

SHAPE BASED CLASSIFICATION OF SEISMIC BUILDING
STRUCTURAL TYPES

A thesis submitted to the Delft University of Technology
in partial fulfillment
of the requirements for the degree of

Master of Science in
Geomatics for the Built Environment

by

Raphael Sulzer

February 2018

Raphael Sulzer: *Shape based classification of seismic building structural types*
(2018)

An electronic version of this thesis is available at:
<https://repository.tudelft.nl/>.

The work in this thesis was supported by:

ARUP Arup Amsterdam
Beta Building, Naritaweg 118
1043 CA Amsterdam
The Netherlands

Supervisors: Dr. ir. Pirouz Nourian
Dr. Jan van Gemert
Co-reader: Dr. Abdoulaye Diakité
Ext. Supervisor: Dr. Michele Palmieri

ABSTRACT

The seismic building structural type (SBST) is an important input for seismic vulnerability and risk assessment, as it describes the main load-bearing structure of a building and, therefore, its behaviour under seismic load. However, for numerous areas in earthquake prone regions SBST information is outdated, unavailable, or simply not existent. Traditional methods to gather this information, such as building-by-building inspections, are costly and highly time-consuming, making them unfeasible for assessing large building inventory. Previous research in the area has shown that datasets available on a large scale, such as remote sensing or cadastral data, can be combined with a limited number of in situ building inspections, in a supervised machine learning approach. This can allow to predict SBSTs of buildings without first inspecting them. This thesis investigates into the usability of such an approach for the building stock of Groningen in the Netherlands.

Within the context of the Groningen Earthquake Structural Upgrading project, Arup Amsterdam processes building inspections to determine SBSTs defined by the Global Earthquake Model (GEM) taxonomy. This results in a dataset where attributes, such as the lateral load resisting system, the floor and exterior wall type are known for a limited number of buildings in the Groningen region. The GEM attributes provide important information to seismically assess the buildings. Since the GEM attributes cannot directly be measured or observed on a large scale, we extract the construction year of the Groningen buildings and geometric features, such as the gutter height and footprint area, from a cadastral dataset and an aerial laser scanning (ALS) point cloud. This allows to represent buildings that have not been inspected, with a feature vector containing relevant information to predict their SBST. Based on this feature representation, a random forest classifier is used to classify a sample of the Groningen buildings. Compared to a buildings ground truth, predictive accuracies between 53% and 93% are achieved, depending on the specific type. Although, we only conduct this process with the eight most common SBSTs in a sample dataset, it can in theory be used for the full Groningen building stock. A prerequisite of the method is that enough training samples for each type are available, as an input for the supervised machine learning algorithm.

Besides the geometric features mentioned above, we also investigate into different ways of describing the geometric shape of a building. We base this step on the assumption that similar SBSTs and similar seismic behaviour of buildings, can be inferred from geometric similarities. To describe the geometric shape of a building we apply Shape DNA, a spectral shape descriptor based on the beginning sequence of the eigenvalues of the Laplace-Beltrami operator. We make use of the finite element discretisation of the Laplace-Beltrami operator that can be applied on a 2-manifold triangle mesh. In an experiment with synthetically generated meshed building models, we succeed in predicting the roof type of buildings with accuracies of 85% and better, only relying on a buildings Shape DNA. The roof of a building thereby serves as a local shape feature, that cannot easily be described and identi-

fied by using traditional shape analysis of buildings. Such shape features are expected to provide relevant information when predicting the SBST or the behaviour of a building under seismic load.

We also propose a process to gain a 2-manifold triangle building mesh from the ALS point cloud and building footprints. Using this process, it is possible to apply Shape DNA in the prediction of SBSTs for the Groningen buildings. However, when applied on our sample dataset, Shape DNA could not improve the predictive accuracies. The reasons for this are manifold: 1) The meshes from the real world dataset are not as regular and dense as the synthetically generated building meshes. This can lead to inaccurate eigenvalues and, thus, to an inaccurate description of the building geometry. A reconstruction of the building geometry in the form of a structured boundary representation is necessary to improve the mesh quality. 2) Our sample dataset may be too small to allow the machine learning algorithm to benefit from the extensive geometric information provided by Shape DNA. More information generally also carries more noise, that can affect the machine learning algorithm adversely. Many algorithms have ways to sort this problem out internally, however, they may require more training data to do so. 3) Our sample dataset mostly includes terraced houses that have a relatively similar geometry. Different kinds of observations, such as visual features, may be necessary to distinguish different SBSTs with similar geometries. 4) The SBST only allows an estimate of the seismic behaviour of buildings. The actual seismic behaviour may have a stronger dependency on the geometry of a building. Thus, supervising a machine learning algorithm with e.g. a seismic vulnerability index instead of SBSTs, described by the GEM taxonomy, may be beneficial for conducting a large scale seismic assessment.

In general, this thesis shows a lot of potential in the combination of knowledge from the domains of Geomatics, machine learning and seismic engineering. We prove that large scale seismic assessment using remote sensing and cadastral data in a supervised machine learning approach is possible. We, thus, also validate previous research in this area for a sample of the Groningen building stock in the Netherlands. Further research for the remaining building stock, large scale seismic assessment in general, and the usability of Shape DNA for shape analysis of buildings is necessary.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to the people who directly or indirectly contributed to this thesis.

First and foremost, I thank my daily supervisor Pirouz Nourian. Major parts of this thesis are based on his ideas. His inspiring thoughts and guidance helped me to shape this research into what it has become. I am very grateful for the tremendous amount of time he invested not only into this thesis, but also into my personal development as a researcher.

Furthermore, I would like to thank my second supervisor Jan van Gemert. Besides his valuable input for the machine learning parts of this research, Jan helped me to sharpen my thoughts many times. His input gave a lot of structure to this thesis.

I would like to thank my external supervisor Michele Palmieri for making it possible to conduct this research at Arup Amsterdam. Michele gave me a lot of freedom to work on my own research, which I am very grateful for. I also thank the rest of the team at Arup Amsterdam for welcoming me with open arms. Furthermore, I gratefully acknowledge Arup for sponsoring the development of this thesis, and Nederlandse Aardolie Maatschappij BV for sponsoring the collection of data described in this work.

This thesis also marks the end of my studies at TU Delft. I would like to thank all the people that are involved in the MSc Geomatics program. The teachers provide a great education and, thus, many of them have also indirectly contributed to this work.

Finally, I would like to thank my family and friends. Without their support I would not even be close to finishing a master's degree in Delft.

CONTENTS

Abstract	iii
Acknowledgements	v
Contents	vii
List of figures	ix
List of tables	xi
Glossary	xiii
1 INTRODUCTION	1
1.1 Motivation and problem statement	1
1.2 Objective and research question	1
1.3 Use case	2
1.4 Scope	3
1.5 Outline and contributions	3
2 BACKGROUND AND RELATED WORK	7
2.1 Seismic vulnerability assessment	7
2.1.1 Introduction	7
2.1.2 Global earthquake model taxonomy	7
2.1.3 Summary and conclusion	10
2.2 Machine learning and pattern recognition	11
2.2.1 Introduction	11
2.2.2 Feature based object representation	12
2.2.3 Generalisation and classification	14
2.2.4 Evaluation	21
2.2.5 Tuning hyperparameters of a classifier	23
2.2.6 Dimension reduction	23
2.2.7 Summary and conclusion	24
2.3 Sensing and modeling the built environment	25
2.3.1 Introduction	25
2.3.2 Sensing the built environment	25
2.3.3 Modeling the built environment	26
2.3.4 Summary and conclusion	27
2.4 Shape analysis	28
2.4.1 Introduction	28
2.4.2 Shape descriptors and deep learning	28
2.4.3 Summary and conclusion	31
2.5 Automatic SBST prediction	32
2.5.1 Introduction	32
2.5.2 Related work	32
2.5.3 Summary and conclusion	34
3 METHODOLOGY	37
3.1 Preprocess input data	37

3.1.1	3D building modeling	37
3.1.2	Classification of building dataset	45
3.1.3	Summary	46
3.2	Synthetic 3D building models	46
3.2.1	Building typologies	46
3.2.2	Procedural modeling	47
3.2.3	Meshing	49
3.3	Feature based building representation	50
3.3.1	Shape analysis and feature extraction	50
3.3.2	Feature scaling	56
3.4	Automatic classification	56
3.4.1	Supervised learning	56
3.4.2	Evaluation	56
4	IMPLEMENTATION AND EXPERIMENTS	59
4.1	Input data	59
4.1.1	Remote sensing data	59
4.1.2	Cadastral data	60
4.1.3	Building dataset	61
4.1.4	Synthetic building models	62
4.2	Software	63
4.2.1	FME	63
4.2.2	Postgres DB	63
4.2.3	Rhinoceros and Grasshopper	63
4.2.4	C#	64
4.2.5	Python and libraries	64
4.2.6	shapeDNA-tria	65
4.3	Implementation	65
4.3.1	Shape analysis	65
4.3.2	Supervised learning	66
4.4	Experiments	67
4.4.1	Shape analysis	68
4.4.2	SBST prediction	79
5	CONCLUSION, DISCUSSION AND RECOMMENDATIONS	87
5.1	Summary and conclusion	87
5.1.1	Limitations	87
5.1.2	Answer to research question	88
5.2	Discussion and recommendations	89
5.2.1	Shape analysis of buildings	89
5.2.2	Automatic SBST prediction	90
5.3	Reflection	92
	Bibliography	93

LIST OF FIGURES

Figure 1.1	Scope of the thesis	3
Figure 1.2	Methodology developed within the thesis	5
Figure 2.1	Examples of lateral load-resisting systems	8
Figure 2.2	Exemplary building models of Groningen building stock	10
Figure 2.3	The process of supervised learning	12
Figure 2.4	Buildings in feature space	13
Figure 2.5	Buildings in feature space with linear SVM applied	15
Figure 2.6	Kernel function applied to SVM	16
Figure 2.7	Buildings in feature space with kernel SVM applied	17
Figure 2.8	Buildings in feature space with k-NN applied	18
Figure 2.9	Buildings in feature space with Decision Tree applied	19
Figure 2.10	Buildings in feature space with RF applied	20
Figure 2.11	Buildings in feature space with ANN applied	21
Figure 2.12	K-fold cross-validation	22
Figure 2.13	Confusion matrix	22
Figure 2.14	Point cloud obtained by ALS	26
Figure 2.15	LODs in CityGML	26
Figure 2.16	Example of a polygon mesh	30
Figure 3.1	Summary of the method of our approach	37
Figure 3.2	Flowchart of 3D building modeling	38
Figure 3.3	Images of process of 3D building modeling	39
Figure 3.4	Region growing algorithm	40
Figure 3.5	Images of region growing algorithm	42
Figure 3.6	Images of region growing algorithm	44
Figure 3.7	Mesh comparison	44
Figure 3.8	Classification of building dataset	45
Figure 3.9	Roof types of synthetic building models	47
Figure 3.10	Synthetic building models	49
Figure 3.11	Synthetic building models with different meshing	50
Figure 3.12	Different approaches to extract building height from point cloud	53
Figure 3.13	Extracting the gutter height of buildings	54
Figure 3.14	Roof feature extraction	55
Figure 4.1	Actueel Hoogtebestand Nederland	60
Figure 4.2	Examples of the building stock with their Global Earthquake Model (GEM) label	62
Figure 4.3	Synthetic building dataset	63
Figure 4.4	FME workbench	65
Figure 4.5	Implementation of region growing algorithm in Grasshopper	66
Figure 4.6	Shape DNA of a sphere	69
Figure 4.7	Synthetic building models with different meshing	71
Figure 4.8	MDS of Shape DNA of synthetic building models with different meshing applied	73
Figure 4.9	MDS of Shape DNA feature space with synthetic building models of different roof types.	74

Figure 4.10	Confusion matrix of roof type classification	74
Figure 4.11	Confusion matrix of roof type classification with extension of $s = 1$	75
Figure 4.12	Confusion matrix of roof type classification with extension of $s = 1$	76
Figure 4.13	Synthetic building models with and without extension	76
Figure 4.14	Accuracy of roof type classification over training sample size	77
Figure 4.15	Confusion matrix of roof type classification with extensions	78
Figure 4.16	MDS plot of buildings in unnormalised "Shape DNA-space"	80
Figure 4.17	MDS plot of buildings in normalised "Shape DNA-space"	81
Figure 4.18	Confusion matrix of LLRS prediction with Shape DNA and year of construction	82
Figure 4.19	Feature importance in LLRS prediction according to RF	83
Figure 4.20	Confusion matrix of LLRS prediction only using footprint area and year of construction	83
Figure 4.21	Feature importance according to RF	84
Figure 4.22	Confusion matrix GEM classification	85
Figure 4.23	Accuracy of GEM classification over training sample size	86

LIST OF TABLES

Table 2.1	Examples of Groningen SBSTs described with GEM taxonomy	11
Table 3.1	Roof types of synthetic building models	48
Table 3.2	Features representing a building	51
Table 4.1	Groningen training samples	61
Table 4.2	Synthetic training samples	62
Table 4.3	Database table with shape features of a building . . .	66
Table 4.4	Database table with Shape DNA of a building	66
Table 4.5	Database table with GEM attributes of a building . .	67
Table 4.6	Experiments conducted in this thesis	68
Table 4.7	Roof count and angle of synthetic building models .	77
Table 4.8	Overview of SBST classification results	84

GLOSSARY

2D	two-dimensional	12
3D	three-dimensional	13
AHN	Actueel Hoogtebestand Nederland	59
ALS	airborne laser scanning	4
ANN	Artificial Neural Network	20
BAG	Basisregistraties Adressen en Gebouwen	60
BIM	building information model	27
CNN	convolutional neural network	29
CR	concrete	8
CV	cross-validation	21
DT	Delaunay triangulation	44
EMS	European Macroseismic Scale	33
EW	outer leaf wall present	9
EWN	no outer leaf wall present	9
FC	concrete floor	9
FDN	fixed distance neighbourhood	40
FEM	finite element model	43
FM	masonry floor	9
FO	other floor	9
FW	wooden floor	9
GEM	Global Earthquake Model	7
k-NN	K-Nearest Neighbour	17
LBO	Laplace-Beltrami operator	29
LH	hybrid lateral load resisting system	8
LLRS	lateral load-resisting system	8
LN	no lateral load resisting system	8
LOD	level of detail	26
LULC	landuse/landcover	33
LWAL	wall based lateral load resisting system	8
MDS	multidimensional scaling	13
MUR	unreinforced masonry	8
NMCA	national mapping and cadastral agency	25
PCA	principle component analysis	24
PDOK	Publieke Dienstverlening Op de Kaart	59
rangle	roof angle	54
RBF	radial basis function	74
rcount	roof segment count	54
RF	Random Forest	19
SBST	seismic building structural type	1
SVM	Support Vector Machine	15
t-SNE	t-distributed stochastic neighbour embedding	13
WHE	World Housing Encyclopedia	33

1 | INTRODUCTION

Seismic risk has been defined as the potential social, economic and environmental consequences of a seismic event. Of the three factors that determine seismic risk, namely the values of elements at risk, their vulnerability and the probability of occurrence of a seismic event, only the first two are amenable to human control [19]. Mitigating seismic risk to a level that is acceptable for affected communities is one of the key steps of seismic risk management. An effective way to reduce seismic risk is the design and construction of earthquake resistant structures. However, the majority of the built environment is erected largely oblivious to this fact, even though the existence of seismic activity is well known to humanity [3]. Thus, knowledge about the seismic vulnerability of existing building stock is of vital importance in seismic risk management, e.g. for the design and development of seismic retrofit strategies.

1.1 MOTIVATION AND PROBLEM STATEMENT

The seismic building structural type (SBST) reflects the main load-bearing structure of a building and therefore its behaviour under seismic load [22]. However, for numerous areas in earthquake prone regions this information is often outdated, unavailable, or simply not existent [21]. Traditional methods to gather this information, such as building-by-building inspections, are costly and highly time-consuming, making them unfeasible for a whole building stock. For this reason the use of remote sensing data and ancillary information (such as cadastral data) has been proposed to allow a fast acquisition of SBST information on urban and regional scale. Subsequently, machine learning algorithms may be used to analyse the gathered information, e.g. to classify a building stock into groups with similar SBSTs [22, 36, 50, 62]. However, existing approaches for such a workflow often deliver highly aggregated results in terms of their spatial or typological granularity, and therefore prevent a precise seismic assessment.

1.2 OBJECTIVE AND RESEARCH QUESTION

This research investigates into the use of remote sensing data and ancillary information to assist large scale seismic vulnerability assessment. In a case study, we aim to develop and implement a methodology to automatically predict a detailed SBST for buildings of the Groningen building stock in the north of the Netherlands. To realise the prediction we aim to apply machine learning techniques. Subsequently, the predicted SBST should allow an as-

assessment of the seismic vulnerability of each building, which is, however, out of the scope of this thesis.

As a second objective, we investigate into the use of different shape descriptors, that can be applied on building geometries extracted from remote sensing and cadastral data. We thereby follow the assumption that similar structural systems and materials can be inferred from geometric similarities [14]. Simply spoken, shape descriptors map (parts of) the shape of an object onto a numerical vector. Subsequently, such a vector may be used in a machine learning algorithm to quantify geometric similarities. The focus of our investigation is Shape DNA, a shape descriptor based on the beginning sequence of the eigenvalues of the Laplace-Beltrami operator (LBO) [48].

The main research question and sub-question for this thesis are stated as follows:

To which extent is it possible to describe geometric similarities of buildings using Shape DNA?

Which shape descriptors are relevant for the prediction of seismic building structural types and how can they be applied on a building representation?

1.3 USE CASE

Arup Amsterdam is commissioned with a large scale seismic risk assessment for the region of Groningen in the Netherlands. In this region, seismic risk is induced by the extraction of gas from the large Groningen gas field [40]. This leads to a unique situation where the traditional, mainly non-resilient building stock of the Groningen region is exposed to recurring seismic events with minor intensities. To prevent damage to buildings, the development of a seismic retrofit strategy is necessary. This requires detailed information about the Groningen building inventory. However, in situ inspections of every building are not feasible, considering the size of the building stock with more than 250.000 buildings. The large amount of openly available remote sensing and cadastral data provided by the Dutch government can complement existing building inspections and help to analyse the building stock in a significantly reduced amount of time.

1.4 SCOPE

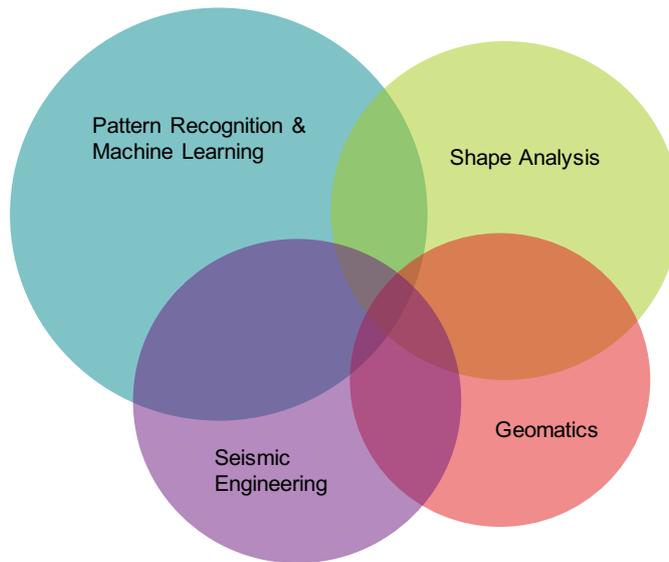


Figure 1.1: Scope of the thesis as the intersection of four domains

In this thesis we focus on the intersection of the four domains depicted in figure 1.1. We will discuss topics that are relevant to achieve the objectives and answer the research question (section 1.2). Furthermore the following remarks are made:

- Our focus is the development and implementation of **SBST** prediction for the presented use case. Beyond this use case we present possible options and give recommendations, but not develop a general methodology for large scale seismic vulnerability assessment.
- We only have a limited amount of data available for this thesis, comprising less than 10% of the building stock of the Groningen region. A quantitative evaluation of the developed methodology can only be given for the available dataset. This entails that we will not actually predict the **SBSTs** for the full building stock. However, we aim to develop and implement a methodology that is able to achieve this goal. Thus, we aim to exclude any specificities from the dataset as good as possible to get an unbiased performance estimate of our implementation.
- For the investigation of different shape descriptors, such as Shape DNA, we do not limit ourselves to the presented use case. For a quantitative evaluation of different approaches we generate synthetic building models with distinct geometries for this part of the research.

1.5 OUTLINE AND CONTRIBUTIONS

In chapter 2 we provide an introduction to the fields of seismic vulnerability assessment, sensing and modeling of the built environment, machine learn-

ing and pattern recognition, and shape analysis. Concepts that are necessary to understand the remainder of the thesis are discussed and the particular datasets and techniques that we use in our approach are motivated. We also mention alternative ways independent of the case study of this thesis, some of them being further discussed in a future work section in chapter 5. These sections can be seen as a conceptual framework for large scale seismic building vulnerability assessment. Furthermore, we briefly present the previously conducted research in this area. A variety of largely heterogeneous approaches have been developed, often tailored to the presented case studies.

In chapter 3 we report on the development and implementation of a methodology for predicting SBSTs. The main idea for this process is depicted in figure 1.2. We reconstruct a three-dimensional (3D) building model using an airborne laser scanning (ALS) point cloud and building footprint polygons. We apply different shape descriptors on this model to extract geometric features of the building. Then we represent every building by a feature vector consisting of the geometric features and the building's year of construction. This representation is used to classify the buildings into groups of similar SBSTs using a supervised learning algorithm. The groups are defined using the Global Earthquake Model taxonomy. Furthermore, we construct synthetic building models with different roof types and extensions to gain a deeper understanding of the suitability of different shape descriptors for shape based building classification.

Chapter 4 reports on experiments conducted with the synthetic building models, and with an available dataset of parts of the Groningen building stock.

In chapter 5 we provide a discussion about the outcomes of the thesis and conclude by answering the research question. Additionally, we give recommendations for possible future work following the ideas of this thesis.

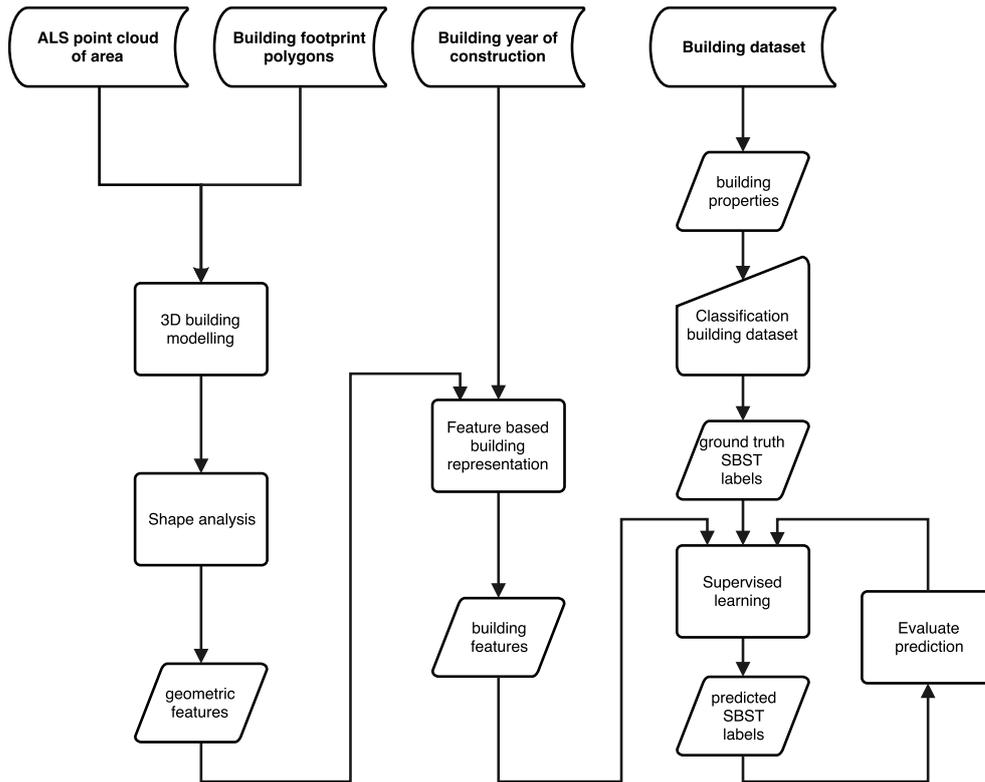


Figure 1.2: Methodology of our approach. A more detailed version of the process is depicted in figure 3.1.

2 | BACKGROUND AND RELATED WORK

2.1 SEISMIC VULNERABILITY ASSESSMENT

2.1.1 Introduction

Considering the case of a single building, its loss E may be defined as the cost of restoring the building to its original state before the earthquake, or demolishing and replacing it by an identical building at the same site. The associated seismic risk R to the building can be calculated as follows [19]:

$$R = E \cdot \int_0^{i_{max}} V_i \cdot p_i \cdot di. \quad (2.1)$$

The vulnerability V_i , thus defined, is expressed by a mathematical function that relates the loss of the building to seismic ground motion of intensity i and probability p_i of such ground motion occurring at the site during a given period of time. Different approaches exist for developing vulnerability functions, such as empirical, analytic or expert based approaches (or a combination thereof) [12]. Before choosing the most appropriate approach it is essential to first establish the project objectives [18]. Even with detailed building inspections, modeling seismic vulnerability of a building is not a trivial and time consuming process. For this reason, a building inventory is often grouped into typologies with similar seismic behaviour [30]. However, defining such a typology, i.e. classifying buildings according to their expected seismic behaviour, is not an obvious process.

2.1.2 Global earthquake model taxonomy

The Global Earthquake Model foundation¹ provides an internationally standardised scheme for seismic risk assessment. Within this scheme, the Global Earthquake Model building taxonomy was developed, to allow the uniform classification of buildings with regard to their expected seismic behaviour. To this end, the Global Earthquake Model (GEM) taxonomy describes a building with 13 attributes (see section 2.1.2.1 to 2.1.2.13) that uniquely determine its SBST.

¹ <https://www.globalquakemodel.org/>

2.1.2.1 Direction

This attribute enables the distinction between different LLRSs in the two main directions of the building. A common example are terraced houses that may have a wall based lateral load resisting system (LWAL) in the direction parallel to the street, but have no lateral load resisting system (LN) in the direction orthogonal to that. In the GEM taxonomy the parameter is used by indexing the LLRS with X or Y. In this thesis we will first write the material and LLRS in X direction following by material and LLRS in Y direction.

2.1.2.2 Material of the lateral load-resisting system

This attribute determines the material of the lateral load-resisting system. Examples may be concrete (CR), wood or unreinforced masonry (MUR)².

2.1.2.3 Lateral load-resisting system

The lateral load-resisting system (LLRS) describes the structural system that provides resistance against horizontal earthquake forces through vertical and horizontal components. Examples are wall based lateral load resisting systems (LWALs) or hybrid lateral load resisting systems (LHs). In the latter case, different types of LLRS are combined in one structure. Figure 2.1 shows some further examples.

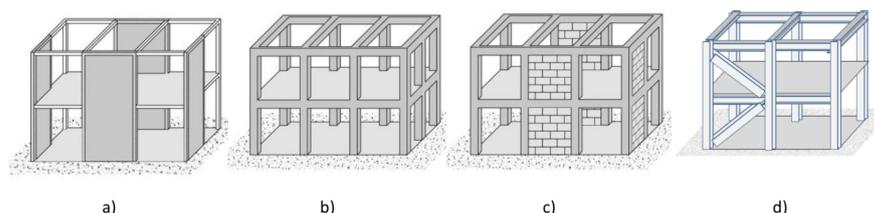


Figure 2.1: Examples of lateral load-resisting systems: a) Wall (LWAL); b) Moment Frame; c) Infilled Frame and d) Braced Frame [11].

2.1.2.4 Height

The building height above ground in terms of the number of storeys (e.g. a building is 3-storeys high); ideally also including information on number of basements (if present). This is an important feature, regarding the seismic behaviour of a building. Even though it can be represented with a single number, it is typically not available on a large scale.

2.1.2.5 Date of construction or retrofit

The date of construction or retrofit identifies the year when the building construction was complete. This parameter is often available on a large scale and can also facilitate the prediction of other GEM attributes.

² In all the figures of this thesis the abbreviation MU is used instead of MUR.

2.1.2.6 *Occupancy*

This attribute is not relevant for the vulnerability of a structure. It is merely used to estimate the exposed people in the building. If no direct information about the occupancy of a structure is available, it may be estimated by looking at the building use, such as residential, commercial, industrial and educational. As an example, educational buildings (e.g. a school) may have a high occupancy during the day but non to little during the night.

2.1.2.7 *Building position within a block*

The building position within a block is especially relevant for terraced houses, where units at the end of a block may behave differently than buildings within the block. Detached buildings are not attached to any other blocks.

2.1.2.8 *Shape of the building plan*

The shape of the building footprint, such as T, L or U shaped may have an influence on the buildings seismic performance.

2.1.2.9 *Structural irregularity*

This attribute can be used to describe a building's structural arrangement, such as one story significantly higher than other stories, an irregular building shape, or change of structural system or material that produces a known vulnerability during an earthquake.

2.1.2.10 *Exterior walls*

The material of exterior walls describe the building enclosure. In this thesis we use this parameter to denote the presence of an outer leaf wall as follows: outer leaf wall present (**EW**) or no outer leaf wall present (**EWN**).

2.1.2.11 *Roof*

This attribute describes the roof shape, material of the roof covering, structural system supporting the roof, and roof-wall connection. The roof shape may be described as gabled, shed, hipped or flat.

2.1.2.12 *Floor*

This attribute describes floor material, floor system type, and floor-wall connection. In this thesis we only describe the material of the building floor, such as wooden floor (**FW**), concrete floor (**FC**), masonry floor (**FM**) or other floor (**FO**) material.

2.1.2.13 Foundation system

The foundation system transmits loads from the building to the underlying soil. Examples are shallow or deep foundation systems, where the choice is mostly depending on the soil conditions.

