Height Inference for all US Building Footprints in the Absence of Height Data

Imke Lánský
Student #4973372

1st supervisor: Hugo Ledoux
2nd supervisor: Balázs Dukai

Date P2: 13/01/2020

1 Introduction

3D city models are used for various analysis applications in different domains (Biljecki et al., 2015). Some of these applications are only based on geometry, while others also include semantic information or even add external data and domain-specific extensions (Ross, 2010). Noise simulations, energy demand estimations and visibility analysis are examples of applications of 3D city models. The 3D city models itself are often generated using building footprints and height data. Nowadays, the building footprints are widely available as open data through government data-portals or as volunteered geoinformation (VGI) (Hecht et al., 2015), but acquiring the height data is still a time-consuming and expensive task, as they are frequently obtained from Lidar (LIght Detection And Ranging) and photogrammetry.

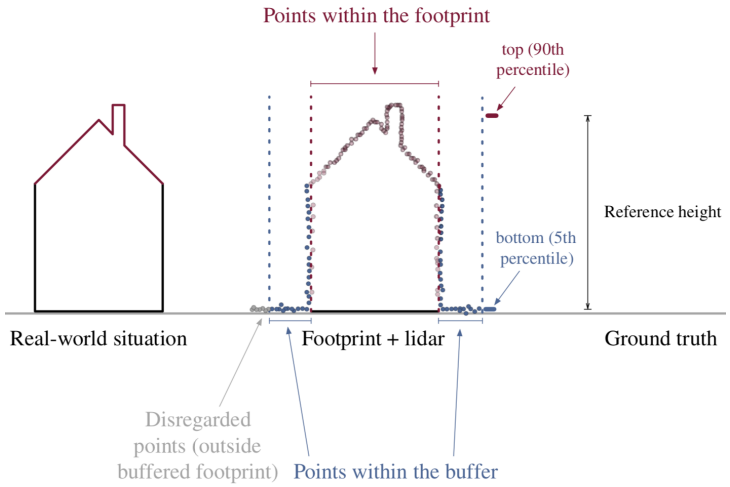


Figure 1: Using Lidar data to compute building heights. *Source*: (Biljecki et al., 2017).

Even if height data is available, it is not always suitable for generating 3D city models. From Lidar data, often only block models can be generated, because not enough information is present to model for example dormers or chimneys. These block models are generated by using the points that fall inside the building footprint, and the footprint is then extruded to the computed height (see Figure 1). The height of the building depends on the selected roof height reference, the height-percentile. A problem that can arise with Lidar data is the 'mismatch' between building footprints and the data points, leaving certain footprints without or with outdated data.

Other data sources might also not always be of high enough quality or resolution to construct 3D city models. An example is the Shuttle Radar Topography Mission (SRTM) dataset, which provides worldwide data coverage in the form of a digital elevation model (DEM), all free of charge. The data has a coarse resolution (e.g. 30m) and is of insufficient accuracy to be used for producing 3D city models (Smith and Sandwell, 2003). In Africa, it is often the only source of elevation data, limiting the possibilities of generating 3D city models in these areas.

To overcome these problems, experiments with machine-learning techniques have been performed to infer building heights in the absence of elevation data (Biljecki et al., 2017). For the United States of America (USA) there is the *Open City Model*; a dataset that contains 3D city models for all 50 states and that contains roughly 125 million buildings (BuildZero, 2019). However, the producers of the data are not open about the techniques that are used to generate the results. Only a statement is made about that the footprint area and the building location are used for the height estimation. The accuracy of these estimations appears to be low, e.g. many buildings are assigned similar heights. Serious doubts arose about what machine-

learning techniques are used, or if they are used at all. Therefore, this thesis will focus on applying different machine-learning techniques to infer the building heights for all building footprints in the USA. The goal is to improve the accuracy of the OCM model and to provide a method for inferring building heights that is not only applicable to the US but can also be adopted in other areas in the world. The latter aspect is especially interesting for the areas where there is a scarcity of accurate enough elevation data. Lastly, the algorithm should be scalable (e.g. have an efficient run-time) since it will be applied to large and diverse regions.

2 Related Work

2.1 Formats & Standards for 3D City Models

3D city models can be stored in different exchange formats. CityGML is an XML-based format designed by the Open Geospatial Consortium (OGC) to create a common definition of the entities, attributes, and relations present in 3D city models (Gröger et al., 2012). It is based on the Geography Markup Language version 3.1.1 (GML3). The data files are often verbose, of a complex and a hierarchical structure, and not very well suited for web applications. With these issues in mind, CityJSON was developed. It provides a JSON encoding for the CityGML data model that is easier to parse and allows for higher data compression than the XML-based format of CityGML (Ledoux et al., 2019).

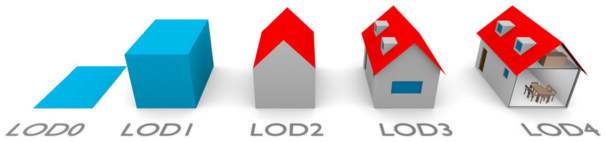


Figure 2: The five LODs as specified by the OGC for CityGML 2.0. *Source*: (Biljecki et al., 2016).

Both encodings support different levels of detail (LOD) to allow the same 3D city objects to be used for different applications; the same object can be represented in different LODs simultaneously (Gröger et al., 2012; Ledoux et al., 2019). Figure 2 shows the five different LODs as defined in CityGML 2.0. When increasing the LOD, both the geometric detail and semantic complexity are increased (Biljecki et al., 2016).

Figure 3: Seven different LOD1 block representations for the same building (in LOD3) when different height references are used. *Source*: (Biljecki et al., 2014).

