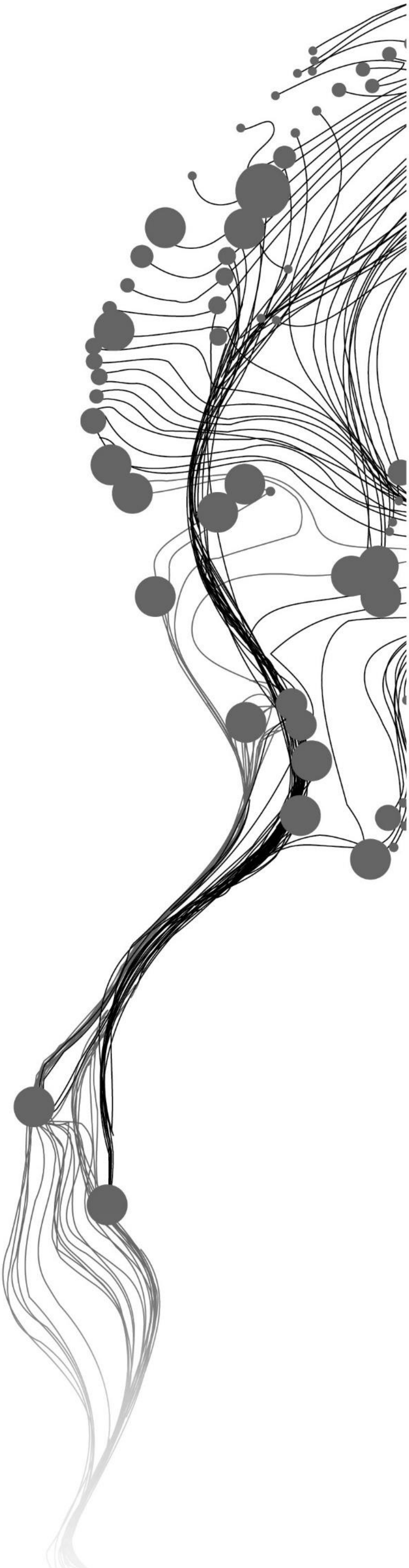
COMPREHENSIVE REMOTE SENSING BASED BUILDING DAMAGE CLASSIFICATION

YALDA SAEDI
March,2015

SUPERVISORS:
Dr. F.C. Nex
Dr.ing. M. Gerke
Dr. N. Kerle



COMPREHENSIVE REMOTE SENSING BASED BUILDING DAMAGE CLASSIFICATION

YALDA SAEDI
Enschede, The Netherlands, March, 2015

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.
Specialization: Geoinformatics

SUPERVISORS:
Dr. F.C. Nex
Dr.ing. M. Gerke
Dr. N. Kerle

THESIS ASSESSMENT BOARD:
Prof.dr.ir. M.G. Vosselman (Chair)
Dr. M.N. Koeva (External Examiner, University of Twente, ITC-PGM)

ABSTRACT

The importance of the building damage assessment after an earthquake is remarkable for rescue, reconstruction and estimating of geographic distribution. The accuracy and speed of this process play a significant role. Conventionally, buildings are evaluated by the specialist through a field survey. This approach is very time-consuming and strongly influenced by experience, skill and perspective of inspectors. State-of-the-art remote sensing technologies provide an opportunity to obtain damage information faster and cheaper. Although there have been several researches carried out to improve techniques for building damage detection, a reliable damage assessment requires semantic integration of remote sensing data based on structural engineering knowledge. This research proposed a framework to represent, classify and analyze detailed remote sensing data based on the semantic and spatial characteristics. Furthermore, the generated framework provides the detailed and overall building damage assessment automatically. The proposed research took a step towards improving the co-operation between remote sensing mappers and structural engineers.

A literature review is conducted to gain an understanding of the necessary information for building damage assessment. Different damage scales, catalogues and references are studied to obtain a holistic overview of the building evaluation process. Required and detectible damage evidence, attributes and relationships from remote sensing data according to the achieved knowledge are provided and classified in this research. The CityGML is an integrative information representation of visual, spatial and semantic properties of city objects hence in this research an extension to CityGML is proposed to store, analyze and represent geometrical, topological and semantic properties of damaged building entities. The 3D model of damaged building provides a visual interpretation of extracted damages for structural engineers in addition to the semantic and spatial properties of damaged building. The data is transformed into The PostgreSQL database which is cable to keep the data integrity and run spatial and non-spatial queries for reasoning. Also, an automatic damage assessment is provided in this study. Two fuzzy expert systems are designed to imitate the behavior of the specialists for building damage assessment. The generated systems capture the knowledge of structural engineers and involve the uncertainty of the decision making to evaluate the damaged structure. The first fuzzy expert system integrates semantic and spatial damage evidence and attributes related to the facade to calculate the damage grade of each façade separately. In the case of occlusion or incomplete 3D model of affected structure the detailed damage evaluation provides damage analysis of the visible parts. The second fuzzy system implements overall building damage assessment and computes the final building damage score.

The technical feasibility of the 3D damage modelling, database consistency and damage evaluation process of the proposed framework are analyzed. The results of damage evaluation of three simulated affected building indicate the proposed framework can integrate detailed damage evidence and calculate the accurate value as damage grade at façade and building scale. In addition, it is proved that the system successfully handles various levels of damage for computing reliable final damage value of the building. As a result, this thesis is a proof of concept for building damage assessment based on semantic and spatial remote sensing data.

Keywords: Remote sensing, Building damage assessment, CityGML, Fuzzy expert system.

ACKNOWLEDGEMENTS

I would like to take this as an opportunity to state my profound gratitude toward my first supervisors Dr. Francesco Nex and Dr.ing. Markus Gerke for their exemplary guidance and monitoring throughout this research. I would like to express my gratitude to my second supervisor Dr. Norman Kerle. A special thanks goes to my dear friend Mr. Felipe De Carvalho Diniz for his precious guidance.

My most lovely gratitude is for my family for their endless love, constant inspiration and encouragement whom I could not have done much without.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Motivation and problem statement

Among natural and man-induced devastating events occur all around the world, earthquake has a significant destructive effect on build-up area and infrastructure (Dell'Acqua & Gamba, 2012). After a catastrophic earthquake, damage assessment is the main source for obtaining the information about the level of damage, its geographic distribution and necessities (Fernandez Galarreta, Kerle, & Gerke, 2015). The speed and accuracy of the damage assessment are critical for the rescue, reconstruction (Kerle, 2013) and economic loss estimation. In order to evaluate damaged structures, there are two main means, ground-based and remote sensing based assessment.

Traditionally, damaged buildings are assessed through a field survey which takes time and requires specialists to visit the area, whereas it provides very detailed and accurate information about the level and the extent of the damage. Ground-based assessment is slow and in some cases dangerous or impossible to access the affected area (Kerle, 2013). On the other hand using remote sensing images for building damage detection provides the opportunity to have low cost and fast damage evaluation without any contact with the area which reduces the risk level for specialists (Dong & Shan, 2013). Very high resolution (VHR) satellite images are adopted by both visual interpretation and automatic damage detection approach. Li, Xu, & Guo (2010) extracted urban building damages from multi-temporal VHR satellite images. Ehrlich, Guo, Molch, Ma, & Pesaresi (2009) discussed earthquake damage assessment based on aerial photography and satellite imagery. In the case of building damage assessment by employing remote sensing (RS) optical data, several factors should be considered such as image resolution and angle of view. Many of the damage features locate on the facades which would not be visible from nadir-looking instruments. Moreover occluded parts of the structure can make it difficult to decide about the damage level of an individual building only based on RS data. Usually, RS damage mappers apply remote sensing data by incorporation of in-situ observation (Geiß, 2014).

The technology of RS data acquisition has been developed rapidly, i.e. Lidar, Radar, UAV, airborne oblique imagery and hyperspectral satellite images. Several researches have proposed various methods for Building Damage Assessment (BDA). Khoshelham, Oude Elberink, & Xu (2013) applied airborne laser scanner data to extract damaged and intact roof of the building. To obtain a better image analysis and comprehensive damage assessment, multi-perspective images are recommended. Thanks to oblique imagery it became possible to collect data not only from the roof but also from façades. Gerke & Kerle (2011) attempted to extract damage features from airborne oblique photographs and estimate the post-building damage score. Using Unmanned Aerial Vehicle (UAV) images which can be captured few hours after earthquake make vital information available from roof and facades for BDA. More detailed damage evidence is required to distinguish the low levels of damage by using remote sensing data. Vetrivel, Gerke, Kerle, & Vosselman (2015a) obtained damage evidence all around the building such as debris, cracks, holes, tilted elements and spalling from multi-perspective UAV images; sub-segment of damages automatically are extracted from imaged-based 3D point-cloud by considering spatial and spectral characteristics. The information provides the opportunity to distinguish different level of intermediate damages (e.g. no damage and moderate damage). A comprehensive and reliable damage scale has a significant role in achieving consistency in damage assessment.

The European Macroseismic Scale 1998 (EMS-98) was generated to have a guidance for ground-based BDA, and it is the second edition of the original 1992 version (Grünthal, 1998). In this scale according to the evidence a proper damage level is assigned to each building. Damage grades are categorized into five main groups according to building material (negligible to slight damage, moderate damage, substantial to heavy damage, very heavy damage and destruction). In some studies EMS 98 is adopted for evaluating the level of damage based on RS data (e.g. Gerke & Kerle (2011)), but it causes some limitation. The most important point that should be considered is that EMS98 is designed for field survey damage assessment. Hence, all types of damage evidence obtained from RS perspective are not included in this catalogue. In addition dealing with occluded parts of building and uncertainties are not supported.

Applying airborne and spaceborne platform data for surveying damaged area is an effective method in the field of building damage assessment. Detailed damage information is provided but still it is not possible to use them in an efficient way and achieve BDA (Dubois, Member, & Lepage, 2014). The question is how this valuable remote sensing data could be adopted to obtain an accurate and comprehensive damage assessment (Fernandez Galarreta et al., 2015). There is a lack of a comprehensive framework to link the knowledge of engineers and state-of-the-art technology of remote sensing data acquisition in the case of BDA. This research proposes a framework for representing, interpreting and integrating RS-based damage evidence with considering occlusions and uncertainties. The occlusion refers to the part of the building which is not observable in the RS data (e.g. one facade is occluded by another building, or one part of the wall is covered with debris). Partial evidence causes limitation to judge about the damage grade of occluded part of the building. Therefore, it is required to find a strategy to address this difficulty. Structural engineering, civil engineering and in situ knowledge are necessary to identify, integrate and interpret required damage evidence, attributes and relationships.

The importance of the building damage assessment after an earthquake is remarkable for rescue, reconstruction and estimating of geographic distribution. The accuracy and speed of this process play a significant role. Conventionally, buildings are evaluated by the specialist through a field survey. This approach is very time-consuming and strongly influenced by experience, skill and perspective of inspectors. State-of-the-art remote sensing technologies provide an opportunity to obtain damage information faster and cheaper. Although there have been several researches carried out to improve techniques for building damage detection, a reliable damage assessment requires semantic integration of remote sensing data based on structural engineering knowledge. This research proposed a framework to represent, classify and analyze detailed remote sensing data based on the semantic and spatial characteristics. Furthermore, the generated framework provides the detailed and overall building damage assessment automatically. The proposed research took a step towards improving the co-operation between remote sensing mappers and structural engineers.

A literature review is conducted to gain an understanding of the necessary information for building damage assessment. Different damage scales, catalogues and references are studied to obtain a holistic overview of the building evaluation process. Required and detectible damage evidence, attributes and relationships from remote sensing data according to the achieved knowledge are provided and classified in this research. The CityGML is an integrative information representation of visual, spatial and semantic properties of city objects hence in this research an extension to CityGML is proposed to store, analyze and represent geometrical, topological and semantic properties of damaged building entities. The 3D model of damaged building provides a visual interpretation of extracted damages for structural engineers in addition to the semantic and spatial properties of damaged building. The data is transformed into The PostgreSQL database which is cable to keep the data integrity and run spatial and non-spatial queries for reasoning. Also, an automatic damage assessment is provided in this study. Two fuzzy expert systems are designed to imitate the behavior of the specialists for building damage assessment. The generated systems capture the knowledge of structural engineers and involve the uncertainty of the decision making to evaluate the damaged structure. The first fuzzy expert system integrates semantic and spatial damage evidence and attributes related to the facade to calculate the damage grade of each façade separately. In the case of occlusion or incomplete 3D model of affected structure the detailed damage evaluation provides damage analysis of the visible parts. The second fuzzy system implements overall building damage assessment and computes the final building damage score.

The technical feasibility of the 3D damage modelling, database consistency and damage evaluation process of the proposed framework are analyzed. The results of damage evaluation of three simulated affected building indicate the proposed framework can integrate detailed damage evidence and calculate the accurate value as damage grade at façade and building scale. In addition, it is proved that the system successfully handles various levels of damage for computing reliable final damage value of the building. As a result, this thesis is a proof of concept for building damage assessment based on semantic and spatial remote sensing data.

1.2. Research identification

The main objective, sub-objectives and research questions are identified as follows.

1.2.1. Research objectives

The main objective of the proposed research is to identify a framework that can represent, interpret and evaluate damaged buildings after an earthquake based on remote sensing data. It adopts human expert knowledge and RS data to provide a faster and more constant assessment to reach damage grade of an individual building. The proposed framework supports different types and attributes of damage evidence extracted from facades and roof. Also, it is required to address the limitation caused by occluded parts of the building and uncertainties. Moreover, the framework is established based on the structural engineering knowledge and expert's experience and perspective. The main objective is achieved through the following sub-objectives:

1. Defining a reliable and efficient approach to represent RS-based damage evidence, attribute, spatial and non-spatial relationships between damage parameters and building elements in order to have a semantic interpretation of RS-based damage evidence.
2. Designing building damage assessment system by considering uncertainties of decision making.
3. Defining a strategy to incorporate occlusion in the framework to acquire a comprehensive representation of visible parts.

1.2.2. Research questions

4. Which damage evidence and attributes are essential for assessing damaged building?
5. Which semantic and spatial relationships between damage evidence and building elements are essential to be represented?
6. Which identified damages, attributes and relationships are detectible from RS data?
7. What is the most proper approach to store, represent and assess damage evidence and relationships?
8. Which method is reliable to involve uncertainty for building damage assessment?
9. How the generated framework can deal with the occlusion and gap?
10. Is the generated framework capable of assessing damaged building which contains damage evidence at different levels?

1.3. Innovation

Although using state-of-the-art RS techniques and technologies have developed building damage detection significantly, there is no method to make an overall BDA through extrapolating detectible damage evidence

(Foulser-Piggott, Spence, Saito, & Brown, 2012). European damage assessment catalogue, EMS 98, was generated more than 15 years ago. Although it is attempted to extend the mentioned catalogue to become useful as a standard scale for global use and involving additional material on various types of building (Foulser-Piggott & Spence, 2013), it is evident that EMS 98 could not support the process of BDA based on remote sensing data completely.

This research proposes a new framework for building damage assessment based on RS data to improve the co-operation between remote sensing specialists and structural engineers during earthquake damage evaluation. It presents an extension to CityGML to model the affected building which provides a comprehensive representation of the structure with available damage information for experts. For the first time, the CityGML is modified to model damaged building. The 3D model can preserve spatial-semantic relationships among different elements. In addition fuzzy expert system is adopted for the phase of reasoning to assess damaged structure, which can mimic the behaviour of a human expert in problem-solving cases contain uncertainty. The 3D city model that includes geometries and attributes of structure and damage evidence are stored in PostgreSQL database. Using PostGIS in this framework allows running spatial queries since the location of damages besides their properties play an important role in BDA. This framework can improve, speed up and simplify the process of building damages assessment after an earthquake through providing a first indication of the typologies of damages to structural engineers in an automated way.

The proposed framework is the first model to integrate detailed RS damage information for building damage assessment automatically. It represents the base for a more detailed damage evaluation. Hence, it could be extended in the future to achieve a complete BDA model. The knowledge and experience of structural engineers and experts are considered to generate a holistic representation of the situation of damaged building only based on remote sensing data. The system classifies and interprets RS-based damage evidence around the building to provide the opportunity for field survey experts to have an overview of the structure and decide about the level of damage of each building from their office. The framework assesses detailed damage information to distinguish the low levels of damage and obtain more reliable and accurate BDA. Moreover, the designed expert system is able to address the limitation caused by occlusion and incomplete damage information of some parts of the building. Therefore, the framework is expected to close the gap between RS mappers and field-survey experts in the case of BDA by integrating state-of-the-art remote sensing technology and knowledge of specialists.

1.4. Related work

Damage scales, data collection technologies, damage mapping and detection methodologies involve in the process of RS-based building damage assessment. An overview of the previous works and researches regarding this study is presented as the following.

1.4.1. Earthquake building damage mapping

One of the most frequent methods for earthquake damage mapping is using post-event optical data (Dell'Acqua & Gamba, 2012). This approach generates valuable information for many purposes such as search and rescue (SAR), emergency response and estimation of reconstruction and economic loss. In this method also reaching the detailed ground-based damage assessment is remain as a challenge, many researches and methods have been conducted to improve the capability of remote sensing data for earthquake damage evaluation. Different properties such as geometry relationship, texture, structure, shadow and shape have been employed to detect structure damages after an earthquake; Dong & Shan (2013) provided a comprehensive evaluation based on different RS methods for building damage assessment. Visual interpretation is one of the basic methods for damage assessment; especially in the case of nadir-looking, even with sub-meter resolution, only roof surfaces are visible, and there is no clue of damages of facades which play a critical role to distinct low levels of damage. Therefore, in the case of heavy damaged and collapsed building nadir-view images would provide useful information for emergency management after natural hazard. Yamazaki et al. (2004) employed Quick Bird images after Zemmouri, Algeria earthquake 2003 for visual interpretation. Different interpreters analysed images visually to assign proper damage grade to an individual building based on EMS98 catalogue (e.g. debris around: grade3, partial collapsed: grade4 and completely destroyed: grade5). Also, pre-event images of the area have been used for the validation. In collapsed cases the results from various interpreters are much closer but in lower levels of damage (no damage, moderate damage and heavy damage) judgment of grade is difficult and it causes large difference among assigned damage grades. Ishii et al. (2002) proposed damage detection based on colour and edge. It is assumed that the brown objects with a uniform orientation of edges represent damaged area; mentioned assumptions cause some limitations for applying this method. . Kerle (2010) discussed the limitation of image-based damage mapping by assessing the damaged maps produced after the 2006 Indonesia earthquake.

1.4.2. Building damage detection

Oblique imagery provides the opportunity to collect data from facades and roof of the buildingGerke & Kerle (2011) applied multi-perspective Pictometry® images for semi-automated building damage mapping, a pre-building damage grade based on EMS98 is obtained after classification (no/moderate damage, heavy damage and destruction). Kerle & Hoffman (2013) discussed how the different perspective of Pictometry® and satellite images can lead to a dissimilar level of damage. Fernandez Galarreta et al., (2015) discussed the usability of UAV-based 3D point-cloud and object-based image analysis for damage detection. Detailed damage parameters are detected through segmentation by applying geometric and radiometric features such as spalling, hole and inclination in facades (Vetrivel et al., 2015a). In order to improve the process of safety and post-earthquake assessment, an automated damage index estimation is introduced for analysing reinforced concrete (RC) columns. In this method columns, cracks, spalling and their properties are extracted from images (Paal, Jeon, Brilakis, & DesRoches, 2015). Also, Zhu, German, & Brilakis (2011) adopted images to detect cracks on concrete structures.

Ma, Sacks, & Zeibak-Shini (2015) proposed a data model of damaged RC building based on the Industry Foundation Classes (IFC, 2013) schema. The extended schema is used for representing surface cracking, spalling and delamination, bending and buckling, shearing and breaking parts of the structures after an earthquake (Ma et al., 2015). Torok et al., (2012) introduced 3D reconstruction method for image-based 3D crack detection algorithm. For safety assessment and damage estimation of reinforced concrete building a new crack classification index is presented by Farhidzadeh, Dehghan-Niri, Moustafa, Salamone, & Whittaker (2013).

1.4.3. Damage scale

There is no coherent description of building damage assessment. Organizations in different countries all around the world have their references for building damage assessment. They consider various factors such as environment, structural design and material of building and previous experiences to design a damage scale. Furthermore, very heavy damaged and collapsed buildings are detectable very well from remote sensing data, but lower levels of damage (such as slight damage, moderate damage and heavy damage), pancake collapse and deformation due to liquefaction are challenging. Hence various grading schemas are designed according to data type and resolution. The Japanese Prime Minister's office proposed a damage evaluation frame comprising four levels (no damage, moderate damage, heavy damage and major damage) (L. Dong & Shan, 2013). Matsuoka & Yamazaki (2005) studied very high resolution (VHR) optical satellite images captured after the 2003 Bam earthquake and classified the results into four grade according to EMS98. Consequently, there is no agreement on the definition and number of damage levels in an RS-Based building damage assessment. European Macroseismic Scale (EMS 98) categorized building in two main groups, masonry and reinforced concrete. Damages related to each cluster are classified in five grades. Grade one: negligible to slight damage, which belongs to no structural damage and very slight non-structural damage. Grade two: moderate damage, this class contains building with slight structural damage and moderate non-structural damage. Grade three: substantial to heavy damage, structural elements with moderate damage and non-structural parts with heavy damage are grouped in the third class. Grade four: very heavy damage, it includes heavy structural damage and very heavy non-structural damage. Grade five: Destruction, it refers to building which contains very heavy structural damage or totally collapsed (Grünthal, 1998). Figure 1 illustrates the five level of damage of masonry and reinforced concrete building based on EMS95 classification.

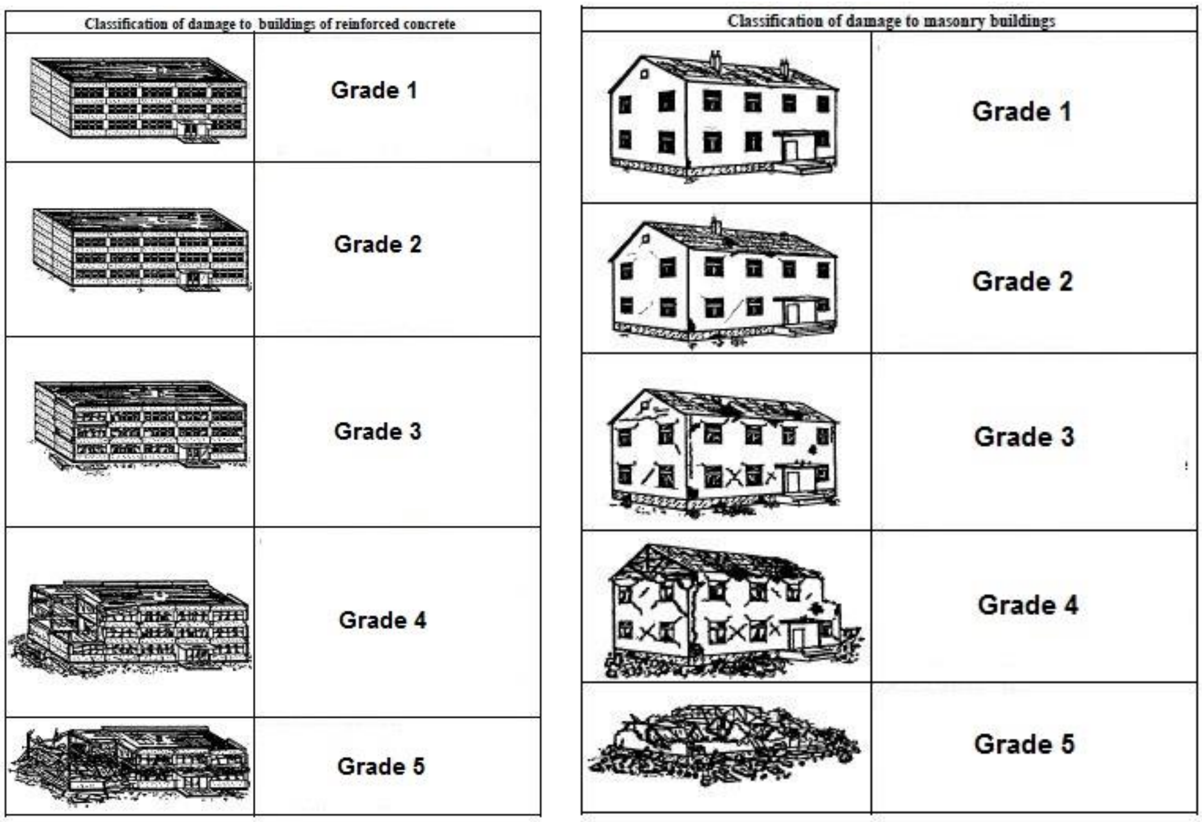


Figure 1: Building damage classification based on EMS 98 (Grünthal, 1998).