2.1.3 Summary and conclusion

In this thesis we focus on the first step of seismic vulnerability assessment of a building stock: the classification of buildings according to their *SBST*. We describe the *SBST* by using six out of the 13 *GEM* attributes. This is due to characteristics of the Groningen building stock, where these attributes reflect the most influential parameters on the vulnerability of the buildings. By first expressing the material and *LLRS* in the main direction, material and *LLRS* in the second direction, floor and exterior walls, a *SBST* can be represented such as the following: *CR_LWAL_CR_LN*. In words, concrete with wall based lateral load resisting system in the main direction and concrete with no lateral load resisting system in the second direction, which is a common *SBST* for a terraced building unit. Figure 2.2 shows some more examples of common building types in the Groningen building stock and table 2.1 describes the *SBST* of the buildings with the six *GEM* attributes that are used in this project.

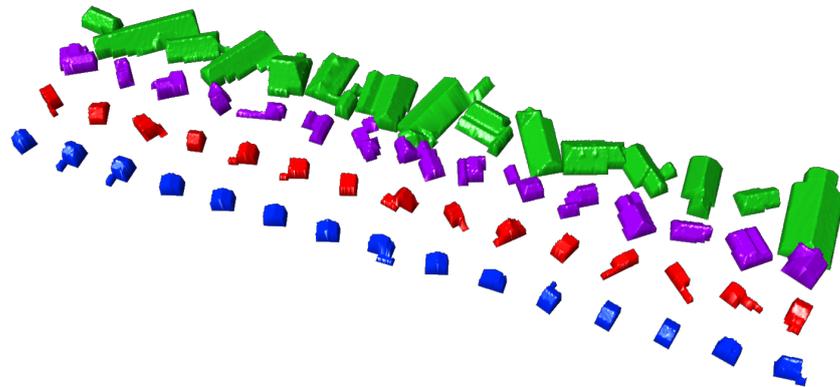


Figure 2.2: Exemplary building models of Groningen building stock. See table 2.1 for a legend. The classification process of these buildings conducted by Arup is further described in section 3.1.2.

Table 2.1: Examples of Groningen SBSTs described with GEM taxonomy

Material _x	LLRS _x	Material _y	LLRS _y	Floor	EW	Description
MUR	LH	MUR	LH	FW	EWN	<i>agricultural</i>
MUR	LWAL	MUR	LWAL	FW	EWN	<i>single unit</i>
MUR	LWAL	MUR	LN	FC	EW	<i>terraced</i>
CR	LWAL	CR	LN	FW	EW	<i>terraced</i>

*The columns and colours used correspond to figure 2.2.

Using the GEM taxonomy an expert in the field of seismic engineering may now be able to classify inspected buildings according to their SBST. In this process he can relate observations of the building to patterns learned over years. As an example, a seismic engineer knows exactly how to distinguish a LWAL from a LH system. However, this manual labelling would be very time consuming for grouping a whole building stock. The fields of pattern recognition and machine learning aim to make the process of detecting and learning of patterns explicit, such that it can be partially or entirely implemented on computers [28]. This will allow a much faster automatic classification process.

2.2 MACHINE LEARNING AND PATTERN RECOGNITION

2.2.1 Introduction

The word pattern can describe slightly different but related concepts [17]: a possible definition of the word pattern is the entire similarity structure in a collection of objects (e.g. buildings), additionally, a pattern can refer to a subset of similar objects in a larger set. In the following, we will use the word pattern class (or simply class) for the second case.

Given a group of objects, there are at least two ways to build an automatic pattern recognition system [61]: 1) supervised learning (e.g. classification, see figure 2.3), where a teacher (e.g. seismic engineer) provides a category label for each pattern class and for each object in a training set, or 2) unsupervised learning (e.g. clustering), in which similar objects are assigned to a hitherto unknown class. In unsupervised learning there is no explicit teacher, and the system forms natural groupings defined by the clustering algorithm itself [16]. Given a particular set of patterns, different algorithms can lead to different clusters. If we want to make sure that the groupings correspond to the predefined SBSTs (section 2.1.3) we need to make use of a supervised learning approach. This entails that we need a training set of buildings, each labelled with its corresponding SBST. To allow a machine to recognise the patterns in the labeled training set, we need a representation

of the buildings in terms of measurable observations that enable a numeric or logic comparison with other buildings in the same problem [17].

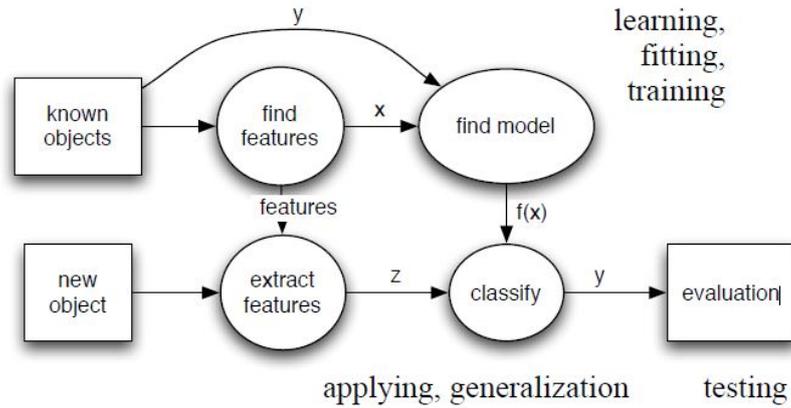


Figure 2.3: The process of supervised learning (Figure taken from [56])

2.2.2 Feature based object representation

In traditional pattern recognition approaches, such as statistical classification, we choose to represent objects by a set of characteristic features. Features are object properties that might be relevant to determine the object's class. Extracting these features from the object is generally problem dependent and thus requires knowledge of the domain [16]. Examples of features that might be relevant for *SBST* classification are the year of construction or footprint area of a building. By extracting these properties from the building model (section 2.3.4), we can build a two-dimensional (2D) feature vector storing the corresponding feature values. The feature vector can be seen as a map of the feature names, e.g. building footprint area and year of construction to a vector of a numerical type. Following this example the first building (top right) of figure 2.2 can be represented with its corresponding feature vector

$$x_1 = \begin{bmatrix} 819 \\ 1940 \end{bmatrix}_{[m, yr.]}$$

Applying such a map to every building also allows to embed the buildings in a vector space spanned by all the feature vectors. We call this vector space the feature space. The exemplary buildings of figure 2.2 are embedded in the feature space spanned by their footprint area and year of construction and visualised with a scatter plot in figure 2.4.

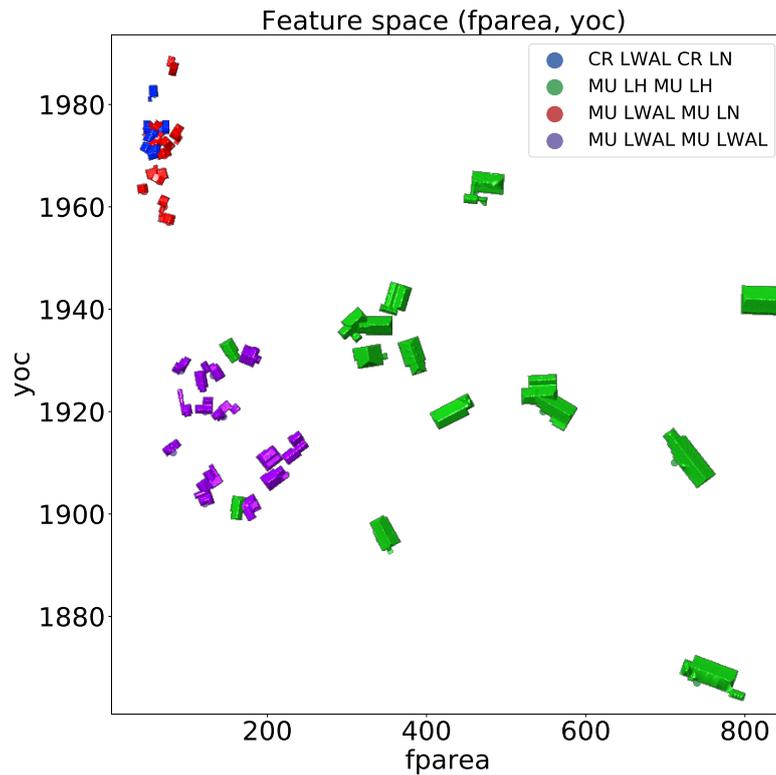


Figure 2.4: Exemplary buildings in feature space spanned by footprint area of the buildings (fparea) and year of construction (yoc)

The feature space offers a direct and intuitive way to see whether a feature representation is a good basis for the subsequent classification problem. If the chosen features really correspond to characteristic properties of the SBSTs, then buildings with the same SBST are close in feature space.

If the dimension of the feature vectors get bigger ($>$ three-dimensional (3D)), it becomes increasingly challenging to plot the data. Techniques like multidimensional scaling (MDS) or the recently developed t-distributed stochastic neighbour embedding (t-SNE) [37] can help to visualise high dimensional data in a two or three dimensional space, e.g. again in the form of a scatter plot.

Representing the buildings by their feature vectors also allows to calculate some type of distance between buildings, e.g. by calculating the euclidean distance between their feature vectors. If the inter-class distances (i.e., between different SBSTs groups) are much larger than the intra-class distances (i.e. inside a group of one SBST class), the classification problem is easy. If they are of similar orders, either the classes overlap, or a more advanced procedure is needed to separate the classes. Obviously, a representation with large inter-class variability and small intra-class variability is desirable [28].

Sometimes it may be a problem if the spread of a feature is much larger compared to other features. If we, for example, look at buildings from the years of 1950 to 1970 with a footprint area of 100 to 600 m^2 the footprint area will have a much bigger influence on most distance measures. If features are considered to be equally important, feature spaces should be created

in which they are equally scaled [17]. This can be done by subtracting the mean of all feature values and dividing by their corresponding standard deviation. This technique is called feature standardisation [43] and is also what we will use for the classification process applied in this thesis.

2.2.3 Generalisation and classification

In a next step, we assume that the patterns found in the observations (i.e. the measured feature values of the buildings) made on the training set are representative for all objects (buildings) in the same problem. This process is called generalisation and can allow to estimate the SBST of unlabelled buildings. Different ideas, developed in multivariate statistics, statistical pattern recognition or machine learning exist to build a decision function, such as a classifier, optimised by learning from the examples in the training set.

One way to build such a decision function is to first estimate the (posterior) probability with which an unknown object belongs to the classes ω_i , given the corresponding feature vector takes the value x . Afterwards, one can simply choose the class membership with the highest probability to classify the unknown object. In general there are two ways to estimate the posterior probability. First, it can be calculated according to the Bayes' theorem [16]:

$$P(\omega_i|x) = \frac{p(x|\omega_i)P(\omega_i)}{p(x)}. \quad (2.2)$$

Following this approach, an optimal decision function can be found, resulting in the least possible classification error. However, to arrive at this goal, the probability of every possible feature value in the given problem (or potentially even every combination for multiple features) has to be known (i.e. the class conditional probability density function $p(x|\omega_i)$). This is in general not possible or at least not feasible. Different techniques have been developed to estimate the class-conditional probability densities based on the available training set. However, often a different approaches, in which the posterior probabilities are estimated directly tend to deliver better results [41]. A variety of different classifiers have been developed following this approach.

Depending on the nature and complexity of the problem and the available input data, different classifiers may be more or less appropriate and lead to different results. In general, there is no single classifier that will work best on any given dataset (cf. No Free Lunch Theorem [16]). A common approach is to simply compute different classifiers, and interpret their results before choosing a final classifier for the classification of the unknown samples. In the following section 2.2.3.1 to 2.2.3.5 we provide an overview of classifiers which are frequently applied to the problem of automatic building and SBST classification [22, 36, 50, 62], and will therefore also be explored in this thesis. For an in depth and mathematical description of the concepts the reader is referred to the work of Duda et al. [16] or Theodoridis and Koutroubas [57]. Besides presenting their main idea, we visualise the decision functions of the different classifiers calculated by using the training sample given in figure 2.2. However, this is merely meant to depict and

compare the concepts of the classifiers rather than drawing conclusions for the automatic classification developed in this thesis. The reason for this is that the bespoke training sample is too small to allow a generalisation of the observations to the complete Groningen building stock.

2.2.3.1 Support Vector Machine

The Support Vector Machine (**SVM**) is a linear classifier that constructs a hyperplane in feature space to separate two classes [15]. The location and orientation of the hyperplane is optimised in an initial training phase. This is done in a way that the margin which the hyperplane leaves to both classes is maximised [57]. In some cases it might be desirable to allow misclassifications, i.e. to allow training samples to be located on the incorrect side of the respective hyperplane. The trade-off between leaving a high margin and keeping the number of misclassifications small is controlled by the cost parameter C . This parameter has to be given by the user before the actual training of the **SVM**.

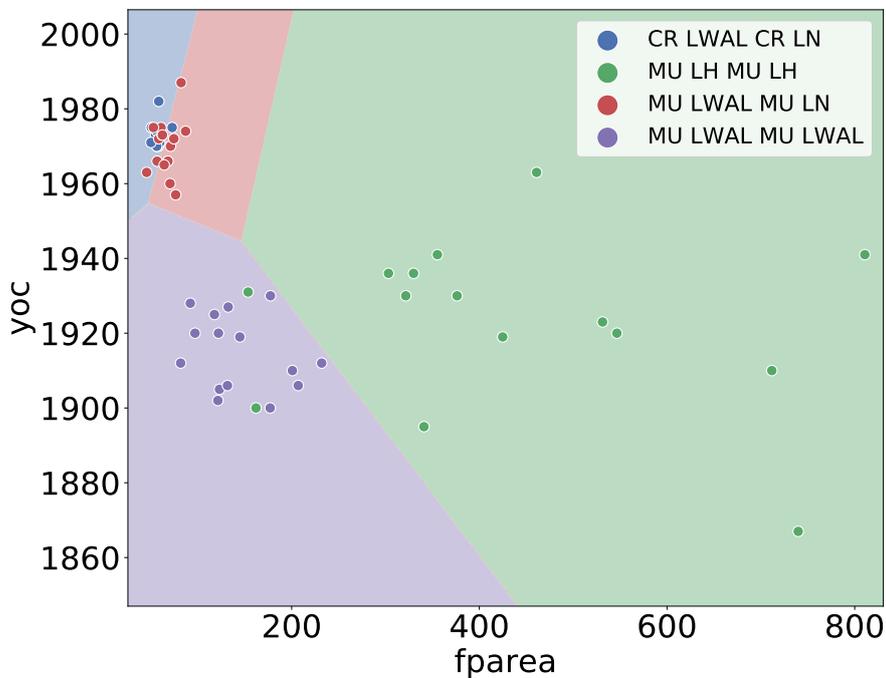


Figure 2.5: Buildings in feature space with linear **SVM** applied

A limitation of the **SVM** is that it will generally not be able to satisfyingly solve arbitrary non-linearly separable classification problems, even with the added flexibility of allowing misclassifications. To this end, the **SVM** can be extended with the so-called kernel trick. In this approach a kernel function $K(\mathbf{x}, \mathbf{y})$ defines a nonlinear mapping to a sufficiently high-dimensional feature space in which the training samples are linearly separable (see figure 2.7). The kernel function thereby replaces the dot product $\langle \mathbf{x}, \mathbf{y} \rangle$ occurring in the mathematical formulation of the **SVM**.

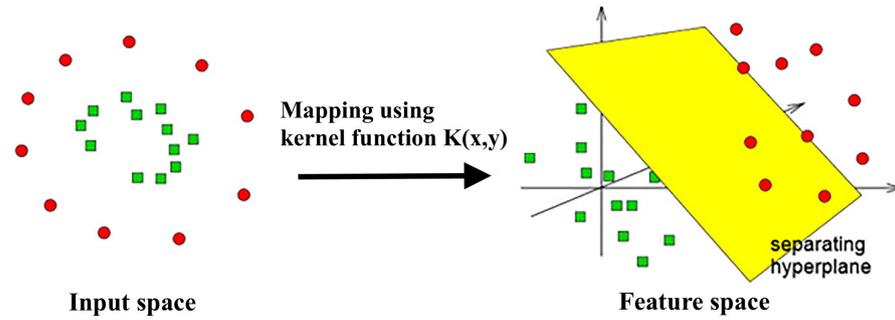


Figure 2.6: Kernel function applied to SVM

Typical examples for kernel functions are [56] radial basis functions (Gaussians) with the form

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{\sigma^2}\right) \quad (2.3)$$

or polynomial functions with the form

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y} + 1)^d. \quad (2.4)$$

The shape of the kernel function, such as the variance of the Gaussian σ or the degree of the polynomial function d , need to be defined by the user.

Once optimal solutions for the type and shape of the kernel function (if necessary) and the cost parameter C have been found, unknown objects can be classified by calculating on which side of the separating hyperplane they lie on. The main advantage of the SVM is that it generalises remarkably well, even in high dimensional feature spaces with relatively small training sets [56].

So far we have seen how SVMs can be used in binary classification problems. It is possible to extend the SVM approach to multiclass problems, i.e. the classification of N classes. This is done by combining multiple binary classifiers, either in a one-vs-one or one-vs-rest approach [57]. In the first case a SVM is calculated for all pairs of classes, resulting in $N \cdot (N - 1)/2$ classifiers [43]. In the second case N SVMs are trained to separate a single class from all others.

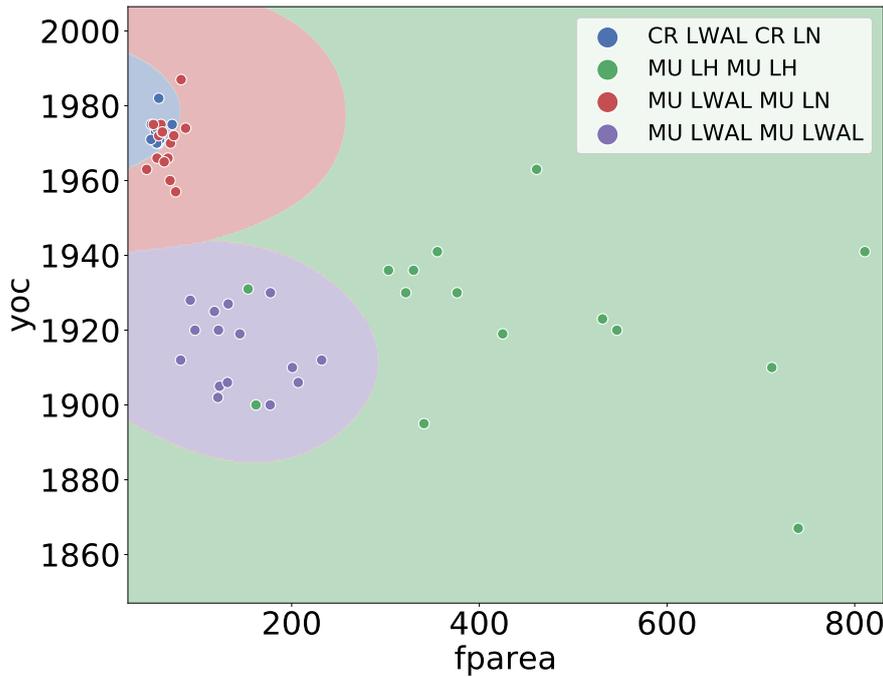


Figure 2.7: Buildings in feature space with kernel SVM applied

Figure 2.5 and 2.7 show the decision boundaries of a linear and a non-linear SVM with a gaussian kernel. Both SVMs are trained using the one-vs-rest approach. This approach will also be used for the classification of SBSTs, as it is generally considered to deliver similar results to the one-vs-one approach while being computationally less expensive [43].

2.2.3.2 K-Nearest Neighbour

The K-Nearest Neighbour (k-NN) classifier requires no training phase to construct a general internal decision function. Instead, unknown samples are classified according to the majority class membership of their k-nearest neighbours in feature space. To find near neighbours, an appropriate distance measure, such as the Euclidean distance, has to be defined. The number of neighbours k allows for a good control over the complexity of the classifier. A value that is not a multiple of the number of classes should be chosen [57]. For a large number of training samples the k-NN classifier often leads to a good classification performance. However, for a large number of features compared to the amount of training samples, the performance may decrease.

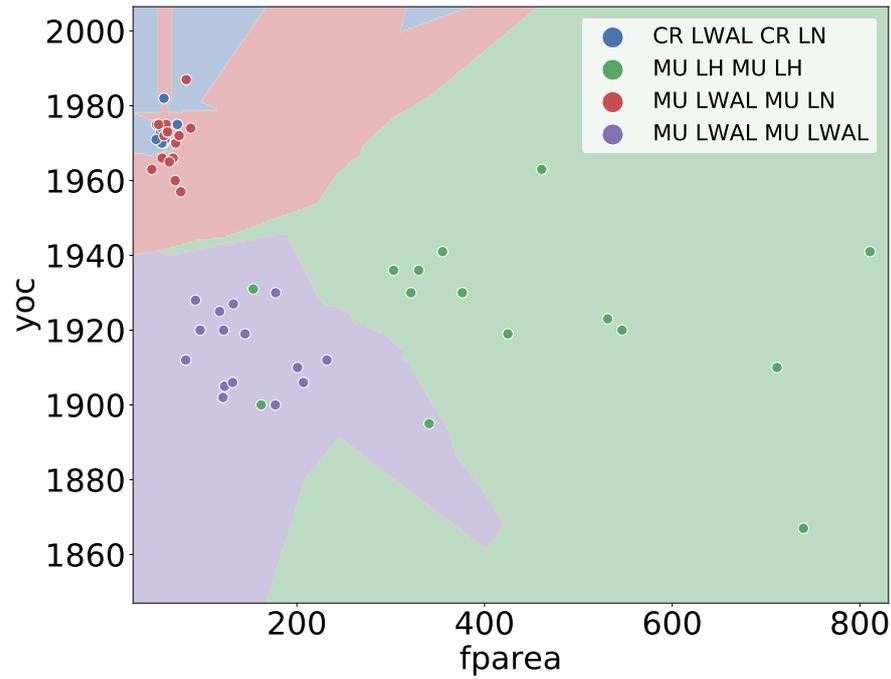


Figure 2.8: Buildings in feature space with k -NN applied

2.2.3.3 Decision Tree

The decision tree is another effective, yet simple classifier [46]. The training step of the decision tree can be seen as asking a sequence of questions in the form of "is feature $x > \alpha$?" [57]. Here, α is a threshold value, that is decided according to a splitting criterion such as entropy or Gini impurity [16]. The questions sequentially split the feature space into several subspaces, each bounded by multiple axis parallel decision boundaries. Following this procedure one could eventually arrive at a state where each subspace only includes a single class. However, this can result in a heavily overtrained classifier. To avoid overtraining there exist multiple stopping criteria, such as limiting the depth of the tree. In the classification phase unknown objects simply traverse the tree by subsequently answering the questions. Eventually objects will arrive at a leaf node classified corresponding to one of the subspaces. The sample is then classified according to the mode of the classes in this subspace.

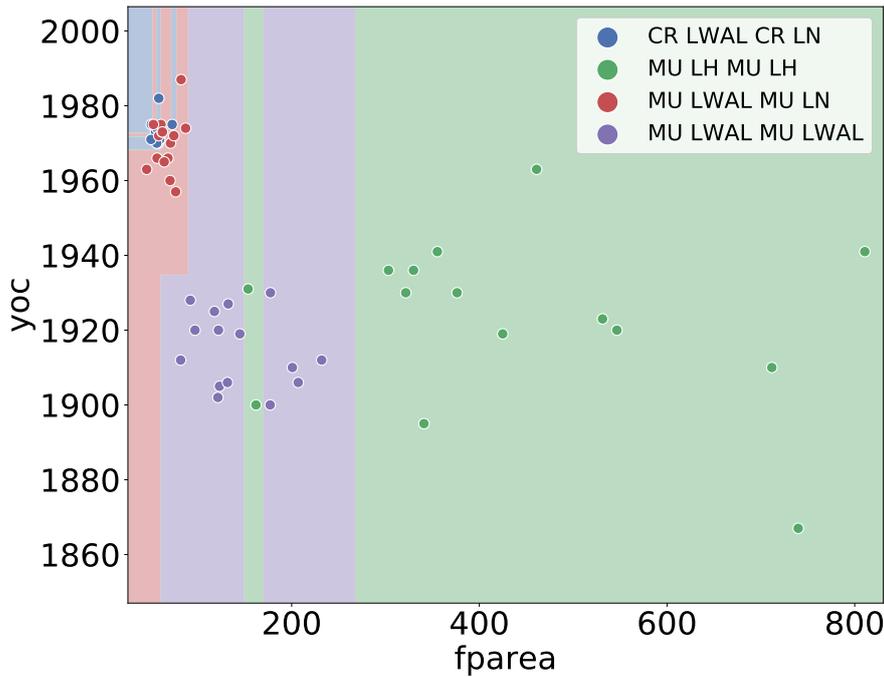


Figure 2.9: Buildings in feature space with Decision Tree applied

The main disadvantage of decision trees is that they may exhibit high variations even on small perturbations of the training set or the order of the features that are used for the splits. This hampers the decision tree method to generalise well for complex datasets.

2.2.3.4 *Random Forest*

The Random Forest (RF) is an ensemble classifier, based on multiple decision trees [10]. This approach follows the common idea of combining several weak classifiers into one strong ensemble. The base classifiers of a random forest are decision trees constructed using random feature sub-sets and random sub-samples of the training set (bagging). The number of trees to construct has to be defined by the operator, and can potentially be large (> 200 trees). In the classification step, every object traverses each decision trees. The final class membership is determined by the mode of the classification results of all the trees. With the help of bagging the variation of the single decision trees can be stabilized. Additionally, some of the decision trees may perform better in particular parts of the feature space even if their performance is not ideal on the whole feature and sample set.

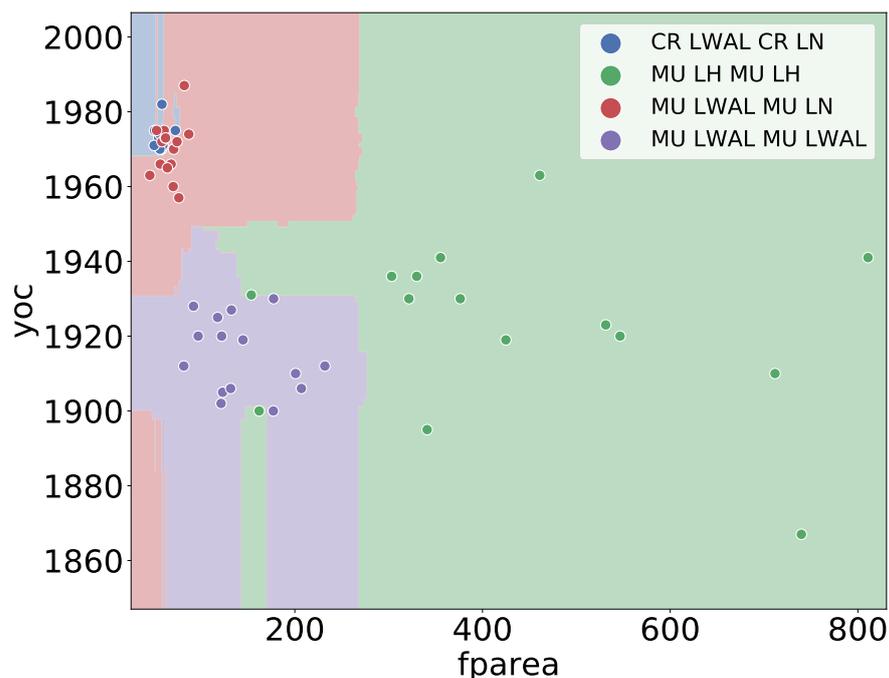


Figure 2.10: Buildings in feature space with RF applied

This is a general feature of ensemble classifiers that in fact allows to construct different classifiers optimized for different parts of the feature space. For [SBST](#) classification this can be a useful property. The presence of many different [SBSTs](#) may be the result of a variety of influences, leading to diverse patterns in the dataset.

2.2.3.5 Artificial Neural Network

Artificial Neural Network ([ANN](#))s are based on the idea of the human nervous system. They consists of several nodes (the neurons) combining multiple inputs into a single output [17]. The neurons are organized in layers forming a network with weighted connections. Usually the network consist of at least one input layer representing the feature values, one or several hidden layers, and an output layer representing the class confidences (cf. posterior probabilities). With this in principle any nonlinear decision boundary can be modeled [66]. [ANNs](#) are trained in a sequential process adjusting the weights of the connections, usually with a gradient descent method, such as backpropagation.

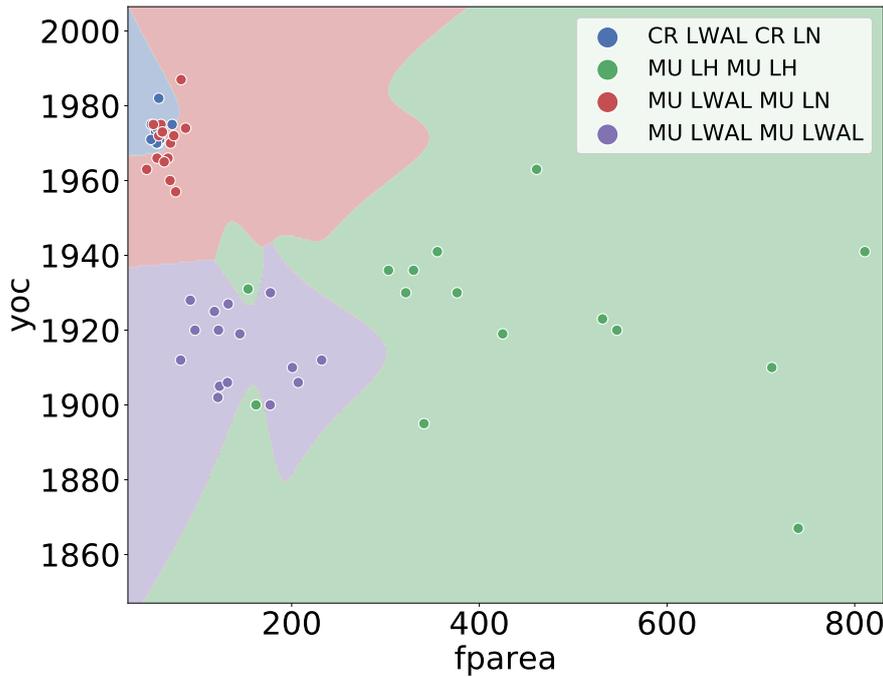


Figure 2.11: Buildings in feature space with ANN applied

ANNs and many variations thereof have proven to be effective, especially for image classification tasks. A disadvantage of ANNs is that they require the tuning of many parameters, such as the number of neurons per layer, the number of hidden layers or the learning rate used in the training phase. Additionally, they may be prone to overfitting their decision boundary onto the specific training set, which can also be seen in figure 2.11.

