GeoBIM for Building Permit in Rotterdam: Phase 2

F. Noardo, N. Hobeika, J. van Liempt, K. Arroyo Ohori,
T.Krijnen, S. Vitalis, J. Stoter

3D geoinformation group
Delft University of Technology NL

August 2021

1 Introduction

In an effort to automate building permit checks, "Investigating the Automation of the Building Permission Issuing process through 3D GeoBIM information" [Noardo et al., 2020] was initiated to issue building permits with the help of digital tools and 3D data including Building Information Models (BIM) and Geographic Information System (GIS) datasets. Thus, the project deals with integrating BIM and GIS datasets and the challenges of having different semantics, level of detail and geometry.

In the first phase of the project, a tool was developed to support municipality officers through several features. The first phase mainly focused on analysing the BIM data rather than integrating BIM and GIS datasets. This means that some checks required the manual input of users in order to be conducted. Additionally, its result was a Python tool which requires to be installed by the user, and there was no focus on user-friendliness on for example the presentation of analysis results.

In order to improve the user-friendliness, the first aim of the second phase of the project is first to develop a web-based tool in which IFC models can be visualised and analysed with a clearer presentation of results. Secondly, to implement functionalities to conduct three new regulation checks based on GeoBIM analysis. The regulations considered are the overhang check while automatically detecting corresponding roads, the height check with regard to the highest road level, and the parcel boundary check.

This report documents the outcomes of the second phase of this project. A specific building in Rotterdam is used as a case study. Section 2 quickly goes over the previously implemented tool and its features. Section 3 first presents details on the architecture of the web-based tool. Then, this section presents some additional guidelines for the input data since the tool now implements

working on analysis that integrates both IFC files and GIS datasets. Finally, it goes over the GeoBIM analysis that was automated with the help of multiple GIS datasets. Section 4 concludes on the second phase of this project and provides some additional future work to further develop the automated building permit checks, and indicates in detail what current issues with the tool are and how these can be improved.

2 Phase 1: implemented tool

In this section, a quick overview of the first phase of the project is given. The analysis provided in the initial tool implemented is briefly presented (see [Noardo et al., 2020]). The analysis includes height calculation, storey overlap calculation, overhang distance calculation and parking regulation check. Other features of the tool are also mentioned in this section.

2.1 Height calculation

The maximum height is the above ground height of the building. This is considered from the entrance level to the highest point in the building. However, in the regulation, the height of a building is measured from the highest road level. This is implemented in phase 2 of the project (see Section 3).

2.2 Storey overlap calculation

The storey overlap calculation extracts the overlap between a storey and the base. The base is defined by the user and should be the ground floor (GF). Each storey's footprint is extracted and stored as a 2D polygon. Using the GF and the storey's polygons, a polygon intersection allows the computation of the overlap percentage between the two polygons.

2.3 Overhang distance calculation

The overhang distance checking computes the overhang distance from one storey to the GF. However, the guidelines for this regulation are connected to the streets that the storeys overhang above. For this purpose, each side that will have an overhang over a road is manually linked to the name of a road by the user. Finally, after getting the overhang distance, the municipality officer compares it to the regulation. Since this calculation still requires manual input and evaluation, in phase 2 of the project (see Section 3), the detection of roads for each side and the comparison to the regulation is implemented automatically.

2.4 Parking regulation check

After defining building units and their function, the area of each apartment and then the number of the minimum parking units (MinPP) are computed. MinPP

is used to check if the parking units meet the regulation.

Other than analysis features, the tool also can provide a sliced plan of a storey at a certain elevation and can export storeys' footprints as Well Known Text (WKT) file.

3 Phase 2: newly implemented features

This section presents the newly implemented features of the tool. This includes the webapp, the input data requirements, and three GeoBIM automated checks. It is possible to retrieve the source code of the application following the instructions provided at `https://github.com/tudelft3d/GeoBIM-building-permit-tool`

3.1 The web application

The main reason to make the tool of Phase 1 into a web-based tool is to increase the user-friendliness, as users don't have install anything and (heavy) processes are executed on the server rather than the user's computer. The application is a fork of Krijnen [2021], which is a full-stack application allowing users to upload IFC files and view them in the browser. IFC files get converted to glTF on a Flask server using IfcOpenShell, and are then visualised using BIMSurfer2.