1.4.4. Expert systems

Expert systems are computer programs that capture knowledge of human experts to solve complicated decision-making problems. Therefore, in this research, the expert system is adopted to make a decision and assess the damaged building automatically. Shu-Hsien Liao (2005) classified more than 150 articles to explore the development of methodologies and applications of Expert System (ES) during the period 1995-2005. It categorized ES methodologies in eleven classes: fuzzy ESs, rule-based systems, object-oriented methodology, neural networks, system architecture, database methodology, knowledge-based systems, intelligent agent systems, case-based reasoning, modeling and ontology. A review of expert systems and their application in the area of science before the 90s is presented in Durkin (1990). Hopgood (2001) explained a broad range of intelligent systems techniques besides the practical view of their applications. This book presented a detailed analysis of design and implementation of expert systems with their pros and cons. The fuzzy expert system is the most proper approach to involve the semantic uncertainty in the decision-making process. Fuzzy logic is able to handle linguistic uncertainties and considers all possible solutions according to the membership degree.

1.5. Thesis approach

The research consists of four main phases: literature review, data modelling, design expert system and evaluation of the proposed framework.

Literature review

In the first stage of the study, a holistic overview of required information for building damage assessment is provided. The damage information is collected through reviewing previous works, analysing damage scales and consulting with experts. In parallel, valuable and useful RS-based damage evidence, attributes and relationships are identified.

Data modeling

The second phase proposes an extension for generating a 3D model of the damaged structure. The CityGML model is a proper 3D modeling language because it is an integrative information model to represent city objects according to their geometric, semantic and visual attributes.

Design expert system

In order to analyse and assess the level of damage to the building, a fuzzy expert system is generated in the third step of the thesis. The fuzzy expert system captures the knowledge of the human experts to integrate different damage evidence and involve semantic uncertainties to make a decision about the detailed and overall damage grades.

Evaluation of the proposed framework

The last step is evaluating the feasibility of the generated framework under different damage conditions.

1.6. Thesis outline

Chapter 1: This chapter includes motivation and problem statement, research objectives and research questions, innovation, related work and thesis approach along with the research outline.

Chapter 2: This chapter consists of concepts, methods and information that are used in the proposed research: building damage assessment information, 3D CityGML model and Fuzzy expert system.

Chapter 3: The third chapter of this research explained the methodology of generating RS-based building damage classification and rules. Also, it includes the process of generating 3D CityGML model of damaged building. In addition importing data into PostgreSQL database and coding fuzzy expert system in Python programming language are described in detail.

Chapter 4: It includes the results and discussion of damage assessment of three simulated damaged building.

Chapter 5: The last chapter includes the conclusion formulated by answering the research questions, recommendations and limitations.

2. THEORETICAL BACKGROUND

This chapter presents information, methods and concepts that participate in the framework. The first part contains a summary of required information for building damage assessment according to the studied damage scales and consulting with specialists. It also discusses the possibility to detect required data from remote sensing data. The second part presents the feasibility of CityGML to model damaged building. The final section provides an outline of the concept and application of the fuzzy expert system for building damage assessment.

2.1. Building damage assesment information

Knowledge and experience of structural engineers are required to recognize, interpret and analyse damages of the building. Therefore, it is critical to identify what type of damage information is important for structural engineers to evaluate damaged building and also which of the required damage information is possible to detect and represent by applying remote sensing data. The first part of this chapter focused on identifying damage properties and their spatial-semantic relationships according to construction and structural engineering knowledge. Following building damage information and its attributes which are important to have BDA are represented.

In this research different damage scales and catalogues are studied to have a comprehensive framework which can be useful in various cases. It was useful to discover the importance and the role of building elements, damage features and information in BDA. EMS 98 (Grünthal, 1998) is the most common reference in Europe for BDA. It is attempted to develop it as an international damage scale. In addition guidance and catalogue for rapid damage assessment of U.S. Federal Emergency Management Agency (ATC, 2005) and Questionnaire for detailed building damage assessment after 2015 Nepal earthquake are considered in this study. Furthermore, consulting with structural engineers and other information sources such as research on comparison among different damage scale and proposed new damage chart e.g. Okada & Takai (2000) are included. Considering the various source of earthquake damage assessment provides an outline of required damage information for different application.

Furthermore, the building damage evidence and attributes that are extractable from RS data based on recent technology and methodology are studied and classified. Several studies carried out into building damage detection after an earthquake. They used various data types and methods to extract evidence and their properties. Recently it is possible to detect holes, spalling, tilted surfaces and columns (Vetrivel, Gerke, Kerle, & Vosselman, 2015b), cracks and their attributes such as direction, width and length (Zhu et al. (2011) & Paal et al. (2015)). Figure 6 illustrates classification of important parts of the building and related RS-based building damage information. It shows the process of dividing one building into its sub-elements and demonstrates what types of damages are possible to measure for each element. One of the factors that should be considered is the damage accumulation can cause a higher level of damage. Beside mentioned sources a questionnaire has been designed to get the view of the specialist about the relationships and their influence on the structure, the questionnaire is represented in Appendix 1.

As discussed above various BDA applications require different criteria and information. Based on results of study in structural engineering and building damage assessment domain various types of damages information and their spatial-semantic relationships and detectible damage information from RS data are explained as follows.

2.1.1. Wall failure

Two important factors of the wall failure are location and size of the damages. Collapsing the non-bearing wall is not a sign of serious damage but for economic loss and insurance issue should be considered. Also load-bearing wall failure refers to the high levels of damage. Collapsed part is one of the critical damage features that is possible to extract from generated 3D point cloud of affected building automatically (Vetrivel et al., 2015a) Thanks to oblique imagery the failures of the roof and façade are visible in the dataset and the area of that is measurable. In the most of the cases, failed roof reveals the heavy damage and have effect on the building functionality.

2.1.2. Crack

The location, width and shape of cracks show the level of severity of damage. The location of the crack is vital. For example, a crack in the structural element is a sign of severe damage in the structure, but the same crack on an infill wall shows a low level of damage. In addition, the severity of crack is another attribute that should be considered. based on 2015 Nepal earthquake physical damage and safety assessment catalogue, the cracks according to their width are classified as follows: severe (crack widths >1/4" (6 mm)), Moderate (crack widths > 1/8" (3 mm) but less than 1/4"), Hairline cracking. The direction of crack also reveals a different level of damage. Diagonal cracks show the serious destruction of the structure. Vertical and long cracks in the wall corners appear because of drift between walls. Figure 2 and Figure 3 are examples of mentioned damages caused by earthquake shakes. Crack in non-load bearing partitions has less importance than load bearing walls. Also, crack propagation can have an effect on the stability and safety of the building. The crack and spalling of the column-beam joint reveal that the column could not support the structure anymore.

Now it is possible to detect cracks in the outside parts of the building (Paal et al. (2015) & Fernandez Galarreta et al. (2015)). As mentioned before, important properties of crack that can give us the information about the level of damage are the position, shape, width and length of the crack. All mentioned properties are measurable from RS data (e.g. Zhu et al. (2011)). It should be noted that other factors such as thickness of building object are necessary to be considered in this classification but in the case of BDA it is not possible to obtain required information remotely.

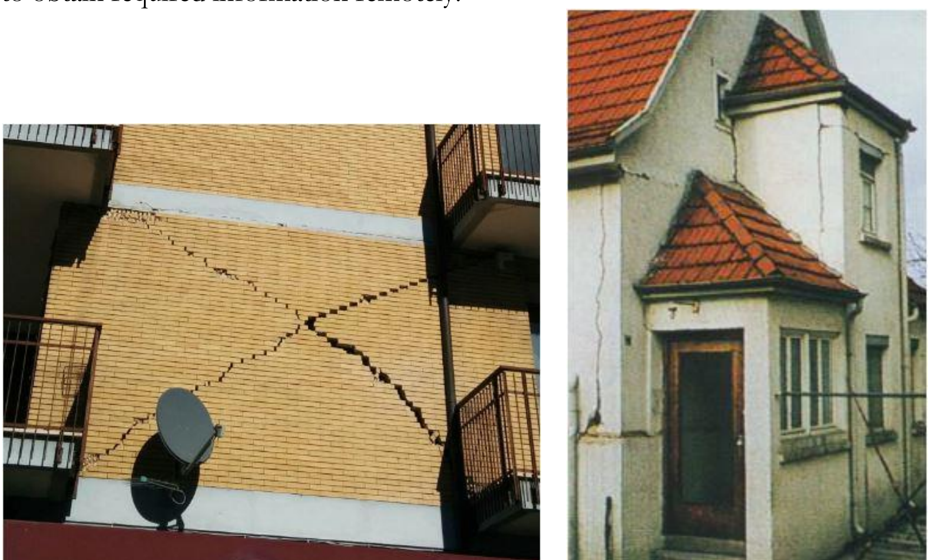


Figure 2: Left image illustrates the intersected crack between windows. Vertical cracks in rigth image reveal the separation between walls (Grünthal, 1998).

Figure 3: Cracks in RC column of building, Image is taken from Haiti earthquake 2010 (Miyamoto, 2010)

2.1.3. Spalling

The extent and location of spalling are important for the evaluation. The spalling of the columns and beam-column joints are essential to participate in the framework (Figure 4). Also, the extent of the spalling in walls represents the low levels of damage. Recently several researches attempted to extract spalling from remote sensing data; this information, especially in the case of column and beam, are one of the important sign of serious damage to structural parts. Spalling on column, beam, structural and non-structural walls are observable and their area is measurable.

2.1.4. Tilted building elements

In general tilted structural parts, e.g. tilted column, can lead to very high level of damage (Figure 4). Although discovering a slight tilt of structural elements are not easy to detect by human eye, but that is possible to measure it from 3D point cloud. The tilted column can reveal the bending of the connected beam and put the building in an unstable situation respect to the position of the column and degree of inclination. Another critical damage evidence is inclination of the whole building, which is the sign of high level of damage.

Figure 4: Left image shows spalling in beam-column joint, right image illustrates tilted building because of collapsed ground floor.

2.1.5. Damage information of inside of the building

Inside survey provides information for detailed damage assessment especially in the case of multi-storey building. The crack of inside walls and columns, failure of celling and partitions and other damage information of inside are not detectible in remote sensing data. Inside information is not possible to obtain remotely from the affected area. Remote sensing data can provide detailed damage information of outside of the structure but the generated framework has the capability to involve the inside damage evidence and information about the year of construction and material of structure can be obtained from cadaster organization.

2.1.6. Damages of roof

Observing damages of the roof is one of the important factors in BDA. Usually, damages to the roof lead to higher level of damage and also have an effect on the safety and functionality of the building. In detailed damage assessment, the failures of roof elements such as chimney and parapet should be considered. Figure 5 illustrates building lost part of its roof.

Figure 5: Damaged roof after 2011Christchurch's earthquake.

2.1.7. Other information

The following more required information for building damage evaluation is described:

- Debris: The location and amount of remained rubble are necessary for emergency management organizations and rescue.
- The material, type of the building and year of construction have remarkable influence in BDA and economic loss estimation. Furthermore, this information can be used for Intensity assessment.
- Fall of pieces of plaster and mortar parts of the joint building elements.
- Damaged door and window are considered in detailed and functionality evaluation e.g. jammed doors can cause the loss of serviceability.

The term of the structural part of the building refers to columns, beams and structural walls, non-structural parts are infill walls, gable roof, window, door and balcony. Visible damage evidence such as crack and spalling that appeared on column's surfaces can be used to detect the level of severity of the damage after an earthquake. These damages caused due to limited compressive and bending resistance of reinforced concrete columns. Buildings contain two type of the walls, load bearing wall and non-load bearing wall. Load bearing walls are the structural part of the building and hold it up. However, the non-structural walls are used as a partition to separate different rooms of a structure. It is obvious that damage on load carrying walls are more severe and can put the whole of the building in a high damage situation. Nowadays most of the small masonry buildings are designed in a way that all walls play the role of structural part. Frame structures are constructed from stronger and lighter material, which can resist against the earthquake better.

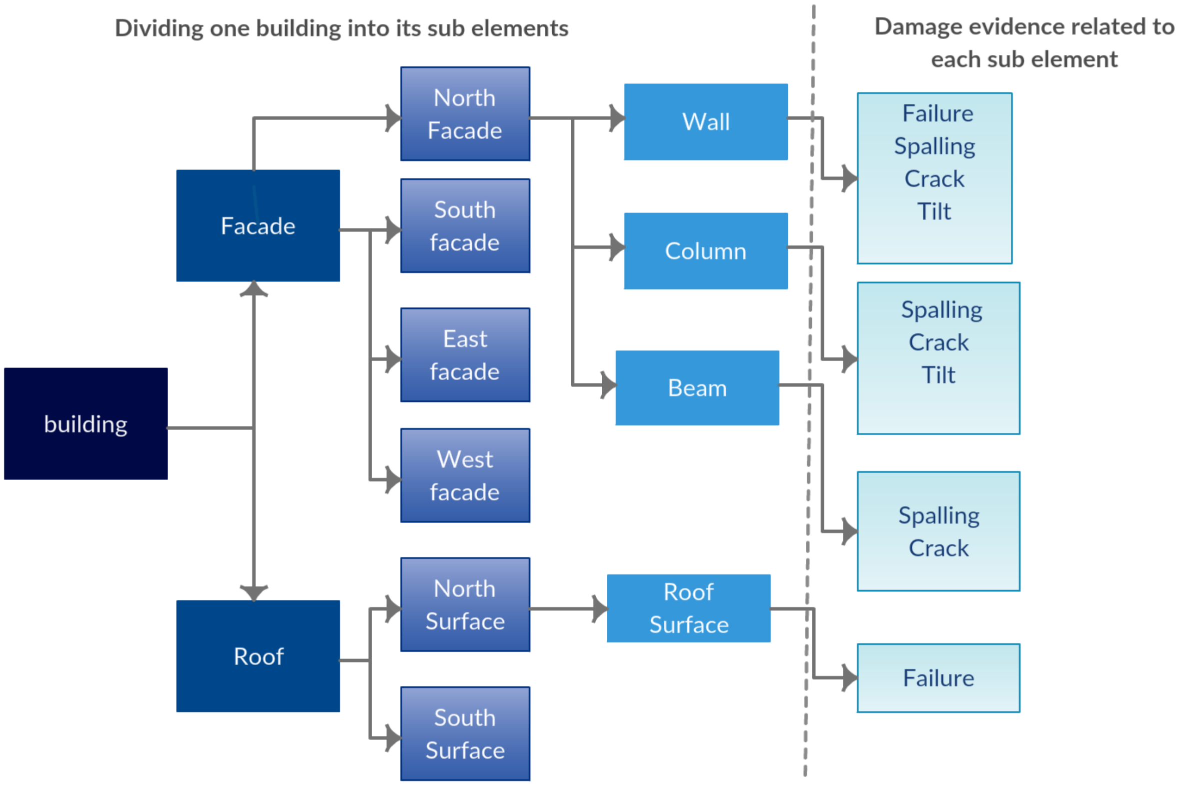


Figure 6: Building elements and related damage evidence illustration.

2.2. CityGML

Nowadays virtual 3D city models are used for various application such as disaster management, urban planning, cadastre and mapping, navigation and environmental simulation. Each application requires different city objects, attributes and relationships represented in a standard framework. Semantic 3D model comprises graphical, spatial and ontological aspects that cover attributes and interrelationships among object elements (Kolbe, 2009). City Geography Markup Language (CityGML) is an international standard for storing, representing and exchanging of semantic virtual 3D city and landscape models. CityGML is implemented based on OGC's Geographic Markup Language 3.1.1 (Gröger, Kolbe, Nagel, & Häfele, 2012). Daum, Borrmann, & Kolbe (2015) proposed a new spatial-semantic query language, QL4BIM, which supports integrated processing of CityGML and IFC models. Building elements and their semantic relationships can be modelled in Industry Foundation Classes (IFC), on the other hand, CityGML is a well-known model to represent various viewable parts of the building in separate classes. Therefore, it introduced an integrated middle level to analyse and make a decision by applying information from both data model. This approach prevents the loss of information during conversion between models.

Increasing use of CityGML for analysing, simulation, 3D visualization and exploration of entire city proves its enormous potential in the field of city planning (Nouvel, Zirak, Dastageeri, Coors, & Eicker, 2014). A successful sample of adopting CityGML model for analysis the energy consumption at city scale can be found in Agugiaro (2015). First the 3D city model of a part of Trento in Italy is generated based on CityGML standard and by using 3DcityDB the city model is imported into a PostgreSQL. Borrmann et al., (2014) shows that CityGML is a proper 3D model in the case of planning multi-scale urban facilities. It can preserve semantic and geometric information in five levels of detail. Also, Nouvel et al., (2014) proved that CityGML is proper for energy analysis at large scale, and it can be even used for national mapping. It presented a new methodology to compute and analyse the energy consumption of 14000 buildings in Ludwigsburg by adopting semantical and geometrical information.

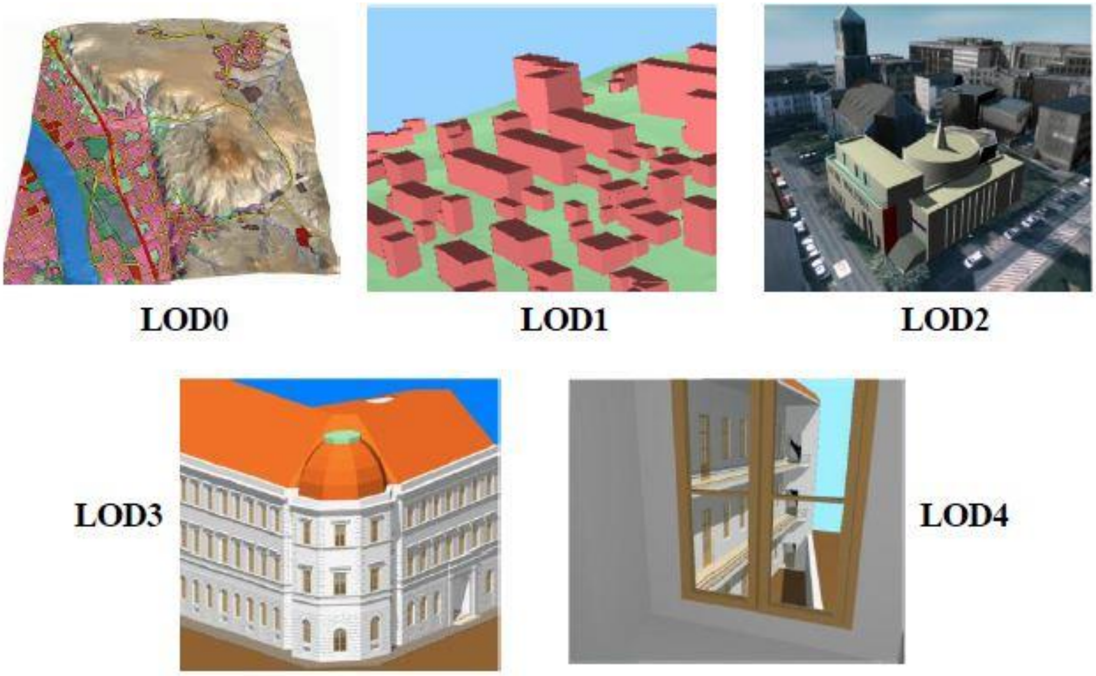


Figure 7: Level of detail (LOD) in CityGML (Gröger et al., 2012).

There are very few studies which used a 3D model for damage assessment such as Dong & Guo (2012) that defined an automatic assessment framework to detect damaged building through analysing the LIDAR data. 3D model of the building is generated by using footprint of structure in GIS software. After that, a comparison between pre and post 3D model is implemented. The mentioned framework is useful to detect severe damaged building. However, 3D CityGML model has not been used for building damages assessment based on post-RS data.

The most important types of virtual 3D city objects have defined in separate classes in CityGML. Thematically it is decomposed into core module and thematic extension modules. This 3D data model supports conjunction with core module with a combination of different extension modules based on the project application (e.g. transportation, bridge and city furniture). In addition, CityGML includes five dedicated Level-Of-Detail (LOD). One object can be represented in different LODs in one data model according to the application and data resolution. Also, it is possible to combine two CityGML model with data at a different level of details (Gröger et al. 2012). Figure 7 demonstrates the five LODs defined in CityGML.

- LOD0 represents building by footprint in 2/5 dimensional of Digital Terrain Model (DTM)
- LOD1 models building in the shape of boxes with flat roofs.
- LOD2 includes boundary surfaces and roof structures.
- LOD3 adds architectural details of walls and roof.
- LOD4 consists LOD3 and interior information of building.

Practical applications would contain various city objects that are not included inside the CityGML also some specific attributes of defined objects would not be covered. Generic is one the thematic extension modules in CityGML, which allows extending this 3D data model. It consists of two classes: Generic City Object and Generic Attribute. A new object (i.e. the building damage) can be presented to the Generic City Object class. To add a new attribute to the object that exists in the model Generic Attribute is helpful, as it integrates new attribute to the model without changing the XML schema. Figure 8 illustrates the UML diagram of Generic Objects and Attributes.

The data type of new attribute inside Generic Attribute can be Integer, Double, URI, String, Date and Measure that allows defining the unit of measure. A Generic City Object can have GML3 geometry and other attributes such as function, class and usage.

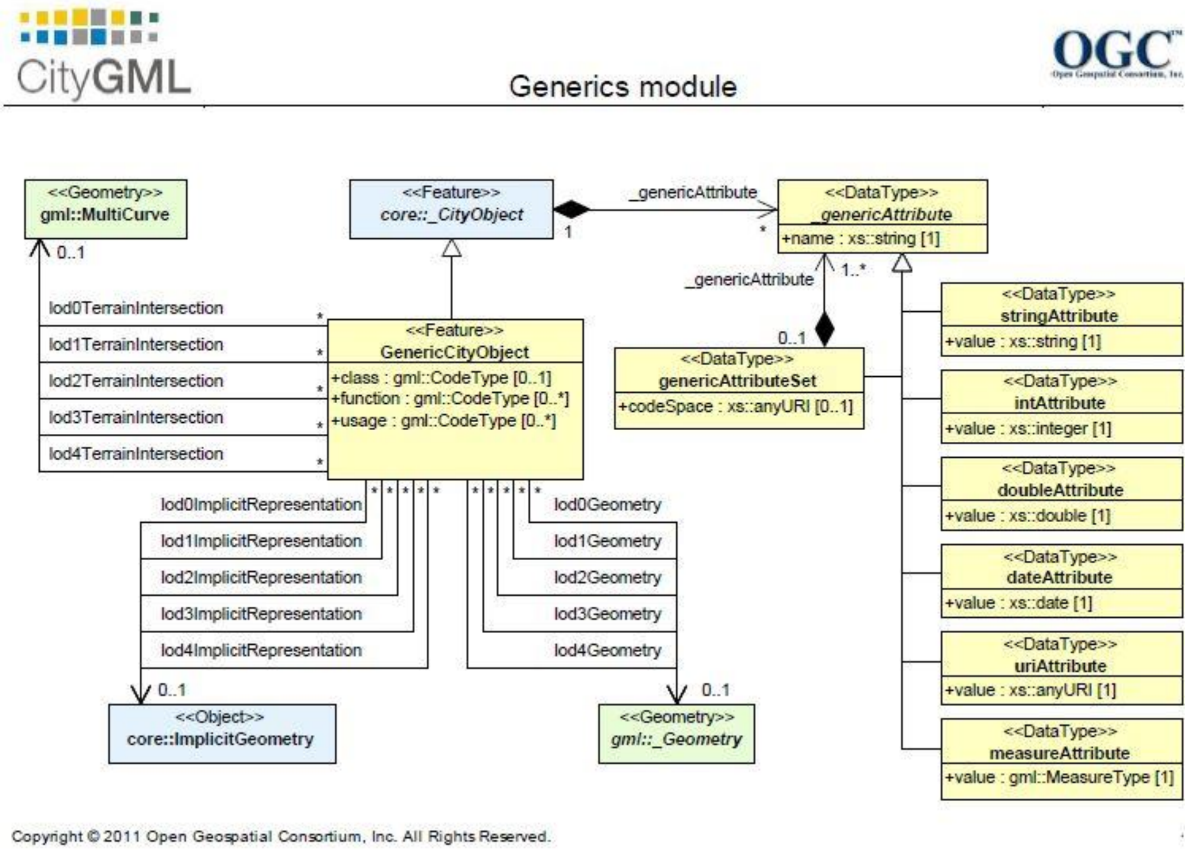


Figure 8: UML diagram of Generic Objects an Attributes in CityGML (Gröger et al., 2012).

