

MSc thesis in Geomatics for the Built Environment

# Bidirectional enrichment of CityGML and Multi-View Stereo Mesh models

Magdalini - Styliani Tryfona  
2016 - 2017







BIDIRECTIONAL ENRICHMENT OF CITYGML AND MULTI-VIEW STEREO  
MESH MODELS

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

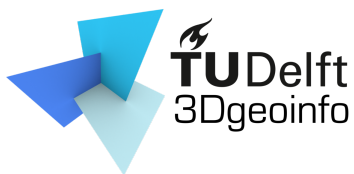
Magdalini - Styliani Tryfona

July 2017

Magdalini - Styliani Tryfona: *Bidirectional enrichment of CityGML and Multi-View Stereo Mesh models* (2017)  
© This work is licensed under a Creative Commons Attribution 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

ISBN 999-99-9999-999-9

The work in this thesis was made in the:



3D geoinformation group  
Department of Urbanism  
Faculty of Architecture & the Built Environment  
Delft University of Technology

Supervisors: Dr. H. Ledoux  
Ir. R.Y. Peters  
Co-reader: Drs. C.W. Quak

# ABSTRACT

The use of 3D city models has increased the last decades due to the evolution of technology. Their use is related to the need for solutions on issues that correspond to the building environment. Modelling aspects, like geometric, semantic information and topology are necessary, in order to provide an integrated result of spatial analysis. The creation of a complete 3D city model that contains all the needed information for an application, is a time consuming and complex process. Furthermore, different 3D city model formats can contain different aspects of the same features. For example, a CityGML model can have the semantic information of a 3D object, while a Multi-View Stereo Mesh model can contain all the geometric and appearance information of the same 3D object.

This thesis is documenting the possible enrichments that can be done bidirectionally between these two different 3D city model formats, when both exist for the same entities. Moreover, it presents the exploration of possible automated enrichment methodologies that can enrich each 3D city model automatically with information from the other one. Finally, an automatic bidirectional enrichment methodology is proposed and is implemented on a testing area, provided in the two different 3D city model formats. This method is based on distance computations between the meshes of the two 3D city models, used to match corresponding features, or part of features, in order to segment semantically the Multi-View Stereo Mesh model (roof, wall, road, terrain, uncertain) and transfer texture to the surfaces of the CityGML model that correspond with the surfaces of the Multi-View Stereo Mesh model.

In addition, distance computations are performed for the validation of the absence of buildings and the shapes of the roofs in the CityGML model, with respect to the information given from the Multi-View Stereo Mesh model. After the implementation of the proposed methodology, it is found that both 3D city model formats can be used for the proposed enrichments of either 3D city model format.

Future improvements are presented for the achievement of using existing information of different formats of 3D city models of the same entities, in order to supplement each 3D city model format with useful information, enhance the performance of spatial analysis and boost the evolution on the use of such models on real life applications.





## ACKNOWLEDGEMENTS

I would like to express my gratitude to all the people involved in helping me throughout this research. First and foremost, I would like to thank my supervisors, Hugo Ledoux and Ravi Peters, for their inspiring thoughts and immense knowledge. Their valuable guidance and suggestions helped me throughout all the stages of my thesis. I would also like to thank Wilko Quak, who was my co-reader and provided me with insightful comments and useful suggestions on my research.

I would like to offer my special thanks to **CycloMedia** and **virtualcitySYSTEMS** for providing the datasets that were used in this thesis.

My appreciation likewise extends to my colleague and friend Lidia, who was willing to lend me her laptop when I needed to run my analyses. I truly value her contribution in the completion of my thesis.

Finally, I must express my very profound gratitude to my parents and to my friends. Nikolas, Liza, Giannis and Sotiria provided me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.



# CONTENTS

1	INTRODUCTION	1
1.1	Characteristics of different 3D city model formats	2
1.2	Research questions and scope	3
1.2.1	Research questions	3
1.2.2	Scope of research	4
1.3	Research contributions	4
1.4	Thesis outline	5
2	THEORETICAL BACKGROUND	7
2.1	CityGML model	7
2.2	MVSM model	9
2.3	Bidirectional enrichment and state-of-the-art in 3D city models	11
2.3.1	Bidirectional enrichment	11
2.3.2	Related work	14
3	BIDIRECTIONAL ENRICHMENT METHODOLOGY	19
3.1	Overview of the methodology	19
3.2	Step 1: Remove the CityGML faces that do not correspond to the MVSM model	20
3.3	Step 2: Match the faces of the MVSM model to the CityGML faces	25
3.4	Step 3: Validate the matching of the MVSM with the CityGML model	30
3.5	Step 4: Provide texture to the CityGML model	34
4	IMPLEMENTATION AND ANALYSIS	39
4.1	Tools	39
4.1.1	Software packages	39
4.1.2	Python packages	39
4.2	Data	40
4.2.1	CityGML input data	41
4.2.2	MVSM input data	45
4.3	Implementation and results	49
4.3.1	1 <sup>st</sup> Implementation of the proposed methodology	49
4.3.2	2 <sup>nd</sup> Implementation of the proposed methodology	60
5	CONCLUSION AND RECOMMENDATIONS	69
5.1	Research questions	69
5.2	Discussion	72
5.3	Conclusion	73
5.4	Future work and recommendations	74





# LIST OF FIGURES

Figure 1.1	Different characteristics between a MVSM (left) and a CityGML (right) model of the same area, representing the same features. . . . .	2
Figure 2.1	Example of a CityGML model (LOD2). . . . .	7
Figure 2.2	The five LODs defined by CityGML (Figure from <a href="#">Biljecki et al. (2016)</a> ). . . . .	8
Figure 2.3	UML diagram of the top level class hierarchy of CityGML (Figure from <a href="#">Kolbe (2009)</a> ). . . . .	8
Figure 2.4	Example of a MVSM model. . . . .	9
Figure 2.5	Steps of the multi-view reconstruction (Figure from <a href="#">Fuhrmann et al. (2014)</a> ). . . . .	10
Figure 2.6	Texture mapping (Figure from <a href="#">Angel et al. (2007)</a> ). . . . .	10
Figure 2.7	Illustrations of the 3D city models that are explored on their bidirectional enrichments. . . . .	11
Figure 2.8	Straightening of a mesh of a MVSM model (Figure from <a href="#">Jonsson (2016)</a> ). . . . .	12
Figure 2.9	Simplification of a mesh of a MVSM model (Figure from <a href="#">Tarini et al. (2010)</a> ). . . . .	13
Figure 3.1	Flowchart of the proposed methodology on the bidirectional enrichment of CityGML and MVSM models. . . . .	20
Figure 3.2	Differences between the corresponding surfaces of the CityGML (transparent) and the MVSM (grey) models. . . . .	20
Figure 3.3	Faces of the CityGML model that are removed (red faces). . . . .	21
Figure 3.4	Flowchart illustrating the parts of step 1 of the proposed methodology. . . . .	21
Figure 3.5	Centroid of the triangle ABC. . . . .	22
Figure 3.6	Point to polygonal mesh distance. . . . .	23
Figure 3.7	Example of the normal vector $\vec{N}$ (cross product) of the triangle $P_2P_3P_1$ . . . . .	24
Figure 3.8	Angle ( $\theta$ ) between two vectors ( $\vec{u}$ and $\vec{v}$ ). . . . .	25
Figure 3.9	Flowchart illustrating the parts of step 2 of the proposed methodology for the segmentation of the MVSM model to roof, wall and uncertain. . . . .	26
Figure 3.10	Unconnected faces on the mesh (scattered triangles and holes). . . . .	27
Figure 3.11	Example of the problem of the connected components. . . . .	28
Figure 3.12	Flowchart illustrating the methodology used for the identification of the different connected components in a semantic class of a MVSM model. . . . .	28
Figure 3.13	Flowchart illustrating the parts of step 2 of the proposed methodology for the segmentation of the MVSM model to road, terrain and uncertain. . . . .	29
Figure 3.14	Flowchart illustrating the process of the quality control of the semantic segmentation of step 2 (Section 3.3) of the proposed methodology. . . . .	32
Figure 3.15	Validation of the shapes of the roofs in the CityGML model with the use of Hausdorff distance computations between the roof class of the MVSM model and the roof surface of the CityGML model. . . . .	33
Figure 3.16	Validation of the absence of buildings in the CityGML model with the use of Hausdorff distance computations between the wall class of the MVSM model and the wall surface of the CityGML model. . . . .	33
Figure 3.17	Flowchart illustrating the process of the texturing of the CityGML model with information from a MVSM model. . . . .	34
Figure 3.18	Example of projection of the triangles ABE, EBC, CDE and EDA to the ABCD plane surface. . . . .	35
Figure 3.19	Example of a triangle that is used for the computation of its area with the use of Heron's formula. . . . .	35
Figure 3.20	Stretch of the faces of the MVSM model, in order to reduce the mismatches on the boundaries of the CityGML model. . . . .	36
Figure 3.21	Topology of adjacent faces. . . . .	37
Figure 4.1	Study area in Amsterdam (pink box). . . . .	40
Figure 4.2	GIS data used for the creation of the CityGML datasets. . . . .	41

Figure 4.3	Dataset of the study area in LOD <sub>1</sub> CityGML model. . . . .	41
Figure 4.4	Dataset of the study area the LOD <sub>2</sub> CityGML model. . . . .	42
Figure 4.5	The study area of the 1 <sup>st</sup> implementation of the proposed methodology (red buildings illustrating the four buildings) and of the 2 <sup>nd</sup> implementation of the proposed methodology (white box illustrating the building block). . . . .	42
Figure 4.6	Testing dataset of the LOD <sub>2</sub> CityGML model (four buildings). . . . .	43
Figure 4.7	Testing dataset of the LOD <sub>1</sub> CityGML model (building block), representing buildings (grey triangulated surfaces), road (transparent triangulated surfaces) and terrain information (grey non-triangulated surfaces). . . . .	44
Figure 4.8	Testing dataset of the LOD <sub>2</sub> CityGML model (building block), representing buildings (grey triangulated surfaces), road (transparent triangulated surfaces) and terrain information (grey non-triangulated surfaces). . . . .	44
Figure 4.9	Height differences between the provided LOD <sub>2</sub> CityGML model (grey) and the one that was corrected (red). . . . .	45
Figure 4.10	Levels of detail of the MVSM. . . . .	46
Figure 4.11	Testing dataset of the MVSM model (four buildings). . . . .	46
Figure 4.12	Testing dataset of the MVSM model (building block). . . . .	47
Figure 4.13	Holes on the testing dataset of the MVSM model (building block). . . . .	48
Figure 4.14	Non-manifold errors (pink dots) on the testing dataset of the MVSM model (four buildings). . . . .	48
Figure 4.15	Non-manifold errors (pink dots) on the testing dataset of the MVSM model (building block). . . . .	49
Figure 4.16	Legend of the classes that derive from the semantic segmentation of the 1 <sup>st</sup> implementation. . . . .	50
Figure 4.17	Differences between the corresponding surfaces of the testing datasets of the CityGML (transparent) and the MVSM (grey) models of the same features. . . . .	50
Figure 4.18	Faces that are removed from the CityGML model. . . . .	51
Figure 4.19	Faces that are chosen from the CityGML model. . . . .	51
Figure 4.20	Result of the step 2 of the implementation of the proposed methodology, after the <i>Heuristic rule 1 - Step 2</i> and <i>Heuristic rule 2 - Step 2</i> (Section 3.3). . . . .	52
Figure 4.21	Characteristics of the facade of the buildings that are avoided of being assigned as uncertain (green), due to the use of the <i>Heuristic rule 2 - Step 2</i> (Section 3.3). . . . .	53
Figure 4.22	Result of the step 2 of the implementation of the proposed methodology, after the <i>Heuristic rule 3 - Step 2</i> (Section 3.3). . . . .	53
Figure 4.23	Unconnected faces assigned as wall class and as uncertain. . . . .	54
Figure 4.24	Result of the implementation of the methodology that is presented in Figure 3.12, in Section 3.3, for the identification of the different connected components in a semantic class of a MVSM model. . . . .	54
Figure 4.25	Result of the step 2 of the 1 <sup>st</sup> implementation of the proposed methodology. . . . .	55
Figure 4.26	Characteristics on the facades of the buildings that are assigned as wall class, while they are not. . . . .	55
Figure 4.27	Example of the distances computed for the quality control of the segmentation of the MVSM faces to the different surfaces of the CityGML model. . . . .	56
Figure 4.28	Projections of the four buildings (roof in red texture and wall in blue texture) to the corresponding plane surfaces (roof and outer walls) of the CityGML model (see Figure 4.19). . . . .	58
Figure 4.29	Example of faces that have wrong orientation due to door recession. . . . .	58
Figure 4.30	Gaps between the projected faces of the MVSM model (red triangles for roof class and blue triangles for wall class) and the plane surface of the CityGML model (grey surfaces). . . . .	59
Figure 4.31	Steps of the projection of the wall class of the 1 <sup>st</sup> building of the testing dataset of the 1 <sup>st</sup> implementation. . . . .	59
Figure 4.32	Projected faces of the MVSM model to the boundary surfaces of the CityGML model (red circles - errors caused by holes on the mesh). . . . .	60

Figure 4.33	Legend of the classes that derive from the semantic segmentation of the 2 <sup>nd</sup> implementation. . . . .	61
Figure 4.34	Faces of the buildings of the CityGML model that correspond to the faces of the MVSM model (2 <sup>nd</sup> Implementation). . . . .	61
Figure 4.35	Result of the step 2 of the implementation of the proposed methodology on the building block in LOD <sub>1</sub> CityGML dataset, after the <i>Heuristic rule 1 - Step 2</i> and <i>Heuristic rule 2 - Step 2</i> (Section 3.3) (2 <sup>nd</sup> Implementation). . . . .	62
Figure 4.36	Result of the step 2 of the implementation of the proposed methodology on the building block in LOD <sub>2</sub> CityGML dataset, after the <i>Heuristic rule 1 - Step 2</i> and <i>Heuristic rule 2 - Step 2</i> (Section 3.3) (2 <sup>nd</sup> Implementation). . . . .	62
Figure 4.37	Result of the step 2 of the implementation of the proposed methodology on the building block in LOD <sub>1</sub> CityGML dataset, after the <i>Heuristic rule 3 - Step 2</i> (Section 3.3) (2 <sup>nd</sup> Implementation). . . . .	63
Figure 4.38	Result of the step 2 of the implementation of the proposed methodology on the building block in LOD <sub>2</sub> CityGML dataset, after the <i>Heuristic rule 3 - Step 2</i> (Section 3.3) (2 <sup>nd</sup> Implementation). . . . .	63
Figure 4.39	Result of the step 2 of the implementation of the proposed methodology on the building block in LOD <sub>1</sub> CityGML dataset, after the correction of the connected components (Section 3.3) (2 <sup>nd</sup> Implementation). . . . .	64
Figure 4.40	Result of the step 2 of the implementation of the proposed methodology on the building block in LOD <sub>2</sub> CityGML dataset, after the correction of the connected components (Section 3.3) (2 <sup>nd</sup> Implementation). . . . .	64
Figure 4.41	Roof of the building B <sub>9</sub> of the MVSM model and the LOD <sub>1</sub> CityGML model (transparent box). . . . .	66
Figure 4.42	Result of the step 4 of the 2 <sup>nd</sup> implementation on the LOD <sub>1</sub> CityGML model. . . . .	66
Figure 4.43	Result of the step 4 of the 2 <sup>nd</sup> implementation on the LOD <sub>2</sub> CityGML model. . . . .	66





## LIST OF TABLES

Table 1.1	Comparison of CityGML and MVSM data model formats (Table from <a href="#">Kavisha and Saran (2015)</a> ). . . . .	3
Table 2.1	Bidirectional enrichments of a CityGML model and a MVSM model. . . . .	12
Table 3.1	Categorisation of the quality control of the semantic segmentation, based on the Hausdorff distance computations. . . . .	32
Table 4.1	Statistics (number of vertices, area) of the different datasets that are used for the implementation of the proposed methodology. . . . .	40
Table 4.2	IDs of the buildings of the CityGML testing dataset used on the 1 <sup>st</sup> implementation of the proposed methodology (see Figure 4.6). . . . .	43
Table 4.3	IDs of the nine buildings of the CityGML testing dataset used on the 2 <sup>nd</sup> implementation of the proposed methodology (see Figures 4.7 and 4.8). . . . .	45
Table 4.4	Non-manifold errors that appear in the testing datasets of the MVSM model. . . . .	48
Table 4.5	Categorisation of the quality control of the semantic segmentation, based on the Hausdorff distance computations. . . . .	57
Table 4.6	Indications of errors that are related to the shape of the roofs and the absence of buildings in the CityGML model. . . . .	57
Table 4.7	Categorisation of the quality control of the semantic segmentation, based on the Hausdorff distance computations (2 <sup>nd</sup> implementation). . . . .	65
Table 4.8	Indications of errors that are related to the shape of the roofs and the absence of buildings in the CityGML model (2 <sup>nd</sup> implementation). . . . .	65



## ACRONYMS

<b>AHN</b>	Actual Height model of the Netherlands
<b>BAG</b>	Basic register for Addresses and Buildings
<b>BIM</b>	Building Information Modelling
<b>CAD</b>	Computer - Aided Design
<b>CGAL</b>	Computational Geometry Algorithms Library
<b>CityGML</b>	City Geography Markup Language
<b>COLLADA</b>	COLLABorative Design Activity
<b>CPA</b>	Closest Point Approach
<b>CRS</b>	Coordinate Reference System
<b>DSM</b>	Digital Surface Model
<b>DTM</b>	Digital Terrain Model
<b>EPSG</b>	European Petroleum Survey Group
<b>GIS</b>	Geographic Information System
<b>GML</b>	Geography Markup Language
<b>GNSS</b>	Global Navigation Satellite System
<b>HOG</b>	Histogram of Oriented Gradient
<b>HTML</b>	HyperText Markup Language
<b>ID</b>	IDentity
<b>IFC</b>	Industry Foundation Classes
<b>ISO</b>	International Organization for Standardization
<b>KML</b>	Keyhole Markup Language
<b>LBS</b>	Location Based Service
<b>LIDAR</b>	LIght Detection And Ranging
<b>LOD</b>	Level Of Detail
<b>MCMC</b>	Markov Chain Monte Carlo
<b>MVS</b>	Multi-View Stereo
<b>MVSM</b>	Multi-View Stereo Mesh
<b>OGC</b>	Open Geospatial Consortium
<b>RANSAC</b>	RANdom SAmples Consensus
<b>RD</b>	RijksDriehoekstelsel

<b>SDIs</b>	Spatial Data Infrastructures
<b>SfM</b>	Structure-from-Motion
<b>TLS</b>	Terrestrial Laser Scanning
<b>UAV</b>	Unmanned Aerial Vehicle
<b>UML</b>	Unified Modelling Language
<b>X3D</b>	Extensible 3D
<b>XML</b>	Extensible Markup Language



The generation of 3D city models that represent the urban landscape is increasing due to the evolution of technology and the need for information in order to provide solutions on issues that correspond to the building environment. Examples of applications that use 3D city models are the estimation of the solar irradiation (Fath et al., 2015), the urban planning (Lu and Wang, 2014) and the emergency response (Tashakkori et al., 2015). A list of identified use cases of 3D city models and their applications, is thoroughly presented by Biljecki et al. (2015), who segmented and categorised the diverse uses of 3D city models, based on a theoretical reasoning. The process of the creation of 3D city models includes the acquisition of the data, the data storage, the analysis, the visualization and the quality control of each stage (Lemmens, 2011). The use of different methods, especially in the data acquisition part, generates 3D city models that contain different characteristics, such as geometric, semantic or topological aspects. Depending on the different advantages and drawbacks that these models have, they are used for different purposes. For instance, the City Geography Markup Language (CityGML) models provide an incorporation of geometric, topological and semantic information, but they are not optimized with respect to efficient visualization (Kolbe, 2009), whereas the Multi-View Stereo Mesh (MVSM) models focus on data visualisation, exchange of geometric data and lack semantic information. In addition, the most important characteristic of MVSM models is their automated creation, where in short time, they provide a large amount of geometric information (Fuhrmann et al., 2014). These differences create difficulties in spatial analysis, because the lack of information on a 3D city model might be of extreme importance for the computation of spatial relationships that are needed for an application. For example, during the use of augmented reality systems for navigation, gaming, sightseeing and other applications, the lack of either geometric data, appearance or semantic information of a 3D city model, lead to fruitless results. In such cases, the solution is to supplement a 3D city model with the additional data that is needed.

The idea of this thesis is to use additional data that can be found into different existing formats of 3D city models of the same area, representing the same features. More specifically, the aim of this thesis is to explore the bidirectional enrichments and the enrichment methodologies that can be conducted between a CityGML model and a MVSM model of the same entities (see Figure 1.1). On this specific field no related work has been found, but some closely related interesting approaches can be used, like for instance the work from Verdie et al. (2015) and Nguatem et al. (2016), who investigated the generation of CityGML models, based on information that is provided from MVSM models. After some research on the methodologies that can be used for the bidirectional enrichment of these two different 3D city model formats, an automatic bidirectional enrichment methodology is proposed and is implemented on a testing area, provided in the two different 3D city model formats. This method is based on distance computations between the meshes of the two 3D city models, used to match corresponding features, or part of features, in order to segment semantically the MVSM model into different thematic classes (roof, wall, road, terrain, uncertain) and to transfer texture to the CityGML model. Furthermore, distance computations are performed for the validation of the shape of the roofs and the absence of buildings in the CityGML model, with respect to the information given from the MVSM model. The achievement of using existing information of different formats of 3D city models of the same entities can supplement each 3D city model format with useful information and as a final goal, it can enhance the performance of spatial analysis.

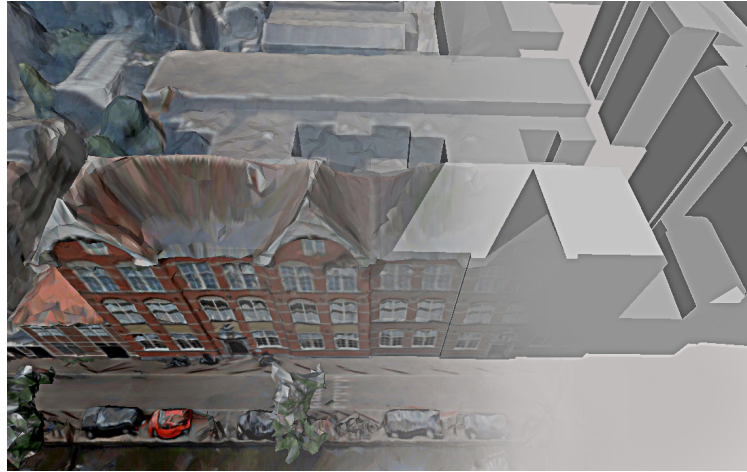


Figure 1.1: Different characteristics between a MVSM (left) and a CityGML (right) model of the same area, representing the same features.

## 1.1 CHARACTERISTICS OF DIFFERENT 3D CITY MODEL FORMATS

The 3D city models can vary from the data that is used for their creation to the representations that they provide, their modelling aspects, their storage and to their file formats. Different geo-data collection technologies and methods from photogrammetry, remote sensing and surveying can be used for their creation, such as Global Navigation Satellite System (GNSS), mobile Geographic Information System (GIS) and Location Based Service (LBS), Terrestrial Laser Scanning (TLS), Photogrammetry and Airborne Light Detection And Ranging (LIDAR) (Kolbe et al., 2009; Lemmens, 2011). 3D city models can represent different features of the real world, such as relief, bridges, tunnels, buildings, vegetation, transportation bodies, water bodies, etc. (Kolbe, 2009). Moreover, 3D city models are characterised by different aspects, including multi-scale representation, semantics, geometry, topology, appearance, spatio-semantic coherence and extensibility (Kolbe, 2009). Furthermore, they can be stored in different ways, either file-based or in a database. In addition, the formats of the 3D city models vary to serve GIS, Building Information Modelling (BIM), Computer - Aided Design (CAD), computer graphics or Web domains, according to their developers (Zlatanova et al., 2012). The different domains provide their own standards for the representation and the exchange of 3D city models. The components of the 3D city models are encoded by common file and exchange formats for 2D raster-based GIS data (e.g. GeoTIFF), 2D vector-based GIS data (e.g. AutoCAD DXF) and 3D models (e.g. CityGML, OBJ, COLLaborative Design Activity (COLLADA), Keyhole Markup Language (KML), Industry Foundation Classes (IFC)). Some formats have been developed as standards by international organizations, while others have been accepted as standards due to their wide use by users and software (Zlatanova et al., 2012).

The two different 3D city models that are used during the research of this thesis are the CityGML model and the MVSM model. The first one is defined by the Open Geospatial Consortium (OGC) as an explicit Extensible Markup Language (XML) - based exchange format for 3D city models, because apart from the geometric data, it can also exchange semantic and topological aspects in a well-defined way (Gröger et al., 2012). Therefore, CityGML models are used in a wide range of applications (Kolbe et al., 2009; Biljecki et al., 2015). On the other hand, the use of MVSM models has increased the last decade due to the fact that the technology behind their creation has evolved and it can provide creations of 3D city models with high amount of geometric and texturing information, in short time (Furukawa et al., 2015). MVSM models are usually provided in a textured triangulated mesh (see Figure 1.1), in different file formats, such as COLLADA and OBJ. The basic different characteristics of these two 3D city model formats are listed in the following table:

Standard/ Criteria	CityGML	MVSM
<b>Geometry</b>	B-rep	Mesh
<b>Semantics</b>	Rich in semantics	No semantics
<b>LOD</b>	5 discrete LODs	Not supported
<b>Texture</b>	Basic support	Rich textures
<b>Web rendering</b>	Supported	Supported
<b>Construction</b>	Semi-automatic	Fully automatic

Table 1.1: Comparison of CityGML and MVSM data model formats (Table from Kavisha and Saran (2015)).

At this point, it has to be added that thoughts on making the CityGML, the standard for 3D representation and exchange of MVSM models have recently arisen (Kolbe et al., 2009). The reason for that is that CityGML models have a flexibility on the structure of the thematic information following the modelling paradigm defined by the International Organization for Standardization (ISO) 191xx standards family. It can be easily realized that both 3D city models are widely used and they provide different characteristics for same entities (see Table 1.1). Therefore, ways of enrichments and automatic enrichment methodologies between the characteristics of the two mentioned 3D city models are of great importance and are still on research.

## 1.2 RESEARCH QUESTIONS AND SCOPE

### 1.2.1 Research questions

For this thesis, the main research question is:

*To what extent can a CityGML model and a MVSM model of the same area, representing the same features, be automatically and bidirectionally enriched?*

The bidirectional enrichment of the two different 3D city models is the main aim of this thesis. The sub-questions that are investigated in order to meet the expectations of the final goal, can be subdivided into three different parts. The first three sub-questions (Sub-questions 1-3) are related to a theoretical part of this research. Next, several sub-questions are investigated for the achievement of a practical part in this thesis that is related to the enrichments of a MVSM model with information from the use of a corresponding CityGML model (Sub-questions 4-7) and to the enrichments of a CityGML model with the use of a corresponding MVSM model (Sub-questions 8-9).

1. What is the concept of bidirectional enrichment?
2. What kind of enrichments are theoretically possible between a MVSM model and a CityGML model?
3. What enrichment methodologies can be performed for the automatic bidirectional enrichments of a MVSM model and a CityGML model of the same features?
4. To what extent can a MVSM model be segmented semantically by matching its faces with the faces of the building elements (roof, wall) and the city object classes of relief features (terrain) and transportation bodies (roads) of a CityGML model?
5. How to transfer and assign attributes from a CityGML model to a MVSM model?
6. How to segment a MVSM model into different buildings?
7. Is it possible to perform other operations like validation and to detect errors in either models, based on bidirectional matches?
8. How to automatically texture the buildings of a CityGML model using the texture from a MVSM model?

### 1.2.2 Scope of research

In this thesis, the concept of bidirectional enrichment between two 3D city models is analysed and the enrichments that can be conducted theoretically between a [CityGML](#) and a [MVSM](#) model are presented. The bidirectional enrichments that are theoretically proposed between the two 3D city models, are limited to the building features of a 3D city model. The other thematic classes, such as vegetation, water bodies etc. are not the core of this research. For the completeness of the list of the enrichments that can be done between the two 3D city models, the level of detail 1, level of detail 2 and level of detail 3 are analysed and not the level of detail 0 and level of detail 4, since there are not relevant here ([Biljecki et al., 2014](#)). One important objective of this thesis is the exploration of the state-of-the-art different enrichment methodologies that can be used between two 3D city models.

Then, a proposed automated enrichment methodology, based on distance computations between the two meshes of the 3D city models is tested for the transfer of aspects from the one 3D city model to the other one, and conversely. The proposed methodology is firstly tested for four buildings of an area in Amsterdam and then, for a building block (nine buildings) of another area in Amsterdam, together with its surroundings in order to test this method to more city object classes, including relief features (terrain) and transportation bodies (road). The proposed methodology is performed with the use of the same building block in the two different 3D city model formats, the [MVSM](#) model and the [CityGML](#) model. The methodology is tested on this building block twice, each time with the use of a different [CityGML](#) level of detail, at the beginning with [CityGML](#) level of detail 1 and then, with [CityGML](#) level of detail 2. The results are compared according to the different information that these two [CityGML](#) level of details provide.

During this research, the execution time of the proposed methodology is not examined. This thesis does not deal with data acquisition and the raw datasets of the two 3D city models are used, where no further cleaning, corrections, simplifications or smoothness are performed. More specifically, according to the proposed automated enrichment methodology, the enrichment of the [MVSM](#) model from the [CityGML](#) model is focused on the semantic segmentation of the mesh into five categories, roof, wall, road, terrain and uncertain. The segmentation that derived from the matching of the faces of the two 3D city models, is validated with the use of distance computations between the meshes of the two 3D city models. On the other direction, the improvements of the [CityGML](#) model from the [MVSM](#) model include the transfer of the texture to the buildings of the [CityGML](#) model and the validation of the buildings into the [CityGML](#) model. During the validation, the absence of buildings and the shapes of the roofs of the buildings in the [CityGML](#) model are investigated with the use of the younger [MVSM](#) model of the same area, based on distance computations between the two meshes of the 3D city models. After these validations, indications are provided for the [CityGML](#) model concerning, the absence of buildings and the shapes of the roofs comparing to the information derived from the [MVSM](#) model.

## 1.3 RESEARCH CONTRIBUTIONS

As it is already mentioned in Section 1, the use of 3D city models, like the [CityGML](#) and the [MVSM](#) models, has increased the last decades due to the evolution of technology, however it is important to notice that different 3D city model formats can contain different aspects of the same entities. The gap that is missing is to record the possible enrichments that can be done bidirectionally between these two different 3D city models, when both exist for the same entities. Moreover, the exploration of possible automated enrichment methodologies that will enrich each 3D city model automatically with information from the other one, will boost the evolution on the use of such models on real life applications. The bidirectional enrichments between a [CityGML](#) model and a [MVSM](#) model that are recorded in this thesis, together with the proposed enrichment methodology, are not related directly to society, but they are mostly focused on the scientific community. Researchers need to use this knowledge to extend the list of the enrichments that can be conducted between these two widely used 3D city models. In addition, tests, improvements and extensions of all the state-of-the-art methodologies that are proposed need to be performed in the future, in order to reach a new strategy on the interoperability of data between different existing 3D city models.

## 1.4 THESIS OUTLINE

In Chapter 2, the theoretical background of the CityGML model and the MVSM model, is analysed. Furthermore, the concept of bidirectional enrichment is presented and the bidirectional enrichments that can be conducted between the two 3D city models, are listed. Moreover, the related work in the frame of methodologies that can be used for the achievement of bidirectional enrichments between a CityGML and a MVSM model, are discussed.