3.1.1 Architecture

An overview model of the architecture of the application is shown in Figure 1. As for the data part, users can either upload a georeferenced IFC file, or choose a preloaded one. Preloaded files are already converted to glTF, which enables the model to be visualised quicker. Additionally, the geoinformation datasets are stored as Shapefiles on the server. These are all used for analysis, and the roads are used for visualisation as well.

The analysis is performed on a Flask server, mainly using IfcOpenShell. It uses part of the original GeoBIM tool of Phase 1 of the project. The client can make a request for analysis to be performed, and the server will return a JSON-formatted response with the outcome. The server handles the uploading, storage, and conversion of IFC models as well.

The frontend is based on Vue.js and Three.js. For the visualisation of IFC, BIMsurfer2 is used, and to visualise geoinformation together with the model, a parses for GeoJSON to Three.js meshes is written. While it is not difficult to use Shapefiles for analysis with Python, for visualisation with Three.js GeoJSON is much easier. Only the road data is visualised, and it gets converted to GeoJSON in the client.

Lastly, the application can be deployed using Docker Compose.

3.1.2 Performance and possible improvements

Since it is necessary to run the tool on a Citrix environment, initially the performance of the tool was problematic due to the visualisation of the model being heavy on the GPU. Therefore, we have implemented a quick performance improvement in BIMsurfer2, by merging all individual Three.js meshes together. This greatly helped the visualisation performance, but as a consequence it is not possible to select individual IFC elements anymore. It can be made possible by parsing the data to Three.js meshes in a similar way as Vitalis [2021] does.

It can also be worth it to try out a new IFC loader for Three.js (`https://threejs.org/examples/webgl_loader_ifc.html`) that was not mature yet when Phase 2 of the project started. Not only for performance reasons, but also to make the application simpler by removing the need for IFC to glTF conversion and BIMsurfer2 as dependency.

Lastly, some of the analysis functions can take a long time, up to around 10 minutes. Besides improving their efficiency, it is also a good idea to make them run properly in the background, using for example Celery.

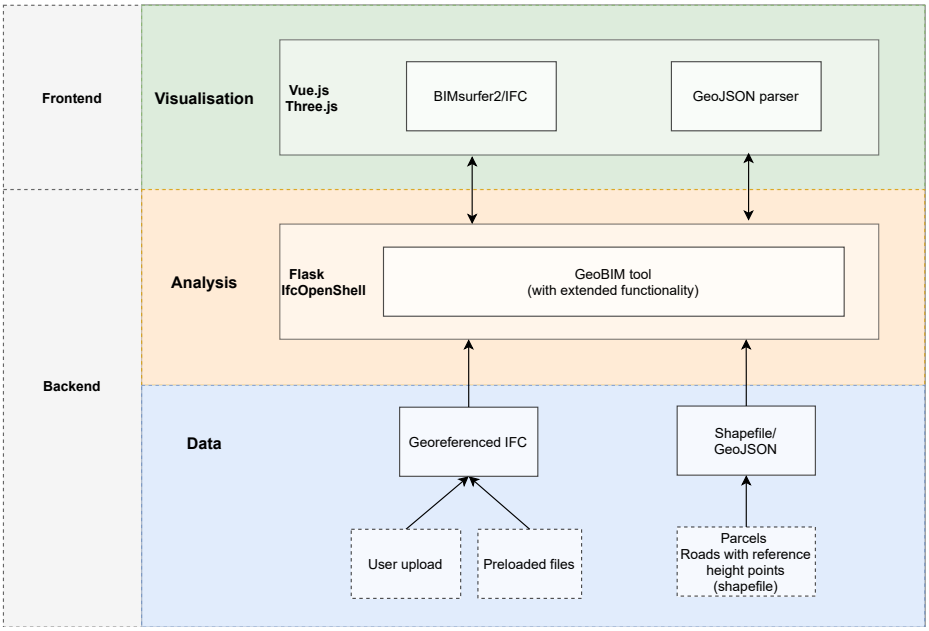


Figure 1: The architecture of the web-based tool

3.2 Input data needed and their requirements

To automate certain checks, integrating GIS datasets with the BIM project is required. This section presents the datasets used for certain checks in the tool.