2.3. Fuzzy expert system

The concept of artificial intelligence is directed toward creating machine which can mimic human mental capabilities, understanding, recognizing and reasoning (Hopgood, 2001). One of the areas of artificial intelligence is Expert Systems (ESs) that is defined to emulate problem-solving skill of human expert through reasoning (Durkin, 1990). The first ESs are designed in the 1970s and expanded in 1980s. Expert systems obtain the knowledge of human experts in a particular field and code this information in a computer program to provide the knowledge of specialists for less experienced users. Expert systems have the ability to simulate the behaviour and judgment of experienced and human expert in a specific domain. The performance of expert systems to solve complicated problems can be very reliable and as good as experienced human (Tripathi, 2011). Another valuable characteristic of ESs is that they have the ability to involve uncertain data to the system.

Vague, inadequate, incomplete and not reliable data involve uncertainty in expert systems. Although by improving data resolution and method of data acquisition can make some reduction in the amount of uncertainty, the linguistic concepts always contribute uncertainty in the system. The uncertainty can be categorized into two main groups: semantic uncertainty and evidential ambiguity. In the case of BDA since it is not possible to have a crisp damage classification and each building is assessed individually based on its damages and topology. For example large number of cracks in concrete structures can lead to different levels of damage because of different perspectives of specialists. The concept of large is vague and different from another point of view. Boolean logic applies crisp distinction and assign 0 or 1 membership to each object in a class. For instance, if there are five cracks in non-structural parts of building it would be low level of damage, but six number of cracks in non-structural elements can change the result to moderate level of damage. On one hand small differences inside input data lead to significant differences in output in Boolean logic, which can create unreliable results. On the contrary, there is no crisp classification in damage scales for building damage assessment, the linguistic concepts are the base of classification in BDA references. Also, experts based on their experience, knowledge, condition of the building and their conception about the linguistic definition on the damage scale (e.g. large part of the roof collapsed) assess the affected building. Therefore, it is essential to involve uncertainty in expert system in order to have a reliable BDA and prevent mistakes caused by ambiguity.

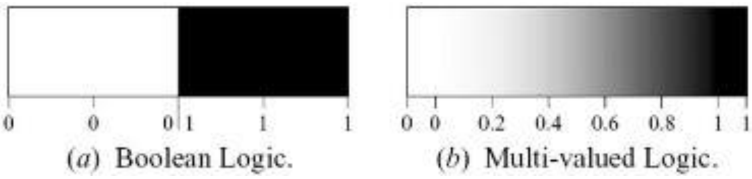


Figure 9: The range of logic value based on Boolean and fuzzy logic (Kahani, n.d.).

Fuzzy set theory is capable of dealing with linguistic concepts. In addition Fuzzy technique is suitable for dealing with uncertain semantic information in problem-solving cases. Fuzzy logic is a multi-valued knowledge representation based on mathematical principle. It is considering continues degree of membership between 0 (not a member) to 1(a full member) for each object in different classes (Kahani, n.d.). Membership functions involve vague classification and ambiguous data. Fuzzy expert systems are capable of assigning one entity to more than one classes based on its membership degree. Figure 9 illustrates the range of logic value based on Boolean and Fuzzy logic.

Jan Lukasiewicz, Polish philosopher and logician, in the 1930s introduced n-valued logic. In 1965, Lotfi A. Zadeh extended this theory and defined fuzzy sets of objects with the continuum degree of membership (Zadeh, 1965). The fuzzy expert systems are decomposed of basic components namely: the universe of discourse, fuzzy set, fuzzy membership function, fuzzy rule and defuzzification. The definitions of mentioned components are explained below:

- The universe of discourse (U): the range of all possible value of each input entity inside a fuzzy system is called universe of discourse.
- Fuzzy set (F): It is a set of members with a membership value in this set in the interval [0,1].
- Fuzzy membership function (μF (x)): this function calculates the degree of membership of each member (x) in a fuzzy set (F). The process of deriving this membership is named fuzzification.
- Fuzzy rule: They are conditional statements that combine the membership value of entities belonged to fuzzy sets. Fuzzy rules are defined to mimic the behaviour of a human expert in different condition.
- Defuzzification: the last step is driving a crisp value from output aggregation.

Mamdani technique is one of the most acceptable fuzzy interference methods for capturing expert knowledge. Figure 10 shows a detailed description of Mamdani system. The most common defuzzification technique is centroid which finds the centre of gravity of output aggregation where the accumulation is divided into two equal mass (Hopgood, 2001).

The required information, method and concepts involved in this research are explained above; the next chapter presents the methodology of adopting mentioned techniques and data to generate building damage assessment framework.

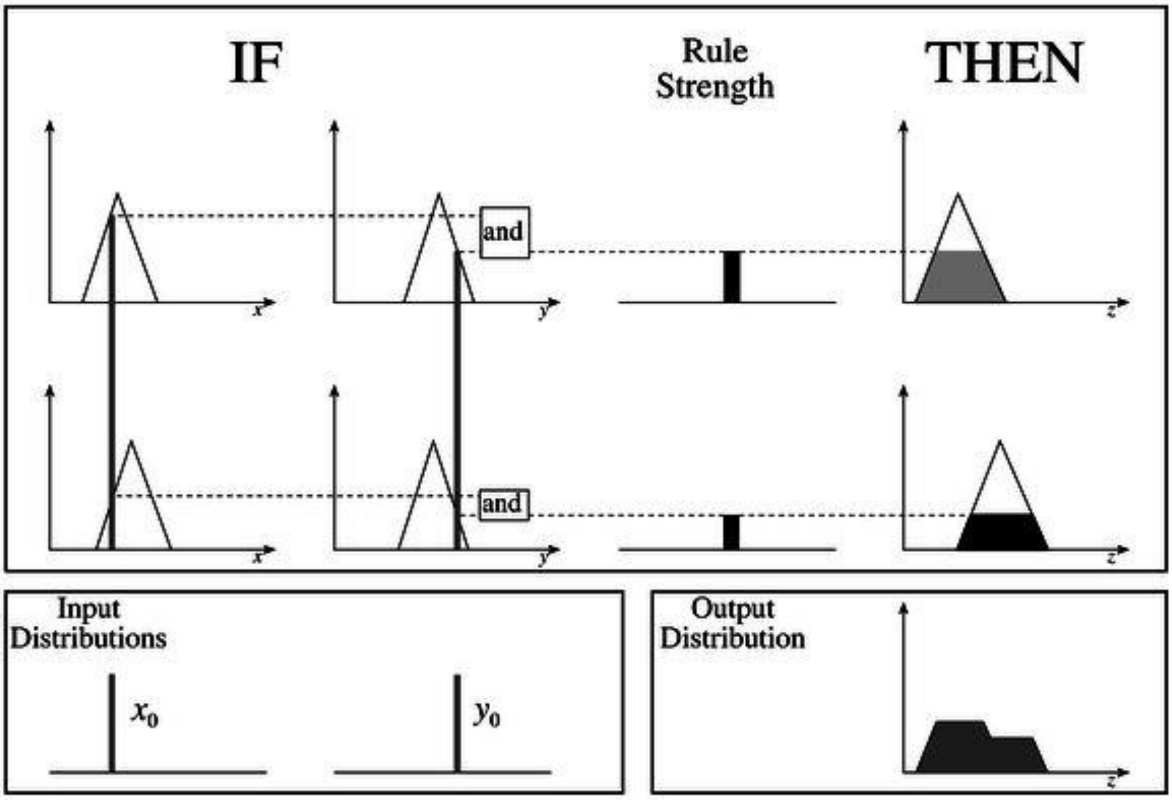


Figure 10: Mamdani interfrence system (Knapp, 2004).

3. METHODOLOGY

In this chapter three main stages of generating building damage assessment framework are described:

- 3D modeling of affected building in CityGML
- Importing 3D city model into the database.
- Generating fuzzy expert system.

The flowchart of the methodology is illustrated in Figure 11. In the 3D modeling phase, the CityGML model of the building is extended to display all types of available damage information that are described in 2.1. The data type and proper place to model required damage attributes are defined. This fact that damages should be retrieved accurately and completely through spatial queries in the fuzzy expert system plays an important role in defining the data type and attribute of damage evidence inside the model. The generated model supports the presentation, analysis and to explore the affected area. In fact, it simulates the field survey process in a three-dimensional environment. In addition, it creates the opportunity for structural engineers to assess damaged building based on remote sensing information. Another purpose of this research is to reach building damage evaluation automatically hence a fuzzy expert system that can imitate human expert understanding, reasoning and decision making by considering uncertainties is generated for BDA application. The process of designing fuzzy system is explained in 3.3. In order to implement reasoning, it is required to use an interface to import model into the database which has the capability to preserve spatial and semantic objects and attributes of 3D dataset.

The second phase consists two parts: choose a proper database and select reliable interface. Based on the criteria and application of the framework PostgreSQL database is chosen which allows running spatial queries to retrieve the semantic-geometric relationship of damage evidence and building elements. Also through testing different methods, the most proper transformation tool is adopted to import 3D model into database. It is critical to transfer data completely and correctly into database.

The last stage is design fuzzy expert system for building damage assessment. The knowledge of structural engineering and fuzzy logic are incorporated to generate the system in Python programming language. The proposed fuzzy system has the ability not only to connect directly to the database and query geometries but also to read the damage information from an auxiliary file. It makes the fuzzy expert system independent from previous steps. Therefore, in can be used in other application regardless of the 3D modeling method and database processing.

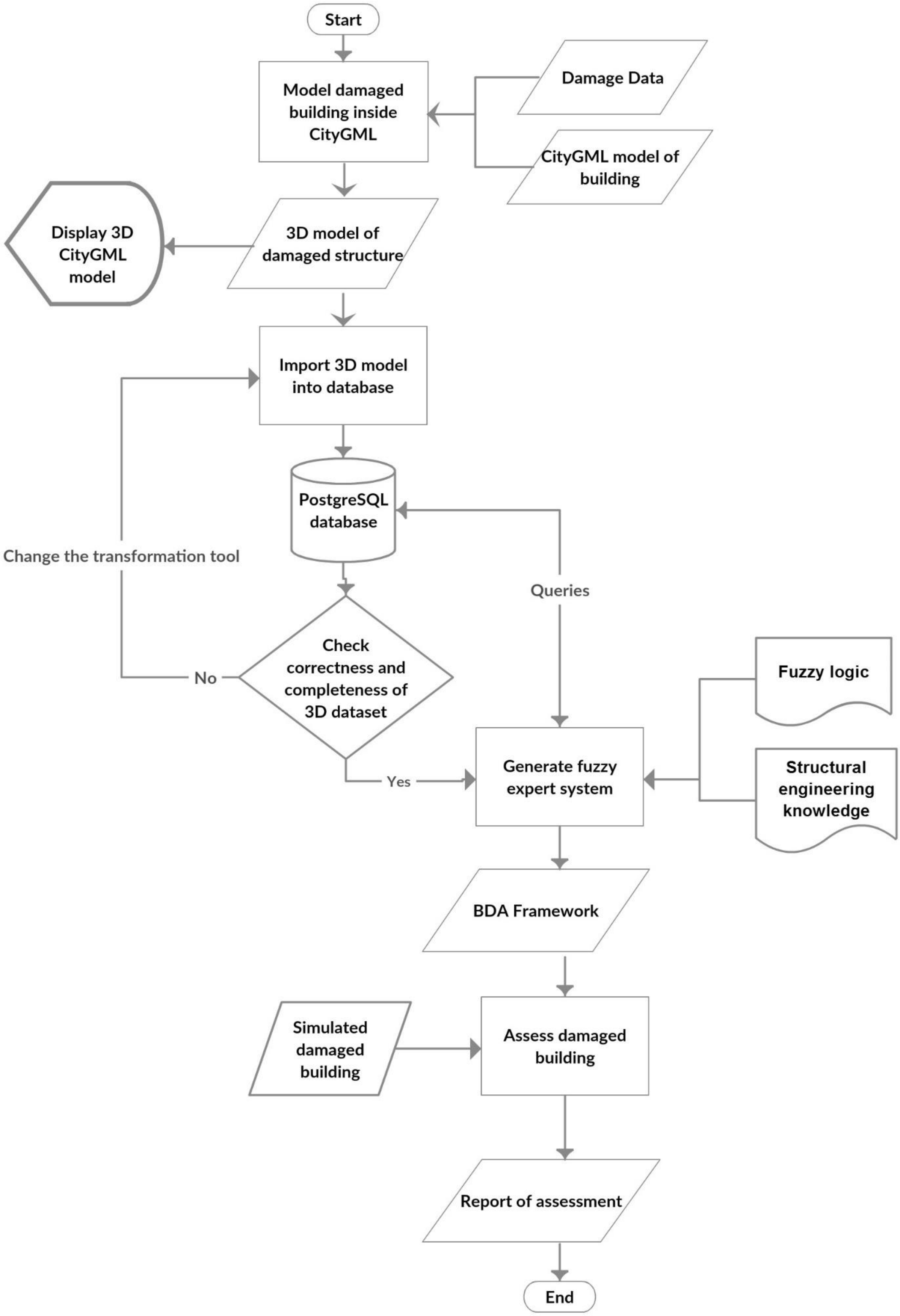


Figure 11: Flowchart of methodology

3.1. Generating CityGML model of damaged building

The details of the 3D modeling damage evidence and attributes are described as follows. Level of details (LOD) 3 is selected to represent damaged building elements and damage properties, because remote sensing data contains only information about the outside of the affected structure and also detailed building elements should be included in the model to represent the location, relationship and distance between damaged objects (more information about different level of details in CityGML format is explained in 2.2). LOD3 classified all relevant external parts of a building such as: window, door, column and beam. There is no defined classes for damage features but it has the capability to be extended for different user cases and for the first time CityGML is adopted to model damaged structures. The proper, data type, unit, location and properties are defined in 3D model that allows to represent and retrieve information completely and accurately in next phases of the framework. Detectible damage information from RS data that is discussed in 2.1 is modelled as Generic city Object and Generic Attributes. RS damage modeling is explained in details below:

1. Crack

   The crack in façade, column or beam is a new entity with three-dimensional geometry and attributes in CityGML. Hence, the *Generic city Object* module is used to model cracks and their attributes. Damage and Crack are introduced as *function* and *class* of this new object respectively. 3D geometry is imported as multi-surface. Also width and direction of each crack are included as object attributes. Figure 12 demonstrates an example of modeling two cracks between windows in CityGML model.

2. Failures

   The collapsed part of the roof and facades are modeled as new *Generic city Object* which its *function* is Damage and its *class* is named Hole. Same Crack, the 3D geometry of collapsed area is imported as multi-surface.

3. Spalling

   Since the spalling is an object that would have overlap with wall surfaces, column and beam is introduced as new *Generic city Object* and its *class* is Spalling under Damage *function.* 3D multi-surface is proper to import spalling geometry.

4. Tilted column and wall

   Wall surfaces and columns are already created in CityGML model by default. Therefore, the measured degree of inclination is introduced as a *Generic Attribute* for wall surfaces, columns and beam. The name of new attribute is Tilt.

5. Other information

   As the explanation is given in 2.2, damages on structural part are more important for BDA. Hence, a new *Generic Attributes* is added to wall surfaces (named Role) to distinguish damaged structural wall from non-structural. The data type of the attribute is set to Boolean. In the case of the structural wall, the value of attribute is 'True'. It allows to consider damages on structural wall as a higher level of damage.

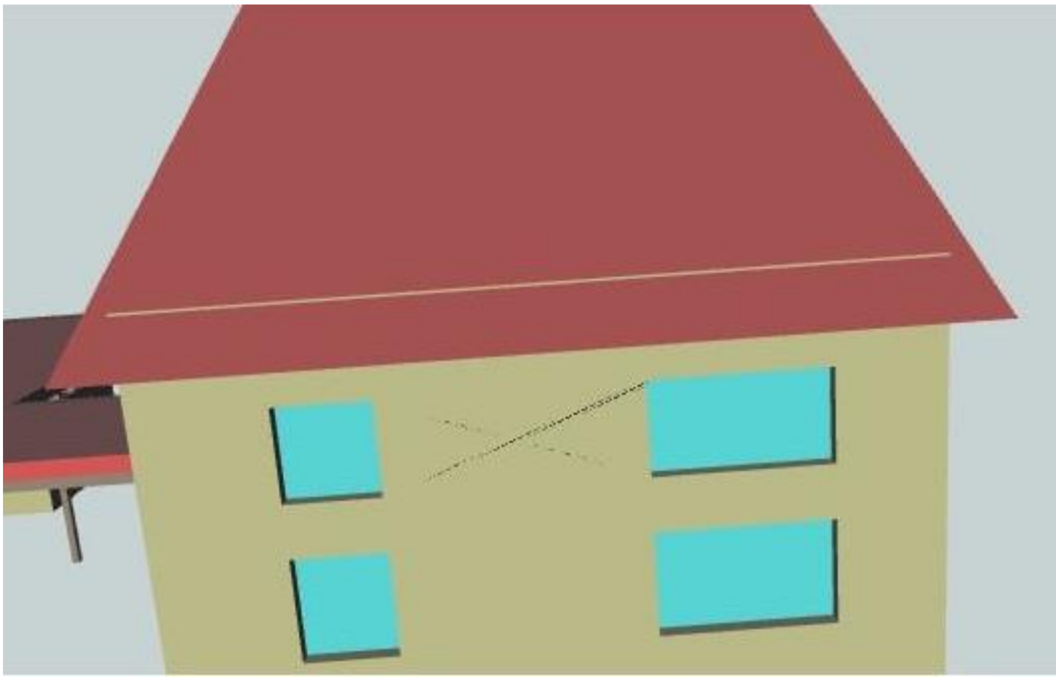


Figure 12: Two diagonal cracks between windows are modeled in CityGML.

3.2. Storing 3D model data into database

In order to start assessment and contribute spatial-semantic information of modeled objects, it is required to import 3D model of damaged building to the database. The database and the transformation tool are two significant elements to store and retrieve data accurately. Appendix 2 displays 3D model inside the database, the process of transferring using FME and an example of testing consistency of database.

3.2.1. Database

Detailed building elements and damaged evidence are modeled in an OGC standard 3D model. It is important that database can store spatial information correctly. Also, it should have the capability to get queries based on their geographic coordinates. In the case of three-dimensional space, the reliability of the result of spatial queries is a controversial issue. In comparison with 2D, numbers of the functions for 3D processing is limited. All desire geometric queries are not available also some new functions have not comprehensively tested. Therefore, the most proper database for not only storing semantic and geometric damaged building information but also for handling 3D spatial queries is chosen.

PostGIS is an open source and OGC complaint spatial database extender for PostgreSQL object-relational database. It supports the storage of geographic objects and spatial query in SQL. PostGIS 2.2.0 was released in September 2015 packaged with PostgreSQL Database Management System and it contains a wrapper library named SFCGAL (Borne, Mercier, Mora, & Courtin, n.d.). This library supports all 3D geometry types of objects inside CityGML model e.g. polyhedral-Surfaces. Also, all required 3D operations for reasoning and assess the damaged building have been tested such as 3D intersection function, 3D area, and distance computation. The results prove that PostGIS is a reliable dataset for processing valid 3D models based on OGC standards.

3.2.2. Transformation tool

3D data transformation should meet a set of criteria to become a proper tool to transfer 3D model into the database in this application. It should be able to:

- Restructure three-dimensional coordinate of objects of the model inside the database as geometry attribute.
- Reproject coordinate system carefully.
- Preserve new objects and attributes along with other defined 3D model elements.

The following methods are tested and analyzed to find the most suitable approach. The weakness and strength of their performance in BDA application are explained below:

Convert Gml format to Shapefile:

The most frequent format to import and store spatial data inside PostGIS is shapefile. However, in the case of the 3D model, it was not successful and could not keep the entire dataset as a three-dimensional modele most common format to import and store spatial data inside PostGIS is shapefile. But in the case of 3D model it was not successful and could not keep the entire dataset as three-dimensional model.

3D City Database:

3DCityDB is a free tool to store, manage, represent 3D city model on top of spatial relational database systems (Kolbe, n.d.). The schema of 3DCityDB is based on CityGML model, and it categorize all information inside defined classes. It creates several empty classes in the database. Also new Generic objects and Attribute are classified inside database very convoluted which adds more complexity to run spatial queries.

FME:

FME has the ability to transfer data in GML format to PostGIS database effectively; it is a powerful transformation tool to preserve the integrity of the original dataset during translation. FME restructures CityGML model in an adequate number of classes and keeps the 3D geometry of information. In this project, FME is chosen as transformation tool because not only it met all required criteria but also it allows to manipulate type of attributes and replace specific table inside database without overwriting and avoids the possible errors.

3.3. Designing fuzzy expert system

As mentioned before in 2.3, to apply reasoning inside the 3D model of damaged building fuzzy expert system is the most proper method to encompass the linguistic uncertainty of building damage assessment. The first step to design a fuzzy expert system of BDA is defining fuzzy membership classification of damages and fuzzy rules. The classes and rules provide the opportunity not only for remote sensing mappers to interpret the damages on the building but also for structural engineers and specialists to evaluate the damaged building according to RS data. The fuzzy expert system is implemented in Python programming language. The main reasons to adopt Python are the following:

- It contains SciKit-Fuzzy package which is a collection of fuzzy logic algorithms.
- It is possible to combine two fuzzy systems to calculate the final level of damage.
- It provides a direct connection to PostgreSQL database to run the spatial queries from the code. It also allows repeating queries multiple times.

The defined BDA framework consists of two fuzzy expert system to achieve a general building evaluation beside façade damage assessment. Calculating levels of damages and damage grade for each façade provides detailed evaluation for specialists. Also in the case of occlusion and incomplete 3D model, it provides damage assessment of visible parts of the building. The second fuzzy expert system assesses entire affected building and generates the final damage grade by considering damages of facades and roof.

The first fuzzy system is developed to assess each façade individually. It contains a loop over all facades to run spatial and non-spatial queries inside the database to detect possible damages related to one façade. Then through defuzzification, it generates a crisp value as the level of damage of facade. It repeats the same process for all facades of the building. The second fuzzy expert system calculates the general level of damage of building. The generated damage values of facades in the first system are used as input in the second fuzzy expert system. Also, the detected damage of the roof is retrieved from the database to incorporate in general assessment. The specifications and details of both systems are described below. Also, the generated fuzzy expert system in Python programming language is illustrated in Appendix 7.

3.3.1. First fuzzy expert system

It is designed to evaluate the level of damage for each façade of structure separately. All detectible types of damages related to façade are contributing in this phase. It provides the opportunity to analyze the different aspects of building individually. Therefore, occluded façade could not influence all the process and still it is possible to analyze the observable parts. The range of universe of discourse of the defined system is from 0 to 100 to represent level of damage and in percentage. Hence, various types of damages related to façade contribute as fuzzy elements. The fuzzy membership classes of each type of damage are generated in mentioned range. The fuzzy classification is based on the damage properties and consulting with experts. Fuzzy rules are generated to classify damage of façade in five levels. To import damage information into the system there are two options. More information about fuzzy membership classification and rules is explained in 3.3.3 and 3.3.4 respectively. The first approach is to retrieve data directly from the database through run queries; the second option is importing data from a file that contains the damage evidence location and attributes. The second option makes the fuzzy expert system independent from previous phases, therefore in the case of problem in database connection or using different 3D data model it can be still used to assess the level of damage of the building. After importing data, degrees of membership of damages of one façade are calculated. According to the result of fuzzification, the fuzzy rules compute the membership degree of façade in five level of damage. The last step is defuzzification. It aggregates all calculated level of damage and finds the center of gravity of aggregation. The calculated crisp value is the percentage of damage (damage grade) of one façade. After generating the same process for all facades second fuzzy expert system starts the overall assessment. Figure 13 illustrates the steps of damage assessment of each façade separately in first fuzzy expert system.

3.3.2. Second fuzzy expert system

The second fuzzy system is designed to analyze damage of building as one integrity and calculate the percentage of overall damage of structure. The evaluation of facades and roof inside fuzzy environment avoids crisp classification and gross errors. The input of this phase is the final value of damage of façades from the first system beside information of roof surfaces. In the second fuzzy system there are five fuzzy rules and three fuzzy membership functions. The first function is to calculate the crisp damage value of each façade in five level of damage and the second function is to compute the degree of membership of damages in the roof. The rules use the output of first and second functions. The results of rules are imported inside the third fuzzy function. The third function has five classes and shows the general level of damage of building. Defuzzification algorithm extracts the crisp damage value to from aggregated classes of the third function. It shows the general condition of the building with considering all visible types of the damages around the structure. The flowchart of the second fuzzy system process is demonstrated in Figure 14.