2.2.4 Evaluation

The evaluation of the classification result is necessary to get an understanding of the quality of a carried out classification. To that end, we need to compare the classifier's predicted class memberships to the corresponding ground truth. The ground truth class membership is in general only known from the objects in the labelled training set. Assessing the quality of a classification based on the prediction of the training set is a methodological mistake that can lead to a heavily biased performance measure [16, 17]. To overcome this issue, we need to split the total amount of labelled data samples into at least two parts: a train and a test set. However, since labelled data samples are usually sparse, we will often face the challenge of compromising between a reasonably sized test set, giving us a reliable measure of the classification performance, and a train set, allowing us to accurately train our classifier. A common way of avoiding this compromise is so-called k-fold cross-validation (CV) (figure 2.12).

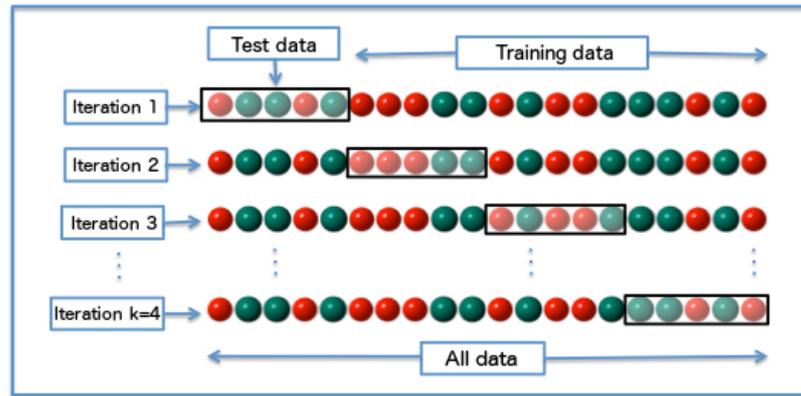


Figure 2.12: K-fold cross-validation

K-fold *CV* is a procedure where the labelled sample set is randomly split into k disjoint sets (folds) of equal size. In an iterative process the performance of the classification is assessed by one of the folds held out as a test set, while all the remaining $k - 1$ folds are used to train the classifier. Given a test set, there are several possibilities to measure the performance. A so-called confusion matrix (see figure 2.13) can give insight into the correct and false predictions per class.

		Predicted label j			
		Type 1	Type 2	...	Type N
True label i	Type 1	$M_{ij} = M_{11}$	M_{12}	...	M_{1N}
	Type 2	M_{21}	M_{22}	...	M_{2N}
	...	⋮	⋮	⋮	⋮
	Type N	M_{N1}	M_{N2}	...	M_{NN}

Figure 2.13: Confusion matrix

As an examples, the entry M_{11} of this matrix gives the number of the correctly classified samples of type 1 normalised by the total number of samples of type 1. However, often it is desirable to assess the performance of a classification with a single figure. An example for such a measurement is the classification accuracy, defined as

$$A = \frac{1}{N} \cdot \sum_i M_{ii}. \quad (2.5)$$

Averaging a performance measure, such as the accuracy, over k iterations of a CV can serve as a reliable measure of the quality of a carried out classification and will be used in this thesis.

2.2.5 Tuning hyperparameters of a classifier

Cross validation can also be used to set the additional user defined parameters of a classifier. This process is also known as tuning of a classifier's hyperparameters [43]. The desired performance measure is used as a criterion based on which the decision for a parameter value is made. Commonly, a suitable range of values is given by the user. This means one can, e.g. train a random forrest with 200, 300 and 400 single decision trees and choose the number of trees that delivers the best classification accuracy. The actual classification of the unknown data is then only carried out using the chosen number of decision trees. If the classifier needs tuning of more than one parameter, this results in a so-called grid search over each possible parameter value combination [43]. Afterwards, the best performing values can be chosen. However, combining a CV with a grid search, i.e. performing a so-called cross validated grid search, can be computationally expensive as it can require the calculation of many possible combinations (i.e. possible parameter combinations \times k-folds).

2.2.6 Dimension reduction

With feature based object representations (section 2.2.2) we are in fact reducing the dimension of an object from infinitely many properties to a finite subset of features. Based on the resulting feature vector, we are able to make an assumption about an object's class membership. One might think that the quality of the assumption increases with the detail of the object's representation, i.e. the amount of features describing the object. However, this is generally not the case. The reasons for this are twofold: First, the higher the dimension, the more difficult it is for a classifier to detect and learn patterns in the data. This is referred to as the curse of dimensionality, and is a common problem in machine learning and pattern recognition. Second, a more detailed description also carries more noise that the classifier is likely to pick up. This effect is called overfitting, and limits the generalisation capabilities of a classifier. To avoid this we can make use of different techniques to reduce the dimensionality of a potentially high dimensional feature space. Dimensionality reduction is commonly divided into feature extraction section 2.2.6.1 and feature selection section 2.2.6.1 methods.

By reducing the dimension of the feature space we not only avoid the curse of dimensionality and overfitting of the classifier, but we can also gain better insight into the classification task and faster models. Additionally, we can use the feature selection methods to get insight into the performance of different feature sets stemming from different data sources. This can be relevant considering data cost, availability and accessibility.

2.2.6.1 *Feature extraction*

In feature extraction, we try to gain a set of new features from the existing representation. This is already partially done by extracting relevant features from the raw sensor data (e.g. shape features from point cloud - section 3.3.1). We can reduce the dimensionality of the resulting feature vector further by employing feature extraction methods, such as principle component analysis (PCA). For this technique, a linear transformation is applied to the original feature space, such that the axis of the transformed space are replaced with axis along the main directions of variance of the data sample. This can help to eliminate noise that is potentially carried by some of the features.

2.2.6.2 *Feature selection*

In feature selection methods, we simply select a subset of the original feature set. There are several possibilities to find the most appropriate subset. The methods can be classified into two different approaches. 1) Filter methods that try to find the best features based on statistical measures prior to the actual classification step. 2) Wrapper methods that evaluate features based on their performance provided by the actual classification algorithm. The latter method can be very time consuming for a large dataset as it requires the calculation of the classifier for each evaluation of a feature subset.

2.2.7 *Summary and conclusion*

Pattern recognition and machine learning introduce the possibility to predict unknown properties (such as the SBST) of objects (such as buildings), even though no complete physical model of the object is known. In our approach we will represent buildings with a feature vector. This feature vector can then be used for a supervised learning algorithm. We will try different classifiers for the prediction and evaluate the results with a CV. As a first step, we have shown how to represent exemplary buildings by their footprint area and year of construction. In the next section we will discuss sources providing such information on a large scale.

2.3 SENSING AND MODELING THE BUILT ENVIRONMENT

2.3.1 Introduction

Data, maps and services provided by national mapping and cadastral agencies (NMCAs) contain a large amount of information about the natural and built-up environment [27]. Building information often focuses on geometry, and semantics are commonly limited to attributes such as address, use and perhaps the construction year of buildings. Explicit information about SBSTs is usually inaccessible (at least in digital form), outdated or simply not available at all. While it is impossible to directly gather such information with remote sensing techniques they at least allow a fast collection of detailed geometric information about buildings that may help to infer their SBST [14].

2.3.2 Sensing the built environment

Historically, data collection of NMCAs was limited to geodetic surveys producing 2D datasets, in which the geometry of buildings is represented with building footprints, i.e. the outline of the building touching the ground.

In the past decade an increasing interest of NMCAs and private organisations towards richer and more realistic representations of the built environment started to shift the collection of geoinformation to 3D datasets. This situation went hand-in-hand with the development of advanced (remote) sensing technologies. These technologies can be loosely classified by their sensors, into photogrammetric approaches (camera), lidar and radar. The sensors can be carried by spaceborne, airborne and terrestrial platforms.

While spaceborne approaches are able to deliver large scale data with high temporal resolution, their 3D positioning accuracy is limited from one to several meters [51, 55]. Terrestrial approaches allow to measure building facades with high resolution and centimeter accuracy. However, city wide acquisition results in big datasets that are not trivial to process. Additionally, they lack information about building roofs. Airborne methods, such as ALS and aerial imagery can be a good compromise between spaceborne and terrestrial methods and therefore represent the biggest part of input datasets for the representation of 3D building geometries [4]. Data acquisition with ALS or aerial stereo imagery usually results in 3D point clouds. A point cloud is an unordered set of point coordinates (X, Y, Z), referenced in a local or global coordinate system (figure 2.14). While lidar and radar sensors mostly deliver geometric information in the form of point clouds, photogrammetric approaches also allow to add visual context by adding additional attributes (such as RGB value) within the same data structure. Recent research also investigates into the combination of different datasets, such as ALS point clouds with aerial and/or terrestrial (stereo) imagery to gain one common and complete textured geometric representation of buildings. However, for many applications the raw data input, needs to be further processed to allow analysis and meaningful visualisation of the data.

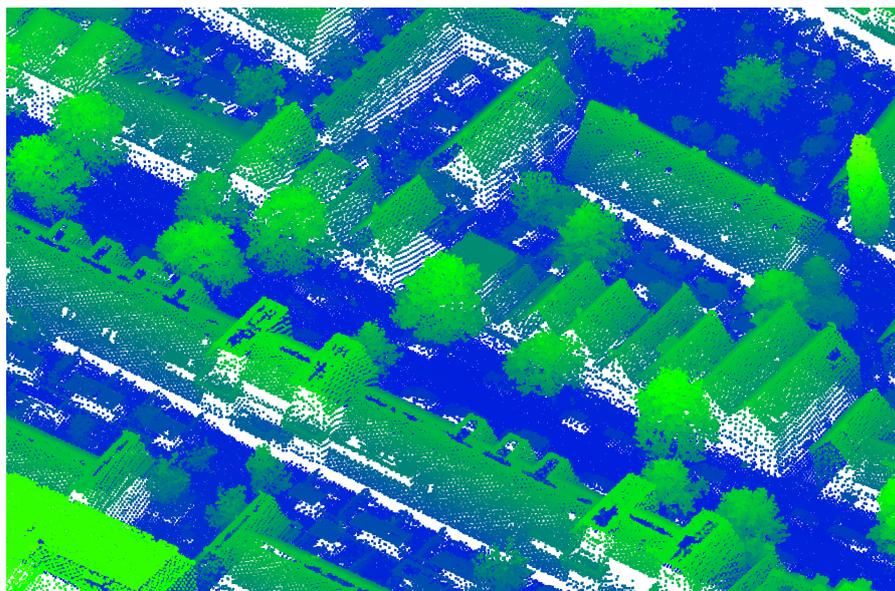


Figure 2.14: Point cloud obtained by ALS. The points are shaded by the value of their Z-coordinate.

2.3.3 Modeling the built environment

A 3D city model may be defined as a collection of common urban objects and structures, with buildings as the most prominent features, described by their boundary surfaces that may be semantically enriched [4, 7]. Conceptual frameworks for such models exist. An example is the CityGML standard by the Open Geospatial Consortium, which defines specifications of building models in different levels of detail (LODs) (see figure 2.15).



Figure 2.15: LODs in CityGML [4]

However the availability of 3D city models is still low, even though 3D reconstruction of buildings from remote sensing data has been an active area of research for more than two decades (see [24] for an early approach). In general, reconstructing the building geometry and representing it with boundary surfaces (boundary representation) or solids (volumetric representation) is the most common approach for 3D building modeling. A variety of methods exist, depending on the input data and the aspired level of detail (for an overview see [25] or [58]). While LOD1 models can be derived automatically, potentially even without elevation data [5], the creation of higher LODs is still a challenging process. Models corresponding to LOD2 are often derived by reconstructing the roof shape from ALS and constraining its extent by the building footprint, e.g. gained from cadastral datasets (from geodetic survey). However, due to the high complexity of building structures, no fully

automatic system is currently available for producing such building models [65]. The creation of LOD3 models, as architecturally detailed models with windows and doors requires a great amount of additional resources and manual modeling effort.

Existing models also differ in their semantic LOD. First, the models may be semantically structured. Unstructured building models only consist of one object per building. Semantically structured building models can be decomposed into building elements such as main structure and extensions, storeys, roof(-segments), chimneys and perhaps even windows and doors. Achieving such a semantic richness already has to be addressed by a suitable reconstruction algorithm. Second, building models may be enriched with additional attributes such as the construction year of the building or its current function. These attributes may be available from other geo-datasets, such as cadastral data, that can be spatially related to the building model. At even higher LODs each building element can have separate attributes such as the material of the roof and walls. However, this type of information is very rare and again requires a great amount of additional resources and modelling effort.

Another valuable data source are already existing building models such as building information models (BIMs), resulting from the design and planning phase of buildings [4]. While this type of data may already explicitly include information about the SBST, it is not commonly available in cadastral or any other type of city wide datasets.

2.3.4 Summary and conclusion

Making a judgment about optimal geo-datasets for predicting SBSTs is not trivial. The choice largely depends on availability of datasets, the scale of the study site, time frame of the project and the necessary or desired accuracy and level of detail of the prediction. By combining 2D geoinformation such as building footprints with semantics such as the year of construction it is already possible to populate some of the GEM attributes (section 2.1.2) directly or with a supervised learning algorithm (section 2.2.3). However, 3D geoinformation can also add vital information, e.g. for the description/prediction of the roof type or detailed building dimensions for analytic vulnerability assessment. On the other hand, this often requires an additional processing step: the construction of detailed and semantically structured building models, such as an LOD2 model or higher.

In this thesis we strive to limit the additional processing step of 3D modeling to a minimum while still incorporating 3D geoinformation in the building classification. We develop a simple approach for modeling a boundary representation of buildings in the form of a polygon surface mesh. We do this by combining an ALS point cloud and building footprint polygons. The advantage of ALS over terrestrial or spaceborne point clouds is that they are often available as one consistent dataset of a whole city or country, while still delivering 3D geographic information with good accuracies (decimeter or better). Additionally, ALS point clouds can often be combined with building footprints, with little to no temporal or spatial mismatch. The building footprints from, e.g. a cadastral dataset, are used to identify outlines of the building and provide semantic attributes, potentially relevant for SBST classification, such as the building year of construction or the primary use of a

building. In our case, both datasets are available as open data on the Dutch geoportal [PDOK](#).

2.4 SHAPE ANALYSIS

2.4.1 Introduction

We have seen that the footprint area of a building can be a relevant feature when trying to predict its [SBST](#). Further geometric properties, locally appropriate to describe the building, such as the span length or the angle of the roof or the gutter height [14] are also discriminative for certain [SBSTs](#). However, extracting generally relevant local properties is not an obvious process. Even though the above mentioned examples are suited to distinguish [LLRSs](#), such as wall based from frame based systems, they lack discriminative power to distinguish them from other [LLRS](#), such as hybrid structures ([LH](#)). Other relevant features may be subject to common building practice in a particular area or construction period. Thus, finding relevant features for a whole building stock can be a cumbersome task, even for an expert in the field. Moreover, extracting these features from a semantically unstructured geometric representation of a building is not trivial. On the other hand, global building properties that are easier to obtain, such as the volume or the surface area of a building might often not be sufficient for describing complex building geometries. It is therefore desirable to describe the geometric shape of a building in a way that all relevant building features can be captured (and extracted). Ideally we would like to make use of the polygonal mesh representing the building geometry (see section 2.3.4). In the fields of computer vision and computational geometry a variety of so-called shape descriptors have been developed, that may be applicable for this case.

2.4.2 Shape descriptors and deep learning

A shape descriptor may be defined as a simplified representation of a [2D](#) or [3D](#) shape in the form of a vector or matrix containing a set of numerical values, or a graph-like structure used to describe the shape topologically [31]. Generally spoken, any retrieval or classification algorithm for shapes requires such a representation, for measuring similarity between two shapes [32]. Shape descriptors may be divided into the two categories of local and global descriptors. The general idea of the former approach is to compute multiple local shape features at sampled points of the shape. Global descriptors can be seen as a mapping from the space of [2D](#) or [3D](#) objects to some finite-dimensional vector space. A vector, defined by one to many local or global shape descriptors, encodes the information about the object's shape by storing numerical attributes. By defining a distance function, such as the euclidean distance, these vectors can be used to assess the similarity between different shapes. Shape descriptors can be implemented on [2D](#) data, representing single or multiple views of the original shape, e.g. in the form of an image or a depth map, or they can be implemented natively on [3D](#) representations, such as point clouds, polygon meshes or voxel grids.

Traditionally, shape descriptors were often hand designed according to a particular geometric property of the shape [54]. Examples are surface normals, curvatures, distances or angles gathered at samples points, or areas and volumes of a shape. More advanced shape descriptors can be properties of functions defined on a representation of the shape. However, a common problem with such hand designed descriptors is that every subsequent process, such as shape classification, is constrained by the representation power of the extracted shape features [45]. To solve this problem, architectures using ANN (or variations thereof) have been developed that combine the process of feature extraction and shape classification into one algorithm. So-called deep learning architectures use ANNs with multiple layers. So far, these architectures have proven to be particularly useful in image classification tasks. As the recognition of 3D models has become more and more important in recent years [54], ANNs that can directly process 3D input are also heavily investigated [63]. A variety of methods for deep learning on geometric shapes have been proposed in recent years (see [64] for an overview). Some of these methods may be appropriate to use for building classification. Wu et al. [63] use a convolutional neural network (CNN) that operates on a 3D voxel grid. They achieve promising results for 3D model classification. Creating a voxel representation of a building may even be achieved with less effort compared to the creation of a boundary representation. Qi et al. [45] developed a method that can directly operate on a point cloud. However, to include the 3D building geometry a reconstruction of the building may still be necessary, as the density of an ALS point cloud is very sparse on the building walls. Another approach would be to use view based 2D data as an input for a deep learning method. Besides the use of aerial or terrestrial imagery (section 5.2.2.2) it could be possible to use a point cloud as a 2D raster image that represents the outline of the building and a height value of the roof in a grid structure.

However, choosing the right shape descriptor or ANN architecture may be a difficult process. In general, there is no single descriptor that is best for all classifications [52]. Additionally, the choice may depend on the available input data (i.e. the shape's representations) and the desired properties of the descriptor. Different descriptors may be invariant to certain transformations of the shape (such as rotation, translation, scaling) making it e.g. possible to match shapes irrespectively of their size or orientation in their current embedding.

2.4.2.1 Shape DNA

Reuter et al. [48] were the first to use the spectrum (i.e. the eigenvalues and eigenfunctions) of the Laplace-Beltrami operator (LBO) of 2D and 3D manifolds as a shape descriptor. In the continuous case, the LBO is defined as

$$\Delta f := \operatorname{div}(\operatorname{grad} f), \quad (2.6)$$

where grad and div are the gradient and divergence on the manifold. The Laplacian eigenvalue problem is given as [48]:

$$\Delta f = -\lambda f. \quad (2.7)$$

The LBO is intrinsic and as a result, it is invariant to isometric (metric-preserving) deformations of the manifold [39]. For this reason it has been

used for the analysis and retrieval of non-rigid shapes (such as organic objects), where deformations are often near isometric [35]. However, non-isometric transformations change the spectrum continuously [35] so the spectrum may also be adequate for describing the shape of buildings and subsequently classifying them.

Reuter et al. [48] also define different discretisations of the LBO, such as the linear FEM operator. This discretisation can be used on a triangular surface mesh $M(V, E, F)$ with vertices $V = \{1, \dots, N\}$.

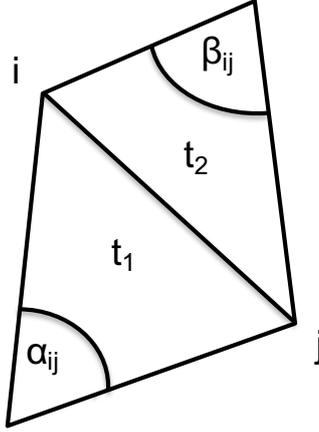


Figure 2.16: Example of a polygon mesh

The mesh has to be 2-manifold, meaning each interior edge $(i, j) \in E$ is shared by exactly two triangular faces t_1 and $t_2 \in F$, and boundary edges belong to exactly one triangular face (see figure 2.16). In this case the LBO is given as an $N \times N$ matrix $\Delta = A^{-1}B$, where the stiffness matrix

$$A(i, j) = \begin{cases} \frac{\cot\alpha_{ij} + \cot\beta_{ij}}{2} & (i, j) \text{ edge,} \\ -\sum_{k \in N(i)} A(i, k) & i = j, \end{cases} \quad (2.8)$$

and the mass matrix

$$B(i, j) = \begin{cases} \frac{|t_1| + |t_2|}{12} & (i, j) \text{ edge,} \\ \frac{\sum_{k \in N(i)} |t_k|}{6} & i = j, \end{cases} \quad (2.9)$$

with $|t_i|$ being the area of the triangle t_i [47]. The first $n \leq N$ eigenvalues of the LBO can be computed by performing the generalised eigendecomposition

$$A\mathbf{f} = -\lambda B\mathbf{f}, \quad (2.10)$$

where $\mathbf{f} = (f_1, \dots, f_n)$ is an $N \times n$ matrix containing as columns the first n discretised eigenfunctions and $\lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ is the diagonal matrix of the corresponding eigenvalues [39]. A truncated version of λ can be used

as a feature vector that describes the geometric shape of the underlying mesh. Reuter et al. [48] named this representation Shape DNA. However, a mesh can only approximate the true underlying manifold. Simply put, the more dense a mesh, the better it can approximate the manifold. Thus, a dense mesh is desirable, especially in areas with concavities or with high surface curvature, since most eigenfunctions will need fine meshes in these areas. However, we do not know beforehand where a dense mesh is needed to represent a specific eigenfunction (and the corresponding eigenvector) [48]. Globally dense meshes may have a large number of vertices and thus make the solution of equation 2.10 difficult or at least very time consuming. How dense a mesh needs to be for efficient shape analysis, such as shape comparison, is problem dependent and cannot be said in general.

Furthermore, it is unclear as to what number of eigenvalues, i.e. n , should be used to form the Shape DNA. Different publications used 11 [47], 20 [42] or 10-15 [35] eigenvalues for shape analysis and retrieval. Arteaga [1] defines the LBO directly on a point cloud and reports about using 50 eigenvalues for accurate shape matching. Gao et al. [20] also discuss this topic and use at most 100 eigenvalues for shape description and [49] mentions that 500 eigenvalues had to be computed for extracting important information from eigenvalues. In view of signal processing, more eigenvalues contain more information of detail and can describe the shape more accurately [20]. However, more eigenvalues can also carry information about non-isometric deformations which might make the detection of shape similarities difficult, especially in view of building classification.

Besides being an isometry invariant, Shape DNA has another nice property: by normalising it, the spectrum can also be made scale-invariant. In [48], Reuter et al. propose different methods for normalising the spectrum, such as dividing it by its first non-zero eigenvalue, multiplying it with the surface area of the underlying manifold, or (according to Weyl's law) dividing by the factor c of the fitting curve

$$f(x) = cx^{\frac{2}{d}}, \quad (2.11)$$

fitting $f(n) := \lambda_n$.

2.4.3 Summary and conclusion

In this thesis we make use of several local and global shape descriptors. Shape features that are easily extractable from the building representation, such as the footprint area, footprint perimeter, or the surface area of the building are used. We also aim to extract information about the roof of the building, such as the gutter height or the number and angle of separate roof segments. However, since it is not straightforward how to define and extract these features and, since this combination of only local shape descriptors might not include all the relevant shape information, we also make use of the so-called Shape DNA as a global shape descriptor of buildings. This shape descriptor can be defined on a 2-manifold triangle mesh and describes the global shape of a building with a one-dimensional vector. This vector can easily be used as a feature input for all of the classification algorithms presented in section 2.2.3. We will initially make use of 50

eigenvalues simply because this seems like a good compromise between the above mentioned numbers and a feasible amount of eigenvalues to extract in view of data handling. We will not investigate deeply into the amount of eigenvalues to be used, however, we do not expect a great influence when altering this number. We will investigate into the density of the mesh, since it is not clear how this will impact building classification. To the best of our knowledge, Shape DNA, or its variants, have not yet been used for shape classification and neither for describing non-rigid shapes.

2.5 AUTOMATIC SBST PREDICTION

2.5.1 Introduction

We have found around a dozen approaches that investigate into large scale seismic assessment with remote sensing data or other types of (geo-)information. There are a few approaches dealing with seismic assessment on city level, however, the developed methods are often too broad and do not have enough spatial detail [38]. Other approaches lack in their typological detail when describing SBSTs [29]. Besides pre-event analysis for seismic risk and vulnerability assessment, there is also work dealing with post-event analysis. He et al. [26] for example define a shape descriptor to detect damaged roofs from an ALS point cloud after the 2010 Haiti earthquake.

Automatic prediction of SBSTs that allow a more detailed pre-event seismic vulnerability assessment has only been attempted in a few, largely heterogeneous approaches [9, 14, 21, 36, 44, 50]. The works differ in their input data, from terrestrial images [44] to aerial imagery and ALS [36] or a combination thereof. They extract topologic [21, 50], geometric [9, 14, 21, 36, 44, 50], spectral³ [9, 21, 36, 44] and geographic features on building and building block level [22, 62]. The biggest variations are reflected in the predicted typologies. Borzi et al. [9], Geiß et al. [22] and Lugari [36] use their own definitions of building typologies, while [62] predict typologies according to the World Housing Encyclopedia (WHE) and European Macroseismic Scale (EMS), and [50] predict different HAZUS-MH⁴ classes.

2.5.2 Related work

Borzi et al. [9] estimate vulnerability functions of industrial structures such as tanks, pipes and chimneys using aerial and satellite imagery. However, their approach includes many assumptions and is tailored to their specific typologies and therefore largely unfeasible for residential buildings.

In the works of Borfecchia et al. [8] and Lugari [36] the vulnerability of the building stock of Avellino, Italy is estimated. In a similar approach to ours, they extract detailed geometric parameters (area, perimeter, volume, type and complexity of roof, total height) of buildings by combining an ALS point cloud with building footprints. They classify roofs into 5 different classes (flat simple, flat multi-level, pitched, complex). Unfortunately they

³ In this section *spectral* refers to the electromagnetic spectrum.

⁴ <https://www.fema.gov/hazus>

do not further specify this procedure. In addition to the geometric features they add another 31 features as the mean and standard deviation of hyperspectral bands from aerial and satellite imagery. This can be done in a similar way to extracting geometric features, by spatially relating the images to the building footprint. Lastly they also approximate the building year by a multitemporal analysis of the satellite images. With this feature combination they train an ANN with 60 in situ training samples to predict either masonry or reinforced concrete buildings. They state that the footprint area was identified as the most important feature by the classification software. Based on 170 test samples they achieve a classification accuracy of 85%. In a last step they link the building types to a predominantly conducted vulnerability study based on post-event data. This significantly hampers the transparency and transferability of their approach.

Wieland et al. [62] developed and partially implemented a methodology to assess seismic vulnerability of building stock on different scales. First they segment satellite images of the city of Bishkek, Kyrgyzstan to identify homogeneous urban structures. In a next step they classify the segments according to the age and landuse/landcover (LULC) of the built-up areas using a SVM. This step is based on the idea that neighboring buildings share vulnerability proxies such as age, typology or materials of construction [62]. The used classes are masonry, concrete, industrial/commercial, mixed built-up or not built-up areas. For the LULC classification, training samples are generated by visual inspection of the satellite images by local experts. The feature vector is composed of spectral values (referring to the electromagnetic spectrum) extracted from the satellite images. Training samples for the identification of the age stem from multitemporal satellite images. The classification accuracy for this step is stated to be around 80% based on 10 test samples per class. In a follow up paper, Pittore and Wieland [44] classify structures on building level into classes according to the World Housing Encyclopedia (WHE). For this classification they only rely on the height of buildings extracted from terrestrial images and the previously predicted block type (LULC and age) containing the building. The terrestrial images were gathered on stratified samples per LULC class. They do not give any performance measure for this part of the implementation, acknowledging that the building classification still needs further research. In a last step Pittore and Wieland [44] also infer vulnerability of the buildings according to European Macroseismic Scale (EMS). However, this step only relies on statistics compiled from WHE data. We think, based on their current implementation, that the building typology prediction is far from being precise and subsequent vulnerability estimation is only based on vague inference from already available vulnerability models. The terrestrial images, however, potentially allow to extract more information on building level, which enables the possibility of a more precise estimation of typology and vulnerability. Additionally, the LULC classification can be useful to give a coarse overview of the city, provide a basis for the collection of training samples, or for extrapolating results from building, to block or even city scale.

In a pioneering work, Sarabandi [50] use decision trees and high resolution satellite images to estimate SBSTs defined by the HAZUS-MH earthquake model. Training samples are gained by the use detailed tax assessor data, which they translate into SBST classes defined by the HAZUS-MH earthquake model. More than 40 classification models with different feature and training sets are calculated. They are able to identify specific HAZUS-

MH classes (14 different types of concrete, steel, masonry and wood structures) using features such as height, footprint area, year of construction, roof type, configuration in plan view and occupancy type (commercial, industrial, mixed, residential), with classification accuracies up to 85%, measured with 10-fold **CV** on almost 2000 training/test samples. Similar to our approach they develop shape descriptors to represent geometric and topological properties. The building configuration in plan view is described by three different shape descriptors (slenderness, convexity, irregularity index) using the footprint polygon as input. Additionally, they describe the roof type of a building with the values flat, low slope, steep slope extracted from **3D** reconstructions of stereo image pairs. Unfortunately they do not make a clear judgement about the influence of these parameters on the classification result.

In their latest work, Geiß et al. [22] use detailed building information gathered by post-event inspections in Pandang, Indonesia to supervise **RF** and **SVM** classification models. Next to geometric and spatial features on building level, they also use spatial features, as well as semantic features on building blocks. Additionally, attributes from multispectral satellite images, such as brightness, contrast or colour of the building roofs are taken into account. Ending up with almost 120 features per building, they also describe different feature selection techniques to identify the most relevant features. The results show that the ratio between the footprint area and perimeter is the most important feature in their classification process. Other important features that they derive in an earlier publication are [23]: the structure type on block level describing the urban morphology (slums, suburbs, low income areas, medium income areas and high income areas), grey level difference vector for some of the spectral bands, floor space.