All the aspects of the proposed methodology that is based on distance computations between the meshes of the two different existing 3D city models, are provided in Chapter 3.

The implementation of the proposed methodology is presented in Chapter 4. In this chapter, the tools that are used, including the programs, the packages that are used in python and the algorithms that are created for this methodology, are specified. The data that is used during the implementation, is analysed and the results of the different enrichments per direction, are given.

In Chapter 5, the summary of this research and the answers to the research question and sub-questions are explained. Discussions and conclusions on this thesis, as well as recommendations for future work are provided.



# 2 | THEORETICAL BACKGROUND

Before taking a dive into the proposed methodology and the results of its implementation, concerning the automatic bidirectional enrichment of a [CityGML](#) and a [MVSM](#) model, the theoretical background associated with these concepts is analysed in this chapter. Firstly, the theories related to the [CityGML](#) and the [MVSM](#) model are described. Then, the concept of the bidirectional enrichment between different 3D city models, together with the possible enrichments between a [CityGML](#) and a [MVSM](#) model, are presented. The related work on the field of enrichment methodologies between 3D city models that is linked with the aim of this thesis, is addressed.

## 2.1 CITYGML MODEL

The background of the [CityGML](#) data models is the urban information modelling. The [CityGML](#) data models are semantic 3D city models and they have been developed to provide data homogeneity and to help the data integration. [CityGML](#) version 2.0 is an [XML](#)-based, open data model and an application schema for the Geography Markup Language ([GML](#)) version 3.1.1 ([GML3](#)) ([Gröger et al., 2012](#)). [GML3](#) is the extensible international standard issued by the [OGC](#) and the [ISO/TC211](#), for the storage and the exchange of virtual 3D city models ([Gröger et al., 2012](#)). It provides an incorporation of geometric, topological and semantic information. An example of such a model is shown in [Figure 2.1](#).

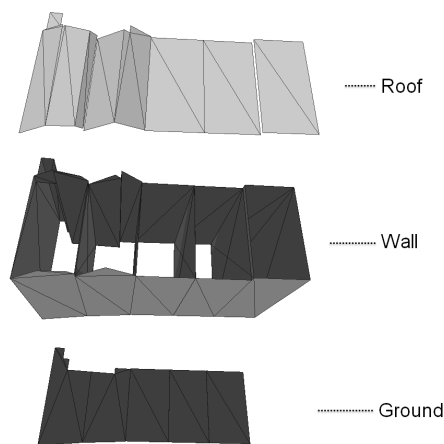


Figure 2.1: Example of a CityGML model (LOD2).

The aim of the development of the [CityGML](#) model is to reach a common definition of the basic entities, attributes and relations of a 3D city model, in order to provide the reuse of the same data for different applications and to aim at the development of Spatial Data Infrastructures ([SDIs](#)) ([Gröger et al., 2012](#)). Nowadays, the need for a semantic 3D city model is related to the need for computer analyses and simulations. Corresponding analyses are important for several applications, such as security scenario, urban planning, energy assessment for improvements of buildings or insulations, emergency purposes, navigation systems, evacuation, etc. ([Biljecki et al., 2015](#)). In these cases, a more analytic answer to these questions is needed. Therefore, the notion of e.g. a roof, a wall, a ground, a window, needs to be modelled into a 3D city model. The [CityGML](#) model contains different modelling aspects, including multi-scale representation, semantics, geometry, topology, appearance, spatio-semantic coherence and extensibility



(Kolbe, 2009). Next, the multi-scale representation, the semantic and the appearance aspects of a CityGML model are presented in detail.

One of the aspects of CityGML models is the multi-scale modelling (5 Level Of Detail (LOD)). LODs are required to reflect independent data collection processes with different application requirements (see Figure 2.2). Buildings may be represented in LOD0 by footprint or roof edge polygons. LOD1 is a block model comprising prismatic buildings with flat roof structures. Furthermore, a building in LOD2 has differentiated roof structures and thematically differentiated boundary surfaces. LOD3 denotes architectural models with detailed wall and roof structures, potentially including doors and windows. LOD4 completes a LOD3 model by adding interior structures for buildings. For example, buildings in LOD4 are composed of rooms, interior doors, stairs, and furniture (Gröger et al., 2012; Gröger and Plümer, 2012).



Figure 2.2: The five LODs defined by CityGML (Figure from Biljecki et al. (2016)).

A CityGML model is characterized by the semantic information that it contains in a well-structured way, for the different 3D objects. The semantic model of CityGML employs the ISO 19100 standards family framework (Gröger et al., 2012). The different geographic features of the real world are modelled by classes, which are formally specified using Unified Modelling Language (UML) notation. The geographic features that are supported in a CityGML model are divided into different city object classes, like buildings, transportation bodies, water bodies, vegetation, etc. The most important city object classes in UML notation are depicted in Figure 2.3, which according to Kolbe (2009), they are the top level class hierarchy of CityGML.

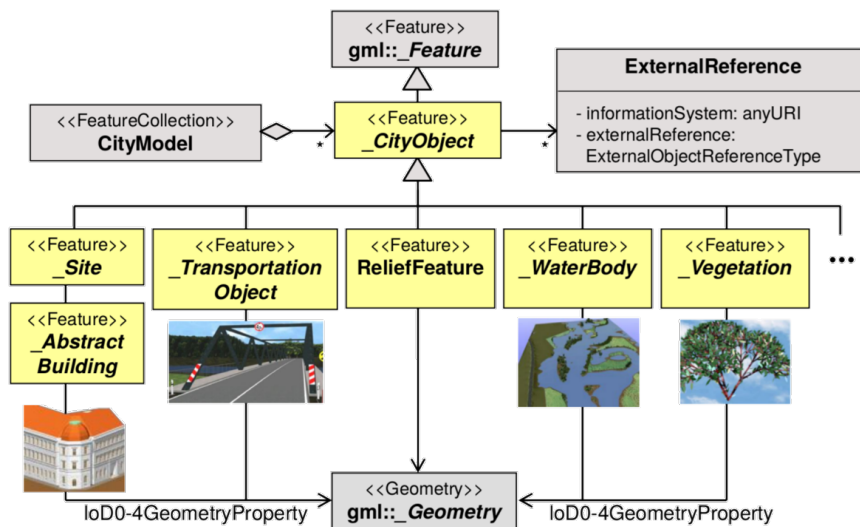


Figure 2.3: UML diagram of the top level class hierarchy of CityGML (Figure from Kolbe (2009)).

The geometry model of GML3 has an aggregation structure, where all feature classes have thematic attributes and relations to different geometry classes. For example, the *Abstract Building* class can be divided into *Buildings* and *Building Parts*. Also, there is a superclass linked with the *Abstract Building* class that is related to the surface boundary (*BoundarySurface*) of the semantic objects. In this case, the lowest aggregation level of CityGML refers to surfaces like roof (*RoofSurface*), wall (*WallSurface*), ground (*GroundSurface*), etc. (Kolbe et al., 2009). Furthermore, it is important to notice that there are also explicit thematic features, for instance windows and doors, which are related only to the LOD3 and the LOD4, and others like rooms, which are related only to the LOD4 (Gröger et al., 2012).



Another important modelling aspect of *CityGML* is the appearance, where the characteristics of the surfaces are captured as they appear to specific sensors (RGB and infrared cameras). The appearance is linked with the surface materials and textures that apart from the visualisation, they are also used for analysis tasks. It has to be noted that the texture images and the material definitions are adopted from the Extensible 3D (*X3D*) and the *COLLADA* file formats (Kolbe, 2009; Gröger et al., 2012). *X3D* and *COLLADA* are file formats for 3D representations in graphics software applications.

The production of *CityGML* models is usually based on GIS data, including Digital Surface Model (*DSM*), Digital Terrain Model (*DTM*), 2D building ground plans, aerial and terrestrial imagery (Kada and McKinley, 2009; Moreira et al., 2013; Elberink et al., 2013). The different data used for the creation of a *CityGML* model vary according to the different *LOD*. Each one of these data can provide all the different needed information. For example, the *DSM* is used for the calculation of the roof types, while the *DTM* is used for the calculation of the base point of the buildings. Moreover, the positions of the buildings are provided from the 2D building footprints, which are closed polygons with unique identifiers. Imagery is used to gain the knowledge on the appearance information, while for visual inspection and quality control of the creation of the *CityGML* model, digital orthophotos can also be used (virtualcitySystems, 2016).

According to Kolbe (2009) "*CityGML* establishes a bindingness or liability of the producer with respect to potential users or customers on the other side".

An important drawback of these models is the amount of work that needs to be done for their creation. Also, *CityGML* is not optimized with respect to efficient visualization (Kolbe, 2009). However, the semantic information given by the explicit association of *CityGML* objects to thematic classes and the provision of thematic attributes, can be exploited to filter objects and to create 3D graphical shapes, appearance properties and materials accordingly (Kolbe, 2009).

## 2.2 MVSM MODEL

*MVSM* models are 3D representations of an object or a scene created exclusively from imagery with the use of photogrammetry. The techniques that are used are based on multi-view algorithms, which have recently been optimized. Therefore, within the last decade their use from a wide range of applications has increased (Furukawa et al., 2015). During the creation of a *MVSM* model, the input is a set of images of the object that is modelled and the output is a digital representation of the environment in the form of a textured triangle mesh (Jonsson, 2016). An example of a *MVSM* model is illustrated in Figure 2.4.

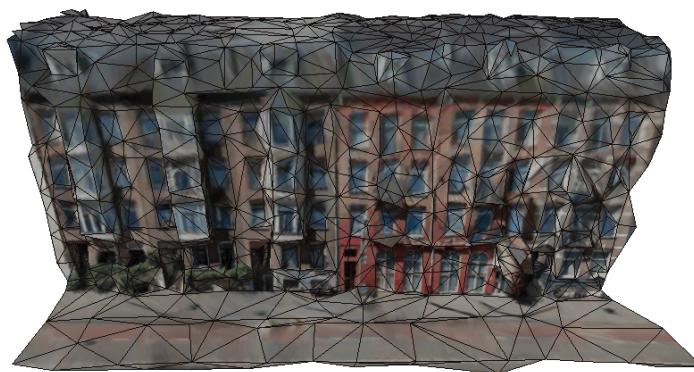


Figure 2.4: Example of a *MVSM* model.

The most important characteristic of *MVSM* models is their automated creation, where in short time, they provide a large amount of geometric information. The process of the mesh generation consists of the image acquisition step, three algorithmic steps for the achievement of the surface reconstruction and the final step is the texturing of the surface of the 3D city model (see Figure 2.5).

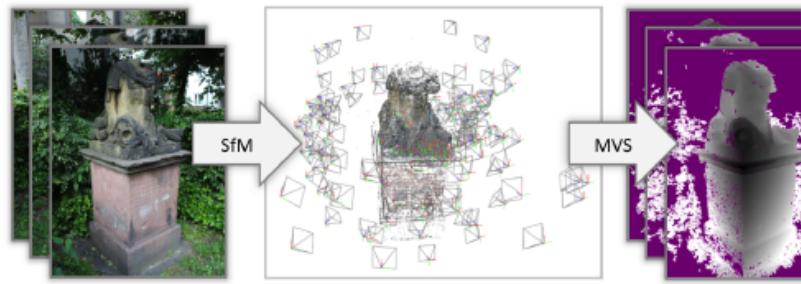


Figure 2.5: Steps of the multi-view reconstruction (Figure from Fuhrmann et al. (2014)).

Firstly, a large set of images with high overlap is used to obtain high quality results for the mesh generation and the texturing. The camera is repositioned for every image, because parallax is required for the triangulation (Fuhrmann et al., 2014). The images can be acquired either by aerial or terrestrial imagery. Cameras are mounted on a specially designed plane, like an aircraft, helicopter, Unmanned Aerial Vehicle (UAV) or supported by a ground-based structure, like a car.

After the image collection, Structure-from-Motion (SfM) algorithms are used for the reconstruction of the parameters of the cameras and for the creation of a sparse set of points of the object (Szeliski, 2010; Wu, 2013).

Next, Multi-View Stereo (MVS) algorithms are used for the reconstruction of a dense 3D geometry, by finding visual correspondences in the images using the estimated camera parameters (Vu et al., 2012).

Furthermore, the surface reconstruction takes place, where the points of the dense point cloud are connected for the creation of the surface of the object (Wachter et al., 2014; Furukawa et al., 2015; Jonsson, 2016).

The final step of the multi-view reconstruction process is the texturing of the surface of the 3D object that is provided from the correspondent images that have been acquired (Wachter et al., 2014). The texturing is created from a UV mapping process, which requires three basic steps, the unwrapping of the mesh, the creation of the texture, and the application of the texture (Mullen, 2011).

After the surface reconstruction of the object, the mesh of the 3D object is consisted of vertices and faces with texture. According to Figure 2.6, for each vertex (A, B or C) of each face ( $f_i$ ) in the mesh, UV coordinates, known as texture coordinates, are generated. These UV values correspond to a UV coordinate spot into a related textured image. Therefore, the combination of the three texture coordinates (a, b, c) of the three vertices of a face, correspond to a face inside the texture image (Murdock, 2008).

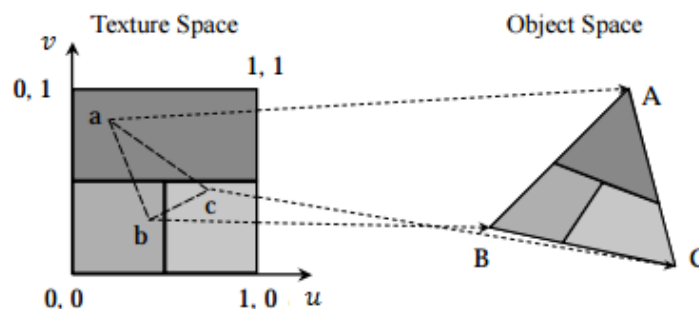


Figure 2.6: Texture mapping (Figure from Angel et al. (2007)).

The texture that MVSM models acquire automatically from the images, is a very important aspect that cannot be easily obtained by other methods. On the other hand, it is important to notice that MVSM models do not contain attributes and semantic information. Therefore, the many triangles of a mesh model cannot be easily distinguished to trees, buildings, streets etc. MVSM models are useful for computer analyses and simulations on visibility-based analysis, where for example shadows are used for solar potential analysis. However, computer analyses and simulations on the basis of virtual 3D city models that require meaningful objects with spatial and thematic properties, cannot be easily realized.

## 2.3 BIDIRECTIONAL ENRICHMENT AND STATE-OF-THE-ART IN 3D CITY MODELS

In this section, the concept of the bidirectional enrichment between two existing 3D city models, is presented and the enrichments that can be conducted, specifically between [CityGML](#) and [MVSM](#) models, are listed in Section 2.3.1. Then, related work referring to methodologies that have been used for such an attempt, plus the ones that have some close relation with the proposed methodology and can be used in the future as enrichment methodologies, are analysed in Section 2.3.2.

### 2.3.1 Bidirectional enrichment

In this section the concept of bidirectional enrichment is introduced (1<sup>st</sup> sub-question from Section 1.2.1) and the possible enrichments between [CityGML](#) and [MVSM](#) models are presented (2<sup>nd</sup> sub-question from Section 1.2.1).

With the use of the concept of bidirectional enrichment, it is meant the either transfer or update of aspects that might or might not exist from one existing 3D city model to another existing 3D city model of the same entities, and conversely. The characteristics of the 3D city models can vary to their modelling aspects, their storage, their file formats etc. (Section 1.1) and these differentiations may affect the aspects and the methods that can be used on different bidirectional enrichments.

As it is already mentioned, this thesis focuses on the bidirectional enrichment of two different 3D city models. The two 3D city models that are explored, are the [CityGML](#) model (Section 2.1) and the [MVSM](#) model (Section 2.2). The different characteristics of the models that are listed in Table 1.1 (Section 1.1) and the limitations of this thesis that are mentioned at the scope of the research (Section 1.2.2), are taken into consideration. The focus of the possible bidirectional enrichments, during this research, is specifically on the enrichments that can be conducted on the buildings between the 3 LODs, LOD1, LOD2 and LOD3 of a [CityGML](#) model and a [MVSM](#) model. Examples of these 3D city models are illustrated in the figure below:

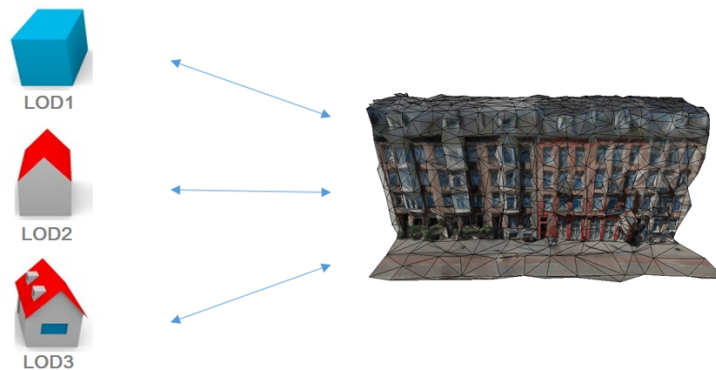


Figure 2.7: Illustrations of the 3D city models that are explored on their bidirectional enrichments.

The following Table 2.1 is created for this thesis, in order to categorize the enrichments that can be done between a [MVSM](#) model and a [CityGML](#) model. The different categories are decided by combining the knowledge of the characteristics of the two different 3D city models (Sections 2.1 and 2.2) and the information from papers that apply similar enrichments on these two models. These papers are presented in Section 2.3.2.

CityGML model to MVSM model	MVSM model to CityGML model
Semantic segmentation	Texturing
Transfer attributes	Validation
Straightening mesh	Update
Simplification	Creation of 3D CityGML model

Table 2.1: Bidirectional enrichments of a CityGML model and a MVSM model.

According to the previous table, the enrichments that can be provided to a MVSM model with the use of information from a CityGML model are the semantic segmentation, the transfer of attributes, the straightening and the simplification of the mesh. On the other direction, with the use of information from existing MVSM models to existing CityGML models, texturing, validation and update of information, plus LOD generation can be provided to the related CityGML model. It is important to notice that in this section, each one of these enrichments is analysed, but methodologies for their achievement and related applications are presented in Section 2.3.2.

**SEMANTIC SEGMENTATION:** Semantic segmentation means the split of the soup of polygons that a MVSM model has. As far as it concerns the building class, the mesh can be separated into faces that correspond to the different aspects of a building, like roof, wall, windows, etc. In this case, during the semantic segmentation of a building in a MVSM model, the related building from the CityGML model is compared. It is obvious that the higher the LOD of the compared CityGML model, the higher the quality of the segmentation of the mesh of the MVSM model that can be conducted.

**TRANSFER ATTRIBUTES:** The CityGML model usually contains multiple information and aggregated structures related to the different attributes of the objects of a 3D city model. For example, age, owners, volume, etc. After finding a correspondence between the faces of the two 3D city models, then the different attributes that are included into the CityGML model can be transferred to the corresponding mesh of the MVSM model. A higher LOD of a CityGML model contains more information than a lower one. Therefore, more attributes can be transferred from a LOD3, where all the information of the building shell is structured, than from a LOD1, which has only the information related to the walls and the planar representations of the roof structures.

**STRAIGHTENING:** The mesh of a MVSM model contains all the geometric information of the facades of a building. Uneven mesh can appear on the parts of the mesh of a MVSM model due to noise caused from the surroundings, during the image acquisition step. Therefore, the straightening of the mesh is a process that usually needs to take place. An example of a MVSM model before and after the straightening of the mesh is illustrated in the following figure:

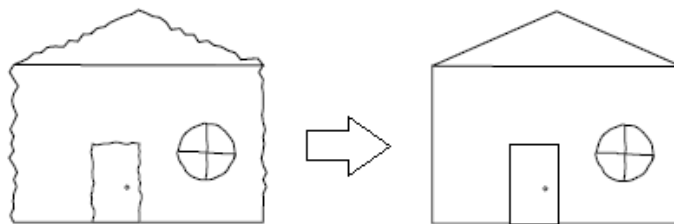


Figure 2.8: Straightening of a mesh of a MVSM model (Figure from Jonsson (2016)).

The straightening of a MVSM model can be conducted with the use of the plane surfaces of the CityGML model on the related faces of the mesh (Jonsson, 2016). The different LODs can provide different quality of the result. The use of a LOD1 CityGML model can provide really good results on the faces of the MVSM model that correspond to the wall surface. A LOD2 CityGML model, apart from the faces that correspond

to the wall surface, it can provide much better results on the straightening of the faces that correspond to a roof surface. Furthermore, the use of a [LOD3 CityGML](#) model, where all the information of the building shell, such as windows and doors, are included, can provide much better results on the straightening of the mesh, especially on the parts of the windows, the doors and the roof surfaces.

**SIMPLIFICATION:** Apart from the straightening of the mesh of the [MVSM](#) model, a simplification of the mesh of the [MVSM](#) model is usually needed. An example of the simplification of a mesh is illustrated in [Figure 2.9](#).

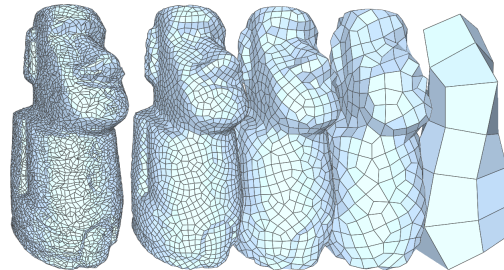


Figure 2.9: Simplification of a mesh of a MVSM model (Figure from [Tarini et al. \(2010\)](#)).

It is known that a [MVSM](#) model consists of a soup of polygons. Some of the polygons, which correspond for instance to surfaces that are on the same plane [CityGML](#) surface and that are related to the same feature, with no differences on their attributes, can be simplified into less and larger triangles. This can lead on the reduce of the information and the enhancement of the computational performance during analysis processes.

**TEXTURING:** The texturing is related to the transfer of texture from the [MVSM](#) model to the [CityGML](#) model. Such an enrichment can provide important information to the [CityGML](#) model for visualization and analysis purposes. It has to be noticed that the use of a higher [LOD](#) can affect the results on the texturing, due to the fact that the matching of the corresponding parts of the texture images can have a better correspondence to the more detailed faces of a building shell, provided by a [LOD3 CityGML](#) model, than by a [LOD1 CityGML](#) model.

**VALIDATION:** A validation of an existing [CityGML](#) model with the use of information from an existing [MVSM](#) model, younger than the [CityGML](#) model, can have many forms. Validation can be conducted on the existence of buildings inside the [CityGML](#) model scene. In this way, the areas with high likelihood of missing buildings can be re-examined. Validation can also be performed for the heights of the buildings in the [CityGML](#) model. A building might have changed, thus an additional or missing storey can be identified. Also, the validation can be conducted for the shape of the roofs of the buildings between a [MVSM](#) model and a [LOD2 CityGML](#) model, which contains this kind of detailed information. Moreover, a validation of a [CityGML](#) model of [LOD3](#), with information from a [MVSM](#) model, can be done for the validation of the shape and existence of windows and doors, together with all the other aspects that are mentioned above.

**UPDATE:** All the aspects that can be validated with the use of [MVSM](#) models in each different [LOD](#) in the previous paragraph, can be updated in the corresponding [CityGML](#) model.

**CREATION OF 3D CITYGML MODEL:** The [MVSM](#) model contains multiple geometric information for a building, concerning the roofs, the walls, the windows, etc. Each information of the [MVSM](#) model and the corresponding existing [CityGML](#) model, is important for the enrichment of the existing [CityGML](#) model and the creation of a [CityGML](#) model of a higher [LOD](#) of the same features. The cases that are explored are the enrichment of a [LOD1 CityGML](#) model for the creation of a [LOD2 CityGML](#) model and the enrichments of either a [LOD1](#) or [LOD2 CityGML](#) model for the creation of a [LOD3 CityGML](#) model.

- *LOD2 CityGML model creation:* The creation of a [LOD2 CityGML](#) model can be conducted with the use of an existing [LOD1 CityGML](#) model and an existing [MVSM](#) model of the same features. The main



difference between a LOD1 and a LOD2 CityGML model is the fact that a building in LOD2 CityGML has differentiated roof structures, while a building in LOD1 CityGML is presented by a block model comprising prismatic buildings with flat roof structures (Section 2.1). Therefore, for the creation of a LOD2 CityGML model by enriching the existing LOD1 CityGML model, the roof information is extracted from the MVSM model. In this way, the roof structures and the new heights of the walls of the buildings are updated to the existing LOD1 CityGML model. Several methodologies, like the use of predefined shapes of the roof structures, creation and comparison of planes or probabilistic methods (Section 2.3.2) can be combined in order to achieve this action.

- *LOD3 CityGML model creation:* The creation of a LOD3 CityGML model can be done with the enrichment of either a LOD1 or LOD2 CityGML model and the use of the corresponding MVSM model. The LOD3 CityGML denotes architectural models with detailed wall and roof structures, potentially including doors and windows (Section 2.1).

In case of using an existing LOD1 CityGML model, the creation of a LOD2 CityGML model is taking place, as it is already mentioned above (*LOD2 CityGML model creation*). Then, the creation of a LOD3 CityGML model can be conducted with the enrichment of the LOD2 CityGML model with the windows and the doors on the building shells. This information can be derived from the exploration of the LOD2 CityGML model and the corresponding MVSM model. The enrichments of a CityGML model with window and door information that can be extracted from the corresponding MVSM model, in order to achieve the creation of a LOD3 CityGML model, are thoroughly analysed below:

- *Window enrichments:* The MVSM model includes all the needed information that is related to the windows of the building shell. It is important to notice that the windows usually protrude from the wall surfaces of a building. In addition, window protrusions or recessions can exist in different heights of a building. Moreover, their basic characteristic that can distinguish them from the doors protrusions or recessions is that windows usually do not touch the ground level of a building and the windows of the same buildings usually have the same shape and size. Therefore, with the use of the surfaces of the existing CityGML model and the detailed information given from the corresponding MVSM model, several methodologies can be performed, like planes comparison, probabilistic approaches and heuristic rules (Section 2.3.2), in order to achieve the enrichments of the windows to the existing CityGML model for the creation of the LOD3 CityGML model.
- *Door enrichments:* Information about the doors of the buildings is provided in the MVSM model. The basic characteristics of the doors are their size and the fact that doors usually touch the ground surface of a building. Furthermore, most of the times the doors appear once on a building facade and they are recesses on the wall surfaces of a building, comparing to the windows that usually protrude from the wall surfaces of a building. The extraction of the door information from the MVSM model and its enrichment on the existing CityGML model for the creation of the LOD3 CityGML model, can be conducted with the same techniques as the ones that are mentioned for the window enrichments, but with different heuristic rules.

### 2.3.2 Related work

Research is conducted on the topic of the automatic and bidirectional enrichment of CityGML and MVSM models, in order to find out if such an attempt has been previously achieved. Specifically on the subject of the enrichment of a CityGML model with information from a MVSM model of the same entities and conversely, no related work has been found. All the topics that are related to the exploration of achieving automatic bidirectional enrichments between these two 3D city models (see Table 2.1), are examined. Several fields that have relation to the goal of this thesis are: interoperability of information, semantic enrichment of 3D city models, straightening of meshes, matching of meshes and mesh comparison. The techniques that have been used on these fields have evolved the last decade and can also be used for the creation, the enhancement or the enrichments of created 3D city models. The choice of a technique is related with the different characteristics that each 3D city model has (Section 1.1) and the characteristics of the foreseen application (Section 2.3.1). The basic methodologies that are used in most of the applications of the different fields, are presented below. They can be used individually or can be combined, in order

to achieve an automatic, semi-automatic or manual enrichment (Section 2.3.1) of a 3D city model (3<sup>rd</sup> sub-question from Section 1.2.1). The methodologies that have been used in similar applications and will be elaborated below, are:

- schemas connection
- creation and comparison of planes
- probabilistic methods
- predefined shapes of features
- template matching techniques
- machine learning techniques
- clustering techniques
- heuristic rules
- distance computation methods

For this thesis, the basic focus of the enrichments that are conducted between the two 3D city models, is elaborated in the scope of research in Section 1.2.2. They are related to the semantic segmentation of the MVSM model on roof, wall, road, terrain and uncertain. Also, the transfer of texture from the MVSM model to the buildings of the CityGML model is explored and the existence of buildings together with the shapes of the roofs of the CityGML model are validated with information from the MVSM model (see Table 2.1 in Section 2.3.1). Next, the different methodologies that are listed above and have already been used into several applications, are analyzed. Furthermore, comments on the relation of these methodologies to the proposed methodology that will be elaborated in the next chapter (Chapter 3), are provided.

**SCHEMAS CONNECTION:** Firstly, some research has been done on the interoperability of information between 3D city models, which focuses on the way that the data is formatted in the format files of different 3D data models. An approach of the transformation of data from one 3D city model to another one, is done by Kavisha and Saran (2015), where basically the transformations from COLLADA to CityGML model and from BIM to CityGML model are quite interesting, but they are focused on the geometrical model of the different 3D city models and not on the semantic or texture information that is related to my research. Moreover, research from Zlatanova et al. (2012) and Kavisha and Saran (2015) has shown that many problems can occur during the conversion of data from one 3D city model to another, such as loss of information, loss of relationships, topological inconsistencies, etc. These problems are expected during the proposed methodology, discussions and proposals on these are presented in Chapter 5.

**CREATION AND COMPARISON OF PLANES:** The creation of planes on a mesh is a method that is usually used into several applications. Plane creation is effective in architectural scenes, but pose problems for non-architectural structures such as trees (Furukawa et al., 2015). Planar structures are used by Verdie et al. (2015), where the detection and the regularization of planar structures on buildings, helped on the classification of a MVSM model into four categories (ground, tree, facade and roof) without other dataset used. Another approach that has worked with plane creation uses airborne laser scanning point clouds and in combination with an outlier detection method, called RANdom SAmple Consensus (RANSAC), the plane fitting is done recursively and adjacent planar intersections are computed (Sun, 2016). One more approach that uses RANSAC shape detection for detecting planar surfaces of roof and walls is by Malihi et al. (2016). The above approaches have resulted into different semantic splits of the meshes. Semantic segmentation of the MVSM model is one of the goals of this thesis during the bidirectional enrichment of the two 3D city models, but due to the fact that plane surfaces already exist from the surfaces of the CityGML model of the same features, then no further planes are planned to be created and the existing ones are used for the semantic segmentation of the MVSM model. In addition, a method that is linked with the straightening of meshes of a MVSM model, is elaborated by Jonsson (2016) and detects the approximately flat surfaces in 3D meshes and replaces them with planes. This approach is focusing on the enhancement

of the quality of the *MVSM* models. In my research, the faces of the *MVSM* model are projected on the related plane surfaces of the *CityGML* model, as it was conducted during the process of the straightening of the mesh by [Jonsson \(2016\)](#). However, in this case, the projection of the *MVSM* model face is done, in order to project the texturing information to the *CityGML* model surfaces.