3.2.1 Parcel dataset

The parcel dataset contains the boundary limits of a piece of owned land. The dataset should represent parcels as polygons. The cleaner the dataset is, the better the automatic detection of a certain parcel is. Polygons within polygons and overlapping polygons should be avoided to circumvent any ambiguities with automatically choosing the correct parcel of the project.

3.2.2 Roads dataset

The roads dataset should represent roads as polygons with their name as an attribute. The name of the roads is important because Dutch regulations are written according to specific roads.

3.2.3 Height points dataset

The height points dataset contains georeferenced 2D points with their height as attribute. These height points should preferably overlap with corner points of the parcel.

3.2.4 IFC file

For the IFC file, storeys should be numerically indexed when they are named, as follows, '00' = GF, '01' = 1^{st} floor, '-01' = 1^{st} underground floor, etc.

3.2.5 JSON-formatted regulations file

This is an example of a machine-readable file that contains regulations and guidelines for building permit checks. It can be used to automatically perform the corresponding analysis and evaluate whether a check passes or fails. We created this ourselves, but there is also the idea to do it in the Gherkin language (see Moult and Krijnen [2020]).

```
{
  "overhang": {
    "rule1": {
      "streetname": "Hertekade",
      "floor": 1,
      "maxMetres": 5
    },
    "rule2": {
      "streetname": "Boompjes",
      "floor": 12,
```

```
      "maxMetres": 3
    }
  },
  "overlap": {
    "rule1": {
      "floor": 1,
      "maxPercentage": 89
    },
    "rule2": {
      "floor": 12,
      "maxPercentage": 92
    }
  },
  "height": {
    "maxMetres": 103.47
  }
}
```

3.3 GeoBIM analysis

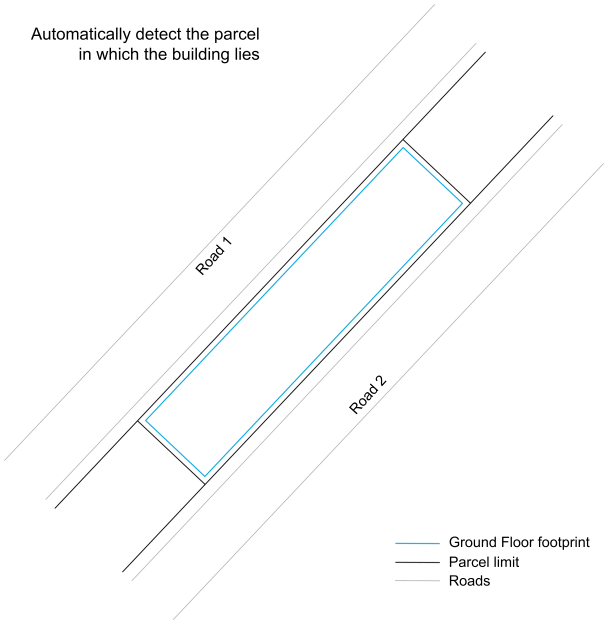
Three new checks are added to the tool: the road overhang check, the height check and the parcel limit check.

