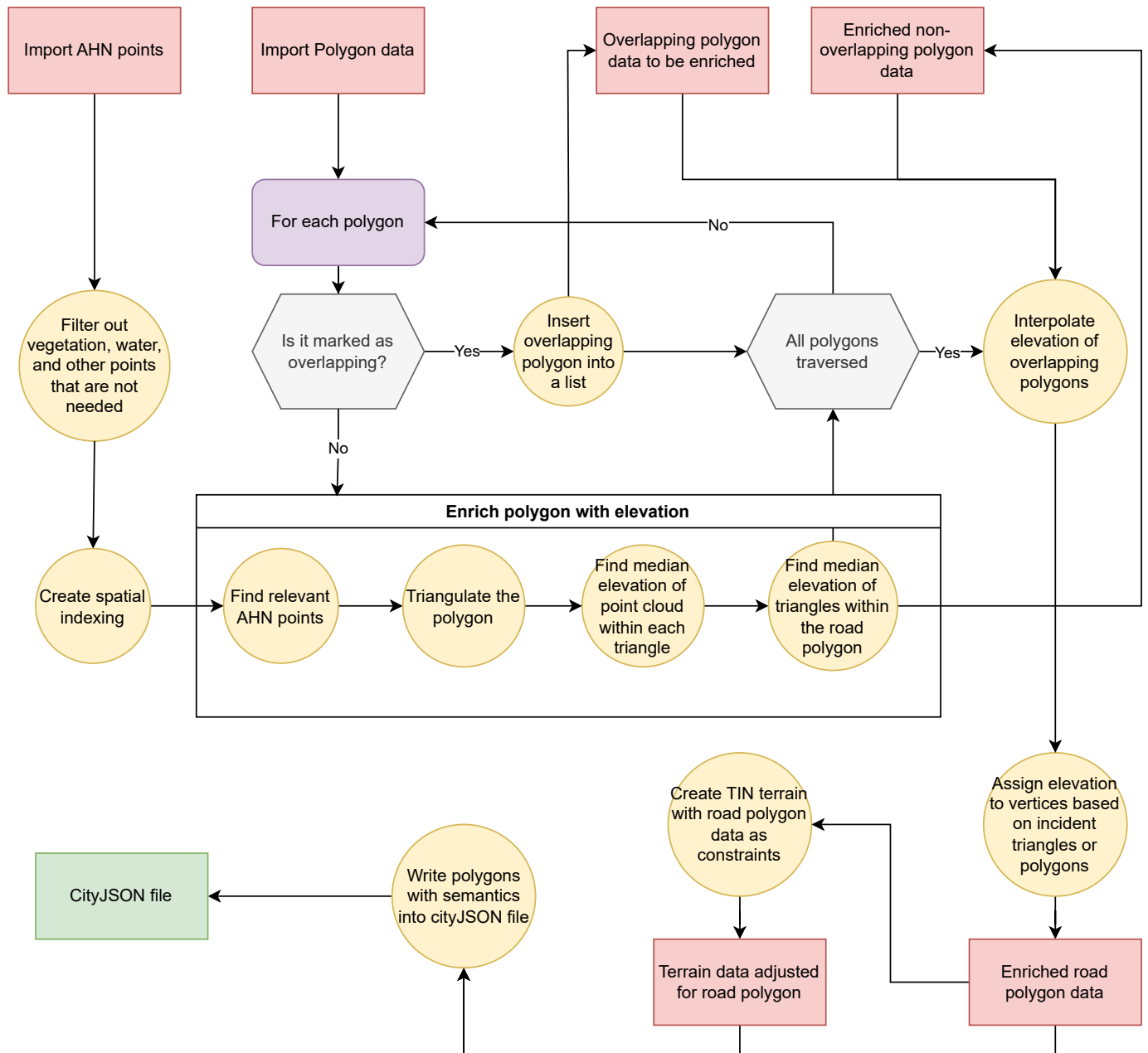


Figure 8: Pipeline of stage 2

4 Results

The methodology described in section 3 was implemented as much as possible and tested on the datasets provided by Tomtom. The results are shown in subsection 4.1. These results are evaluated in subsection 4.2. The advantages and disadvantages of the methods are discussed in subsection 4.3.

4.1 Results

Although both of the aforementioned stages that this synthesis project tackled were implemented sufficiently enough as individual entities, the combination of both of them in order for a complete implementation of the integration of 3D roads and terrain for navigation purposes couldn't succeed due to small discrepancies within the stages or strict constraints set. More specifically, ambiguities in storing the produced 2D polygons using the boundary representation method couldn't be resolved by the team while at the same time, the implementation regarding the assignment of elevation to the polygons' vertices couldn't handle more than ideally topologically and geometrically constructed polygons.

4.1.1 Stage 1 Results



Figure 9: The result of the test case (Aerial image from Kadaster [1])

The final result of the process consists of a comprehensive representation of the road network which includes various elements such as road polygons, junction polygons, and road connectors. Road polygons represent the physical layout of individual road segments within the network. These polygons typically follow the centerline of the roads and extend to their defined widths. They help depict the spatial extent and characteristics of each road, including its shape, width, and location within the network. Junction polygons are representations of the areas around road intersections. They encompass the space where multiple roads converge. Junction polygons are created by buffering the road segments that meet a junction. They are modified and smoothed to improve the visual appearance of the junction. Finally, road connector polygons facilitate smooth transitions between roads with different numbers of lanes. They help in maintaining a consistent and aesthetically pleasing representation of the road network. Topological attributes in polygons store information about the connectivity between polygons. Specifically, they define which polygon is the "from" and which is the "to" polygon in the road network, aiding in navigation, routing, and spatial analysis.

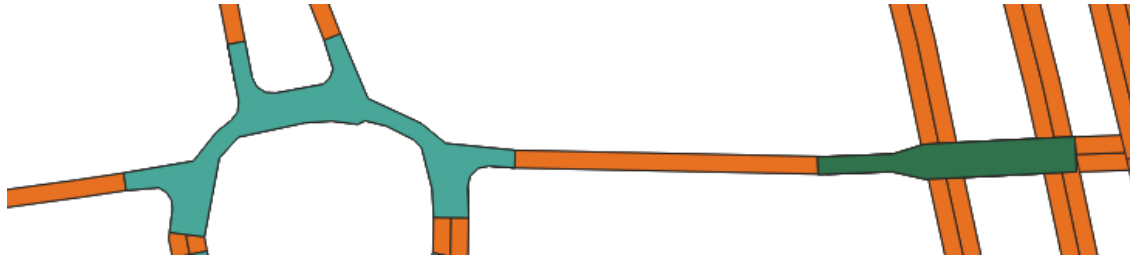
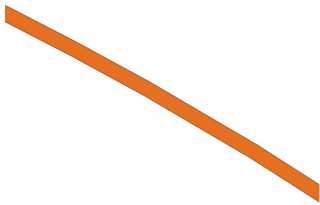


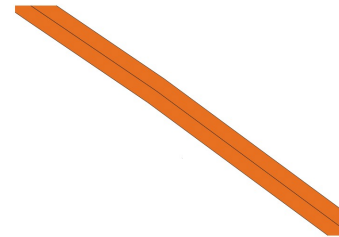
Figure 10: Visualisation of part of the road network. On the left side a junction is appeared and on the right a road connector is shown with green color.

Below there are some visual illustrations displaying the results, showcasing regular roads, junctions, and road connectors.