Geiß et al. [22] also address one of the biggest problems when using a supervised learning approach: the quality and lack of complete training samples. In situ **SBST** ground truth is time consuming and costly to obtain and at the same time afflicted with uncertainties induced by a challenging assignment process [22]. For this reason they first deploy a one-class **SVM** to eliminate training samples for which the object label cannot be regarded as reliable. In a next step, they generate synthetic training samples, by oversampling existing ground truth data. They thereby ensure to have at least 60 training samples for each of the seven classes confined masonry, reinforced concrete high, reinforced concrete low, steel frame, timber frame residential, timber frame non-residential and unreinforced masonry. Using all of their extracted features leads to a classification accuracy over 90% using a **SVM** measured with 10-fold cross validation. However, they acknowledge that the accuracy estimates are very optimistic by being at least partially based on the synthetically generated training samples.

2.5.3 Summary and conclusion

In this thesis, we follow the idea of a supervised classification by using in situ building inspections as training samples. However, we propose the use of significantly different input and target variables for this approach. Several of the aforementioned studies mention the importance of geometric features as relevant proxies to determine the **SBSTs**. Thus, we make use of Shape DNA to describe the building geometry in a complete way, beyond

footprint characteristic or discrete roof types. This can enable the machine learning algorithm to detect geometric and topological patterns inside a SBST class, at any position of the building. Furthermore, we chose to predict the SBST described by the GEM building taxonomy. The taxonomy was developed to classify buildings in a uniform manner and enable large scale seismic risk assessment. It is modular and designed to be applicable on a global level and can therefore be adopted to several use cases.

However, we also acknowledge some of the shortcomings of our method. Several of the aforementioned studies use visual information, e.g. extracted from satellite, aerial or terrestrial images. While we cannot say with certainty that this information is important, we believe that it is worth to investigate into its use, especially considering the vast amount of sources for this type of data. Furthermore, in our approach we only use information on building level for the classification. Generally, the building geometry allows characterising individual buildings, whereas information on building block level characterises the geographic setting in which the respective buildings are embedded in [22]. This can be valuable information in many stages of the process. In the classification process it can help to prevent misclassifications due to unlikely geometries of single structures. It can provide information on different spatial scales, also with regard to data collection. This should be address in future implementations. Furthermore, some of the presented works deliberately try to restrict their approach to globally available dataset, such as satellite imagery. These methods are well justified by the lack of data availability in many countries. In this thesis we investigate into the potential gain in detail of the estimated typologies if detailed (geometric) data is available. We also acknowledge that our approach gains from (is enabled by) the favourable situation of openly available geo-data which may not be the case in other countries.

3 | METHODOLOGY

This chapter describes the general methodology that was used in this thesis to classify SBSTs as summarised in figure 3.1. The terms used in figure 3.1 are consistent with the (sub-)section names in this chapter. Furthermore, in section 3.2 we describe the generation of synthetic building models that are mainly used for investigations into shape analysis of buildings.

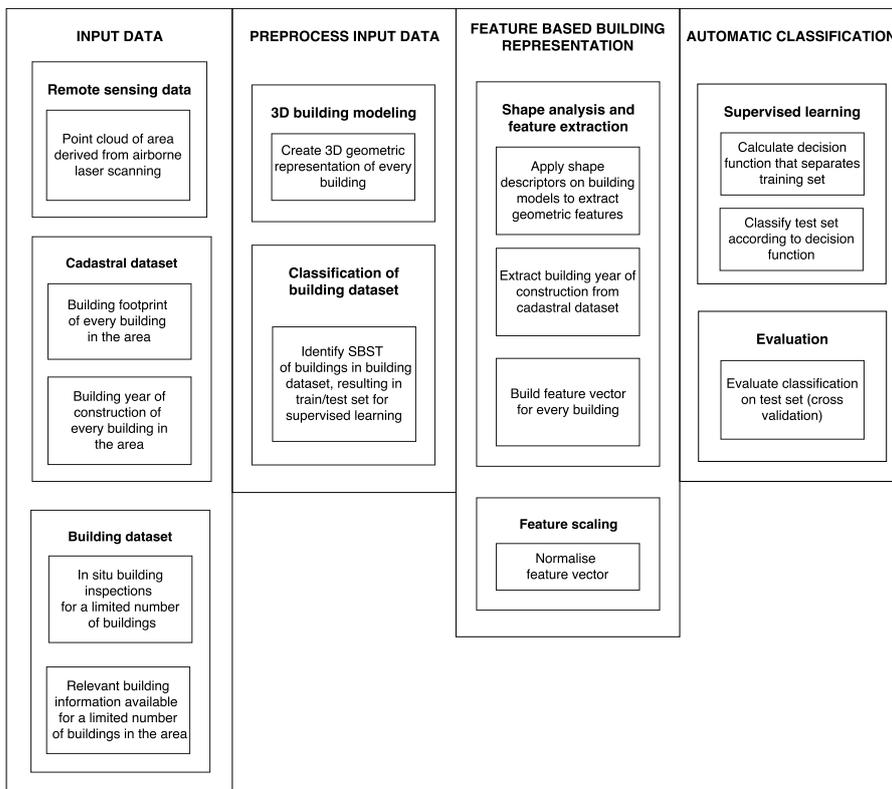


Figure 3.1: Summary of the method of our approach. The terms used in this diagram are consistent with the (sub-)section names in this chapter.

3.1 PREPROCESS INPUT DATA

3.1.1 3D building modeling

In this section we report on the generation of 3D building models. Using the ALS point cloud of the area (figure 3.3a) and the building footprint polygons we reconstruct the boundary surface of every building and represent this surface with a polygon mesh. The full process is depicted in the

flowchart in figure 3.2. The polygon mesh, as well as intermediate results of the modeling process are used for the shape analysis of the buildings (section 3.3.1).

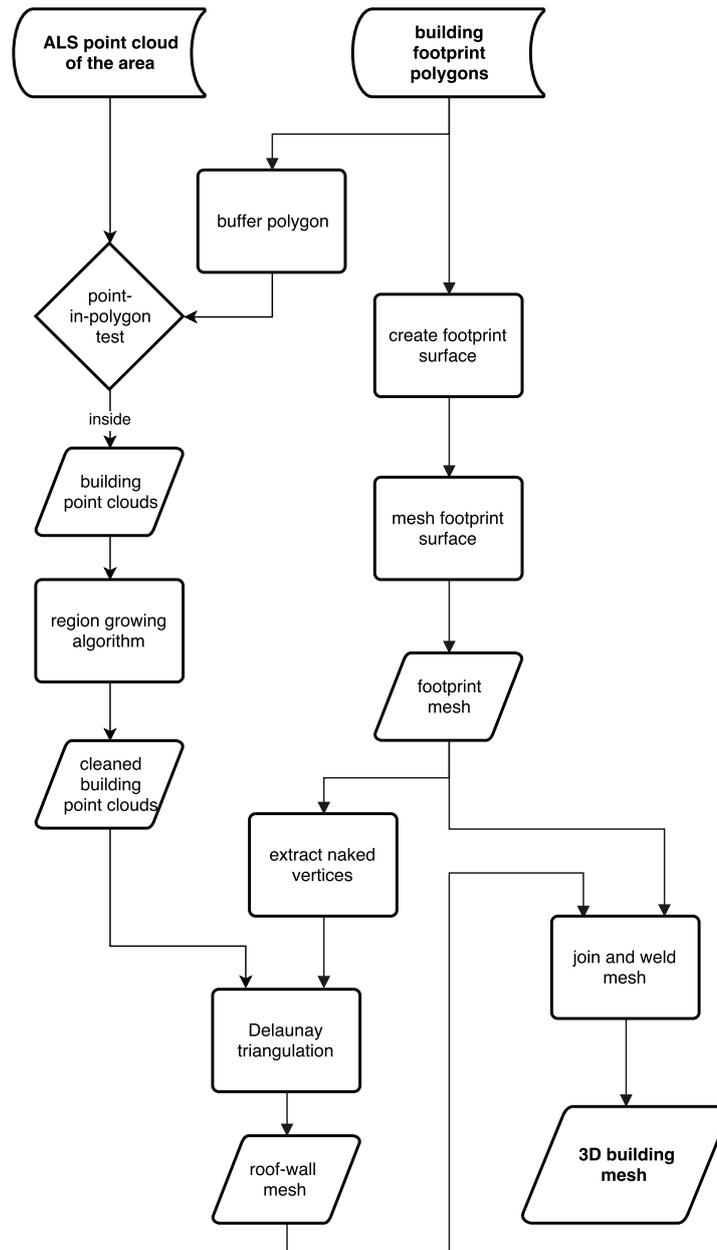


Figure 3.2: Flowchart of 3D building modeling

In a first step we create a buffer around the line segments of each building footprint polygon (figure 3.3a). We identify the points of the area point cloud that lie inside the buffered footprint polygons and subsequently associate these points with the corresponding footprint. This results in a "small" point cloud of every building, which we will call the building point cloud in the following (figure 3.6d). The buffer around the footprint is used to make sure that the building point cloud only consists of points that lie on the roof of the building. This is important for the meshing algorithm (see section 3.2.3). However, in many cases not all the points on vertical walls

can be removed in this way. In figure 3.6d an example of such points can be seen on the left and front wall of the building. Furthermore, the building point cloud may still include points of trees, or other objects that lie inside the building polygon and occlude the roof (figure 3.3a). To remove such points and remaining points on the wall we have to further clean the building point clouds. This will be done by applying a region growing algorithm to each building point cloud.

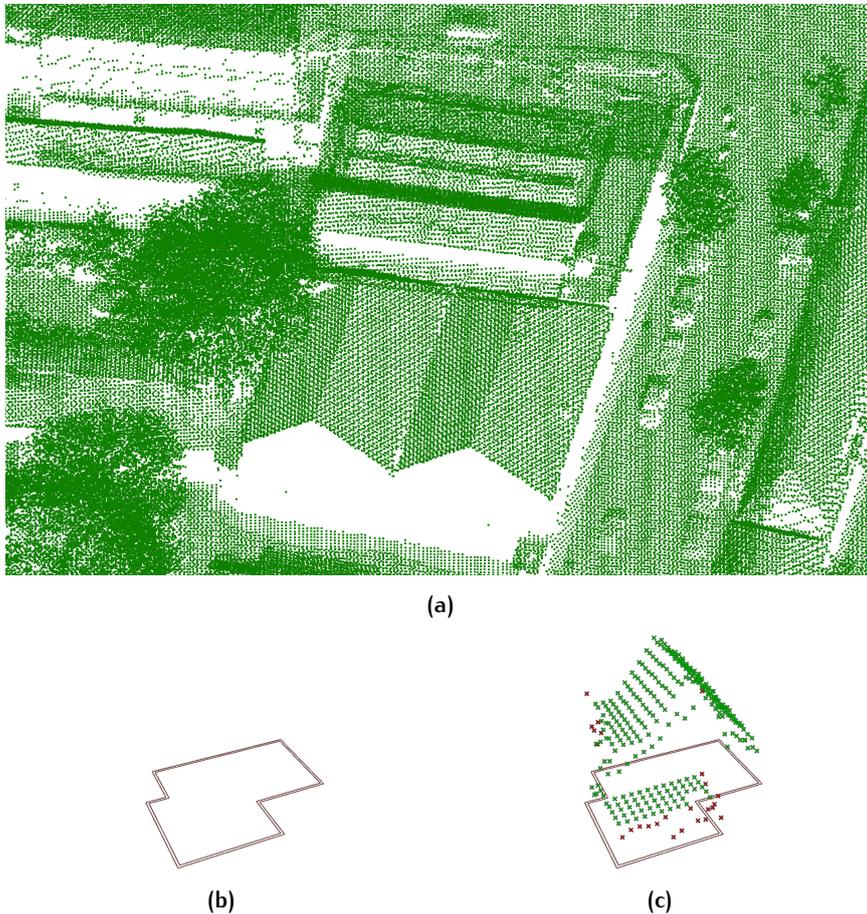


Figure 3.3: The steps of processing an ALS point cloud to polygon meshes of buildings: (a) input point cloud (visualised in CloudCompare); (b) polygon footprint of an example building; (b) building footprint with associated points of point cloud; (c) region growing algorithm applied to building point cloud; (d) polygonal mesh of building

3.1.1.1 Region growing algorithm

The region growing algorithm can identify planar regions in the building point clouds. These planar regions are very likely to reflect the roof structure (as separate roof segments) of the building. Subsequently, we can remove points of the building point cloud that are not a part of a planar region, and thus, not likely to be a part of the roof. A flowchart of the region growing algorithm is depicted in figure 3.4.

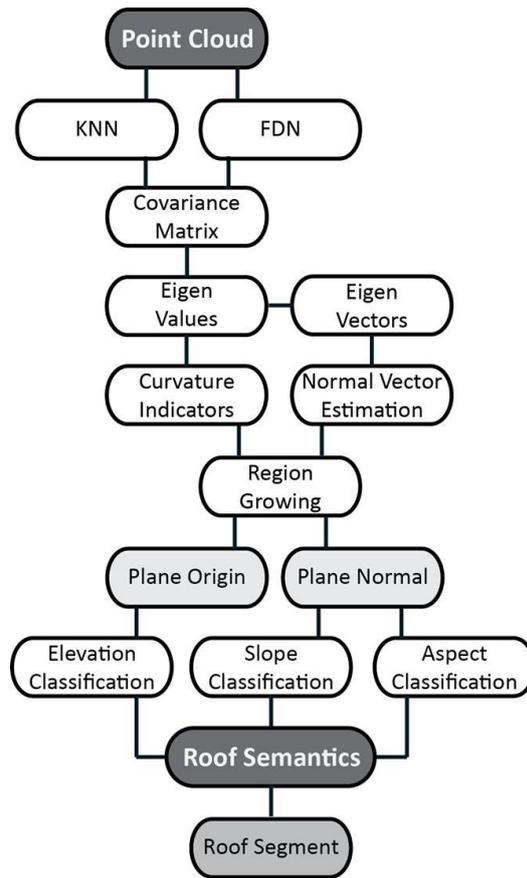


Figure 3.4: Region growing algorithm

In a first step it is necessary to define a neighbourhood for every point in the building point cloud. This can be done either with a k -nearest neighbour¹ algorithm 3.1 or fixed distance neighbourhood (FDN) algorithm 3.3 algorithm.

The first approach is independent of the resolution of the local density of the point cloud and always gives k -neighbours. The second approach, however, requires an understanding of the distribution of the points inside the point cloud. The range-parameter has to be big enough to always return at least two neighbours for every point in order to avoid complications in the following steps. For highly irregular point clouds a dynamic estimation of the range could be a possibility. In a covariance matrix as a dispersion indicator for every neighbourhood is calculated. for each point A as Point3D in the point cloud calculate centroid C as Point 3D from neighborhood of point A for each neighbor k as Point3D N end

This returns a 3×3 symmetric matrix for each point in the building point cloud. A principal component analyses (PCA) of this matrix gives an idea of the data distribution inside the neighbourhood. The first eigenvector of the covariance matrix is the direction in which the points in this neighborhood vary the most. The second eigenvector indicates the direction with the most variation orthogonal to the thirist one. Therefor the third eigenvector is a

¹ This algorithm is in principle the same as the k -NN algorithm in pattern recognition (section 2.2.3.2), but will be used slightly different here, and is thus treated separately.

Algorithm 3.1: K-nearest neighbour algorithm

```

1 K-nearest neighbour algorithm ( $P, k$ ) :
   Input:
       an array  $P$  of points point3d
       in integer  $k$  as the number of neighbours
   Output:
       an array  $N$  with arrays of  $k$  point3d neighbours for each point
2  $N \leftarrow$  initialise array  $N$  with  $|P|$ ;
3 for  $p1$  in  $P$  do
4      $D \leftarrow$  initialise 2d array  $D$  with  $|P|$ ;
5     for  $p2$  in  $P$  do
6          $d \leftarrow$  Euclidean distance between  $p1$  and  $p2$ ;
7          $D[1, i].append(d)$ ;
8          $D[2, i].append(p2)$ ;
9     sort  $D$  by increasing value of  $D[1]$   $n \leftarrow$  array of second to  $k + 1$ 
       entries of  $D[2, :]$ ;
       // first entry of  $D[1]$  is zero as it is the distance of
        $p1$  to itself
10     $N.append(n)$ ;
11 return  $N$ 

```

Algorithm 3.2: Fixed distance neighbourhood algorithm

```

1 Fixed distance neighbourhood algorithm ( $P, d$ ) :
   Input:
       an array  $P$  of points point3d
       a float  $d$  as the maximum distance between to points
   Output:
       an array  $N$  with arrays of point3d neighbours for each point
2  $N \leftarrow$  initialise array  $N$  with  $|P|$ ;
3 for  $p1$  in  $P$  do
4      $n \leftarrow$  initialise list  $n$ ;
5     for  $p2$  in  $P$  do
6          $s \leftarrow$  Euclidean distance between  $p1$  and  $p2$ ;
7         if  $s \leq d$  then
8              $n.append(p2)$ ;
9      $N.append(n)$ ;
10 return  $N$ 

```

Algorithm 3.3: Calculation of covariance matrix

1 **covariance_matrix** (N) :

Input:

an array N with points p and their neighbours n

Output:

an array C with covariance matrices for each point

```

2  $N \leftarrow$  initialise array  $N$  with  $|P|$ ;
3  $C \leftarrow$  initialise array for covariance matrices ;
4 for  $p$  in  $N$  do
5    $M \leftarrow$  initialise matrix  $M$ ;
6   for  $np$  in  $n$  do
7      $x, y, z \leftarrow$  X, Y and Z distance of  $np$  and  $p$ ;
8      $M.append([X, Y, Z])$ 
9    $C.append(c)$ ;
10 return  $C$ 

```

good estimation for a normal vector of this set of points. The eigenvalues of the covariance matrix are calculated with a trigonometric solution of the covariance matrix? characteristic equation [53]. This is an efficient way to get the eigenvalues of a 3×3 symmetric matrix in descending order. The third eigenvector can then be calculated by using the Cayley-Hamilton theorem. Additionally, the third eigenvalue normalised by the sum of eigenvalues gives a good indication of the curvature in each point.

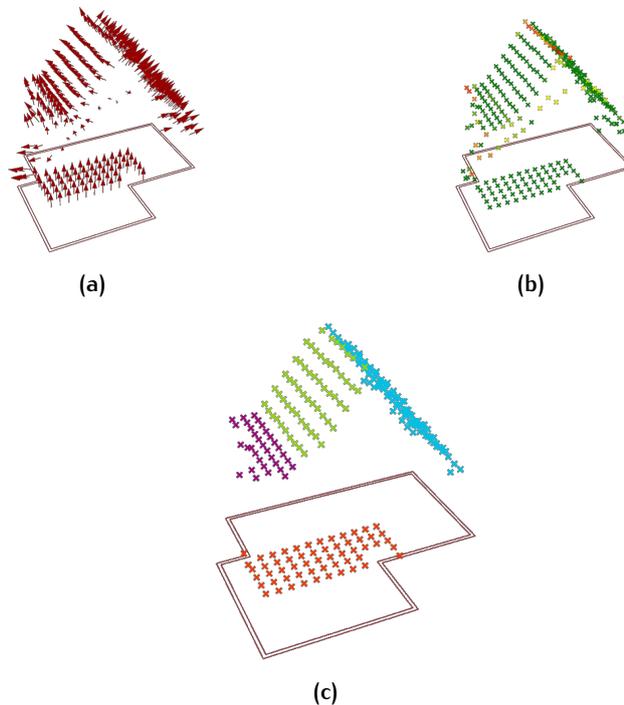


Figure 3.5: Region growing algorithm: (a) estimated normal vectors, (b) estimated curvature, (c)

Based on curvature and normal vectors for every point in the building point cloud we now apply the region growing algorithm algorithm 3.4.

Algorithm 3.4: Region growing algorithm

```

initialize List B of already processed Points
initialize List of Regions
for i = 0, i < Pts.Count, i++
    initialize CurrentRegion and CurrentSeeds as List of Point IDs
    initialize Seed as ID of Pt with min Curvature
    if Point is already processed
        continue
    CurrentRegion.Add(Seed);
    CurrentSeeds.Add(Seed);
    for z = 0, z < CurrentSeeds.Count, z++
        initialize list of neighbours for current seed
        for k = 0; k < neighbourhood.Count; k++
            initialize current neighbour as Pj
            if B does not contain Pj and
            angle between Normal of Pj and Seed is below threshold
                add Pj to CurrentRegion
                if curvature of Pj is below threshold
                    add Pj to CurrentSeed

                    add Pj to B
                    set Curvature of Pj to 99999
        if CurrentRegion.Count > RegionTh
            add CurrentRegion to List of Regions
    add Seed to B
    set Curvature of Seed to 99999
1

```

This returns an integer indicating the region a point belongs to. By setting a threshold to the region size we do not assign points to a region that consists of less than 15 points. Doing this we identify the "main" segments as the roof segments of the building point cloud figure 3.5c. Besides cleaning the building point cloud the region growing algorithm is used to extract local shape features of the building (section 3.3.1).

3.1.1.2 Generate polygon mesh of building

In this section we report on the generation of a polygon mesh for every building that will serve as an input for the shape analysis (section 3.3.1). To apply the finite element model (FEM) discretisation of the LBO presented in section 4.4.1.3 we need a 2-manifold triangle mesh. So far, we have the cleaned building point cloud consisting only of points on the roof of the building. To include information about the bottom part of the building, we now make use of the footprint again. The building footprint is a planar polygon. We can describe the inside of this polygon with a surface and approximate the surface with a mesh (figure 3.6).

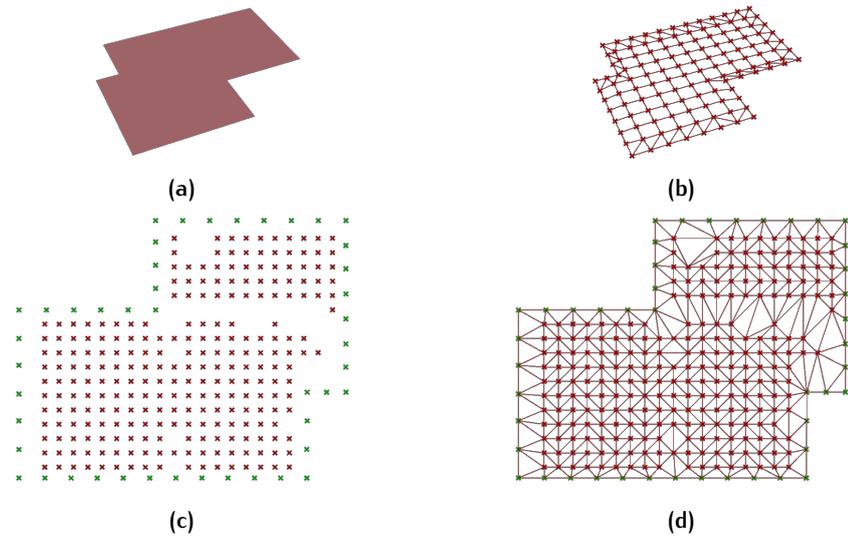


Figure 3.6: Footprint meshing: (a) surface trimmed with footprint curve; (b) footprint approximated with mesh; (c) naked vertices of footprint mesh (green) and cleaned building point cloud projected to XY plane (red).

We mesh the surface with the same density as the building point cloud. Now we extract only the naked (outer) vertices of the footprint mesh. Furthermore we project the cleaned building point cloud on the global XY plane in which also the footprint mesh lies. In this 2D space, the Delaunay triangulation (DT) provides a fast and good triangulation of a number of points N . We apply a DT to the points, but only edges that are inside the footprint polygon. Figure 3.7 shows the resulting mesh in 3D. As a result of our process the mesh has a much higher density on the roof and ground of the building than on the walls. This may be unwanted for the extraction of Shape DNA section 4.4.1.3. For this reason we apply a topological smoothing to the building mesh with the goal to even out mesh density (figure 3.7).



Figure 3.7: Different building meshes: (a) raw mesh after Delaunay; (b) smoothed mesh.

3.1.2 Classification of building dataset

In this process existing information about the building stock is used to label around 20.000 buildings of the Groningen building stock. The result of this process will be used as the label training set for the automatic building classification developed in this thesis. The building dataset stems from different sources, including detailed inspections of building plans and in situ buildings. Furthermore, a dataset containing every agricultural building in the area is included. Within the context of the Groningen Earthquakes Structural Upgrading project, Arup Amsterdam developed a methodology to process this information to gain i.a. the six attributes (see section 2.1.3) defined by the GEM taxonomy for each inspected building. A conceptual overview of this process is depicted in figure 3.8. The details of the labeling process are out of the scope of this thesis. However, in the following we mention some of the characteristics that were relevant for the design of the experiments conducted in section 4.4.2.

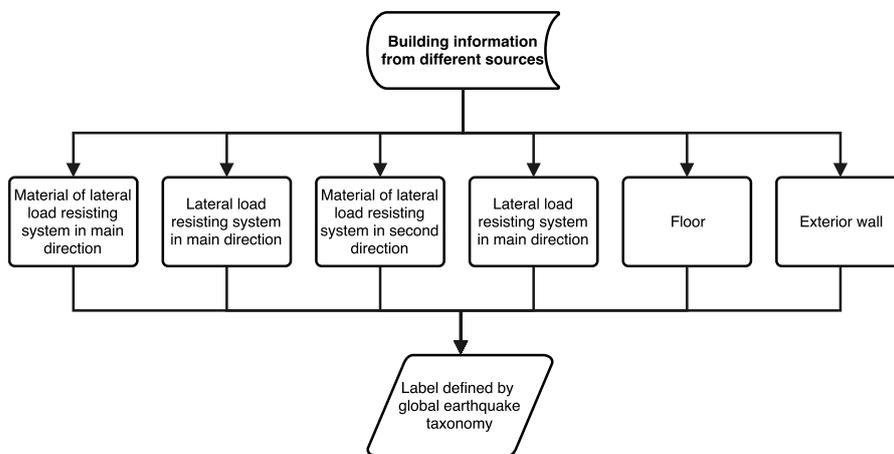


Figure 3.8: Classification of building dataset

- For several reasons it is not always possible to identify each of the six attributes for every building in the labeling process. In this thesis we consider the distinct combinations of the attributes as one class label and therefore only use buildings for the training set that are fully labelled with all six attributes. Otherwise there would be classes such as `unknown_LH_unknown_LH_FW_EWN`. A way to circumvent this problem and increase the labeled training set could be to predict each of the six GEM attributes separately. Although this is possible in a so-called multi-label classification, we decided to not use this technique in this thesis. The reason for this is that patterns formed by classes represented with the combination of several attributes may be easier to detected than those from classes grouped by single GEM attributes. As an example, a building with a FW can still occur in many different shapes, while the combination with other GEM attributes describes the building more precisely.
- The distinct combination of the attributes results in more than 30 different classes that are present in our dataset. However, most of these classes are only comprised of around 10 to 100 buildings. In this the-

sis we only include **SBST** classes with more than 100 samples in our training set, to get a more reliable measure of the classification accuracy of these **SBST**s. There are eight **SBST** classes with more than 100 inspected buildings in our dataset.

- Stemming from the dataset including only agricultural buildings, an educated guess is made towards the **SBST** of these buildings. All buildings in this dataset are labelled with the **GEM** combination **MUR_LH_MUR_LH_FW_EWN**. This guess is not always correct and may therefore also influence the measured classification accuracy.
- The remaining seven **SBST** classes are only comprised of residential buildings. For this reason we do not include the main use of the building as a feature as it would immediately identify agricultural buildings.
- In a previous classification all buildings that are directly adjacent to a similar building were identified. These are terraced houses which are assigned to have no lateral load resisting system (**LN**) in the second direction. For this reason we do not include an adjacency feature in the classification as it would immediately identify six out of the eight **SBST** classes as terraced houses.

3.1.3 Summary

In this section we have shown how to generate a **3D** building model from the **ALS** point cloud and building footprints. The building model, and intermediate results of the modeling process are used for the shape analysis of the buildings in section 4.4.1. The need for a shape analysis is based on the assumption that the shape of a building is (partially) determined by its **SBST**. Furthermore, we have shown how the building dataset is classified into groups of similar **SBST**s described by the **GEM** taxonomy. This process is often based on assumptions and therefore the resulting training set can not always be considered as ground truth. In this thesis we are also interested in the shape analysis of the buildings by using different shape descriptors, such as Shape DNA. To design and conduct experiments without the influence of the assumptions in the building classification process (section 3.1.2) we generate a synthetic training set of building models with different shapes.

3.2 SYNTHETIC **3D** BUILDING MODELS

3.2.1 Building typologies

In this section we report on the generation of synthetic **3D** building models. We aim to create building models similar to the ones shown in the previous section. However, we focus on building models with a distinct geometric shape. More precisely, we model buildings with a flat, shed and gabled roof type. Furthermore, we also add extensions with different sizes to the building models.

3.2.2 Procedural modeling

In a first step we choose a point in 3D space O . Next we construct a boundary rectangle spanned by two vectors width $\vec{w} = \vec{OW}$ and length $\vec{l} = \vec{OL}$. Then we extrude this rectangle in the third dimension by a vector height $\vec{h} = \vec{OH}$. For the flat roof type the resulting geometry is now simply closed with boundary surfaces. The two other roof types are modelled using simple trigonometric functions.