3.3.1 Road overhang check

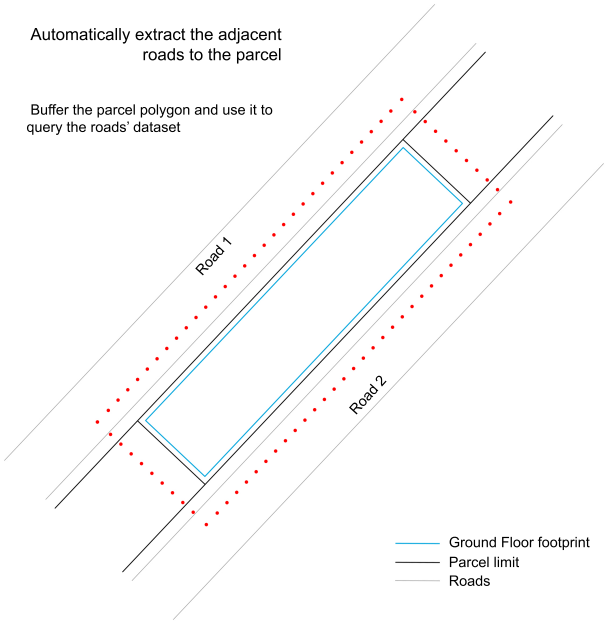
To perform this check, the parcel of the project, the roads adjacent to the parcel and the guidelines attached to those roads are needed. First, the GF footprint is used to automatically determine the parcel that contains the project (Figure 2a). The parcel's boundary is buffered and used to automatically detect the adjacent roads from a roads' dataset (Figure 2b). Once adjacent roads are detected, each floor of the building, starting from the first floor, is simplified into its Oriented Bounding Box (OBB) (Figure 2c). The normal of each vertical side of the OBB is computed and is checked against the adjacent roads (Figure 2d, 2e, 2f, 2g).

If the normal intersects a road, the name of the road is checked for the admissible overhang and the distance from the furthest 2D point of the tested vertical side to the GF is calculated (Figure 2d, 2e). Those two values are then compared to see if the side of the floor passes the overhang check or not . A side is marked in green if it passes the check and in red if it fails the check (Figure 2h).

If the normal does not intersect any road, it is marked in black as a side not facing a road (Figure 2f, 2g).