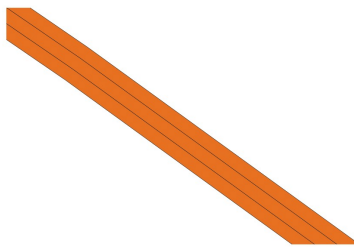
- regular roads



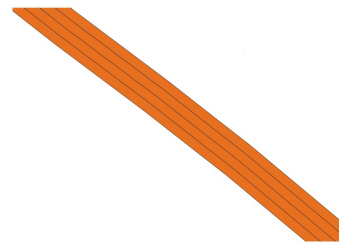
(a) Part of road polygon with 1 lane



(b) Part of road polygon with 2 lanes

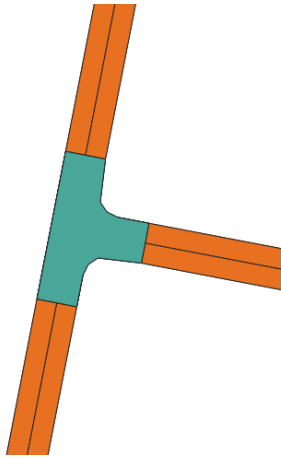


(c) Part of road polygon with 3 lanes

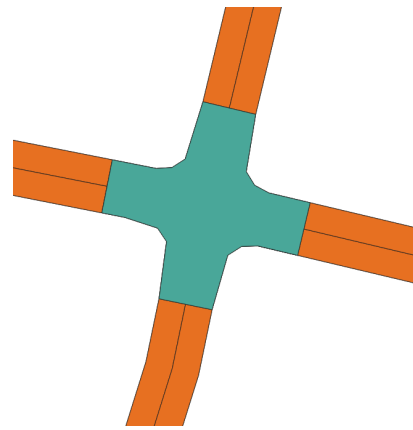


(d) Part of road polygon with 4 lanes

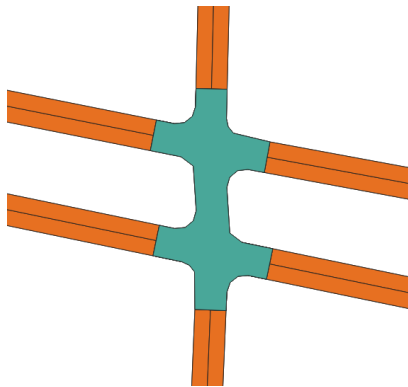
- junctions



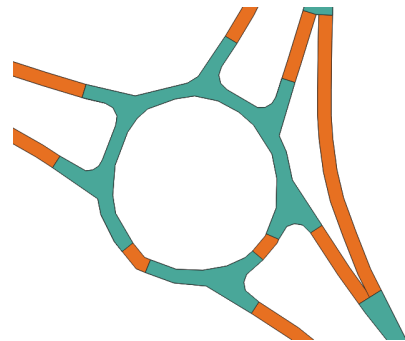
(a) Junction with 3 connected roads



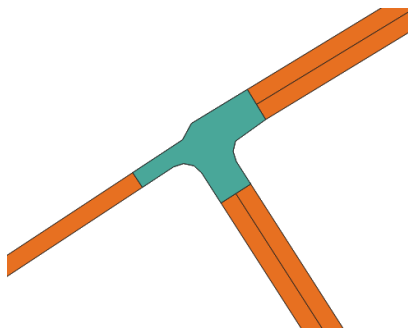
(b) Junction with 4 connected roads



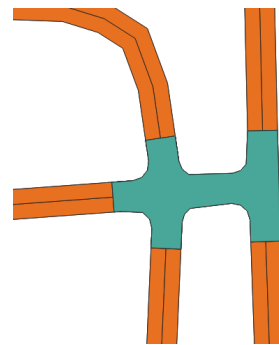
(c) Two adjacent junctions



(d) Roundabout



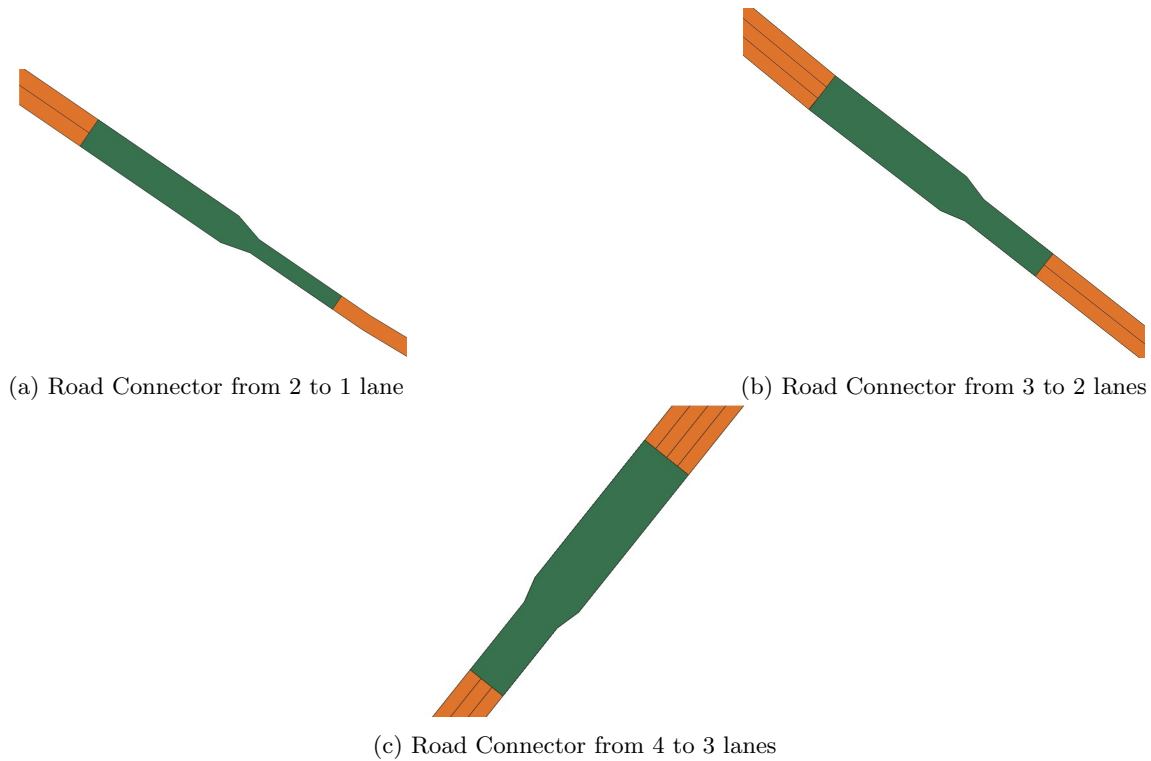
(e) Junction with connected roads having multiple count of lanes



(f) Two adjacent junctions

Figure 12: Junctions

- road connectors



4.1.2 Stage 2 Results

The results of stage 1 could not be used as input due to the reasons explained at the start of subsection 4.1. Instead, stage 2 was tested using three test cases constructed from the polygon dataset provided by Tomtom. The input into the stage 2 software has some requirements, which are listed below.

- 1. The Polygons should be valid.
- 2. Overpasses should not share vertices with the road below.
- 3. Polygons need to have as an attribute its adjacent polygons.
- 4. Polygons need to have as attributes which polygons it overlaps with and a number indicating the vertical order of the polygons (which is highest).

The three test cases used are listed below. the 'Overpasses' test case needed to be prepared first to fulfill the above requirements.

- **Overpasses:** This is a test case where roads cross over each other. It also includes connector roads between normal roads and a highway. The area of this test case is visualized in Figure 14.
- **Residential area:** This is a test case in a standard residential area. The area of this test case is visualized in Figure 15.
- **Hills:** This test case is in an area with larger elevation differences. The area of this test case is visualized in Figure 16.



Figure 14: The Overpasses test case (Aerial image from Kadaster [1])



Figure 15: The Residential area test case (Aerial image from Kadaster [\[1\]](#))



Figure 16: The Hills test case (Aerial image from Kadaster [1])

The output is visualized in Figure 17 for the overpasses, Figure 18 for the residential area, and Figure 19 for the hills. The roads in the results look promising. The road polygons that should be connected are connected, and the roads that have a lot of elevation change increase in height smoothly. In the overpasses test set, the terrain does not look ideal aesthetically, but the overall shape conforms to reality (except right next to the road, which would look better if constraint triangulation is used). In the hills and residential area test sets, the terrain does not look very good. In the residential area this is due to the lack of ground points where there are buildings. In the hills the terrain has no triangulated irregular network (TIN) points in certain areas. This might be caused by the random thinning of the point cloud. There are less ground points in this area due to the many trees.