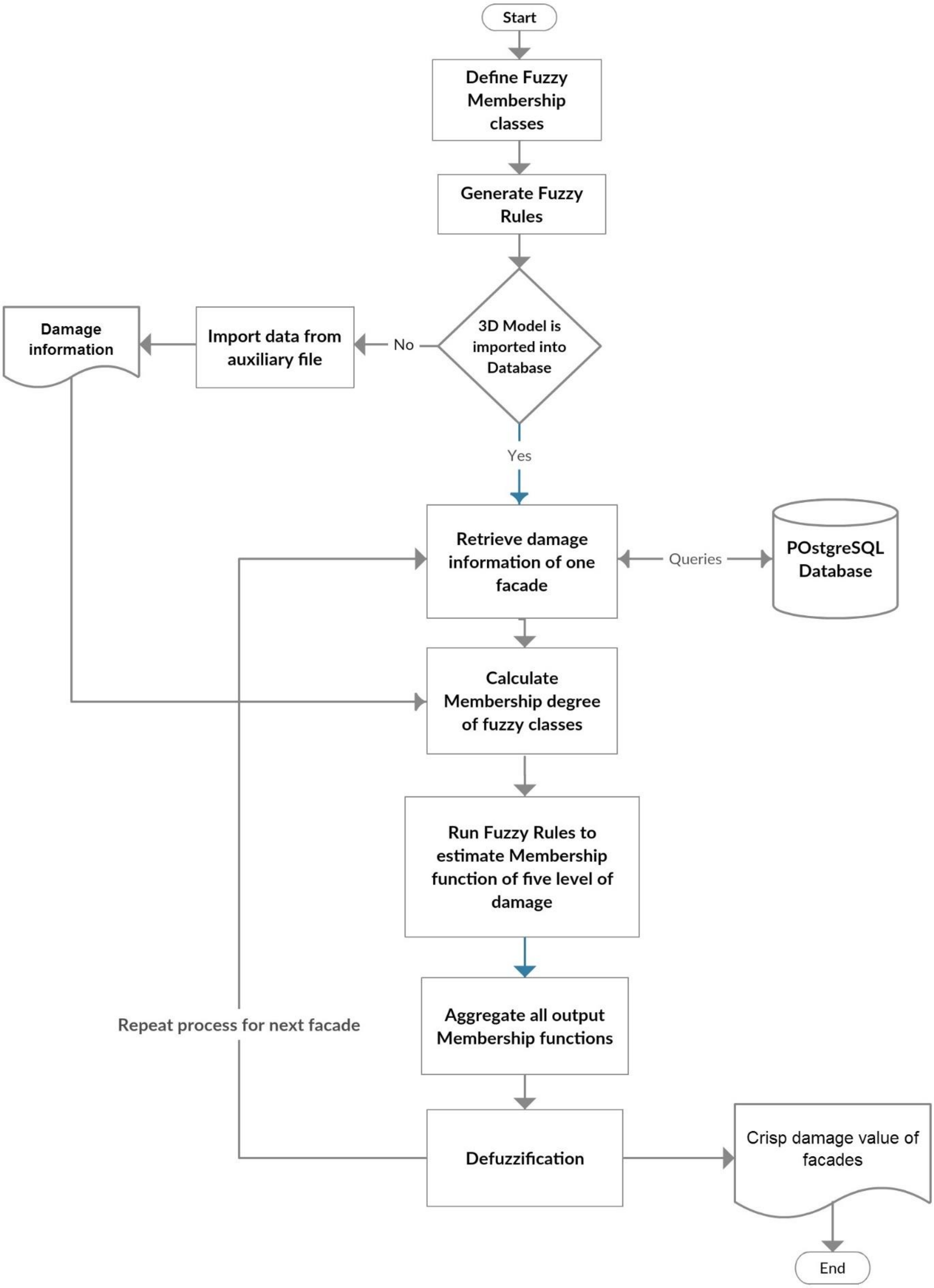


Figure 13: Flowchart of first fuzzy expert system.

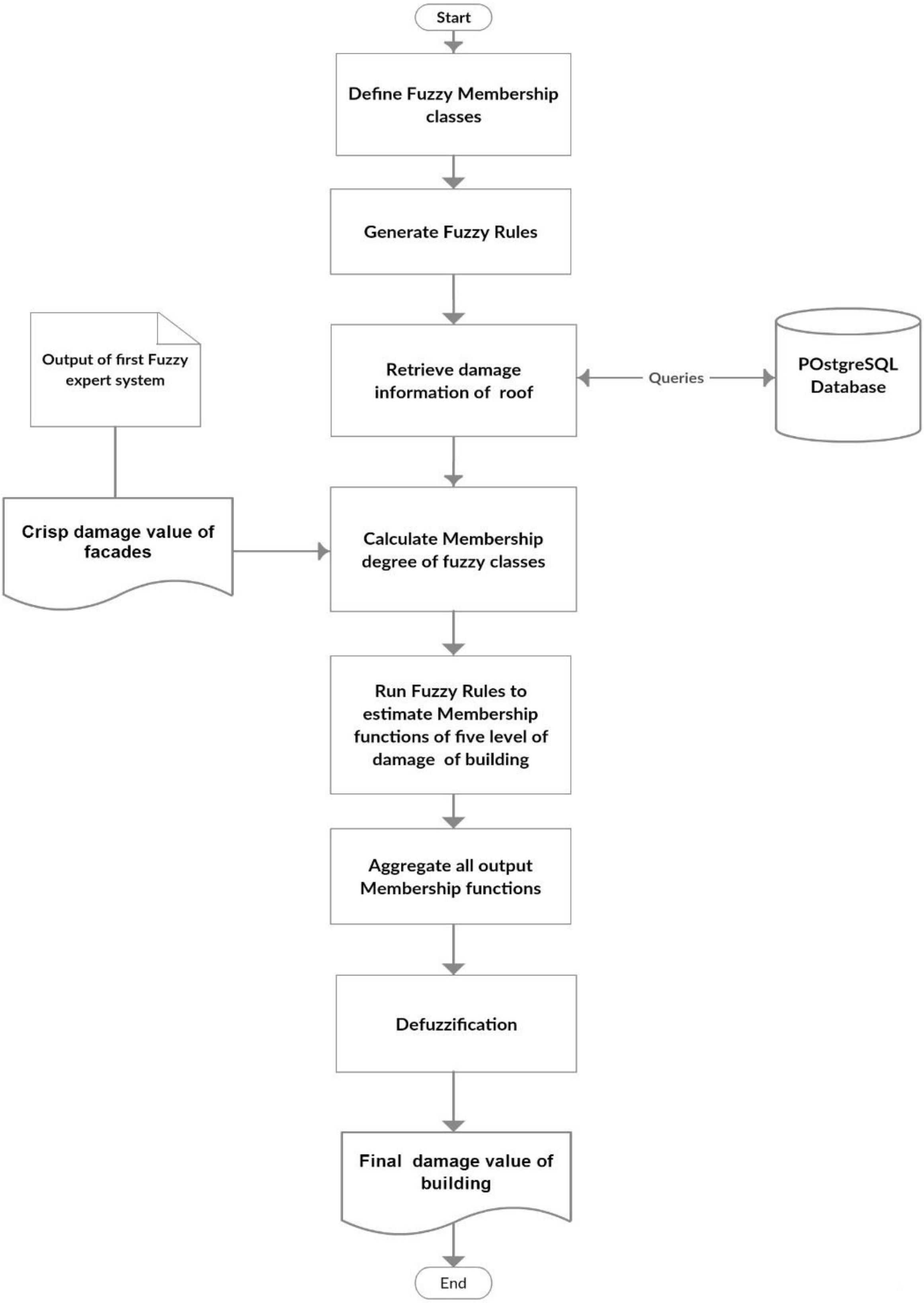


Figure 14: Flowchart of second fuzzy expert system.

3.3.3. Fuzzy membership classification

This study attempts to involve useful RS damage evidence to obtain fast and accurate evaluation. There is a lack of proper damage scale for BDA to involve and interpret different type of RS damage evidence. Therefore, during this research visible damage evidence from RS data is classified by considering their properties, semantic and geometric attributes. The concept of the classification is based on damage scales, questionnaire and consulting with structural engineers and its structure could be adopted and improved in future studies.

Type of the building (masonry, reinforced concrete, steel structures and timber structures) is a critical factor in process of defining boundary of fuzzy classes and set the assessment rules. The rules and classes of damage evidence are generated by considering two types of building, reinforced concrete and masonry building. It leads to obtaining a reliable assessment and avoid a gross mistake. It should be mentioned that other types of building require modification to the classification and rules of the system.

Fuzzy objects consist of building elements which are categorized according to the RS damage evidence and structural knowledge. Fuzzy objects are normalized between 0 and 100 and classified based on the level of severity (Low, Moderate, and High). Figure 15 illustrates classification factors and fuzzy classes of objects inside the designed system. For example, the area of roof failure is categorized in three classes, but a crack in the column is introduced into the system in two groups, Low and High. Each object has an exclusive category, for instance, the Low roof failure is different from Low structural wall failure. Because of better visualization and avoid repetition in Figure 15 fuzzy classes of various objects are demonstrated in the same box. Each class contributes in related fuzzy rule to assess the damage of façade and building. The details of fuzzy membership classification are explained as follows.

1. Failure:
   - Structural wall failure
   - Non-structural wall failure
   - Roof failure
2. Spalling:
   - Spalling on wall
   - Spalling on column
   - Spalling on beam
3. Tilt:
   - Tilted structural wall
   - Tilted non-structural wall
   - Tilted column
4. Crack:
   - Crack in column
   - Crack in beam
   - Crack at Level 1 on structural wall
   - Crack at Level 2 on structural wall
   - Crack at Level 3 on structural wall
   - Crack at Level 1 on non-structural wall
   - Crack at Level 2 on non-structural wall
   - Crack at Level 3 on non-structural wall
5. Damage close to structural building elements.

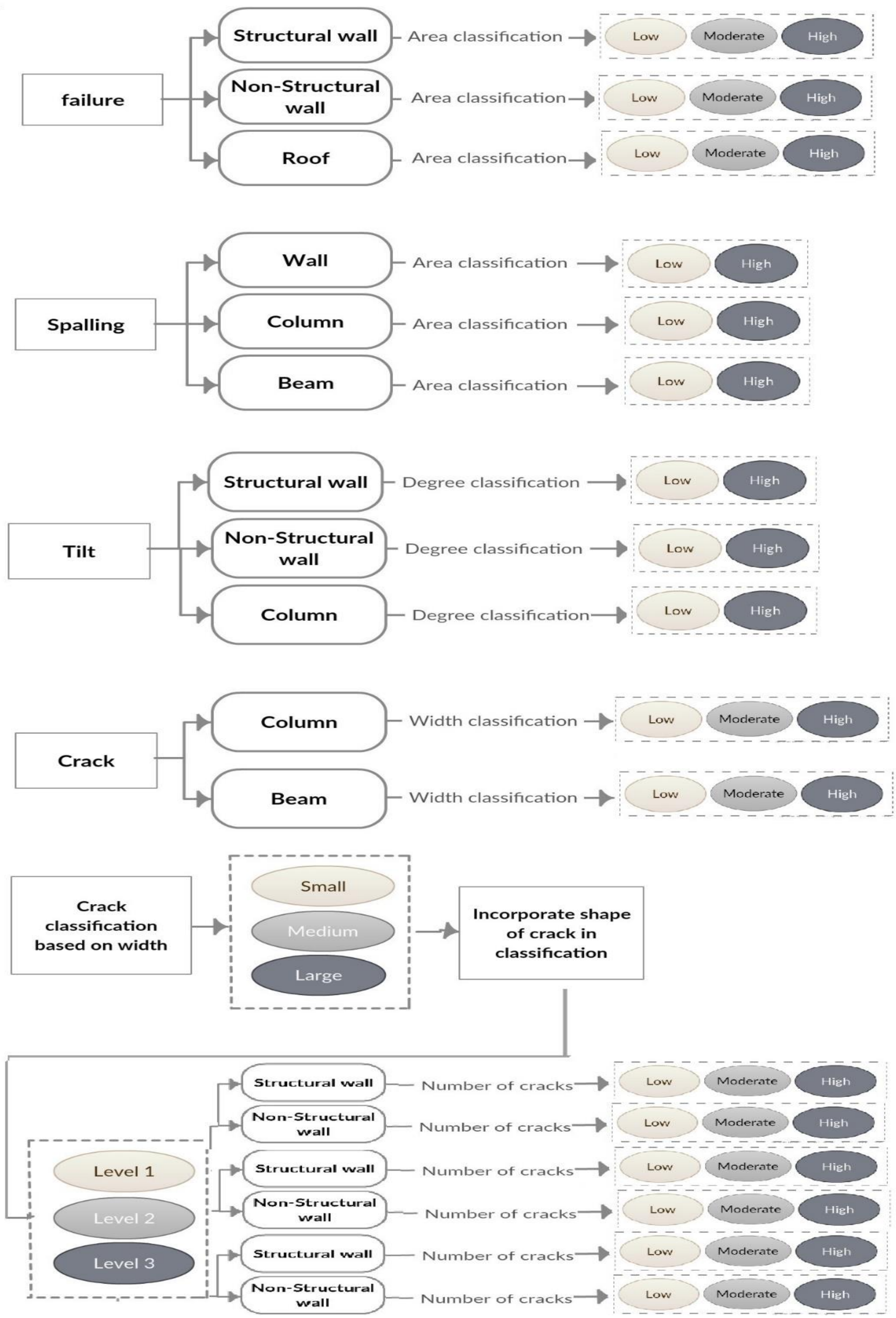


Figure 15: Fuzzy membership classification of RS building damage evidence. The light gray, dark gray and dark blue represent Low, Moderate and High classes respectively.

3.3.3.1. Structural, non-structural wall and roof failure

Multi-storey building or some cases based on its architectural design, one façade can contain both structural and non-structural wall. Also, it is possible to observe wall failure in different parts of a façade. The percentage of total area of the non-structural wall is classified in the fuzzy system to have a complete façade assessment. Figure 16 illustrates the fuzzy membership function of the percentage of failure at the non-structural wall. The red line in the graph represents a low level of damage in trapezoidal shape which 1% to 10% have the highest degree of membership (1) and 20% has the lowest degree (0). The value between 10% and 20% has degree less than 1 and more than 0. In other words, if in general only 5% of the non-structural wall are failed, the membership degree of the faced to Low class is 1. Also, the degree of fuzzy membership of a facade with 15% failure is 0.5 in Low class. The purple line shows triangular membership class of Moderate level of damage. The range of this class is from 10% to 40% with the peak at 20%. Trapezoid-shape of High level damage in blue colour illustrates that all value of failures more that 40% are belonged to this class with the highest degree of membership (one). Table 1 contains the summary of non-structural wall, structural wall and roof failure classification. The structural wall and roof failure also are defined with the same concept of percentage of total damage over the total area. In addition, two mentioned damage factors are classified into three classes (Low, Moderate and High damage) because the importance of failure is high and three classes can help to have a more detailed evaluation. In Figure 16 displays the details of non-structural failure classification. Low damage in the structural wall leads to serious damage in contrast with Low damage of non-structural wall. Each object has exclusive fuzzy classes.

Table 1: Structural, non-structural wall and roof failure fuzzy classification information

Type of damage	Low damage	Moderate damage	High damage
Non-structural wall failure	10%	20%	30%
Structural wall failure	10%	20%	30%
Roof failure	5%	10%	40%

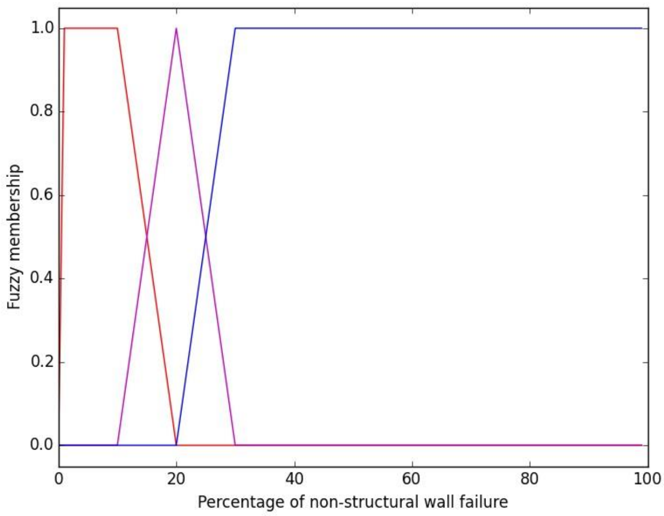


Figure 16: Fuzzy membership function of non-structural wall failure

3.3.3.2. Spalling on wall surface, column and beam

In the case of assessing a level of damage of the spalling on the wall surface, column and beam two classes can provide sufficient detail inside the fuzzy expert system. Accordingly the percentage of total area of spalling over the calculated area of the object is categorized in two trapezoidal classes (Low and high damage). Table 2 contains the specifications of fuzzy classes. The fuzzy membership function of damage classification caused by spalling on wall surfaces is illustrated in Figure 17. The trapezoid in red colour indicates less than 10% spalling of the total area of the façade belongs to Low level of damage and its membership value is 1, but from 10% to 40% the membership degree is decreasing to 0. Also, the degree from 10% to 40% of spalling is increasing inside the class of High level of damage, and all spalling damages more than 40% completely belong to this class.

Table 2: Spalling on wall surface, column and beam fuzzy classification information.

Type of damage	Low damage	High damage
Spalling on wall surface	10%	40%
Spalling on column	10%	20%
Spalling on beam	5%	10%

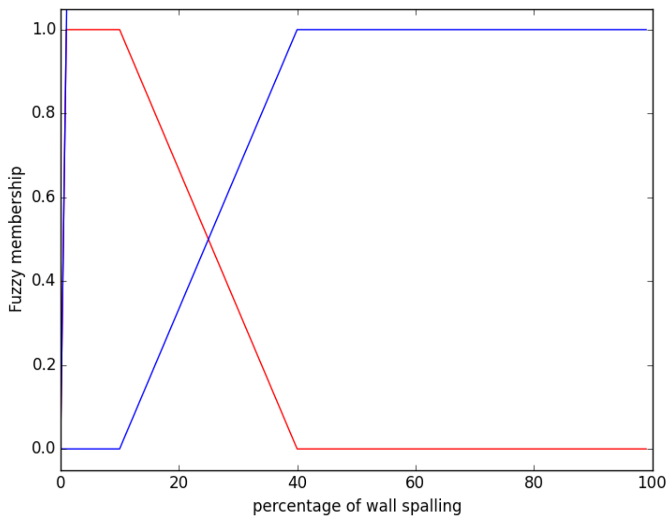


Figure 17: Fuzzy classification of wall spalling according to the percentage.

3.3.3.3. Tilted column, structural and non-structural wall

In the case of tilted building element there are two options:

- The building object is still safe of falling despite detecting a degree of inclination from a 3D model.
- Even though the wall surface or column is standing but the probability of falling is very high.

RS damage detection is difficult to detect small tilt caused by an earthquake. Therefore, the damage evidence of tilting is classified in two groups Low and high with trapezoid-shape. This classification is based on the detected degree of inclination. The information about the membership function classification in the fuzzy system can be found in Table 3. Also, the graph of membership classification is represented in Figure 18.

Table 3: Tilted column, structural and non-structural wall fuzzy membership classification information.

Type of damage	Low damage (degree)	High damage (degree)
Tilted non-structural wall	10	20
Tilted structural wall	10	20
Tilted column	10	20

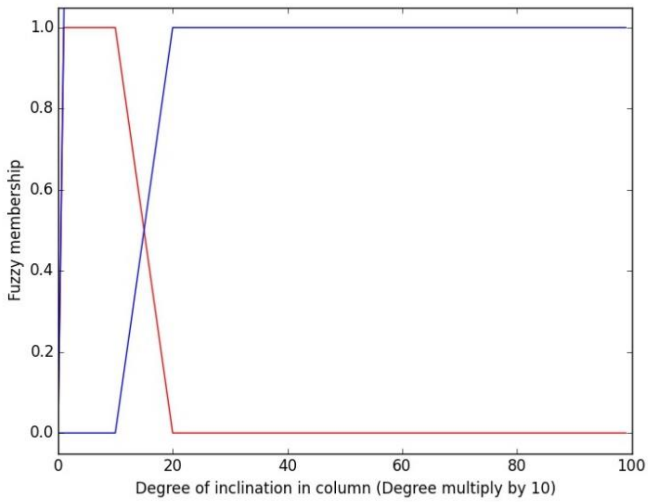


Figure 18: Fuzzy menbership of tilted column.

3.3.3.4. Crack in column and beam

After an earthquake crack is imperative evidence to evaluate the level of damage of a column or beam. The width of crack is the key that can show the severity of damage to columns and beams caused by an earthquake. Therefore, the cracks in mentioned structural elements are classified into three classes (Low, moderate and high damage) based on the largest detected width of the object. The value of membership function in the trapezoid and triangular shape are represented in Table 4. To implement fuzzification we need to have a uniform range for all membership function; hence the width of extracted cracks which are modeled in meter in 3D CityGML model, are multiplied by 10 to be included in the range of 0 to 100.

Table 4: Crack in column and beam fuzzy membership classification information.

Type of damage	Low damage	Moderate damage	High damage
Crack in column	20	50	70
Crack in beam	20	50	70

3.3.3.5. Crack at level 1, 2 and 3 in structural and non-structural wall surfaces

The width, shape and location of the crack are the factors that should be considered to decide on the level of severity of damage to wall surfaces. The combination of these three factors can lead to different results. Firstly, crack is classified based on their width to integrate them accurately. According to the references: crack with the width less than 3 mm is classified as Small, if the width of crack is more than 3 mm but less than 6 mm it is grouped as Medium level and crack which has the width more than 6 mm is in Large level class (Table 5). Secondly, the combination of each three classes and shape of the cracks result in three level of damage. There are three main shapes of cracks: diagonal, horizontal and vertical. The connected cracks are added as the fourth shape type and named intersected cracks. The intersected cracks are important for BDA and can show the severe level of damage, and they indicate damage accumulation. Therefore, it is possible to involve the spatial relationship of damage features in the process of assessing. Table 6represents the importance of each crack according to their shape and width. It is clear that the location of each level of crack (based on Table 6) plays a different role in BDA. Hence, building elements are analysed separately in generated system. For example, a diagonal crack which its width is less than 3 mm is classified in Level 2, or intersected cracks which the maximum width of them is more than 6 mm are classified in Level 3 of damage.

Table 5:Classification of cracks based on their width

Level of severity of crack	Width (mm)
Small	Width =< 3
Medium	3 < Width < 6
Large	Width >= 6

Table 6: Classification of level of damage of cracks based on combination of their shape and width

Level of damage of crack	Diagonal	Intersected	Horizontal	Vertical
Level 1			Low	Low
Level 2	Low	Low	Moderate	Moderate &High
Level 3	Moderate &High	Moderate &High	High	

Besides width, shape and location, the number of cracks on a façade also is important to judge the condition of that façade, for example detecting six cracks on a façade should be more significant than a façade that contains only one crack. Consequently, in the fuzzy system each level of crack is classified based on the number of detected crack on a façade (structural and non-structural wall separately). It comprises three classes to represent low, moderate and high level of damage. Figure 19 demonstrates the fuzzy membership classification of Level 3 of crack on a non-structural wall. Since the number of cracks should be normalized between 0 and 100 to participate in fuzzification, the measured number is multiplied by 10. The generated classes are the same for all three level of cracks in both structural and non-structural wall surfaces because they follow the same criteria. The information of fuzzy classification of crack on the wall surface is represented in Table 7.

Table 7: Crack in wall surface fuzzy membership classification information.

Type of damage	**Low damage** (number of crack in one façade multiply by 10)	**Moderate damage** (number of crack in one façade multiply by 10)	**High damage** (number of crack in one façade multiply by 10)
Crack in wall surface	20	40	60

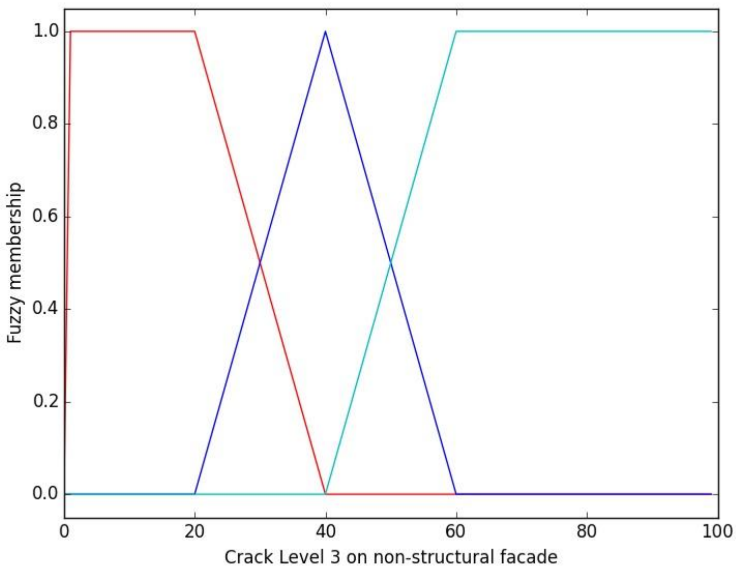


Figure 19: Fuzzy membership function of crack at Level 3 on non-structural wall.