**PROBABILISTIC METHODS:** The probabilistic approach is one more methodology that can be useful for the achievement of enrichments on 3D city models. [Mayer and Reznik \(2005\)](#) used Markov Chain Monte Carlo (MCMC) to reconstruct buildings, where they obtained facade planes and made several assumptions based on aspect ratios, rectangularity and pixel information, in order to detect windows. Moreover, [Diakit  et al. \(2014\)](#) provided an automatic propagation approach guided by heuristic rules, designed in a way that the user can decide how many rules are needed to supervise the semantic attribution of a 3D model. Two really interesting methodologies from [Verdie et al. \(2015\)](#) and [Nguatem et al. \(2016\)](#) have resulted on the creation of *CityGML* models from the use of *MVSM* models. Firstly, [Verdie et al. \(2015\)](#) created a workflow that produces semantically rich *CityGML* models from a triangular mesh, with a classification method that is based on a Markov random field. In this case, a classification step, an abstraction step and a reconstruction step are performed on the *MVSM* model, in order to generate meaningful *CityGML* models of different LODs. [Nguatem et al. \(2016\)](#) presents a workflow for the automatic generation of building models with LOD 1 to 3 *CityGML* after the creation of a *MVSM* model. This work is an extension of the previous work that is done by [Nguatem et al. \(2013\)](#) and [Nguatem et al. \(2014\)](#). A stochastic method was created for the roof model selection from a *MVSM* model with the use of predefined roof shapes ([Nguatem et al., 2013](#)). Next, by [Nguatem et al. \(2014\)](#) the method was extended with the use of a stochastic method for the identification of the window model from the *MVSM* model. In the method of [Verdie et al. \(2015\)](#) and [Nguatem et al. \(2016\)](#), the generation of the *CityGML* models is based on the information that is given from the *MVSM* model. However, in this thesis the *CityGML* model of the same features already exists, therefore for example the shape of the roofs do not need to be defined from scratch. Also, the LOD of the proposed methodology is on LOD2, so a stochastic method for the detection of the windows is not needed. On the other hand, the methodologies that are used by [Verdie et al. \(2015\)](#) and [Nguatem et al. \(2016\)](#) can be very useful for some of the other enrichments that can be conducted between the two 3D city models (see [Table 2.1](#)), such as the creation of a higher LOD of a *CityGML* model with information from the *MVSM* model.

**PREDEFINED SHAPES OF FEATURES:** Within computer vision, template matching is a standard method for identifying objects in images ([Sonka et al., 2014](#)). This methodology has been used into several applications, in order to identify objects of the 3D scene, but only two recent works have been found that are related to the topic of this thesis. A paper that presents an approach of 3D building reconstruction, has achieved to overcome some of the limitations of planar fitting procedures by the use of template matching ([McClune et al., 2016](#)). More specifically, roof vertices were extracted from true-orthophotos using edge detection and were converted to roof corner points. It has to be added that this method has a lot of potential for the reconstruction of complex 3D building models at *CityGML* LOD1 and LOD2 specification ([McClune et al., 2016](#)). A recent approach by [Slade et al. \(2017\)](#) uses computer vision techniques including the Histogram of Oriented Gradient (HOG) template matching, in order to enrich *CityGML* models with information about windows and doors. These methodologies can be extended to be used for the bidirectional enrichment of the two 3D city models, like for the creation of higher LODs, such as LOD3 and LOD4, where the information of the roof details, the windows and the doors are important (see [Table 2.1](#)). In this thesis, the enrichments tested with the proposed methodology are not into such details, due to the fact that the tested LODs are the LOD1 and LOD2.

**MACHINE LEARNING AND CLUSTERING TECHNIQUES:** Machine learning techniques can be used for better labelling of the different features of a *MVSM* model. An example of such an attempt is made by [Kalogerakis et al. \(2010\)](#). They created an algorithm, which uses hundreds of geometric and contextual label features and learns different types of segmentations for different tasks, without requiring manual parameter tuning. Another example that uses a machine learning approach for the labelling of the regions of triangles, is the work by [Rook et al. \(2016\)](#). In this case, a soup of polygons serves as input and the different spatial features are recognised, distinguished and structured, following the *CityGML* definitions. After the clustering of the triangles into different classes by region growing algorithm, heuristic rules are used for the distinction of



the classes. Then, the regions with high likeliness score on the different classes, served the decision space used on the machine learning techniques (Rook et al., 2016). Machine learning and clustering techniques are two methodologies that can be used during the process of all the enrichments of a MVSM model with information from a CityGML model and on the LOD generation of the opposite direction (see Table 2.1). However, it has to be noted that none of the above approaches has attempted to enrich semantically a MVSM model with the use of information from an existing CityGML model and vice versa. In the proposed methodology, the two existing models are used and the different information that can be derived from their comparison is tested.

**DISTANCE COMPUTATION METHODS:** More relevant work with the approach that is used in the proposed methodology (Chapter 3) for the comparison of the two 3D city model surfaces and the attempt of the semantic segmentation, is the field of the mesh comparison and the matching of 3D meshes or shape matching. Studies on the mesh comparison of different simplifications are done by Roy et al. (2002) and Roy et al. (2004), who assessed respectively, the geometric and the attribute mesh quality after a simplification process, by a comparison between the original and its simplified representation. The simplification of the meshes is out of the scope of this research (Section 1.2.2), but some approaches that are used in that field and also, in the mesh matching field (Jesorsky et al., 2001; Veltkamp, 2001; Ericson, 2004) have shown that point-to-polygonal-mesh distance and Hausdorff distance are more relevant to the approach of this thesis. The point-to-polygonal mesh distance is judged to be more reliable (Guezlec, 2001; Ericson, 2004), for the semantic segmentation of the MVSM model with the use of the different layers of the CityGML model. According to Ericson (2004) and Eberly (2006) a way of obtaining the closest point between a triangle and a point, is to use a vector calculus approach. The minimum of this function occurs in one of three cases, at a vertex, on an edge or in the interior of the triangle. This approach is used, in order to compare the meshes of the two different existing 3D data models and segment the meshes of the MVSM model according to the surfaces of the CityGML model. Then, the Hausdorff distance is used for the quality control of the conducted matching of the surfaces of the MVSM model and for the validation of the existence of buildings and the validation of the shapes of the roofs on the corresponding CityGML model. It is the maximum of all the minimum distances from a point in one model, to all the points in the other model, so it provides a global and not a local comparison between two meshes (Veltkamp, 2001).



# 3

## BIDIRECTIONAL ENRICHMENT METHODOLOGY

The literature review in Section 2.3.2 of Chapter 3 did not identify any complete method for the automatic bidirectional enrichment of existing CityGML (Section 2.1) and MVSM (Section 2.2) models of same features. Hence, a new methodology is proposed, which is based on distance computations between the two 3D city models. This technique is decided due to the fact that both models are available and a comparison between the meshes of the two 3D city models should lead to fruitful results. The goal of this methodology is, to be able to segment semantically the mesh of the MVSM model into five categories (roof, wall, road, terrain, uncertain) and transfer the attributes of each building from the CityGML to the MVSM model. The quality of the results from the semantic segmentation is tested for the roof and the wall surfaces between the meshes of the two 3D city models. Next, enrichments of the CityGML model are achieved, including the transfer of the texture of the buildings and the validation of the buildings into the model. During the validation, the shapes of the roofs of the buildings and the absence of buildings in the CityGML model are investigated. The limitations of this research are mentioned in Section 1.2.2. It has to be noticed that the methodology is focused on the enrichments of CityGML models of LOD1 and LOD2. An overview of the proposed methodology is presented in Section 3.1 and the different basic steps of the proposed methodology are thoroughly explained in Sections 3.2, 3.3, 3.4 and 3.5.

### 3.1 OVERVIEW OF THE METHODOLOGY

The process of the proposed bidirectional enrichment methodology between CityGML and MVSM models of the same entities, consists of four basic steps, which are presented in Figure 3.1. Initially, for the proposed methodology the two 3D city models are the input data. The basic aspects of the two models that are important for the achievement of the proposed methodology are, for the CityGML model, the semantics (roof, wall, ground) and the identifier of each building, and for the MVSM model the texture information. First steps of the methodology are the removal of the CityGML faces that do not correspond to the faces of the MVSM model and then, the matching of the faces of the MVSM model to the remaining faces of the CityGML model. After these steps, the semantic segmentation of the mesh of the MVSM model is conducted, where the mesh is divided into five classes (roof, wall, road, terrain and uncertain). After this action, a quality control of the matching of the previous step is conducted. Moreover, the validation of the shapes of the roof and the validation of the existence of buildings in the CityGML model take place and the results of the validations are exported. Finally, the texture included in the MVSM model is transferred to the CityGML model and a textured CityGML model is created.

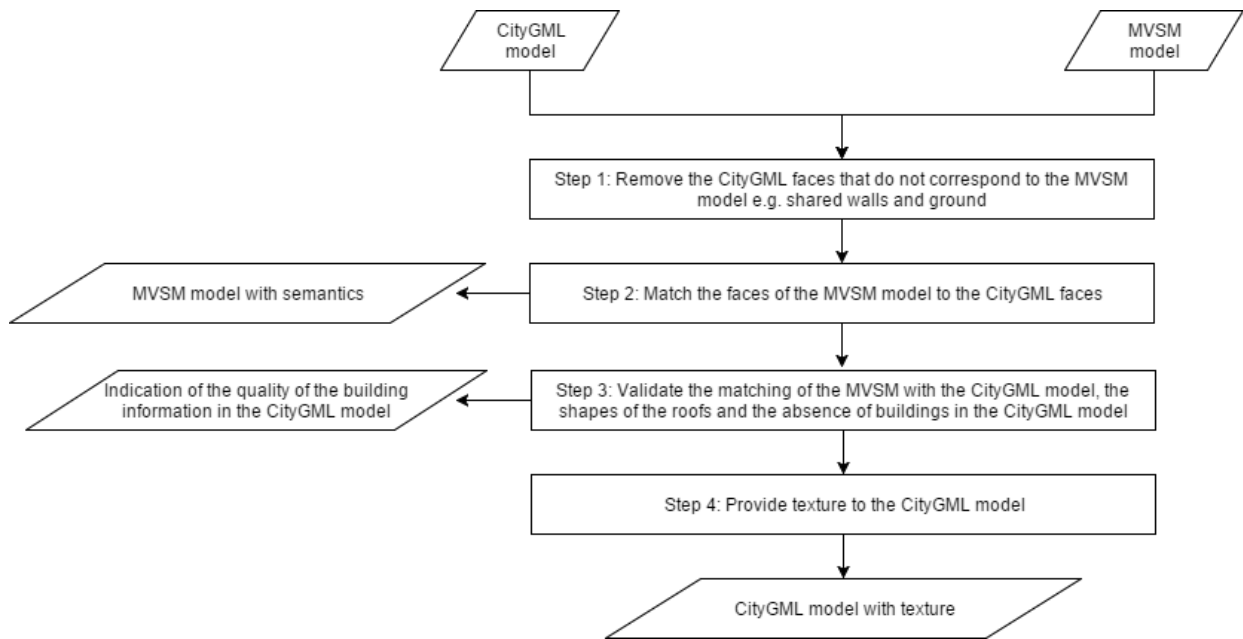


Figure 3.1: Flowchart of the proposed methodology on the bidirectional enrichment of CityGML and MVSM models.

### 3.2 STEP 1: REMOVE THE CITYGML FACES THAT DO NOT CORRESPOND TO THE MVSM MODEL

During the proposed methodology, the meshes of the two 3D city models are compared, therefore the faces of the models need to correspond to the same features of the 3D scene. It has to be added in this point that the semantic information given from a LOD1 or LOD2 of a CityGML model that models buildings in 3D space, is related to the roof, the wall and the ground of these buildings. In Figure 2.2, examples of these surfaces of a LOD2 CityGML model of four buildings are illustrated as transparent polygons. However, the mesh of the MVSM model (grey triangles in Figure 3.2) does not correspond to all the surfaces of this CityGML model. Considering the creation of the MVSM models, the parts of the buildings that are captured on the image acquisition step (Section 2.2), in relation to the faces of a CityGML model are, the roof surfaces and the outer walls of the buildings (see Figure 2.4), because only these parts of the buildings are visible from the terrestrial and the aerial imagery that is used for the MVSM model creation. Figure 3.2 illustrates the differences between the corresponding surfaces of the two 3D city models.

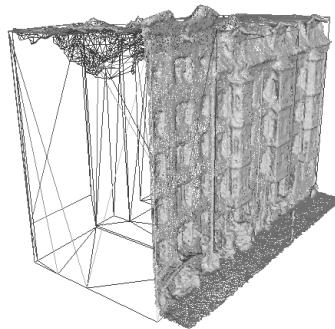


Figure 3.2: Differences between the corresponding surfaces of the CityGML (transparent) and the MVSM (grey) models.

In Figure 3.2, it is obvious that the faces of the CityGML model that make sense to be compared with the MVSM model's mesh are, the roof surfaces and the outer wall faces of each building that are not shared with other wall faces. Therefore, in order to prevent mismatches during face matching, the faces of the CityGML model that correspond to the ground and the shared walls of the buildings, are removed in this step (see Figure 3.3).

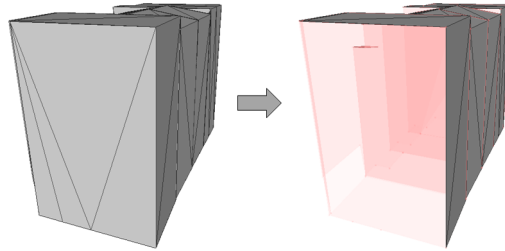


Figure 3.3: Faces of the CityGML model that are removed (red faces).

A flowchart that explains the steps of the removal of the CityGML faces that do not correspond to the MVSM model is shown below (4<sup>th</sup> sub-question from Section 1.2.1):

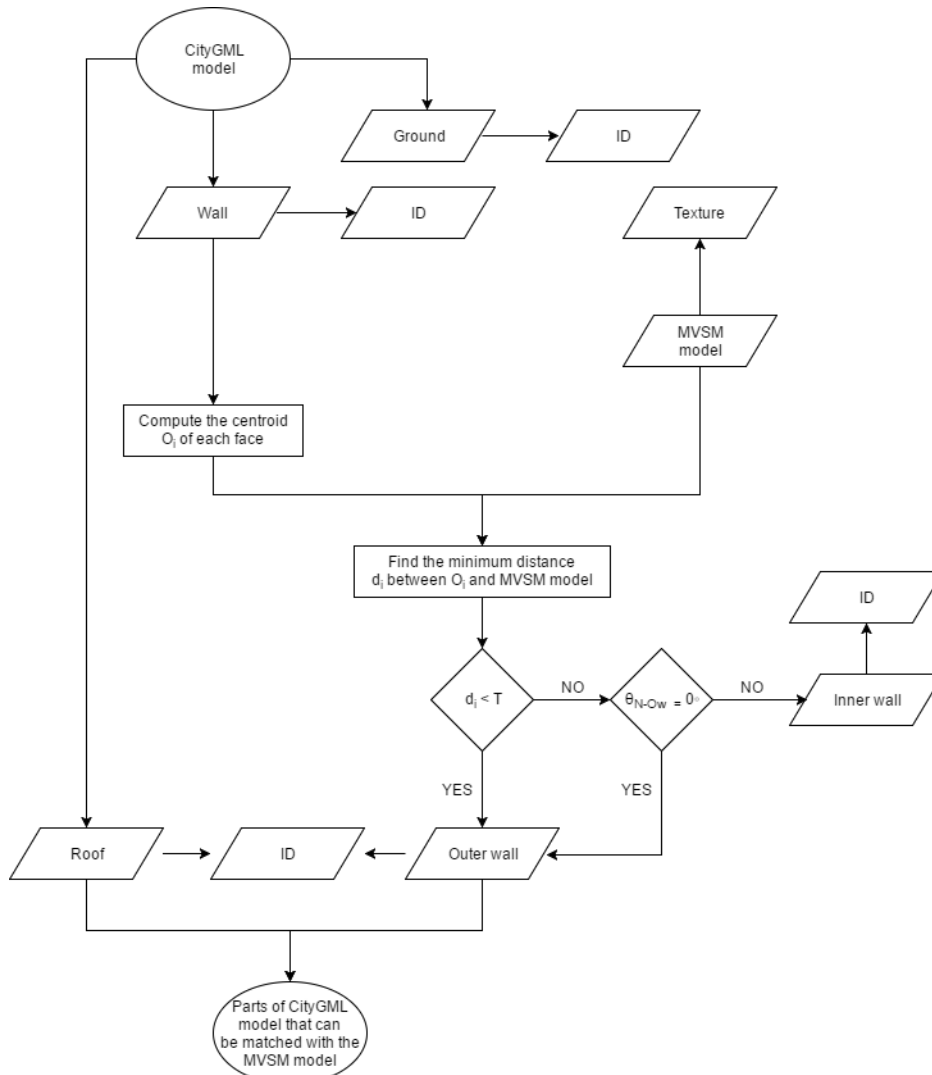


Figure 3.4: Flowchart illustrating the parts of step 1 of the proposed methodology.

According to the flowchart in Figure 3.4, the ground surfaces of the CityGML model, which correspond to the ground of the buildings of the 3D scene are removed. Then, the roof surfaces of the CityGML model are chosen for the next step of the proposed methodology, together with the outer walls of the buildings. The process that is done to remove the wall faces of the CityGML model that do not correspond to the MVSM model is firstly, based on the distances  $d_i$  between the centroids  $O_i$  of the faces of the CityGML model and the faces of the MVSM model. The centroids of the faces of the CityGML model are chosen for the distance  $d_i$  computations, because the vertices of the same faces might touch the MVSM model, so they cannot be a good judge for the choice of the faces of the CityGML model that correspond to the MVSM model.

The centroid  $O_i$  of a triangle is the point, where the three medians of the triangle meet. A median of a triangle is a line segment from one vertex to the midpoint on the opposite side of the triangle (Vince, 2006). The following figure depicts the triangle ABC, the medians AF, BD, CE and the centroid O:

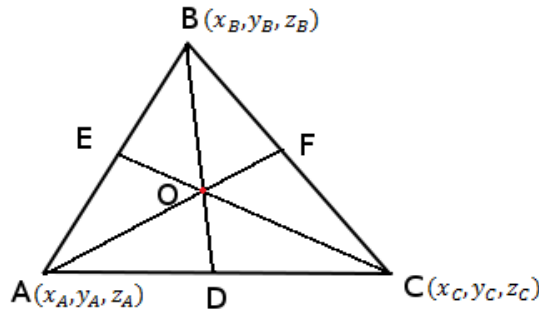


Figure 3.5: Centroid of the triangle ABC.

The coordinates of the centroid O of the triangle that is depicted in Figure 3.5 are equal to:

$$O = \left( \frac{x_A + x_B + x_C}{3}, \frac{y_A + y_B + y_C}{3}, \frac{z_A + z_B + z_C}{3} \right) \quad (3.1)$$

After the computation of the centroids  $O_i$  of each face of the CityGML model, according to the equation 3.1, the minimum distances  $d_i$  between these centroids  $O_i$  and each face of the MVSM model are calculated. The minimum distance between a point and a triangle in 3D space is found based on the point-to-polygonal mesh distance. The point-to-polygonal mesh distance is a way of obtaining the closest point (Q) between a triangle (ABC) and a point (P) (see Figure 3.6). The Closest Point Approach (CPA) between two points (Q and P) that are dynamically moving in straight lines is computed. The CPA refers to the positions at which two dynamically moving objects reach their closest possible distance. It is done with the use of a vector calculus approach and the minimum of this function must occur in one of the three cases that are shown in Figure 3.6: at a vertex (Q'), on an edge (Q'') or in the interior of a triangle (Q''') (Ericson, 2004; Eberly, 2006).

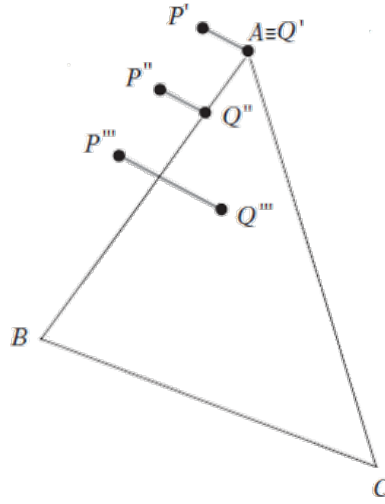


Figure 3.6: Point to polygonal mesh distance.

The distance between the points P and Q is computed based on the equation of the Euclidean distance:

$$d_{PQ} = \sqrt{(x_P - x_Q)^2 + (y_P - y_Q)^2 + (z_P - z_Q)^2} \quad (3.2)$$

A threshold of the distance  $d_i$  (calculated from equation 3.2) is decided, in order to choose the faces of the CityGML model that are more relevant to be matched with the MVSM faces.

**Heuristic rule 1 - Step 1:**

The faces of the CityGML model that are less than 0.3m ( $d_i < 0.3m$ ) closer to at least one face from the MVSM model, are used for the matching process of the meshes of the two 3D city models (Section 3.3).

The above heuristic rule gives an indication of all the faces of the CityGML model that are close enough to the corresponding MVSM model and are more relevant to be matched with the MVSM faces. After the above rule (Heuristic rule 1 - Step 1), for several faces of the CityGML model that are part of the same facade of a building and happened to have the centroid of their face at a distance bigger than the threshold T, a second heuristic rule is used.

**Heuristic rule 2 - Step 1:**

The normals  $N_i$  of the faces of the CityGML model that are more than 0.3m ( $d_i > 0.3m$ ) from all the faces of the MVSM model, but share at least one edge with at least one of the faces  $Ow_i$  of the CityGML model that have derived from the Heuristic rule 1 - Step 1, are computed. Then, the angle  $\theta_{N-Ow}$  between the normal vectors  $N$  and  $Ow$  of the related faces of the CityGML model is calculated. In case the normals of these faces have an angle  $\theta_{N-Ow}$  equal to  $0^\circ$  ( $\theta_{N-Ow}=0^\circ$ ), then the face  $N$  of the CityGML model is assigned as an outer wall and is used for the matching process of the meshes of the two 3D city models (Section 3.3).

The equations used for the computation of the normal vector of a triangle and the equations that compute the angle of two vectors in 3D space, are analyzed below. A normal vector of a triangle  $P_2P_3P_1$ , is a vector  $\vec{N}$  that is perpendicular to that triangle (see Figure 3.7).

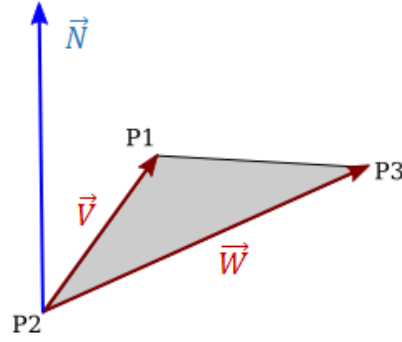


Figure 3.7: Example of the normal vector  $\vec{N}$  (cross product) of the triangle  $P_2P_3P_1$ .

The normal vector  $\vec{N}$  is obtained by taking the cross product of  $\vec{W}$  and  $\vec{V}$ , which are vectors that lie along the two sides of the triangle (Vince, 2006). If the cross product of the given triangle is  $\vec{N}$ , then the coordinates of the vector  $\vec{N}$  are:

$$\vec{N} = (N_x, N_y, N_z) \quad (3.3)$$

In order to find these coordinates, it is known that the cross product of two sides of the triangle, equals the surface normal. Therefore, if:

$$\vec{V} = P_2 - P_1 \quad (3.4)$$

$$\vec{W} = P_3 - P_1 \quad (3.5)$$

Then, the cross product can be expressed as:

$$\begin{aligned} \vec{N} &= \vec{W} \times \vec{V} \\ &= \begin{vmatrix} i & j & k \\ W_x & W_y & W_z \\ V_x & V_y & V_z \end{vmatrix} \end{aligned} \quad (3.6)$$

$$= (W_y \cdot V_z - W_z \cdot V_y)i + (W_z \cdot V_x - W_x \cdot V_z)j + (W_x \cdot V_y - W_y \cdot V_x)k$$

Therefore, the components of the resulting vector are:

$$N_x = (W_y \cdot V_z) - (W_z \cdot V_y) \quad (3.7)$$

$$N_y = (W_z \cdot V_x) - (W_x \cdot V_z) \quad (3.8)$$

$$N_z = (W_x \cdot V_y) - (W_y \cdot V_x) \quad (3.9)$$

In addition, the angle  $\theta$  between the normal vectors of two triangles is used, in order to decide whether or not two triangles have small or big difference on their orientation (Heuristic rule 3 - Step 2). The angle  $\theta$  between two vectors  $\vec{u}$ ,  $\vec{v}$  in 3D space is illustrated in the following figure:



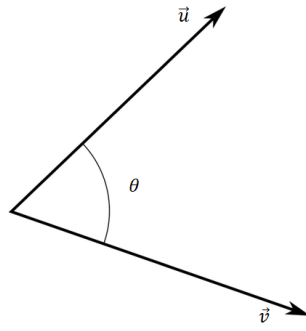


Figure 3.8: Angle ( $\theta$ ) between two vectors ( $\vec{u}$  and  $\vec{v}$ ).

The two vectors are expressed as:

$$\vec{u} = (u_1, u_2, u_3) = u_1i + u_2j + u_3k \quad (3.10)$$

$$\vec{v} = (v_1, v_2, v_3) = v_1i + v_2j + v_3k \quad (3.11)$$

It is known that the dot product of two vectors is equal to:

$$\vec{u} \cdot \vec{v} = \|\vec{u}\| \|\vec{v}\| \cos \theta \quad (3.12)$$

The dot product can be also described as:

$$\vec{u} \cdot \vec{v} = u_1 \cdot v_1 + u_2 \cdot v_2 + u_3 \cdot v_3 \quad (3.13)$$

Therefore, with the use of the equations 3.12 and 3.13, the angle  $\theta$  between two vectors is equal to:

$$\begin{aligned} \theta &= \arccos \left( \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \|\vec{v}\|} \right) \\ &= \arccos \left( \frac{u_1 \cdot v_1 + u_2 \cdot v_2 + u_3 \cdot v_3}{\|\vec{u}\| \|\vec{v}\|} \right) \end{aligned} \quad (3.14)$$

### 3.3 STEP 2: MATCH THE FACES OF THE MVSM MODEL TO THE CITYGML FACES

The goal of this step is to use the knowledge of the semantic information of the CityGML model for the buildings (roof and wall), in order to categorize semantically the MVSM model into three classes, roof, wall and uncertain. Then, the class of the uncertain is categorized based on the semantic information of the CityGML model of the relief features and the transportation bodies into three classes, terrain, road and uncertain (4<sup>th</sup> sub-question from Section 1.2.1) (step 2 in Figure 3.1). After the previous step of the proposed methodology (Section 3.2), the roof and the outer wall faces of a CityGML model, which correspond to the faces of a MVSM model, are automatically selected. These faces of the CityGML model provide in this step, a better matching between the faces of the two 3D city models. The matching of the two 3D city models is done with a combination of distance computations, normal vector comparisons and heuristic rules. The following flowchart shows the process of this step until the categorization of the MVSM model into the three classes, roof, wall and uncertain:

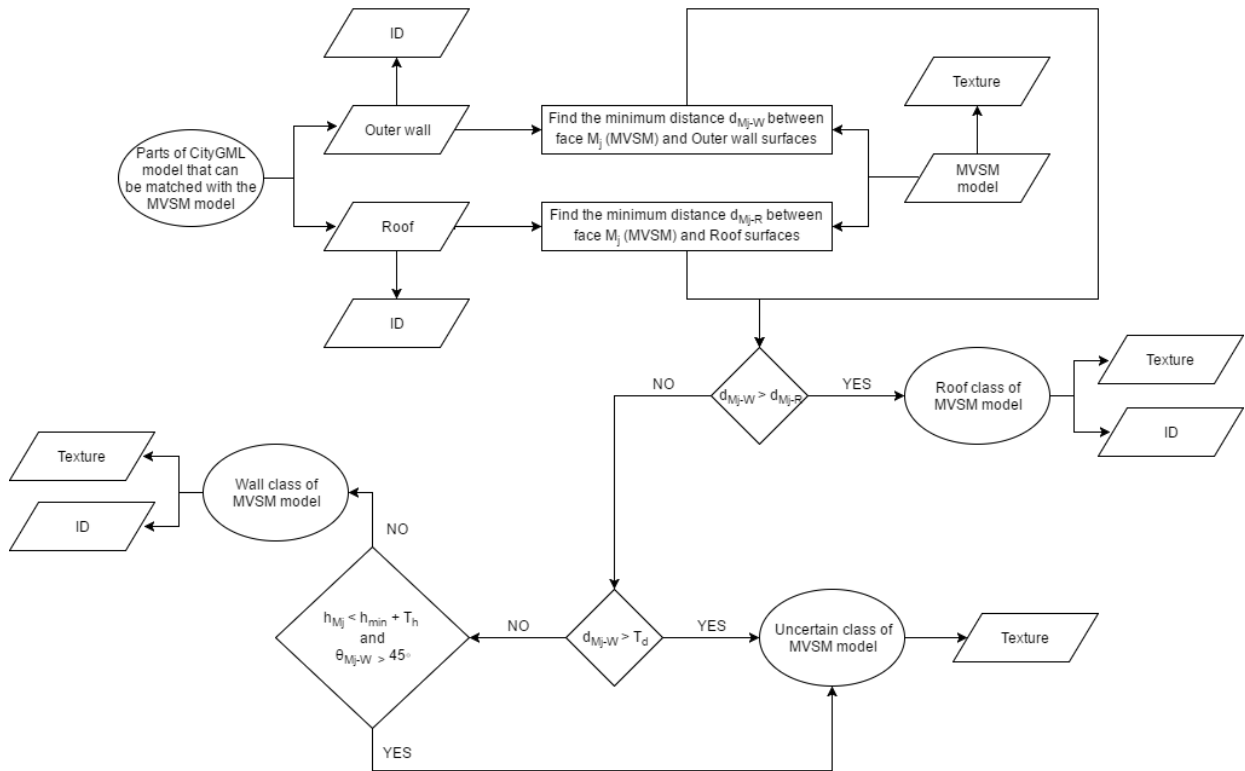


Figure 3.9: Flowchart illustrating the parts of step 2 of the proposed methodology for the segmentation of the MVSM model to roof, wall and uncertain.

The first computation is the minimum distance between each face  $M_j$  of the MVSM model with each face of the CityGML model with the use of the point-to-polygonal distance (described in Section 3.2). The minimum distance  $d_{M_j-W}$  between the face  $M_j$  of the MVSM model and the wall surfaces  $W$  of the CityGML model is compared with the minimum distance  $d_{M_j-R}$ , between the face  $M_j$  of the MVSM model and the roof surfaces  $R$  of the CityGML model.

#### Heuristic rule 1 - Step 2:

Each face  $M_j$  of the MVSM model is assigned as roof class, if the distance between  $M_j$  and the roof surface  $R$  of the CityGML model is smaller than the distance between  $M_j$  and the wall surface  $W$  of the CityGML model ( $d_{M_j-R} < d_{M_j-W}$ ).

It has to be added that in case a face of the MVSM model is assigned as roof class after the Heuristic rule 1 - Step 2 of the methodology, the IDentity (ID) of the related roof surface of the CityGML model that is closer to this MVSM face, is assigned to this face. On the other hand, the faces of the MVSM model that are closer to the wall surfaces of the CityGML model ( $d_{M_j-W} < d_{M_j-R}$ ) are proceeded to further tests.

#### Heuristic rule 2 - Step 2:

In case the distance  $d_{M_j-W}$  between the face  $M_j$  of the MVSM model and the wall surface  $W$  of the CityGML model is bigger than a threshold distance  $T_d$  ( $d_{M_j-W} > T_d$ ), then the face  $M_j$  of the MVSM model is assigned as uncertain.

The threshold  $T_d$  is decided for this thesis around 1m. This value is an average value related to the windows and the doors recesses that are part of the wall surface of the MVSM model, which needs to be

matched to the plane wall surface  $W$  of the CityGML model. Therefore, the objects that have  $d_{M_j-W}$  distance more than  $T_d$  can either be transportation, relief features or other thematic classes.

Moreover, the remained faces of the MVSM model are checked in another rule, which takes into consideration the minimum height  $h_j$  of each face  $W$  of the wall surface of the CityGML model and the angle  $\theta_{M_j-W}$  between the normal vector of the related wall surface of the CityGML model and the normal vector of the face  $M_j$  of the MVSM model.