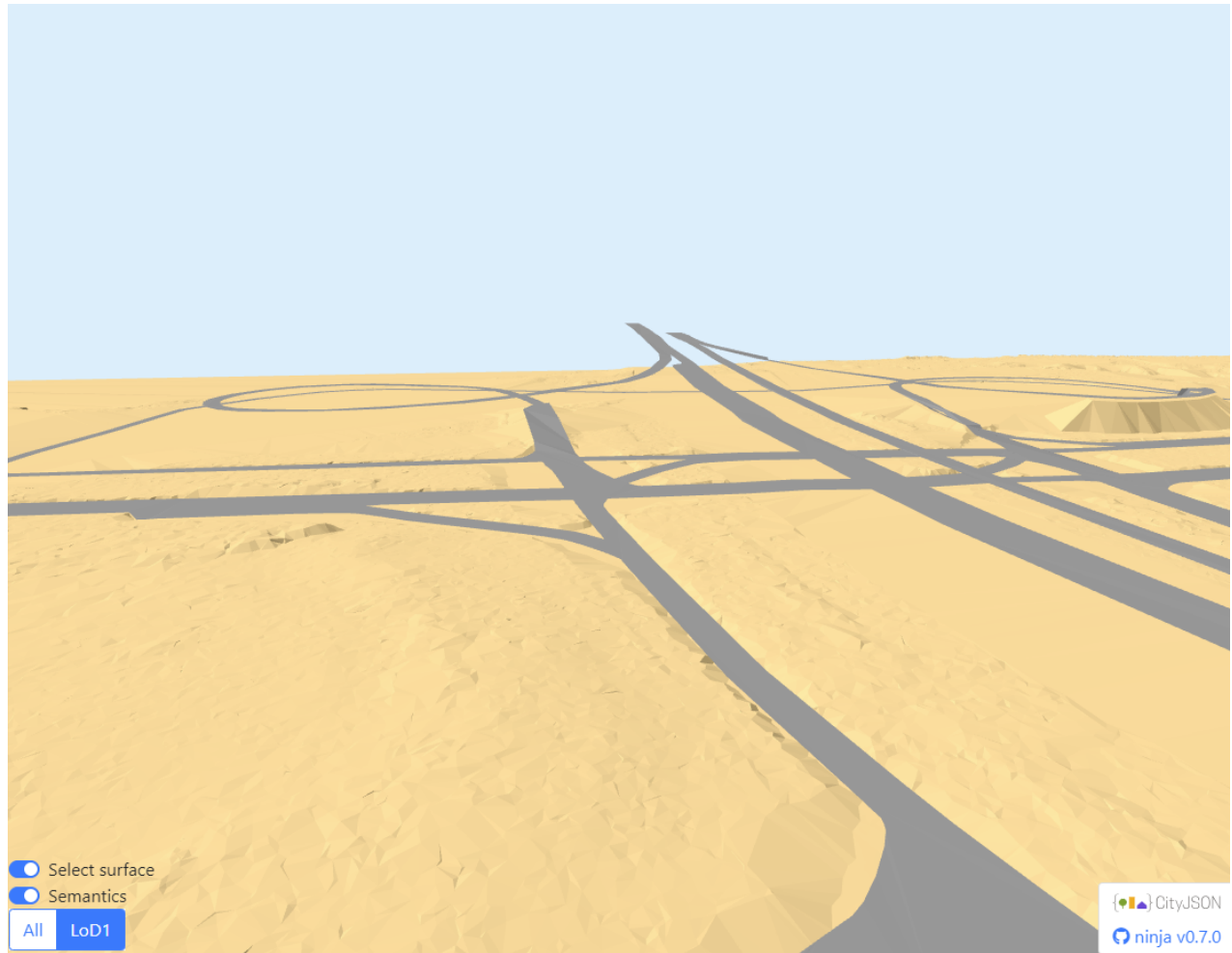


Figure 17: The Overpasses test case results

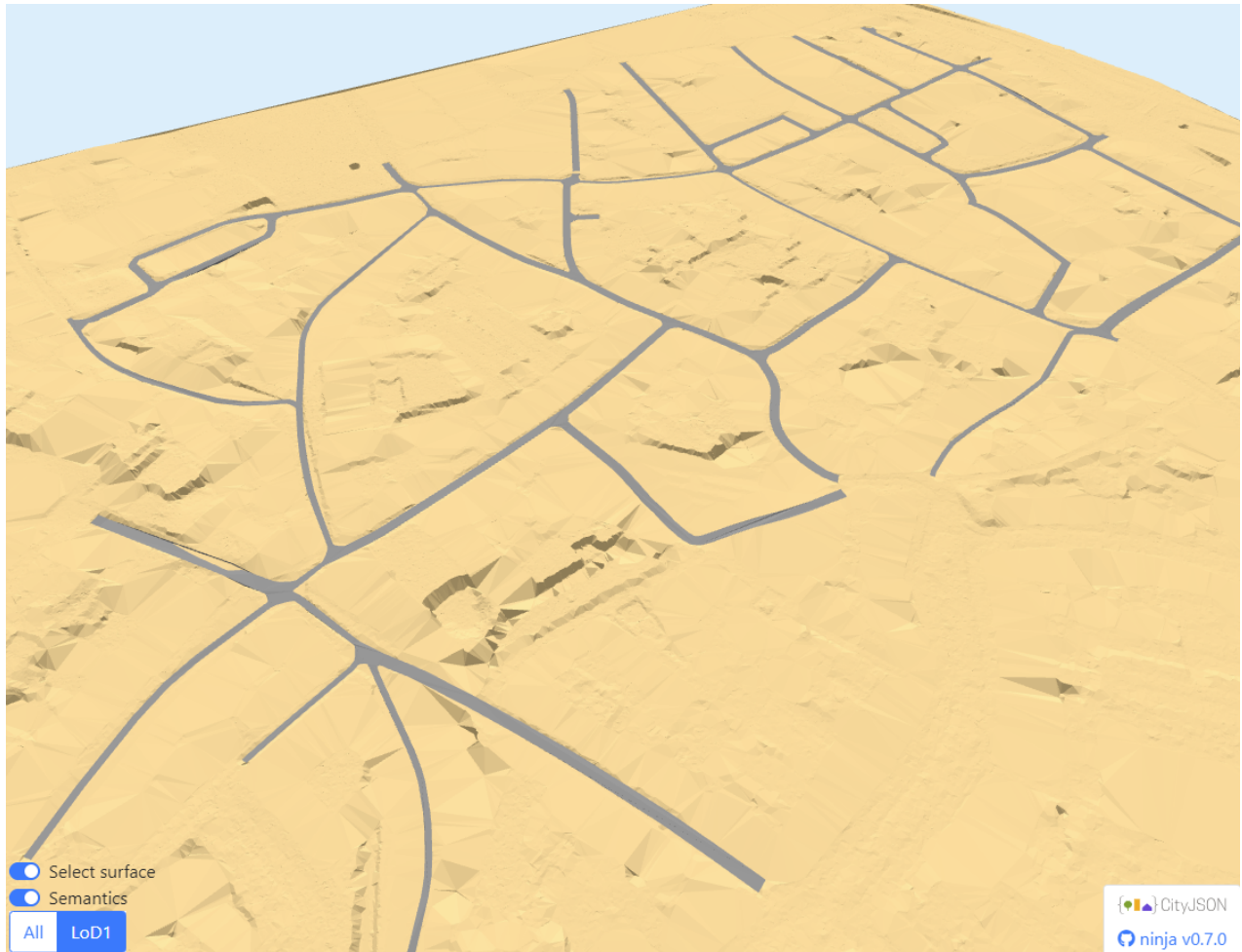


Figure 18: The Residential area test case results

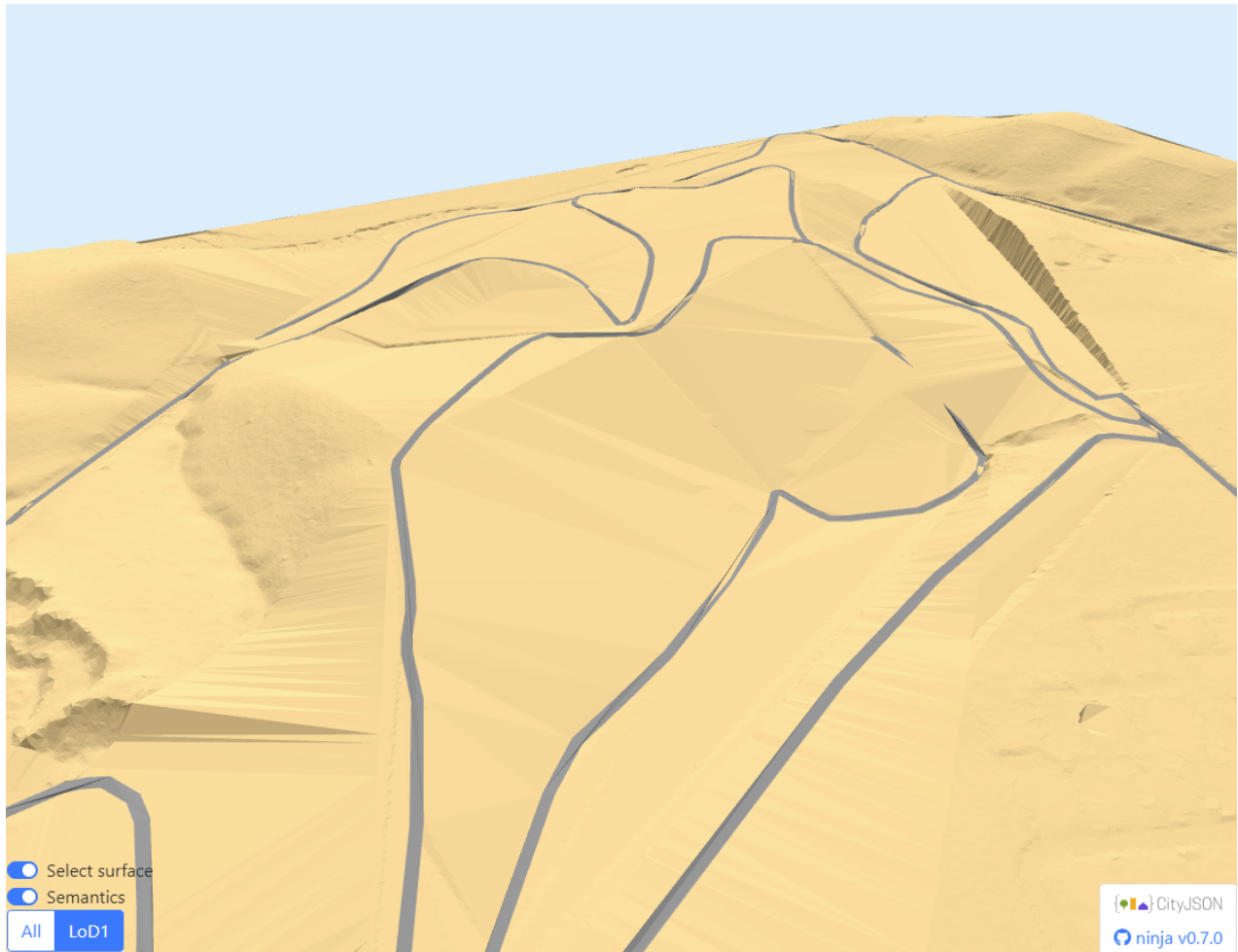


Figure 19: The Hills test case results

4.2 Evaluation of Results

4.2.1 Evaluation of Polygons

The code generally produces the expected output with well-visualized road polygons, junctions, and road connectors. However, there are still numerous errors in the code that need to be addressed.

- At times, spikes appear between two polygons representing the same road segment, particularly when there's an angle in that segment's alignment. While the centerlines of the same road segments are successfully merged, the issue arises during the polygon creation phase when the centerlines are buffered. These spikes introduce inaccuracies in the representation, and addressing this anomaly is crucial for maintaining a seamless and visually coherent road network depiction. Since the neighboring roads are not connected, there are no detected shared vertices within these roads. This could lead to continuity problems of the whole road polygon network.

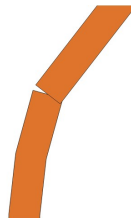


Figure 20: Gap in road turn

- One of the flaws of the generated road polygons is the presence of dangling polygons. Dangling usually forms near the junctions or road connector, due to the clip operation that was performed. These segments often result in irregular

shapes, creating invalid polygon geometries within the network representation. Additionally, due to these danglers, certain vertices within the polygons may appear multiple times, potentially impacting the overall data consistency.

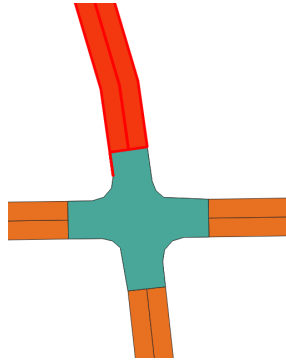


Figure 21: Dangling polygon

- In certain instances, road connector polygons are not being generated, even when there is a disparity in the number of lanes between road segments. This issue requires attention, as the expected creation of road connector polygons is not consistently occurring, potentially impacting the accuracy of the road network representation.

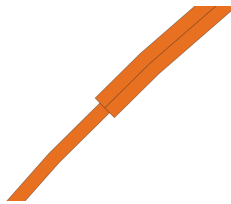


Figure 22: Road Connector is missing

geometry checks