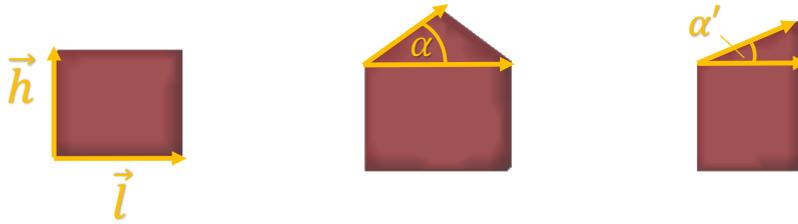


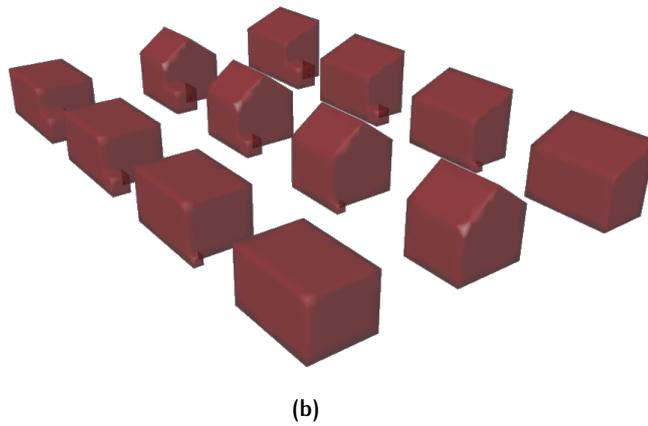
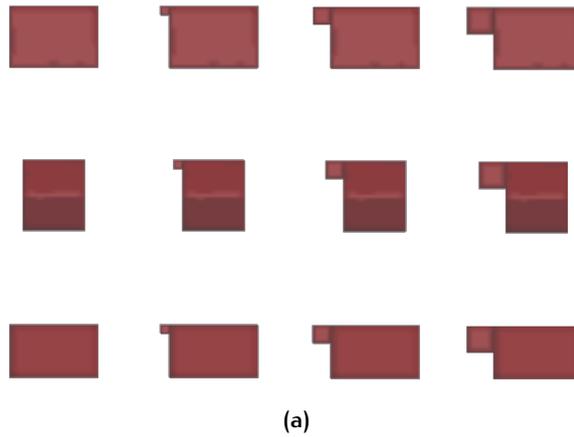
Figure 3.9: Roof types of synthetic building models

For the shed roof type we model two more surfaces. The first one, representing the roof of the building, is inclined by angle α and the second one inclined by 90° , to extend the side wall with height $\frac{|\vec{l}|}{\tan \alpha}$. The span length of the roof is determined by $\frac{|\vec{l}|}{\cos \alpha}$ and the width by the building width $|\vec{w}|$. For the gabled roof type we also model one surface inclined by α with span length $\frac{|\vec{l}|}{2 \cdot \cos \alpha}$ and mirror this surface for the second half of the roof. Again the width of the roof is determined by the building width $|\vec{w}|$. The characteristics of the different types are summarised in table 3.1. We now vary building length, width, height and roof angle to get a dataset with around 300 buildings per class with different sizes.

Table 3.1: Roof types of synthetic building models (without extension)

Roof type	Flat	Shed	Gabled
Number of roof segments	1	1	2
Angle of roof segments	$\alpha = 0^\circ$	$\alpha > 0^\circ$	$\alpha > 0^\circ$
Footprint area	$ \vec{l} \times \vec{w} $		
Footprint perimeter	$2 \vec{l} + 2 \vec{w} $		
Gutter height	$ \vec{h} $		

For three more datasets we add extensions to the building models. This is simply done by modeling an additional, smaller building with a flat roof adjacent to the existing building. We control the size of the extension with a varying parameter $s = \{1, 2, 3\}$, being the length, width and height of the extension. Figure 3.10a and 3.10b show the resulting models.



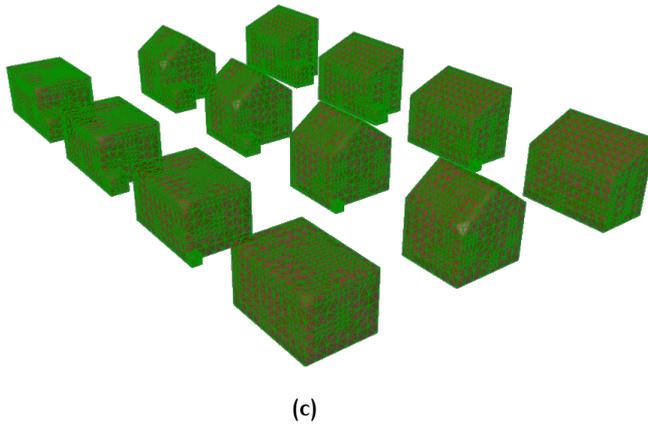
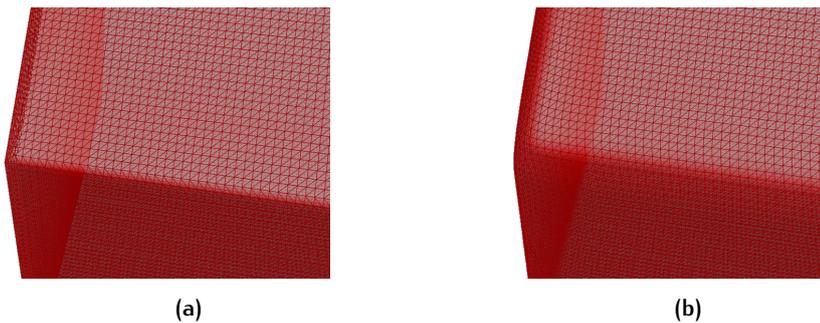


Figure 3.10: Synthetic building models with their boundary represented with surfaces (a,b) and meshes (c)

3.2.3 Meshing

In the last step we convert the boundary representations from (poly-)surfaces to meshes, to allow the extraction of Shape DNA from the synthetic building models section 3.3.1.1. In contrast to the mesh generation of the real building models () in this case we have full control over the density of the mesh, as it is based on the constructed surfaces. Thus, in theory we can make an arbitrary dense mesh. However, creating a mesh with edge length of 0.1cm for a building with $|w| = |l| = |h| = 5$, already leads to more than 30000 vertices. Even with a fast implementation, extracting the Shape DNA of this mesh takes around 5 seconds. For 250.000 building models it would take more than 800 days. Thus, such a density is not feasible. To get a further understanding of the mesh density and quality we apply 6 different meshings on one building model of each roof type. Figure 4.7 shows a close up of the different meshings. We will extract the Shape DNA of each of these meshes and compare the results.



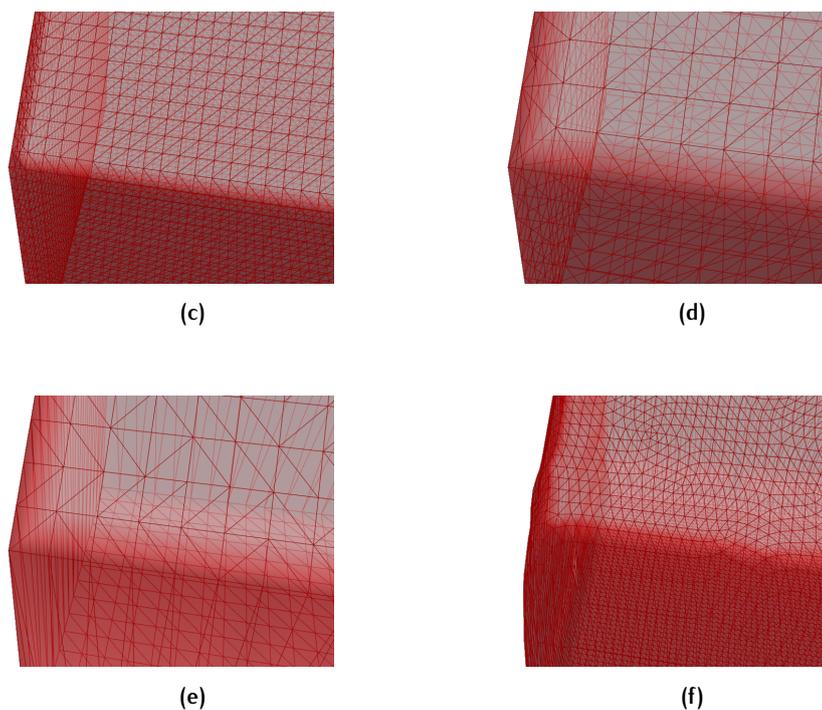


Figure 3.11: Synthetic building models with different meshing applied: (a) mesh with 0.1cm edge length - 34000 vertices; (b) same as (a), but mesh smoothing applied; (c) mesh with 0.2cm edge length - 8000 vertices; (d) mesh with 0.5cm edge length - 1300 vertices; (e) same as (d) but no vertices on the walls of the building to resemble meshing of real building models - 540 vertices; (f) same as (d) but mesh smoothing applied to achieve better mesh quality on the walls - 8000 vertices.

3.3 FEATURE BASED BUILDING REPRESENTATION

3.3.1 Shape analysis and feature extraction

In this section we will report on the feature extraction with the goal to represent a building with a feature vector that can be used for the supervised learning approach (section 3.4.1). We mainly extract shape features from geometric representations of the building and the year of construction from ancillary cadastral data. All features are summarised in table 3.2.

Table 3.2: Features representing a building

Abbreviation	Description	Input representation
fparea	Area of the footprint	Footprint polygon
perimeter	Perimeter of the footprint	Footprint polygon
gutter_height	Gutter height of the building	Building point cloud
rcount	Number of roof segments, based on region growing algorithm	Building point cloud
rangle	Average angle of roof segments, based on region growing algorithm	Building point cloud
sarea	Surface area of the building	Building mesh
ev	Shape DNA, based on eigenvalues of the Laplace-Beltrami operator	Building mesh
yoc	Year of construction of the building	-

3.3.1.1 Extract Shape DNA

According to the theory and the equations given at the end of section 2.4.2 we extract the first 50 eigenvalues of the **LBO** of the building mesh using the **FEM** discretisation [48]. To this end, we developed an algorithm that takes the building meshes as input and returns the eigenvalues of the discrete **LBO** of the meshes (see algorithm 3.5). We first build the stiffness matrix A and mass matrix B according to equation 2.8 and 2.9. Both matrices are symmetric, and of size $|V| \times |V|$, where $|V|$ corresponds to the number of vertices of the mesh. The building meshes we generate in section 3.2.3 typically have around 500 to 2500 vertices, depending on the size of the building. Using the matrices we build the generalised symmetric eigenvalue problem given in equation 2.10. Due to the size of the matrices it is computationally expensive to solve this problem with a full eigendecomposition. In this thesis, we use the implicitly restarted Arnoldi method to iteratively solve the generalised eigenvalue problem. Because iterative methods work best on large eigenvalues, the problem is first inverted into $B\mathbf{f} = \frac{1}{\lambda}A\mathbf{f}$, which is called the shift-invert method [47]. A shift-invert Arnoldi method is implemented in the linear algebra package ARPACK [34]. Solving equation 2.10

this way leads to a n -dimensional array describing the global shape of each building.

Algorithm 3.5: Extract Shape DNA

```

1 function ShapeDNA ( $M, n$ ) :
    Input:
        a 2-manifold triangular mesh  $M$  with vertices  $V$ ,
        edges  $E$  and faces  $F$ 
        an integer  $n < |V|$  as the desired number of eigenvalues
    Output:
         $\lambda$ s as an array of  $n$  eigenvalues  $\lambda$ 
2  $A \leftarrow$  initialise stiffness matrix  $A$  with  $|V|$  rows and  $|V|$  columns;
3  $B \leftarrow$  initialise mass matrix  $B$  with  $|V|$  rows and  $|V|$  columns;
4 for  $E$  in  $M$  do
5      $(i, j) \leftarrow$  get indices of  $V$  incident to  $E$ ;
6      $cot = 0$ ;
7     for  $angle \leftarrow$  opposite of  $E$  do
8          $cot+ = cotangent(angle)$ ;
9      $A(i, j) = A(j, i) = 0.5 \cdot cot$ ;
10     $ar = 0$ ;
11    for  $F$  incident to  $E$  do
12         $ar+ = area(F)$ 
13     $B(i, j) = B(j, i) = ar/12$ ;
14  $A = -A + rowsum(A)$ ;
15  $B = B + rowsum(B)$ ;
16 for  $iter \leftarrow 0$  to  $n$  do
17     iteratively solve  $Af = -\lambda Bf$ ;
18      $\lambda$ s.append( $\lambda$ );
19 return  $\lambda$ s

```

3.3.1.2 Extract other geometric features

Next to Shape DNA we extract more geometric features, with the goal to gain more insight into the suitability of different descriptors for SBST classification.

Area of the footprint polygon

The area of a polygon can simply be calculated with Gauss' area formula. In our case this area is already explicitly included as an attribute of every building footprint polygon in the cadastral dataset. This attribute can often give a good impression of the size of a building, or at least of its 2D extent.

Perimeter of the footprint polygon

We also extract the perimeter of the footprint polygon. As several previous methods suggest [8, 36, 22] (see also section 2.5), this may be a representa-

tive features for [SBST](#) classification. Combined with the footprint area, the perimeter can help to describe the shape of the footprint polygons.

Gutter height

To also incorporate [3D](#) information when comparing Shape DNA to the traditional geometric features we extract the height of the buildings from the point cloud. Different possibilities to extract the height of a building roof point cloud can be seen in [figure 3.12](#).

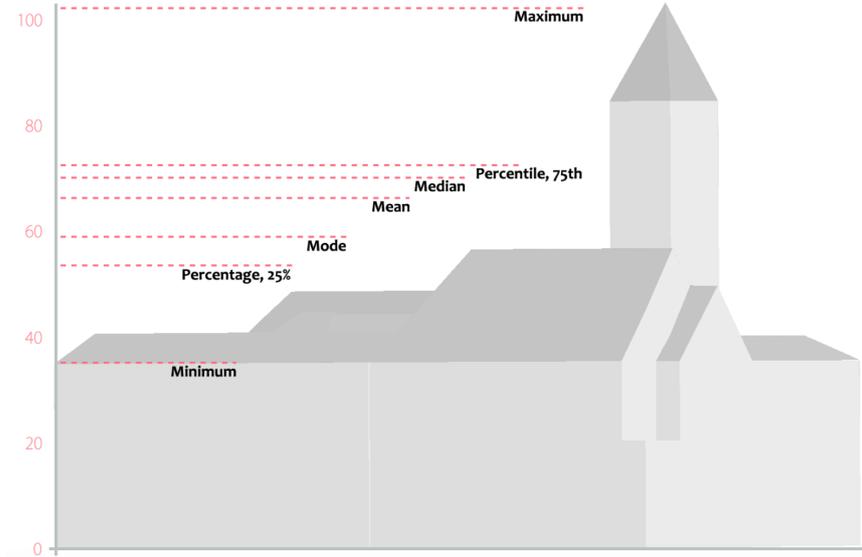


Figure 3.12: Different approaches to extract building height from point cloud [\[33\]](#).

In this thesis we chose to take the mode of the Z-coordinates (see [section 2.3.2](#)) of all the points of the cleaned building point cloud (see [section 3.1.1](#)). We mainly use this approach because it is fairly simple to implement. We call this feature gutter height, although, it may not represent the actual gutter height of a building. The actual gutter height of a building is not trivial to extract, especially if a building has several roof segments with different heights. To extract a more precise gutter height, one could use the buffered building polygon (see [section 3.1.1](#)) and only use the height information (such as the maximum Z-coordinates) of the points of which the [2D](#) projection lies inside the buffered footprint polygon. [Figure 3.13](#) shows an example for such an approach.

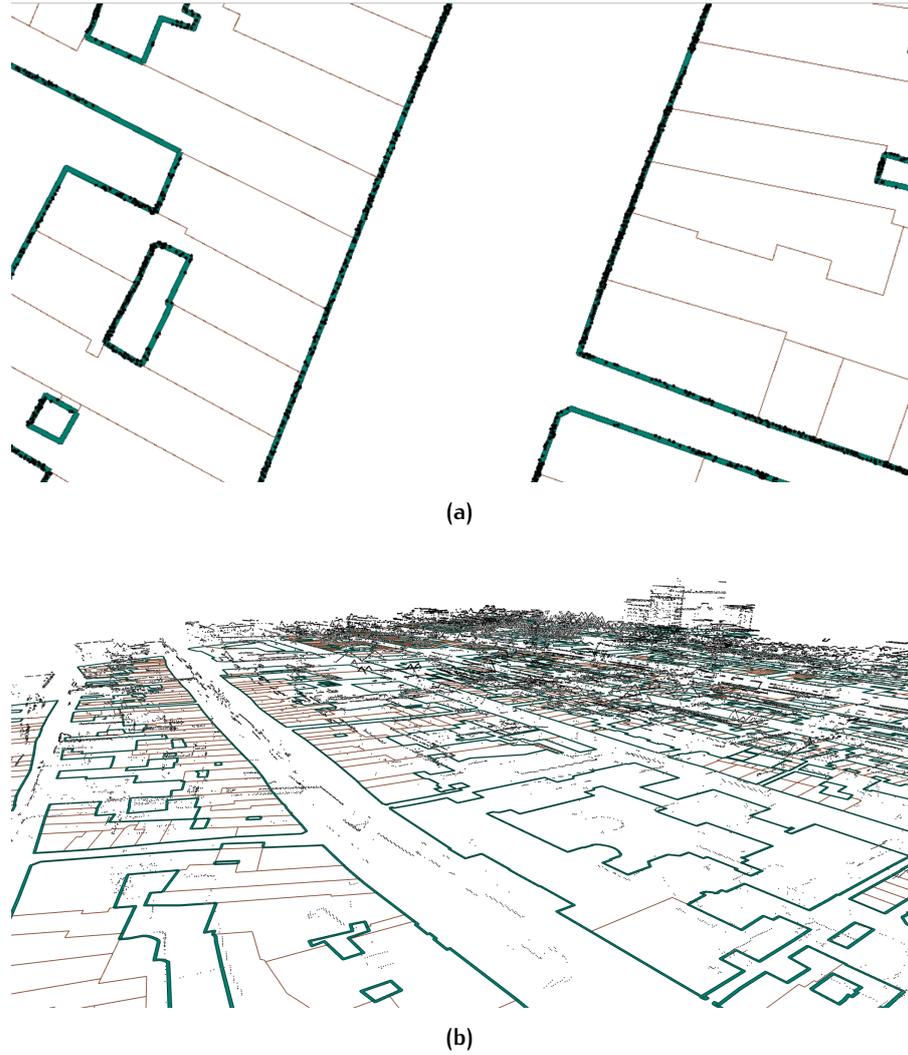


Figure 3.13: Extract gutter height of buildings: (a) 2D projection of points intersecting with buffered footprint polygon (b) Multiple buildings point cloud intersecting with buffered footprint polygon

Roof segment count and average roof angle

First, as a byproduct of the region growing algorithm (section 3.1.1.1), we can extract the number of planar roof segments, as the roof segment count (**rcount**). Furthermore, by fitting a plane to the segments, we can also get the zenith angle α of these segments, using the normal vector \vec{n} of the fitted plane (see figure 3.14). The zenith angle is defined as

$$\cos \alpha = \frac{\vec{n} \cdot \vec{z}}{|\vec{n}| \cdot |\vec{z}|}, \quad (3.1)$$

with $\vec{z} = [0 \ 0 \ 1]^T$. To use this measurement as a feature, we average the zenith angles of every roof segment, as roof angle (**rangle**). However, this can result in situation in which a building with two roof segments with zenith angles of 5° and 25° has the same feature value as a building with one roof segment of 15° . Especially in situation where one of the roof segments is much smaller than another one this may be unwanted. A way

to circumvent this problem would be to weight the zenith angles with the number of points a roof segment consists of.

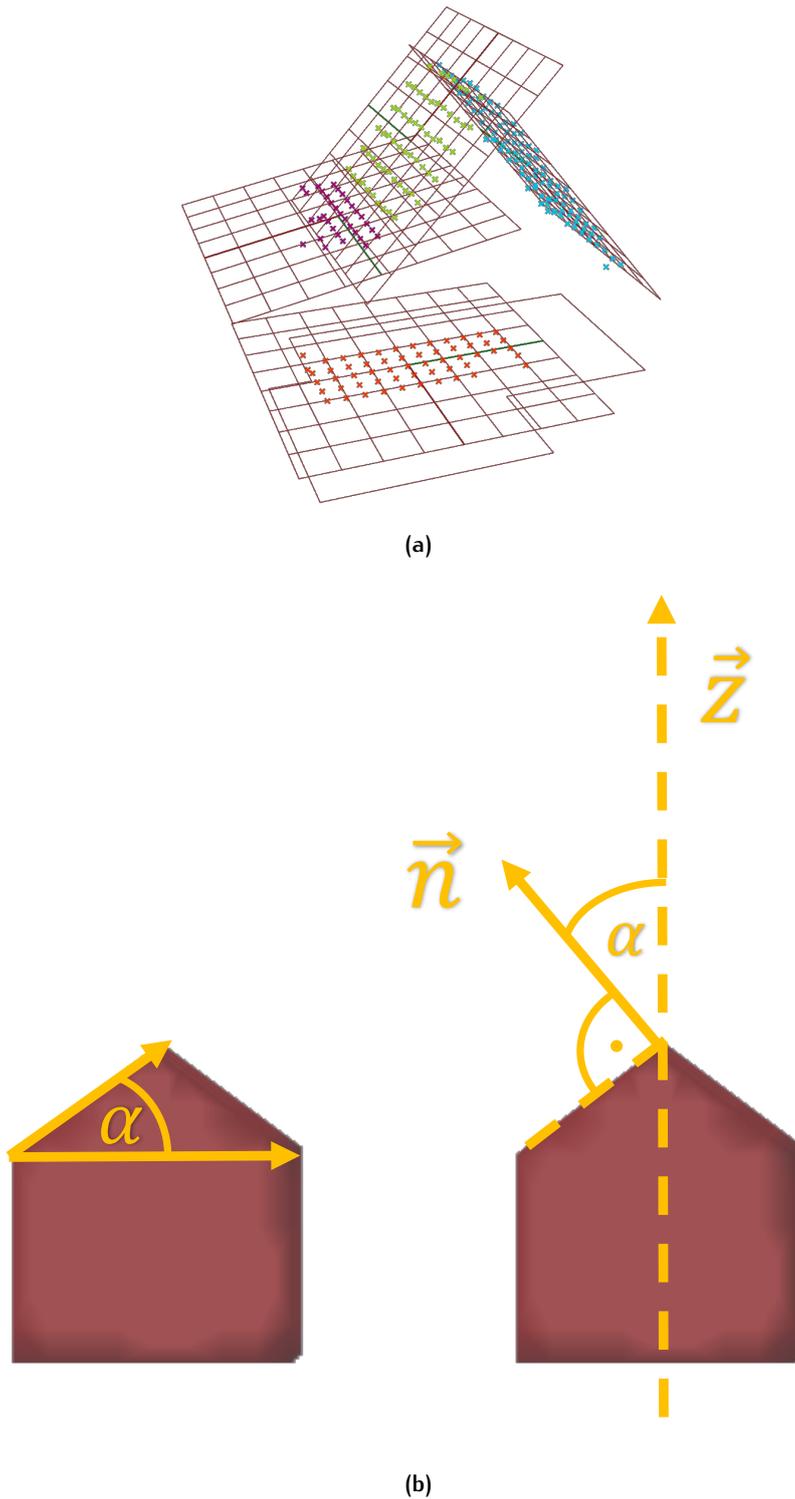


Figure 3.14: Roof angle extraction (notice that α represents the same angle as the inclination angle defined on the synthetic building models)

Surface area

For this feature we simply extract the surface area of the building mesh created in section 3.2.3. This is simply the sum of the area of each mesh face, which is a good approximation of the real surface area of the building. The feature thereby delivers a good description of the global size of the building.

3.3.1.3 *Extract non-geometric features*

The only non-geometric feature we use is the building year of construction. In our case the year of construction is included as an attribute of every building in the cadastral dataset. Further attributes included in this dataset are not used because they may introduce a bias in the SBST classification of our dataset (see section 3.1.2).

3.3.2 Feature scaling

All the features extracted in section 3.3.1 are standardised (section 2.2.2). This means we also standardise the Shape DNA as the spread of the higher order eigenvalues is often bigger than the spread of the lower order ones. However, the higher order eigenvalues are not regarded to be more important (section 4.4.1.3). It is important to include the feature scaling process inside the inner loop of a CV [13]. This means we make sure that the feature scaling is done on each train and test set individually. Otherwise information about the test set would be included in the tuning of the hyperparameters and evaluation process which is unwanted for a good generalisation of the model.

3.4 AUTOMATIC CLASSIFICATION

3.4.1 Supervised learning

For the SBST classification we use a linear (SVM) and several non-linear (k-NN, kernel SVM, RF, ANN) classifiers (section 2.2.3). The hyperparameters of the classifiers are tuned with a 5-fold cross validated grid search evaluating the best classification accuracy on the respective test sets.

Furthermore we use different feature (sub-)sets to gain a better insight into the importance of feature types, such as local or global shape descriptors applied on 2D or 3D representations, or the year of construction.

3.4.2 Evaluation

We evaluate different classification models based on their classification accuracy with the best performing hyperparameters. The classification accuracy is the mean value of the performance on each of the 5 test sets in the cross validation. We also generate a mean confusion matrix within this process

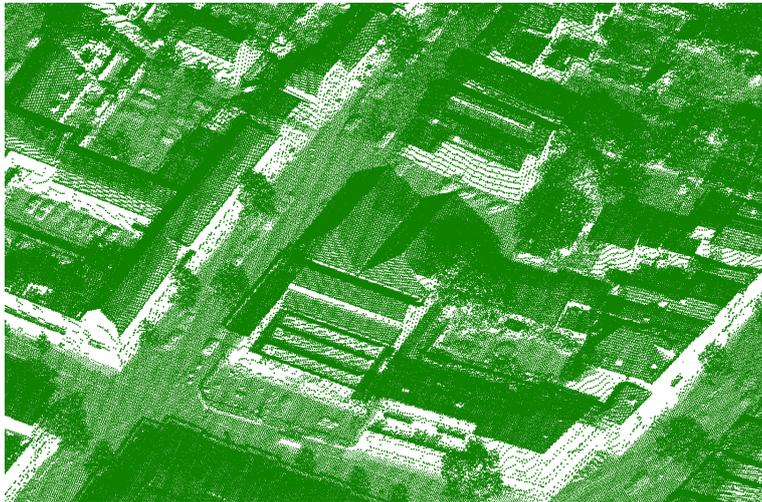
to get an insight into the prediction of the separate classes. When comparing the performance of different models (classifiers and feature subsets) we make sure that it is based on the same input samples.

4 | IMPLEMENTATION AND EXPERIMENTS

4.1 INPUT DATA

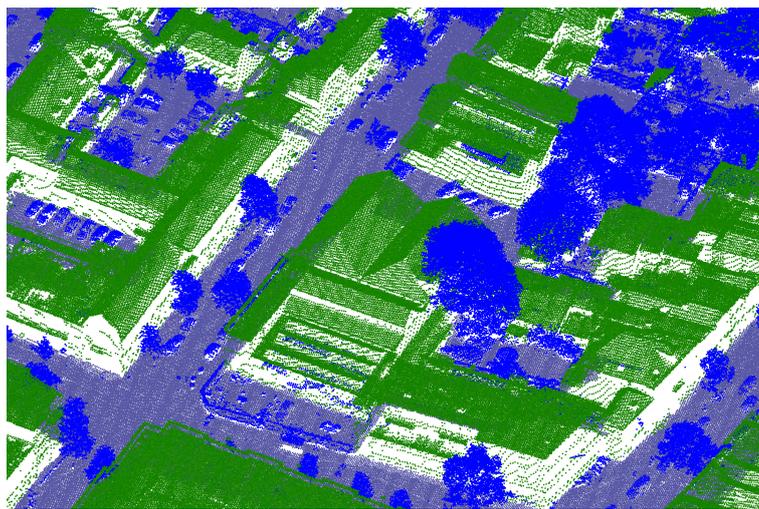
4.1.1 Remote sensing data

We make use of the Actueel Hoogtebestand Nederland (AHN)¹, a digital surface model in the form of a point cloud. The point cloud is obtained by ALS, and in total contains more than 600 billion elevation points covering the whole Netherlands. The AHN dataset is available for download free of charge at the Dutch national geoportal Publieke Dienstverlening Op de Kaart (PDOK) in a *.las* file format (see [2] for a documentation of the dataset and downloading instructions). The point cloud is preprocessed, such that the points are the centre points of a regular grid with $0.5m \times 0.5m$ grid cells. Different versions (AHN1, AHN2 and AHN3) of this dataset exist for different regions of the country (figure 4.1). At the time of writing this thesis only AHN2, measured from 2011 through 2012, was available for the Groningen region. In contrast to AHN3, the points of the AHN2 point cloud are not classified according to classes, such as ground level, buildings or vegetation (figure 4.1). This means the extraction of buildings from the point cloud requires an additional processing step and the building roofs may be occluded by trees (section 3.1.1).

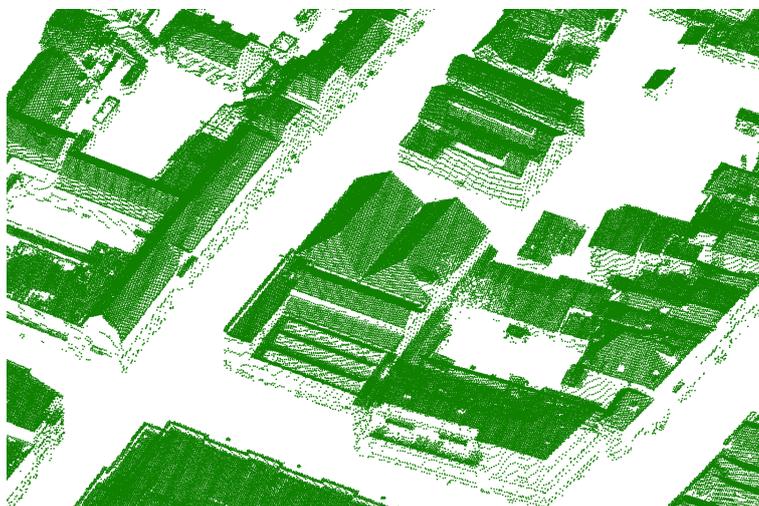


(a)

¹ <http://www.ahn.nl/index.html>



(b)



(c)

Figure 4.1: Different versions of [AHN](#): (a) [AHN2](#); (b) [AHN3](#) with classified ground (purple), vegetation (blue) and buildings (green); (c) only buildings in [AHN3](#).

4.1.2 Cadastral data

The cadastral dataset used in this project stems from the Basisregistraties Adressen en Gebouwen ([BAG](#)), which is also available on the Dutch geoportaal [PDOK](#). The dataset includes two features that are used in this project, namely the year of construction and the footprint polygon of the buildings. In general, the footprint polygon represents the outline of the building touching the ground. However, if the roof- or substructure (e.g. basement) of the building deviates more than $1m$ from this outline, the respective outline of the roof- or substructure is represented [60].