3.3.3.6. Detecting damages based on spatial-semantic relationships

In practice, there are some damaged parts that do not have any sign of damage on their surfaces. These affected elements are surrounded by damage evidence without any connection. For example, it is possible that a crack in 5 cm of a structural wall has penetrated inside that, but this damage is not observable from RS data. Another sample could be failure close to a column that it could put the column in an unstable situation. Structural elements located in a specific distance of damage evidence are involved in the fuzzy expert system to have a spatial-semantic building assessment. In this research number of detected structural elements within 0.1 meters (m) distance in 3D space from all type of damages are classified in two trapezoid-shape level of damage (Low and High). Table 8 contains the information of the classification. Façade which contains less than 10 structural parts close to damage categorized in Low level with membership degree equal 1, and façade with more than 15 possible damaged elements are in High level with highest membership degree.

Table 8: fuzzy classification of spatial-semantic damage assessment.

Type of damage	Low damage	High damage
Number of structural elements clos to damages	10	20

3.3.3.7. Level of damage classification

The process of fuzzification requires a fuzzy classification for the output of the defined rules. In this research, the output classes are named "Level of damage". The output of each fuzzy rule refers to one level of damage. These levels are fuzzy classes normalized between 0 and 100 similar to previous fuzzy classes. The first expert system calculates the results of five rules by considering extracted damages of facades, for example, the output of the first rule shows the degree of membership of façade in Level one. Levels represent the percentage of damage. Figure 20 demonstrates fuzzy classes of the level of damage in triangle and trapezoid. The second fuzzy expert system has the same level of damage classification. The description of damage levels are as follows:

- Level one: no structural damage and slight non-structural damage (rule 1).
- Level two: slight structural damage and moderate non-structural damage (rule 2).
- Level three: moderate structural damage and heavy non-structural damage (rule 3).
- Level four: heavy structural damage and very heavy structural damage (rule 4).
- Level five: very heavy structural damage (rule 5).

The higher levels of damage have a more severe effect on the building. In other words, the role of the higher levels is more significant than the lower classes in damage assessment. The structure of the fuzzy expert system does not let to add a new factor to give different weight to levels of damage. To overcome this limitation, the boundary of the damage levels in fuzzy classification are not equal. Therefore, the boundary of levels in classification is increases based on the influence and importance of damage level in BDA. The centre of the gravity of aggregated levels is the final damage grade. Consequently, the Level five involves more weight to the aggregated levels by default and affects the location of the centre of the gravity. Table 9 represents the peak of trapezoidal and triangular shapes of damage levels.

Table 9: Level of damage fuzzy classification.

Level of damage	Level one	Level two	Level three	Level four	Level five
Percentage of damage	10%	20%	35%	50%	75%

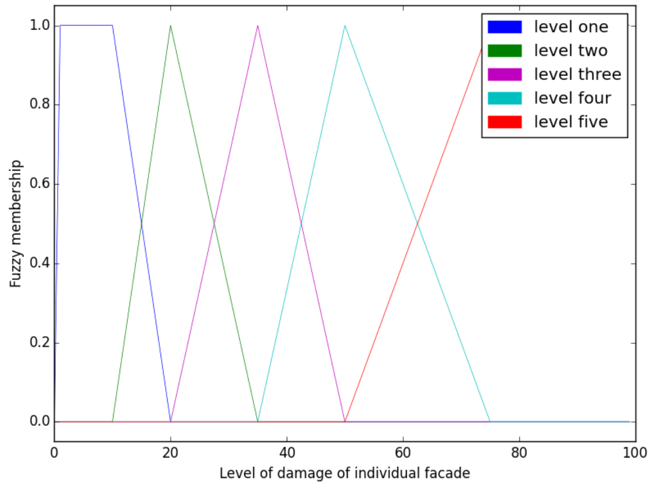


Figure 20: Fuzzy membership classification of level of damage

3.3.4. Define fuzzy rules based on RS data

After determining fuzzy membership functions of required and available damage evidence based on their spatial, semantic and topology information, fuzzy rules are designed to combine fuzzy classes and define a correct level of damage of façade and building. Based on references each fuzzy membership class belongs to the specific level of damage. There are two set of rules, first set is used to evaluate the level of damage for each façade separately and the second one generates a general level of damage of a building by considering results of the first set and fuzzy membership function of the roof. Both sets of rules use Mamdani technique to generate the result of fuzzy classes. Figure 21 illustrates fuzzy classes of first fuzzy expert system rules. The components of defined rules are as follows:

Rules of first fuzzy system:

Rule 1: Low non-structural damage. The result of Rule 1 is classified as damage Level one of facade.

Rule 2: Moderate non-structural damage Or Low structural damage. The result of Rule 2 is classified as damage Level two of façade.

Rule 3: High non-structural damage OR Moderate structural damage. The result of Rule 3 is classified as damage Level three of facade.

Rule 4: High structural damage. The result of Rule 4 is classified as damage Level four of facade.

Rule 5: Very High structural damage. The result of Rule 5 is classified as damage Level five of façade.

Rules of second fuzzy system:

Rule 1: Damaged façade at Level one.

Rule 2: Damaged façade at Level two

Rule 3: Damaged façade at Level three OR roof failure at Low level.

Rule 4: Damaged façade at Level four OR roof failure at Moderate level.

Rule 5: Damaged façade at Level five OR roof failure at High level.

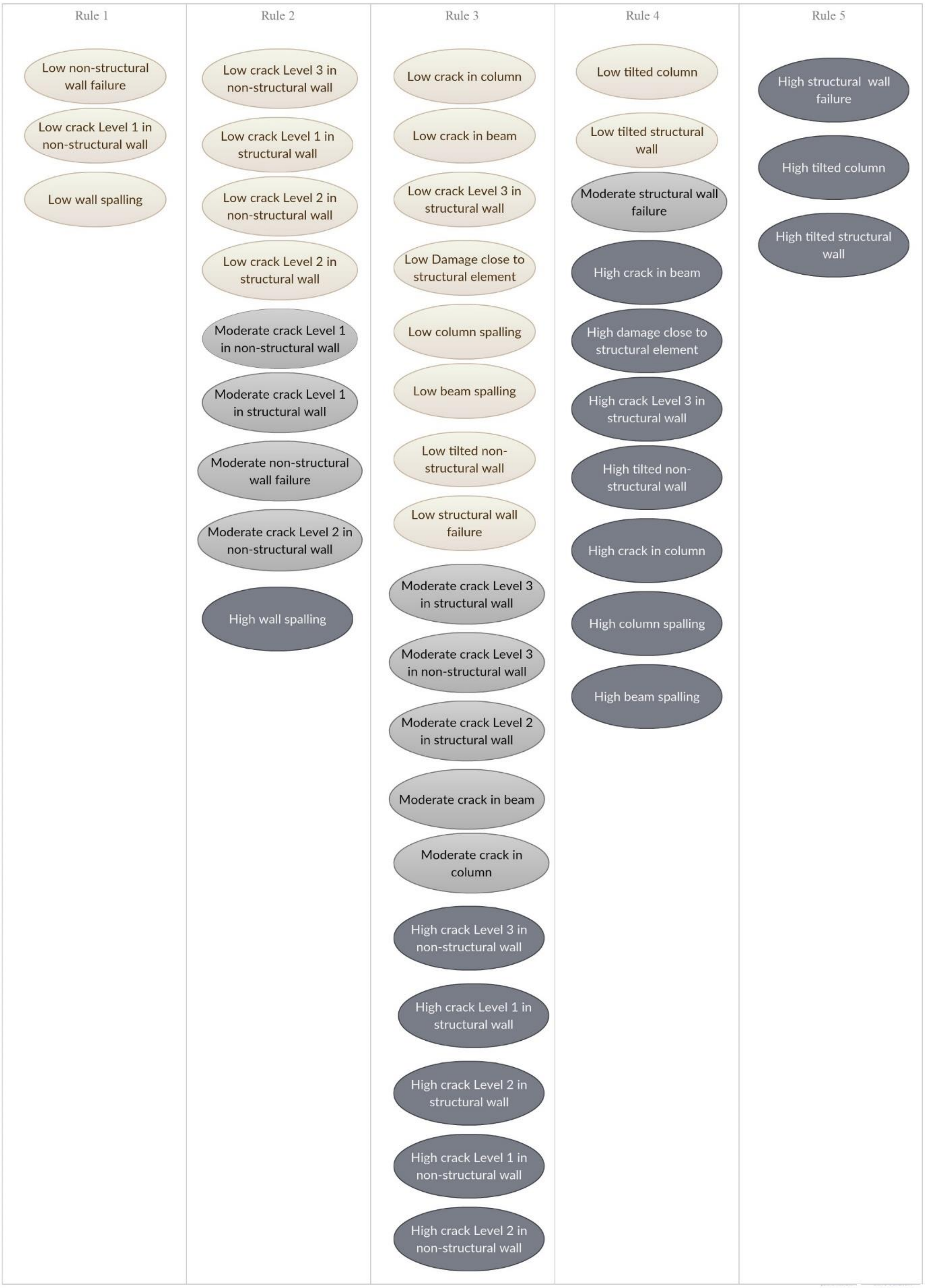


Figure 21: Components of first fuzzy expert system rules. The generated rules apply fuzzy classes to compute level of damage of facades. The light gray, dark gray and dark blue represent Low, Moderate and High level of damage respectively.

4. RESULTS AND DISCUSSION

The feasibility of performance of the CityGML, fuzzy expert systems and consistency of database to represent, store and assess damaged building based on remote sensing data are studied in this research. Appendix 3 illustrates GML code of extended CityGML to model damage evidence and attributes. In addition, it includes the results of consistency test of PostGIS to store, process and retrieve data based on three-dimensional geometry. This chapter presents the results and discussion of the technical feasibility evaluation of the proposed framework by adopting three simulated damaged structures.

The samples contain masonry and reinforced concrete building in different levels of damage. It is attempted to figure out strengths and weaknesses of the framework. The proposed fuzzy expert system runs the queries to assess and calculates the final percentage of damage of each façade individually. Then the final level of damage of building is calculated by aggregating the final damage percentage of the façades and damages of the roof. The simulated samples information and results of damage assessment are discussed as follow.

4.1. Simulated sample 1

Sample 1 is a simulation of damaged reinforced concrete building. Figure 22 shows the 3D visualization of the building and damages. The 3D model represents a three-storey structure. It is composed of columns, beams, structural and non-structural walls. The structural walls are modeled in dark gray color. The details of damage evidence are listed in below. There is no damage to the roof and structural elements (column, beam, and structural wall) and damages in the non-structural wall are at Low and Moderate level. According to the designed rules and expert knowledge the final grade of damage of facades and building should not exceed Level 2.

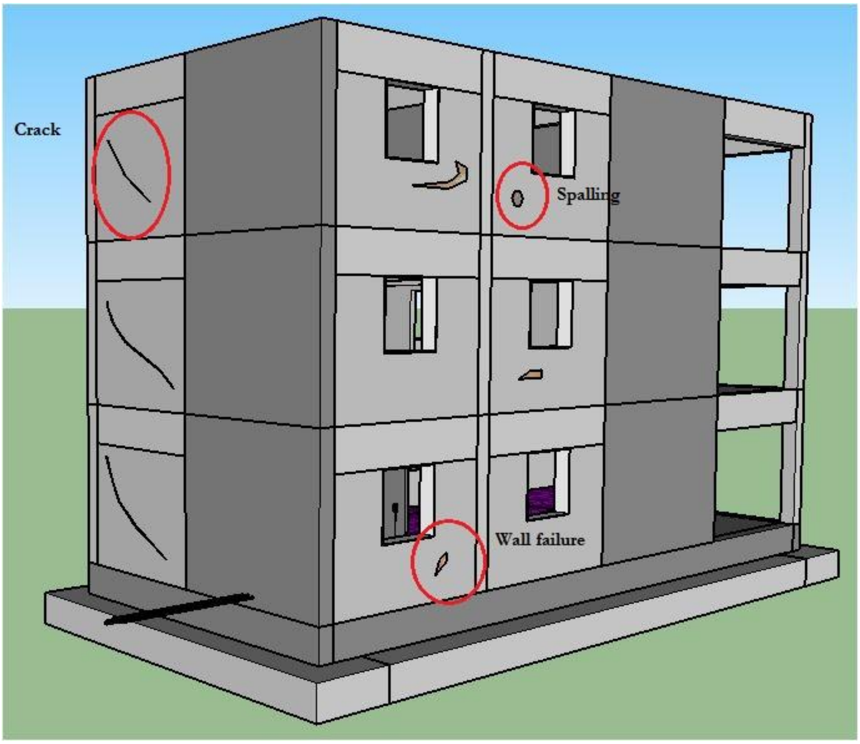


Figure 22: 3D model of sample 1 represents the damage evidence in non-structural building elements.

Damage description of sample 1:

North Façade:

- 5% of the non-structural walls are failed.
- 6% of walls are spalled.

East façade:

- 3 diagonal cracks in non-structural wall (width of crack: 0.002 m).

South façade:

- 9% of the non-structural walls are failed.

West façade:

- No damage.

Roof:

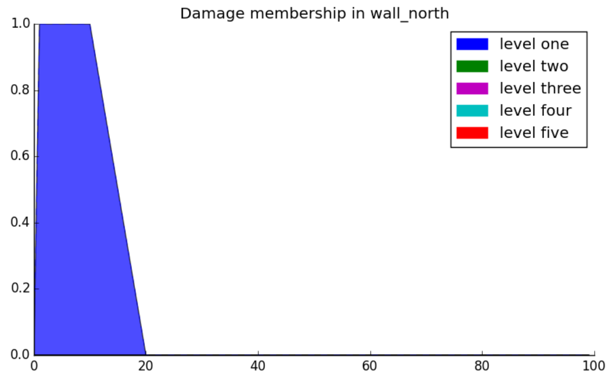
- No damage.

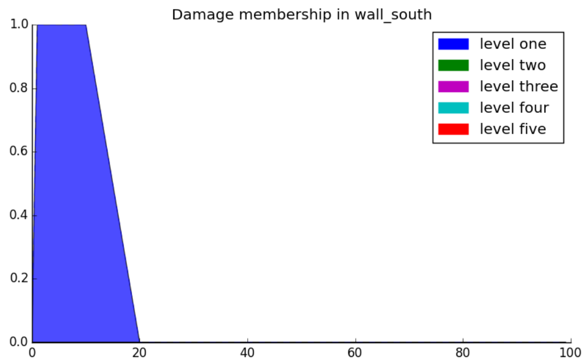
The results of first and second fuzzy expert systems of the façades and building assessment are explained in the following figures and table. Figure 23 shows the results of the evaluation of the facades individually; the horizontal axes represent the percentage of damage, and vertical axes illustrate the degree of membership. As mentioned in 3.3.3 the percentage of damages are classified into five levels; Level one refers to the slight damage and Level five belongs to very heavy damage. According to the detected damages from specific façade degrees of membership of the five levels are calculated. The levels which have the membership degree more than zero are illustrated in the graphs.

According to the damage description, non-structural walls belong to the north façade and south façade contain less than 10% failure and in Figure 23 they are classified in Level one. East façade has 0.5 degrees of membership of Level two because of cracks in non-structural walls. The degree of membership of the west façade is zero in all damage levels due to no damage evidence. The final crisp damage grades of the facades and building are provided in Table 10. Less than 10 percent of north and south façade are damaged and east façade has about 20% damaged elements.

Figure 24 shows the building has damages in Level one, two and three in general. By considering the membership degrees, it is strongly affected by the Level one and Level two. The final damage grade of the building after aggregating all contributed fuzzy classes is 16 percent (Figure 24). Therefore, the simulated sample 1 can be classified in Level two because of moderate non-structural damage and an intact roof Appendix 4 represents the computed final damage grades of facades from aggregated membership degrees.

The results of the second fuzzy expert system (Figure 24) indicates that the framework can integrate intact and damaged facades. It successfully imitates the expert decision making in BDA and considers detailed damage evidence. Although general assessment of building reveals about 0.1 membership degree of Level three, the calculated overall damage value is 16% which is located in overlap boundary of Level one and Level two. Therefore, the small membership degree does not have a significant effect on the final damage grade that makes the system more constant and reliable. Also, the intact façade is not classified as Level one that prevents misclassification. The fuzzy membership degrees of damage classes gives a better understanding of the influence of detected damages on the building. Applying two separate fuzzy expert systems provides detailed and holistic evaluation.





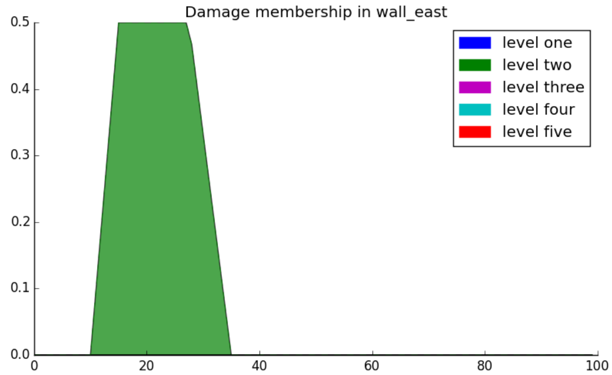


Figure 23: Facade damage assessment of sample 1. Each graph represents the level of damage and its degree of membership of specific facade.

.

Table 10: The final damage value of facades and overall percentage of damage of the building.

	North facade	East facade	South facade	West facade	Overall
Damage grade of Sample 1	8%	21%	8%	0	16%
Damage grade of Sample 2	29%	26%	30%	35%	30%
Damage grade of Sample 3	45%	35%	68%	41%	62%

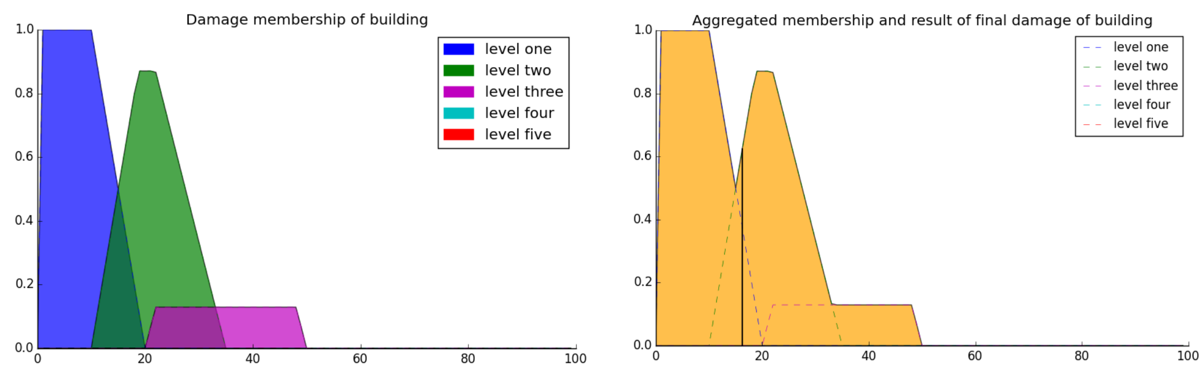


Figure 24: Left image represents calculated levels of damage of the building of sample 1 after overall assessment. Right image shows the aggregated damage levels, the overall damage value of the building is illustrated in black line.

4.2. Simulated sample 2

The second sample is a simulation of reinforced concrete building that includes damaged structural elements besides damaged non-structural parts. 3D model of the affected building with simulated damages is represented in Figure 25. Although there is no damage to the roof surfaces, some damage evidence is extracted from structural elements. For example, there are partially collapsed of the structural walls in third and second floors. Also, there are spalling and cracks in beams and columns due to severe shakes of the earthquake. More detailed information on damages of the structure is listed as follows.

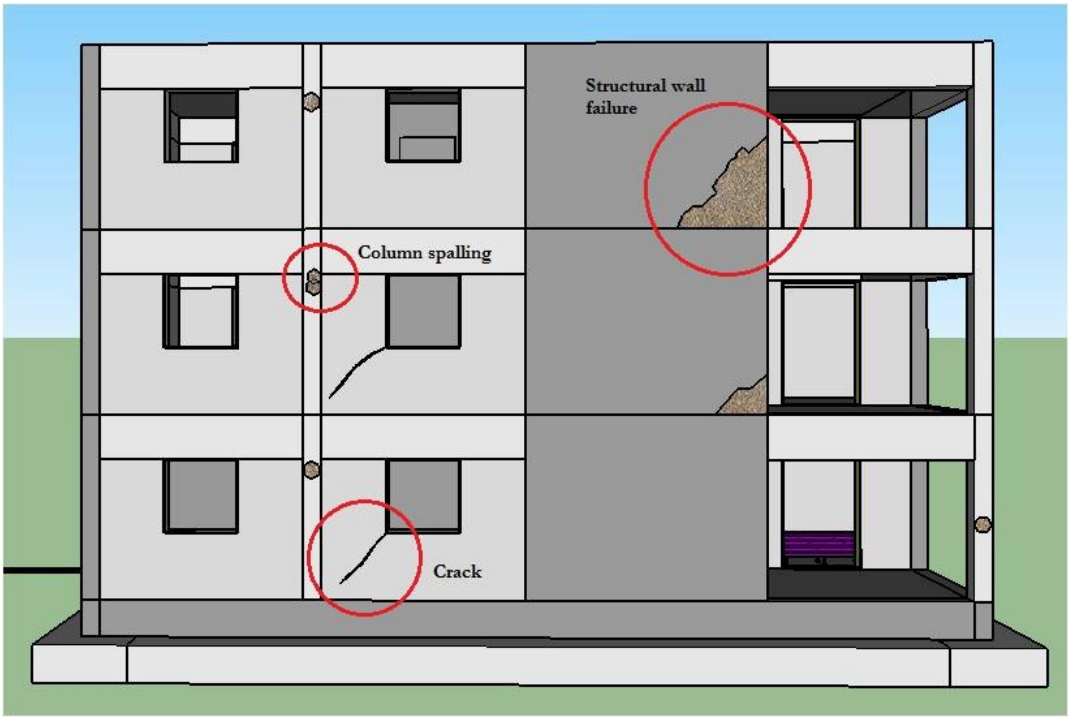


Figure 25: 3D model of sample 2 represents damage evidence in structural and non-structural elements.

Damage description of sample 2:

North façade:

- 5% of columns are spalled.
- 10% of structural walls are failed.
- 2 intersected cracks in non-structural walls (width of crack more than 0.006m).

East façade:

- 17% of the non-structural walls are failed.
- Column contains crack (width of crack less than 0.003m).
- Beam contains crack (width of crack less than 0.003 m).

South façade:

- 25% of the non-structural walls are failed.
- Beam contains crack (width of crack less than 0.006 m and more than 0.003m).

West façade:

- Column contains crack (width of cracks less than 0.002 m).
- Beam contains crack (width of crack less than 0.006 m and more than 0.003 m).

Roof:

- No damage.

The degrees of membership of facades in five damage levels are illustrated in Figure 26. The Level three is common to all façades, although some damages in the first and second levels are observed. For example, the east façade has 0.3 membership degree of Level one due to low non-structural failure. The system should have the ability to handle various levels of damage and generate a reasonable damage grade. The calculated final damage value in both fuzzy systems are represented in Table 10. Even though the Level one and Level two are observed in the east façade assessment, its highest membership degree belongs to the Level three because of the cracks in the columns and beams. The final grade of the east facade is 26 percent. This value indicates moderate structural damage that proves the correct performance of the BDA system in the case of various levels of damage. About 30% of the north façade is damaged (Table 10). As mentioned before, the classification of the damage levels gives more weight to the higher level. This value belongs to the Level three more than the Level two although the north façade has equal membership degree in the both damage levels. The graphs of aggregated damage levels and calculated damage value of facades are provided in Appendix 5.