Visual inspections were carried out to assess the resemblance of the generated results to the provided TomTom road polygon dataset. Additionally, further geometry checks were conducted in QGIS, using OpenStreetMap [9] as a reference, to evaluate the degree of accuracy in representing real-world road geometries. These checks aimed to ensure that the created polygons closely match the actual road network in the physical environment. During the process, gaps and other visible invalid geometry can also be found. To manage invalid geometries that are difficult to identify visually, the "Check Validity" tool in QGIS is employed.



(a) Sample of produced polygons



(b) Sample of TomTom polygons dataset

attribute checks

The process of conducting attribute checks involves randomly sampling multiple representative polygons within the study area using QGIS. This method assesses whether their attributes align with the expected results, focusing specifically on the "near" attribute. At times, within the "near" attribute, there are occurrences where it contains the "fid" of polygons that are not actual neighbors of the specified polygon.

4.2.2 Evaluation of Elevation

To evaluate the quality of the elevation assigned to the road network a few evaluation methods were used. First, some visual checks were performed. Then, some logical checks were performed, after which the elevation of the polygons was compared to the elevation of the point cloud.

visual checks

Some visual checks were performed to check if the results look similar to what the roads look like in real life. The visual checks were performed using Google Earth [2]. For the overpasses the comparison is shown in Figure 24a and Figure 17. For the hills the comparison is shown in Figure 25a and Figure 25b. The shape of the roads look similar in both test cases.

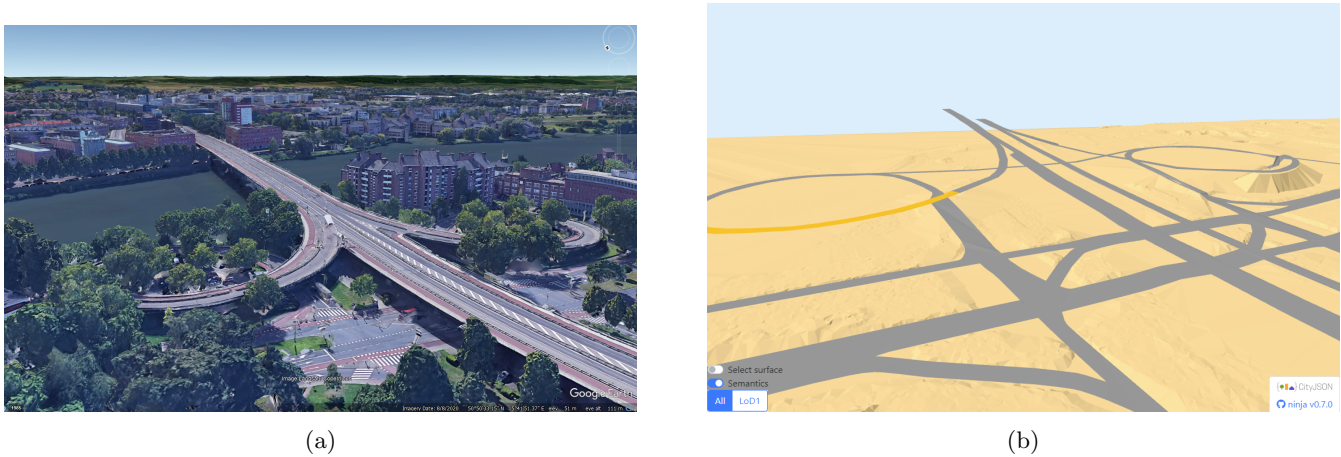


Figure 24: Overpasses test case on Google Earth (a) [2] and results (b) in the same location



Figure 25: Hills test case on Google Earth (a) [2] and results (b) in the same location

Manual sanity checks

Some simple manual checks were performed to check if the results make sense. For overlapping polygons it was checked if the difference in height is at least three meters. This was the case for all the test cases.