When geometric data is represented with an LOD1 block model, different height references can be used for its roof surface (Biljecki et al., 2014). CityGML does not standardise how to store the geometric reference of a model; there is no metadata available for expressing the different options. Figure 3 shows seven different LOD1 block models for the same building, according to different height references. One can, for example, take the height at the top of the roof, also include the constructions on the roof such as chimneys, or decide to take

a median height at half of the height of the roof. Biljecki et al. (2014) show that the chosen height reference can greatly influence the results of the analysis in certain cases (e.g. volumetric computations). It can also affect the Root Mean Square Error (RMSE) of the 3D city model; if the chosen height reference does not match the ground truth height measurements well, the RMSE will increase.

In this thesis, LOD1 reference models and training data will be generated from point clouds, where the points inside the building footprints are used to compute the building height (see Figure 1). The height reference that is used for the building roofs is thus of high importance and can (significantly) impact the final results of the building height estimations.

2.2 Machine-Learning for 3D City Models

Biljecki et al. (2017) describe how to use the Random Forest (RF) regression machine-learning technique to infer building heights for 200,000 buildings in the city of Rotterdam, the Netherlands. The attributes (features) are extracted from cadastral and statistical data and the geometry of the building footprints. The former two are available through external data sources, while the latter is always available, as they are derived from the 2D geometries of the footprints. Different combinations of features are used to cover a wide range of possible real-world scenarios, and each feature's importance is computed after the regression algorithm is run. The method shows promising results when only the geometric features are used; a mean absolute error (MAE) of 1.8 metres is achieved. Point cloud data is used to generate ground truth models for the building heights, which are used to analyse the results.

A similar method is applied by Anh et al. (2018) to the city of Hanoi, Vietnam. The same geometric properties as proposed by Biljecki et al. (2017) are used, and the building usage is added as an extra feature. The diversity in features is therefore much lower. Since no point cloud data is available for the study area, actual field surveys were conducted to obtain ground truth data for the building heights. Cross-validation and grid-search techniques were used to adjust the model parameters and to make it more accurate. However, with an MAE of 7.12 metres, the performance of the predictor model is less accurate than the one of Biljecki et al. (2017).

These 3D city models can be further enriched (Biljecki and Sindram, 2017; Henn et al., 2012; Biljecki and Dehbi, 2019). The number, and complexity, of the features required for the enrichment process, depends highly on the use-case. These models, if accurate enough, can be used for various applications and different types of analyses.

2.3 Knowledge Gap

This research will build upon this previous work and extend it in the following ways:

Scaling of the machine-learning algorithm is necessary to deal with the millions of building footprints. This also introduces the challenge of dealing with different morphologies of build-up areas. Now, not only cities are considered but also rural areas. Different ways to distinguish between these morphologies will be researched.

The number of geometric features that can be derived from 2D building footprints will be exploited further to try and improve the results shown in previous research.

Lastly, different roof height references will be considered. 3D city models that were generated based on different height percentiles will be compared to the 3D city models generated with machine-learning techniques. This can provide insight into what range of the height percentiles the estimated heights often lie.

3 Research Questions

The main research question for this thesis is:

Can the 125 million US building footprints be assigned a height without making use of height data, and what accuracy can be achieved?

The goal of this research is to implement a machine-learning algorithm that can infer heights for 2D building footprints, preferably based on only their geometric properties. The estimations should be of high enough accuracy to be useful for further analysis applications. To achieve these goals, the following sub-questions are defined:

a) *What methods can be used to assess the accuracy of the building height estimations? And when are the estimations deemed accurate enough?*

b) *Are the geometric properties of building footprints as training features sufficient for meeting the accuracy requirements?*

c) *What other features, besides the geometric properties of the building footprints, can be used in the machine-learning algorithm to estimate building heights? And does the inclusion of these features, even if they are incomplete, improve the accuracy of the estimations?*

3.1 Scope

This thesis will focus on the height inference of building footprints in the USA using different machine-learning techniques, including Random Forest Regression, Support Vector Regression with linear kernels, and Multiple Linear Regression. These methods provide a balance between the results they generate and the runtime needed to perform the estimations. The training features for the models are based on the geometric properties of the building footprints. The resulting models will be in LOD1, i.e. the use of block models where no roof structures are considered. If the accuracy of the results and the algorithm performance are satisfying enough, an extension to also include Canada could be made. Lastly, if time allows it, a trial can be performed to explore the possibilities of shadows in satellite imagery as an extra feature for learning. The focus will then be on the city of Rotterdam in the Netherlands, because of the availability of high-quality data for this area.

4 Methodology

The method to infer the heights for all building footprints requires several steps. Figure 5 displays a flowchart of the proposed steps, including pre-processing of the data, feature extraction and the machine-learning algorithm for the height inference itself.

4.1 Data Pre-Processing

The goal of the data pre-processing step is to create datasets suitable for input in a database. The data of the `USBuildingFootprints` dataset lacks unique `IDs` for the building footprints. Therefore the first data pre-processing step involves generating these unique `IDs`. One option is to use the abbreviation of the state name where the footprint is located, i.e. `NY` for New York, together with a unique number, e.g. `NY_1234`. Another interesting option is the Unique Building Identifier (UBID) designed by the US Department of Energy (DOE). It is the "north axis-aligned 'bounding box' of the building's footprint represented as a centroid along four cardinal extents" (Wang et al., 2019), see Figure 4. A possible difficulty with implementing

this type of identifier is that the north axis-alignment information is not always available. It requires the use of a CRS with coordinates in longitude and latitude.