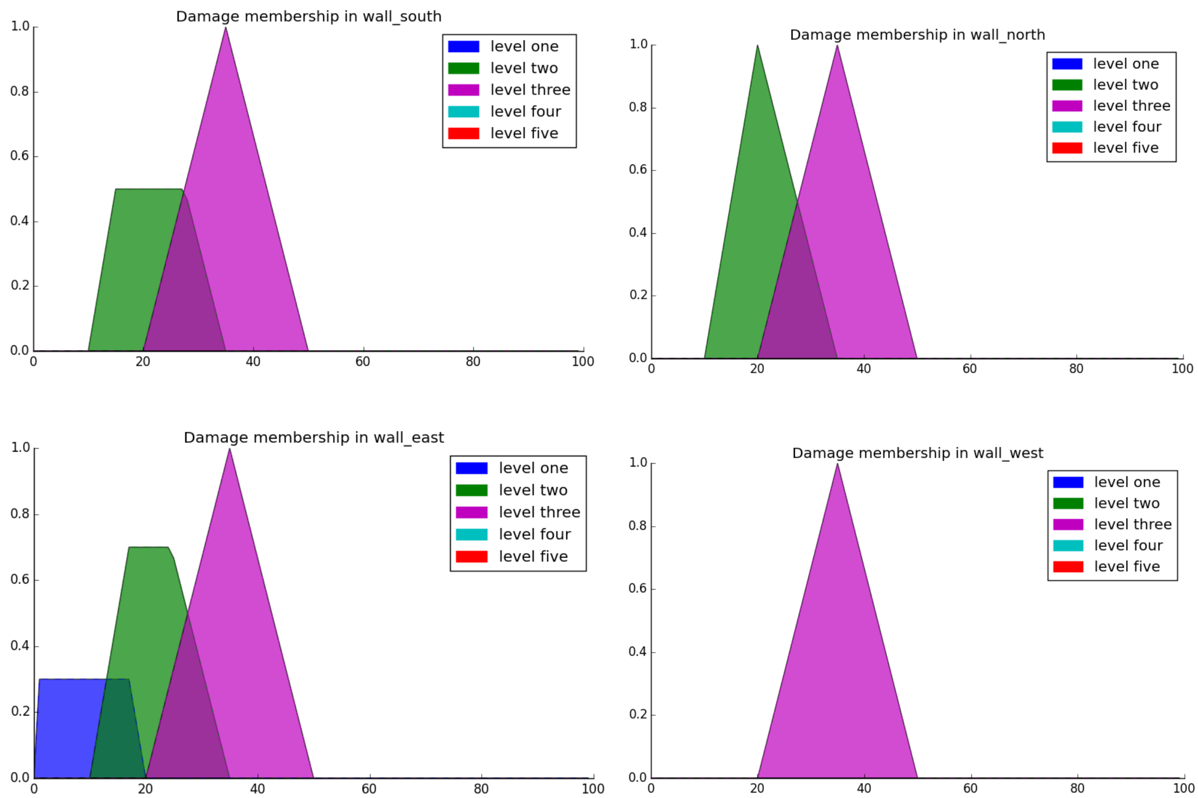


Figure 26: Facade damage assessment of sample 2. Each graph represents levels of damage and their membership degrees of specific facade

Figure 27 illustrates the results of the second fuzzy expert system. In the process of overall damage assessment, the affected building has around 0.5 and 1 degree of membership in Level two and Level three respectively. The simulated damaged structure belongs to Level three mostly with moderate structural damage and heavy non-structural damage. The small membership degree of the east façade in Level one does not have a significant effect on the second fuzzy system. The final damage score of facades calculated in the first fuzzy system are around 30 percent, and there is no damage to the roof surfaces. Hence the final calculated damage grade of the building is 30% (Table 10). The detailed and overall assessment is implemented correctly according to the references.

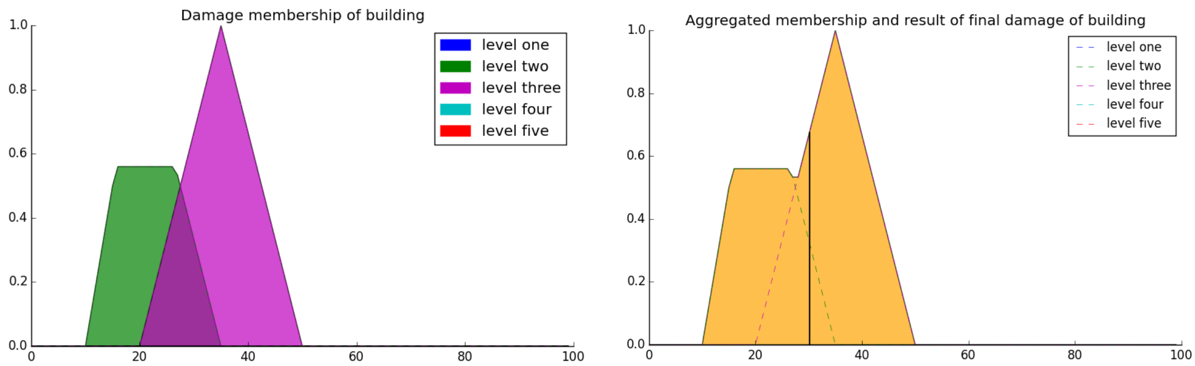


Figure 27: Left image represents calculated levels of damage of the building of sample 2 after overall assessment. Right image shows the aggregated damage levels, the overall damage value of the building is illustrated in black line.

4.3. Simulated sample 3

The third sample is a simulated masonry building of which large part of its facades and roof are collapsed. Although most of the building elements are intact, and three façades contain moderate structural damages, heavy damages of the roof and the south façade should have a remarkable influence on the overall damage assessment. Figure 28 shows the 3D model of masonry building and three types of damage are displayed in red circles. The detailed information about simulated sample 3 are provided below.

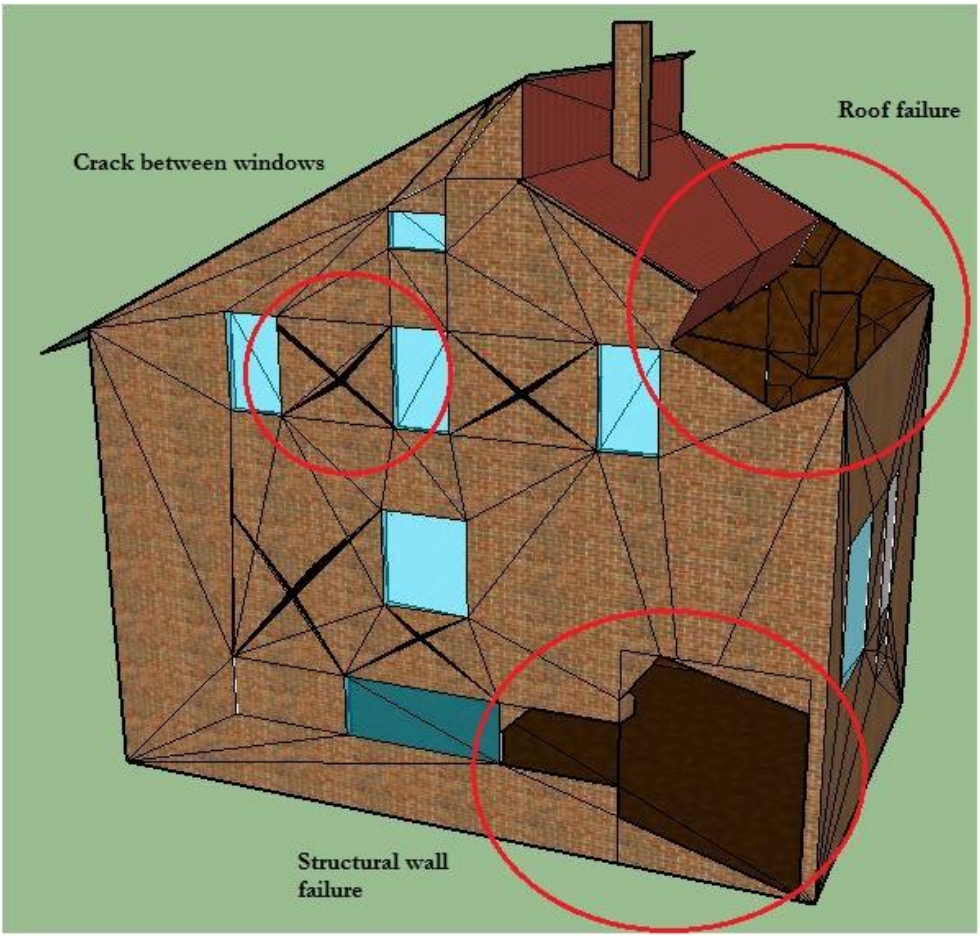


Figure 28: 3D model of sample 3 represents a damaged masonry building with heavy structural damages.

Damage description of sample3:

North façade:

- 20% of the structural walls are failed.
- 4 intersected cracks in the structural walls (width of cracks more than 0.006 m).

East façade:

- 10% of the structural walls are failed.

South façade:

- 30% of the structural walls are failed.
- 5 diagonal cracks in the structural walls (width of cracks less than 0.006m and more than 0.003).

West façade:

- 12% of the structural walls are failed.

Roof:

- 30% of the roof surfaces are collapsed.

Figure 29 illustrates involved damage levels of each façade individually. Both of the north and south facades include cracks at Level 3 and structural wall failure. However, 10% difference in wall failure leads to a significant change in the result of the membership function. The south façade has 0.5 degrees of membership in Level three and four equally but it more belongs to fifth damage level. While the north façade is classified in Level three and four with highest membership degree. Consequently, the calculated damage scores of the south and north façade are 68 and 45 percent respectively (Table 10). In general, the south façade has very heavy structural damage and the north façade contain heavy structural damage. Both cases prove that the system is influenced by higher levels more that lower damage levels, it is able to calculate more reliable and reasonable damage value. The east wall has only moderate structural damage and around 41% of the west wall is damaged that leads to classifying both mentioned facades in third damage level. Appendix 6 demonstrates calculating damage score of facades from aggregated fuzzy classes.

Figure 30 displays the damage levels and final percentage of the affected building. Three damage levels (Level three, four and five) are the result overall assessment. Although three facades do not belong to Level five, the building is highly influenced by damages at Level five due to heavy failures in roof surfaces and the west façade. In practice, observing heavy structural damages results to classify building as Level four or five regardless to the other detected damage evidence. The simulated masonry building is classified in Level four with 62% damages, even though the degree of membership of Level three is higher than the others. The results assessment reveal that the designed expert system imitates the human expert behavior in the case of heavy non-structurally and structurally damaged building successfully.

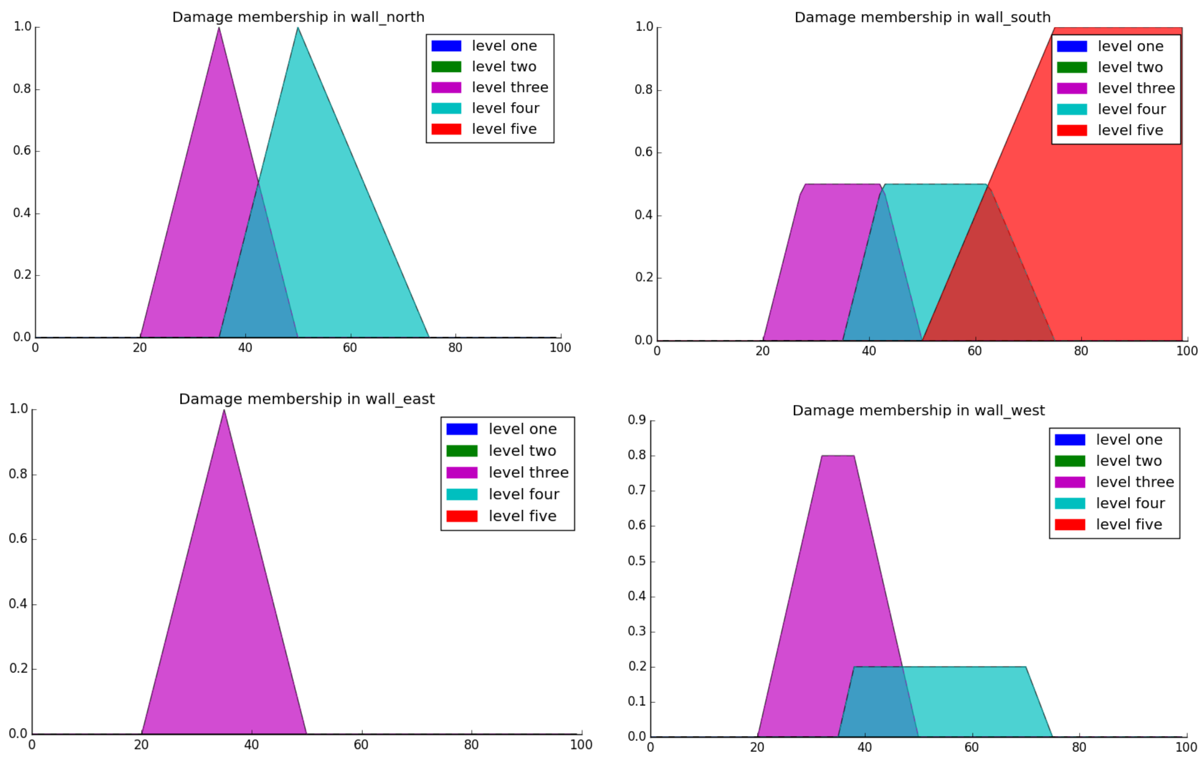


Figure 29: Façade damage assessment of sample 3. Each graph represents levels of damage and their membership degrees of specific façade.

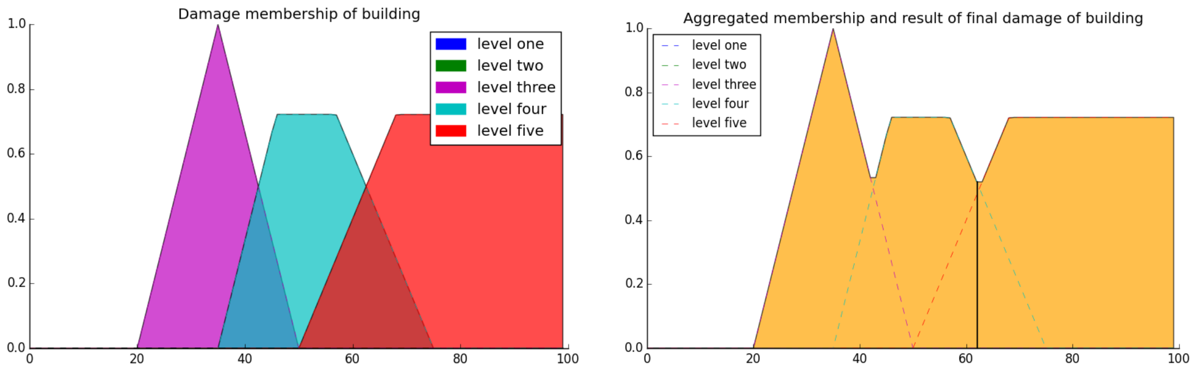


Figure 30: Left image represents calculated levels of damage of the building of sample 3 after overall assessment. Right image shows the aggregated damage levels, the overall damage value of the building is illustrated in black line.

5. CONCLUSION AND RECOMMENDATIONS

Modern remote sensing technologies are able to provide most of the required information for building damage assessment. Although several RS techniques and state-of-the-art technologies have successfully developed to detect damage evidence from all around the building, there is a gap to adopt this valuable information to interpret and assess damaged building individually. This research proposes a framework to link structural engineering knowledge and advanced remote sensing technologies to assess damaged structures faster and more constant in comparison with field survey. This study is a proof of concept for building damage evaluation according to semantic and spatial properties of RS-based damage evidence, further investigations are required to make it fully operative using real dataset.

It is attempted to identify and use valuable RS data for this application and involve structural engineering knowledge to obtain a reliable evaluation. The extension modules of CityGML are adopted to model and provide a 3D representation of damaged structure which preserves semantic and geometric attributes. It gives an opportunity for structural engineers to interpret remote sensing damage evidence and have a general overview of the building without visiting the area. The proposed fuzzy expert system is an artificial intelligence program to emulate the behavior of human experts for damage evaluation with considering the uncertainty of decision making. The fuzzy expert system captures the knowledge of structural engineers to implement detailed and overall building damage assessment automatically. The proposed system is composed of two individual fuzzy expert systems. The first system executes a detailed evaluation of facades separately and the second system implements overall building damage assessment by considering damages of facades and roof surfaces. Three simulated damaged buildings are generated to analyze and evaluate the performance of the framework under different damage conditions. The results of evaluations indicate that the designed system can assess low, moderate and heavy damaged building. Furthermore, the fuzzy system is capable of handling different damage levels to calculate the accurate final damage score of facades and building. In addition to 3D visualization, individual façade assessment allows exploring various aspects of the damaged structure. The answers to research questions are explained as follows:

1. Which damage evidence and attributes are important to assess the damaged building?
2. Which semantic and spatial relationships between damage evidence and parts of the building are essential to be represented?

   Two first research questions are answered by studying damage scales and BDA catalogues. The references are selected from different countries that consider various damage factors according to their environment and criteria. Also, a questionnaire is designed to obtain more detailed information on building evaluation based on structural engineering knowledge. A holistic collection of required damage evidence and attributes are made beside the influence of damage accumulation.

3. Which identified damages, attributes and relationships are detectible from RS data?

   Various types of damages, damage attributes and relationships that ate detectable from remote sensing data according to the modern methods and advanced technologies such as oblique imagery are identified and classified.

4. What is the most proper approach to store, represent and assess damage evidence and relationships?

   The CityGML model is adopted and extended to store and represent damaged building. The semantic and geometric attributes and relationships of building elements and damage evidence are preserved. The 3D model is imported into PostgreSQL database for assessment based on spatial and non-spatial criteria. The designed expert system is able to capture human expert knowledge and assess the level of damage automatically.

5. Which method is reliable to involve uncertainty for building damage assessment?

   The generated expert system is based on fuzzy logic to deal with linguistic uncertainty for building damage assessment.

6. How the generated framework can deal with the problem of lack of information?

   The building damage assessment is implemented in two steps, façade assessment and building evaluation. In the case of occlusion or insufficient information from particular part of the building, the first fuzzy expert system is able to assess facades. It can provide an overview of the building condition for specialists to judge about the level of damage or ask for more information through a field survey.

7. Is the generated framework capable of assessing damaged building which contains damage evidence at different levels?

   The results of damage assessment of simulated building indicate that proposed framework has the ability to store, represent and evaluate damaged structures. It successfully classifies, simplifies and interprets RS damage data by capturing structural engineers.

Limitations:

- Type, material and environment of the building directly affect classification and rules of damage assessment. Therefore, the generated rules and classes require modification under different conditions.
- The result of building damage assessment is highly influenced by the damage description, experience and knowledge of structural engineers. This research is a proof of concept for remote sensing based BDA based on studied damage documentation and knowledge of structural engineers. New analysis and questionnaire and a deeper interaction with experts could be necessary to define a more extensive framework.
- The framework is designed for representing, analysing and assessing data based on OGC standard. Therefore, 3D model of building and damage features should be generated according to the standard otherwise the fuzzy expert system could not execute spatial queries. Even if the input data are not following the OGC standard, it is possible to implement damage assessment through importing damage information into the fuzzy expert system as an auxiliary file. It would require some manual effort.
- In the case of occlusion, it is not possible to compute overall damage grade. The damage assessment of façades could be implemented to analyse and interpret the detected damage evidence from visible parts.
- The accuracy of the calculated percentage of damage is highly dependent on the accuracy and completeness of the extracted damage evidence and generated 3D model.
- The 3D spatial functions for analysing and processing 3D model are limited.
- The spatial queries to retrieving information in the fuzzy expert system are designed based on CityGML schema. The queries require small modification in the case of converting another 3D model to CityGML format.

Recommendations:

- A comprehensive evaluation of the framework usability could be achieved through assessing a real dataset of damaged building a participating of structural engineers.
- The proposed fuzzy classification and fuzzy rules could be improved by making a comparison between levels of damages calculated by the framework and structural engineers.
- Further research can involve data uncertainty into the framework, which requires modification of the expert system. It would lead to obtaining more uncertain damage level.
- The proposed framework represents the base for a complete and detailed models. The current work could be therefore extended and amplified in the future to involve additional information such as inside damage evidence.
- Damage accumulation is considered for individual façade. Involving the influence of connected damages in separate facades could lead to a more accurate overall level of damage.

LIST OF REFERENCES

Agugiaro, G. (2015). Energy planning tools and CityGML-based 3D virtual city models: experiences from Trento (Italy). *Applied Geomatics*. doi:10.1007/s12518-015-0163-2

ATC. (2005). ATC-20 Building Safety Evaluation Forms. Retrieved October 1, 2015, from https://www.atcouncil.org/products/downloadable-products/briefing-papers/45-downloadable/downloads/107-atc-20-download

Borne, M., Mercier, H., Mora, V., & Courtin, O. (n.d.). SFCGAL. Retrieved November 5, 2015, from http://www.sfcgal.org/

Borrmann, a., Kolbe, T. H., Donaubauer, A., Steuer, H., Jubierre, J. R., & Flurl, M. (2014). Multi-Scale Geometric-Semantic Modeling of Shield Tunnels for GIS and BIM Applications. *Computer-Aided Civil and Infrastructure Engineering*, *30*(4), 263–281. doi:10.1111/mice.12090

Daum, S., Borrmann, A., & Kolbe, T. H. (2015). A Spatio-semantic Query Language for the Integrated Analysis of City Models and Building Information Models. *Proceedings of the 3D Geoinfo*, 1–15.

Dell'Acqua, F., & Gamba, P. (2012). Remote sensing and earthquake damage assessment: Experiences, limits, and perspectives. *IEEE*, *100*(10), 2876–2890. doi:10.1109/JPROC.2012.2196404

Dong, L., & Shan, J. (2013). A comprehensive review of earthquake-induced building damage detection with remote sensing techniques. *ISPRS Journal of Photogrammetry and Remote Sensing*, *84*, 85–99. doi:10.1016/j.isprsjprs.2013.06.011

Dong, P. L., & Guo, H. D. (2012). A framework for automated assessment of post-earthquake building damage using geospatial data. *International Journal of Remote Sensing*, *33*(1), 81–100. doi:10.1080/01431161.2011.582188

Dubois, D., Member, S., & Lepage, R. (2014). Fast and efficient evaluation of building damage from very high resolution optical satellite images. *IEEE*, *7*(10), 4167–4176.

Durkin, J. (1990). Research Review : Application of Expert Systems in the Sciences. *Ohio Journal of Science*, *90*(5), 171–179.

Ehrlich, D., Guo, H. D., Molch, K., Ma, J. W., & Pesaresi, M. (2009). Identifying damage caused by the 2008 Wenchuan earthquake from VHR remote sensing data. *International Journal of Digital Earth*, *2*(4), 309–326. doi:10.1080/17538940902767401

Farhidzadeh, a., Dehghan-Niri, E., Moustafa, a., Salamone, S., & Whittaker, a. (2013). Damage Assessment of Reinforced Concrete Structures Using Fractal Analysis of Residual Crack Patterns. *Experimental Mechanics*, *53*(9), 1607–1619. doi:10.1007/s11340-013-9769-7

Fernandez Galarreta, J., Kerle, N., & Gerke, M. (2015). UAV-based urban structural damage assessment using object-based image analysis and semantic reasoning. *Natural Hazards and Earth System Science*, *15*(January), 1087–1101. doi:10.5194/nhess-15-1087-2015

Foulser-Piggott, R., & Spence, R. (2013). Extending EMS-98 for more convenient application outside Europe I : Review of field experience using EMS-98. In *Vienna Congress on Recent Advances in Earthquake Engineering and Structural Dynamics 2013* (pp. 1–9).

Foulser-Piggott, R., Spence, R., Saito, K., & Brown, D. . (2012). The use of remote sensing for post-earthquake damage assessment: lessons lessons from from recent recent events, events, and future assessment: prospects. In *15 WCEE*. Lisbon.

Geiß, C. (2014). *Seismic vulnerability assessment of built enviroment with remote sensing*. Humboldt University of Berlin.

Gerke, M., & Kerle, N. (2011). Automatic Structural Seismic Damage Assessment with Airborne Oblique Pictometry© Imagery. *Photogrammetric Engineering & Remote Sensing*, *77*(9), 885–898.

doi:10.14358/PERS.77.9.885

Gröger, G., Kolbe, T. H., Nagel, C., & Häfele, K.-H. (2012). Open Geospatial Consortium OGC City Geography Markup Language (CityGML) En- coding Standard. Retrieved from http://www.opengeospatial.org/legal/

Grünthal, G. (1998). *European Macroseismic Scale 1998*. *European Center for Geodynamics and Seismology* (Vol. 100). Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:European+Macroseismic+Scale+1998#0

Hopgood, A. A. (2001). *Intelligent Systems Engineers and Scientists*. *New York* (Second.). CRC press.

IFC. (2013). Industry Foundation Classes. Retrieved November 1, 2015, from http://www.ifcwiki.org/index.php/Main_Page

Ishii, M., Goto, T., Sugiyama, T., Saji, H., & Abe, K. (2002). Detection of Earthquake Damaged Areas from Aerial Photographs by Using Color and Edge Information. In *The 5th asian conference on computer visio*. Melborn.