Elevation comparison polygons and point cloud

The modeling choices made in the project mean that the roads are smoother and more flat than the point cloud they are based on. The roads are made up of planar parts, which means the elevation is different from the point cloud. This is needed for the purposes of this project, but it is important to measure how much the elevation of the road polygons differ from the elevation in the point cloud. This is a measure of the precision of the modeling methods.

This is achieved by randomly sampling points on road polygons. The elevation of the polygon in that location is found using linear interpolation. The point cloud is then clipped to the extents of the polygon, after which the closest 10 points are used for inverse distance weighted interpolation to obtain the elevation of the point cloud. The point cloud is clipped first to prevent comparing the elevation with points that are far away below road. For example, if a polygon on a bridge is tested, the sample points might be on the edge of the bridge. The closest points used for interpolation are based only

on horizontal coordinates. If the point cloud is not clipped, the closest points could include points on the ground, meters below the road. This method does not evaluate polygons that are below other polygons, since the point cloud can not be used there. The results of this evaluation test for the three test cases are shown in Table 4.1.

Table 4.1: Average difference between road polygon elevation and pointcloud elevation in meters

Test case	Mean difference [m]	Mean squared difference [m ²]
Overpasses	0.0394	0.00555
Residential area	0.0589	0.0137
Hills	0.298	0.241

4.3 Evaluation of the Methods

After analyzing the results, the methodology itself can be evaluated. The design choices in the modeling process determine what the methodology can and can not do. These choices have many advantages and disadvantages, the most essential of which will be discussed in this section.

4.3.1 Evaluation of Stage 1 Methods

The evaluation of the methodology applied in constructing the road polygons from the centerlines gives some strengths and weaknesses. Certain errors may arise from either implementation blunder or a less suitable approach in designing the methodology. The points below will only evaluate the methodology's side.

1. Constructing junction polygons

The junction polygons have been constructed as anticipated, effectively connecting intersections where multiple roads intersect. Their visual representation also appears smooth and realistic, courtesy of the geometry refinement function. This not only elevates the visual representation of the road network but also significantly enhances the practical aspects of routing and navigation. The clipping process for the road that overlaps with the junction is also implemented well for the roads with the same elevation. The clip does not apply to the road that goes under or over the junction.

2. Assigning width of the roads

During the implementation of assigning width, the existing literature on road width in the Netherlands has limitations, primarily providing guidelines for road design aimed at optimizing accessibility, safety, and environmental considerations[10]. Concrete data about the actual width of roads in the country is relatively rare. Consequently, the road width is calculated by multiplying the number of lanes by the width of each traffic lane, as specified in international comparison of width of traffic lanes guidelines provided by The World Road Association (PIARC)[11]. The decision to use one value of 3.5 meters has its benefits and drawbacks. Some of the benefits of this standardized value is that it simplifies the modeling process and ensures uniformity in road width representation across dataset. However, this might contribute to some drawbacks. The road model would not capture the real-world variation of the actual roads, since they often have varying lane widths based on road type and categories. It can also lead to visual inaccuracies while using the road model for routing.

3. Finding neighboring polygons

This project employs junction IDs for identifying neighboring polygons, which offers both advantages and disadvantages when it comes to retrieving topological information. Using junction IDs can increase the effectiveness of finding neighboring polygons as it directly connects road segments through their junction nodes. It liberates the process from strict geometric constraints, irrespective of their geometrical validity. Junction IDs also facilitate the organization of road network data, making it easier to manage and retrieve topological information efficiently. Nonetheless, some drawbacks are found. The accuracy and reliability of the road data output are intricately tied to the quality of the centerline data. Additionally, an extra step needs to be done separately to store the vertices of the polygon, due to lack of detection of shared vertices in the initial process. The challenge persists when the IDs stored in the road data do not align with their actual status, causing the function to fail in recognizing them as adjacent polygons. Adapting the function to accommodate this scenario by incorporating every polygon that uses either of the two junction IDs poses another issue. This adjustment not only includes neighboring polygons but also extends to the neighbors of those neighbors, rendering the results unusable.

Moreover, most of the spatial analysis that is used in this part of the methodology is within the GeoPandas library[5]. While Geopandas dataframes are useful for spatial data management, they may not provide the best tools for network analysis and topology creation. This is because Geopandas primarily focuses on the representation and manipulation of spatial data using GeoDataFrames. It lacks the extensive network analysis and graph algorithms that are needed

to identify neighboring polygons.

4. Evaluating road connectors

A similar issue arises in the creation of road connectors as in determining neighboring polygons, because it also relies on the consistency of junction IDs and the number of lanes. In instances where the function fails to identify continuity, the road connector would not be created, resulting in abrupt breaks where the lanes of the road either merge or diverge.

5. Cleaning the polygons

It can be seen in the section 4.2.1 that some of the imperfections of the polygons happened because the cleaning method is not being implemented well. The method to eliminate polygon smaller than the threshold worked quite well, this also eliminates the polygon particles as the result of the clipping. However, the presence of dangling was found in the middle of the implementation process. Dangles are often formed to the polygons near junctions or road connectors. To address these challenges, it's crucial to implement a process for detecting and resolving dangling polygons to enhance the overall quality of the road network representation. Some fixing and cleaning efforts are done, including incorporating v.clean method from the Geographic Resources Analysis Support System or GRASS GIS. This is a spatial tool that is used to clean the polygons by removing gaps, discarding dangling geometries[12]. However, the tool does not work as well as it was predicted, resulting in polygons that are not clean enough until the end of the process.

4.3.2 Evaluation of Stage 2 Methods

To be able to model the road network modeling choices were made which have some advantages and disadvantages. The choice was made to compute the median height of the point cloud within triangles within polygons. This means not all the information in the point cloud is used. Using more detailed information from the point cloud could make the roads more bumpy. Furthermore, there is enough detail in the road polygon elevation for navigation visualisation purposes.

The methodology in place can be very fast and efficient, partly due to the simplifications and partly due to the possibility of parallelization.

Using the median of the points inside the triangles of the roads also means the method is robust against slightly inaccurate polygons. For example, a polygon on a bridge that is partly outside of the bridge. Computing the elevation for the entire polygon at once would be more robust, but this would not give enough detail in roads with a significant slope.

There are also some significant downsides to the methodology. Due the elevation of vertices being interpolated from the adjacent triangles the vertices of a polygon are not coplanar. This might hinder some further processing of the polygons.

The choice of finding the elevation per triangle instead of per polygon also has some downsides. If part of a polygon is below a structure but not below another road polygons, the point cloud will be used for the triangles in that area, but the elevation would be wrong. Computing the elevation per polygon would be more robust to these kinds of errors. However, that would require different kinds of interpolation or smoothing within the polygon.

Using triangles could also make some polygons more bumpy. However, since roads are usually not that bumpy, this would mostly happen where polygons are in the wrong place (not in the location of the real road).

Using a Delaunay triangulation for the terrain model is a very efficient and robust method. It does mean that the triangles can be seen in the output, which might not be desired for visualization.

5 Conclusion

Thinking about the initial research question that was set in Section 2, exploring the development of a 3D model of road networks from center-line datasets and elevation data, has seen significant progress, yet it remains partially answered. Throughout the investigation, there have been successful advancements in transforming centerlines to 2D polygons and subsequently translating these polygons into a 3D format. However, the integration of these two stages—connecting the 2D polygons derived from center-line data with the elevation data to create a comprehensive 3D model—has presented challenges and remains an area that requires further attention and development.