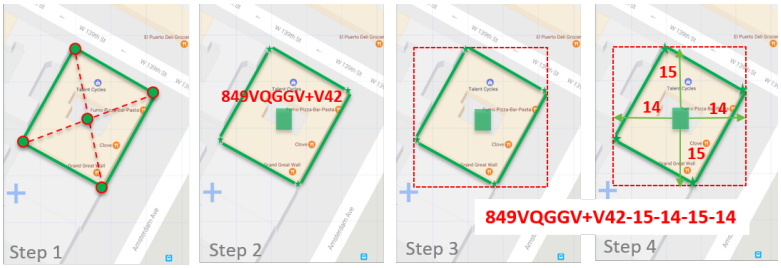


Figure 4: The steps in constructing the Universal Buiding Identifier. *Source:* `energy.gov`

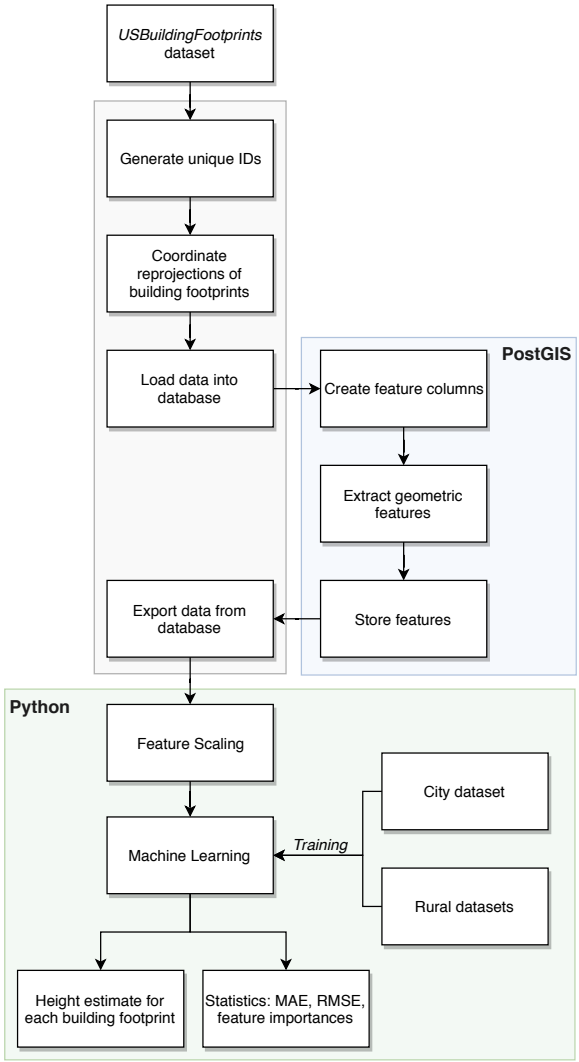


Figure 5: The steps of the methodology; from downloading the data and data pre-processing, to feature extraction and running the machine-learning algorithm.

The next step is to re-project the building footprints to a coordinate reference system (CRS) that uses Cartesian coordinates, instead of longitude and latitude as is used with WGS84. This is required because of the spatial operations that will be performed on the data, such as computing the area of the building footprints. One option is to use the *Universal Transverse*

Mercator (UTM) CRS, which divides Earth into 60 different zones, each of which is of 6° of longitude in width. Every location on Earth has a zone and an x,y-coordinate in that plane. Since the USA is a wide-spread country, it covers many UTM zones (see Figure 6). One state can be part of multiple UTM zones, making it impossible to directly re-project all building footprints located in the same state. Per building footprint, in a state dataset, it must then be checked to which zone it belongs and the state dataset should be split accordingly to only contain data of the same UTM zone.

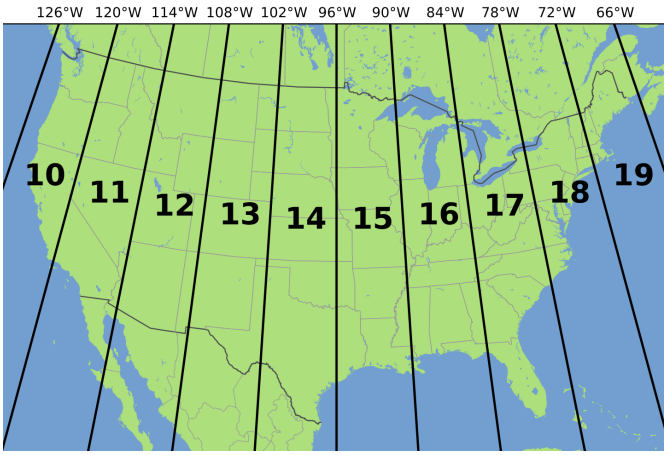


Figure 6: The different UTM zones that cover the USA. *Source:* Wikipedia.

A more user-friendly method would be to re-project all coordinates to a US-wide CRS, provided that it minimises the distortion and that the coordinates are (or can be transformed into) Cartesian coordinates. Two possible options include the *Albers Equal Area Conic* and the *Lambert Conformal Conic* projections. The first minimises the shape and linear scale distortion between the two standard parallels, and the latter portrays shapes more accurately than areas if they are along middle latitudes (Kennedy and Kopp, 2000). The State Plane Coordinate System (SPCS), which divides each state into six zones and uses Cartesian coordinates, makes use of the Lambert Conformal Conic projection for its mapping along the east-west axis (U.S. Geological Survey, 2017).

4.2 Random Forest Regression

For the machine-learning method, Random Forests (RF) can be used, which is a supervised learning method for both regression and classification problems. Supervised learning methods require the data to have both *features* and *labels*. In the case of building height prediction, the features describe characteristics of the 2D building footprints (e.g. area, number of neighbours, perimeter, etc.), and the labels are the actual building heights. Since this problem deals with numerical values, the focus will be on RF regression.

In RFs, many decision trees are generated (see Figure 7). Unlike splitting the nodes based on the best split among all variables (as in standard trees), the RF chooses the best split from a random subset of predictors chosen at that node (Breiman, 2001). It also makes use of averaging methods, making it robust against over-fitting. A strong point of RFs is the computation of the feature importance, which can be complex to calculate as it depends on the interaction with other variables (Liaw and Wiener, 2002). The importance of a feature is estimated by looking at how much the prediction error increases when the data for that feature is changed, while all other features are left unchanged. The provided feature ranking is useful for designing predictive models; only include features that are important and thus minimise the number

of features used in total (Grömping, 2009).