Kahani, M. (n.d.). Fuzzy Expert System. Retrieved January 10, 2016, from http://www.powershow.com/view2b/42528d-ZjFiY/Fuzzy_Expert_System_Fuzzy_Logic_powerpoint_ppt_presentation

Kerle, N. (2010). Satellite-based damage mapping following the 2006 Indonesia earthquake—How accurate was it? *International Journal of Applied Earth Observation and Geoinformation*, *12*(6), 466–476. doi:10.1016/j.jag.2010.07.004

Kerle, N. (2013). *Remote sensing based post - disaster damage mapping with collaborative methods*. *Intelligent Systems for Crisis Management*. doi:10.1007/978-3-642-33218-0

Kerle, N., & Hoffman, R. R. (2013). Collaborative damage mapping for emergency response: The role of Cognitive Systems Engineering. *Natural Hazards and Earth System Science*, *13*(1), 97–113. doi:10.5194/nhess-13-97-2013

Khoshelham, K., Oude Elberink, S., & Xu, S. (2013). Segment-based classification of damaged building roofs in aerial laser scanning data. *IEEE Geoscience and Remote Sensing Letters*, *10*(5), 1258–1262. doi:10.1109/LGRS.2013.2257676

Knapp, R. B. (2004). Fuzzy Sets and Pattern Recognition. Retrieved January 10, 2016, from http://www.cs.princeton.edu/courses/archive/fall07/cos436/HIDDEN/Knapp/fuzzy004.htm

Kolbe, T. H. (n.d.). 3D City Database. Retrieved November 1, 2016, from http://www.3dcitydb.net/3dcitydb/3ddatabase/

Kolbe, T. H. (2009). Representing and Exchanging 3D City Models with CityGML. In *3D Geo-Information Sciences* (pp. 15–31). doi:10.1007/978-3-540-87395-2_2

Li, P., Xu, H., & Guo, J. (2010). Urban building damage detection from very high resolution imagery using OCSVM and spatial features. *International Journal of Remote Sensing*, *31*(13), 3393–3409. doi:10.1080/01431161003727705

Ma, L., Sacks, R., & Zeibak-Shini, R. (2015). Information modeling of earthquake-damaged reinforced concrete structures. *Advanced Engineering Informatics*, (November). doi:10.1016/j.aei.2015.01.007

Miyamoto. (2010). kitmiyamoto. Retrieved November 5, 2015, from http://kitmiyamoto.com/earthquake-field-reports/2010-haiti-earthquake-journal-entry-1/

Nouvel, R., Zirak, M., Dastageeri, H., Coors, V., & Eicker, U. (2014). Urban Energy Analysis Based on 3D City Model for National Scale Applications. In *Presented at the IBPSA Germany Conference* (Vol. 8, pp. 83–90).

Okada, S., & Takai, N. (2000). CLASSIFICATIONS OF STRUCTURAL TYPES AND DAMAGE

PATTERNS OF. *In Proceedings of the 12th World Conference on Earthquake Engineering*, (paper 0705), 1–8.

Paal, S. G., Jeon, J.-S., Brilakis, I., & DesRoches, R. (2015). Automated Damage Index Estimation of Reinforced Concrete Columns for Post-Earthquake Evaluations. *Journal of Structural Engineering*, (SEPTEMBER), 04014228. doi:10.1061/(ASCE)ST.1943-541X.0001200

Shu-Hsien Liao. (2005). Expert system methodologies and applications—a decade review from 1995 to 2004. *Expert Systems with Applications*, *28*(1), 93–103. doi:10.1016/j.eswa.2004.08.003

Torok, M., Golparvar-Fard, M., & Kochersberger, K. (2012). Image-Based Automated 3D Crack Detection for Post-disaster Building Assessment. In *2012 International Conference on Computing in Civil Engineering*.

Tripathi, K. P. (2011). A Review on Knowledge-based Expert System : Concept and Architecture. *Artificial Intelligence Techniques - Novel Approaches & Practical Applications*, *4*(4), 19–23. doi:10.5120/2845-226

Vetrivel, A., Gerke, M., Kerle, N., & Vosselman, G. (2015a). Identification of damage in buildings based on gaps in 3D point clouds from very high resolution oblique airborne images. *ISPRS Journal of Photogrammetry and Remote Sensing*, *105*, 61–78. doi:10.1016/j.isprsjprs.2015.03.016

Vetrivel, A., Gerke, M., Kerle, N., & Vosselman, G. (2015b). Segmentation of UAV-based images incorporating 3D point cloud information. In *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* (Vol. XL–3/W2, pp. 261–268). doi:10.5194/isprsarchives-XL-3-W2-261-2015

Yamazaki, F., Kouchi, K., Kohiyama, M., Muraoka, N., & Matsuoka, M. (2004). Earthquake damage detection using high-resolution satellite images. *IEEE International Geoscience and Remote Sensing Symposium*, *4*(c), 2280–2283. doi:10.1109/IGARSS.2004.1369739

Zadeh, L. A. (1965). Fuzzy Sets-Information and Control-1965.pdf. *Information and Control*, (8), 338–353.

Zhu, Z., German, S., & Brilakis, I. (2011). Visual retrieval of concrete crack properties for automated post-earthquake structural safety evaluation. *Automation in Construction*, *20*(7), 874–883. doi:10.1016/j.autcon.2011.03.004

APPENDIX 1

Building Damage Classification Questionnaire

1. Please classify the severity of a crack in a concrete wall in 3 classes (sever, moderate, low) based on the width of the crack in mm. (it should be mentioned that other factors such as thickness of wall is important but in the case of building damage assessment it is not possible to have all this information and also this process is not for detailed evaluation therefore it is required to have a general reference).
2. Please classify the severity of crack on a reinforced concrete column in 3 classes (sever, moderate, low) based on the width of the crack in mm.
3. Please classify the severity of crack on a masonry wall in 3 classes (sever, moderate, low) based on the width of the crack in mm. (e.g. in one of the earthquake catalogue the damage was classified in Sever (crack width > 6 mm), Moderate (6 mm > crack width > 3 mm) and Low (crack width < 3mm))
4. Please classify the severity of crack on a reinforced concrete column-beam joint in 3 classes (sever, moderate, low) based on the width of the crack in mm.
5. Please classify the severity of tilt on a reinforced concrete column in 3 classes (sever, moderate, low) based on the degree of inclination.(in sever situation the possibility of collapsing is high, in the moderate level it is highly damaged but the possibility of collapsing is low, the low level refers to not serious danger).
6. Please define a threshold for tilted masonry wall based on the degree, which can be used to distinguish unstable walls from others.
7. Please classify the severity of tilt on a reinforced wall in 3 classes (sever, moderate, low) based on the degree of inclination.
8. Please classify tilted building (masonry and reinforced concrete building) after an earthquake in two classes (danger and safe) based on the degree. E.g. if whole structure is tilted 20°.

9. Please give a level of damage from 1 to 5 (1 refers to low and 5 dedicates to high level of damage) to each bellow damaged part of building:

- Column contains few moderate cracks.
- Column contains a lot of moderate cracks.
- Column contains sever cracks.
- Beam-Column joint contains moderate cracks.
- Beam-Column joint contains sever cracks.
- Column contains spalling.
- Column contains spalling and crack.

10. One of factors that is interesting in this research is defining the damage accumulation, therefore identifying the some general relationships that can reveal the damage accumulation in a structure is a proposed solution in this research. Bellow some possible relationships are stated. Which of connection between damaged parts of building is important to be considered in building assessment and can increase the total level of damage or put the building in an unsafe situation? Please explain.

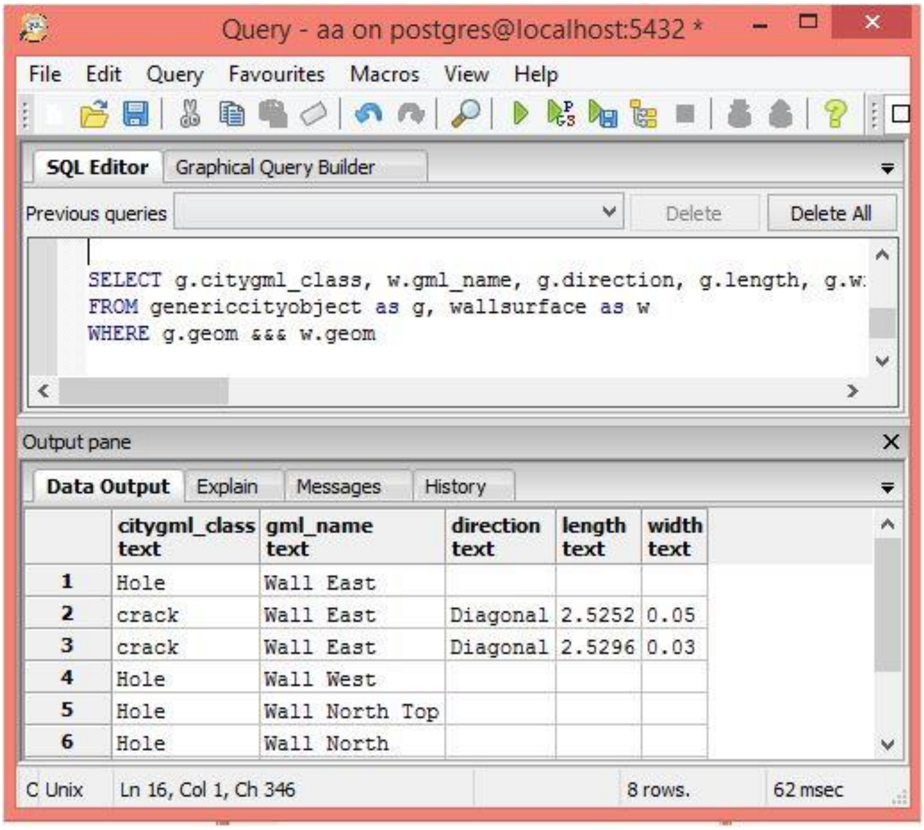
- Tilted column in the corner of the building and connected to two damaged wall (Damaged wall refers to any type of the damage such as failure part of the wall, tilt, crack and spalling). Please specify which of wall damages id important in the case of damage accumulation.
- Damaged wall connected to (or close to) collapsed part of the roof.
- Partial collapsed wall connected to column that contains sever crack.
- Crossed cracks.
- Is it true to that the intersection of cracks and spalling on building elements can increase the level of damage on that specific part?
- Sever vertical cracks reveal the separation between walls.
- Is it true that tilted column can reveal the connected beam is bended?
- Please add more important relationships which is not mentioned here.

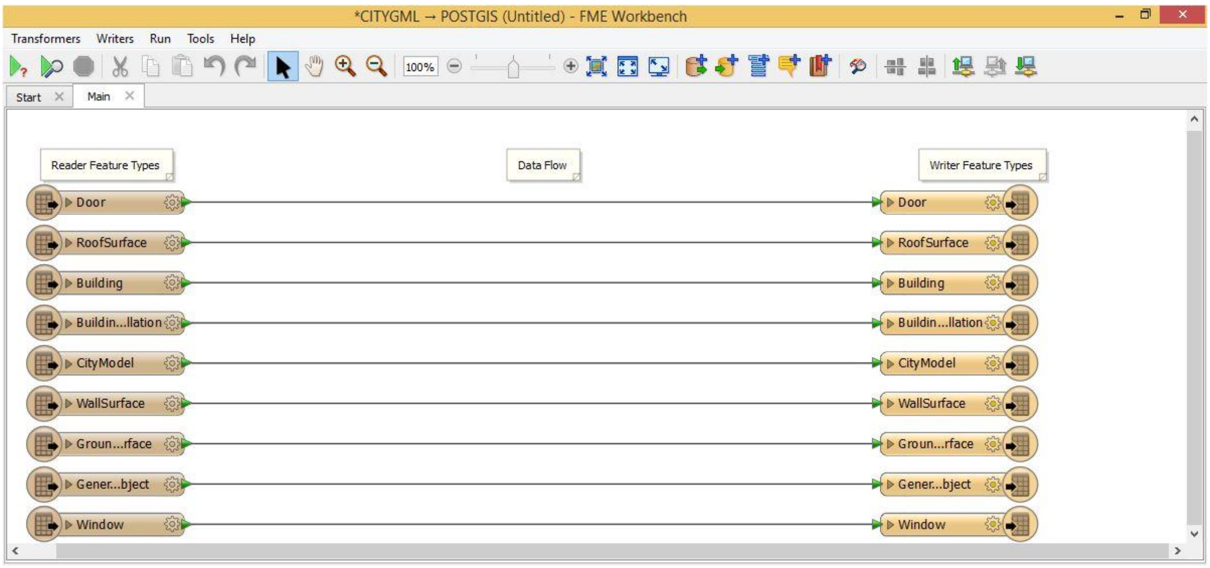
11. What is the distance in centimeter between crack and building elements that we can guess the crack penetrated to the building element but it is not visible (e.g. if we have a crack on the wall within 10 cm of a beam, it can be assumed that crack penetrated into the beam but from images it is not possible to see)?

12. Is there any relationship between the hole on the roof and other damage on the facades of the building? If yes what type of damages are they?

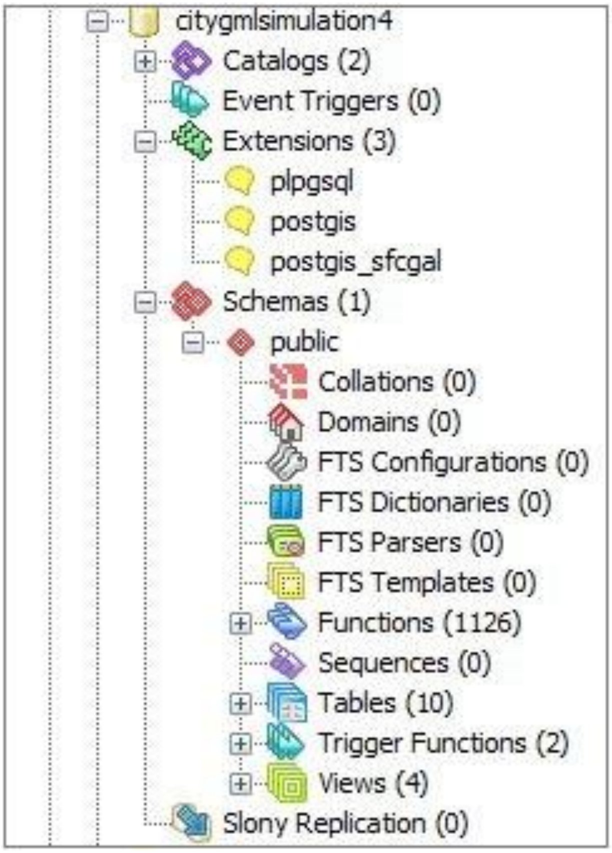
APPENDIX 2

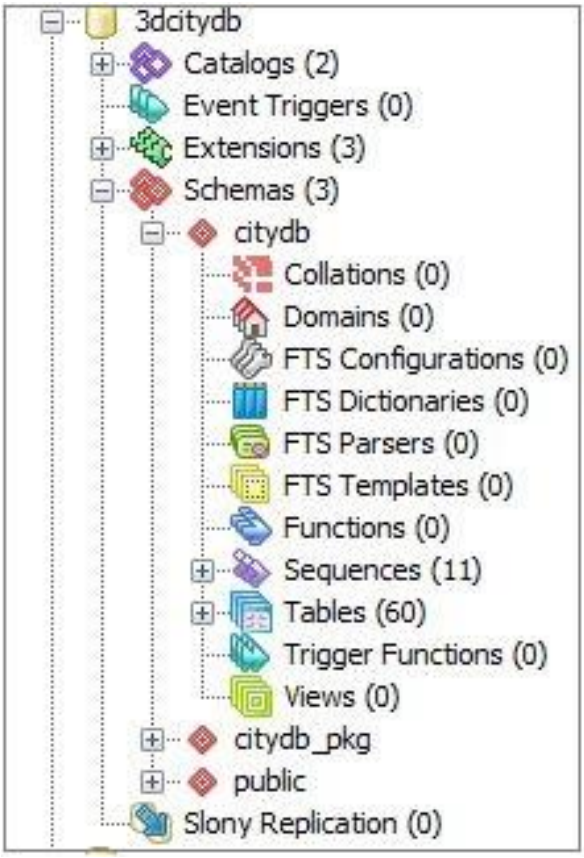
Top image illustrates an example of consistency test of PostGIS for 3D spatial queries. Bottom image displays the process of transferring 3D model into PostgreSQL database using FME software. FME creates a specific table for each table of CityGML model.





Left image illustrates the 3D model inside database, it shows that the PostGIS and SFCGAL are added as extension to run spatial queries. Right shows the same 3D model inside database which is transferred by 3DcityDB, which created 50 tables more than FME for the same dataset.





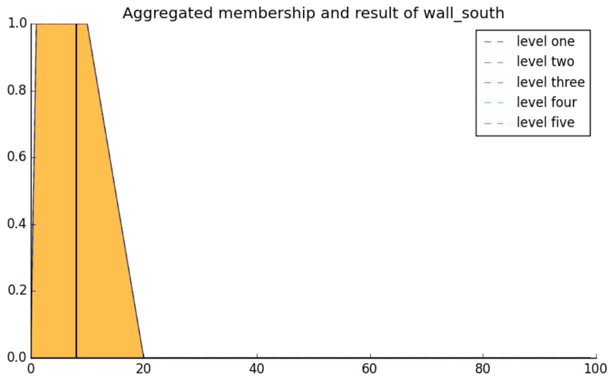
APPENDIX 3

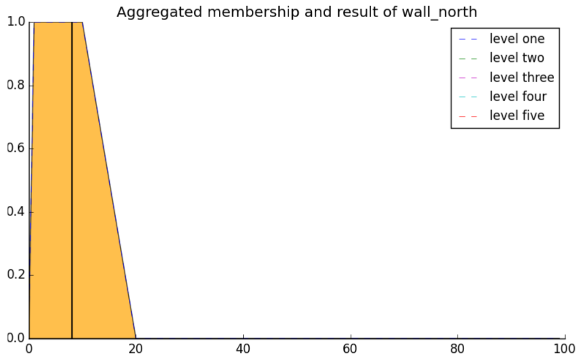
The image represents a part of CityGML file which is extended to contain damage evidence. The 3D geometry and attributes of the crack are imported as new *Generic City Object*.

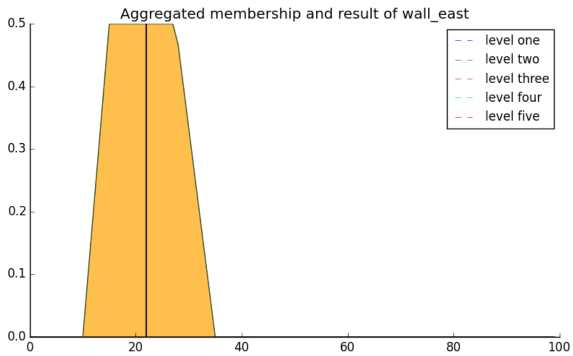
```
<cityObjectMember>
    <gen:GenericCityObject>
        <gen:doubleAttribute name="Width">
            <gen:value uom="#m">0.005</gen:value>
        </gen:doubleAttribute>
        <gen:stringAttribute name="Direction">
            <gen:value>Diagonal</gen:value>
        </gen:stringAttribute>
        <gen:function>damage</gen:function>
        <gen:class>crack</gen:class>
        <gen:lod1Geometry>
            <gml:MultiSurface>
                <gml:surfaceMember>
                    <gml:Polygon gml:id="PolyID1112">
                        <gml:exterior>
                            <gml:LinearRing
                                gml:id="PolyID1112_0">
                                <gml:pos>445541.3370 5444898.1621 3.8250</gml:pos>
                                <gml:pos>445541.3370 5444898.1621 3.7750</gml:pos>
                                <gml:pos>445539.0526 5444897.5500 2.9400</gml:pos>
                                <gml:pos>445539.0526 5444897.5500 2.9900</gml:pos>
                                <gml:pos>445541.3370 5444898.1621 3.8250</gml:pos>
                            </gml:LinearRing>
                        </gml:exterior>
```

APPENDIX 4

The result of aggregating damages classes of each façade and generating damage value of sample 1 in first fuzzy expert system is represented below.

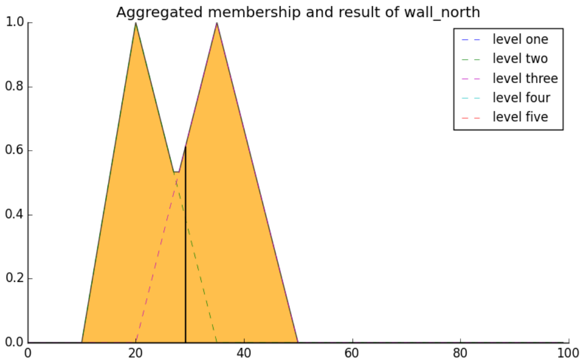


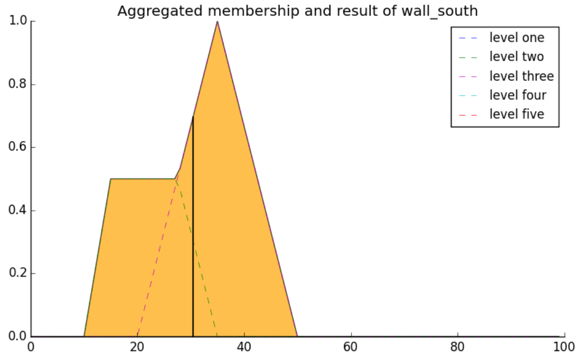


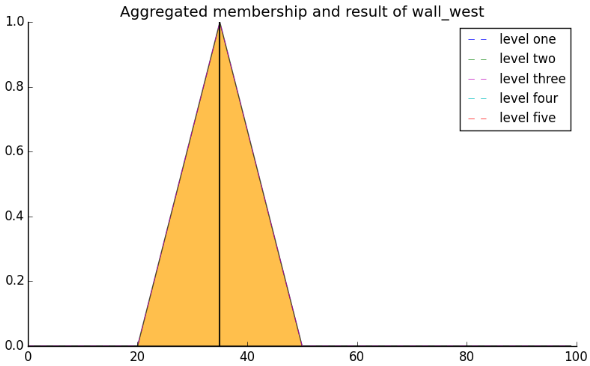


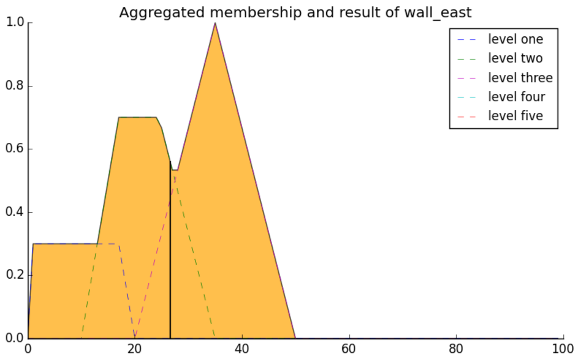
APPENDIX 5

The result of aggregating damages classes of each façade and generating damage value of sample 2 in first fuzzy expert system is represented below.