4.1.3 Building dataset

The building dataset comprises information of about 20.000 buildings of the Groningen building stock. In a classification process (section 3.1.2) this information is used to predict the **SBST** class membership of each building according to the **GEM** taxonomy. The distinct combination of the **GEM** attributes results in more than 60 different **SBST** classes that are present in the dataset. For the experiments conducted in this thesis we use 800 sample buildings equally distributed over the eight most common **SBST** classes in the dataset. This sample is summarised in table 4.1. Additionally, exemplary buildings of each class are shown in figure 4.2.

Table 4.1: Groningen training samples

	Material of lateral load resisting system in direction X	Lateral load resisting system in direction X	Material of lateral load resisting system in direction Y	Lateral load resisting system in direction Y	Floor type	Exterior Wall – Presence of outer leaf
MU LWAL MU LWAL FW EWN	Masonry unreinf.	Wall	Masonry unreinf.	Wall	Wood	No
MU LWAL MU LN FM EW	Masonry unreinf.	Wall	Masonry unreinf.	No system	Masonry	Yes
MU LWAL MU LN FW EW	Masonry unreinf.	Wall	Masonry unreinf.	No system	Wood	Yes
MU LWAL MU LN FC EW	Masonry unreinf.	Wall	Masonry unreinf.	No system	Concrete	Yes
MU LH MU LH FW EWN	Masonry unreinf.	Hybrid	Masonry unreinf.	Hybrid	Wood	No
CR LWAL CR LN FC EW	Concrete	Wall	Concrete	No system	Concrete	Yes
CR LWAL CR LN FC EWN	Concrete	Wall	Concrete	No system	Concrete	No
CR LWAL CR LN FO EW	Concrete	Wall	Concrete	No system	Other	Yes

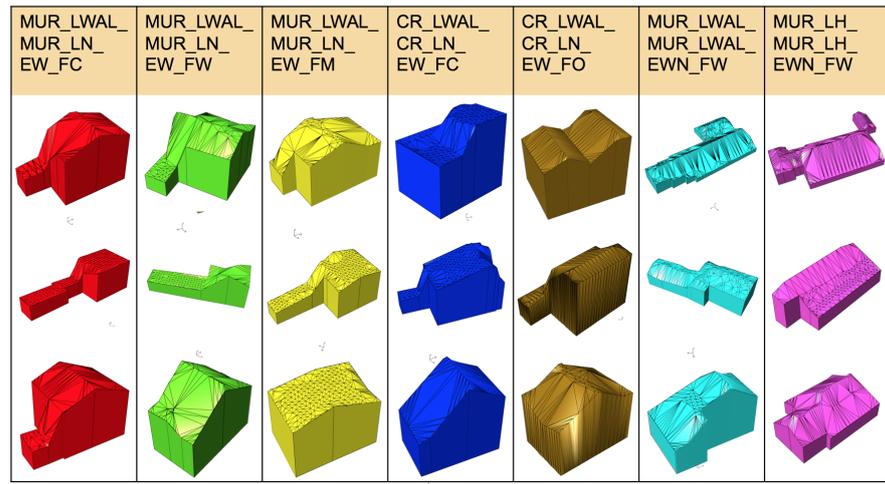


Figure 4.2: Examples of the building stock with their GEM label

4.1.4 Synthetic building models

For the shape analysis we mainly use the synthetic building models created in this thesis (section 3.2). In total we modeled 12 different classes (Table 4.2) with around 300 buildings in each class.

Table 4.2: Synthetic training samples

Abbreviation	Description
fl_wo	Flat roof type without extension
fl_e1	Flat roof type with extension of $s = 1$
fl_e2	Flat roof type with extension of $s = 2$
fl_e3	Flat roof type with extension of $s = 3$
ga_wo	Gabled roof type without extension
ga_e1	Gabled roof type with extension of $s = 1$
ga_e2	Gabled roof type with extension of $s = 2$
ga_e3	Gabled roof type with extension of $s = 3$
sh_wo	Shed roof type without extension
sh_e1	Shed roof type with extension of $s = 1$
sh_e2	Shed roof type with extension of $s = 2$
sh_e3	Shed roof type with extension of $s = 3$

The buildings vary in their length, width and height from $3m$ to $10m$, in their roof angle from around 30° to 70° , and in their extension from $1m$ to $3m$ (width, length and height). Figure 4.3 shows an overview of the variation for one of the classes. For most of the experiments we use a random subset of 100 buildings of a class.

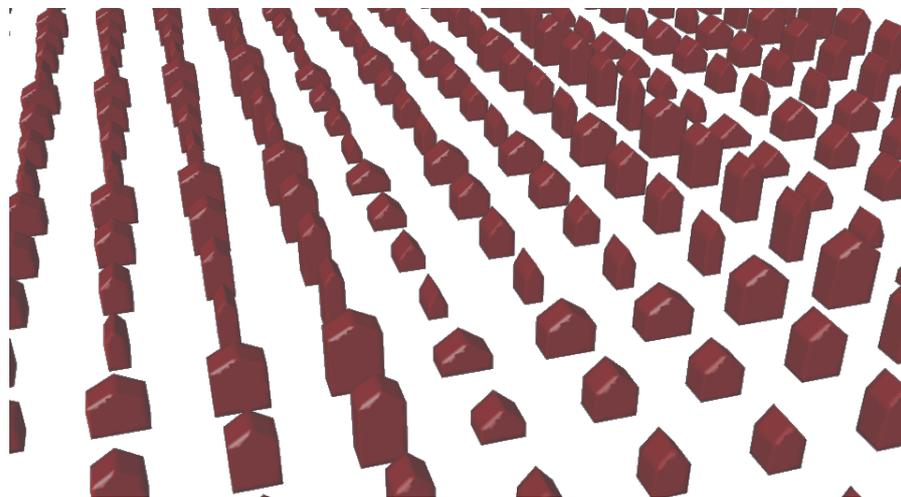


Figure 4.3: Synthetic building dataset

4.2 SOFTWARE

4.2.1 FME

FME² is a GIS software that allows the creation of custom workflows. Components with common GIS functions can be used in a visual programming approach. FME is useful for handling large datasets such as the ALS point clouds and applying simple spatial operations. In this thesis it is used to implement the point-in-polygon test for the AHN point cloud and the building footprints.

4.2.2 Postgres DB

Postgres DB³ is a database management system based on the the SQL standard. It allows fast data handling for large datasets. All the data that is used in this thesis is stored in several tables of a Postgres DB.

4.2.3 Rhinoceros and Grasshopper

Rhinoceros⁴ is a 3D computer graphics and computer-aided design application software. Grasshopper is a visual scripting language add-on for

² <https://www.safe.com/>

³ <https://www.postgresql.org/>

⁴ <https://www.rhino3d.com/>

Rhinoceros. The add-on also allows to integrate the programming language C# as individual scripting components. In this thesis Rhinoceros and Grasshopper are used for a large part of the geometric processing.

4.2.4 C#

C# is a general-purpose, object-oriented programming language. In this thesis it is used for data processing within the Grasshopper add-on of Rhinoceros.

4.2.5 Python and libraries

Python is an interpreted high-level programming language for general-purpose programming. Besides Python's large standard library, additional libraries are used for data handling (analysing, storing, exporting, visualisation) and for the machine learning algorithms.

SciPy

SciPy is a library for scientific computing. We use SciPy for solving the generalised eigenvalues problem in algorithm 3.5.

Pandas

Pandas is an open source library providing high-performance, easy-to-use data structures and data analysis tools. In this thesis it is mainly used for storing and exporting results of the classification process.

Psycopg

Psycopg is the most popular PostgreSQL adapter for the Python programming language. At its core it fully implements the Python DB API 2.0 specifications. Several extensions allow access to many of the features offered by PostgreSQL.

Imbalance-learn

The imbalanced-learn package is a Python package implementing re-sampling techniques for datasets showing strong between-class imbalance. We use imbalanced-learn to undersample the majority classes in our training datasets for the machine learning algorithms.

Scikit-learn

Scikit-learn [43] is the machine learning library used in this thesis. It features various classification algorithms including SVM, RF or k-NN and is designed to interoperate with SciPy.

Matplotlib

Matplotlib is a Python 2D plotting library which can produce high quality figures. In this thesis it is used for visualising data, such as results of the machine learning processes.

Seaborn

Seaborn extends the functionality of Matplotlib with a focus on statistical data visualisation.

4.2.6 shapeDNA-tria

'shapeDNA-tria'⁵ is a software developed by Reuter et al. [48] implementing different discretisations of the LBO on polygon meshes and their spectral decompositions. The software, thus, allows to extract Shape DNA from a polygon mesh. We use this software to compare results and performance to our own implementation for extracting Shape DNA.

4.3 IMPLEMENTATION

4.3.1 Shape analysis

In a first step the BAG and AHN data is loaded into the software FME. The *PointAreaOverlayer* transformer is used to implement the point-in-polygon-test (figure 3.2). The points that lie inside a footprint polygon are exported to a database table, where each point is assigned to a specific building determined by a building ID.

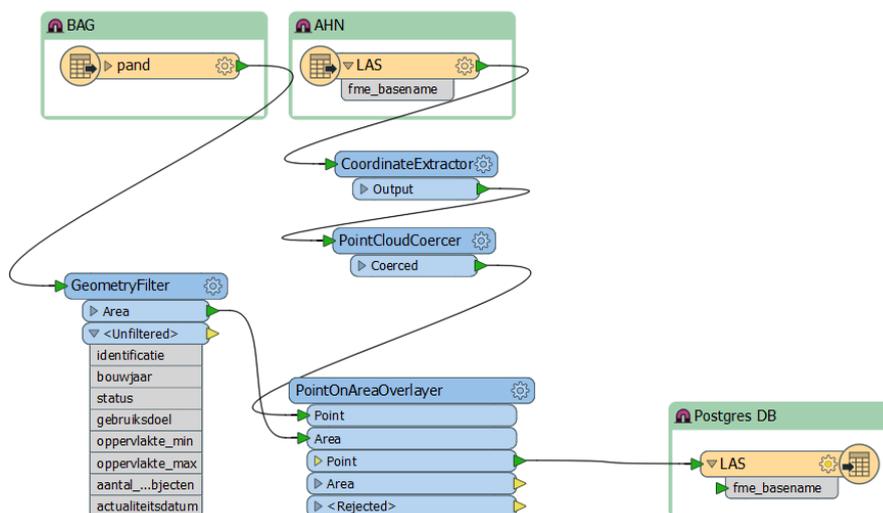


Figure 4.4: FME workbench

Subsequently the points and polygons are loaded into a Grasshopper script where the mesh generation is implemented. As a first step, the region growing algorithm is applied on the point cloud. The region growing algorithm is implemented with C# scripting components as described in algorithm 3.4.

⁵ <http://reuter.mit.edu/software/shapedna>

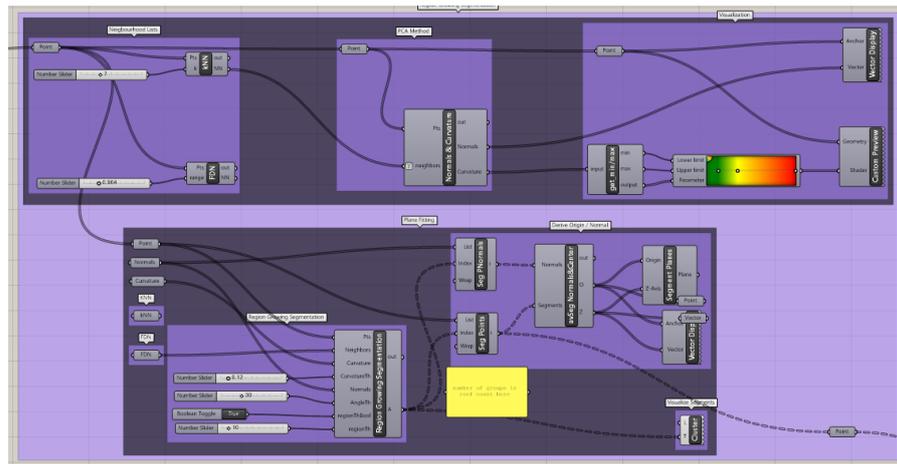


Figure 4.5: Implementation of region growing algorithm in Grasshopper

Figure 4.5 shows an overview of the implementation. The rest of the meshing is implemented with a combination of standard Grasshopper components. Geometric features, such as the footprint area and perimeter, roof count and angle, surface area etc., are exported to a database table of a Postgres database (table 4.3). The mesh of each building is saved as a Wavefront *.obj* file.

Table 4.3: Database table with shape features of a building

Building-ID	fparea	perim.	sarea	rcount	rangle	gutter
XXX						

Subsequently, a Python script was created that reads the *.obj* file and implements algorithm 3.5. This part of the implementation can be found on Github⁶. The eigenvalues are saved in another database table (table 4.4).

Table 4.4: Database table with Shape DNA of a building

Building-ID	ev1	ev2	ev3	ev4	ev5	...
XXX						

4.3.2 Supervised learning

In a first step the label training set is prepared. A database table with separate columns for each GEM attribute is created. A row of this table, thus,

⁶ <https://github.com/muha321/ShapeDNA>

contains one building identified by a building ID and its corresponding GEM attributes (table 4.5). Subsequently, Table 4.3, 4.4 and 4.5 are joined by the building ID.

Table 4.5: Database table with GEM attributes of a building

Building-ID	LLRS_x	M_x	LLRS_y	M_y	Floor	EW
XXX						

A python script implements an *Importer* class that allows to import an arbitrary combination of GEM attributes and features per building. A possible usage of the class can be seen below. The synthetically generated dataset can be treated in the same manner. Possible label types for this dataset are *roof* and *extension*.

```
# initialise importer class instance
im = Importer()

# define labels as lateral load resisting system in x and y
  direction and ancillary material
la = ['llrs_x', 'm_x', 'llrs_y', 'm_y']

# define type that should not be imported (optional)
ex = [('llrs_x', 'LWAL')]

# define minimum amount of samples per class (optional)
lim = 100

# create the database table
im.create(labels = la, exclude = ex, limit = lim)

# load data
im.load(building_id = True, fparea = True, yoc = True)

# return data as numpy array
bid = im.building_id    # returns list of building ID's used
data = im.features     # returns label
label = im.label       # returns feature
```

The data and label arrays can then be used in any common Scikit-learn function [43]. For the usage of these functions the reader is referred to the Scikit-learn documentation⁷.

4.4 EXPERIMENTS

In the following section we conduct multiple experiments to gain insight into the usability of the methodology developed in this thesis. Table 4.6

⁷ <http://scikit-learn.org/stable/documentation.html>

shows an overview of the conducted experiments. In section 4.4.1 we investigate into the use of different shape descriptors for classifying the synthetic building dataset. The focus of these experiments lies on the usability of Shape DNA as a global descriptor of a building shape. In section 4.4.2 we apply the results of the preceding experiments on the Groningen building dataset and predict the SBST with different feature combinations and training set sizes.

Table 4.6: Experiments conducted in this thesis

Section	Experiment	Question
Shape analysis	1.1	Does our algorithm to extract Shape DNA from a mesh return valid results?
	1.2	How do density and quality of a building mesh influence Shape DNA?
	1.3	How does Shape DNA perform at predicting the roof type of a building?
	1.4	How does Shape DNA perform at predicting the roof type of a building with an extension?
	1.5	How do other shape descriptors perform at predicting the roof type of a building with an extension?
SBST prediction	2.1	How do the features extracted in section 3.3.1 perform at predicting the LLRS?
	2.2	How do the features extracted in section 3.3.1 perform at predicting the SBST?
	2.3	How does the number of training samples influence SBST prediction?

4.4.1 Shape analysis

4.4.1.1 *Does our algorithm to extract Shape DNA from a mesh return valid results?*

Shape DNA is the beginning sequence of the eigenvalues of the LBO defined on a manifold mesh and thus the solution to equation 2.10. Equation 2.10 is the discretised version of the so-called Helmholtz equation, given in equation 2.7. Analytic solutions of equation 2.7 are only known for a limited number of shapes, such as the sphere, the cylinder or the solid ball. The eigenvalues for the unit 2-sphere for example are

$$\lambda_i = i(i + 1), \quad i \in \mathbb{N}_0, \quad (4.1)$$

with multiplicity $2i + 1$ [49].

By comparing the solution of equation 4.1 for some i , to the solution of algorithm 3.5, with a meshed sphere as input, we can validate if algorithm 3.5 returns the correct results. To this end, we calculate the analytic solution

for a unit sphere with a simple Matlab script. Furthermore, we constructed the meshed sphere with the constraint of a maximum edge length of $0.1m$ which lead to an approximation of the sphere with around 2000 vertices (see figure 4.6a). The sphere is meshed and the first 50 eigenvalues are extracted with the implementation of algorithm 3.5 and with the software 'shapeDNA-tria' (section 4.2.6). A plot of the results can be seen in figure 4.6b. The different implementations of Shape DNA are not shown, as they led to the same result.

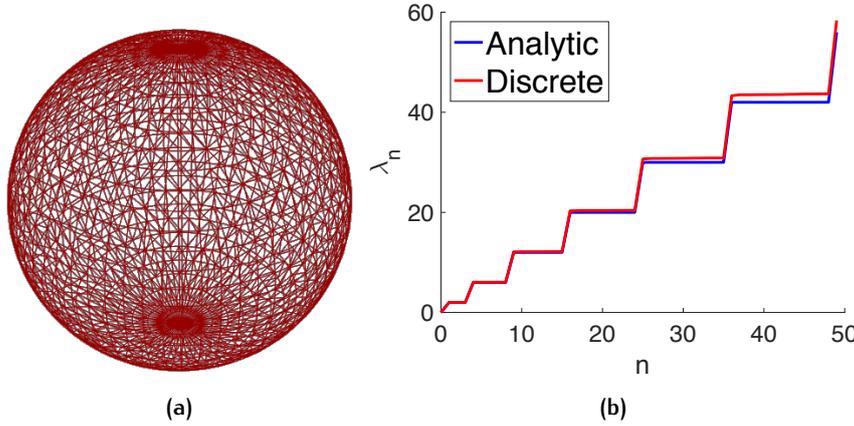


Figure 4.6: (a) polygon mesh of a sphere; (b) first 50 eigenvalues of the LBO eigenvalues of a sphere: analytic solution calculated with equation 4.1 and discrete solution calculated with algorithm 3.5

We may now define the two vectors constructed from the eigenvalues of the analytic and discrete solution as feature vectors of the corresponding spheres. The euclidean distance between the two vectors can be calculated with

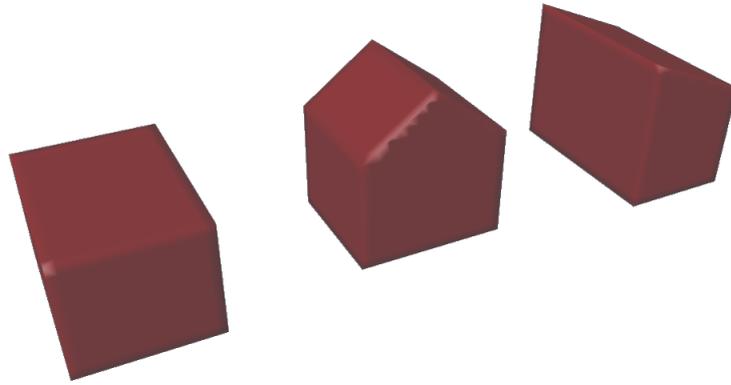
$$d_{sphere} = \sqrt{\sum_{n=1}^{50} (\lambda_n^{Analytic} - \lambda_n^{Discrete})^2} = 3.016 \quad (4.2)$$

Since both feature vectors represent the shape of a unit-sphere we desire d_{sphere} to be 0. However, the meshed sphere is only a discrete approximation of the continuous unit sphere. If we increase the number of vertices N of the meshed sphere we can create a better approximation of the real unit-sphere, and thus d_{sphere} will decrease (section 4.4.1.3). However, based on figure 4.6b, and the fact that our implementation returns the same result as the 'shapeDNA-tria' software we conclude that algorithm 3.5 works as expected.

4.4.1.2 How do density and quality of a building mesh influence Shape DNA?

For shapes such as buildings, no analytic solution is known for equation 2.7. Therefore, we can only rely on the result of our previous experiment. However, a building mesh with $0.1m$ edge length mesh has around 30.000 vertices. Extracting the Shape DNA of a whole building stock with such models is not feasible (section 3.2.3). To get an understanding, how a different meshing changes the precision of Shape DNA, we apply 6 different meshings on

a building model of each roof type. Figure 4.7 shows a close up of the different models. In this experiment we will extract the first 50 eigenvalues of the LBO of the meshes, as Shaphe DNA of each of theses meshes and compare the results. A smaller number of eigenvalues does not significantly change the results.



(a)

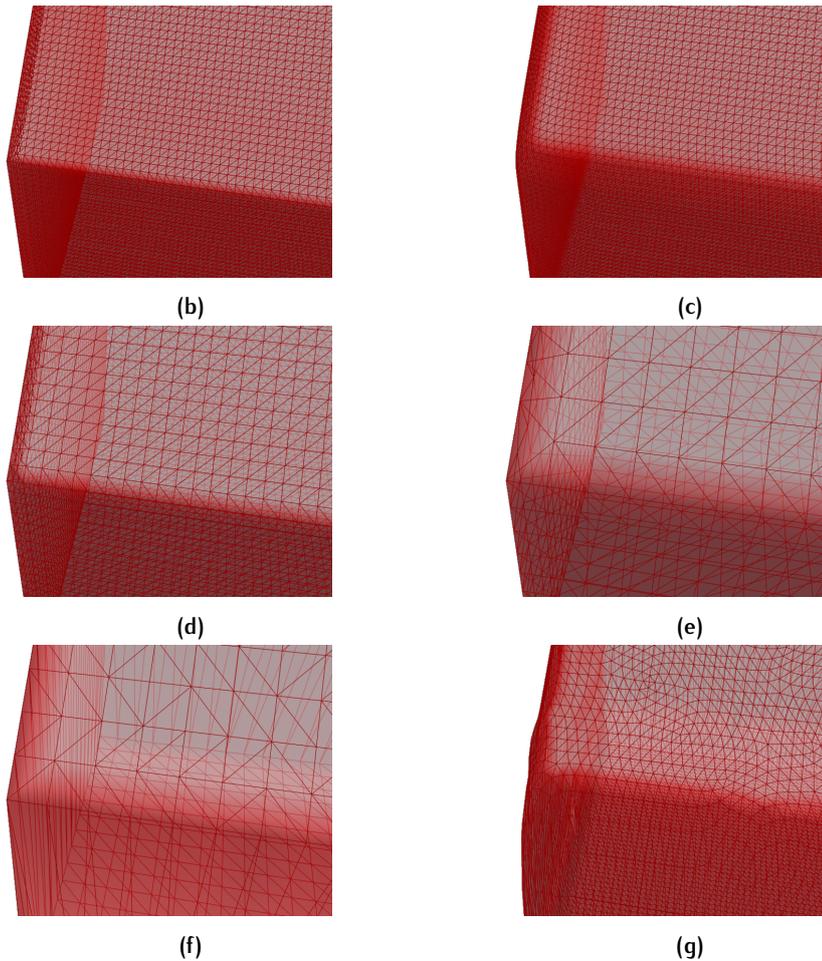
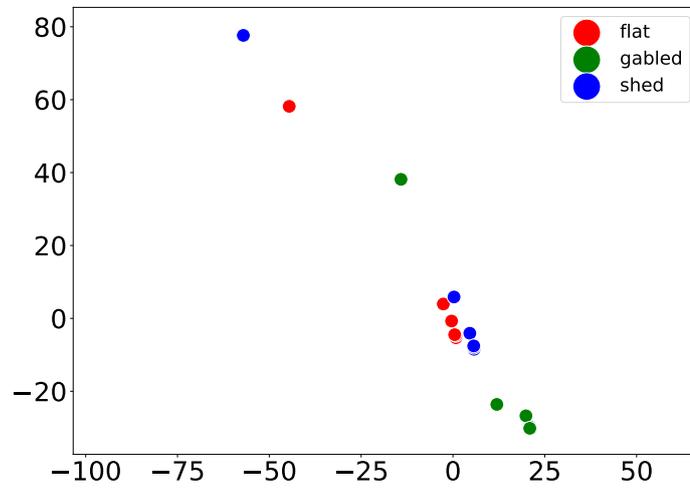


Figure 4.7: Synthetic building models with different meshing applied: (a) flat, gabled and shed roofed building model; (b) mesh with 0.1cm edge length - 34000 vertices; (c) same as (b), but mesh smoothing applied; (d) mesh with 0.2cm edge length - 8000 vertices; (e) mesh with 0.5cm edge length - 1300 vertices; (f) same as (e) but no vertices on the walls of the building to resemble meshing of real building models - 540 vertices; (g) same as (f) but mesh smoothing applied to achieve better mesh quality on the walls - 8000 vertices

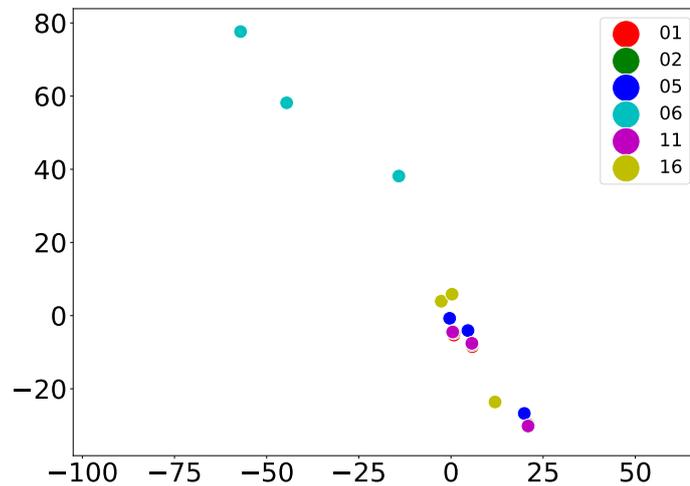
We use the extracted Shape DNA as feature vectors of the building models and plot the results with the help of MDS (section 2.2.2). Figure 4.8b shows the different meshings. The points in magenta represent the building models that were meshed with 0.1m edge length. A geometric smoothing was applied to “round” the corners and edges (roof-wall, wall-floor intersection) of the model. A topologic smoothing was applied to produce equally sized triangles on the mesh. These meshes are considered to represent the underlying manifold most accurately. The points in cyan represent building models that were meshed to resemble the meshing of the real building models (section 3.2.3). We can see that the Shape DNA extracted from these models is far from the red points and therefore considered to be imprecise. When applying a topologic mesh smoothing to the “magenta”-models, we can increase the precision of the Shape DNA greatly (“yellow”-models). In figure 4.8c we see that the meshes with 0.1 (with and without smoothing) and the meshes with 0.2m produce very similar results. The later models

consist of around 8000 vertices and are therefore still feasible to use for shape analysis of the whole building stock. Thus, we will continue using models with this meshing applied.

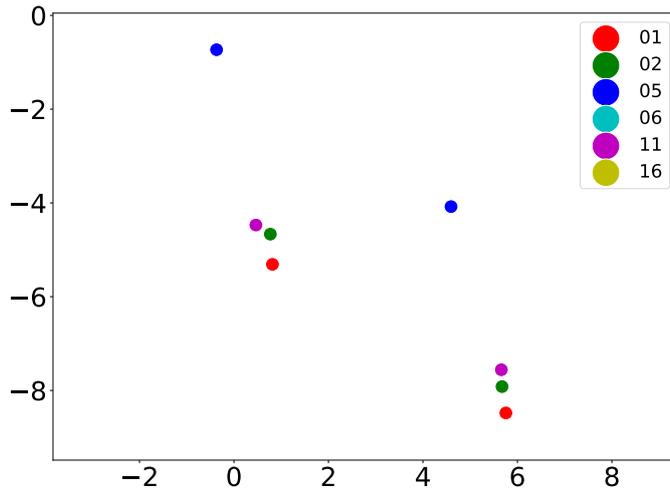


(a)

However, what we can already see by looking at figure 4.8a is that buildings with different roof types may have similar feature vectors, and are, thus, close in feature space. This is not desirable if we want to classify the buildings according to their roof type.



(b)



(c)

Figure 4.8: MDS of Shape DNA of synthetic building models with different meshing applied: (a) models coloured according to flat, gabled and shed roof type; (b,c) models coloured according to meshing: Red: $0.1cm$ edge length and 34000 vertices; Magenta: same as red, but mesh smoothing applied; Green: mesh with $0.2cm$ edge length and 8000 vertices; Blue: mesh with $0.5cm$ edge length and 1300 vertices; Magenta: same as blue but no vertices on the walls of the building to resemble meshing of real building models - 540 vertices; Yellow: same as magenta but smoothing applied to achieve better mesh quality on the walls - 8000 vertices.

4.4.1.3 How does Shape DNA perform at predicting the roof type of a building?

In this experiment we extract Shape DNA of synthetic building models with three different roof types and without an extension. We normalise the Shape DNA by multiplying it with the first eigenvalue (), however, using another normalisation method lead to the same results. We use 100 building models of each roof type for this experiment. We select the building models randomly from the pool of 300 buildings per roof type (section 4.1.4). The models are meshed with $0.2m$ edge length. We represent the buildings with their feature vector comprised of Shape DNA with 50 eigenvalues. This leads to the best classification result. However, changing this parameter in the range of 15 to 50 eigenvalues only has a marginal impact on the classification of around 1-2%. Figure 4.9 shows a MDS plot of the buildings in the Shape DNA feature space. In this scaled down space, the buildings do not seem to be separable. However, this is not necessarily an accurate representation of the data in the 50-dimensional "Shape DNA space".

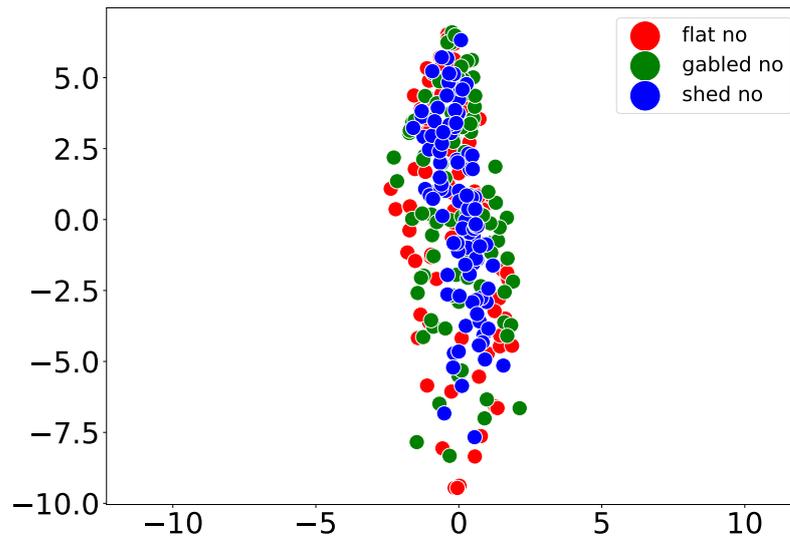


Figure 4.9: MDS of Shape DNA feature space with synthetic building models of different roof types.