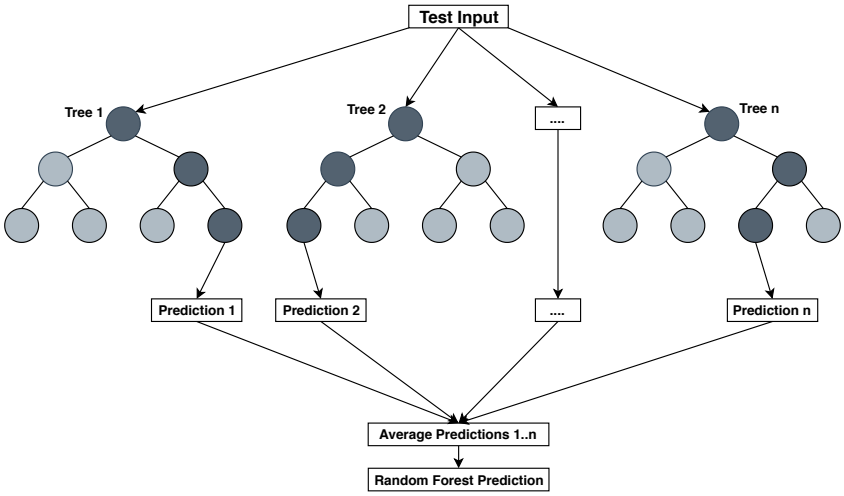


Figure 7: Decision trees that are generated by the RF regression method for prediction.

The features can be on different scales; the area of a building footprint can have values greater than one hundred, while the shape complexity is a number between zero and one. These large differences can cause certain features to dominate the prediction. Feature scaling normalises the range of the features. Several methods are available, but in this research, we will standardise features by removing their mean and then scaling them to have unit variance. This is shown in Equation 1, where x' is the normalised value for the feature, $\bar{x}$ the mean of the feature vector, and σ the standard deviation.

$$x' = \frac{x - \bar{x}}{\sigma} \tag{1}$$

4.2.1 Alternatives to RF Regression

RF regression is not the only possible regression machine-learning technique that can be used to infer building heights. Two other options are Support Vector Regression (SVR) and Multiple Linear Regression (MLR).

SVR makes use of a loss function and a distance measure. The method requires prior knowledge about the underlying distribution of the data, and the loss function is selected based on this knowledge (Gunn, 1998). When non-linear kernels are used in the model, the complexity for the data fitting is more than quadratic with the number of samples, making it unsuitable for datasets with more than a couple of ten-thousand samples (TheKernelTrip, 2018). This introduces a problem for the USA, as it contains over 125 million building footprints. Linear kernels are faster, but might not always fit the data. However, the results of a linear kernel with the ϵ-insensitive loss function do show that it can be an interesting option to investigate further alongside RF regression, see Table 4 in Section 5. The ϵ-insensitive loss function implements a constant that determines a trade-off between the amount up to which deviations larger than ϵ are tolerated and the flatness of the function (Smola and Schölkopf, 2004), see Figure 8.

Some other differences from RF regression are that SVR does not include the importance of the different features and it cannot handle categorical data. All data must be converted to continuous numerical data before it can be used.

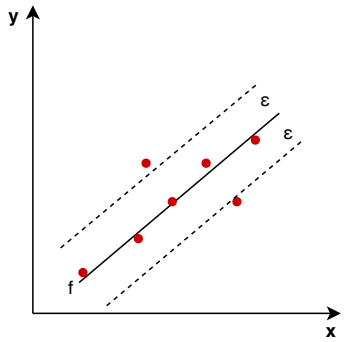


Figure 8: Epsilon loss; the function f with a tolerance value ϵ.

In linear regression, the relationship between an independent and dependent variable is plotted. With multivariate linear regression multiple linear regression models are used, where there are multiple independent variables (Tipireddy, unknown). In this research, the independent variables are the different features (e.g. footprint area, complexity, etc.), and the dependent variable is the building height. It is assumed that there is a linear relationship between the two. As with SVR, the data must be continuous and no feature importances are computed. Table 4 in Section 5 shows that MLR can be an interesting option, but its MAE is higher than the one for RF regression and SVR with a linear kernel for the same test set-up.

4.3 Feature Extraction

4.3.1 Geometric Features

An important part of this thesis is extracting the different geometric features from the 2D building footprints. These features are the input for the RF regression model. Table 1 shows different features that can be extracted from the 2D building footprints. The fact that no additional data sources are needed for deriving this data is a big advantage. The footprint area, complexity and the number of neighbours were implemented in the research of Biljecki et al. (2017). This research includes extra features to see if it makes the prediction model better, resulting in more accurate building height estimations.

	Feature	Description	Computation
1.	Area	The area of the building footprint	-
2.	Complexity	The Normalised Perimeter Index (NPI)	$\frac{2\sqrt{\pi A}}{P}$
3.	Number of neighbours	Buildings within a range of 100 metres of the footprint	Centroid distance
4.	Number of adjacent objects	Buildings within 1 metre of the footprint	Buffers
5.	Length	Longest edge of MBR	-
6.	Width	Shortest edge of MBR	-
7.	Slimness	Ratio of the sides	$\frac{F_{length}}{F_{width}}$
8.	Number of vertices	Total number of vertices in the footprint	-

Table 1: The features that can be derived from the 2D geometries of the building footprints.

The *area* of the building footprint is the surface that the building covers. It is used to investigate if the footprint area is proportional to building height.

The *footprint complexity* is defined by the Normalised Perimeter Index (NPI), which uses the equal area circle and the perimeter of the polygon; $\frac{2\sqrt{\pi A}}{P}$. Here, A is the area and P the perimeter. It can be used to identify features with irregular boundaries because it compares the input perimeter to the most compact polygon with the same area (the equal area circle). A high NPI value means fewer irregularities in the polygon shape than a low NPI value. The normalisation makes the measure independent of the size of the polygon (Angel et al., 2010).