The methodology for constructing road polygons from the given centerline dataset shows both strengths and weaknesses. A comprehensive representation of the road network took place, including polygons, junctions, and road connectors, all derived from a single centerline dataset. The evaluation highlights several areas where improvements are necessary. Challenges encompass issues such as spikes between road polygons, the presence of dangling polygons where junctions and road connectors are located, and the inconsistent creation of road connectors when there's a disparity in the number of lanes. Finding neighboring polygons based on junction IDs offers advantages in topological data retrieval but can be affected by the quality and consistency of centerline data. Effective cleaning processes are crucial to address the imperfections observed in the polygons. On the other side, the methodology's use of junction IDs for identifying neighboring polygons and organizing topological information simplifies data management and retrieval, demonstrating the potential for increased efficiency in stage 2.

The initial research sub-questions regarding the methodology and the implementation of assigning elevation to the constructed polygons can be analyzed in three different axes. The first one depicts how the input data were handled due to incompatibility that comes because of the different formats, where the manipulation of the input data was efficient enough, and thus, the initial methodology could be applied to them. The second one shows, to what extent this methodology can be implemented. The initial idea for directly assigning elevation to the polygon's vertices once the polygon elevation was found was skipped and instead assigning elevation to the vertices constitutes a different step since this parallelized approach can lead to more evenly distributed elevation to vertices incident to big polygons, such as those represent junctions. At the same time, the initial idea of creating a constraint triangulation by using road polygons was difficult to implement, and thus, a less accurate approach was adopted. The final axis describes what is the relation between the real 3D scene and the final output as well as how exactly this output can be used. Unfortunately, the semantic information was not totally stored, and applying navigation algorithms on top of that output can not happen immediately. However, Section 4 shows promising visualizations that tend to look like the real scene and can indicate an important initiative to build on.

5.1 Requirements

To answer the research questions and in order for the students, the supervisors, and the client to have the same expectations from the project the MoSCoW method[4] was initialized at the beginning, and below is depicted whether the expectations were completed (✓), partially completed (!) or not completed at all (✗).

Must

- ✗ Extract a 2D polygon representation of the roads based on the center-lines dataset.
The polygons were partially completed according to what was mentioned in Section 4.2.1.
- ✓ Integrate terrain elevation to the road polygons.
The elevation to the polygons' vertices was successfully assigned according to Section 3.3.
- ✓ Model and store the terrain around the roads with a 3D representation.
The terrain was successfully integrated for the examined test cases according to Section 4.1.2.
- ✓ Handle topology in the 3D dataset correctly.
The topology was handled correctly while the output CityJSON file for the roads and terrain is topologically valid.
- ✗ Store the 3D representation of the road network effectively to use for navigation purposes.
Applying navigation algorithms to the produced output can not be directly implemented. However, it could be partially resolved by combining the underlying center-line data with the FID of the produced 3D polygons.
- ✓ Store the directions and the lanes of the road.
The directions and the lanes were successfully stored for the produced polygons according to Section 3.1.4

Should

- ✓ Create a 3D representation of overpasses like intersections and overpass bridges.
It was successfully implemented based on Section 3.2.1.

- ✓ Compared the accuracy of the resulting 2D polygon with the polygon data provided by Tomtom.
It was successfully implemented on Section 4.2.1.

Could

- ✗ Model the objects around the main roads (sidewalks, buildings) with a 3D representation.

Will not

- Develop a 3D visualization application that is dedicated to 3D navigation purposes.

6 Further Development

This project is a first step towards 3D road networks for navigation purposes. There are many improvements and additions to be made for this to be commercially viable. The most important of those that have come to light in this project are discussed in this section.

6.1 Alternative Approach in Managing Polygon Topology

Establishing the topological relationships among road polygons by considering their geometry rather than relying on junction IDs, which portray inconsistencies. To ensure accurate road connectivity, the approach involves deducing connections based on shared vertices between adjacent road centerlines. This method would provide a way of establishing reliable road network topology, particularly in cases where junction ID information may be unreliable or inconsistent. This also points out that the preprocessing of input data should include verifying whether the data for the junction IDs along each road centerline remains uniform throughout the road and if any segments exhibit a conflicting direction compared to the overall road alignment.

Another approach to polygon topology management is to use NetworkX library methods. NetworkX provides useful tools for designing, manipulating, and analysing complicated networks, such as road networks. It can generate the full network graph from road data, and it makes locating specific node IDs easier, making network-wide iterations more efficient. Moreover, finding neighboring polygons becomes straightforward, since there are a built-in graph properties like `G.adj`, which identifies adjacent nodes within the network graph. Its ability to handle complex network structures and computations adds depth to the topology management process.

Additionally, instead of creating separate polygons for multiple lanes within a single road, a more efficient approach is to merge them into a single polygon while retaining lane information as attributes. This simplifies the process of establishing the topology between polygons, as each polygon typically shares edges with just two neighboring polygons. This approach will not only simplify data management but also makes analysing and dealing with the road network easier.

6.2 Obtain the Road Width from Other Data

Another means of obtaining road width can be employed for further development. Remote sensing technology provides detailed imagery of roads all over the country, which allows us to extract road width information from satellite images. This approach has the potential to enhance the spatial and temporal accuracy of the resulting road model[13]. Enhancing spatial accuracy becomes important because not all roads comply with government regulations, particularly those built before the regulations were enforced. Additionally, for recently renovated roads where the width may have changed, using current satellite images can improve temporal accuracy. However, achieving this may require a more advanced algorithm, potentially involving techniques like deep learning for image analysis. Choosing the proper satellite imagery that gives the necessary resolution can also be a challenge.

6.3 Enrich the Attributes of the Junctions

The newly constructed junction polygons currently possess only fundamental attributes. Although, a dedicated polygon made solely for junction can open more possibility in improving routing and navigation support. There is a need to consider supplementing them with additional information. It's common for drivers to encounter confusion at road intersections, and providing details such as directional flow and turn restrictions can significantly enhance their ability to navigate routes effectively. These attributes serve as essential aids in ensuring smooth and accurate journeys. Besides, including other semantic attributes such as road type information including highways or residential streets aids in route planning based on the driver's preferences.

6.4 Scale up to Large Dataset

This synthesis project uses specific AHN3 tiles for the elevation to be assigned to polygon vertices. This means that the used point cloud data are not being retrieved automatically from a database or a server but instead, they have been downloaded and directly used for checking the algorithm's capabilities. Further development on top of what was discussed so far, would indicate an efficient way of directly retrieving the point cloud data. This can be done by using the free open-source WebGL-based point cloud renderer, Potree ([14]). While Potree itself is primarily designed for visualization within a web browser, one can access the point cloud data and use it within the Python code by scraping or programmatically downloading the resources from the Potree server. Potree often serves point cloud data in the LAS/LAZ format, which was one of the basic inputs for implementing stage two.

6.5 Different Treatment of Some of the Overlapping Polygons

In the second stage of the implementation, each polygon's elevation is assigned based on the point cloud points lying in its bounding box. However, for the polygons marked as overlapping, only the adjacent polygons were used for computing this value. This means that if only a small part of the polygon was overlapped, the whole polygon was treated as overlapping and thus the point cloud was not used at all. Section 3.3 could be an initiative in this modification because each polygon is subdivided into triangles and thus, retrieving information on whether or not a polygon is fully overlapped based on the triangles lying in it can be obtained.

6.6 Lightweight Terrain Integration

The current integration of the terrain around the examined road network is being succeeded by triangulating a thinned version of the used point cloud. However, random thinning of a point cloud poses two critical challenges in the creation of a Digital Terrain Model (DTM). Firstly, it leads to a significant loss of information, potentially compromising the accuracy and intricacy of the model. When points are removed without a systematic approach, essential data crucial for precisely representing the terrain's features might be lost, impacting the DTM's overall detail. Secondly, the process introduces a risk of bias in the dataset's sampling. Randomly removing points can create uneven representation across the terrain, leading to some areas being overrepresented while others are underrepresented. This skewed sampling can result in inaccuracies, particularly in regions with a reduced number of data points, thereby impacting the final model's reliability and precision for various applications. When addressing the challenges of random thinning in point cloud data for Digital Terrain Model (DTM) creation, an effective alternative involves applying grid thinning. Grid thinning entails dividing the dataset into different-sized grid cells based on the elevation distribution within each cell. This method allows for a more structured and systematic approach to point reduction. By correspondingly assigning a representative point to each cell, based on the elevation characteristics within the cell, the grid thinning technique maintains crucial data points while reducing the dataset's size.