### Heuristic rule 3 - Step 2:

The minimum height  $h_{M_j}$  of each face  $M_j$  of the MVSM model is compared with the minimum height of the CityGML model and a threshold  $T_h$ . In case the  $h_{M_j}$  height of the face  $M_j$  is lower than the  $h_{min}$  plus the threshold  $T_h$  ( $h_{M_j} < h_{min} + T_h$ ), it might not be part of the wall surface of a building and it might be part of another thematic class, such as transportation. In order to check that, the angle  $\theta_{M_j-W}$  between the normal vector of the related face of the CityGML model and the normal vector of the face  $M_j$  of the MVSM model is tested. If and only if the height  $h_{M_j}$  of the face  $M_j$  of the MVSM model is lower than the minimum height of the CityGML model, plus a threshold  $T_h$ , and if the angle  $\theta_{M_j-W}$  is lower than  $45^\circ$  ( $\theta_{M_j-W} < 45^\circ$ ), then the face  $M_j$  of the MVSM model is assigned as uncertain class, while the rest of the faces of the MVSM model that do not follow this rule, are assigned as wall class.

It has to be noted that, as it is already mentioned for the roof class of the MVSM model, the ID of the building of the related roof surface of the CityGML model that is closer to this MVSM face, is assigned to this face. The same is also provided to the faces of the MVSM model that correspond to the wall class (see flowchart in Figure ??), (5<sup>th</sup> and 6<sup>th</sup> sub-question from Section 1.2.1).

After the distinction of the MVSM faces into the three different classes (roof, wall, uncertain), some faces have been categorized as roof surfaces, while they are wall surface, or uncertain. The same happens for the other two classes, as a result there are holes and some scattered triangles after the face matching of the models (see Figure 3.10).

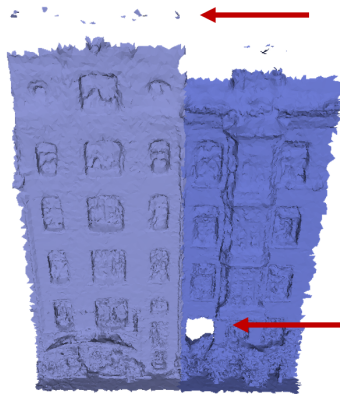


Figure 3.10: Unconnected faces on the mesh (scattered triangles and holes).

The solution is to take into consideration the connected components. Connected faces can be found in the 3D city model, because they share an edge. It can be said that the faces that are not connected together, are not part of the same group.

Before proceeding to the solution for the triangular network that is used for this thesis, a simple 2D example of the connected component's problem is illustrated in Figure 3.11. On the left image the graph shows three different groups of connected components assigned as same class (green), for example the wall class. In order to find out whether these groups are all connected together, each point is tested with a neighbour and added it to a set. After a recursive check of all the neighbours, all the points that have been assigned to the same sets are connected components, while the ones that are not assigned together are unconnected. The result is given on the right figure below, where the 3 different connected groups are found (red, green, blue) and they are assigned to the three different groups, with which they are

connected e.g. roof, wall and uncertain, respectively.

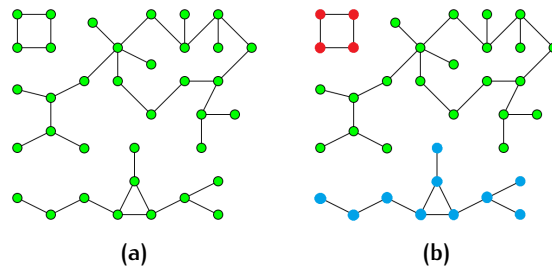


Figure 3.11: Example of the problem of the connected components. (a) Three different groups of connected components, assigned as same class (green). (b) Three different groups of connected components, assigned as three different classes (red, green, blue).

The same logic is used, in order to find the different connected components for each one of the three classes (roof, wall, uncertain) of the semantic segmentation of the mesh. The following flowchart shows the different steps:

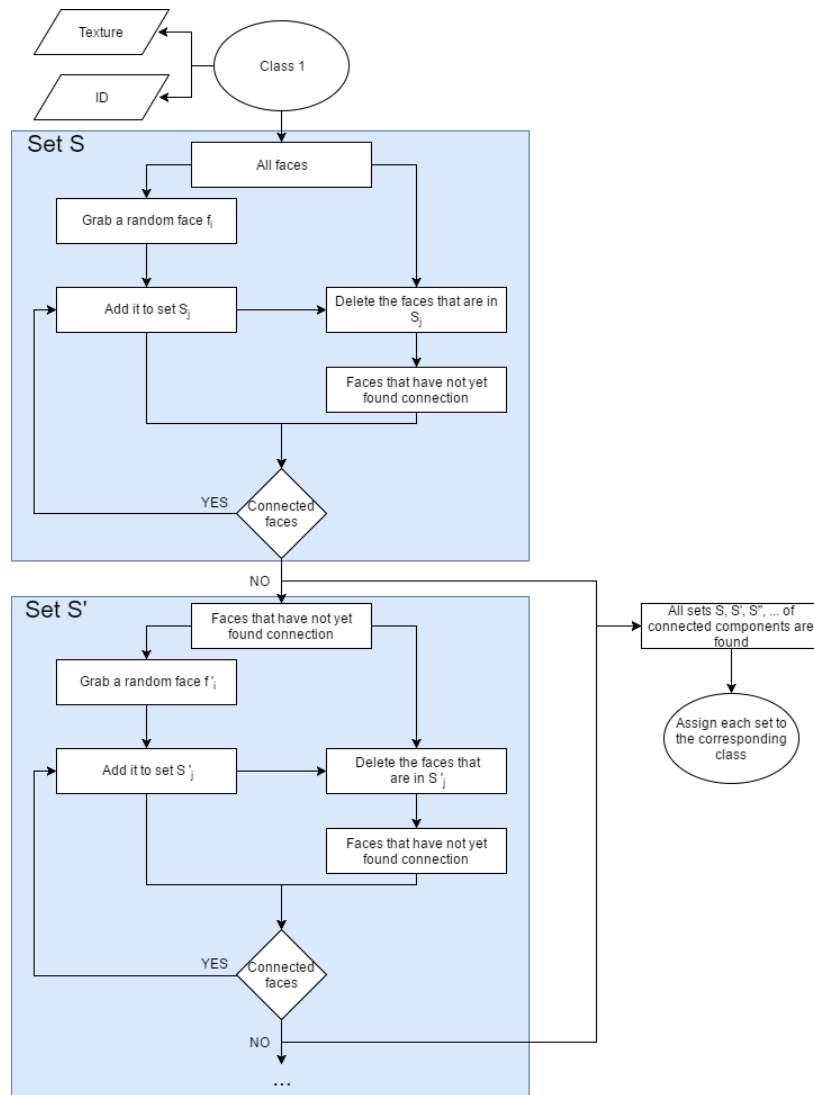


Figure 3.12: Flowchart illustrating the methodology used for the identification of the different connected components in a semantic class of a MVSM model.

The process that is explained in Figure 3.12 is done for all the three classes (roof, wall, uncertain), derived from the matching of the meshes of the two 3D city models (see Figure 3.9). First of all, from the faces of a class (Class 1) a random face  $f_i$  is chosen and it is added to a new set  $S_j$ . The faces of the Class 1 that are not in the set  $S_j$ , are tested for connectivity. This means that every face of the set  $S_j$  is tested with every face from the Class 1. If during the test, faces that share at least one of their vertices are found, then these faces are connected with the set  $S_j$ . Therefore, they are assigned to the set  $S_j$  and removed from the faces of the Class 1. This process is done iteratively, until all the points of the set  $S_j$  and all the remained points of the Class 1 are checked for connectivity. In case no more faces from Class 1 are imported to the set  $S_j$ , then a new set  $S'_j$  is created, following the same process. First choosing a random face  $f'_i$  from the remaining faces into the Class 1, until no more faces from the Class 1 are imported to the second set  $S'_j$ . The same can be done until all the faces of the Class 1 are assigned to the related sets. The different created sets  $S_j, S'_j, \dots$  are gathered and the biggest set is left as Class 1. Finally, the rest of the created sets are tested with the other classes of the model and each one of them is assigned to the class, with which it has more connections.

Furthermore, the faces of the MVSM model that are categorized as uncertain (see Figure 3.9) are proceeding into further investigation, following the process of the flowchart shown in Figure 3.13.

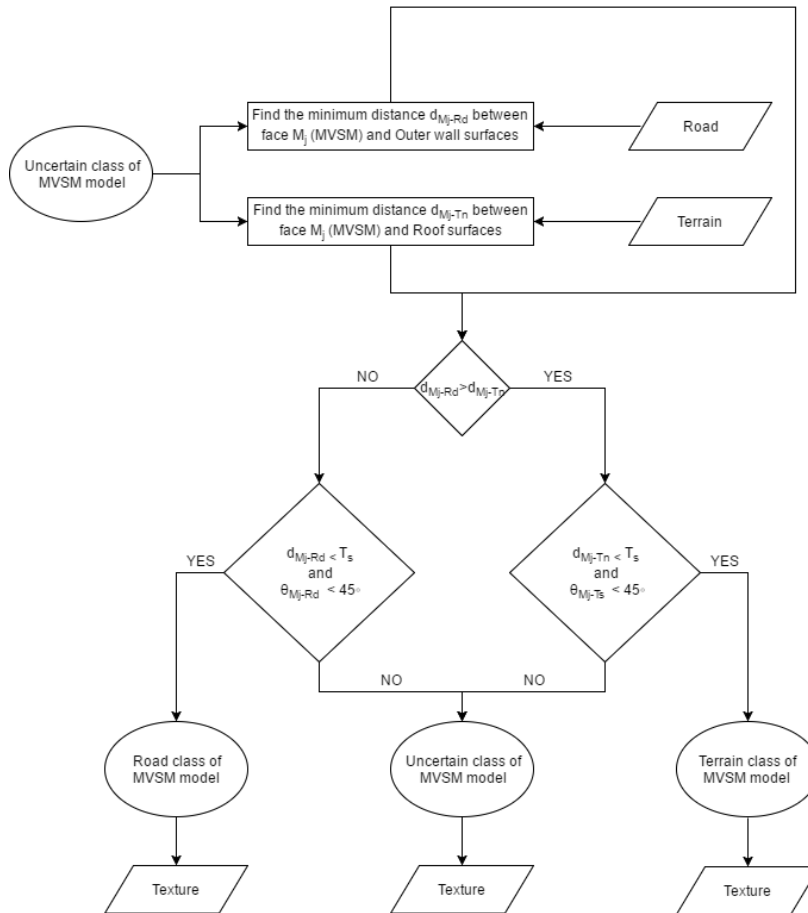


Figure 3.13: Flowchart illustrating the parts of step 2 of the proposed methodology for the segmentation of the MVSM model to road, terrain and uncertain.

The first computation for the faces of the MVSM model that are assigned from the previous heuristic rules as uncertain, is the minimum distance between each face  $M_j$  of the MVSM model with each face of the CityGML model that correspond to the road or the terrain class. The distance computations are made

with the use of the point-to-polygonal distance (described in Section 3.2). The minimum distance  $d_{M_j-Rd}$  between the face  $M_j$  of the **MVSM** model and the road surfaces  $Rd$  of the **CityGML** model is compared with the minimum distance  $d_{M_j-Tn}$ , between the face  $M_j$  of the **MVSM** model and the terrain surfaces  $Tn$  of the **CityGML** model.

**Heuristic rule 4 - Step 2:**

*Each face  $M_j$  of the **MVSM** model is proceeding into further investigation for its assignment as terrain class, if the distance between  $M_j$  and the terrain surface  $Tn$  of the **CityGML** model is smaller than the distance between  $M_j$  and the road surface  $Rd$  of the **CityGML** model ( $d_{M_j-Tn} < d_{M_j-Rd}$ ). On the other hand, each face  $M_j$  of the **MVSM** model is proceeding into further investigation for its assignment as road class, if the distance between  $M_j$  and the road surface  $Rd$  of the **CityGML** model is smaller than the distance between  $M_j$  and the terrain surface  $Tn$  of the **CityGML** model ( $d_{M_j-Rd} < d_{M_j-Tn}$ ).*

The faces of the **MVSM** model that are closer to the road surfaces of the **CityGML** model ( $d_{M_j-Rd} < d_{M_j-Tn}$ ) are proceeding to the *Heuristic rule 5 - Step 2*, while the faces of the **MVSM** model that are closer to the terrain surfaces of the **CityGML** model ( $d_{M_j-Tn} < d_{M_j-Rd}$ ) are proceeding to the *Heuristic rule 6 - Step 2*, in order to be distinguished as road, terrain or uncertain.

**Heuristic rule 5 - Step 2:**

*The distance between  $M_j$  face of the **MVSM** model and the road surface  $Rd$  of the **CityGML** model is compared with a threshold  $T_s$ . In case the  $d_{M_j-Rd}$  distance is lower than the threshold  $T_s$  ( $d_{M_j-Rd} < T_s$ ), it might not be part of the road surface of a building and it might be part of an uncertain class, such as vegetation. In order to check that, the angle  $\theta_{M_j-Rd}$  between the normal vector of the related face of the **CityGML** model and the normal vector of the face  $M_j$  of the **MVSM** model is tested. If and only if the  $d_{M_j-Rd}$  distance is lower than the threshold  $T_s$  and if the angle  $\theta_{M_j-Rd}$  is lower than  $45^\circ$  ( $d_{M_j-Rd} < T_s$  and  $\theta_{M_j-Rd} < 45^\circ$ ), then the face  $M_j$  of the **MVSM** model is assigned as road class, while the rest of the faces of the **MVSM** model that do not follow this rule, are assigned as uncertain.*

**Heuristic rule 6 - Step 2:**

*The distance between  $M_j$  face of the **MVSM** model and the terrain surface  $Tn$  of the **CityGML** model is compared with a threshold  $T_s$ . In case the  $d_{M_j-Tn}$  distance is lower than the threshold  $T_s$  ( $d_{M_j-Tn} < T_s$ ), it might not be part of the terrain surface of a building and it might be part of an uncertain class, such as vegetation. In order to check that, the angle  $\theta_{M_j-Tn}$  between the normal vector of the related face of the **CityGML** model and the normal vector of the face  $M_j$  of the **MVSM** model is tested. If and only if the  $d_{M_j-Tn}$  distance is lower than the threshold  $T_s$  and if the angle  $\theta_{M_j-Tn}$  is lower than  $45^\circ$  ( $d_{M_j-Tn} < T_s$  and  $\theta_{M_j-Tn} < 45^\circ$ ), then the face  $M_j$  of the **MVSM** model is assigned as terrain class, while the rest of the faces of the **MVSM** model that do not follow this rule, are assigned as uncertain.*

The threshold  $T_s$  is decided for this thesis around 0.5m. All the faces of the **MVSM** model that are further away from either the road or the terrain surfaces of the related **CityGML** model, are assigned as uncertain.

### 3.4 STEP 3: VALIDATE THE MATCHING OF THE MVSM WITH THE CITYGML MODEL

After the completion of the previous step of the proposed methodology, the **MVSM** model is segmented into five classes, roof, wall, road, terrain and uncertain. This step of the proposed methodology is related to the validation of the matching of the faces of the two 3D city models that is conducted in the previous step (Section 3.3). The two layers of the roof and the wall, derived from the semantic segmentation of the **MVSM** model, are tested with the related surfaces roof and wall, from the **CityGML** model. In the previous step of the methodology, the IDs of the **CityGML** buildings are assigned to the corresponding faces of the **MVSM** model. Therefore, the faces that correspond to the same buildings in the **MVSM** and in the **CityGML** model, can be compared. The validation of the two classes is conducted with the use of the Hausdorff distance between the points of the two 3D city models. These distances are tested with a proposed categorization

(see Table 3.1), in order to distinguish the quality of the matching of the faces into excellent, good, medium or bad matching. According to the resulted indication (excellent, good, medium or bad) of the matching of the faces, it is decided whether or not to resegment the MVSM model with different parameters (repeat the process described in Section 3.3). The resegmentation of the mesh can take place as many times as the changes of the parameters improve the indication of the quality of the matching of the faces between the two 3D city models (the indications, excellent, good, medium and bad, derive from the categorisation that is shown in Table 3.1).

After the quality control of the semantic segmentation of the step presented in Section 3.3 and the resegmentation of the MVSM model, the Hausdorff distance between the points of the two classes (roof and wall) between the two meshes of the 3D city models, is computed again. In this way, the indication that derives from the Hausdorff distance of the roof and the wall of each building between the roof and the wall of the MVSM model and the corresponding roof and wall surface of the CityGML model, indicates the quality of the building information in the CityGML model (the indications derive from the categorisation that is shown in Table 3.1). In this way, the validation of the shapes of the roofs of the buildings can take place, together with a validation considering the existence of buildings in the CityGML model (7<sup>th</sup> sub-question from Section 1.2.1). The results of the Hausdorff distance computations can provide an indicator, showing whether or not a roof in the CityGML model might not correspond to the shape of the roof that is captured in a younger MVSM model. Also, the results of these computations can indicate whether or not a building might be missing in the CityGML model.

More specifically, the Hausdorff distance provides a global comparison between the two meshes (A, B). It is the maximum of all the minimum distances from a point in the one model, to all the points in the other model (Veltkamp, 2001).

The computation of the minimum distances between every two points of the two 3D city models is computed with the use of the equation 3.2 (mentioned in Section 3.2). In this case, the Euclidean distance is equal to:

$$d_{x_i y_j} = \sqrt{(x_{x_i} - x_{y_j})^2 + (y_{x_i} - y_{y_j})^2 + (z_{x_i} - z_{y_j})^2} \quad (3.15)$$

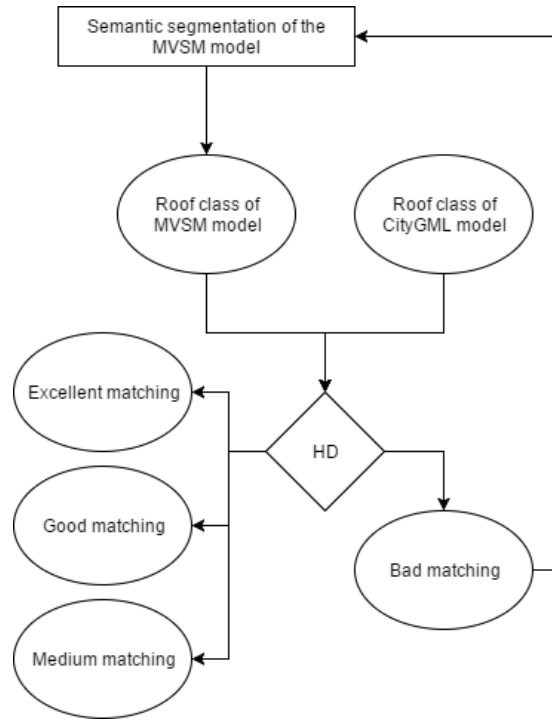
Given two finite point sets  $A = \{a_1, \dots, a_p\}$  and  $B = \{b_1, \dots, b_q\}$ , the Hausdorff distance is defined as:

$$HD(A, B) = \max(h(A, B), h(B, A)) \quad (3.16)$$

where

$$h(A, B) = \max_{a \in A} \min_{b \in B} (h(a, B), h(B, a)) \quad (3.17)$$

The process of the quality control of the matching of the faces after the semantic segmentation of the MVSM from the CityGML model (Section 3.3), is described in the flowchart in Figure 3.14.



**Figure 3.14:** Flowchart illustrating the process of the quality control of the semantic segmentation of step 2 (Section 3.3) of the proposed methodology.

According to the flowchart in Figure 3.14, the faces of the two 3D city models that correspond to the same class, for example the roof class, are checked for their quality. The minimum distances between all the vertices of the two models are computed and then, the maximum of these distances determine the related Hausdorff distance (HD in Figure 3.14). The same process is followed for the wall class of the two 3D city models. In order to categorize the results of the Hausdorff distances and indicate the quality of the matching of the faces between the two 3D city models (excellent, good, medium, bad), the categorisation shown in Table 3.1 is created.

Category	Interval of HD
Excellent	0m - 5m
Good	5m - 8m
Medium	8m - 11m
Bad	>11m

**Table 3.1:** Categorisation of the quality control of the semantic segmentation, based on the Hausdorff distance computations.

**Heuristic rule 1 - Step 3:**

If the Hausdorff distance, HD, between a MVSM and a CityGML model that correspond to the same class (roof, wall), is resulted as bad, based on the categories shown in Table 3.1, then the classes roof and wall that have derived from the matching of the faces between the two 3D city models in step 2 (Section 3.3), are resegmented with different parameters (repetition of the process presented in Section 3.3). The parameters that change during the resegmentation process are the:

- $T_d$  (Heuristic rule 2 - Step 2 in Section 3.3)
- $T_h$  (Heuristic rule 3 - Step 2 in Section 3.3)
- $T_s$  (Heuristic rule 5 - Step 2 and Heuristic rule 6 - Step 2 in Section 3.3)



After the resegmentation of the *MVSM* model into roof and wall classes, the validation of the shapes of the roofs and the validation of the existence of buildings in the *CityGML* model take place. The processes are provided in Figure 3.15 and in Figure 3.16, for the validation of the shapes of the roofs and the validation of absence of buildings, respectively.

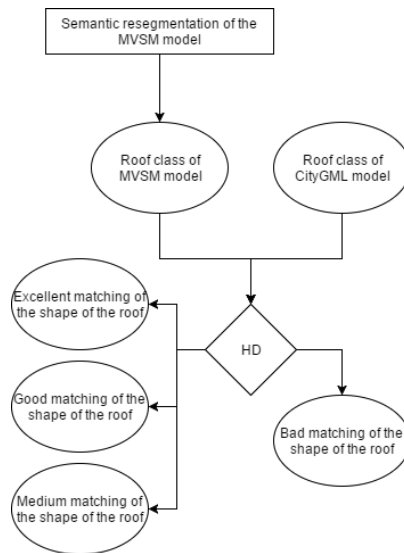


Figure 3.15: Validation of the shapes of the roofs in the *CityGML* model with the use of Hausdorff distance computations between the roof class of the *MVSM* model and the roof surface of the *CityGML* model.

**Heuristic rule 2 - Step 3:**

If the Hausdorff distance, *HD*, between a *MVSM* and a *CityGML* model that correspond to the roof, is resulted as bad matching of the shape of the roof, based on the categories shown in Table 3.1, then an indication is given for a wrong roof shape on the specific building in the corresponding *CityGML* model.

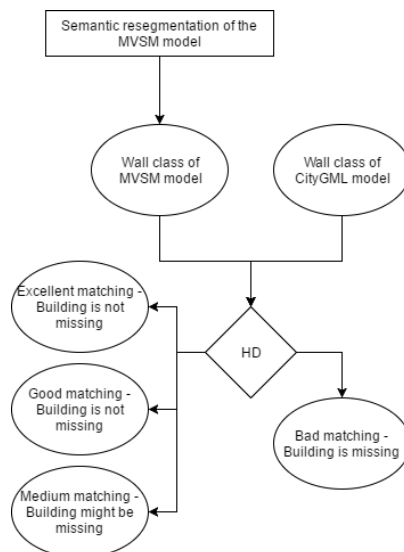


Figure 3.16: Validation of the absence of buildings in the *CityGML* model with the use of Hausdorff distance computations between the wall class of the *MVSM* model and the wall surface of the *CityGML* model.

**Heuristic rule 3 - Step 3:**

If the Hausdorff distance,  $HD$ , between a *MVSM* and a *CityGML* model that correspond to the wall, is resulted as bad matching, based on the categories shown in Table 3.1, then an indication is given for a possible missing building in the *CityGML* model.

In the validation processes shown in Figure 3.15 and in Figure 3.16, the different quality of the matching, that indicates a different shape of roof on a building between the two meshes and the absence of a building, are divided into the same four different categories that correspond to the intervals of Hausdorff distances that are used for the quality control of the matching of the faces between the two 3D city models (see Table 3.1).

### 3.5 STEP 4: PROVIDE TEXTURE TO THE CITYGML MODEL

The last step of the proposed methodology is related to the transfer of texture from the *MVSM* model to the *CityGML* model (8<sup>th</sup> sub-question from Section 1.2.1). In this part, the buildings of the two 3D city models that are assessed, in the previous step, to have excellent, good or medium matching (Section 3.4), are the ones that are proceeding to this step of the methodology. The following flowchart describes the process that is proposed for the achievement of the texturing of the *CityGML* model:

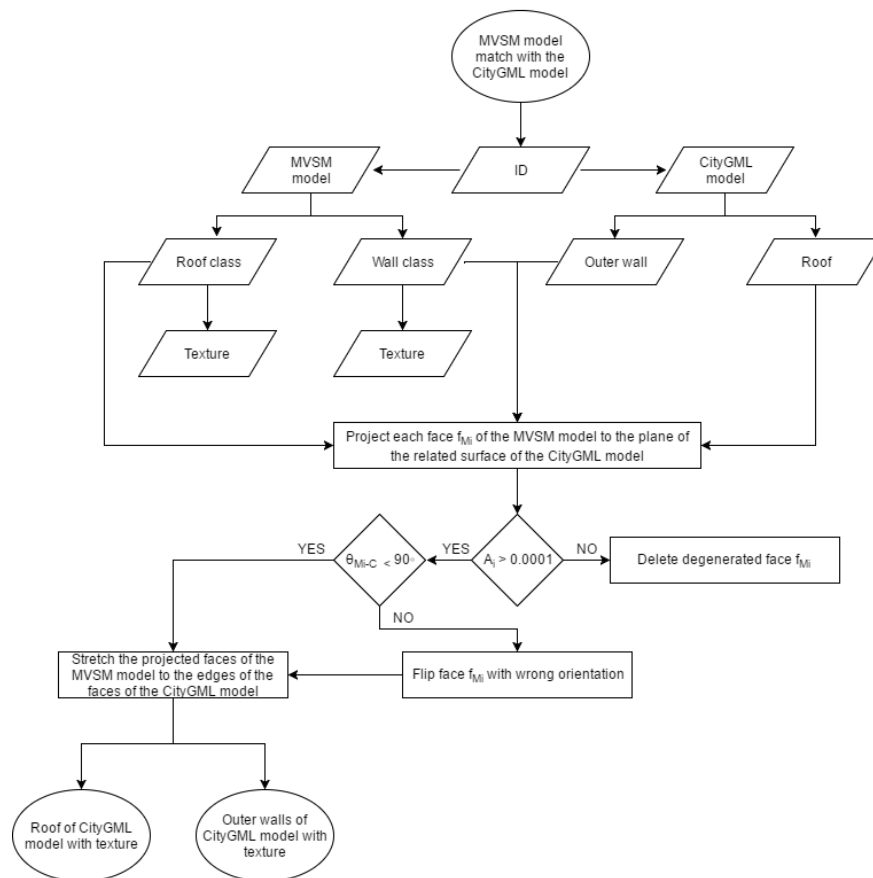


Figure 3.17: Flowchart illustrating the process of the texturing of the *CityGML* model with information from a *MVSM* model.

First of all, the faces of the two 3D city models that correspond to a building (same ID) are selected. The classes roof and wall of this building that correspond to the different 3D city models are used. For the achievement of the texturing of the CityGML model, the roof and the wall surfaces follow the same process separately (see Figure 3.17). The process starts with the projection of each face  $f_{Mi}$  of the MVSM model on the plane of the related surface of the CityGML model. The method that is used for the projection of a face, is the computation of the closest point from a point to a triangle, the CPA (described in Section 3.2). Each point of each face  $f_{Mi}$  of the MVSM model is mapped to the closest point on the plane of the CityGML surface. After the projection, the coordinates of the points of each face  $f_{Mi}$  are changing to the coordinates of the closest point. An example of the mapping is illustrated in the following figure:

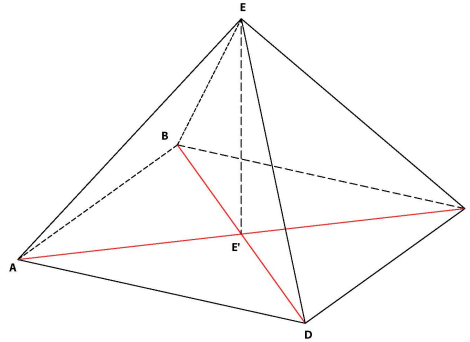


Figure 3.18: Example of projection of the triangles ABE, EBC, CDE and EDA to the ABCD plane surface.

According to Figure 3.18, the triangles ABE, EBC, CDE and EDA are projected to the plane surface ABCD. The closest points of the points A, B, C, D and E to the plane ABCD are respectively the A, B, C, D and E' points. Therefore, the new triangles that are created, are the ABE', E'BC, CDE' and E'DA.

After the mapping of the faces of the MVSM model to the plane surface of the related faces of the CityGML model, some degenerated faces and some faces with opposite orientation, are created. The cause of these problems is linked with the big amount of polygons that exist in a MVSM model, especially in the parts of the windows, the doors and the corners of the roofs. In order to delete the unnecessary faces, two heuristic rules are created:

**Heuristic rule 1 - Step 4:**

*If and only if the area  $A_i$  of a face  $f_{Mi}$  is bigger than 0.0001 ( $A_i > 0.0001$ ), the face  $f_{Mi}$  is not deleted.*

**Heuristic rule 2 - Step 4:**

*In case the angle  $\theta_{Mj-C}$  between the normal vector of a face  $f_{Mi}$  of the MVSM model and the normal vector of the related face of the CityGML model  $f_C$ , is lower than  $90^\circ$  ( $\theta_{Mj-C} < 90^\circ$ ), the orientation of the face is correct. The faces that correspond to a different angle, are flipped to the opposite orientation.*

The area of a triangle is computed with the use of Heron's formula (Vince, 2006). An example of a triangle and of the related equations needed for the computation of the area, are provided below.

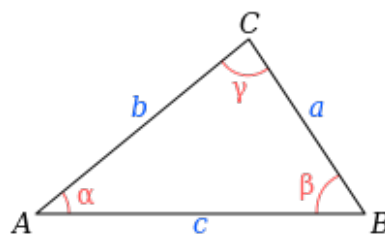


Figure 3.19: Example of a triangle that is used for the computation of its area with the use of Heron's formula.

The area  $A$  given from the Heron's formula is equal to:

$$A = \sqrt{s \cdot (s - a) \cdot (s - b) \cdot (s - c)} \quad (3.18)$$

$$s = \frac{a + b + c}{2} \quad (3.19)$$

In order to compute the lengths  $a$ ,  $b$  and  $c$ , the Euclidean distance (Section 3.2) is used:

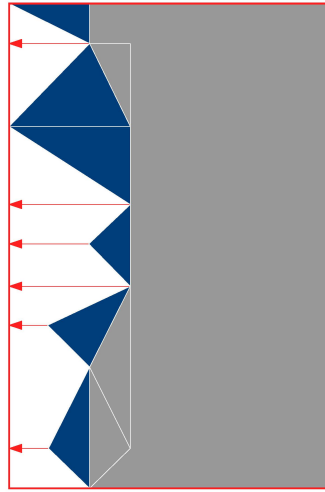
$$a = \sqrt{(x_B - x_C)^2 + (y_B - y_C)^2 + (z_B - z_C)^2} \quad (3.20)$$

$$b = \sqrt{(x_C - x_A)^2 + (y_C - y_A)^2 + (z_C - z_A)^2} \quad (3.21)$$

$$c = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2} \quad (3.22)$$

Moreover, the angle  $\theta_{M_j-C}$  between two vectors of two different faces can be computed, as it is already described in Section 3.3.