The *number of neighbours* might give information about the type of area that the building is located in. It is expected that in rural areas the number of neighbours is lower than in a city. A distance of 100 metres between buildings is selected. For each 2D footprint, its centroid is computed, and the number of other centroids within this 100-metre radius defines the number of neighbours of a building. It must be noted that taking the centroid of a building with a big footprint might affect the results because the distance from the centroid to the footprint edges is also bigger than for buildings with smaller footprint areas. However, computing the number of neighbours using buffers is much more computationally expensive than the centroid method, making it infeasible for the 125 million building footprints in the US.

A similar measure to the number of neighbours is the *number of adjacent objects*, which defines the number of footprints that are directly touching another footprint. As before, it is expected that this number is higher in cities than in rural areas, since for the latter buildings are more likely to be spread out over a larger area. The computation of this feature does require buffers; for each footprint, a one-metre buffer is generated and intersected with nearby buildings.

The next two features include the footprint *length* and *width*. These values are derived from the minimum bounding rectangle (MBR) of the footprint, where the longest edge in the MBR represents the length of the footprint and the shortest edge the width. Then, the *footprint slimness* is computed as the ratio between the length and width of the footprint.

Lastly, the *number of vertices* that make up the building footprint are counted. More vertices might also provide an indication of how complex the footprint shape is.

4.3.2 Non-Geometric Features

Besides the geometric properties that can be directly derived from the 2D building footprints, other data sources can provide additional information about the buildings

OpenStreetMap (OSM) provides data exports of building footprints where the `building`-tag can indicate the type of building. This field is not always filled, and sometimes it only provides 'yes' to indicate that it is indeed a building. The `amenity`-tag can provide extra information about the specific building types. Lastly, there is the `other tags`-tag, that might also include information about the building type. Data describing only one feature is spread out over multiple attributes, and often they are not filled.

Cadastral data is another data source that can provide additional features. Biljecki et al. (2017) used features such as the *building use*, *year of construction* and *number of storeys above ground* for the Netherlands. While in the Netherlands this data is readily accessible, in the USA cadastral data is spread out over local governments (Coalition of Geospatial Organizations, 2018). A national database comprising all relevant information about public and private parcels is not available yet. This characteristic of the US national spatial infrastructure makes it difficult to incorporate such information. Some states provide state-wide databases with

cadastral data, which could be an alternative option.

Adding all these extra features to the prediction model can be useful and might result in better height predictions, even if the information is incomplete.

4.4 Algorithm Scaling

Scaling of the algorithm is an important aspect of this research, as we need to estimate the height for over 125 million building footprints. The scaling involves mostly what kind of trained network(s) to use for the height predictions. The two main options are the following: input the building footprints into a network trained on both rural and city areas, or create separately trained networks for the rural and city areas.

The first option considers all training data and creates a random forest or a fitted function based on the characteristics found in both the rural and city area data. With this option, it is interesting to test how a network trained on rural data predicts the heights for city areas and vice versa. This can provide insights into the kind of errors that may arise. When a combined network is used, the network should be able to distinguish well between the two different types of morphologies present in the data, which might be difficult.

The second option includes two prediction networks, trained on the two different area morphologies. Once these networks are generated, a metric is needed to identify in what type of environment the building footprint is located. A binary feature can be added to each building footprint, providing true or false for city or no city respectively. For each building, a radius of a few kilometres around its centroid can be used to find the buildings within this radius. The ratio between the area covered by buildings and the total area of the radius can then provide an indication of the type of environment; for rural areas, this ratio should be lower than for cities.

In terms of scaling for computation time of the algorithm, the number of jobs that can be run in parallel can be adjusted as a hyperparameter of the random forest regression model. More jobs mean that the data fitting and height prediction is parallelised over the different trees, resulting in faster run-times. In theory, all processors on a system can be used. For MLR a similar hyperparameter is present to adjust, but for SVR no such option is available.

5 Preliminary Results

The first steps of this research included exploring the possibilities of `scikit-learn` for `Python` based on the research performed by Biljecki et al. (2017). A subset of 14,189 buildings for the city of Rotterdam is extracted from the 'Basisregistratie Grootschalige Topografie' (BGT). The building footprints are then assigned height values with the `3dfier` tool, where the 'Actueel Hoogtebestand Nederland 3' (AHN3) is used as point cloud input. The 2D building footprints are loaded into the database and the area, the number of neighbours and shape complexity are computed.

	Feature	**Importance**
1.	Area	0.489
2.	Complexity	0.306
3.	Number of neighbours	0.205

Table 2: The feature importance computed based on data for the city of Rotterdam.

The RF regression network is trained with 60% of the enriched data, and the other 40% is used as test input for predicting the building heights. A total of 1000 trees are used in the RF; more trees will result in better predictions. It takes 20.34 seconds to train the RF prediction model and predicting the values for the test data only takes around 1.06 seconds. Both these values are averaged over five runs. Table 2 shows the importance of the three features used in the model; the footprint area influences the height prediction the most, followed by the shape complexity and the number of neighbours. The height predictions have an MAE of 4.23m and an RMSE of 10.16m. Biljecki et al. (2017) reached an MAE of 1.8m and an RMSE of 3.5m with the same three geometric features. Possible explanations for this big difference in accuracy include the use of other model hyperparameters, such as the number of trees, and a difference in the size and coverage of the (training) dataset.