Thus, we try to classify this dataset according to the roof type. Using a SVM with radial basis function (RBF) kernel leads to the best results. We determine the hyperparameters with a 5-fold CV. Figure 4.10 shows the average confusion matrix over the 5-folds of the best hyperparameters.

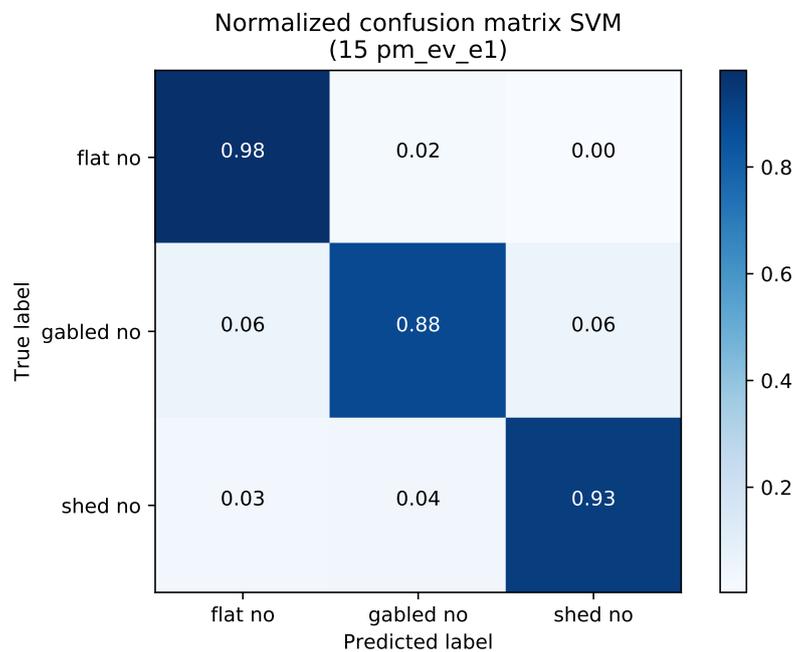


Figure 4.10: Confusion matrix of roof type classification

The classification accuracy is 93% as an average of all classes. This can be seen as a very good result. However, in reality buildings often have one

or more extensions and do not only vary in their size and roof type (see figure 2.2). Thus, in a next experiment we add noise to the building models in the form of differently sized extensions.

4.4.1.4 How does Shape DNA perform at predicting the roof type of a building with an extension?

In this experiment we extract Shape DNA of synthetic building models with three different roof types and extensions of three different sizes. First, we use 100 building models of each roof type with an extension of $s = 1$. Again, we select the building models randomly from the pool of 300 buildings per roof+extension type (section 4.1.4). This results in a dataset with the exact same building models as in the previous experiment, only that the buildings have a $1m \times 1m \times 1m$ extension this time. The results of classifying this dataset with a kernel SVM can be seen in the confusion matrix in figure 4.11.

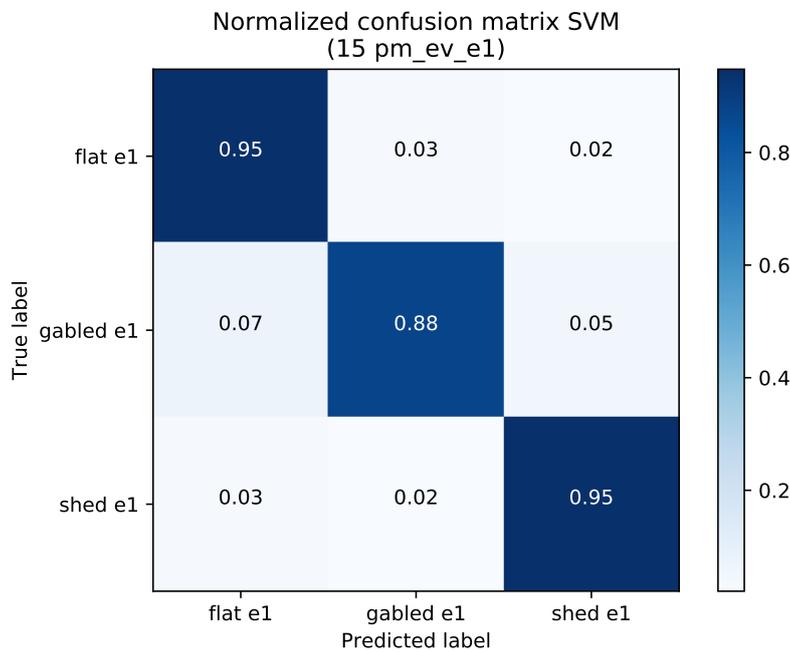


Figure 4.11: Confusion matrix of roof type classification with extension of $s = 1$

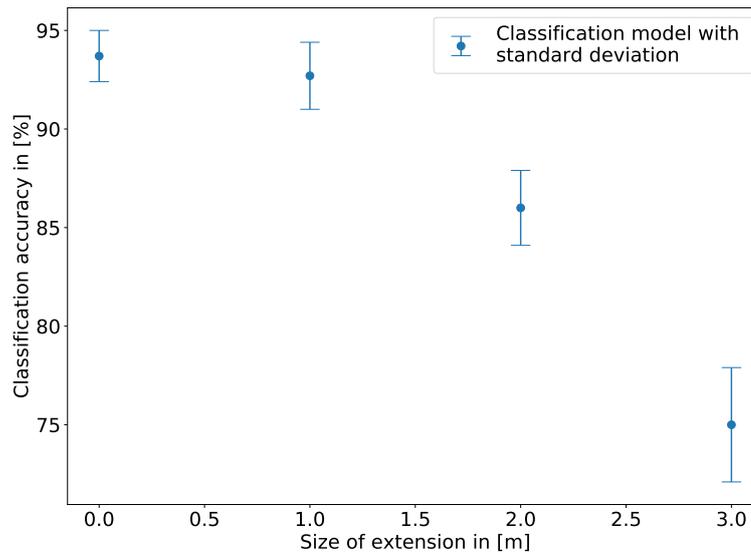


Figure 4.12: Confusion matrix of roof type classification with extension of $s = 1$

The classification accuracy is 92%. By comparing this to the result of the previous experiment we conclude that the extension did not have a significant impact on the detection of the roof type. We now repeat the same experiment with extensions of $s = 2$ and $s = 3$. The classification accuracies of all the roof type predictions, with and without an extension can be seen in figure 4.12. We can see that the classification accuracy goes down to 75% with an extension of $s = 3$.

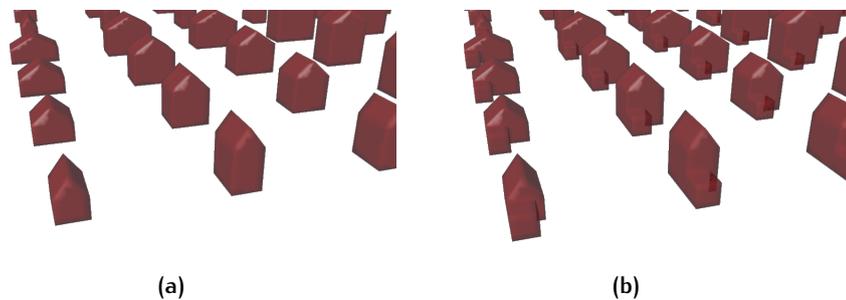


Figure 4.13: Synthetic building models with and without extension

Figure 4.13 shows some exemplary buildings with extensions of $s = 3$ and the same buildings without an extension. We can see a relevant change in the shape of the buildings, especially for buildings with smaller dimensions. This is the reason why Shape DNA, being a global shape descriptor, cannot accurately detect the local similarity (same roof) in the buildings anymore. We can further see this effect by drawing a random sample of 100 buildings per roof type with all possible extension sizes including buildings without an extension. Classifying this dataset only results in a 70% classification accuracy. However, we have now theoretically reduced the training size for each type (roof+extension) to 1/4 of the previous experiments, as we still only draw 100 buildings per roof type including all 4 extension types.

If we increase the sample size to 400 buildings in total, the classification accuracy rises to 85%, which is a promising result. Figure 4.14 shows how the classification accuracy increases with an increasing number of training samples. In the next experiment we will see how other shape descriptors applied on the same dataset perform in a classification.

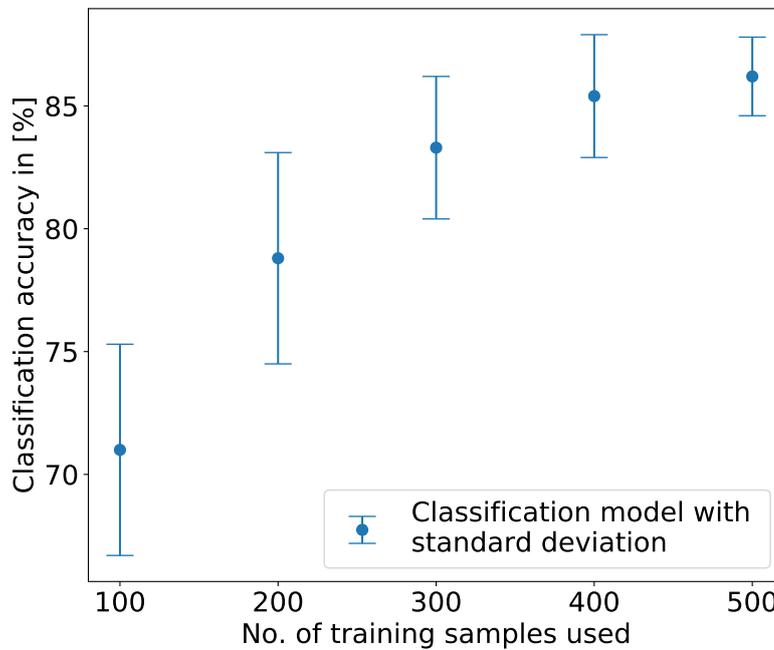


Figure 4.14: Accuracy of roof type classification based on SVM classifier using Shape DNA and a varying number of training samples.

4.4.1.5 How do other shape descriptors perform at predicting the roof type of a building?

In this experiment we extract the footprint area and perimeter, gutter height, surface area, roof count and roof angle from the synthetic building models. By looking at the roof count and roof angle we can already classify the three roof types with simple rules (table 4.7).

Table 4.7: Roof count and angle of synthetic building models

Roof type	Flat	Shed	Gabled
Number of roof segments	1	1	2
Angle of roof segments	$\alpha = 0^\circ$	$\alpha > 0^\circ$	$\alpha > 0^\circ$

This does not change, when adding an extension to the models, as the number of roof segments increases for each type by one. Since the extensions are flat, the roof angle will stay the same for flat buildings and decrease for shed and gabled building types (section 3.3.1.2). Thus, including roof count

and roof angle features leads to classification accuracies of 100%. Footprint area, perimeter and gutter height do not include any information about the roof type of the building as they stem from the same distribution of width, length and height values. However, if we add the surface area to that feature set, the combination of these features can allow to induce the roof type. We try to classify the model with 100 buildings per roof type and a random presence of extensions. We, again, use a kernel SVM for the classification. Figure 4.15 shows a confusion matrix of this classification.

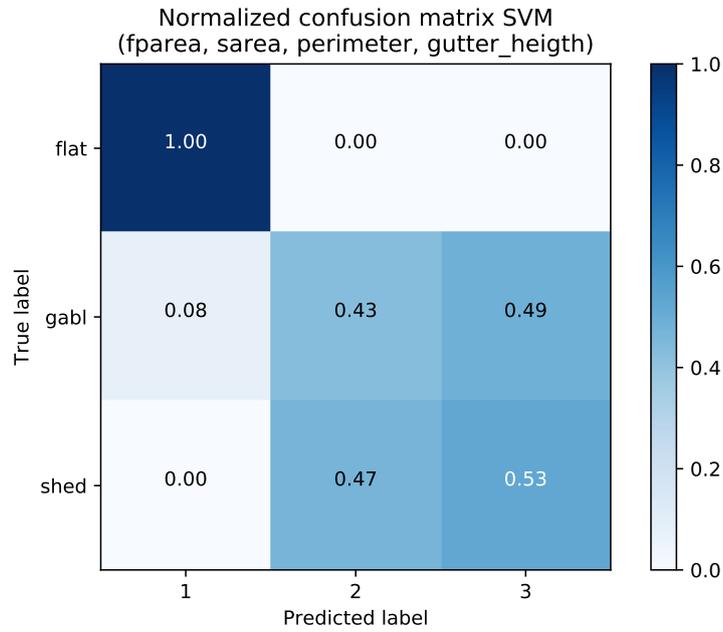


Figure 4.15: Confusion matrix of roof type classification with extensions

The building models with a flat roof can be predicted with 100% accuracy. This is a result of the way we generated the synthetic building set. Building models with equal length, width and height always have a greater surface if they are of type gabled or shed. Remarkably, this already allows the detection of buildings with flat roofs. The probability of correctly classifying a building of one of the remaining classes shed and gabled, is thus 50%. With 43% and 53% accuracy respectively, the classification of these types can be seen close to random.

4.4.1.6 Summary and conclusion

With the experiments conducted in the previous sections we have shown that we need a regular mesh to extract precise Shape DNA. The density of the mesh used in the experiments was $0.2m$. We do not expect a marginal influence for building classification if we increase or decrease this number slightly. However, the meshes with highly irregular meshing created in section 3.2.3 may not be adequate for extracting Shape DNA, due to their irregular density. By applying a topologic smoothing to the mesh the precision of the Shape DNA can be increased. Regarding the classification of building models with Shape DNA the results are mixed. When trying to classify building models that only change in their roof type, Shape DNA delivers

good results. Provided that we cannot extract the relevant local feature directly section 2.4.1, Shape DNA can be a useful global shape descriptor for building classification. However, the classification accuracy drops significantly when extensions are added to the building, and, thus, the influence of the roof type on the global shape of the building is reduced. This may be problematic for SBST classification, as high intra-class variabilities of the global shape are likely section 2.2.2. Furthermore, the local shape features allowed to detect the flat roof types with high accuracy, as a result of the consistently smaller surface area of this type. While this is of course not the case in reality, similar shape patterns may be possible. However, Shape DNA could not capture this pattern as good as the relation of surface area and local shape features could.

4.4.2 SBST prediction

In the following experiment we use all the extracted features from the building models and try to predict the SBST in different levels of detail using different feature combinations. We first construct a subset from the available training set representing all the available typologies with at least 100 samples. We then undersample to the minority class which leaves us with 800 samples evenly divided into 8 classes (see table 4.1). However, this sample represents less than 1% of the building stock of the Groningen region. The main reason we choose to undersample the data is the runtime of the classification. With 100 samples per class a 5-fold cross validation for tuning the classifiers hyperparameters and simultaneous evaluation of the classification results can be conducted within a few minutes. This allows to conduct multiple classifications with different settings, while still using a reasonable amount of buildings for a reliable measure of the classification performance.

We only show the classification results either using a SVM or a RF classifier as they resulted in the best classification performance. A k-NN classifier often performs slightly worse than the RF and an ANN usually performs worse than the other classifiers on this dataset.

4.4.2.1 *How do the features extracted in section 3.3.1 perform at predicting the LLRS?*

Looking at the building models and the preliminary experiments in section 2.2.2 we conclude that the LLRS is the most influential parameter on the geometry of the building. First, we extract the Shape DNA of the building models. Figure 4.16 shows the exemplary building of figure 2.2 in a feature space spanned by the first 15 eigenvalues of the LBO.

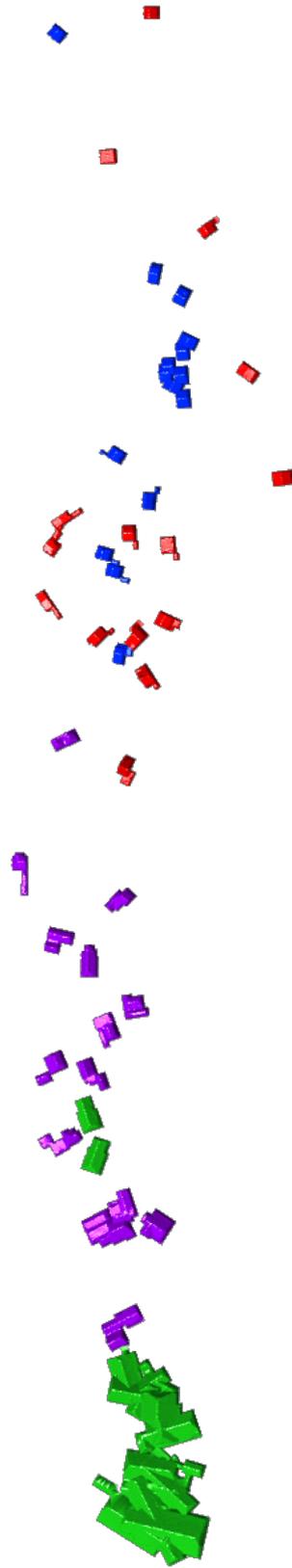


Figure 4.16: MDS plot of buildings in unnormalised "Shape DNA-space"

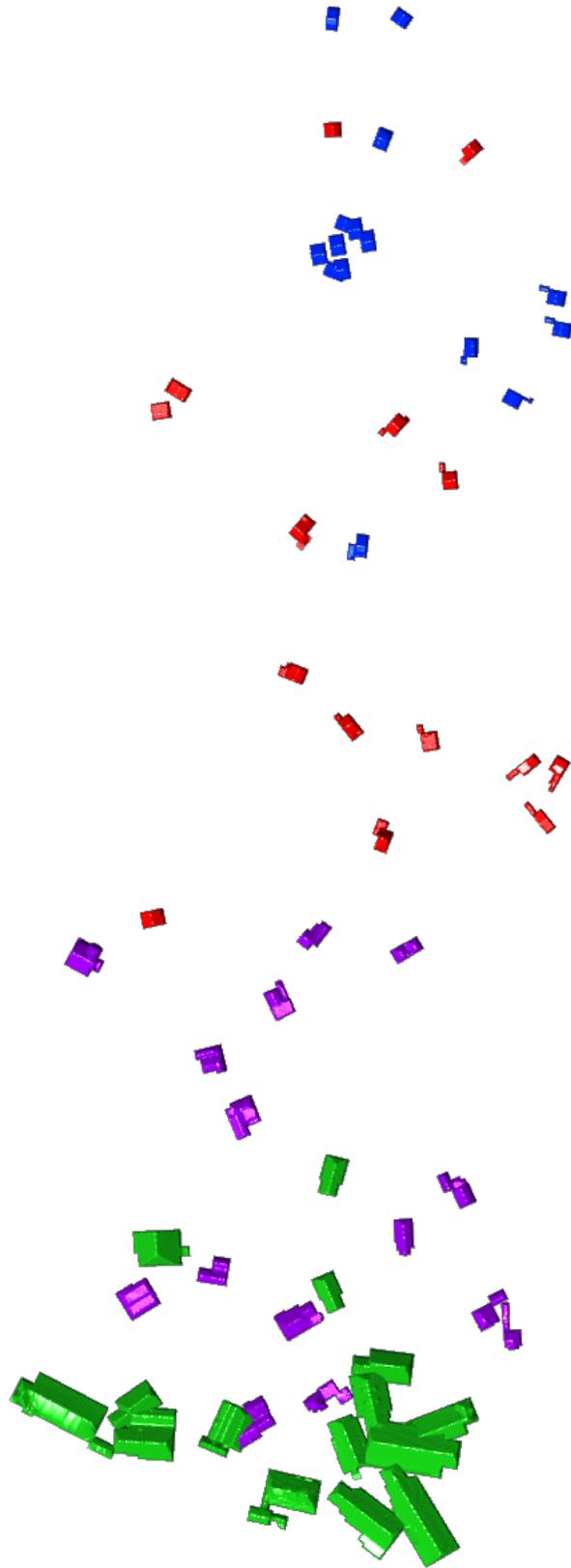


Figure 4.17: MDS plot of buildings in normalised "Shape DNA-space"

We try to classify the **LLRS** using the Shape DNA features and the year of construction of the sample including 800 buildings. We thereby try to answer the question if Shape DNA is useful for the classification of the **LLRS**. Similar to the inspection of figure 4.16 and figure 4.17 the classification using the unnormalised and normalised spectrum does not allow a clear conclusion whether to prefer one over the other (2% better performance with 6% standard deviation). Nevertheless, both classification results can be regarded as good. By reducing the number of eigenvalues from 50 to 15 we can even achieve slightly (not marginally) better results, mostly in the detection of terraced houses. Figure 4.18 shows the confusion matrix of the **LLRS** classification using the first 15 unnormalised eigenvalues and the year of construction of the buildings.

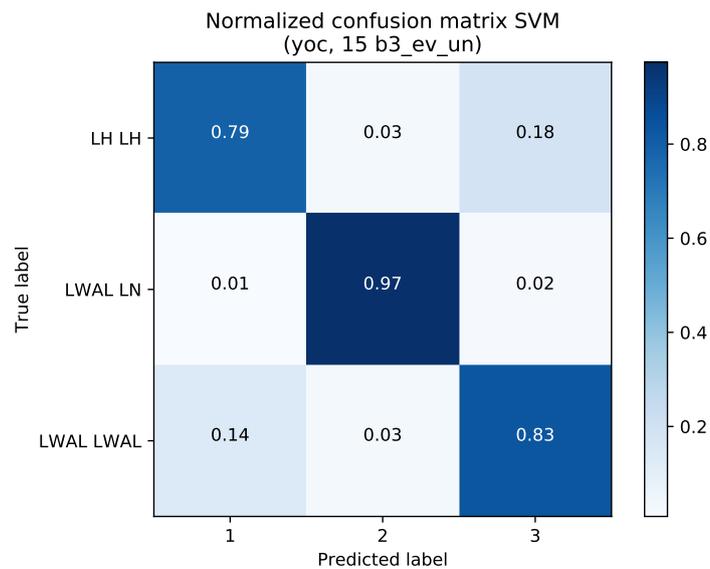


Figure 4.18: Confusion matrix of **LLRS** prediction with Shape DNA and year of construction

In a next experiment we compare the result to different feature sets to further rank the performance of the previous experiment. First we deploy a **RF** using the features year of construction, footprint area, gutter height, roof count and roof angle. To judge the influence of the **3D** information extracted from the point cloud we plot the feature importance of the random forest in figure 4.19.

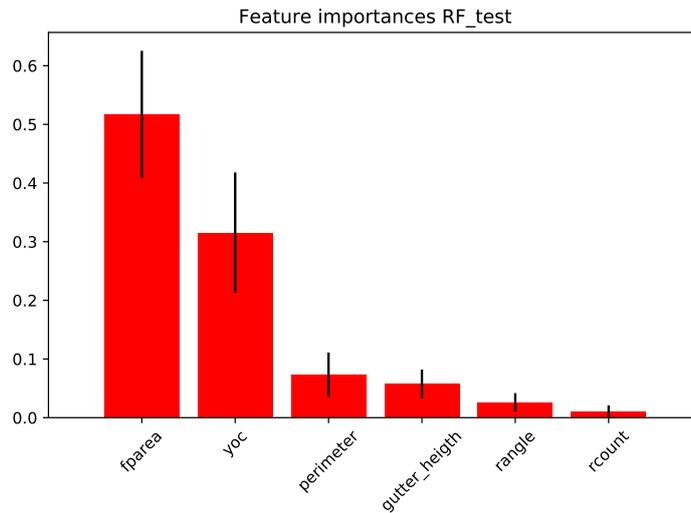


Figure 4.19: Feature importance in LLRS prediction according to RF

This shows that the year of construction and the footprint area are the most important features. We deploy a SVM only using these two features and get a classification accuracy as a mean over all classes of 87%, which is matching the results of the previous classification using Shape DNA. The classification accuracy per class can be seen in figure 4.20).

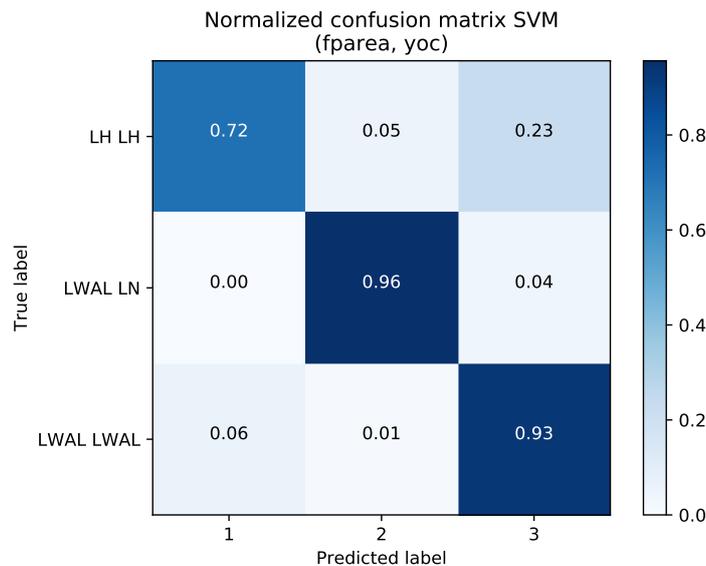


Figure 4.20: Confusion matrix of LLRS prediction only using footprint area and year of construction

4.4.2.2 How do the features extracted in section 3.3.1 perform at predicting the SBST?

We first deploy a RF again to inspect the feature importance for the classification (figure 4.21). The year of construction and footprint area are still the most important features, but 3D information gains importance in this classification compared to the prediction of the LLRS.

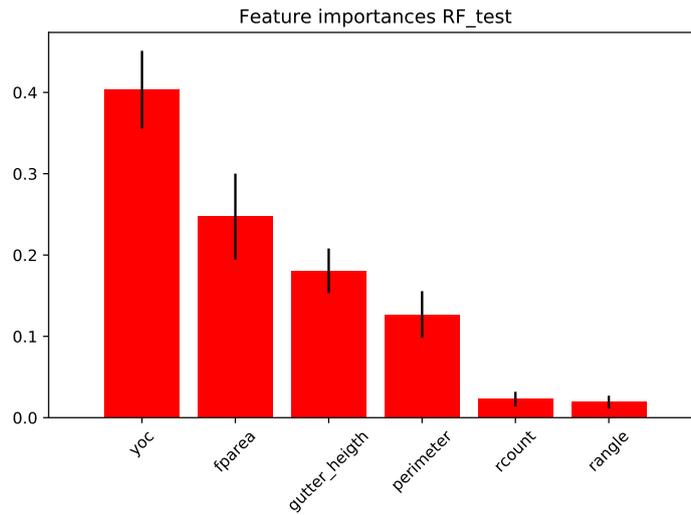


Figure 4.21: Feature importance according to RF. See table 3.2 for a description of the features.

2D		3D					semantic	Classification accuracy and (standard deviation) in %
fparea	perimeter	gutter_height	rcount	rangle	sarea	Shape DNA	yoc	
•	•							60 - (2)
•	•						•	75 - (2)
•	•				•		•	78 - (3)
•	•	•	•	•	•		•	79 - (4)
•	•					•	•	75 - (3)

Table 4.8: Overview of SBST classification results. See table 3.2 for a description of the features.

This time the RF also significantly performs better than a SVM with a classification accuracy of 79% (4% std. dev.) vs. 71% (2% std. dev.). Figure 4.22 shows the confusion matrix of the best performing GEM classification.

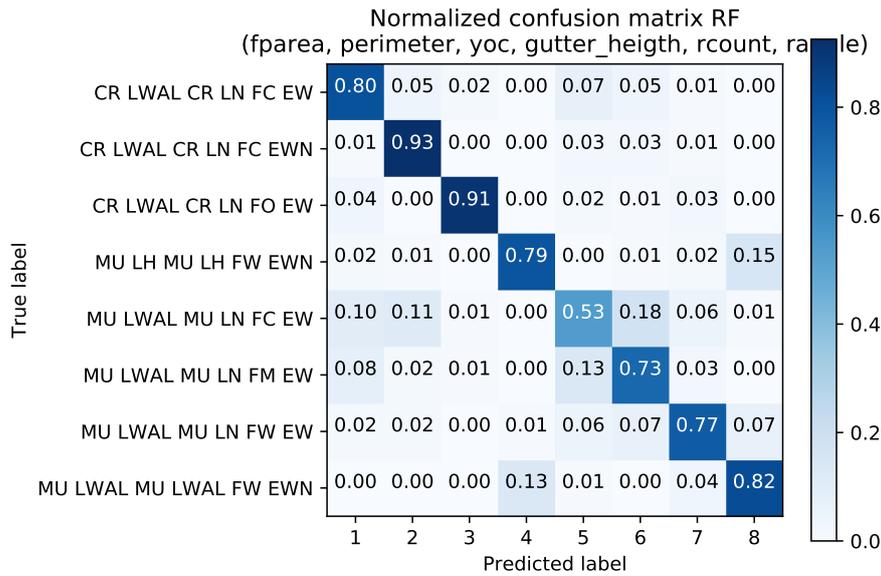


Figure 4.22: Confusion matrix GEM classification

Similar to the classification of the LLRS, Shape DNA cannot improve the classification and performs slightly worse the local shape features with an accuracy of 75% (3% std. dev.).

4.4.2.3 How does the number of training samples influence SBST prediction?

In this experiment we want to see how the size of the training set influences the SBST prediction. This is important, considering that training data with label SBSTs is expensive and difficult to gather. Thus, it is desirable to achieve adequate prediction accuracies, even with a small training set. For this experiment we use the third model of table 4.8 (footprint area, perimeter, gutter height, surface area) with a RF classifier. Then we randomly draw 80, 60, 40 and 20 samples from the initial 100 training samples, always equally distributed over the classes. We repeat this experiment 5 times. In every run the classification accuracy is measured with a 5-fold CV. This, however, means that the test set, e.g. for the run with 20 samples is as little as 4 building samples. The results can be seen in figure 4.23.