After the abstraction of the faces that correspond to degeneralized faces and after the flip of the faces with wrong orientation (*Heuristic rule 1 - Step 4* and *Heuristic rule 2 - Step 4*), some projected faces of the **MVSM** model are not parallel to the boundary line of the faces of the **CityGML** model. In order to fix that, a stretch of these faces needs to be performed. An example illustrating this problem is shown below:



**Figure 3.20:** Stretch of the faces of the MVSM model, in order to reduce the mismatches on the boundaries of the CityGML model.

According to the above figure, the faces of the **MVSM** model that have one or more edges that are not connected between other faces of the mesh (blue triangles), are the faces that are at the boundary of the connected **MVSM** mesh (grey area). The stretching of these faces (blue triangles) is done by projecting each one of the two vertices of the edges that do not touch other faces of the mesh, to the closest point (red arrow) found on the boundary line (red line) of the **CityGML** model that is closer to these vertices. The coordinates of these vertices change to the coordinates of the closest point (red arrow) located on the boundary line (red line). The method that is used for the projection of a face, is the computation of the closest point from a point to an edge, the **CPA** (described in Section 3.2). Each point of each face  $f_{M_i}$  of the **MVSM** model is mapped to the closest point on the closest edge of the **CityGML** surface. It is important

to add that these coordinates are replaced to all the different faces that these vertices appear (Wang et al., 2004).

It should be noticed that an edge that is shared by two triangles is recorded in an opposite direction in the records of two adjacent faces. Therefore, during the search of the triangles of the *MVSM* model in order to find the faces that are on the boundary of the mesh, the searching of the edges of one triangle to the edges of all the other triangles needs to be conducted oppositely. An example that shows the topology of adjacent triangles is depicted in the following figure:

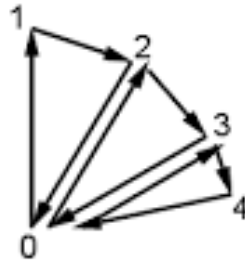


Figure 3.21: Topology of adjacent faces.

Finally, the roof and the outer wall surfaces of the *CityGML* model are enriched with the texture information provided by the related existing *MVSM* model.



# 4

## IMPLEMENTATION AND ANALYSIS

This chapter includes the implementation of the proposed methodology presented in Chapter 3. Firstly, the tools and the data that are used for this venture are analysed in the Sections 4.1 and 4.2 respectively. The presented data, including the two 3D city models, the CityGML and the MVSM model, is used for the application of the methodology. The implementation of the proposed methodology is conducted two times (Section 4.3), first for four buildings presented in Section 4.3.1 and then, for a building block that is analysed in Section 4.3.2.

### 4.1 TOOLS

In this section, the tools that are used in this thesis, are shown. More specifically the programs that are used during this thesis, are provided in Section 4.1.1. Section 4.1.2 addresses the python packages that are used for the automation of the process of the proposed methodology.

#### 4.1.1 Software packages

In this thesis, the following software packages are used for the creation of CityGML models, and for conversion and visualisation purposes:

- **CloudCompare**: Software used for the visualisation of the MVSM models.
- **FME**: Software used for the file format conversions of the 3D city models and the creation of the LOD1 building block dataset of the CityGML model (see Figure 4.3).
- **MeshLab**: Software used for the visualisation of the 3D city models.

#### 4.1.2 Python packages

In this thesis, the Python scripting language is used for the automation of the bidirectional enrichments between CityGML and MVSM models, according to the proposed methodology presented in Section 3. The packages that have been used, are shortly listed and described below:

- **lxml**: Library for processing XML and HyperText Markup Language (HTML) in the Python language.
- **Numpy**: Package for scientific computing with Python.
- **matplotlib**: Python 2D plotting library.
- **cgal-bindings 0.0.7**: The Computational Geometry Algorithms Library (CGAL) bindings project allows the use of some packages of CGAL in languages other than C++, as for example Java and Python. The bindings are implemented with SWIG.

The CGAL packages that are used are:

- **2D and 3D Linear Geometry Kernel**, use of 3D point, 3D segment and 3D triangle objects
- **3D Fast Intersection and Distance Computation (AABB Tree)**, use of closest point and squared distance functions

## 4.2 DATA

In this section, the data that is provided for this thesis is related to the two 3D city models, a [CityGML](#) and a [MVSM](#) model. The two datasets are described in Section 4.2.1 and Section 4.2.2, respectively. The datasets that are used, correspond to an area in Amsterdam, which is  $\approx 720000.000 \text{ m}^2$  and includes 655 buildings. The [CityGML](#) model has less than 1 point per  $\text{m}^2$ , while the [MVSM](#) model has  $\approx 60$  points per  $\text{m}^2$ . The location of the given area, regarding Amsterdam, is illustrated in Figure 4.1.

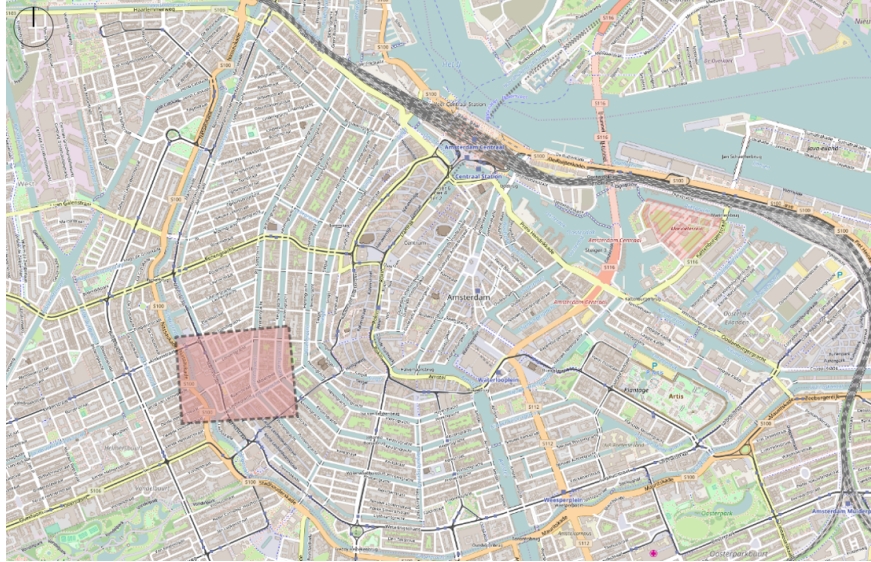


Figure 4.1: Study area in Amsterdam (pink box).

Both 3D city models are provided in the same Coordinate Reference System (CRS), which is the European Petroleum Survey Group (EPSG):28992 Netherlands, Amersfoort RijksDriehoekstelsel (RD) 2008 datum, New System.

The proposed methodology that is presented in Section 3 is implemented in Section 4.3 for two different implementations. The 1<sup>st</sup> implementation is on a small area of four buildings for a [MVSM](#) model and a [CityGML](#) model of [LOD2 CityGML](#) standard. Next, the 2<sup>nd</sup> implementation of the proposed methodology is for a larger area of a block of buildings containing nine buildings. This implementation is conducted for a [MVSM](#) model and a [CityGML](#) model of [LOD1 CityGML](#) standard and then, at the same block of buildings for a [MVSM](#) model and a [CityGML](#) model of [LOD2 CityGML](#) standard in order to proceed into comparison on the results of the different LODs after the implementation of the proposed methodology. Statistics on the different datasets of the study area, where the proposed methodology is implemented, are provided in Table 4.1.

Dataset	Number of vertices in CityGML model	Number of vertices in MVSM model	Area ( $\text{m}^2$ )
4 buildings (LOD2 CityGML - MVSM)	62	81132	$\approx 1222.406$
Building block (LOD1 CityGML - MVSM)	907	737745	$\approx 14247.427$
Building block (LOD2 CityGML - MVSM)	967	737745	$\approx 14247.427$

Table 4.1: Statistics (number of vertices, area) of the different datasets that are used for the implementation of the proposed methodology.



#### 4.2.1 CityGML input data

The study area of the CityGML model that is used for this thesis (area in Amsterdam depicted in Figure 4.1) is provided in a XML file format in LOD1 and in LOD2 CityGML standard (more details on the theory in Section 2.1).

The LOD1 is created in FME software with the use of the Basic register for Addresses and Buildings (BAG) that contains the 2D footprints of the buildings. Moreover, for the heights of the objects of the 3D city model, the DTM and the DSM from the Actual Height model of the Netherlands (AHN)<sup>3</sup> are used. The LOD2 is created from virtualcitySYSTEMS with the use of the BuildingReconstruction software. virtualcitySYSTEMS created the CityGML model with the use of the BAG that contains the 2D footprints of the buildings, the DTM and the DSM from the AHN3 and finally, an orthophoto of the area (see Figure 4.2) (Kada and McKinley, 2009). Examples of the GIS data that are used for the creation of the study area in LOD1 and in LOD2 are depicted in Figure 4.2.

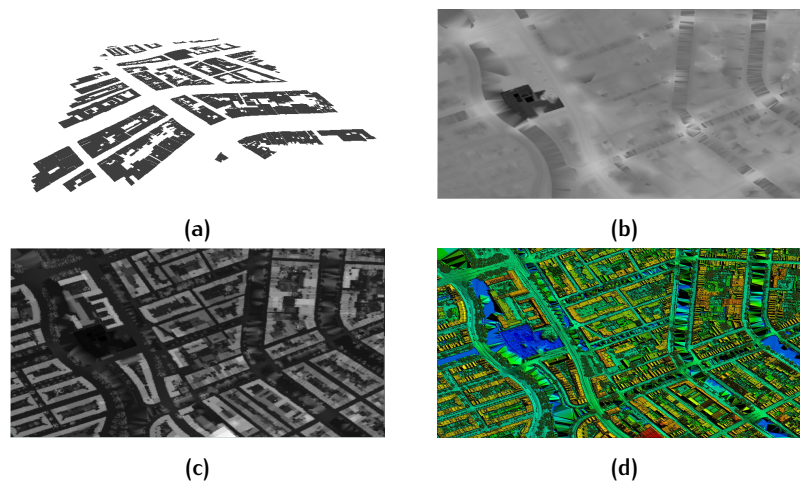


Figure 4.2: GIS data used for the creation of the CityGML datasets. (a) 2D building footprints. (b) DTM. (c) DSM. (d) Orthophoto.

The LOD1 and the LOD2 of the CityGML models of the study area that is depicted in Figure 4.1, are illustrated in Figures 4.3 and 4.4 respectively. The differences of the two levels are related to the roof structures that in LOD1 are depicted as flat roof structures and in LOD2, as differentiated roof structures. These differences are shown in detail in the black boxes, in the following figures.

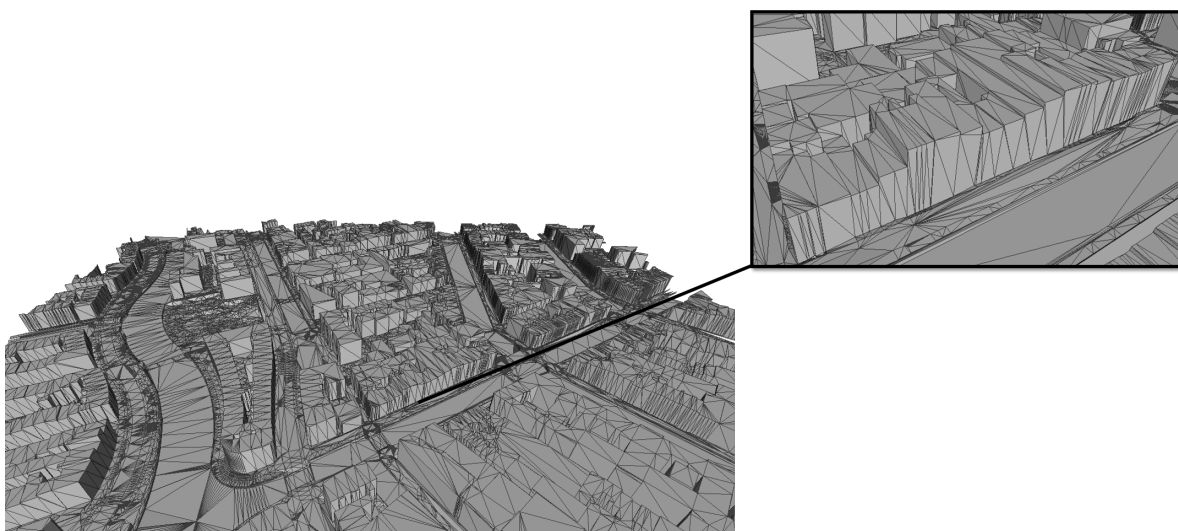


Figure 4.3: Dataset of the study area in LOD1 CityGML model.

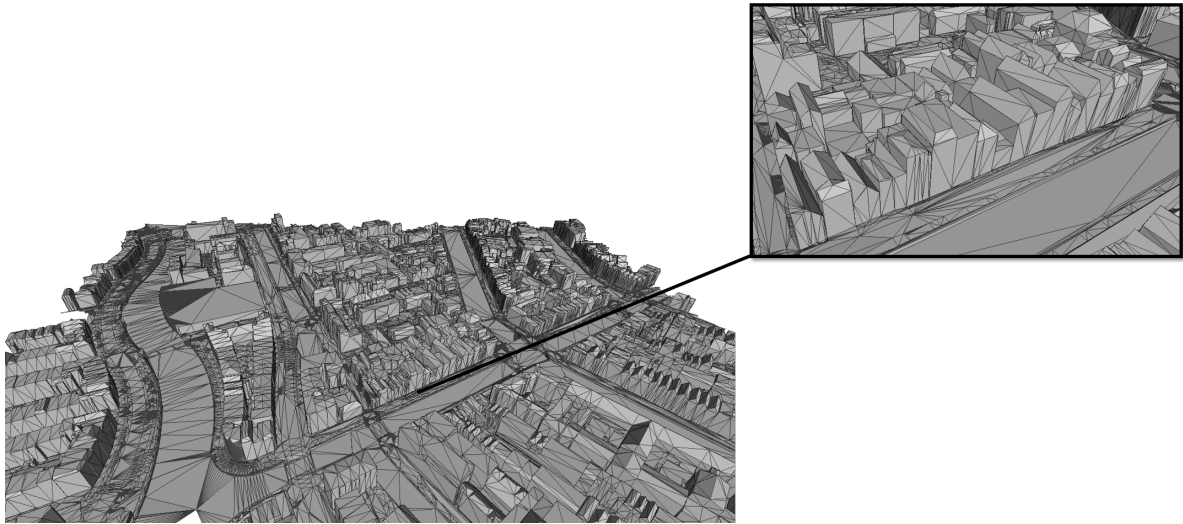


Figure 4.4: Dataset of the study area the LOD2 CityGML model.

The datasets of the study area that are mentioned in Table 4.1, are presented in Figure 4.5. The red buildings indicate the four buildings that are used during the 1<sup>st</sup> implementation of the proposed methodology in LOD2 CityGML standard (Section 4.3.1) and the white box indicates the building block (nine buildings) that is used for the 2<sup>nd</sup> implementation of the proposed methodology once, in LOD1 and then, in LOD2 CityGML standard (Section 4.3.2).

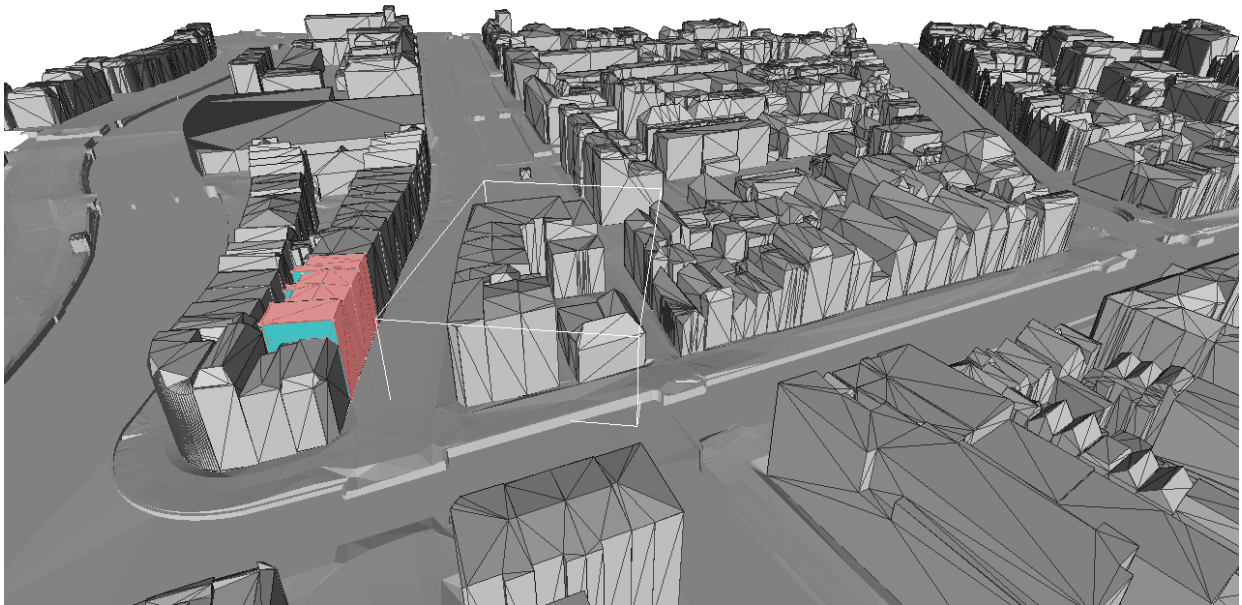


Figure 4.5: The study area of the 1<sup>st</sup> implementation of the proposed methodology (red buildings illustrating the four buildings) and of the 2<sup>nd</sup> implementation of the proposed methodology (white box illustrating the building block).

### Testing dataset of the CityGML model for the 1<sup>st</sup> implementation of the proposed methodology

The testing dataset of the 1<sup>st</sup> implementation of the proposed methodology (Section 4.3.1) is taken from the LOD2 CityGML model (see Figure 4.4 and Figure 4.5). It is illustrated in Figure 4.6, it is composed of four buildings and its characteristics are previously presented in Table 4.1.

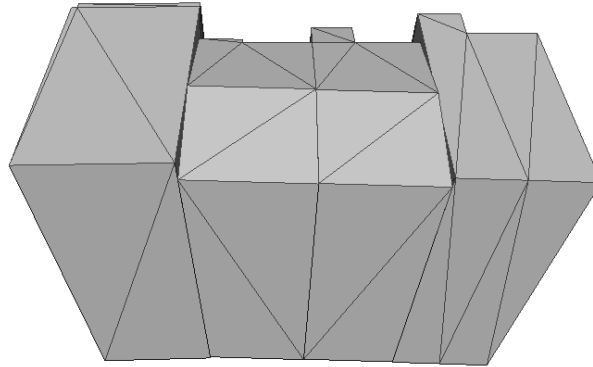


Figure 4.6: Testing dataset of the LOD2 CityGML model (four buildings).

The four buildings of the testing dataset contain the semantic information (roof, wall and ground), while at the same time the faces of the CityGML model that correspond to the same building, contain the same ID. According to Figure 4.6, from the left to the right, the buildings are referring to  $B_1$ ,  $B_2$ ,  $B_3$  and  $B_4$ , respectively. The different IDs of the four building are provided in Table 4.2.

Building name	Building ID
$B_1$	BID363100012174364
$B_2$	BID363100012174363
$B_3$	BID363100012174368
$B_4$	BID363100012174367

Table 4.2: IDs of the buildings of the CityGML testing dataset used on the 1<sup>st</sup> implementation of the proposed methodology (see Figure 4.6).

### Testing dataset of the CityGML model for the 2<sup>nd</sup> implementation of the proposed methodology

In the 2<sup>nd</sup> implementation of the proposed methodology (Section 4.3.2) the building block that is chosen from the study area (see Figure 4.5) is taken from the LOD1 (see Figure 4.3) and LOD2 (see Figure 4.3) CityGML models that are created for this thesis. The characteristics of these testing datasets are previously presented in Table 4.1. The reason why this building block is chosen from the study area is the fact that it contains buildings with different characteristics, such as adjacent buildings, building on street crossing, building with no adjacent buildings etc. Apart from the building information, this building block contains the surfaces of the road and the terrain, which are tested during the 2<sup>nd</sup> implementation (Section 4.3.2), for the exploration of the results of the automatic proposed methodology of bidirectional enrichment between MVSM and CityGML models. The LOD1 and LOD2 CityGML models of the building block that are used for the 2<sup>nd</sup> implementation of the proposed methodology in Section 4.3.1, are presented respectively in Figures 4.7 and 4.8.

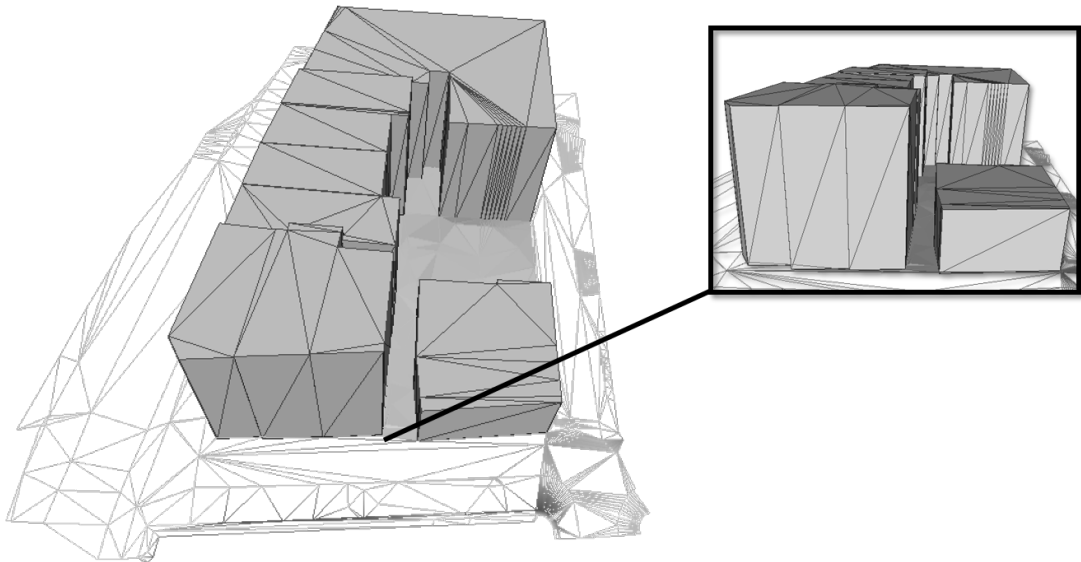


Figure 4.7: Testing dataset of the LOD<sub>1</sub> CityGML model (building block), representing buildings (grey triangulated surfaces), road (transparent triangulated surfaces) and terrain information (grey non-triangulated surfaces).

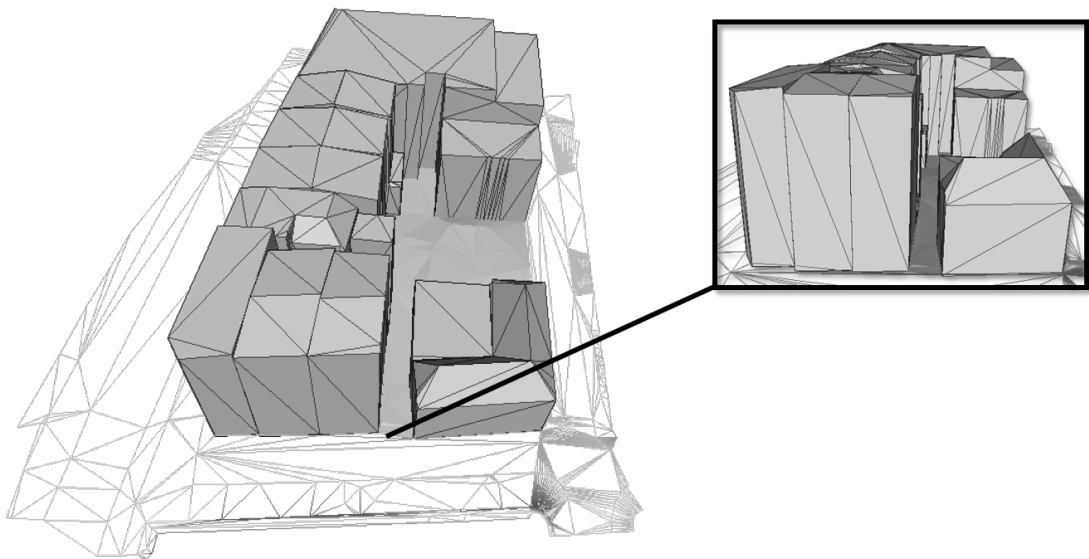


Figure 4.8: Testing dataset of the LOD<sub>2</sub> CityGML model (building block), representing buildings (grey triangulated surfaces), road (transparent triangulated surfaces) and terrain information (grey non-triangulated surfaces).

The building block of the testing dataset contains the semantic information (roof, wall, road and terrain), while at the same time the faces of the CityGML model that correspond to the same building, contain the same ID. According to Figures 4.7 and 4.8, from the bottom right and clockwise, the buildings are referring to  $B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8$  and  $B_9$ , respectively. The different IDs of the nine building are provided in the Table 4.3.

Building name	Building ID
B1	BID363100012169866
B2	BID363100012169865
B3	BID363100012169869
B4	BID363100012169868
B5	BID363100012169861
B6	BID363100012169867
B7	BID363100012167463
B8	BID363100012169936
B9	BID363100012169937

Table 4.3: IDs of the nine buildings of the CityGML testing dataset used on the 2<sup>nd</sup> implementation of the proposed methodology (see Figures 4.7 and 4.8).

#### Problem in the CityGML dataset

According to [virtualcitySystems \(2016\)](#), the height information of the ground surface of the buildings is taken from the [DTM](#). This guarantees that the buildings perfectly sit on the terrain model, rather than floating over or sinking into it.

It is found that in the provided [LOD2 3D city model](#) (see Figure 4.4), the above statement is not true. The heights of the ground surface of the given [CityGML](#) model are around  $h_g=0.9m$ , while the heights from the [DTM](#) of the same area, are around  $h_{DTM}=2.0m$ . Therefore, the given [CityGML](#) model is sinking into the terrain model. In order to fix the heights of the [LOD2 CityGML](#) model, the heights of the buildings that derived from the creation of the [LOD1 CityGML](#) model (see Figure 4.3) are used. In Figure 4.9, the differences of the heights of the [CityGML](#) model between the model that was provided from [virtualcitySYSTEMS](#) (grey) and the one that is corrected (red) are depicted.

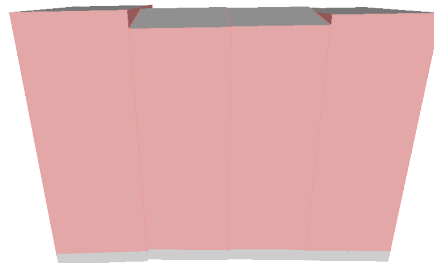


Figure 4.9: Height differences between the provided LOD2 CityGML model (grey) and the one that was corrected (red).

#### 4.2.2 MVSM input data

The [MVSM](#) model is given for the same study area in Amsterdam, as the [CityGML](#) model (see Figure 4.1). It is provided by [CycloMedia](#) in [COLLADA](#) file format for seven different [LODs](#). The [LODs](#) that are given, are numbered in the opposite direction of what is standard. For example, [LOD0](#) is created from the original deliverables and contains  $\approx 60$  points per  $m^2$ . However, each [LOD](#) numbered higher, was simplified and smoothed. As a result, the data that remained in [LOD6](#) contains less than 1 point per  $m^2$ . Examples of the different levels are shown in Figure 4.10. For this thesis, the [LOD0](#) is used, but the other [LODs](#) can be used



for future improvements and testings of the proposed methodology (Chapter 3). In this way, it would be interesting to explore the different correspondences between the CityGML model and the smoother and more simplified LODs of the MVSM model.

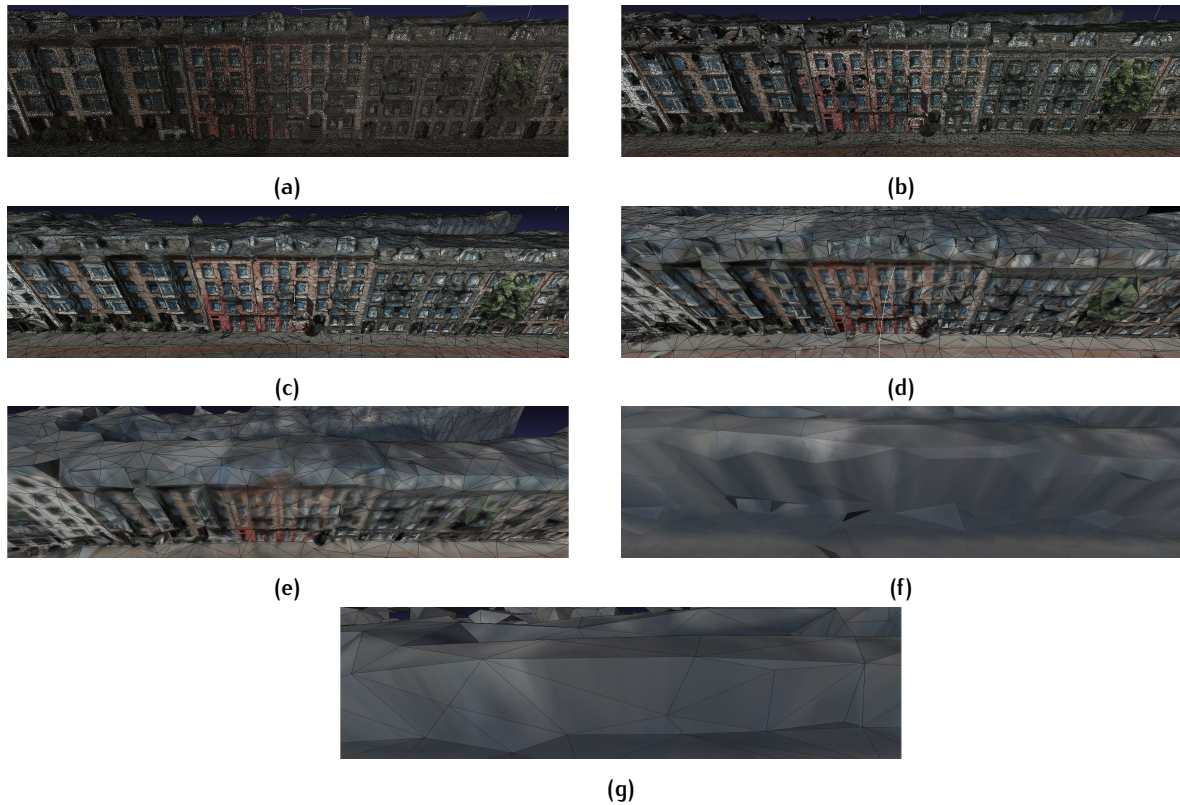


Figure 4.10: Levels of detail of the MVSM. (a) LOD0. (b) LOD1. (c) LOD2. (d) LOD3. (e) LOD4. (f) LOD5. (g) LOD6.

#### *Testing dataset of the MVSM model for the 1<sup>st</sup> implementation of the proposed methodology*

The testing dataset used during the 1<sup>st</sup> implementation of the proposed methodology is illustrated in Figure 4.11. It is composed of four buildings and its characteristics are previously presented in Table 4.1.



Figure 4.11: Testing dataset of the MVSM model (four buildings).