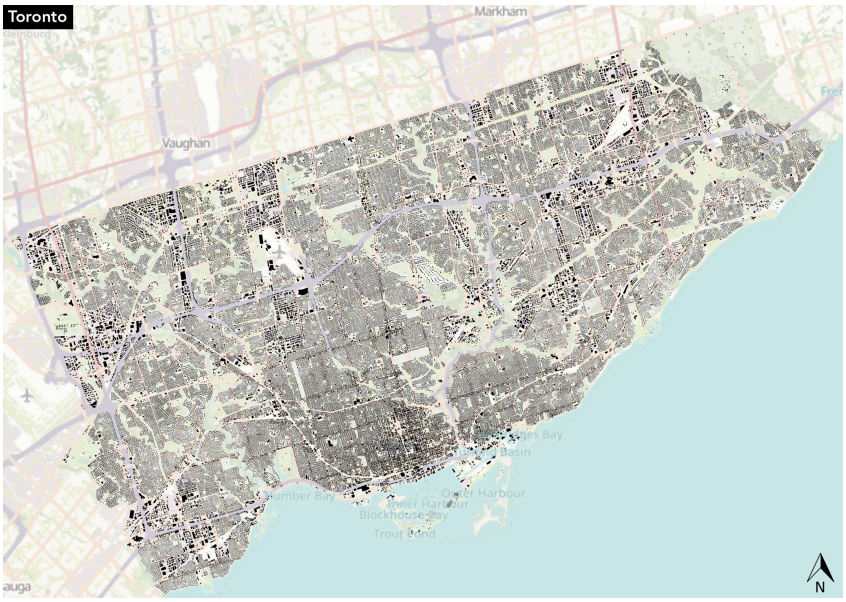


Figure 9: Building footprints for the city of Toronto.

The next step is to look at data that could be suitable for training the RF regression network for predicting building heights in the USA. Ideally, this data already contains height values for the building footprints. For the city of Toronto (Canada) such a dataset is available (City Planning Toronto, 2019). The source of the height data, e.g. Lidar or photogrammetry, is indicated in a separate attribute. The average height is chosen as the building height for training the prediction model, which is the maximum average height of the building footprint in metres according to the dataset metadata.

Even though Toronto is not in the US, it is close to the border and the city shows similarities to other US cities. Figure 9 shows the coverage of the 420,852 building footprints in this dataset.

	Feature	**Importance**
1.	Number of neighbours 25m	0.203
2.	Number of neighbours 50m	0.212
3.	Number of neighbours 75m	0.260
4.	Number of neighbours 100m	0.324

Table 3: The feature importance for the number of neighbours based on different distance radii for the city of Toronto.

First, the 'optimal' distance for the number of neighbours query is determined based on the feature importance of different distance radii. 20% of the data is used for training, and 80% is used for testing the prediction model. A total of 500 trees are used in the random forest. Table 3 shows the results for distances of 25m, 50m, 75m and 100m. The 100m distance radius has the highest importance in the prediction process of the building heights and is therefore used in combination with the other geometric features.

Method	Training time [s]	Predicting time [s]	MAE [m]	RMSE [m]
RF	171.74	47.91	2.66	7.47
SVR	5.46	0.004	2.42	7.97
MLR	0.009	0.003	2.79	7.79

Table 4: Results for three different machine-learning techniques on the city of Toronto dataset. All values are averaged over five runs.

	Feature	Importance
1.	Area	0.434
2.	Complexity	0.392
3.	Number of neighbours	0.173

Table 5: The feature importance computed based on data for the city of Toronto with the RF regression method.

Next, the building heights are estimated based on the footprint area, shape complexity and the number of neighbours for each building to make the results comparable to Rotterdam. Besides RF regression, also the SVR and MLR are run for the Toronto dataset. The results are shown in Table 4; all values are averaged over five runs. The feature importances, computed by the RF regression method, are shown in Table 5. The importance ranking of the features is the same as for the Rotterdam dataset, but the shape complexity plays a bigger role for the Toronto dataset. The RF method has a lower error for Toronto than the Rotterdam dataset. The Toronto dataset contains almost 30 times as many buildings as the Rotterdam dataset; a lot more buildings are used during the training of the RF network. Another explanation for the higher accuracy is a possibly higher diversity in the 'type' of buildings in the training dataset. Even though the error decreased, an MAE of 2.66m is still too high as this is almost one entire floor for a building.

When comparing the three methods, we see that SVR has the lowest MAE, followed by RF regression and MLR. The time needed for training and the predictions is a lot higher for the RF. When running the RF regressor on all processors by changing the number of jobs, the training time goes down to 36.65 seconds, and the prediction time to 13.77 seconds.

It must also be noted that these results do not include the different roof height references for the LOD1 models yet. Therefore, not much meaning can be derived from these preliminary results.

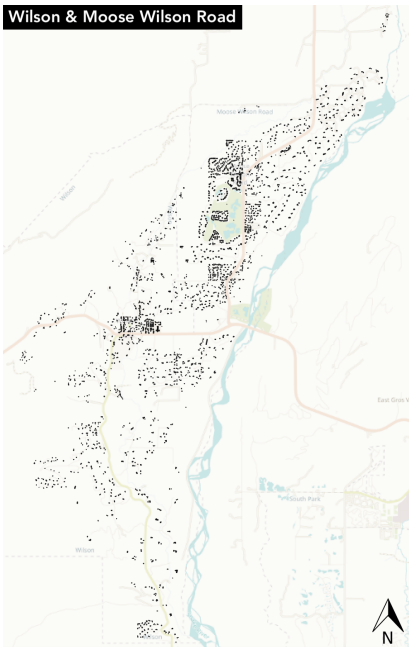
From these preliminary results, it can be seen that the hyperparameters for the different methods should be tuned more. `scikit-learn` provides this option for all three methods.

For RF, the number of trees and the maximum number of features that the RF should consider when splitting a node can be adjusted. More trees result in more stable predictions, but it slows down the computation. Changing the maximum number of features forces the algorithm to choose other splits at the start, resulting in more variation in the trees. In this way,

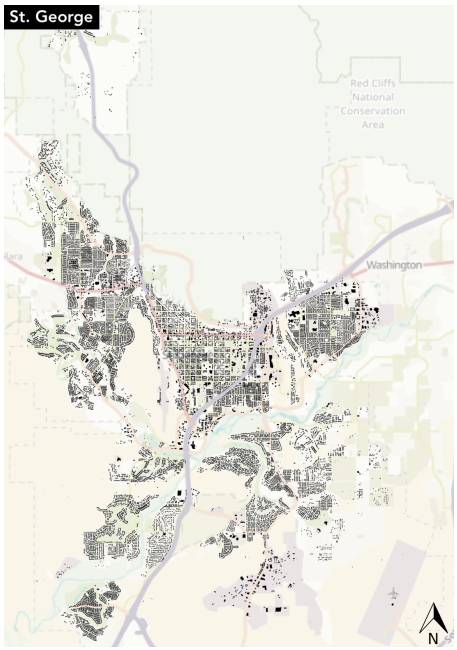
more generalised trees are created with less correlation between them. The higher variety in the trees can increase the prediction performance.