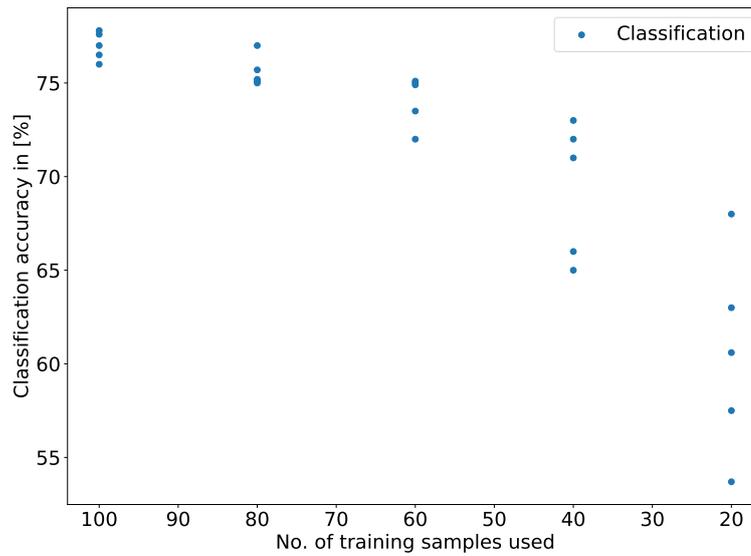


Figure 4.23: Accuracy of **GEM** classification based on **RF** classifier using footprint area, perimeter, gutter height, surface area and a varying number of training samples.

4.4.2.4 Summary and conclusion

The experiments in this section show that the prediction of **SBSTs** in our dataset is already possible by using the footprint area and year of construction of the building resulting in accuracies better than 70%. For the **LLRS** prediction we cannot prove that **3D** geoinformation can help to predict the **LLRS** in this dataset. By only using the footprint area and the year of construction we can already achieve accuracies better than 85%. The classification of the **LLRS** its materials, the floor and exterior walls of a building for the eight most common building types in the region of Groningen lead to an average accuracy of 78%. Here **3D** shape features slightly improve the classification result. From these **3D** the gutter height has the biggest influence in this classification. This influence can, however, not be justified from an engineering point of view. It is therefore possible that this is an effect of patterns from building practice and regulations in the building stock, or our tested dataset. Shape DNA could not improve the classification of the **LLRS** and resulted in a decrease in classification accuracy when applied in the **SBST** prediction.

5

CONCLUSION, DISCUSSION AND RECOMMENDATIONS

5.1 SUMMARY AND CONCLUSION

In this thesis we have investigated into the use of remote sensing and cadastral data to assist large scale seismic vulnerability assessment. Our main goal was to develop and implement a methodology to predict the **SBST** for buildings in the Groningen region. To that end, we generate **3D** building models, apply local and global shape descriptors to extract geometric features from the building model and combine these features with the year of construction to represent a building by a feature vector. We use this representation to predict the **SBST** of a building in a supervised machine learning approach. We describe the **SBST** of the Groningen buildings by six **GEM** attributes, namely the **LLRS** in two distinct directions, the material of the **LLRS** in both directions, the floor of a building and a description of its exterior walls. This information can allow empirical assessment of the seismic vulnerability of a building. A **RF** classifier delivers the best results and allows to predict the **SBST** with accuracies between 53% and 93%, depending on the **SBST**. Furthermore, we show the importance of different features for the classification process. The year of construction proved to be the most important feature for the presented **SBST** prediction, followed by the footprint area. Both features are directly available from the **BAG** dataset, which makes the preprocessing steps, such as the **3D** model generation, basically unnecessary. In general, we show that the prediction of **SBSTs** for seismic vulnerability assessment on a large scale is possible. We have, thus, also validated the results of previous case studies in related work for a sample of **SBSTs** from the Groningen building stock.

5.1.1 Limitations

We have only conducted our experiments on a small sample consisting of less than 1% of the Groningen building stock, mainly to reduce the runtime of the experiments. Furthermore, only the eight most common **SBSTs** are predicted in our workflow. This is due to the fact that we only have a small number of training samples available for the remaining types. The full building stock of Groningen includes at least 30 different **SBSTs**. The supervised learning approach requires a reasonable number of training samples for each **SBST** and the collection of further training samples is still a non trivial process. To facilitate this process we show the influence of the sample size on the predictive performance. This experiment hints that even with as little as 40 training samples per class, **SBST** prediction can still deliver adequate results.

5.1.2 Answer to research question

Which shape descriptors are relevant for the prediction of seismic building structural types and how can they be applied on a building representation?

Local shape descriptors that describe the building on specific points and their local surroundings can be seen as the most relevant for **SBST** prediction (section 2.4.1). In our test dataset of the Groningen building stock, the footprint area was the most important shape feature, followed by the footprint perimeter. The extraction of these features does not require the reconstruction of the building geometry and not even the use of **3D** data. The use of further local shape descriptors, such as the gutter height or roof angle, and the surface area as a global shape descriptor, only improve the classification slightly. Approximations of these features can be extracted directly from the roof point cloud of a building, or a polygon mesh. The use of Shape DNA resulted in a decline of classification accuracy for the prediction of the **SBST** described by six **GEM** attributes. The reconstruction of a structured semantically enriched **3D** building model can be helpful to extract further local shape features, such as the number and size of extensions. First steps towards the creation of such a model are implemented in this thesis, e.g. with the identification of planar roof segments.

To which extent is it possible to describe geometric similarities of buildings using Shape DNA?

Shape DNA can measure geometric similarities of buildings by describing the global shape of a building with a one-dimensional vector. This vector can then be used for shape analysis, such as comparison or classification. In this thesis we have used between 15 and 50 eigenvalues of the **FEM** discretisation of the **LBO** to form the Shape DNA. Varying the number of eigenvalues did not have a significant influence on shape classification. A requirement for extracting precise eigenvalues is a dense and regular polygon mesh of the building geometry. On the real world dataset from the Groningen region, we have gained a good approximation of such a mesh, only using a point cloud of building roofs and building footprints. Further improvements of the mesh quality require the geometric reconstruction of the building first. It is still unclear to which extent the mesh quality and therefore the precision of the eigenvalues also has an effect on shape classification.

To gain further insight, independent of the mesh quality and use case of **SBST** classification, we also generate synthetic building models that allow to gain a dense and regular polygon mesh of a building. As an example for shape based building classification, we succeeded to predict the roof type (flat, gabled, shed) of buildings using Shape DNA in combination with a **SVM**. We have also shown that an extension of $2m \times 2m \times 2m$ on a building already limits the predictive accuracy significantly. In this case the roof has a smaller impact on the global shape of the building. Local features, only describing the shape of the roof have a clear advantage in such a situation. Shape DNA may be useful in situations where local features, such as the roof angle or number of extensions of a buildings, are unknown or difficult to extract. However, any non-isometric change of the building shape also changes the Shape DNA. This means, if buildings differ at many local points, the ability of Shape DNA to only represent relevant local features for a

specific classification task is limited. A sufficiently large number of training samples with different types of geometric noise may allow to overcome this limitation. However, to what extent a real building stock can be classified is still unclear, as occurring buildings may include many different shape patterns.

Future work in the area of shape analysis for [SBST](#) classification and Shape DNA as a shape descriptor for buildings is necessary.

5.2 DISCUSSION AND RECOMMENDATIONS

5.2.1 Shape analysis of buildings

Considering [SBST](#) classification beyond the dataset used in this thesis and shape based building classification in general, investigation into shape analysis of buildings is still necessary. As a first step, we suggest to improve the input, namely the building model, for the shape analysis. By making use of the segmented building point cloud generated in this thesis an [LOD2](#) model could be created, with little to no manual modeling effort. Such a model can also facilitate to extract further semantic information. From a scientific point of view, a reconstructed building model is desirable to compare different methods independent of representation and quality of the input data. For further use of Shape DNA, meshes such as the ones used for the synthetic building models allow to exclude influence of the meshing. Besides Shape DNA, investigations into the following methods may be of interest.

5.2.1.1 *Deep learning*

In general, deep learning methods combine the steps of feature extraction and classification, and thus allow to extract only relevant features for a specific problem (section [2.4.2](#)). Thereby, they may, however, require a larger number of training samples. This can be problematic for [SBST](#) prediction, as the number of training samples is often very sparse. Here, low dimensional local shape descriptor can have a significant advantage. However, with a sufficiently large number of training samples available, deep learning architectures may be appropriate to use for building or [SBST](#) classification.

5.2.1.2 *Simple shape features*

The most important shape features for the [SBST](#) classification in this thesis are the footprint area and perimeter. Related work in the area often comes to the same conclusion. The combination of these features can detect simple patterns in a building stock mostly related to the size of a building. For similar [LLRS](#) (and their materials) but different exterior walls and floor types, the influence of the building size can often not be justified from an engineering point of view. It is more likely that patterns regarding these types are a result of building practices during specific times and regulations in the region. However, it is questionable if such patterns are present beyond the size of a building. Thus, simple local shape features may often be enough to

classify buildings according to their [SBST](#). Using such features also allows a more transparent classification process. This can also be justified if we think about, how humans would classify buildings. Beyond the size and roof type there are often no attributes solely describing the shape of a building, especially on a scale that can be captured with aerial or satellite remote sensing data. Further classification may be based on spatial relations, such as adjacency, or visual information, e.g. describing the materials of a building (section [5.2.2.2](#)).

5.2.2 Automatic [SBST](#) prediction

Future work is necessary in order to investigate into the suitability of the developed methodology for the remaining building stock of Groningen or for different regions. The following sections describe some of the possibilities to improve and extent the current workflow.

5.2.2.1 *Preprocess training samples*

Training samples for [SBST](#) prediction are not always reliable. They may stem from different sources (such as in our case), that are not always consistent and correct, due to a challenging assignment process of [SBST](#) labels. To improve the prediction of unseen data an outlier detection applied on the training set is necessary. This could be done with a one-class [SVM](#) in the same way Geiß et al. [[22](#)] describe this process (see section [2.5](#)).

5.2.2.2 *Potential of different datasets and features*

Our implementation only considers data from two different sources ([BAG](#) and [AHN](#)), and two different types of features (geometric and semantic). In view of extending the workflow to more [SBST](#) classes, additional data sources may be necessary.

Geographic features

Similar to the assumption that geometric features reveal patterns from building practice and regulations, such patterns may also be identified in geographic information. Buildings in the same region or area of a city are often more likely to have the same [SBST](#) as buildings that are far apart. Thus, encoding the location of a building in relation to the city and its neighbours may provide valuable information for the classification.

Image features

Images may provide valuable information for [SBST](#) prediction. Similar to a human interpretation of building classification, building patterns may be visible in image data, such as the use and arrangement of different materials on the outer wall and roof of the building. Tutzauer and Haala [[59](#)] have shown that it is possible to predict building use from terrestrial images with a [CNN](#). We believe if such a pipeline is supervised with [SBSTs](#) ground truth it could be a good alternative, or extension of the approach taken in this thesis.

Alternative training samples

BIM or CAD models may be useful training samples, even for the prediction of real buildings. The geometry as well as the SBST in such models is often known, and thus could be treated in the same way as inspected buildings combined models gained from remote sensing data.

Contextual information

Contextual information such as known building codes or common building practice may be useful information for SBST classification. This could lead to a workflow where knowledge about the building stock is integrated into a probabilistic model.

5.2.2.3 *Suitability in different region*

In theory, our method can also work in different regions outside of Groningen and the Netherlands. However, several parameters may need to be adapted. First, the GEM attributes used in this thesis may need to be adjusted to reflect the most influential parameters on the vulnerability of the building stock in question (see section 2.1.2). This process is strongly connected to building inspections and subsequent generation of the training set, which is a necessary step in our approach. Furthermore, input datasets such as the point cloud may not be available. However, we proved in this thesis that information from the building footprint and the building year of construction can already lead to good classification results. These two features are available in many regions all over the world [6].

5.2.2.4 *Direct large scale seismic building vulnerability assessment*

Instead of predicting a SBST it may be possible to conduct large scale seismic vulnerability assessment in a more direct way. Analytic vulnerability assessment with a building model automatically reconstructed from remote sensing data can be a next step for large scale seismic vulnerability assessment. However, traditional approaches for computer aided vulnerability assessment may still be too time consuming without grouping a building stock first. To circumvent this problem it may be possible to directly predict vulnerability parameters instead of SBST labels in a supervised learning approach.

5.2.2.5 *Usefulness in other hazardous situations*

Knowledge about the structural building type, such as the structural system and materials of (many) buildings may also be of importance in other hazardous situation, such as fires or floods. Besides seismic load, fires or floods can also affect the structural behaviour of a building and thus result in economic loss or life safety. To mitigate impact of such hazardous situations it is important to have fast access to a detailed structural model of a building. Region wide digital city models enriched with structural information about the buildings is thus of great importance.

5.3 REFLECTION

The originality of this thesis is the combination and integration of the three main parts, geomatics, pattern recognition and seismic engineering. The main contribution is a literature study summarising relevant aspects in these parts, and a prototype to classify SBSTs. Initial tests show that the ideas developed in this thesis have potential. The research also led to the realisation of the relevance for a structurally aware building model created from remote sensing data and ancillary (geo-)information, potentially gained by using machine learning techniques. Besides a geometric reconstruction with sufficient accuracy such a model should include information about the structural system and structurally relevant materials of the building. Such models - available for large building stocks of cities, regions and countries - could be of great value for pre-, peri- and post-event analysis and decision making in hazardous situations (e.g. fire, flood, earthquake). Currently available standards for building models, such as CityGML, are focusing on visualisation or navigation purposes and analysis where mostly the geometry is important, but not the structural system and materials of the building. The data model of the GEM taxonomy is compatible with BIMs which could serve as a basis. Insight and understanding, and some of the developed tools of this thesis can be used towards the creation of such models.

BIBLIOGRAPHY

- [1] Arteaga, R. J. [2014], Laplace-beltrami spectra as ‘shape-dna’ of surfaces using the closest point method, Master’s thesis, Simon Fraser University.
- [2] Baart, F. and de Boer, G. [2014], ‘Dataset documentation ahn2’.
URL: <https://publicwiki.deltares.nl/display/OET/Dataset+documentation+AHN2>
[accessed: 05 January 2018]
- [3] Bilham, R. [2009], ‘The seismic future of cities’, *Bulletin of Earthquake Engineering* **7**(4), 839.
URL: <https://doi.org/10.1007/s10518-009-9147-0>
- [4] Biljecki, F. [2017], Level of detail in 3D city models, PhD thesis, Delft University of Technology, Delft, the Netherlands.
- [5] Biljecki, F., Ledoux, H. and Stoter, J. [2017], ‘Generating 3d city models without elevation data’, *Computers, Environment and Urban Systems* **64**, 1 – 18.
URL: <http://www.sciencedirect.com/science/article/pii/S0198971516302617>
- [6] Biljecki, F. and Sindram, M. [2017], *Estimating building age with 3D GIS*, Vol. IV-4/W5.
- [7] Billen, R., Cutting-Decelle, A., Marina, O., DUARTE de ALMEIDA, J.-P., Caglioni, M., Falquet, G., Leduc, T., Métral, C., Moreau, G., Perret, J., Rabino, G., García, R., Yatskiv, I. and Zlatanova, S. [2014], *3D City Models and urban information: Current issues and perspectives*.
- [8] Borfecchia, F., Pollino, M., Cecco, L. D., Lugari, R., Martini, R., Porta, L. L., Ristoratore, E. and Pascale, C. [2010], ‘Active and passive remote sensing for supporting the evaluation of the urban seismic vulnerability’, *Italian Journal of Remote Sensing* pp. 129–141.
- [9] Borzi, B., Dell’Acqua, F., Faravelli, M., Gamba, P., Lisini, G., Onida, M. and Polli, D. [2011], ‘Vulnerability study on a large industrial area using satellite remotely sensed images’, *Bulletin of Earthquake Engineering* **9**(2), 675–690.
URL: <http://dx.doi.org/10.1007/s10518-010-9211-9>
- [10] Breiman, L. [2001], ‘Random forests’, *Machine Learning* **45**(1), 5–32.
URL: <https://doi.org/10.1023/A:1010933404324>
- [11] Brzev, S., Scawthorn, C., Charleson, A. W., Allen, L., Greene, M., Jaiswal, K. and Silva, V. [2013], Gem building taxonomy (version 2.0), Technical report, GEM Foundation.
URL: <http://pubs.er.usgs.gov/publication/70058718>
- [12] Calvi, G. M., Pinho, R., Magenes, G., Bommer, J. J., Restrepo-Vélez, L. F. and Crowley, H. [n.d.], ‘Development of seismic vulnerability assessment methodologies over the past 30 years’.

- [13] Cawley, G. C. and Talbot, N. L. [2010], 'On over-fitting in model selection and subsequent selection bias in performance evaluation', *J. Mach. Learn. Res.* **11**, 2079–2107.
URL: <http://dl.acm.org/citation.cfm?id=1756006.1859921>
- [14] Christodoulou, A., Kokkos, A. and Palmieri, M. [2017], Automated building stock data mining and classification using open source data, in 'IASS 2017, At Hamburg, Germany'.
- [15] Cortes, C. and Vapnik, V. [1995], 'Support-vector networks', *Machine Learning* **20**(3), 273–297.
URL: <https://doi.org/10.1007/BF00994018>
- [16] Duda, R. O., Hart, P. E. and Stork, D. G. [2000], *Pattern Classification (2Nd Edition)*, Wiley-Interscience.
- [17] Duin, R. and Pekalska, E. [2015], *Pattern Recognition: Introduction and Terminology*, 37 Steps ebook.
- [18] Ferreira, T., Vicente, R., Varum, H., Mendes Silva, R. and Costa, A. [2012], *Vulnerability assessment of urban building stock: a hierarchic approach*.
- [19] Fournier d'Albe, E. M. [1982], 'An approach to earthquake risk management', *Engineering Structures* **4**(3), 147–152.
URL: <http://www.sciencedirect.com/science/article/pii/0141029682900025>
- [20] Gao, Z., Yu, Z. and Pang, X. [2014], 'A compact shape descriptor for triangular surface meshes', *Computer-Aided Design* **53**, 62–69.
URL: <http://www.sciencedirect.com/science/article/pii/S0010448514000645>
- [21] Geiß, C. [2015], Seismic vulnerability assessment of built environments with remote sensing, <http://dx.doi.org/10.18452/17104>, Humboldt-Universität zu Berlin.
- [22] Geiß, C., Aravena Pelizari, P., Marconcini, M., Sengara, W., Edwards, M., Lakes, T. and Taubenboeck, H. [2015], 'Estimation of seismic building structural types using multi-sensor remote sensing and machine learning techniques', *ISPRS Journal of Photogrammetry and Remote Sensing* **104**, 175–188.
URL: <http://www.sciencedirect.com/science/article/pii/S0924271614002007>
- [23] Geiß, C., Taubenböck, H., Tyagunov, S., Tisch, A., Post, J. and Lakes, T. [2013], 'Assessment of seismic building vulnerability from space', *Earthquake Spectra* **30**(4), 1553–1583.
URL: <https://doi.org/10.1193/121812EQS350M>
- [24] Haala, N., Brenner, C., Karl-Heinrich, A., for Photogrammetry, I. S. and Sensing, R. [1997], *Generation of 3D City Models from Digital Surface Models and 2D GIS*, INTERNATIONAL ARCHIVES OF PHOTOGRAMMETRY AND REMOTE SENSING, International Society for Photogrammetry and Remote Sensing.
- [25] Haala, N. and Kada, M. [2010], 'An update on automatic 3d building reconstruction', *ISPRS Journal of Photogrammetry and Remote Sensing* **65**(6), 570–580.
URL: <http://www.sciencedirect.com/science/article/pii/S0924271610000894>

- [26] He, M., Zhu, Q., Du, Z., Hu, H., Ding, Y. and Chen, M. [2016], 'A 3d shape descriptor based on contour clusters for damaged roof detection using airborne lidar point clouds', *Remote Sensing* **8**(3), 189.
URL: <http://www.mdpi.com/2072-4292/8/3/189>
- [27] Hecht, R., Meinel, G. and Buchroithner, M. [2015], 'Automatic identification of building types based on topographic databases – a comparison of different data sources', *International Journal of Cartography* **1**(1), 18–31.
URL: <http://dx.doi.org/10.1080/23729333.2015.1055644>
- [28] Jain, A. and eds. R.L. Gregory, R. D. [2004], *The Oxford Companion to the Mind, Second Edition*, second edition edn, Oxford University Press, Oxford, UK, chapter Pattern Recognition, pp. 698–703.
- [29] Karimzadeh, S., Miyajima, M., Hassanzadeh, R., Amiraslanzadeh, R. and Kamel, B. [2014], 'A gis-based seismic hazard, building vulnerability and human loss assessment for the earthquake scenario in tabriz', *Soil Dynamics and Earthquake Engineering* **66**(Supplement C), 263 – 280.
URL: <http://www.sciencedirect.com/science/article/pii/S026772611400150X>
- [30] Kazantzi, A., Vamvatsikos, D., Porter, K. and Ho Cho, I. [2014], Analytical vulnerability assessment of modern highrise rc moment-resisting frame buildings in the western usa for the global earthquake model, in 'European Conference on earthquake engineering'.
- [31] Kazmi, I. K., You, L. and Zhang, J. J. [2013], A survey of 2d and 3d shape descriptors, in '2013 10th International Conference Computer Graphics, Imaging and Visualization', pp. 1–10.
- [32] Knopp, J. [2015], Large-scale Classification and Retrieval of 3D Shapes, PhD thesis, KU Leuven.
- [33] Labetski, A., Ledoux, H. and Stoter, J. [2017], *Generalising 3D Buildings from LoD2 to LoD1*.
- [34] Lehoucq, R., Maschhoff, K., Sorensen, D. and Yang, C. [2018], 'Arpack'.
URL: <http://www.caam.rice.edu/software/ARPACK/>
- [35] Lian, Z., Godil, A., Bustos, B., Daoudi, M., Hermans, J., Kawamura, S., Kurita, Y., Lavoué, G., Nguyen, H. V., Ohbuchi, R., Ohkita, Y., Ohishi, Y., Porikli, F., Reuter, M., Sipiran, I., Smeets, D., Suetens, P., Tabia, H. and Vandermeulen, D. [2011], SHREC '11 Track: Shape Retrieval on Non-rigid 3D Watertight Meshes, in H. Laga, T. Schreck, A. Ferreira, A. Godil, I. Pratikakis and R. Veltkamp, eds, 'Eurographics Workshop on 3D Object Retrieval', The Eurographics Association.
- [36] Lugari, A. [2014], Active and passive remote sensing techniques and artificial neural networks in support of buildings seismic vulnerability assessment, PhD thesis, TOR VERGATA UNIVERSITY, ROME, ITALY.
- [37] Maaten, L. J. P. V. D. and Hinton, G. [2008], *Visualizing High-Dimensional Data using t-SNE*, Vol. 9.
- [38] Maqsood, T., Edwards, M., Ioannou, I., Kosmidis, I., Rossetto, T. and Corby, N. [2016], 'Seismic vulnerability functions for australian buildings by using gem empirical vulnerability assessment guidelines', *Natural Hazards* **80**(3), 1625–1650.
URL: <https://doi.org/10.1007/s11069-015-2042-x>

- [39] Masci, J., Boscaini, D., Bronstein, M. M. and Vandergheynst, P. [2015], Geodesic convolutional neural networks on riemannian manifolds, in '2015 IEEE International Conference on Computer Vision Workshop (ICCVW)', pp. 832–840.
- [40] Muntendam-Bos, A. G., Roest, J. P. A. and De Waal, H. [2015], *A guideline for assessing seismic risk induced by gas extraction in the Netherlands*, Vol. 34, Society of Exploration Geophysicists.
- [41] Ng, A. Y. and Jordan, M. I. [2002], On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes, in T. G. Dietterich, S. Becker and Z. Ghahramani, eds, 'Advances in Neural Information Processing Systems 14', MIT Press, pp. 841–848.
URL: <http://papers.nips.cc/paper/2020-on-discriminative-vs-generative-classifiers-a-comparison-of-logistic-regression-and-naive-bayes.pdf>
- [42] Niethammer, M., Reuter, M., Wolter, F.-E., Bouix, S., Peinecke, N., Koo, M.-S. and Shenton, M. E. [2007], *Global Medical Shape Analysis Using the Laplace-Beltrami Spectrum*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 850–857.
- [43] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, É. [2011], 'Scikit-learn: Machine Learning in Python', *Journal of Machine Learning Research* .
URL: <http://hal.inria.fr/hal-00650905>
- [44] Pittore, M. and Wieland, M. [2013], 'Toward a rapid probabilistic seismic vulnerability assessment using satellite and ground-based remote sensing', *Natural Hazards* **68**(1), 115–145.
URL: <https://doi.org/10.1007/s11069-012-0475-z>
- [45] Qi, C. R., Su, H., Mo, K. and Guibas, L. J. [2016], 'Pointnet: Deep learning on point sets for 3d classification and segmentation'.
URL: <https://arxiv.org/abs/1612.00593>
- [46] Quinlan, J. [1987], 'Simplifying decision trees', *International Journal of Man-Machine Studies* **27**(3), 221 – 234.
URL: <http://www.sciencedirect.com/science/article/pii/S0020737387800536>
- [47] Reuter, M., Biasotti, S., Giorgi, D., Patanè, G. and Spagnuolo, M. [2009], 'Discrete laplace-beltrami operators for shape analysis and segmentation', *Computers and Graphics* **33**(3), 381 – 390. IEEE International Conference on Shape Modelling and Applications 2009.
URL: <http://www.sciencedirect.com/science/article/pii/S0097849309000272>
- [48] Reuter, M., Wolter, F.-E. and Peinecke, N. [2006], 'Laplace-beltrami spectra as 'shape-dna' of surfaces and solids', *Comput. Aided Des.* **38**(4), 342–366.
URL: <http://dx.doi.org/10.1016/j.cad.2005.10.011>
- [49] Reuter, M., Wolter, F.-E., Shenton, M. and Niethammer, M. [2009], 'Laplace-beltrami eigenvalues and topological features of eigenfunctions for statistical shape analysis', *Computer-Aided Design* **41**(10), 739 – 755. Selected Papers from the 2007 New Advances in Shape Analysis and Geometric Modeling Workshop.
URL: <http://www.sciencedirect.com/science/article/pii/S0010448509000463>

- [50] Sarabandi, P. [2007], Development of algorithms for building inventory compilation through remote sensing and statistical inferencing, PhD thesis, Stanford University.
- [51] Shahzad, M. and Zhu, X. X. [2015], 'Robust reconstruction of building facades for large areas using spaceborne tomosar point clouds', *IEEE Transactions on Geoscience and Remote Sensing* **53**(2), 752–769.
- [52] Shilane, P., Min, P., Kazhdan, M. and Funkhouser, T. [2004], The Princeton shape benchmark, in 'Shape Modeling International'.
- [53] Smith, O. K. [1961], 'Eigenvalues of a symmetric 3 by 3 matrix', *Commun. ACM* **4**(4), 168–.
URL: <http://doi.acm.org/10.1145/355578.366316>
- [54] Su, H., Maji, S., Kalogerakis, E. and Learned-Miller, E. [2015], 'Multi-view convolutional neural networks for 3d shape recognition'.
URL: <https://arxiv.org/abs/1505.00880>
- [55] Tack, F., Buyuksalih, G. and Goossens, R. [2012], '3d building reconstruction based on given ground plan information and surface models extracted from spaceborne imagery', *ISPRS Journal of Photogrammetry and Remote Sensing* **67**(Supplement C), 52 – 64.
URL: <http://www.sciencedirect.com/science/article/pii/S0924271611001134>
- [56] Tax, D. M. J. and Loog, M. [2017], 'Pattern recognition', Lecture slides TU Delft course.
- [57] Theodoridis, S. and Koutroumbas, K. [2008], *Pattern Recognition, Second Edition*, 2th edn, Academic Press.
- [58] Tomljenovic, I., Höfle, B., Tiede, D. and Blaschke, T. [2015], *Building Extraction from Airborne Laser Scanning Data: An Analysis of the State of the Art*, Vol. 7, Remote Sensing.
- [59] Tutzauer, P. and Haala, N. [2017], 'Processing of Crawled Urban Imagery for Building Use Classification', *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* pp. 143–149.
- [60] van den Brink, L., Krijtenburg, D., van Eekelen, H. and Maessen, B. [2018], 'inwinningsregel bgt en bag'.
URL: <http://imgeo.geostandaarden.nl/def/imgeo-object/pand/inwinningsregel-bgt-en-bag> [accessed 05 January 2018]
- [61] Watanabe, S. [1985], *Pattern Recognition: Human and Mechanical*, John Wiley and Sons, Inc., New York, NY, USA.
- [62] Wieland, M., Pittore, M., Parolai, S., Zschau, J., Moldobekov, B. and Begaliev, U. [2012], 'Estimating building inventory for rapid seismic vulnerability assessment: Towards an integrated approach based on multi-source imaging', *Soil Dynamics and Earthquake Engineering* **36**(Supplement C), 70 – 83.
URL: <http://www.sciencedirect.com/science/article/pii/S026772611200005X>
- [63] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X. and Xiao, J. [2014a], '3d shapenets: A deep representation for volumetric shapes'.
URL: <https://arxiv.org/abs/1406.5670>

- [64] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X. and Xiao, J. [2014b], '3d shapenets: Modelnet benchmark'.
URL: <http://3dshapenets.cs.princeton.edu/> [accessed: 05 January 2018]
- [65] Xiong, B., Jancosek, M., Oude Elberink, S. and Vosselman, G. [2015], 'Flexible building primitives for 3d building modeling', *ISPRS Journal of Photogrammetry and Remote Sensing* **101**(Supplement C), 275–290.
URL: <http://www.sciencedirect.com/science/article/pii/S0924271615000143>
- [66] Zhang, G. P. [2000], 'Neural networks for classification: a survey', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **30**(4), 451–462.
URL: <http://ieeexplore.ieee.org/document/897072/>

COLOPHON

This document was typeset using \LaTeX . The document layout was generated using the `arsclassica` package by Lorenzo Pantieri, which is an adaption of the original `classicthesis` package from Andre Miede.