6.7 Using Distance-based Weighting Instead of "Topology" Steps for Assigning Polygon Elevation

The method of assigning elevation to overlapping polygons based on steps in topology, while functional, can be limited in its accuracy and representation of spatial reality. Using distance-based weighting for determining elevation offers a more intuitive and realistic approach. Distance-based weighting considers the actual physical distance between polygons, reflecting a more accurate representation of spatial relationships. By using distance as a measure, the influence of nearby polygons on the elevation of a specific area is more reflective of the real-world scenario. This method aligns better with how elevation and influence diminish over physical space, creating a more continuous and smoothly varying representation of the road network. Distance-based weighting provides a more nuanced and precise understanding of the real terrain. Overall, employing distance-based weighting in assigning elevation to polygons provides a more accurate and true-to-life representation of geographic features compared to the limitations posed by a topology-based approach.

6.8 Triangulation Refinement for Elevation Assignment

The current triangulation approach is effective in assigning elevations to road polygons and vertices under typical conditions. However, the refinement techniques outlined in Section 3.3, specifically the smoothing filter and conforming triangulation, are designed to address specific degenerate scenarios and have not been integrated into our system. Incorporating these methods can enhance the quality and utility of the resulting 3D road dataset substantially. Therefore, one of the future developments of the workflow could be apply the refinement methods.

A Project Organisation

A.1 Team Members



Dimitris Mouzakidis

Nationality: Greek

Background: Int. MSc Rural and Surveying Engineering (Faculty of Engineering, AUTH Greece)

Work Experience: Survey engineer (International research team of the ASCSA)

Mail: D.MOUZAKIDIS@student.tudelft.nl



Puti Nabila Riyadi

Nationality: Indonesian

Background: BSc Geodesy and Geomatics Engineering (Bandung Institute of Technology, Indonesia)

Work Experience: Geomatics Analyst (Pertamina Internasional EP - Indonesian state-owned oil and natural gas corporation), GIS Engineer (Esri Indonesia)

Mail: PutiNabilaRiyadi@student.tudelft.nl



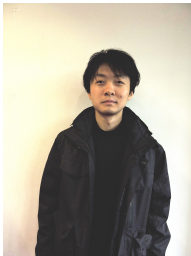
Georgios Konstantinos (Akis) Nestoras

Nationality: Greek

Background: Int. MSc Spatial Planning and Development (Faculty of Engineering, AUTH Greece)

Work Experience: Geospatial analyst using remote sensing - Intern (IrRADIARE, Lisbon)

Mail: G.K.Nestoras@student.tudelft.nl



Longxiang Xu (Leon)

Nationality: Chinese

Background: Bsc Geographic Information Science (China Universit of Geosciences Beijing), BES Geomatics (Universit of Waterloo)

Work Experience: Big Data Analsis Intern (Chinese Academy of Sciences Computer Network Information Center)

Mail: L.Xu-18@student.tudelft.nl



Na Liu

Nationality: Chinese

Background: Bsc Geographic Information Science (Lanzhou University)

Mail: N.Liu-5@student.tudelft.nl



Stein Köbben

Nationality: Dutch

Background: Bsc Aerospace Engineering (Delft University of Technology)

Mail: S.Kobben@student.tudelft.nl

A.2 Responsibilities of Team Members

The project initially outlined roles and responsibilities for each team member based on their strengths and interests, as detailed in the Project Initiation Document (PID) [15]. Weekly meetings were scheduled with both supervisors and the client. To maintain an equitable distribution of duties, team members rotated in delivering updates on the progress made and in recording meeting minutes.

Project manager: responsible for planning, executing and overseeing the project from initiation to completion	Stein Köbben
Secretary: defines the agenda and leads the meetings, books project rooms, keeps minutes during meetings	Na Liu
Report manager: defines the report requirements and checks if they are met.	Akis Nestoras
Lead software manager: leading role in software development	Longxiang Xu
Communication manager: facilitates communication between the team, the coaches and the client	Dimitris Mouzakids
Quality Manager: ensures that data and work meets the quality standards and goals	Puti Nabila Riyadi

Table A.1: Roles and responsibilities of team members

A.3 Client

TomTom is a Dutch company known for its global navigation and mapping products and services. They specialize in GPS navigation devices, software, and digital maps for various industries, including automotive, consumer electronics, and location-based services. TomTom's products and technologies help individuals and businesses with accurate and reliable navigation solutions, making it easier to find routes and locations efficiently.

A.4 Responsibilities of Client

Throughout the project's timeline, the client has been actively engaged with the team, offering valuable collaboration and guiding us with a spirit of cooperation, all in pursuit of achieving the optimal outcome. Furthermore, the client maintained open and consistent communication regarding their needs and goals, fostering a smooth and fruitful partnership.

A.5 Supervisors



Dr. Ken Arroyo Ohori

Nationality: Mexico

Position: Lecturer and researcher at the 3D Geoinformation Group of TU Delft

Mail: K.Ohori@tudelft.nl



Dr. Stelios Vitalis

Nationality: Greek

Position: Software Engineer, TomTom

Mail: s.vitalis@tudelft.nl



Jantien Stoter

Nationality: Dutch

Position: Professor 3D Geoinformation Group of TU Delft

Mail: j.e.stoter@tudelft.nl

A.6 List of needs

Compiling a list of requirements and determining their sources is a crucial aspect of project planning as mentioned in the Project Identification Document. It is important to note that the team was not responsible for acquiring data or obtaining specialized equipment. Instead, the primary focus of this project revolved around the dataset required for developing a strategy to integrate and store 3D road and terrain information. Below, the data requirements are outlined.

Software	Use
CloudCompare	Point Cloud viewer
CityJSON ninja	CityJSON viewer
Blender	CityJSON viewer
Google Earth	

Programming Languages & libraries	Use
Python 3.10	Algorithm Development
(Geo)Pandas (Python library)	Manipulation of 3D geodata
Open3D (Python library)	Manipulation of Point Clouds
LasPy (Python library)	Manipulation of Point Clouds
Shapely (Python library)	Manipulation of geometric objects
SciPy (Python library)	Mathematical algorithms and convenience functions
StartinPy (Python library)	Modelling of terrains
Concave_hull	Computational Geometry

Table A.2: Software & Programming languages to be used

- **Centerline Data:** High-quality road centerline data for the area of interest in *.shp* or *ascii* format.
Provider: Tomtom (Collaborating company)
- **Elevation Data:** Accurate elevation data in *.laz* format.
Provider: Government agencies (e.g. PDOK) and/or the collaborating company.
- **Road Polygons:** 2D polygons of the roads of the study area in *.shp* format.
Provider: Tomtom (Collaborating company)

To fulfill the project requirements mentioned earlier, the team utilized the following software packages, programming languages, and libraries to process the aforementioned data.

A.7 Communication plan

The project team carefully tracked each team member's availability to enable effective communication. Communication involved regular meetings, occurring 2-3 times weekly, complemented by a weekly session with both supervisors on Fridays. Additionally, a Discord channel facilitated communication with supervisors outside of meetings, and a WhatsApp group allowed quick informal exchanges among the team members.

For document sharing and collaboration, Google Drive served as the primary platform for sharing materials like the Project Initiation Document (PID) and datasets. GitHub facilitated code sharing, enhancing collaboration, while Overleaf was chosen for collaborative report writing. These platforms were selected to ensure smooth information exchange and teamwork among team members and supervisors.

A.8 Planning

A planning was made during the start of the project. For the most part, the planning as followed. Developing the initial stage 1 and sage 2 software took longer than expected, this continued into week 9.