MLR has limited options for adjustments as it is a simple method, but SVR includes the adaptation of the ϵ value for the loss function among others. Since it is mathematically more complex than the other methods, it includes many more hyperparameters that must be tuned.

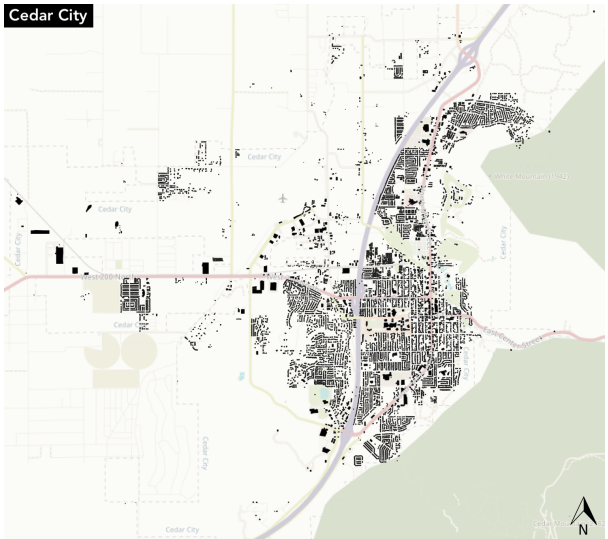
Including more geometric features, or including more diverse training data, can also improve the prediction results. For the latter aspect, rural areas should be included next to the cities. Figure 10 shows the three different areas that were extracted from the `USBuildingFootprints` datasets: Wilson and Moose Wilson Road (Wyoming) with 2227 buildings, Cedar City (Utah) with 8846 buildings, and St. George (Utah) with 26,996 buildings. This makes a total of 38,069 buildings for training data for rural areas.



(a) Wilson & Moose Wilson Road (Wyoming)



(b) St. George (Utah)



(c) Cedar City (Utah)

Figure 10: The building footprint datasets of the rural area training data.

6 Time Planning

The Gantt chart in Figure 11 shows the time planning for the thesis project, including an estimation of the different deadlines for the five Ps.

6.1 Meetings

Every two weeks a one-hour meeting will be held with the first supervisor. This might switch to a weekly meeting of thirty minutes if that seems a better fit. The second supervisor will provide additional guidance and feedback when needed. The co-reader for this thesis is yet to be decided on.

7 Tools and Datasets used

7.1 Tools

Several tools are required to read, pre-process and export the data before it can be used as input for the machine-learning algorithm. Pre-processing of the data requires `Python` and the `Feature Manipulation Engine` (`FME`) by Safe Software Inc. (2019). Afterwards, the data is loaded into a database using the `ogr2ogr` (Warmerdam et al., 2019) or the `pgsql2shp` (Strobl, 2008) command-line tool. The database is extended with `PostGIS`, and its spatial analysis tools are used to extract the geometric features for each building footprint. This pre-processed data is the input to the `Python` program that makes use of the `scikit-learn` library to perform the machine-learning tasks (Pedregosa et al., 2011). `3dfier` is used to create training data for rural areas (3D Geoinformation TU Delft, 2019a). Its other application is to create reference models to check the accuracy of the output of the algorithm. `val3dity` can be used to check the geometric validity of the created city models (Ledoux, 2013). For visualisation of the models, `Azul` (3D Geoinformation TU Delft, 2019b), `FME Data Inspector` (Safe Software Inc., 2019) or `QGIS` (QGIS Development Team, 2019) can be used, depending if it is in `CityGML` or `CityJSON` format. The latter two can also be used for visualising the 2D building footprints.

7.2 Data

The building footprints for the USA are obtained from the `USBuildingFootprints` dataset created by Microsoft (2018). This dataset contains 125,192,184 computer-generated building footprints in the `GeoJSON` data format and covers all 50 states. The training dataset for Toronto is obtained through the open data portal of the city of Toronto (City Planning Toronto, 2019). `Lidar` data is used to compute building heights for training datasets that are not enriched with height attributes. The point clouds are also used to provide ground truth-models to examine the accuracy of the machine-learning algorithm. Both `OpenTopography` and the `USGS` (U.S. Geological Survey) provide open Lidar datasets for the USA (OpenTopography, 2019; U.S. Geological Survey, 2019). Lastly, the `OCM` datasets (BuildZero, 2019) are used to compare their height inference results to the results obtained in this thesis.

Figure 11: Gannt chart showing the time planning and different stages of the thesis project.

References

3D Geoinformation TU Delft. Takes 2D GIS datasets and "3dfies" them by lifting each polygon to its height (obtained with LiDAR), 2019a. https://github.com/tudelft3d/3dfier (accessed: 03.12.2019).

3D Geoinformation TU Delft. 3D city model viewer for Mac, 2019b. https://github.com/tudelft3d/azul (accessed: 03.12.2019).

S. Angel, J. Parent, and D. L. Civco. Ten compactness properties of circles: measuring shape in geography. *The Canadian Geographer / Le Géographe Canadien*, 54(4):441–461, 2010. doi:10.1111/j.1541-0064.2009.00304.x.

P. Anh, C. T. Vu, B. Q. Hung, N. T. N. Thanh, and N. V. Ha. Preliminary Result of 3D City Modelling For Hanoi, Vietnam. In *NAFOSTED Conference on Information and Computer Science (NICS)*, pages 294–299, 11 2018. doi:10.1109/NICS.2018.8606867.

F. Biljecki and Y. Dehbi. Raise the Roof: Towards Generating LoD2 Models Without Aerial Surveys using Machine Learning. In *3D Geoinfo 2019 Proceedings*, 2019. doi:10.5194/isprs-annals-IV-4-W8-27-2019.