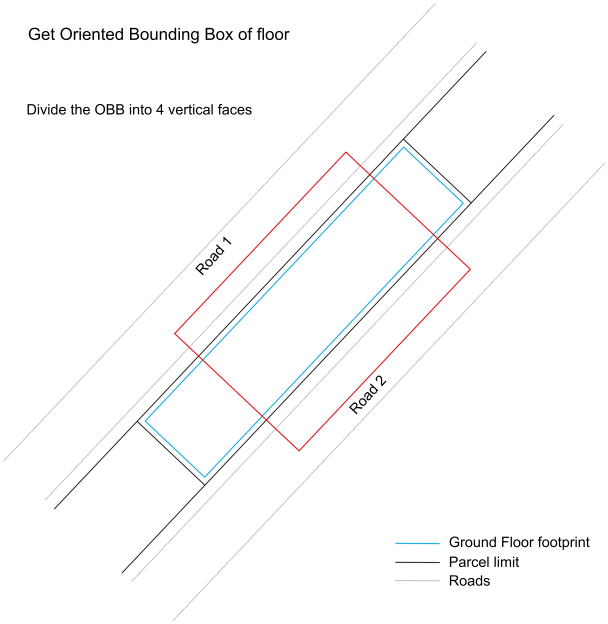


(a) Detecting the parcel of the building

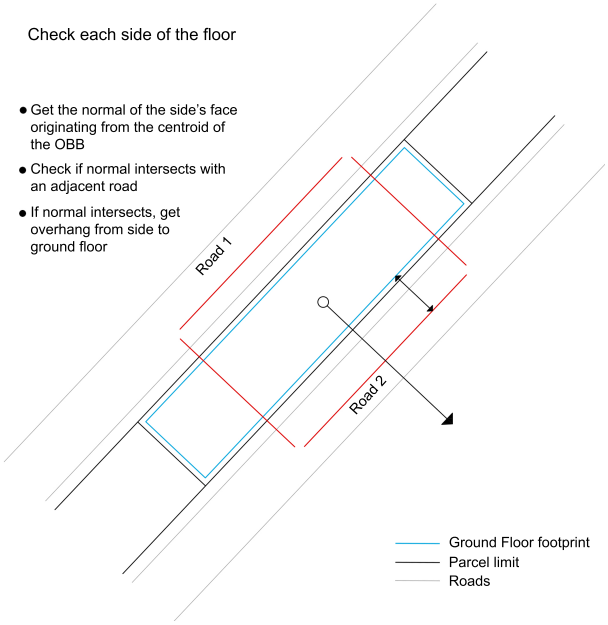


(b) Detecting the adjacent roads to the parcel

Figure 2: Steps to automatically check the overhang over roads

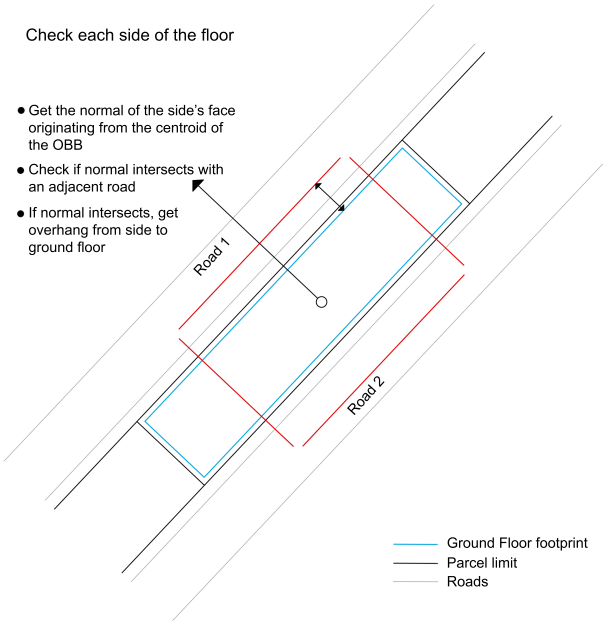


(c) Simplifying a floor into its OBB

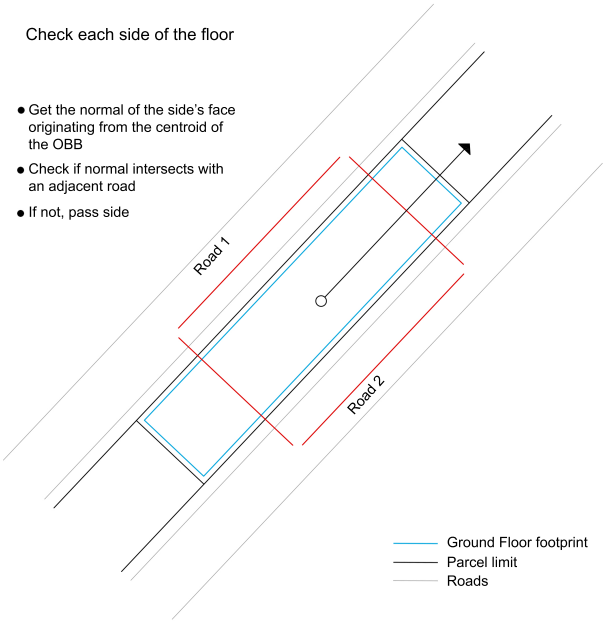


(d) Checking if a side overhangs over a road - case that does overhang

Figure 2: Steps to automatically check the overhang over roads

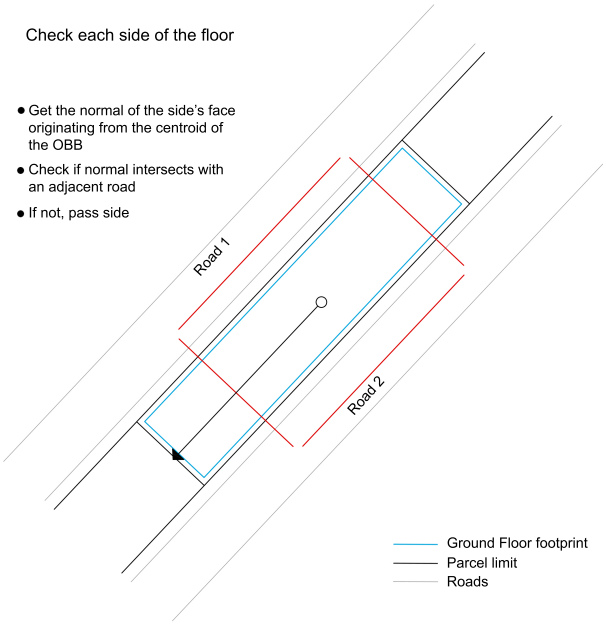


(e) Checking if a side overhangs over a road - case that does overhang

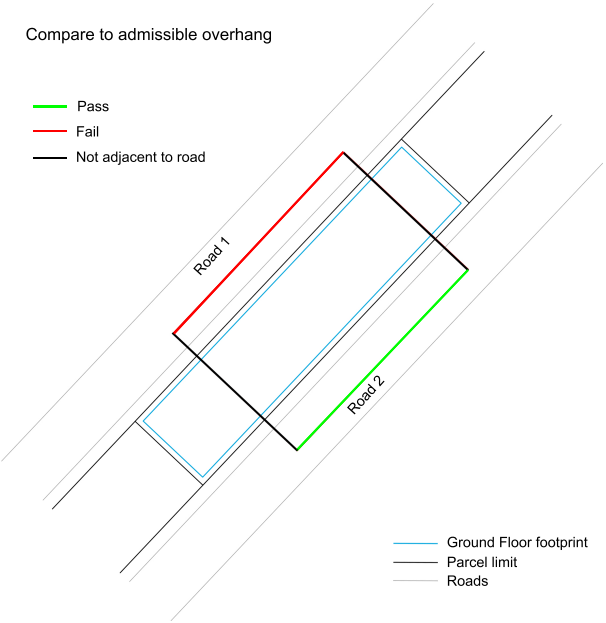


(f) Checking if a side overhangs over a road - case that does not overhang

Figure 2: Steps to automatically check the overhang over roads



(g) Checking if a side overhangs over a road - case that does not overhang



(h) Checking if sides pass or fail the regulation's guideline

Figure 2: Steps to automatically check the overhang over roads

3.3.2 Height calculation from road levels

To perform this check, the parcel, the height points of roads and the maximum allowed height should be given. First, the overall height of the building is computed from the IFC file. Then, the parcel of the project is detected to filter the corresponding height points from the height points dataset. Afterwards, the highest road level is automatically determined from the filtered height points. The overall height is adjusted according to the highest road level (Figure 3a).