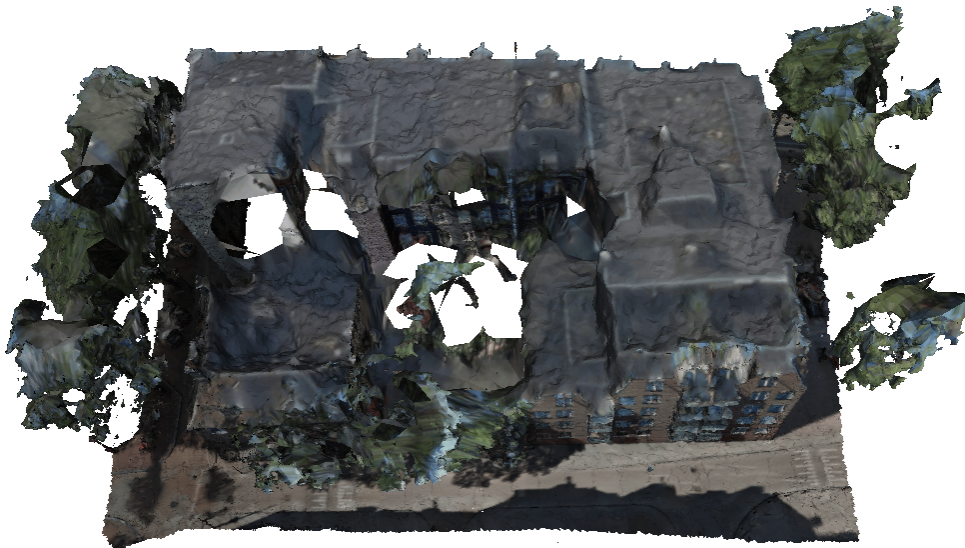
The features shown in Figure 4.11 depict the same features with the testing area of the CityGML model, shown in Figure 4.6.

### Testing dataset of the MVSM model for the 2<sup>nd</sup> implementation of the proposed methodology

In the 2<sup>nd</sup> implementation of the proposed methodology the building block that is chosen from the study area (see Figure 4.5) is chosen from the given MVSM model. The characteristics of the testing dataset is previously presented in Table 4.1. The building block of the MVSM model that is used during the 2<sup>nd</sup> implementation of the proposed methodology is illustrated in Figure 4.12.



(a)



(b)

Figure 4.12: Testing dataset of the MVSM model (building block). (a) West side of the building block. (b) East side of the building block.

### Quality of the MVSM model

The quality of the mesh of the MVSM model is related with the amount of non-manifold errors, border edges, invalid orientation of faces, singularities, self-intersections, degenerated elements, spikes etc. In the MVSM model that is used in this thesis, the focus of the exploration of the quality of the visioned mesh is on the non-manifold errors. Non-manifold errors are related to two types of errors, the holes and the unwanted faces (Cirak and Long, 2011). Unwanted faces are the extra faces or edges in a model. With



the use of **MeshLab** software, the amount of the holes and the amount of the non-manifold vertices and edges, plus the number of faces over non-manifold vertices and edges, are presented in the Table 4.4.

Dataset	Holes	Non-manifold vertices	Faces over non-manifold vertices	Non-manifold edges	Faces over non-manifold edges
4 buildings	115	99	752	276	905
Building block	509	2531	19957	3793	13032

Table 4.4: Non-manifold errors that appear in the testing datasets of the MVSM model.

Examples of the non-manifold errors that appear in the testing datasets of the given **MVSM** model are depicted in the Figures 4.13, 4.14 and 4.15.

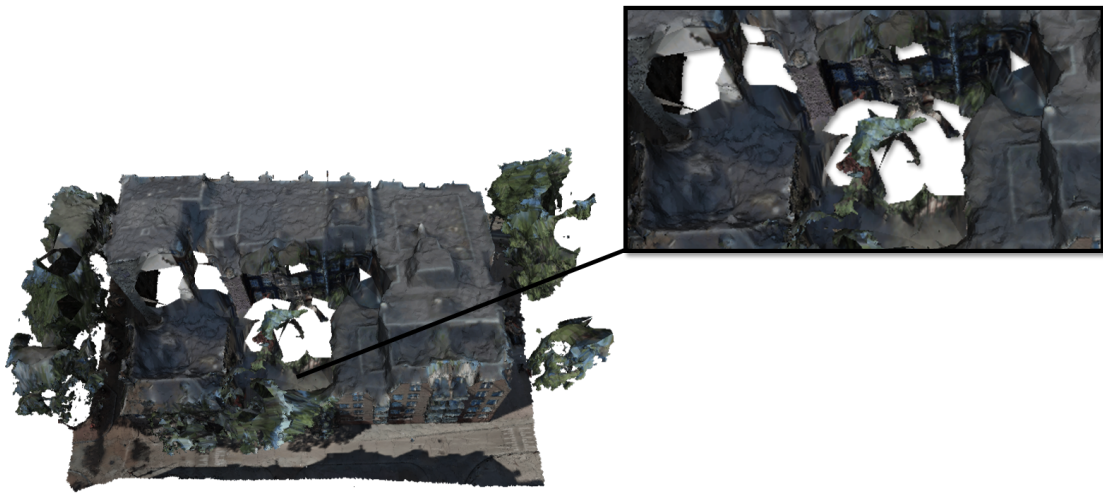


Figure 4.13: Holes on the testing dataset of the MVSM model (building block).

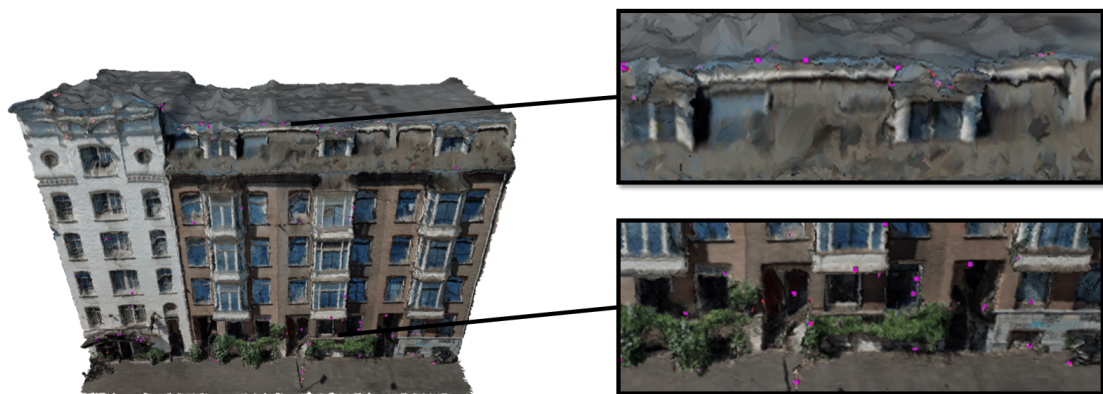


Figure 4.14: Non-manifold errors (pink dots) on the testing dataset of the MVSM model (four buildings).



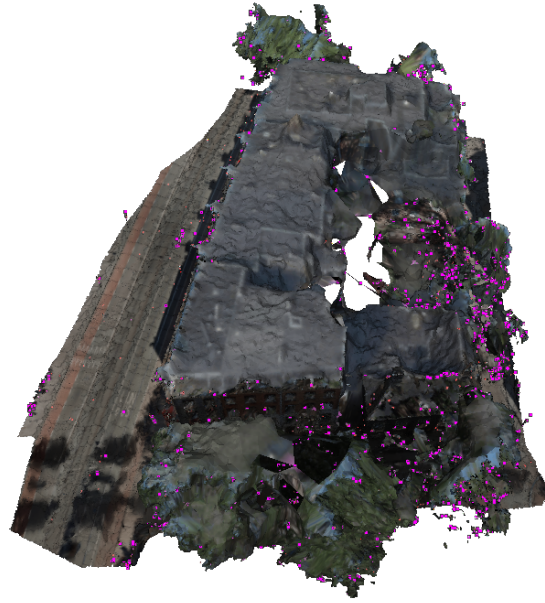


Figure 4.15: Non-manifold errors (pink dots) on the testing dataset of the MVSM model (building block).

## 4.3 IMPLEMENTATION AND RESULTS

The tools and the data presented in the previous Sections 4.1 and 4.2 respectively, are used for the implementation of the proposed methodology (see Figure 3.1). The implementation of the methodology is conducted twice. At the 1<sup>st</sup> implementation (Section 4.3.1), the small area of four buildings of the MVSM model (see Figure 4.11) and of the CityGML model in LOD2 CityGML standard (see Figure 4.6) are used. Next, the 2<sup>nd</sup> implementation of the proposed methodology is conducted for a larger area of a block of buildings containing the nine buildings, the road and the terrain information, presented in Section 4.2 (see Figure 4.5). This implementation is performed for the MVSM model (see Figure 4.12) and the CityGML model of LOD1 CityGML standard (see Figure 4.7) and then, at the same block of buildings for the MVSM model (see Figure 4.12) and the CityGML model of LOD2 CityGML standard (see Figure 4.8). Comparisons on the results that derive from the different LODs during the implementation of the proposed methodology, are presented in this part. The results of the four basic steps of the proposed methodology are provided separately in every implementation. In each step of the implementation, the results are thoroughly analysed.

### 4.3.1 1<sup>st</sup> Implementation of the proposed methodology

The 1<sup>st</sup> implementation of the proposed methodology is related to the implementation of the methodology presented in Section 3 and the use of the testing dataset that is linked to the four buildings in the MVSM model (see Figure 4.11) and in the LOD2 CityGML model (see Figure 4.6). The semantic segmentation is resulting into three classes, the roof, the wall and the uncertain, with texture colors red, blue and green, respectively. In Figure 4.16 a legend of the colors that will appear during the 1<sup>st</sup> implementation is depicted.

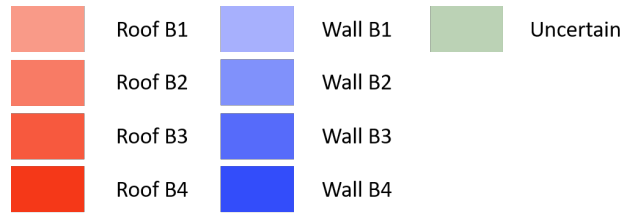


Figure 4.16: Legend of the classes that derive from the semantic segmentation of the 1<sup>st</sup> implementation.

### Step 1

First of all, the correspondence of the faces of the CityGML model with the faces of the MVSM model is tested. As it is already mentioned in Section 3.2, the MVSM model does not correspond to all the faces of the CityGML model, because it provides only the data that is captured from the aerial and the terrestrial imagery (Section 2.2). The differences of the faces between the testing datasets of the CityGML (see Figure 4.6) and the MVSM (see Figure 4.11) models of the same features are presented in Figure 4.17.

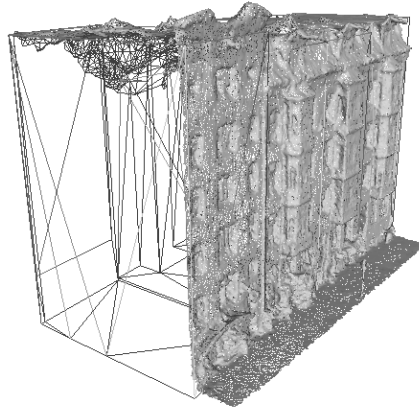


Figure 4.17: Differences between the corresponding surfaces of the testing datasets of the CityGML (transparent) and the MVSM (grey) models of the same features.

From the above figure it can be easily observed that the roof surfaces of the CityGML model and the surfaces of the CityGML model that correspond to the walls of the facades of the four buildings, have a correspondence with the MVSM model of the same entities. Therefore, the ground and the shared wall faces of the CityGML model are removed, based on the proposed methodology presented in Figure 3.4, in Section 3.2.

More specifically, the centroids of all the faces of both 3D city models are computed. The minimum distances between the centroids of the faces of the CityGML model and the centroids of the faces of the MVSM model are calculated. These distances are compared with a threshold  $T$ , equal to  $0.3m$  ( $T = 0.3m$ ). The comparison of the distances with this threshold  $T$  provides an indication to the faces of the CityGML model that are close enough to the corresponding MVSM model, to proceed on the matching with the MVSM faces. After the above rule (*Heuristic rule 1 - Step 1*), for several faces of the CityGML model that are part of the same facade of a building and happened to have the centroid of their face at a distance bigger than the threshold  $T$ , the second heuristic rule presented in Section 3.2 is used (*Heuristic rule 2 - Step 1*) and the faces of the CityGML model that are removed from the testing dataset (see Figure 4.6) are illustrated in Figure 4.18.

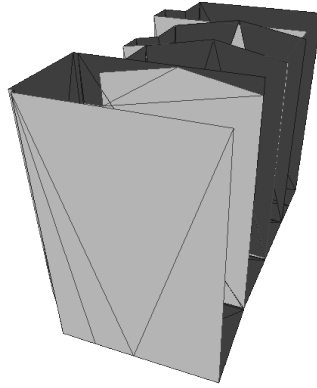


Figure 4.18: Faces that are removed from the CityGML model.

The remaining faces (roof and outer walls) of the CityGML model that can be matched with the MVSM model are depicted in Figure 4.19.

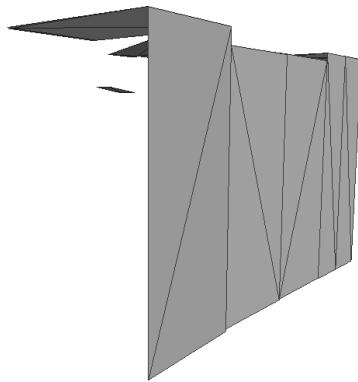


Figure 4.19: Faces that are chosen from the CityGML model.

The remaining faces depicted in Figure 4.19, are the faces that were expected to be remained from the implementation of this step of the methodology on the testing dataset (see Figure 4.6). The same success cannot be expected, if the same step of the methodology is implemented in a different dataset. More tests have to be conducted in different building types, in order to check the correctness of the decided threshold  $T$ . The reason for that is that each building has different characteristics that can affect the result. For example, if a building contains large window and door recesses, specifically in the parts of the MVSM model, where it happens to be around the centroid of the corresponding CityGML face, then the distance between these centroid points will be larger than the proposed threshold  $T$  (*Heuristic rule 1 - Step 1* in Section 3.2).

Furthermore, it has to be added that the use of simplified and smoothed MVSM models, such as the ones that are depicted in Figure 4.10, will have a different impact on the expected results of this step. The reason for that is that a MVSM model that is simplified and smoothed, contains less geometric details than the original MVSM model, therefore it is closer to the representation of the CityGML surfaces. In these cases, the threshold  $T$  of the *Heuristic rule 1 - Step 1*, in Section 3.2 can be lower, when the given CityGML model is either LOD1 or LOD2 (Section 2.1). In addition, the opposite case needs to be taken into consideration. For instance, if the LOD of the CityGML model is LOD3 (Section 2.1) and the MVSM model is simplified and smoothed in a high level (see Figure 4.10g), then the threshold  $T$  of the *Heuristic rule 1 - Step 1*, in Section

3.2 needs to be higher, in order to take into consideration the window and the door recesses, plus the details of the roof structures provided by the CityGML model.

In general, the use of different LODs of CityGML models and levels of simplification and smoothness of MVSM models, plus the characteristics of the different building types, can affect the result and need to be explored, in order to provide a proper threshold  $T$  and enhance this step of the proposed methodology.

### Step 2

The 2<sup>nd</sup> step of the 1<sup>st</sup> implementation of the proposed methodology is related to the segmentation of the MVSM model into three different classes (roof, wall, uncertain) with the use of the corresponding faces of the CityGML model (see Figure 4.6). The automatic matching of the faces of the two 3D city models is conducted based on the combination of distance computations, normal vector comparisons and heuristic rules, according to the step 2 of the proposed methodology, described in Section 3.3.

First of all, the faces of the CityGML model that are used for the implementation of this step (see Figure 4.19), are the ones that are chosen from the previous step (Section 4.3.1). As reported by the flowchart of the step 2 in Figure 3.9, the minimum distances between the faces of the MVSM model (see Figure 4.11) and the roof and the wall faces of the related CityGML model (see Figure 4.19) are computed. Based on the *Heuristic rule 1 - Step 2*, presented in Section 3.3, the faces of the MVSM model that are closer to the roof faces of the CityGML model are assigned as roof class. In addition, the faces of the MVSM model that are closer to the wall faces of the CityGML model and lower than a threshold  $T_d$  (see Figure 3.9), are proceeding for further investigation on whether or not these faces will be assigned as wall class. The rest of the faces of the MVSM model are assigned as uncertain (*Heuristic rule 2 - Step 2*). The result of the above process (*Heuristic rule 1 - Step 2* and *Heuristic rule 2 - Step 2*) is presented in Figure ??, where the roof class is assigned with red color, the faces of the MVSM model that are assigned as uncertain have a green color and the rest of the faces of the MVSM model are assigned with light blue color. The four different intensities of the red texture of the roof class are indicating the different unique identifiers of the buildings (see Table 4.2 in Section 4.2.1), with which the related faces of the MVSM model correspond to. In this way, the attributes that correspond to the specific buildings in the CityGML model are transferred to the related buildings in the MVSM model.

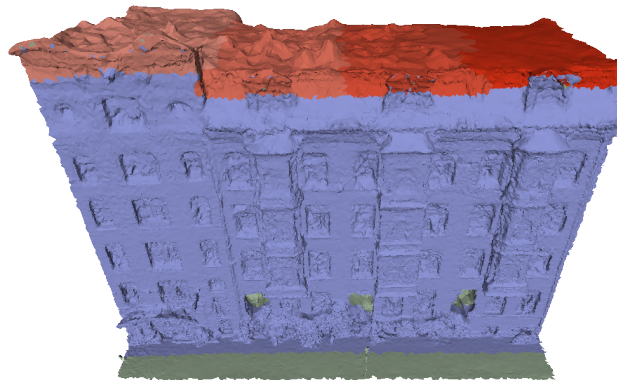
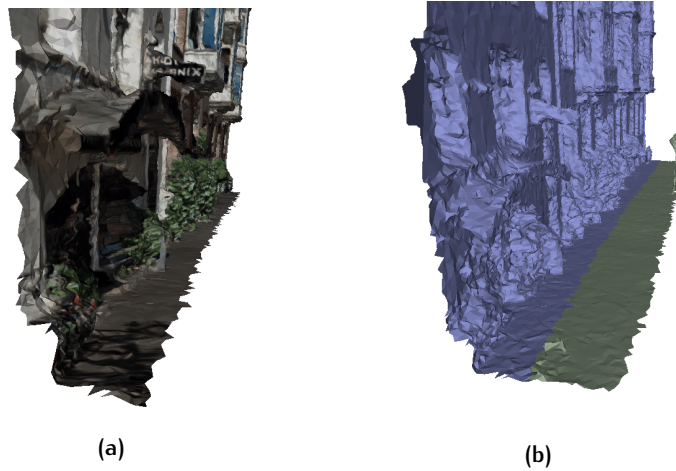


Figure 4.20: Result of the step 2 of the implementation of the proposed methodology, after the *Heuristic rule 1 - Step 2* and *Heuristic rule 2 - Step 2* (Section 3.3).

The threshold  $T_d$  of the minimum distances between the faces of the two 3D city models, which is used to assign as uncertain the faces of the MVSM model, is equal to  $1m$  ( $T_d = 1m$ ) (*Heuristic rule 2 - Step 2* in Section 3.3). This value takes into consideration an average value of the recesses and protrusions on the facade of the buildings (see Figure 4.21). In this way, windows, doors, awnings, balconies, etc. are proceeding into further investigation with the use of the *Heuristic rule 3 - Step 2* of the proposed methodology,

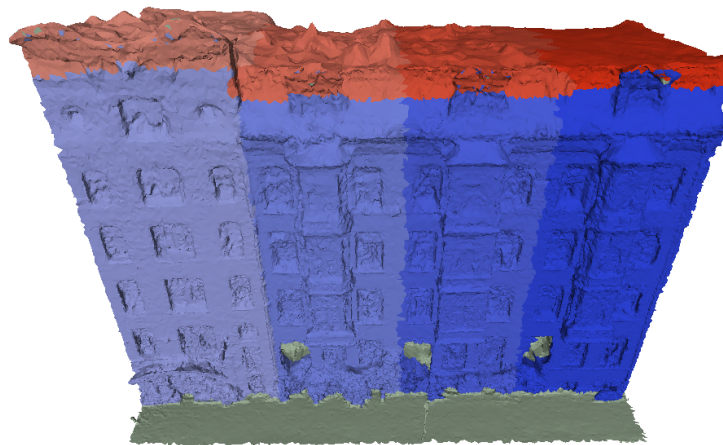
presented in Section 3.3 (light blue). Also, this value  $T_d$  protects the other thematic classes, like transportation bodies and relief features, of being assigned as wall class, into the next steps (green). An example of these characteristics of the facade of the buildings that are avoided of being assigned as uncertain (green), is illustrated in the following figure.



**Figure 4.21:** Characteristics of the facade of the buildings that are avoided of being assigned as uncertain, due to the use of the *Heuristic rule 2 - Step 2* (Section 3.3). (a) Example of the implementation of *Heuristic rule 2 - Step 2* (Section 3.3) with real texture information. (b) Example of the implementation of *Heuristic rule 2 - Step 2* (Section 3.3) with the texture representation (green) of the result provided in Figure 4.20.

From Figure 4.21 it can be observed that the characteristics of the facades of the buildings are kept for further investigation (light blue texture), while the lamp of the road is assigned as uncertain (green).

Furthermore, in order to segment the remaining faces of the *MVSM* model (light blue from Figure 4.20), into wall class and uncertain (see flowchart in Figure 3.9 in Section 3.3), the 3<sup>rd</sup> rule of the step 2 is implemented. The minimum height of the related face of the *CityGML* model is compared to a threshold  $T_h$ . Moreover, for the same face, the angle between its normal vector and the normal vector of the closest *CityGML* face is computed. The faces that are lower than the threshold  $T_h$  and at the same time have an angle of normal vectors lower than  $45^\circ$ , are assigned as uncertain, while the rest as wall class (*Heuristic rule 3 - Step 2* in Section 3.3). The result of the *Heuristic rule 3 - Step 2* is presented in Figure 4.22. As it is already mentioned, the four different intensities of the red texture of the roof class and the blue texture of the wall class are indicating the different unique identifiers of the buildings (see Table 4.2 in Section 4.2.1), with which the related faces of the *MVSM* model correspond to.



**Figure 4.22:** Result of the step 2 of the implementation of the proposed methodology, after the *Heuristic rule 3 - Step 2* (Section 3.3).



The result shown in Figure 4.22, depicts the segmentation of the *MVSM* model into the three classes (roof class (red), wall class (blue) and uncertain (green)), after the use of the *Heuristic rule 3 - Step 2* (Section 3.3). It has to be noticed that the wall class and the uncertain contain some unconnected faces. This is shown more specifically in Figure 4.23.

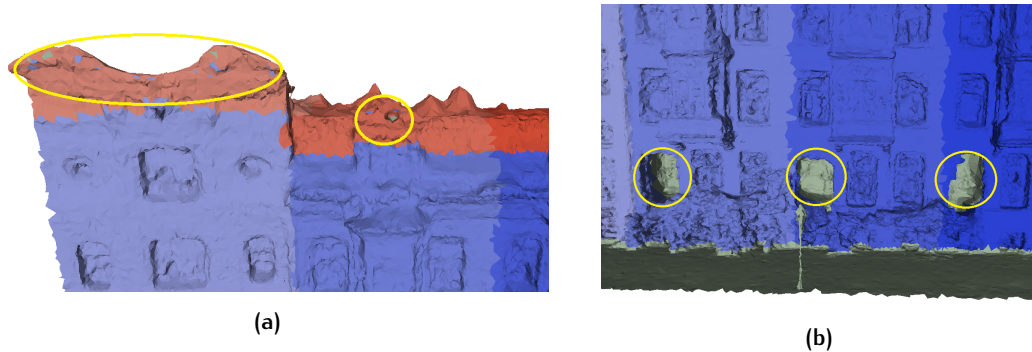


Figure 4.23: Unconnected faces assigned as wall class and as uncertain. (a) Parts of the roof class that are assigned as wall class (blue) or as uncertain (green). (b) Parts of the wall class that are assigned as uncertain (green).

According to Figure 4.23, it can be observed that there are some parts of the roof class that are assigned as wall class (blue) or uncertain (green), and some parts of the wall class that are assigned as uncertain (green). Therefore, the connectivity of the faces of the *MVSM* model is explored according to the methodology for the identification of the different connected components in a semantic class of a *MVSM* model, presented in the flowchart in Figure 3.12, in Section 3.3. The result of the implementation of this methodology is illustrated in the following figure:

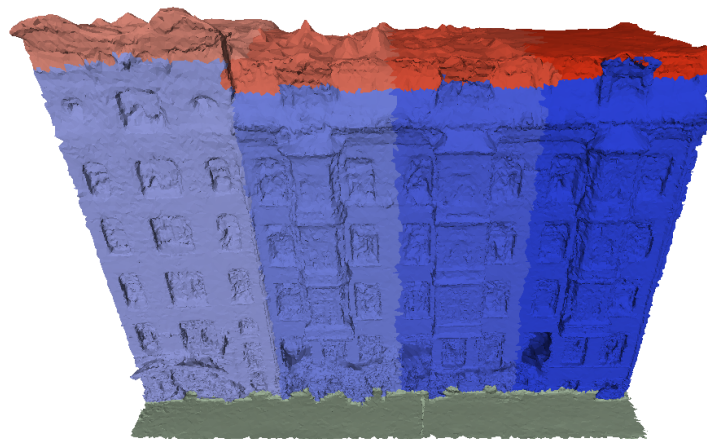
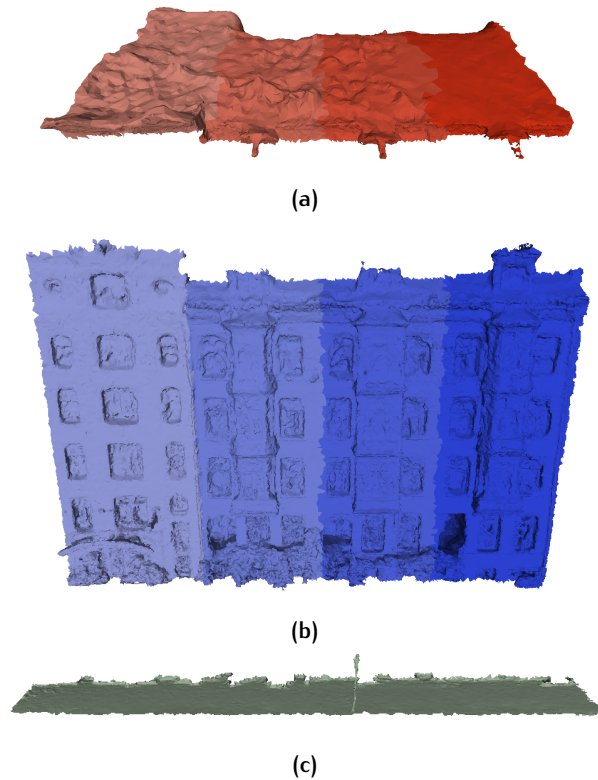


Figure 4.24: Result of the implementation of the methodology that is presented in Figure 3.12, in Section 3.3, for the identification of the different connected components in a semantic class of a *MVSM* model.

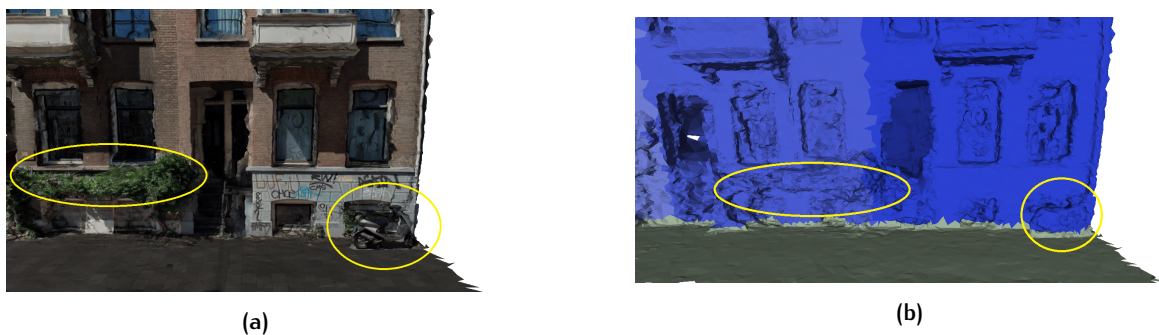
The three different classes that derive from the segmentation step of the *MVSM* model (see Figure 4.11) with the use of information from the related *CityGML* model (see Figure 4.6) of the same features are shown in Figure 4.25.



**Figure 4.25:** Result of the step 2 of the 1<sup>st</sup> implementation of the proposed methodology. (a) Faces of the *MVSM* model assigned as roof class. (b) Faces of the *MVSM* model assigned as wall class. (c) Faces of the *MVSM* model assigned as uncertain.

In general, after the implementation of this step of the proposed methodology, it can be concluded that some enhancements can be performed. It is obvious that the roof class (red) shown in Figure 4.25a, contains parts of the walls of the buildings. The reason for that is that the roof class is created based on the roof surfaces of the *CityGML* model.

Moreover, it can be observed that after the implementation of the proposed methodology, the wall class includes all the characteristics of the facades of the buildings that are close enough to the related wall surface of the *CityGML* model. These characteristics are not always related to the building facades, but it might also contain vegetation or other obstacles that should not be part of this class. An example illustrating this problem is depicted in the figure below:



**Figure 4.26:** Characteristics on the facades of the buildings that are assigned as wall class, while they are not. (a) Vegetation and obstacles on the facade of the buildings, before the segmentation step. (b) Vegetation and obstacles on the facade of the buildings, after the segmentation step assigned as wall class.

It can be concluded that further investigation needs to be done, in order to be able to improve the proposed methodology and be able to identify the differences on the characteristics of the facades of the buildings that should or should not be assigned as wall class.

Moreover, the same methodology can be extended for the identification of other thematic classes, such as relief features or transportation bodies. The reason for that is that the faces that are characterised as uncertain, are successfully removed from the buildings and can be related to the other thematic classes. The flowchart depicted in Figure 3.13 shows the process that needs to be conducted for this implementation. The performance of the *Heuristic rule 4 - Step 2*, *Heuristic rule 5 - Step 2* and *Heuristic rule 6 - Step 2* that are related to the semantic segmentation of a MVSM model into the classes road, terrain and uncertain, is not tested on the 1<sup>st</sup> implementation of the proposed methodology (Section 4.3.1), but on the 2<sup>nd</sup> implementation that is analysed in Section 4.3.2.

### Step 3

After the completion of the previous step of the proposed methodology, the MVSM model is segmented into the three classes, roof, wall and uncertain. This step of the proposed methodology is related to the quality control of the matching of the faces of the two 3D city models that is conducted in step 2, in Section 4.3.1. The two layers of the roof and the wall of each building, derived from the semantic segmentation of the MVSM model, are tested with the related surfaces, roof and wall, of the corresponding building from the CityGML model. The validation of the two classes is conducted with the use of the Hausdorff distance between the points of the two 3D city models, as it is already mentioned in Section 3.4.

Examples of the results of the minimum distances, between every point of the MVSM model (roof and wall class) and the points of the CityGML model (roof and wall surfaces), are presented for the 1<sup>st</sup> building of the testing dataset, in the following diagrams in Figure 4.27. In every diagram, the red line represents the maximum distance (Hausdorff distance) of all the minimum distances (blue dots) between the points of the two 3D city models.