F. Biljecki and M. Sindram. Estimating Building Age with 3D GIS. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 4, pages 17–24, 10 2017. doi:10.5194/isprs-annals-IV-4-W5-17-2017.

F. Biljecki, H. Ledoux, and J. Stoter. Height references of CityGML LOD1 buildings and their influence on applications. *Proceedings. 9th ISPRS 3DGeoInfo Conference*, 2014. doi:10.4233/uuid:09d030b5-67d3-467b-babb-5e5ec10f1b38.

F. Biljecki, J. Stoter, H. Ledoux, S. Zlatanova, and A. Çöltekin. Applications of 3D City Models: State of the Art Review. *ISPRS International Journal of Geo-Information*, 4:2842–2889, 2015. doi:10.3390/ijgi4042842.

F. Biljecki, H. Ledoux, and J. Stoter. An improved lod specification for 3d building models. *Computers, Environment and Urban Systems*, 59:25–37, 2016. ISSN 0198-9715. doi:10.1016/j.compenvurbsys.2016.04.005.

F. Biljecki, H. Ledoux, and J. Stoter. Generating 3D city models without elevation data. *Computers, Environment and Urban Systems*, 64:1–18, July 2017. doi:10.1016/j.compenvurbsys.2017.01.001.

L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, October 2001. ISSN 1573-0565. doi:10.1023/A:1010933404324.

BuildZero. Open CityGML data for the United States, 2019. https://github.com/opencitymodel/opencitymodel (accessed: 03.12.2019).

City Planning Toronto. 3D Massing, 2019. https://open.toronto.ca/dataset/3d-massing/ (accessed: 15.12.2019).

Coalition of Geospatial Organizations. *Second Report Card on the U. S. National Spatial Data Infrastructure*. COGO, December 2018.

G. Gröger, T. H. Kolbe, C. Nagel, and K.-H. Häfele. *OGC City Geography Markup Language (CityGML) Encoding Standard*. Open Geospatial Consortium, April 2012. URL http://www.opengis.net/spec/citygml/2.0.

U. Grömping. Variable Importance Assessment in Regression: Linear Regression versus Random Forest. *The American Statistician*, 63(4):308–319, 2009. doi:10.1198/tast.2009.08199.

S. R. Gunn. Support Vector Machines for Classification and Regression. Technical report, University of Southamptop, May 1998.

R. Hecht, G. Meinel, and M. Buchroithner. Automatic identification of building types based on topographic databases – a comparison of different data sources. *International Journal of Cartography*, 1(1):18–31, 2015. doi:10.1080/23729333.2015.1055644.

A. Henn, C. Römer, G. Gröger, and L. Plümer. Automatic classification of building types in 3D city models. *GeoInformatica*, 16(2):281–306, April 2012. doi:10.1007/s10707-011-0131-x.

M. Kennedy and S. Kopp. *Understanding Map Projections: GIS by ESRI*. ESRI, 2000.

H. Ledoux. On the Validation of Solids Represented with the International Standards for Geographic Information. *Computer-Aided Civil and Infrastructure Engineering*, 28(9):693–706, 2013. doi:10.1111/mice.12043.

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*, 4(1-12), 2019. doi:10.1186/s40965-019-0064-0.

A. Liaw and M. Wiener. Classification and Regression by randomForest. *R News*, 2(3):18–22, 2002.

Microsoft. Computer generated building footprints for the United States, June 2018. `https://github.com/Microsoft/USBuildingFootprints` (accessed: 03.12.2019).

OpenTopography. OpenTopography: High-Resolution Topography Data and Tools, 2019. `https://opentopography.org/` (accessed: 03.12.2019).

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

QGIS Development Team. QGIS - A Free and Open Source Geographic Information System, 2019. `https://qgis.org/en/site/` (accessed: 03.12.2019).

L. Ross. *Virtual 3D City Models in Urban Land Management - Technologies and Applications*. PhD thesis, Technische Universität Berlin, Berlin, Germany, 2010.

Safe Software Inc. FME - Data Integration Platform, 2019. `https://www.safe.com/fme/` (accessed: 03.12.2019).

B. Smith and D. Sandwell. Accuracy and resolution of shuttle radar topography mission data. *Geophysical Research Letters*, 30(9), 2003. doi:10.1029/2002GL016643.

A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14:199–222, 2004.

C. Strobl. *PostGIS*, pages 891–898. Springer US, Boston, MA, 2008. ISBN 978-0-387-35973-1. doi:10.1007/978-0-387-35973-1_1012.

TheKernelTrip. Computational complexity of machine learning algorithms, 2018. https://www.thekerneltrip.com/machine/learning/computational-complexity-learning-algorithms/ (accessed: 03.01.2020).

S. R. Tipireddy. Multivariate linear regression, unknown. https://www.hackerearth.com/practice/machine-learning/linear-regression/multivariate-linear-regression-1/tutorial/ (accessed: 03.01.2020).

U.S. Geological Survey. What is the State Plane Coordinate System? Can GPS provide coordinates in these values?, 2017. https://www.usgs.gov/faqs/what-state-plane-coordinate-system-can-gps-provide-coordinates-these-values?qt-news_science_products=0#qt-news_science_products (accessed: 15.12.2019).

U.S. Geological Survey. The National Map (TNM) Download v1.0, 2019. https://viewer.nationalmap.gov/basic/ (accessed: 03.12.2019).

N. Wang, A. Vlachokostas, M. Borkum, H. Bergmann, and S. Zaleski. Unique Building Identifier: A natural key for building data matching and its energy applications. *Energy and Buildings*, 184:230–241, 2019. ISSN 0378-7788. doi:10.1016/j.enbuild.2018.11.052.

F. Warmerdam, E. Rouault, et al. ogr2ogr - GDAL Documentation, 2019. https://gdal.org/programs/ogr2ogr.html (accessed: 03.12.2019).