This means that if the highest road level is lower than the entrance level of the building, then the difference between the entrance level and the highest road level is added to the overall height. Otherwise, the difference between the entrance level and the highest road level is subtracted from the overall height. The final adjusted height is then compared to the maximum allowed height (Figure 3b) and the result is given as a 'Pass' or 'Fail' and the maximum allowed height is visualised as a plane in the tool with a green color if it passed the check or a red color if it failed the check.

3.3.3 Parcel limit check

To perform this check, the parcel of the project is needed. First, the GF footprint is used to automatically determine the parcel that contains the project. Then, the GF footprint is checked against the parcel polygon. The result is either a 'Pass' or 'Fail' (Figure 4).

4 Conclusion

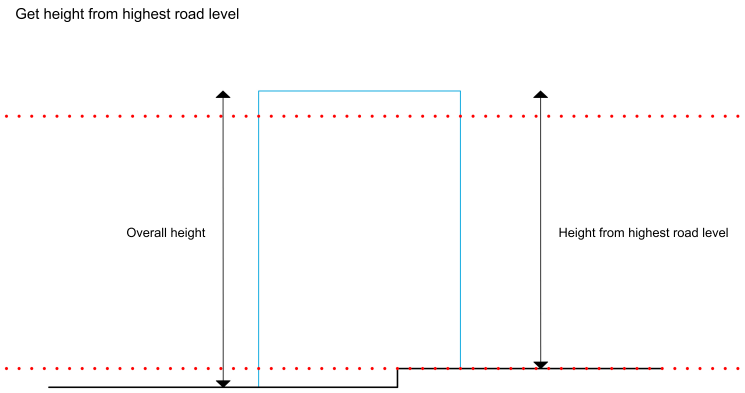
This document presents the second phase of automating building permit checks. A web-based tool was implemented to visualise the results of several checks and analysis. Three new checks, that make use of GIS datasets, were added to the initial tool. These checks detect the parcel that contains the building to be checked, the adjacent roads to that parcel and the height of those roads. Consequently, these new features support fully-automated GeoBIM analysis. Testing the tool with a specific case in Rotterdam using municipality datasets and a real full project helped deduce some requirements for the input data that make the checks faster and less prone to errors.

The outcomes of the second phase show that it is possible to use GIS datasets to automate the checks. This would be helpful to support municipality officers and make some checks faster and easier to evaluate with the visualisation in the tool.