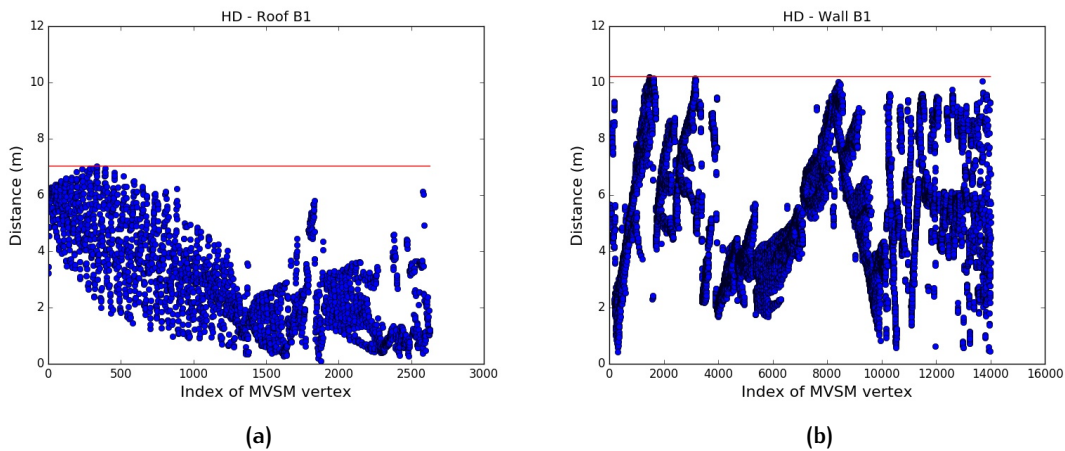


Figure 4.27: Example of the distances computed for the quality control of the segmentation of the MVSM faces to the different surfaces of the CityGML model. (a) Quality control of the roof surface of the 1<sup>st</sup> building. (b) Quality control of the wall surface of the 1<sup>st</sup> building.

It can be observed that the roof surface of the MVSM and the CityGML model have a better matching, than the faces of the wall class of the same 3D city models. This can be observed from the values of the Hausdorff distances and the variance of the different distances. The diagram of the roof class of the 1<sup>st</sup> building has less fluctuations, than the diagram of the wall class of the same building. Also, the maximum distance (red line) is higher for the wall class ( $\approx 7\text{m}$  for the roof class in Figure 4.40a and  $\approx 10\text{m}$  for the wall class in Figure 4.38b).

The results of the Hausdorff distances that derive after the quality control of the roof and wall classes of the four buildings, are compared to the proposed categorization provided in Table 3.1, in Section 3.4 and are presented in Table 4.5.



Building name	Roof class	Quality	Wall class	Quality
B1	7.027m	Good	10.209m	Medium
B2	4.221m	Excellent	9.527m	Medium
B3	4.102m	Excellent	9.458m	Medium
B4	5.819m	Good	9.399m	Medium

**Table 4.5:** Categorisation of the quality control of the semantic segmentation, based on the Hausdorff distance computations.

According to the resulted indication (excellent, good, medium or bad) of the matching of the faces, it is decided whether or not to resegment the *MVSM* model with different parameters (repeat the step 2 of the 1<sup>st</sup> implementation in Section 4.3.2). The results shown in Table 4.5 are the best results that have derived after resegmentations of the *MVSM* model with different parameters, during the previous steps of the implementation. It can be observed that the roof matching of every building is better than the wall matching. This can be explained due to the fact that the wall surfaces contain window and door recesses, which reduce the quality of the matching. In general, it can be concluded that the semantic segmentation of the previous step of the implementation of the methodology ended with a good matching result.

After the quality control of the semantic segmentation, the final indications that have derived from the Hausdorff distance of the roof and the wall of each building between the roof and the wall of the *MVSM* model and the corresponding roof and wall surface of the *CityGML* model, are presented in Table 4.5. These values are used for the quality check of the building information in the *CityGML* model. In this way, the validation of the shapes of the roofs of the buildings can take place, together with a validation considering the existence of buildings in the *CityGML* model. The results of the Hausdorff distance computations, together with the processes that are depicted in the flowcharts in Figures 3.15 and 3.16, result in indications about the matching of the shape of the roofs and the possible absence of buildings in the *CityGML* model. These results are provided in Table 4.6 and can show errors of the *CityGML* model to the user, after its comparison with a younger *MVSM* model.

Building name	Matching with the shape of the roof	Absence of building
B1	Good	No
B2	Excellent	No
B3	Excellent	No
B4	Good	No

**Table 4.6:** Indications of errors that are related to the shape of the roofs and the absence of buildings in the *CityGML* model.

The above indications show that the shapes of the roofs between the testing datasets of the *MVSM* model and the *CityGML* model that are used during the 1<sup>st</sup> implementation of the proposed methodology have either a good or an excellent matching and no further investigation needs to be done for the correction of the shapes of the roofs in the related testing dataset of the *CityGML* model. Moreover, after the validation of the models, no absence of any building is detected in the testing dataset of the *CityGML* model that is used for this implementation.

#### Step 4

In the final step of the 1<sup>st</sup> implementation of the proposed methodology, for the automatic bidirectional enrichment of the two 3D city models, the texture of the *MVSM* model is transferred to the related faces of the *CityGML* model. The segmented meshes of the *MVSM* model from step 2, with the related faces from the *CityGML* model, are used in this step.

In the previous step (Step 3), all the four buildings (roof and wall classes) of the testing dataset of the *MVSM* model, resulted with good correspondence with respect to the surfaces (roof and wall surfaces) of the testing dataset of the *CityGML* model. Therefore, all the buildings of the testing dataset are used during this step. The steps of the proposed methodology that are described in the flowchart in Figure

3.17 (Section 3.5) are used for the achievement of this venture. Every face of the *MVSM* model is projected to the corresponding plane surface of the roof or the related outer wall of the *CityGML* model (see Figure 4.19). The results of the projections of the four buildings (roof and wall classes) are shown in Figure 4.28.

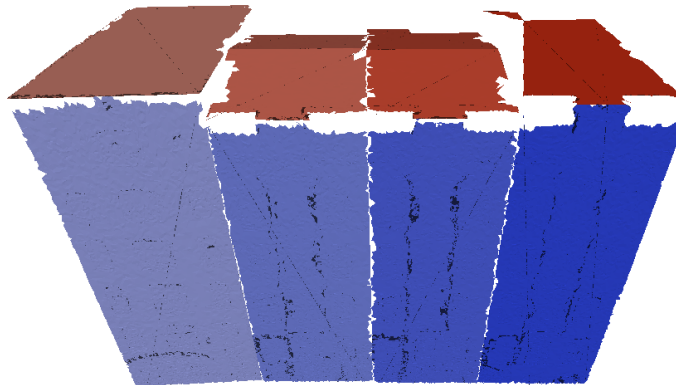


Figure 4.28: Projections of the four buildings (roof in red texture and wall in blue texture) to the corresponding plane surfaces (roof and outer walls) of the *CityGML* model (see Figure 4.19).

As it is already mentioned in Section 3.5, after the projection of the faces of the *MVSM* model, some degenerated faces and some faces with opposite orientation are expected. Such faces are mostly created on areas, where the mesh is composed of many polygons, for example on areas that are related to window and door recesses and protrusions, vegetation or other obstacles. However, in areas where the *MVSM* model is in a way parallel to the related surface of the *CityGML* model, these problems do not appear. An example of faces that exist on the recesses of a door of a building and have wrong orientation after the *MVSM* projection, is illustrated in Figure 4.29.



Figure 4.29: Example of faces that have wrong orientation due to door recession.

After the projection of the faces of the *MVSM* model to the plane surfaces of the corresponding *CityGML* model, the faces that correspond to degenerated faces are deleted (*Heuristic rule 1 - Step 4*). Moreover, the faces that have wrong orientation based on the related *CityGML* surface, are flipped (*Heuristic rule 2 - Step 4*).

In addition, on the boundary line of the faces of the *CityGML* model, the projected faces of the *MVSM* model are not parallel, as it is shown in Figure 4.30.

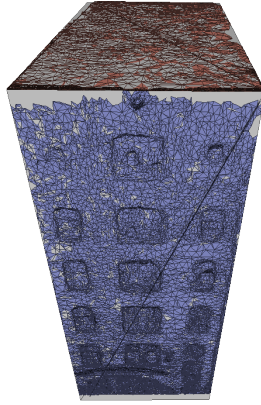


Figure 4.30: Gaps between the projected faces of the MVSM model (red triangles for roof class and blue triangles for wall class) and the plane surface of the CityGML model (grey surfaces).

In order to solve the gaps on the boundary line of the CityGML model between the projected faces of the MVSM model and the related plane surfaces of the CityGML model, the faces of the MVSM model that are closer to the boundaries of the CityGML model have to be stretched. The stretching of these faces (blue and red triangles in Figure 4.30) is done by projecting each one of the two vertices of the edges that do not touch other faces of the mesh, to the closest point found on the boundary line of the CityGML model (grey surface in Figure 4.30) that is closer to these vertices. The computations during this process are described in Section 3.5. An example of the steps of the projection of the wall class of the 1<sup>st</sup> building of the testing dataset of this implementation, is shown in Figure 4.32.

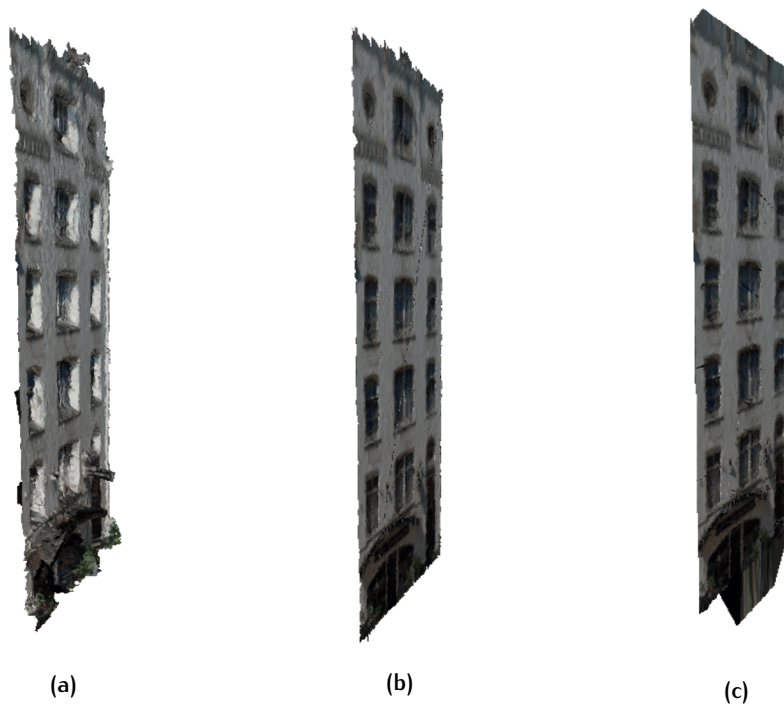


Figure 4.31: Steps of the projection of the wall class of the 1<sup>st</sup> building of the testing dataset of the 1<sup>st</sup> implementation. (a) Wall class of the 1<sup>st</sup> building of the testing dataset of the 1<sup>st</sup> implementation. (b) Projection of the wall class of the MVSM model to the plane surface of the CityGML model. (c) Projection of the boundary edges of the projected faces of the MVSM model to the closest edges of the CityGML model.

During the projection of the boundary edges of the MVSM model to the closest edges of the CityGML model, some issues occur at the corners of the boundary lines and on the texturing of the mesh. During

the stretching of the projected faces to the boundary lines of the CityGML surfaces, the corners of the boundary lines are not covered and further projection techniques need to be investigated in order to complete the texturing of the surfaces of the CityGML model that correspond to the existing surfaces of the MVSM model. In addition, as it is already mentioned in the quality of the visioned mesh in the previous section 4.2.2, the mesh of the MVSM model contain many holes that cause problems during the projection phase. While searching the faces of the MVSM model that are part of the boundary of the mesh (faces with edges that do not touch other faces of the MVSM mesh), the faces of the MVSM model that correspond to the holes of the mesh, are also projected to the boundary edges of the closest CityGML edge. An example of this problem is illustrated in Figure 4.32.

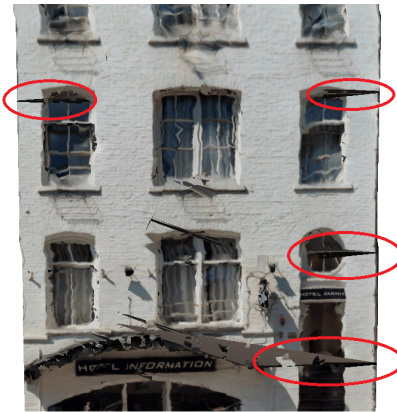


Figure 4.32: Projected faces of the MVSM model to the boundary surfaces of the CityGML model (red circles - errors caused by holes on the mesh).

The holes of the mesh of the MVSM model need to be closed, before proceeding onto the projection of the faces to the boundary line of the CityGML model, in order to enhance the texturing step of the CityGML model, with information from an existing corresponding MVSM model.

#### 4.3.2 2<sup>nd</sup> Implementation of the proposed methodology

The 2<sup>nd</sup> implementation of the proposed methodology is related to the implementation of the methodology presented in Chapter 3. In this implementation the testing dataset that is used, is the building block that is provided in the MVSM model (see Figure 4.12) and in the LOD1 and LOD2 CityGML models (see Figure 4.7 and 4.8 respectively). The steps that are implemented in Section 4.3.1, are repeated in this part. Details on the process of the methodology are mentioned per step, in the previous implementation (Section 4.3.1). Therefore, only comments on the comparison between the results that the two different LODs provide in each step, are discussed in this part. The semantic segmentation is resulting into five classes, the roof, the wall, the road, the terrain and the uncertain, with texture colors red, blue, grey, brown and yellow, respectively. In Figure 4.33 a legend of the colors that will appear during the 2<sup>nd</sup> implementation, is depicted.

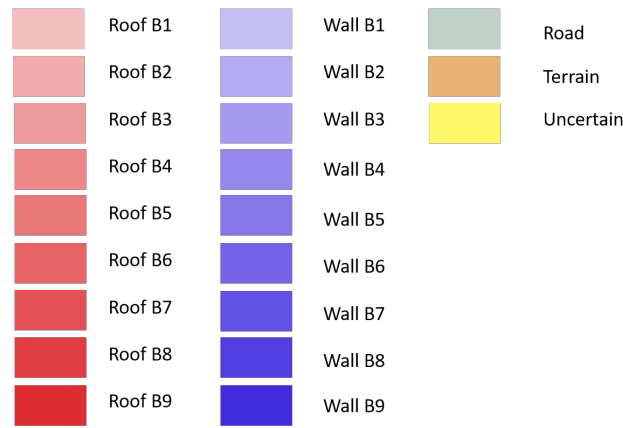


Figure 4.33: Legend of the classes that derive from the semantic segmentation of the 2<sup>nd</sup> implementation.

### Step 1

At this step of the implementation the correspondence of the faces of the buildings of the CityGML model with the faces of the MVSM model is tested. The results of both LODs with the corresponding MVSM model are shown in the following figure:

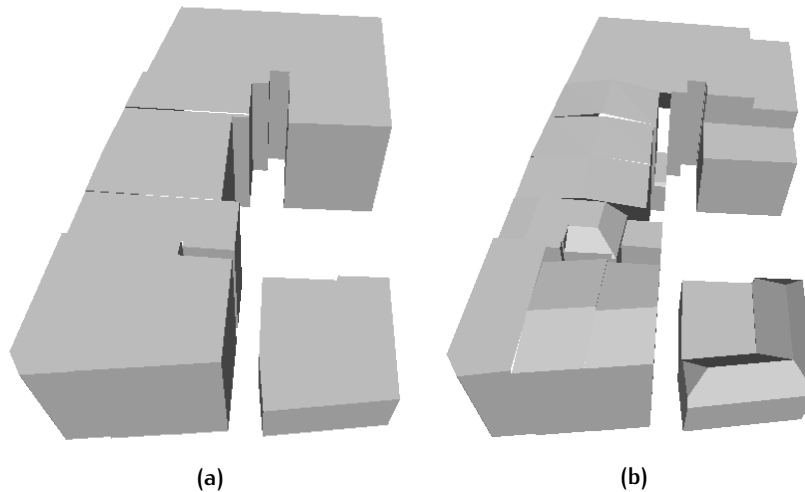


Figure 4.34: Faces of the buildings of the CityGML model that correspond to the faces of the MVSM model (2<sup>nd</sup> Implementation). (a) Faces of the buildings of LOD<sub>1</sub> CityGML model that correspond to the faces of the MVSM model. (b) Faces of the buildings of LOD<sub>2</sub> CityGML model that correspond to the faces of the MVSM model.

Both the MVSM model with the LOD<sub>1</sub> and the MVSM model with the LOD<sub>2</sub>, kept all the outer walls of the buildings of the CityGML model, while at the same time the ground surfaces and the shared walls of the buildings of the CityGML model are removed. It can be observed that this step of the proposed methodology is succeeded in both LODs. The remained faces of the buildings, together with the road and the terrain surfaces of the CityGML models (see Figures 4.7 and 4.8) can be matched with the faces of the MVSM model and are proceeding into the next step of the proposed methodology.

### Step 2

The 2<sup>nd</sup> step of the 2<sup>nd</sup> implementation of the proposed methodology, is related to the segmentation of the building block of the MVSM model into five different classes (roof, wall, road, terrain and uncertain) with the use of the road and the terrain surfaces of the CityGML models that are depicted in Figures 4.7 and 4.8, and the corresponding faces of the buildings of the CityGML model that are shown in Figure 4.34.



The automatic matching of the faces of the two 3D city models is conducted based on the combination of distance computations, normal vector comparisons and heuristic rules, according to the step 2 of the proposed methodology, described in Section 3.3.

The result of *Heuristic rule 1 - Step 2* and *Heuristic rule 2 - Step 2* of the LOD1 and LOD2 CityGML model with the MVSM model of the building block, are presented respectively in Figures 4.35 and 4.36.

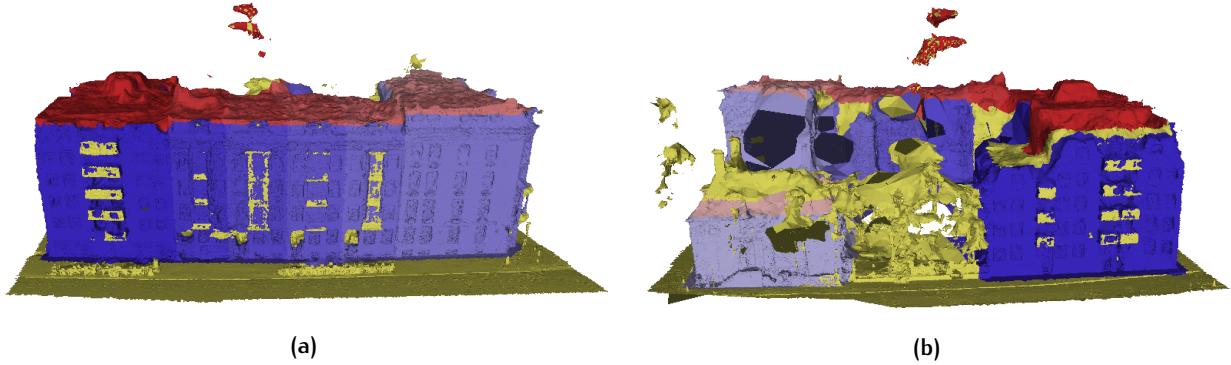


Figure 4.35: Result of the step 2 of the implementation of the proposed methodology on the building block in LOD1 CityGML dataset, after the *Heuristic rule 1 - Step 2* and *Heuristic rule 2 - Step 2* (Section 3.3) (2<sup>nd</sup> Implementation). (a) West side of the building block. (b) East side of the building block.

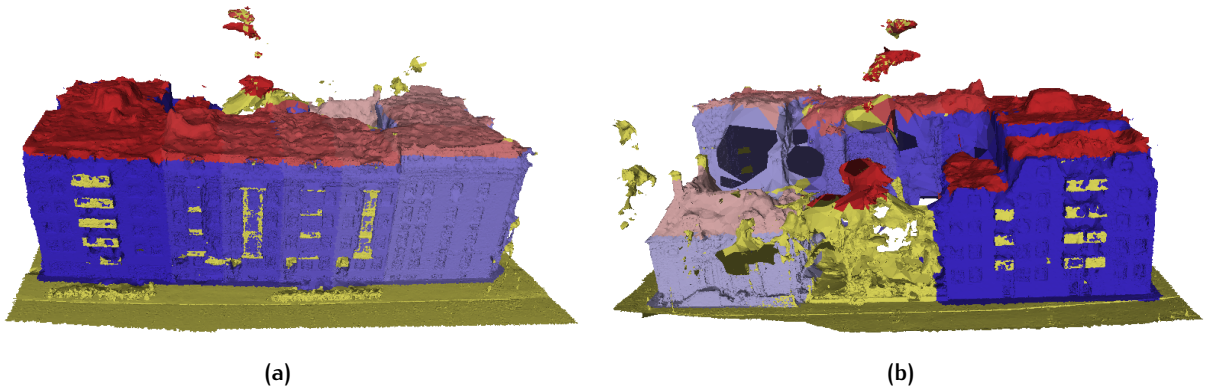


Figure 4.36: Result of the step 2 of the implementation of the proposed methodology on the building block in LOD2 CityGML dataset, after the *Heuristic rule 1 - Step 2* and *Heuristic rule 2 - Step 2* (Section 3.3) (2<sup>nd</sup> Implementation). (a) West side of the building block. (b) East side of the building block.

The results of the first two heuristic rules on the two different LODs, provide some differences on the roof class of the buildings that have complex roof structures. According to the numbering of the buildings of the building block that is decided in Section 4.2.1, it has to be noticed that the building  $B_1$  and the building  $B_9$  have more complicated roof structures, than the other seven buildings of the building block. By comparing the Figures 4.35b and 4.36b, it can be easily observed that a big part of the roofs of these buildings that are included in the LOD1 CityGML model, are assigned as uncertain (yellow) after the *Heuristic rule 1 - Step 2* and *Heuristic rule 2 - Step 2*. On the other hand, the same roofs of the same buildings in the LOD2 CityGML model, are mostly assigned as roof class (red). These differences on the results were expected, because the representation of the roofs in the LOD1, specifically for the mentioned buildings, do not represent the reality.

After the *Heuristic rule 1 - Step 2* and *Heuristic rule 2 - Step 2*, the *Heuristic rule 3 - Step 2* is implemented. The results on the building block, for the two different LODs are depicted in Figures 4.37 and 4.38.

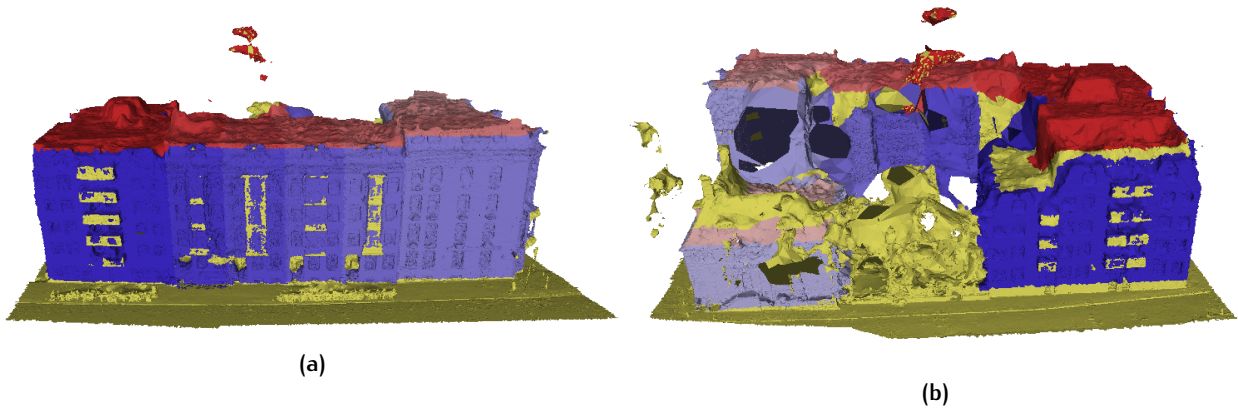


Figure 4.37: Result of the step 2 of the implementation of the proposed methodology on the building block in LOD1 CityGML dataset, after the *Heuristic rule 3 - Step 2* (Section 3.3) (2<sup>nd</sup> Implementation). (a) West side of the building block. (b) East side of the building block.

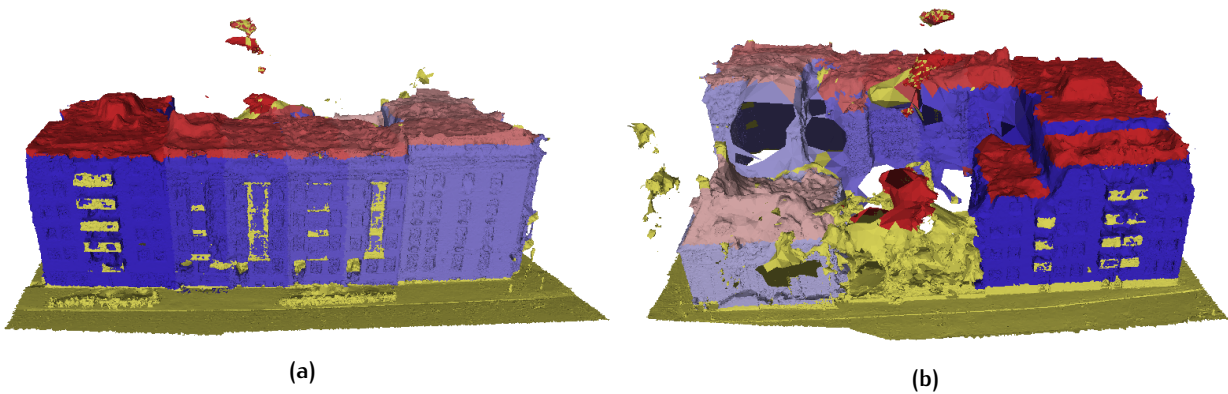
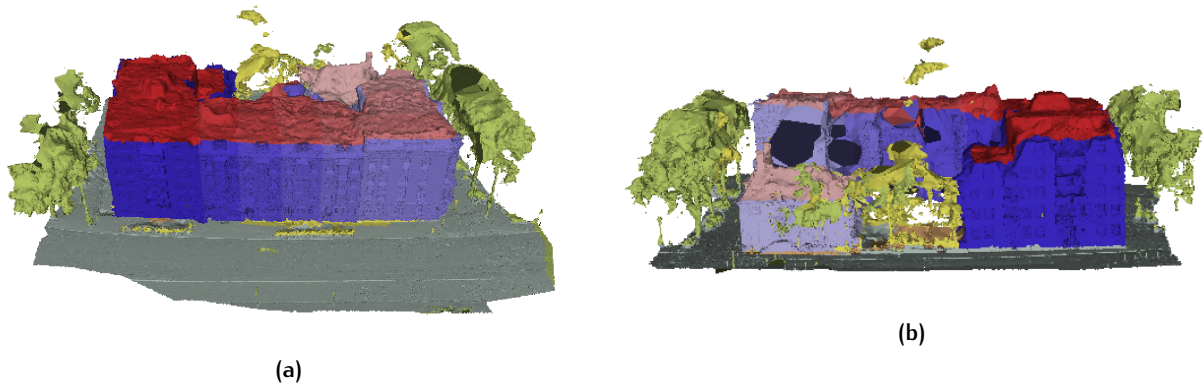
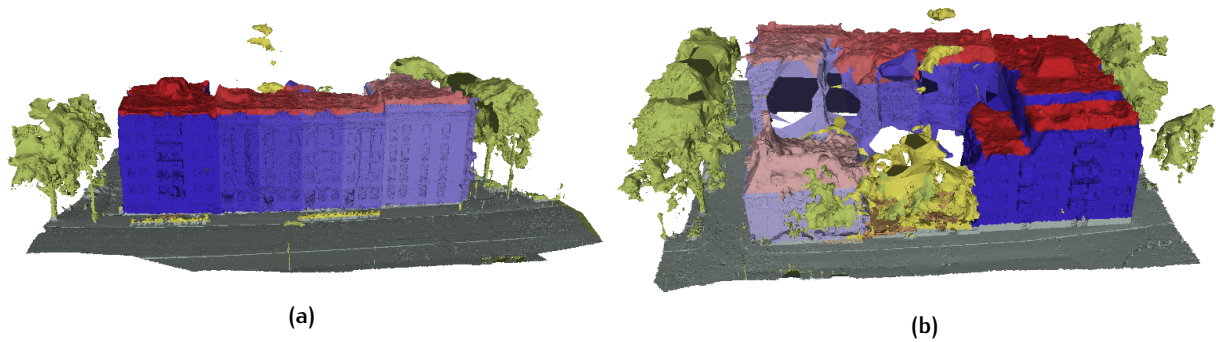


Figure 4.38: Result of the step 2 of the implementation of the proposed methodology on the building block in LOD2 CityGML dataset, after the *Heuristic rule 3 - Step 2* (Section 3.3) (2<sup>nd</sup> Implementation). (a) West side of the building block. (b) East side of the building block.

The results after the *Heuristic rule 3 - Step 2* show the segmentation of the MVSM model into the three classes (roof class (red), wall class (blue) and uncertain (yellow)). The different intensities of the red texture of the roof class and the blue texture of the wall class are indicating the different unique identifiers of the buildings (see Table 4.3 in Section 4.2.1), with which the related faces of the MVSM model correspond to. The connected components are detected, in order to fix the unconnected components that are part of wrong classes. After the correction of the connected components the uncertain (yellow) is proceeding into the *Heuristic rule 4 - Step 2*, *Heuristic rule 5 - Step 2* and *Heuristic rule 6 - Step 2* that are analysed in Section 3.3 and it is divided into road (grey), terrain (brown) and uncertain (yellow). The results of the above computations are depicted for the testing dataset that uses the LOD1 CityGML model in Figure 4.39 and for the one that uses the LOD2 CityGML model in Figure 4.40. The unconnected faces that are detected are fixed in the following figures:



**Figure 4.39:** Result of the step 2 of the implementation of the proposed methodology on the building block in LOD1 CityGML dataset, after the correction of the connected components (Section 3.3) (2<sup>nd</sup> Implementation). (a) West side of the building block. (b) East side of the building block.



**Figure 4.40:** Result of the step 2 of the implementation of the proposed methodology on the building block in LOD2 CityGML dataset, after the correction of the connected components (Section 3.3) (2<sup>nd</sup> Implementation). (a) West side of the building block. (b) East side of the building block.

The results of this step of the implementation with the use of LOD1 CityGML model and LOD2 CityGML model, together with the corresponding MVSM model seem very similar. The only major difference that needs to be highlighted is in the building  $B_9$ , where the roof structure of the two different LODs differs a lot. It can be seen that the result from the LOD2 provides a better representation of the reality, because the roof structure of the building  $B_9$  is assigned at the wall (blue) or the roof (red) class (see Figure 4.40b), while on the result of the LOD1 depicted in Figure 4.39b, parts of the wall of the MVSM model are assigned as roof (red).

### Step 3

After the semantic segmentation of the MVSM model that correspond to the building block, the quality control of the matching of the roof and wall classes of the different buildings is presented in Table 4.7.



Building name	LOD1 Roof class	Quality	LOD2 Roof class	Quality	LOD1 Wall class	Quality	LOD2 Wall	Quality
B1	8.526m	Medium	6.524m	Medium	8.399m	Medium	7.830m	Good
B2	6.170m	Good	4.872m	Excellent	10.386m	Medium	10.320m	Medium
B3	7.583m	Good	4.200m	Excellent	10.342m	Medium	10.108m	Medium
B4	7.667m	Good	7.666m	Good	10.590m	Medium	10.407m	Medium
B5	6.200m	Good	3.541m	Excellent	9.590m	Medium	9.345m	Medium
B6	8.064m	Medium	4.923m	Excellent	9.767m	Medium	9.430m	Medium
B7	7.555m	Good	4.600m	Excellent	9.845m	Medium	9.603m	Medium
B8	7.895m	Good	4.878m	Excellent	9.865m	Medium	9.395m	Medium
B9	11.974m	Bad	9.622m	Medium	10.106m	Medium	10.500m	Medium

**Table 4.7:** Categorisation of the quality control of the semantic segmentation, based on the Hausdorff distance computations (2<sup>nd</sup> implementation).