1. Organisational phase

- **Defining the project** At the project's outset, the team discussed its scope with the supervisor and client. The team extensively researched the topic and relevant theories with the help of them
- **Communication Planning** The team devised a comprehensive communication plan for the project, outlining how and when meetings would take place among the team, supervisor and client. The protocols for internal and external team communication were also established (subsection A.7)
- **Assigning Responsibilities** A distribution of roles within the team was established, and clear responsibilities were defined.
- **Writing Project Identification Document** A document was created and submitted, outlining the project's future course and objectives.

2. Research and data analysis

- **Literature Review** Extensive research into relevant theories and methodologies were conducted.
- **Analyze given data** The team will explore and analyze the provided datasets, determining how they will be used and identify any need for additional external data.
- **Obtain elevation data** To incorporate elevation data, the team selected an external dataset (AHN3) that will be processed accordingly when the dataset will be provided by the client.
- **Develop methodology** The team defined the methodology for the software, specifying methods to be employed at each stage of the modeling process, outlining the interplay between different software components, and delineating data storage procedures.

3. Midterm Software Development

- **Write midterm report**
- **Prepare midterm presentation**
- **Develop stage 1 software** Coding for the first stage of the software was commenced.
- **Develop preliminary stage 2 software** This phase involved laying the foundation for stage 2 of the software, which included tasks such as creating a terrain model and assigning elevation to road polygons, excluding the handling of special cases like overpasses and tunnels.
- **Test and review** Preliminary results were shared with supervisors and client, incorporating their feedback, suggestions, and necessary modifications into the code.

4. Finalisation

- **Write final report**
- **Prepare final presentation**
- **Geomatics day presentation**
- **Write task distribution document**

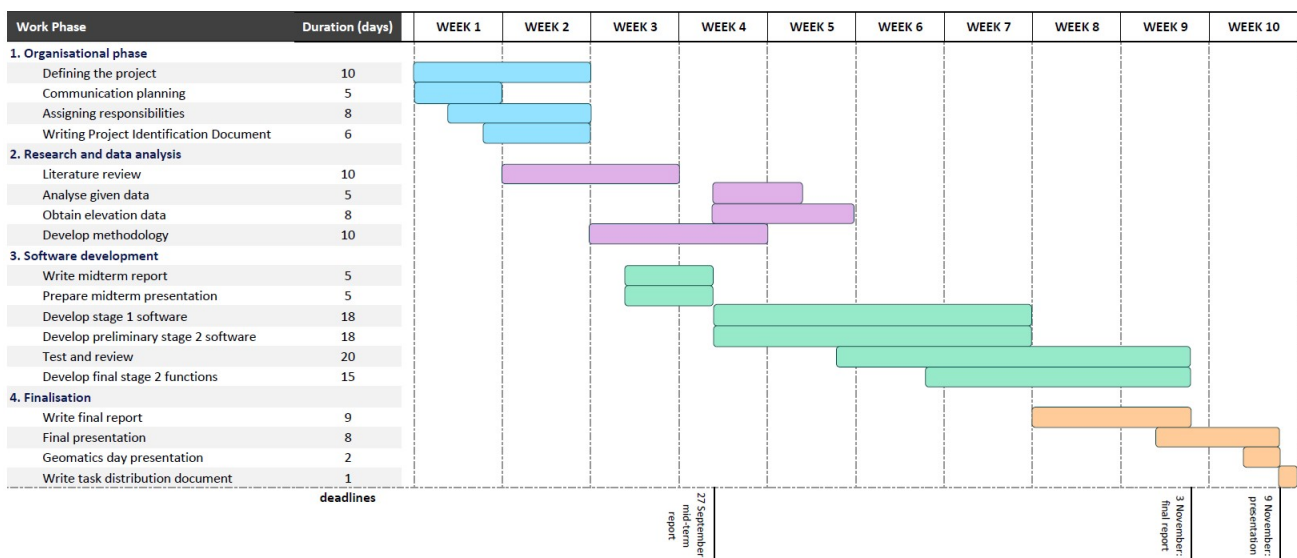


Figure 26: Gantt diagram of the project planning

A.9 Risk Overview

Below is the list of risks identified at the start of the project. Most risks did not occur during the project. The first external risks did cause some difficulties when preparing the polygon dataset for testing purposes, but this was not a big issue.

Internal risk

1. Resource constraints. The presence of internal resource constraints, such as limited computing hardware capabilities, has the potential to result in project delays. For instance, the computers may lack the necessary processing power to handle large datasets effectively.
2. Communication issue. Delays in project deliverables may arise due to miscommunications or misunderstandings among team members, with the client, or with the project supervisor.
3. Ineffective main approach. The initial project methodology planned during the early stages may not prove optimal, primarily because the actual dataset has not yet been assessed. Furthermore, team members have not had the opportunity to test the libraries that will be utilized.

External risk

1. Data quality and compatibility. Variations in data quality and accuracy from different sources can pose challenges when integrating these data sources for model development. This may lead to compatibility issues during the later stages of the project.
2. Software Dependencies. The project's reliance on specific software tools for data processing and model development may introduce potential problems related to licensing or software compatibility.

References

- [1] Kadaster. Dataset: Pdok luchtfoto rgb. [Online]. Available: <https://www.pdok.nl/introductie/-/article/pdok-luchtfoto-rgb-open>
- [2] Google. Google earth. [Online]. Available: <https://earth.google.com/web>
- [3] TomTom. About us. [Online]. Available: <https://www.tomtom.com/company/>
- [4] E. Miranda, “Moscow rules: A quantitative exposé,” in *Agile Processes in Software Engineering and Extreme Programming*, V. Stray, K.-J. Stol, M. Paasivaara, and P. Kruchten, Eds. Cham: Springer International Publishing, 2022, pp. 19–34.
- [5] K. Jordahl, J. V. den Bossche, M. Fleischmann, J. Wasserman, J. McBride, J. Gerard, J. Tratner, M. Perry, A. G. Badaracco, C. Farmer, G. A. Hjelle, A. D. Snow, M. Cochran, S. Gillies, L. Culbertson, M. Bartos, N. Eubank, maxalbert, A. Bilogur, S. Rey, C. Ren, D. Arribas-Bel, L. Wasser, L. J. Wolf, M. Journois, J. Wilson, A. Greenhall, C. Holdgraf, Filipe, and F. Leblanc, “geopandas/geopandas: v0.8.1,” Jul. 2020. [Online]. Available: <https://doi.org/10.5281/zenodo.3946761>
- [6] Q.-Y. Zhou, J. Park, and V. Koltun, “Open3D: A modern library for 3D data processing,” *arXiv:1801.09847*, 2018.
- [7] H. Ledoux, R. Peters, K. A. Ohori, and M. Pronk, “Computational modelling of terrains,” 2020.
- [8] H. Ledoux, K. A. Ohori, K. Kumar, B. Dukai, A. Labetski, and S. Vitalis, “Cityjson: a compact and easy-to-use encoding of the citygml data model.” *Open Geospatial Data, Software and Standards*, 2019.
- [9] Openstreetmap - geographic map view. [Online]. Available: <https://www.openstreetmap.org/#map=4/39.74/-87.10&layers=T>
- [10] SWOV. Principles for a safe road network. [Online]. Available: <https://swov.nl/en/fact-sheet/principles-safe-road-network>
- [11] “V. traffic lanes and carriageway.” [Online]. Available: <https://tunnels.piarc.org/sites/tunnels/files/public/wysiwyg/import/Chapters%20PIARC%20reports/2001%2005.11.B%20Chap%205%20EN.pdf>
- [12] G. D. Team. (2023, Feb) v.clean - toolset for cleaning topology of vector map. [Online]. Available: <https://grass.osgeo.org/grass83/manuals/v.clean.html>
- [13] T. Keijzer, A. Schipper, J. Meijer, and W. Nijland. PBL publication, 2022.
- [14] M. Schuetz, “Potree: Rendering largepoint clouds in webbrowsers,” 2016.
- [15] D. Mouzakidis, P. N. Riyadi, G. K. A. Nestoras, L. Xu, N. Liu, and S. Köbben, “Project identification document - integrate and store 3d roads and terrain for navigation purposes,” 2023.