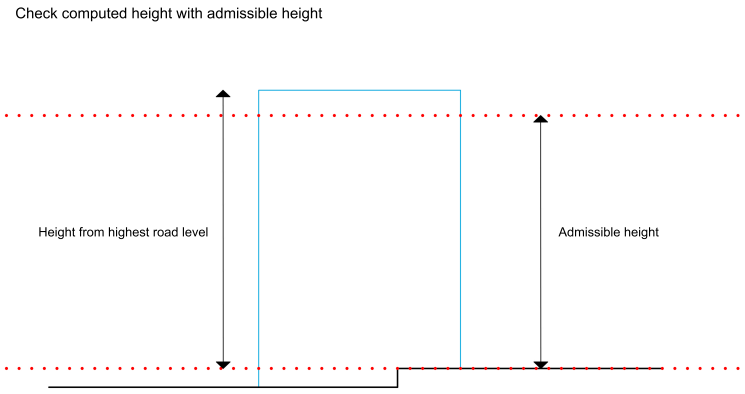
4.1 Future work

The tool can be further developed by adding new features and fully automating more analysis.

- Required number of parking places analysis (on which there is an ongoing thesis in our group);



(a) Getting the height of the building from the highest road level



(b) Compare computed height to admissible height

Figure 3: Steps to automatically check the height from roads

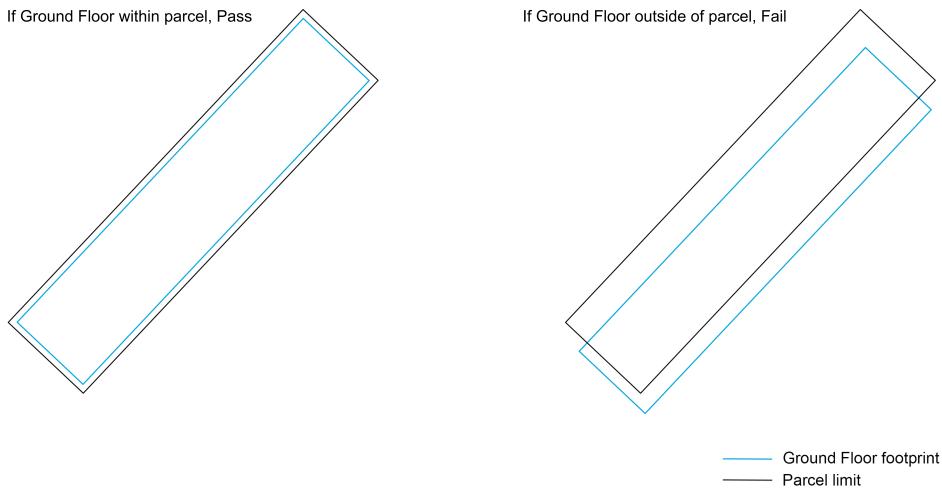


Figure 4: Steps to automatically check whether the building's GF is within the parcel boundary

- Automatic reading of building permit documents ("toepasbare regels Omgevingswet");
- Automatic retrieval of apartments and parking places of the BIM;
- Conduct the analysis using the 3D alpha shape of the building or the storeys;
- More performant viewer and analysis, better result visualisations;
- Have the roads dataset in 3D in the viewer and place the building correctly relative to those roads;
- Use an API (BGT API or internal Rotterdam one) to retrieve geoinformation (rather than shapefiles);
- Other applications? For example flood analysis, noise calculations, etc.

References

Krijnen, T. (2021). ifc-pipeline. `https://github.com/AECgeeks/ifc-pipeline`.

Moult, D. and Krijnen, T. (2020). Compliance checking on building models with the gherkin language and continuous integration. In *Proceedings of the EG-ICE 2020 Workshop on Intelligent Computing in Engineering*. Technische Universität Berlin.

Noardo, F., Wu, T., Ohori, K. A., Krijnen, T., and Stoter, J. (2020). *Investigating the Automation of the Building Permits Issuing process through 3D GeoBIM information - Final Deliverable.*

Vitalis, S. (2021). ninja. `https://github.com/cityjson/ninja`.