From the previous table of the quality control of the matching of the faces, it is very interesting to notice the big values of the building B<sub>1</sub> and especially of the building B<sub>9</sub>. Both buildings are buildings that had more complex roof structures. Therefore, the low quality of the matching of the roof of the building B<sub>9</sub>, especially in the LOD1 was expected. The roof and wall classes are tested for the detection of errors, concerning the shapes of the roofs of the CityGML model and the absence of buildings in the CityGML model. The results of the validations are shown in Table 4.8.

Building name	LOD1 Matching with the shape of the roof	LOD2 Matching with the shape of the roof	LOD1 Absence of building	LOD2 Absence of building
B1	Medium	Medium	No	No
B2	Good	Excellent	No	No
B3	Good	Excellent	No	No
B4	Good	Good	No	No
B5	Good	Excellent	No	No
B6	Medium	Excellent	No	No
B7	Good	Excellent	No	No
B8	Good	Excellent	No	No
B9	Bad	Medium	No	No

**Table 4.8:** Indications of errors that are related to the shape of the roofs and the absence of buildings in the CityGML model (2<sup>nd</sup> implementation).

The above results indicate that the roof of the building B<sub>9</sub> in LOD1 CityGML model needs to be checked. The Figure 4.41 illustrates the MVSM model and the LOD1 CityGML model (transparent box in Figure 4.41) of the building B<sub>9</sub>.



Figure 4.41: Roof of the building  $B_9$  of the MVSM model and the LOD1 CityGML model (transparent box).

It is obvious that the roof structure of the building  $B_9$  is more complicated than the flat roof of the LOD1 CityGML model. Therefore, the indication that is resulted from the Table 4.8 is correct.

#### Step 4

The transfer of the texture from the MVSM model to the CityGML model, is conducted in this step. The results of the projection of the faces of the MVSM model to the surfaces of the CityGML model are depicted in the Figures 4.42 and 4.43 on the surface of the LOD1 and LOD2 CityGML model, respectively.

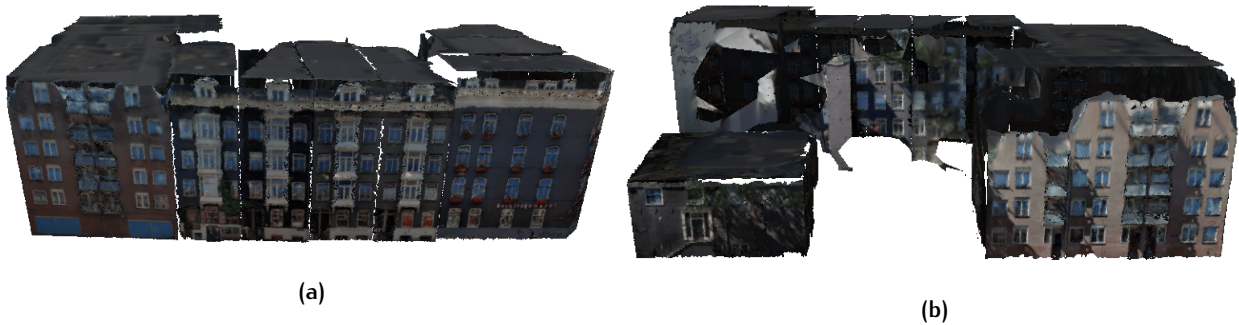


Figure 4.42: Result of the step 4 of the 2<sup>nd</sup> implementation on the LOD1 CityGML model. (a) West side of the building block. (b) East side of the building block.



Figure 4.43: Result of the step 4 of the 2<sup>nd</sup> implementation on the LOD2 CityGML model. (a) West side of the building block. (b) East side of the building block.

The results that are illustrated on the previous figures, depict the transfer of texture from the MVSM model to the CityGML models. It has to be highlighted that the transfer of the texture from the MVSM model is

transferred to the different structures, according to the different **LOD** that is related to. These differences can be seen specifically in the roof structures of the building  $B_1$  and the building  $B_9$  in the Figures [4.42b](#) and [4.43b](#). Further improvements need to be performed, such as the projection of the textured faces to the boundary lines of the surfaces of the **CityGML** model.



# 5

## CONCLUSION AND RECOMMENDATIONS

The aim of this thesis is to enrich automatically 3D city models with the use of additional data that can be found into different existing formats of 3D city models of the same area, representing the same features. More specifically, it is achieving to provide enrichments of CityGML models with information from MVSM models and enrichments of MVSM models with information from CityGML models. In this part, the research questions that are presented in Section 1.2.1, are answered. Then, several discussions on the methodologies that are used in this thesis are analysed. Conclusions and recommendations for further research on this topic are presented.

### 5.1 RESEARCH QUESTIONS

The research questions that are mentioned in Section 1.2.1 are analysed in this part in order to answer the main research question of this thesis. The first three sub-questions are related to a theoretical part of the research of the bidirectional enrichment. Next, the presented sub-questions are linked with the focus on a practical part, in order to first explore practically the enrichments of a MVSM model with information from the use of a corresponding CityGML model and then, the enrichments of a CityGML model with the use of a corresponding MVSM model.

*What is the concept of bidirectional enrichment?*

This thesis has been initiated by the need of introducing the concept of bidirectional enrichment. Before proceeding to the answer of this sub-question, it is important to add that there are many different 3D city model formats that have different characteristics, such as geometric, semantic or topological aspects, based on their creation.

The bidirectional enrichment is the concept of enriching bidirectionally two existing 3D city models with information from one to the other, according to the missing data that each 3D city model needs. More specifically, the bidirectional enrichment can be characterized as the either transfer or update of aspects that might or might not exist from one existing 3D city model to another existing 3D city model of the same entities, and conversely. The characteristics of the 3D city models can vary to their modelling aspects, their storage, their file formats etc. and these differentiations may affect the aspects and the methods that can be used on different bidirectional enrichments.

*What kind of enrichments are theoretically possible between a MVSM model and a CityGML model?*

One of the main contributions of this thesis is related to the answer of this sub-question (see Table 2.1). The enrichments that can be provided to a MVSM model with the use of information from a CityGML model are the *semantic segmentation*, the *transfer of attributes*, the *straightening* and the *simplification of the mesh*. On the other direction, with the use of information from existing MVSM models to existing CityGML models, the categorisation of the enrichments are specified to the *texturing*, *validation* and *update of information*, plus *LOD creation* of CityGML model. Each one of these enrichments are thoroughly analysed in Section 2.3.1.

*What enrichment methodologies can be performed for the automatic bidirectional enrichments of a MVSM model and a CityGML model of the same features?*

After some research on the topic of the automatic and bidirectional enrichment of CityGML and MVSM models, it is concluded that at the subject of the enrichment of a CityGML model with information from a MVSM model of the same entities and conversely, no related work has been found. Several fields that are partly related to methodologies that can be used for the bidirectional enrichments presented in Table 2.1,

are the interoperability of information, the semantic enrichment of 3D city models, the straightening of meshes, the matching of meshes and the mesh comparison.

The techniques that have been used on these fields have evolved the last decade and can be used for the creation, the enhancement or the enrichments of created 3D city models. Therefore, it is concluded that the basic methodologies that can be used for the automatic and bidirectional enrichment of *CityGML* and *MVSM* models, are the schemas connection, creation and comparison of planes, probabilistic methods, predefined shapes of features, template matching techniques, machine learning techniques, clustering techniques, heuristic rules and distance computation methods. They can be used individually or can be combined, in order to achieve an automatic, semi-automatic or manual enrichment (Section 2.3.1) of a 3D city model. It has to be noticed that the choice of a technique is related to the different characteristics that each 3D city model has (Section 1.1) and the characteristics of the foreseen application (Section 2.3.1).

*To what extent can a MVSM model be segmented semantically by matching its faces with the faces of the building elements (roof, wall) and the city object classes of relief features (terrain) and transportation bodies (roads) of a CityGML model?*

A *MVSM* model is segmented semantically by matching its faces with the corresponding faces of the building elements (roof, wall) and the city object classes of relief features (terrain) and transportation bodies (roads) of a *CityGML* model. The automatic matching of the faces of the two 3D city models is conducted in this thesis based on the combination of distance computations, normal vector comparisons and heuristic rules.

Two implementations are provided in order to test this methodology (Section 4.3.1 and 4.3.2). The faces of the *MVSM* model are semantically segmented into five classes, roof, wall, terrain, road and uncertain. After the performance of this methodology on the testing datasets presented in Section 4.2, it is found that this methodology is successful after the use of either a *LOD1 CityGML* model or a *LOD2 CityGML* model.

A problem of unconnected faces occurs during this process. It usually appears in some of the five classes that happen to end up with holes, while other classes end up with some unconnected faces. This is solved with the check of the connectivity of the faces of the *MVSM* model and the identification of the different connected components. Moreover, after the implementation of the semantic segmentation, it can be concluded that some enhancements can be performed. Most of the times the roof class contains parts of the walls of the buildings. The reason for that is that the roof class is created based on the roof surfaces of the *CityGML* model. These surfaces do not correspond to the complexity of a building in reality, therefore the segmentation of the *MVSM* model cannot be correctly realized. Moreover, it can be observed that after the implementation of the proposed methodology, the wall class includes all the characteristics of the facades of the buildings that are close enough to the related wall surface of the *CityGML* model. These characteristics are not always related to the building facades. They might also contain vegetation or other obstacles that should not be part of this class. It can be concluded that further investigation needs to be done, in order to be able to improve the semantic segmentation and be able to identify the differences on the characteristics of the objects, like the identification of the characteristics of the facades of the buildings that should or should not be assigned as wall class.

*How to transfer and assign attributes from a CityGML model to a MVSM model?*

The transfer of the attributes of the *CityGML* model to the faces of the *MVSM* model is done as a result of the semantic segmentation. During the semantic segmentation of the mesh of the *MVSM* model, the faces of the mesh are segmented based on a combination of distance computations, normal vector comparisons and heuristic rules, according to the step 2 of the proposed methodology, described in Section 3.3. After the segmentation, each face of the *MVSM* model is assigned as one of the five classes, roof, wall, road, terrain, uncertain. The faces that are assigned as roof and wall class are related to a specific *CityGML* surface. The unique identifier of the surface of the *CityGML* model is provided to the related face of the *MVSM* model and in this way, the attributes that correspond to the specific buildings in the *CityGML* model are transferred to the related buildings in the *MVSM* model.

*How to segment a MVSM model into different buildings?*

The segmentation of the *MVSM* model into different buildings is conducted as a result of the semantic segmentation. As it is already mentioned in the previous sub-question, the *CityGML* model has a different unique identifier per building. The faces of the *MVSM* model that correspond to the same unique identifier after the semantic segmentation are part of the same building in the 3D city model.

*Is it possible to perform other operations like validation and to detect errors in either models, based on bidirectional matches?*

A bidirectional match between the *CityGML* and the *MVSM* model is conducted with the use of the Hausdorff distance computation. This distance is the computation of the maximum distance of all the distances between all the points of one model to all the points of the other model. After the semantic segmentation of the *MVSM* model, the transfer of attributes to the faces of the segmented mesh of the *MVSM* model and the segmentation of that model into the different buildings, further operations can be performed.

Firstly, in this thesis, the quality control of the matching of the faces of the two 3D city models is conducted. In this part, the two layers of the roof and the wall of each building, derived from the semantic segmentation of the *MVSM* model, are tested with the related surfaces, roof and wall, of the corresponding building from the *CityGML* model.

Next, a quality check of the building information in the *CityGML* model is performed. In this way, the validation of the shapes of the roofs of the buildings take place, together with a validation considering the existence of buildings in the *CityGML* model. This is possible considering that the *MVSM* model is younger than the *CityGML* model. In this case, indications on the matching of the faces between the two 3D city models provide whether or not the information of the *CityGML* model is valid.

*How to automatically texture the buildings of a CityGML model using the texture from a MVSM model?*

The faces of the *MVSM* model that correspond to the roof and the wall class, after the semantic segmentation, can be related to the corresponding surfaces of the roof and the wall surfaces of the *CityGML* model. Every face of the *MVSM* model is projected to the corresponding plane surface of the roof or the related wall of the *CityGML* model and the texture of the *MVSM* model is transferred to the *CityGML* model.

After the projection of the faces of the *MVSM* model, some degenerated faces and some faces with opposite orientation are expected. Such faces are mostly created on areas, where the mesh is composed of many polygons, for example on areas that are related to window and door recesses and protrusions, vegetation or other obstacles. However, in areas where the *MVSM* model is in a way parallel to the related surface of the *CityGML* model, these problems do not appear.

Moreover, after the projection of the faces of the *MVSM* model, some gaps occur around the boundary line of the *CityGML* model, between the projected faces of the *MVSM* model and the related plane surfaces of the *CityGML* model. In order to solve these gaps, the faces of the *MVSM* model that are closer to the boundaries of the *CityGML* model have to be stretched. During the projection of the boundary edges of the *MVSM* model to the closest edges of the *CityGML* model, some issues occur at the corners of the boundary lines and on the texturing of the mesh. During the stretching of the projected faces to the boundary lines of the *CityGML* surfaces, the corners of the boundary lines are not covered and further projection techniques need to be investigated, in order to complete the texturing of the surfaces of the *CityGML* model that correspond to the existing surfaces of the *MVSM* model. In addition, in case of errors on the textured mesh of the *MVSM* model, such as holes, further investigation needs to be conducted for the transfer of texture to a *CityGML* model.

Based on the above answers on the sub-questions of this research, it can be concluded that the main research question of this thesis is answered. The main research question is the following:

*To what extent can a CityGML model and a MVSM model of the same area, representing the same features, be automatically and bidirectionally enriched?*



After the definition of the concept of the bidirectional enrichment between two 3D city models, the possible enrichments between a *MVSM* model and a *CityGML* model are analysed. The related work on this field and on similar fields that can provide some techniques that can be used during enrichment methodologies between a *MVSM* model and a *CityGML* model, has shown that this venture is possible. In order to proceed on the exploration of the automation of the bidirectional enrichment between the two 3D city models, a proposed methodology is created. There, the semantic segmentation of the *MVSM* model is explored by matching the faces of the *MVSM* model to the faces of the *CityGML* model. The mesh of the *MVSM* model is segmented semantically into the classes, roof, wall, road, terrain and uncertain. The attributes from the correspondent *CityGML* surfaces are transferred to the related faces of the *MVSM* model and according to them, the *CityGML* model is segmented per building. The correspondence of the faces between the two 3D city models allowed the performance of other operations like validation of the matching of the semantic segmentation, plus the detection of errors in either models, such as the shapes of the roofs and the absence of buildings in the *CityGML* model. Finally, the texture of the faces of the *MVSM* model is transferred to the surfaces of the *CityGML* model. In general, the *MVSM* model is segmented semantically and attributes are transferred to the faces of the mesh, while on the other direction, information related to the shapes of the roofs and the absence of buildings is validated on the *CityGML* model and texture from the *MVSM* model is transferred to the surfaces of the *CityGML* model. Therefore, it can be concluded that a *CityGML* model and a *MVSM* model of the same area, representing the same features, can be automatically and bidirectionally enriched.

## 5.2 DISCUSSION

Discussions concerning the contribution on the field of the bidirectional enrichment of *CityGML* and *MVSM* models and the methodologies that are used during this thesis, are presented in this part.

At the beginning of this thesis, the concept of bidirectional enrichment and the enrichments between two 3D city models were not previously defined. Therefore, the concept of the bidirectional enrichment between different 3D city models is introduced. More specifically, the characteristics of the *CityGML* and the *MVSM* models are explored. This was important for the further investigation and exploration of the enrichments that can be conducted between these two 3D city model formats. The enrichments that can be provided to a *MVSM* model with the use of information from a *CityGML* model are categorised in this thesis to the *semantic segmentation*, the *transfer of attributes*, the *straightening* and the *simplification of the mesh*. On the other direction, with the use of information from existing *MVSM* models to existing *CityGML* models, the categorisation of the enrichments are specified to the *texturing*, *validation* and *update of information*, plus *LOD creation* of *CityGML* model. These enrichments are thoroughly analysed in Section 2.3.1 and are presented in Table 2.1. The proposed categorisation can be expanded and each enrichment can be tested, in order to establish and proof in practice the theoretical enrichments that are listed.

After the categorisation of the possible theoretical enrichments that can be conducted between *CityGML* and *MVSM* models, some of them are chosen to be tested practically. More specifically, the *MVSM* model is segmented semantically and attributes are transferred to the faces of the mesh, while on the other direction, information related to the shapes of the roofs and the absence of buildings is validated on the *CityGML* model and texture from the *MVSM* model is transferred to the surfaces of the *CityGML* model. It has to be noticed that the subject of the enrichment of a *CityGML* model with information from a *MVSM* model of the same entities and conversely, is new. Therefore, a methodology is proposed for the achievement of the automatic bidirectional enrichment of the two 3D city models. This methodology is based on the combination of distance computations, normal vector comparisons and heuristic rules. Improvements on the steps of the methodology and the methods that are used, are necessary for future implementations. It is believed that a complete automated technique can be created for the bidirectional enrichment of *CityGML* and *MVSM* models.

The main distance computations that are performed during the proposed methodology are the point-to-polygonal mesh distance for the achievement of the semantic segmentation of the *MVSM* model with the use of the different layers of the *CityGML* model and for the projection of the texture of the *MVSM* model to the surfaces of the *CityGML* model. According to Ericson (2004) and Eberly (2006), point-to-polygonal mesh distance is a way of obtaining the closest point between a triangle and a point with the use of a

vector calculus approach. The minimum of this function occurs in one of three cases, at a vertex, on an edge or in the interior of the triangle. This approach is used, in order to compare the meshes of the two different existing 3D data models and segment the meshes of the *MVSM* model according to the surfaces of the *CityGML* model. It is a computation that provides easily the closest point between a point to an edge or to a triangle, in 3D space.

Another distance that is used, is the Hausdorff distance, for the quality control of the conducted matching of the surfaces of the *MVSM* model and for the validation of the existence of buildings and the validation of the shapes of the roofs on the corresponding *CityGML* model. It is the maximum of all the minimum distances from a point in one model, to all the points in the other model, so it provides a global and not a local comparison between two meshes (Veltkamp, 2001). It is not always recommended, due to the fact that it is a global comparison and outliers may affect a lot the result.

## 5.3 CONCLUSION

The use of 3D city models has increased the last decades due to the evolution of technology. Their use is related with the need to provide solutions on issues that correspond to the building environment. Modelling aspects, like geometric, semantic information and topology are necessary, in order to provide an integrated result of spatial analysis. Therefore, this thesis is recording the possible enrichments that can be done bidirectionally between *MVSM* and *CityGML* models, when both exist for the same entities. In this way, both 3D city models can be enriched with the missing information from the other 3D city model. More specifically, two enrichments are achieved per direction with the use of a proposed methodology that is based on distance computations between the meshes of the two 3D city models.

At the beginning, the concept of bidirectional enrichment between two 3D city models is identified and the possible enrichments that can be conducted between a *MVSM* model and a *CityGML* model are listed for the *LOD1*, *LOD2* and *LOD3* in Section 2.3. Next, enrichment methodologies that have been used from similar applications and methodologies that can be used in the future for the achievement of bidirectional enrichments, are discussed. The automatic enrichment of the two models with two enrichments per direction is conducted with the use of a proposed methodology that is created for the needs of this thesis. During that process, the matching of the two 3D city models is realized by matching the faces of the models in 3D space, using distance computations. In this way, the semantic segmentation of the mesh of the *MVSM* model is done with the use of the corresponding faces of the *CityGML* model. The mesh is divided into five classes (roof, wall, road, terrain and uncertain). The overall semantic segmentation that is conducted, is resulted with good matching between the faces of the same classes of the two 3D city models. The same methodology is tested for a small area of four buildings and for a block of buildings. The methodology is also performed for different *LODs* (*LOD1* and *LOD2*) of the *CityGML* model. The results of the different implementations showed that the matching between the meshes of the two 3D city models can achieve the semantic segmentation of the *MVSM* model with success. Then, the faces of the *MVSM* model that are matched to the related surfaces of the *CityGML* model, can be segmented into the different buildings that exist in the *CityGML* model. In this way, the transfer of the attributes from the surfaces of the *CityGML* model to the related faces of the *MVSM* model is achieved. With respect to this correspondence between the two meshes, two operations are performed in order to detect errors on the information of the buildings of the *CityGML* model. With the condition of having a younger *MVSM* model, the *CityGML* buildings can be validated for the existence of buildings and for the shapes of the roofs in the models. The final implementation that is achieved in this thesis, is related to the transfer of texture from the *MVSM* model to the surfaces of the *CityGML* model. The previous matching of the faces between the two models is used for the correct projection and transfer of texture information from the *MVSM* model to the *CityGML* model.

The contributions of this thesis are mainly the concept, the investigation and the implementation of bidirectional enrichments between *MVSM* and *CityGML* models. The main goal of this contribution is related to the interoperability of information between 3D city models. As it is already mentioned, the bidirectional enrichment is new. Therefore, further investigation has to be done on all the enrichments that are listed in Section 2.3. The proposed methodology that is presented in this thesis together with the theoretical part of the bidirectional enrichment should be extended from other researchers, with tests, improvements

and extensions on the proposed methodology, in order to reach a new and automated strategy on the exchange of useful information from one 3D city model to the other one.

## 5.4 FUTURE WORK AND RECOMMENDATIONS

Some future work and recommendations that derive from a new set of research questions, are presented in this part.

- A redefinition of the concept of the bidirectional enrichment.
- An extension of the Table 2.1 into further enrichments between a MVSM model and a CityGML model, plus the investigation of extending this table into more 3D city formats.
- The exploration of methodologies that can be used for the achievement of automatic bidirectional enrichments. Moreover, the extension of these methodologies in implementation with different LODs of the CityGML model.
- Test all the enrichments that are proposed theoretically in Table 2.1.
- Expand the proposed methodology (Chapter 3) to other thematic classes, such as vegetation, water bodies etc.
- Find ways on improving the proving the proposed methodology (Chapter 3), specifically on the improvement of the step of the transferring of texture.
- Fix the non-manifold errors in the MVSM model.
- Test the proposed methodology (Chapter 3) with the use of a LOD3 CityGML model.
- Test the proposed methodology (Chapter 3) with the use of smoother and more simplifier MVSM models.
- Test the proposed methodology on more complex buildings and compare the results with the use of different testing datasets from different LODs.
- Reduce the execution time of the process of the proposed methodology.
- Test the categorization of the Hausdorff distances in bigger areas, in order to determine better intervals for the detection of errors based on the bidirectional matching of the meshes.

## BIBLIOGRAPHY

- E. Angel, D. Shreiner, and V. Shreiner. An interactive introduction to opengl programming. In *SIGGRAPH Courses*, pages 1–124, 2007.
- F. Biljecki, H. Ledoux, J. Stoter, and J. Zhao. Formalisation of the level of detail in 3D city modelling. *Computers, Environment and Urban Systems*, 48:1–15, 2014. doi: <http://dx.doi.org/10.1016/j.compenvurbsys.2014.05.004>.
- F. Biljecki, J. Stoter, H. Ledoux, S. Zlatanova, and A. Çöltekin. Applications of 3D city models: State of the art review. *ISPRS International Journal of Geo-Information*, 4(4):2842–2889, 2015. doi: <http://dx.doi.org/10.3390/ijgi4042842>.
- F. Biljecki, H. Ledoux, and J. Stoter. An improved lod specification for 3d building models. *Computers, Environment and Urban Systems*, 59:25–37, 2016.
- F. Cirak and Q. Long. Subdivision shells with exact boundary control and non-manifold geometry. *International Journal for Numerical Methods in Engineering*, 88(9):897–923, 2011.
- A. A. Diakit , G. Damiand, and G. Gesquiere. Automatic semantic labelling of 3d buildings based on geometric and topological information. In *Proc. of 9th International 3DGeoInfo Conference (3DGeoInfo)*. Karlsruhe Institute of Technology, 2014.
- D. H. Eberly. *3D game engine design: a practical approach to real-time computer graphics*. CRC Press, 2006.
- S. O. Elberink, J. Stoter, H. Ledoux, and T. Commandeur. Generation and dissemination of a national virtual 3d city and landscape model for the netherlands. *Photogrammetric engineering & remote sensing*, 79(2):147–158, 2013.
- C. Ericson. *Real-time collision detection*. CRC Press, 2004.
- K. Fath, J. Stengel, W. Sprenger, H. R. Wilson, F. Schultmann, and T. E. Kuhn. A method for predicting the economic potential of (building-integrated) photovoltaics in urban areas based on hourly radiance simulations. *Solar Energy*, 116:357–370, 2015.
- S. Fuhrmann, F. Langguth, and M. Goesele. Mve-a multi-view reconstruction environment. In *GCH*, pages 11–18, 2014.
- Y. Furukawa, C. Hern andez, et al. *Multi-view stereo: A tutorial*. Citeseer, 2015.
- G. Gr oger and L. Pl umer. CityGML—interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 71:12–33, 2012.
- G. Gr oger, T. Kolbe, C. Nagel, and K. H afele. Ogc city geography markup language (citygml) encoding standard, version 2.0, ogc doc no. 12-019. *Open Geospatial Consortium*, 2012.
- A. Guezlec. “meshsweeper”: dynamic point-to-polygonal mesh distance and applications. *IEEE Transactions on visualization and computer graphics*, 7(1):47–61, 2001.
- O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz. Robust face detection using the hausdorff distance. In *International Conference on Audio-and Video-Based Biometric Person Authentication*, pages 90–95. Springer, 2001.
- M. Jonsson. Make it flat: Detection and correction of planar regions in triangle meshes, 2016.
- M. Kada and L. McKinley. 3d building reconstruction from lidar based on a cell decomposition approach. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(Part 3):W4, 2009.

- E. Kalogerakis, A. Hertzmann, and K. Singh. Learning 3d mesh segmentation and labeling. *ACM Transactions on Graphics (TOG)*, 29(4):102, 2010.
- K. Kavisha and S. Saran. Citygml based interoperability for the transformation of 3d data models. Master's thesis, Andhra University, India, 2015.
- T. H. Kolbe. Representing and exchanging 3d city models with citygml. In *3D geo-information sciences*, pages 15–31. Springer, 2009.
- T. H. Kolbe, C. Nagel, and A. Stadler. Citygml–ogc standard for photogrammetry. In *Photogrammetric week*, volume 9, pages 265–277, 2009.
- M. Lemmens. *Geo-information: technologies, applications and the environment*, volume 5. Springer Science & Business Media, 2011.
- S. Lu and F. Wang. Computer aided design system based on 3d gis for park design. *Computer, Intelligent Computing and Education Technology; CRC Press: London, UK*, pages 413–416, 2014.
- S. Malihi, M. Zoj, M. Hahn, M. Mokhtarzade, and H. Arefi. 3d building reconstruction using dense photogrammetric point cloud. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 71–74, 2016.
- H. Mayer and S. Reznik. Building facade interpretation from image sequences. In *Proceedings of the ISPRS Workshop CMRT*, pages 55–60, 2005.
- A. P. McClune, J. P. Mills, P. E. Miller, and D. A. Holland. Automatic 3d building reconstruction from a dense image matching dataset. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 641–648, 2016.
- J. M. Moreira, F. Nex, G. Agugiaro, F. Remondino, and N. J. Lim. From dsm to 3d building models: a quantitative evaluation. *Int Arch Photogramm Remote Sens Spatial Inf Sci XL1 W*, 1:213–219, 2013.
- T. Mullen. *Mastering blender*. John Wiley & Sons, 2011.
- K. L. Murdock. *3ds Max 2009 bible*, volume 560. John Wiley & Sons, 2008.
- W. Nguatem, M. Drauschke, and H. Mayer. Roof reconstruction from point clouds using importance sampling. *Ann. Photogrammetry, Remote Sens. Spat. Inf. Sci. II–3 W*, 3:73–78, 2013.
- W. Nguatem, M. Drauschke, and H. Mayer. Localization of windows and doors in 3d point clouds of facades. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2(3):87, 2014.
- W. Nguatem, M. Drauschke, and H. Mayer. Automatic generation of building models with levels of detail 1-3. In *XXIII ISPRS Congress, Technical Commission III*, volume 41, pages 649–654, 2016.
- M. Rook, F. Biljecki, and A. Diakit . Towards automatic semantic labelling of 3d city models. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, 2016.
- M. Roy, S. Fougou, and F. Truchetet. Generic attribute deviation metric for assessing mesh simplification algorithm quality. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 3, pages 817–820. IEEE, 2002.
- M. Roy, S. Fougou, and F. Truchetet. Mesh comparison using attribute deviation metric. *International Journal of Image and Graphics*, 4(01):127–140, 2004.
- J. Slade, C. B. Jones, and P. L. Rosin. Automatic semantic and geometric enrichment of citygml building models using hog-based template matching. In *Advances in 3D Geoinformation*, pages 357–372. Springer, 2017.
- M. Sonka, V. Hlavac, and R. Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014.

- Y. Sun. *3D building reconstruction from spaceborne TomoSAR point cloud*. PhD thesis, Technical University of Munich, 2016.
- R. Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
- M. Tarini, N. Pietroni, P. Cignoni, D. Panozzo, and E. Puppo. Practical quad mesh simplification. In *Computer Graphics Forum*, volume 29, pages 407–418. Wiley Online Library, 2010.
- H. Tashakkori, A. Rajabifard, and M. Kalantari. A new 3d indoor/outdoor spatial model for indoor emergency response facilitation. *Building and Environment*, 89:170–182, 2015.
- R. C. Veltkamp. Shape matching: similarity measures and algorithms. In *Shape Modeling and Applications, SMI 2001 International Conference on.*, pages 188–197. IEEE, 2001.
- Y. Verdie, F. Lafarge, and P. Alliez. Lod generation for urban scenes. Technical report, Association for Computing Machinery, 2015.
- J. Vince. *Geometry for computer graphics: formulae, examples and proofs*. Springer Science & Business Media, 2006.
- virtualcitySystems. *BuildingReconstruction 2015.2 Manual*, 2016.
- H.-H. Vu, P. Labatut, J.-P. Pons, and R. Keriven. High accuracy and visibility-consistent dense multiview stereo. *IEEE transactions on pattern analysis and machine intelligence*, 34(5):889–901, 2012.
- M. Waechter, N. Moehrle, and M. Goesele. Let there be color! large-scale texturing of 3d reconstructions. In *European Conference on Computer Vision*, pages 836–850. Springer, 2014.
- C. C. Wang, Y. Wang, K. Tang, and M. M. Yuen. Reduce the stretch in surface flattening by finding cutting paths to the surface boundary. *Computer-Aided Design*, 36(8):665–677, 2004.
- C. Wu. Towards linear-time incremental structure from motion. In *3DTV-Conference, 2013 International Conference on*, pages 127–134. IEEE, 2013.
- S. Zlatanova, J. Stoter, and U. Isikdag. Standards for exchange and storage of 3d information: Challenges and opportunities for emergency response. In *Proceedings of the 4th International Conference on Cartography & GIS, Volume 2, Albena, June 2012, pp. 17-28*. International Cartographic Association, 2012.

## COLOPHON

This document was typeset using L<sup>A</sup>T<sub>E</sub>X. The document layout was generated using the `arsclassica` package by Lorenzo Pantieri, which is an adaption of the original `classicthesis` package from André Miede.