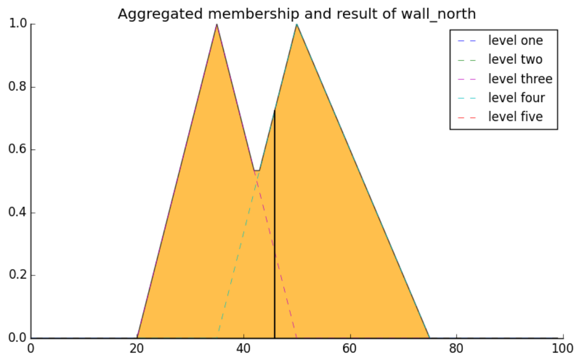


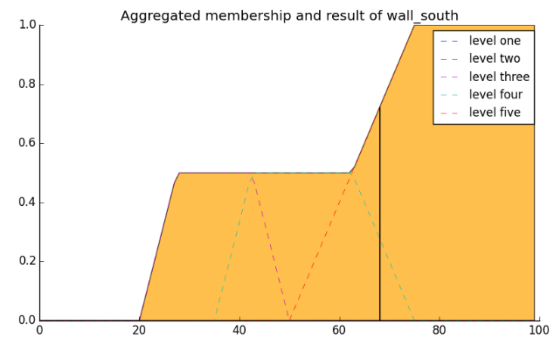


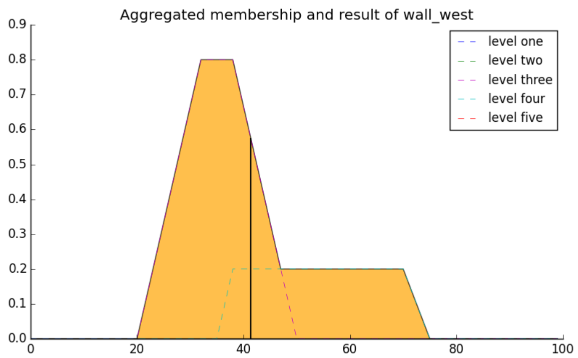


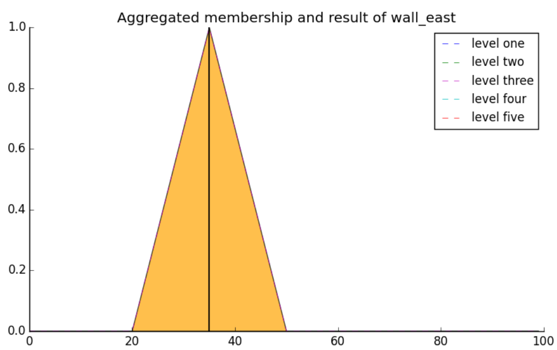
APPENDIX 6

The result of aggregating damages classes of each façade and generating damage value of sample 3 in first fuzzy expert system is represented below









APPENDIX 7

```
import numpy as np
import matplotlib.pyplot as plt
import skfuzzy as fuzz
import psycopg2
import csv
import matplotlib.patches as mpatches

#if set to True it queries the geometries from the database. If set to false it uses the
result from an auxiliary file
databaseData = False
database_var = 'ysimulation_roof_hole'
host_var= 'localhost'
user_var = 'postgres'
password_var = '3155'
port_var = 5432

#in case of not using database need to specify the file
file_name = 'sample3.csv'


##Generate universe function
#defines the range of each of the fuzzy elements
array =np.arange(0,100,1)

# The following rules are for Reinforced Concrete Building
# Membership functions for non_strucural_wall_failure
nwf_low = fuzz.trapmf(array, [1,1 ,10, 20])
nwf_moderate = fuzz.trimf(array, [10, 20, 30])
nwf_high = fuzz.trapmf(array, [20, 30, 100,100])

#Membership function for column spalling
cs_low = fuzz.trapmf(array, [1,1,10,20])
cs_high =fuzz.trapmf(array, [10,20,100,100])

#Membership function for beam spalling
bs_low =fuzz.trapmf(array, [1,1,10,20])
bs_high = fuzz.trapmf(array, [10,20,100,100])

# Membership functions for strucural_wall_failure
wf_low = fuzz.trapmf(array, [1,1,10,20])
wf_moderate = fuzz.trimf(array, [10,20,30])
wf_high = fuzz.trapmf(array, [20, 30, 100,100])

# Membership functions for wall_spalling
ws_low = fuzz.trapmf(array, [1,1,10,40])
ws_high = fuzz.trapmf(array, [10,40, 100,100])

# Membership functions for roof_failure
rf_low = fuzz.trapmf(array, [1,1,5,10])
rf_moderate = fuzz.trimf(array, [5, 10, 40])
rf_high = fuzz.trapmf(array, [10,40, 100,100])

#first the cracks are classified based on their width and shape in three classes( Level1,
Level2, Level3)
#for each facade the number of crack*10 to have same shape in fuzzy aggregation
# Membership functions for level1_crack_non_structural_wall
l1nw_low = fuzz.trapmf(array, [1,1,20,40])
l1nw_moderate = fuzz.trimf(array, [20, 40, 60])
l1nw_high = fuzz.trapmf(array, [40,60, 100,100])

# Membership functions for level2_crack_non_structural_wall
l2nw_low = fuzz.trapmf(array, [1,1,20,40])
l2nw_moderate = fuzz.trimf(array, [20, 40, 60])
l2nw_high = fuzz.trapmf(array, [40,60, 100,100])

# Membership functions for level3_crack_non_structural_wall
l3nw_low = fuzz.trapmf(array, [1,1,20,40])
l3nw_moderate = fuzz.trimf(array, [20, 40, 60])
l3nw_high = fuzz.trapmf(array, [40,60, 100,100])
```

```
# Membership functions for level1_crack on structural_wall
l1w_low = fuzz.trapmf(array, [1,1,20,40])
l1w_moderate = fuzz.trimf(array, [20, 40, 60])
l1w_high = fuzz.trapmf(array, [40,60, 100,100])

# Membership functions for level2_crack on structural_wall
l2w_low = fuzz.trapmf(array, [1,1,20,40])
l2w_moderate = fuzz.trimf(array, [20, 40, 60])
l2w_high = fuzz.trapmf(array, [40,60, 100,100])

# Membership functions for level3_crack on structural_wall
l3w_low = fuzz.trapmf(array, [1,1,20,40])
l3w_moderate = fuzz.trimf(array, [20, 40, 60])
l3w_high = fuzz.trapmf(array, [40,60, 100,100])

#crack on column and beam are classified based on their width*10 only
# Membership functions for crack_column the width of the crack *10000
cc_low = fuzz.trapmf(array, [1,10,20,50])
cc_moderate = fuzz.trimf(array, [20, 50, 70])
cc_high = fuzz.trapmf(array, [50,70,100,100])

# Membership functions for crack_beam based on the width *10000
cb_low = fuzz.trapmf(array, [1,10,20,50])
cb_moderate = fuzz.trimf(array, [20, 50, 70])
cb_high = fuzz.trapmf(array, [50,70,100,100])

# Membership functions for number of structural parts that are within specific 3d distance of
damage evidence
ds_low = fuzz.trapmf(array, [1,5,10,15])
ds_high = fuzz.trapmf(array, [10,15,100,100])

# Membership function for tilted_structural wall based on the degree of inclination
tw_low = fuzz.trapmf(array, [1,1,10,20])
tw_high = fuzz.trapmf(array,[10,20,100,100])

#Membership function for tilted non_structural_wall
tnw_low= fuzz.trapmf(array, [1,1,10,20])
tnw_high= fuzz.trapmf(array, [10,20,100,100])

#Membership fuction for tilted column
tc_low= fuzz.trapmf(array,[1,1,10,20])
tc_high=fuzz.trapmf(array,[10,20,100,100])

# Membership functions for level_wall_damage
lwd_one = fuzz.trapmf(array, [1,1 ,10, 20])
lwd_two = fuzz.trimf(array, [10, 20, 35])
lwd_three = fuzz.trimf(array, [20, 35, 50])
lwd_four = fuzz.trimf(array, [35,50,75])
lwd_five = fuzz.trapmf(array, [50, 75, 100,100])

# Example
# Visualize membership functions for non_strucural_wall_failure
fig, ax = plt.subplots()
ax.plot(array, lwd_one, 'b', linewidth=0.5, linestyle='-')
ax.plot(array, lwd_two, 'g', linewidth=0.5, linestyle='-')
ax.plot(array, lwd_three, 'm', linewidth=0.5, linestyle='-')
ax.plot(array, lwd_four, 'c', linewidth=0.5, linestyle='-')
ax.plot(array, lwd_five, 'r', linewidth=0.5, linestyle='-')
ax.set_ylabel('Fuzzy membership')
ax.set_xlabel('Level of damage of individual facade')
ax.set_ylim(-0.05, 1.05);
plt.savefig('1')
plt.tight_layout()
one= mpatches.Patch(color='b', label='level one')
two = mpatches.Patch(color='g', label='level two')
three = mpatches.Patch(color='m', label='level three')
four = mpatches.Patch(color='c', label='level four')
five = mpatches.Patch(color='r', label='level five')

plt.legend(handles=[one,two,three,four,five])
plt.show()


if(databaseData):
    # Get a database connection for our database:
```

```
    db = psycopg2.connect(database=database_var,host=host_var,user=user_var,
password=password_var, port=port_var)
    # Get a cursor to be used:
    mycursor = db.cursor()
    #import data from database...
    #first select and assess each facade seperately
    mycursor.execute("SELECT gml_name FROM wallsurface")
    walls = mycursor.fetchall()
    mycursor.close
    wallsurfaces = []
    roof = []

    mycursor = db.cursor()
    mycursor.execute("select (st_3darea(g.geom))/(st_3darea(r.geom))*100 as rf\
        from (select st_collect(ARRAY((select geom from roofsurface))) as geom) as r,\
        (select st_collect(ARRAY((select g.geom from genericcityobject as g, roofsurface as r
where st_3dintersects(g.geom,r.geom) ))) as geom) as g")
    rf = mycursor.fetchall()
    mycursor.close
    aux = {}
    if len(rf)>0 and not rf[0][0] is None:
        aux['rf'] = rf[0][0]
    else:
        aux['rf'] = 0
    roof.append(aux)

    for wall in walls:
        aux = {}
        aux['wall_name'] = wall[0]

        mycursor = db.cursor()
        mycursor.execute("select (st_3darea(g.geom))/(st_3darea(w.geom))*100 as nwf\
        from wallsurface as w,\
        (select st_collect(ARRAY((select g.geom from genericcityobject as g, wallsurface as w
where st_3dintersects(g.geom,w.geom) and g.citygml_class='hole' and w.role=false and
w.gml_name = %s))) as geom) as g\
        where w.gml_name = %s",(wall,wall))
        nwf = mycursor.fetchall()
        mycursor.close
        if len(nwf)>0 and not nwf[0][0] is None:
            aux['nwf'] = nwf[0][0]
        else:
            aux['nwf'] = 0

        mycursor = db.cursor()
        mycursor.execute("select (st_3darea(g.geom))/(st_3darea(w.geom))*100 as wf\
        from wallsurface as w,\
        (select st_collect(ARRAY((select g.geom from genericcityobject as g, wallsurface as w
where st_3dintersects(g.geom,w.geom) and g.citygml_class='hole' and w.role=true and w.gml_name
= %s))) as geom) as g\
        where w.gml_name = %s",(wall,wall))
        wf = mycursor.fetchall()
        mycursor.close
        if len(wf)>0 and not wf[0][0] is None:
            aux['wf'] = wf[0][0]
        else:
            aux['wf'] = 0

        mycursor = db.cursor()
        mycursor.execute("select (st_3darea(g.geom))/(st_3darea(w.geom))*100 as ws\
        from  wallsurface as w,\
        (select st_collect(ARRAY((select g.geom from genericcityobject as g, wallsurface as w
where st_3dintersects(g.geom,w.geom) and g.citygml_class='spalling' and w.gml_name = %s ))) as
geom) as g\
        where w.gml_name = %s",(wall,wall))
        ws = mycursor.fetchall()
        mycursor.close
        if len(ws)>0 and not ws[0][0] is None:
            aux['ws'] = ws[0][0]
        else:
            aux['ws'] = 0

        mycursor = db.cursor()
        mycursor.execute("select (st_3darea(g.geom))/(st_3darea(b.geom))*100 as cs\
        from genericcityobject as g, buildinginstallation as b, wallsurface as w\
```

```
        where st_3dintersects(b.geom,g.geom) and g.citygml_class='spalling'    and
b.gml_name='column'  and st_3dintersects(b.geom,w.geom)and w.gml_name = %s",(wall))
        cs = mycursor.fetchall()
        mycursor.close
        if len(cs)>0:
            aux['cs'] = cs[0]
        else:
            aux['cs'] = 0

        mycursor = db.cursor()
        mycursor.execute("select (st_3darea(g.geom))/(st_3darea(b.geom))*100 as bs\
        from genericcityobject as g, buildinginstallation as b, wallsurface as w\
        where st_3dintersects(b.geom,g.geom) and g.citygml_class='spalling'    and
b.gml_name='beam' and st_3dintersects(b.geom,w.geom)and w.gml_name = %s",(wall))
        bs = mycursor.fetchall()
        mycursor.close
        if len(bs)>0 and bs[0][0] is None:
            aux['bs'] = bs[0][0]
        else:
            aux['bs'] = 0

        mycursor = db.cursor()
        mycursor.execute("select count(*)*10 as l1nw\
        from genericcityobject as g, wallsurface as w \
        where (st_3dintersects(g.geom,w.geom) and g.width::numeric <= 0.003 and g.direction =
'vertical' and w.gml_name = %s  and w.role=false) or (st_3dintersects(g.geom,w.geom) and
g.width::numeric <= 0.003 and g.direction = 'horizontal' and w.role=false and w.gml_name =
%s)",(wall,wall))
        l1nw = mycursor.fetchall()
        mycursor.close
        if len(l1nw)>0 and l1nw[0][0] is None:
            aux['l1nw'] = l1nw[0][0]
        else:
            aux['l1nw'] = 0

        mycursor = db.cursor()
        mycursor.execute("select count(*)*10 as l2nw\
        from genericcityobject as g, wallsurface as w \
        where (st_3dintersects(g.geom,w.geom) and g.width::numeric <= 0.003 and g.direction =
'diagonal' and w.gml_name = %s and w.role=false) or (st_3dintersects(g.geom,w.geom) and
g.width::numeric <= 0.003 and g.direction = 'intersected' and w.gml_name = %s and
w.role=false) or (st_3dintersects(g.geom,w.geom) and g.width::numeric > 0.003 and
g.width::numeric < 0.005 and g.direction = 'horizontal' and w.gml_name = %s and w.role=false)
or (st_3dintersects(g.geom,w.geom) and g.width::numeric > 0.003 and g.direction = 'vertical'
and w.gml_name = %s and w.role=false)",(wall,wall,wall,wall))
        l2nw = mycursor.fetchall()
        mycursor.close
        if len(l2nw)>0 and l2nw[0][0] is None:
            aux['l2nw'] = l2nw[0][0]
        else:
            aux['l2nw'] = 0

        mycursor = db.cursor()
        mycursor.execute("select count(*)*10 as l3nw\
        from genericcityobject as g, wallsurface as w \
        where (st_3dintersects(g.geom,w.geom) and g.width::numeric > 0.003 and g.direction =
'diagonal' and w.gml_name = %s and w.role=false) or (st_3dintersects(g.geom,w.geom) and
g.width::numeric > 0.003 and g.direction = 'intersected' and w.gml_name = %s and w.role=false)
or (st_3dintersects(g.geom,w.geom) and g.width::numeric >= 0.006 and g.direction =
'horizontal' and w.gml_name = %s and w.role=false)",(wall,wall,wall))
        l3nw = mycursor.fetchall()
        mycursor.close
        if len(l3nw)>0 and l3nw[0][0] is None:
            aux['l3nw'] = l3nw[0][0]
        else:
            aux['l3nw'] = 0

        mycursor = db.cursor()
        mycursor.execute("select count(*)*10 as l1w\
        from genericcityobject as g, wallsurface as w\
        where (st_3dintersects(g.geom,w.geom) and g.width::numeric <= 0.003 and g.direction =
'vertical' and w.gml_name = %s and w.role=true) or (st_3dintersects(g.geom,w.geom) and
g.width::numeric <= 0.003 and g.direction = 'horizontal' and w.gml_name = %s and
w.role=true)",(wall,wall))
        l1w = mycursor.fetchall()
```

```
        mycursor.close
        if len(l1w)>0 and l1w[0][0] is None:
            aux['l1w'] = l1w[0][0]
        else:
            aux['l1w'] = 0

        mycursor = db.cursor()
        mycursor.execute("select count(*)*10 as l2w\
        from genericcityobject as g, wallsurface as w\
        where (st_3dintersects(g.geom,w.geom) and g.width::numeric <= 0.003 and g.direction =
'diagonal' and w.gml_name = %s and w.role=true) or (st_3dintersects(g.geom,w.geom) and
g.width::numeric <= 0.003 and g.direction = 'intersected' and w.gml_name = %s and w.role=true)
or (st_3dintersects(g.geom,w.geom) and g.width::numeric > 0.003 and g.width::numeric < 0.005
and g.direction = 'horizontal' and w.gml_name = %s and w.role=true) or
(st_3dintersects(g.geom,w.geom) and g.width::numeric > 0.003 and g.direction = 'vertical' and
w.gml_name = %s and w.role=true)",(wall,wall,wall,wall))
        l2w = mycursor.fetchall()
        mycursor.close
        if len(l2w)>0 and l2w[0][0] is None:
            aux['l2w'] = l2w[0][0]
        else:
            aux['l2w'] = 0

        mycursor = db.cursor()
        mycursor.execute("select count(*)*10 as l3w\
        from genericcityobject as g, wallsurface as w \
        where (st_3dintersects(g.geom,w.geom) and g.width::numeric > 0.003 and g.direction =
'diagonal' and w.gml_name = %s and w.role=true) or (st_3dintersects(g.geom,w.geom) and
g.width::numeric > 0.003 and g.direction = 'intersected' and w.gml_name = %s and w.role=true)
or (st_3dintersects(g.geom,w.geom) and g.width::numeric >= 0.006 and g.direction =
'horizontal' and w.gml_name = %s and w.role=true)",(wall,wall,wall))
        l3w = mycursor.fetchall()
        mycursor.close
        if len(l3w)>0 and l3w[0][0] is None:
            aux['l3w'] = l3w[0][0]
        else:
            aux['l3w'] = 0

        mycursor = db.cursor()
        mycursor.execute("select g.width*10000 as cc\
        from genericcityobject as g, buildinginstallation as b, wallsurface as w\
        where st_3dintersects(b.geom,g.geom) and g.citygml_class='crack' and
b.gml_name='column' and w.gml_name = %s and st_3dintersects(b.geom,w.geom)",(wall))
        cc = mycursor.fetchall()
        mycursor.close
        if len(cc)>0 and cc[0][0] is None:
            aux['cc'] = cc[0][0]
        else:
            aux['cc'] = 0

        mycursor = db.cursor()
        mycursor.execute("select g.width*10000 as cb\
        from genericcityobject as g, buildinginstallation as b, wallsurface as w\
        where st_3dintersects(b.geom,g.geom) and g.citygml_class='crack' and b.gml_name='beam'
and w.gml_name = %s and st_3dintersects(b.geom,w.geom)",(wall))
        cb = mycursor.fetchall()
        mycursor.close
        if len(cb)>0 and cb[0][0] is None:
            aux['cb'] = cb[0][0]
        else:
            aux['cb'] = 0

        mycursor = db.cursor()
        mycursor.execute("select count(*) as ds\
        from genericcityobject as g, wallsurface as w, buildinginstallation as b\
        where (ST_3DDWithin(g.geom,w.geom, 0.10) and w.role=true and w.gml_name = %s) or
(ST_3DDWithin(g.geom,b.geom,0.10) and b.gml_name='beam' and st_3dintersects(b.geom,w.geom) and
w.gml_name = %s) or (ST_3DDWithin(g.geom,b.geom,0.10) and b.gml_name='column' and
st_3dintersects(b.geom,w.geom) and w.gml_name = %s)",(wall,wall,wall))
        ds = mycursor.fetchall()
        mycursor.close
        if len(ds)>0 and ds[0][0] is None:
            aux['ds'] = ds[0][0]
        else:
            aux['ds'] = 0
```

```
        mycursor = db.cursor()
        mycursor.execute("select w.tilt as tw\
        from wallsurface as w\
        where w.role= true and w.gml_name = %s",(wall))
        tw = mycursor.fetchall()
        mycursor.close
        if len(tw)>0 and tw[0][0] is None:
            aux['tw'] = tw[0][0]
        else:
            aux['tw'] = 0


        mycursor = db.cursor()
        mycursor.execute("select w.tilt as tnw\
        from wallsurface as w\
        where w.role= false and w.gml_name = %s",(wall))
        tnw = mycursor.fetchall()
        mycursor.close
        if len(tnw)>0 and tnw[0][0] is None:
            aux['tnw'] = tnw[0][0]
        else:
            aux['tnw'] = 0

        mycursor = db.cursor()
        mycursor.execute("select b.tilt as tc\
        from buildinginstallation as b, wallsurface as w\
        where b.gml_name='column' and st_3dintersects(b.geom,w.geom) and w.gml_name =
%s",(wall))
        tc = mycursor.fetchall()
        mycursor.close
        if len(tc)>0 and tc[0][0] is None:
            aux['tc'] = tc[0][0]
        else:
            aux['tc'] = 0


        #after all queries append aux
        wallsurfaces.append(aux)
else:
    with open(file_name, 'rt') as csvfile:
        reader = csv.reader(csvfile, delimiter=',')
        wallsurfaces = []
        roof = []
        firstline = True
        for row in reader:
            if firstline:
                firstline = False
                headers = row
                continue
            aux = {}
            for i,val in enumerate(row):
                if headers[i] == 'wall_name':
                    aux[headers[i]] = row[i]
                else:
                    aux[headers[i]] = float(row[i])
            if row[0] == 'roof':
                roof.append(aux)
            else:
                wallsurfaces.append(aux)

#damages hold the value of level_damage for each iteration
facade_damage = []


#loops in each wallsurface
for wallsurface in wallsurfaces:
    print (wallsurface['wall_name'])
    print(wallsurface)
    nwf_low_level = fuzz.interp_membership(array, nwf_low, wallsurface['nwf'])
    nwf_moderate_level = fuzz.interp_membership(array, nwf_moderate, wallsurface['nwf'])
    nwf_high_level = fuzz.interp_membership(array, nwf_high, wallsurface['nwf'])

    cs_low_level = fuzz.interp_membership(array,cs_low,wallsurface['cs'])
    cs_high_level = fuzz.interp_membership(array,cs_high,wallsurface['cs'])
```

```
    bs_low_level = fuzz.interp_membership(array,bs_low,wallsurface['bs'])
    bs_high_level = fuzz.interp_membership(array,bs_high,wallsurface['bs'])

    tnw_low_level= fuzz.interp_membership(array, tnw_low,wallsurface['tnw'])
    tnw_high_level= fuzz.interp_membership(array, tnw_high,wallsurface['tnw'])

    tc_low_level=fuzz.interp_membership(array,tc_low,wallsurface['tc'])
    tc_high_level=fuzz.interp_membership(array,tc_high,wallsurface['tc'])

    tw_low_level = fuzz.interp_membership(array, tw_low,wallsurface['tw'])
    tw_high_level = fuzz.interp_membership (array,tw_high,wallsurface['tw'])

    wf_low_level = fuzz.interp_membership(array,wf_low,wallsurface['wf'])
    wf_moderate_level = fuzz.interp_membership(array, wf_moderate, wallsurface['wf'])
    wf_high_level = fuzz.interp_membership(array, wf_high, wallsurface['wf'])

    ws_low_level = fuzz.interp_membership(array, ws_low, wallsurface['ws'])
    ws_high_level = fuzz.interp_membership(array, ws_high, wallsurface['ws'])

    l1nw_low_level = fuzz.interp_membership(array, l1nw_low, wallsurface['l1nw'])
    l1nw_moderate_level = fuzz.interp_membership(array, l1nw_moderate, wallsurface['l1nw'])
    l1nw_high_level = fuzz.interp_membership(array, l1nw_high, wallsurface['l1nw'])

    l2nw_low_level = fuzz.interp_membership(array, l2nw_low, wallsurface['l2nw'])
    l2nw_moderate_level = fuzz.interp_membership(array, l2nw_moderate, wallsurface['l2nw'])
    l2nw_high_level = fuzz.interp_membership(array, l2nw_high, wallsurface['l2nw'])

    l3nw_low_level = fuzz.interp_membership(array, l3nw_low, wallsurface['l3nw'])
    l3nw_moderate_level = fuzz.interp_membership(array, l3nw_moderate, wallsurface['l3nw'])
    l3nw_high_level = fuzz.interp_membership(array, l3nw_high, wallsurface['l3nw'])

    l1w_low_level = fuzz.interp_membership(array, l1w_low, wallsurface['l1w'])
    l1w_moderate_level = fuzz.interp_membership(array, l1w_moderate, wallsurface['l1w'])
    l1w_high_level = fuzz.interp_membership(array, l1w_high, wallsurface['l1w'])

    l2w_low_level = fuzz.interp_membership(array, l2w_low, wallsurface['l2w'])
    l2w_moderate_level = fuzz.interp_membership(array, l2w_moderate, wallsurface['l2w'])
    l2w_high_level = fuzz.interp_membership(array, l2w_high, wallsurface['l2w'])

    l3w_low_level = fuzz.interp_membership(array, l3w_low, wallsurface['l3w'])
    l3w_moderate_level = fuzz.interp_membership(array, l3w_moderate, wallsurface['l3w'])
    l3w_high_level = fuzz.interp_membership(array, l3w_high, wallsurface['l3w'])

    cc_low_level = fuzz.interp_membership(array, cc_low, wallsurface['cc'])
    cc_moderate_level = fuzz.interp_membership(array, cc_moderate, wallsurface['cc'])
    cc_high_level = fuzz.interp_membership(array, cc_high, wallsurface['cc'])

    cb_low_level = fuzz.interp_membership(array, cb_low, wallsurface['cb'])
    cb_moderate_level = fuzz.interp_membership(array, cb_moderate, wallsurface['cb'])
    cb_high_level = fuzz.interp_membership(array, cb_high, wallsurface['cb'])

    ds_low_level = fuzz.interp_membership(array, ds_low, wallsurface['ds'])
    ds_high_level = fuzz.interp_membership(array, ds_high, wallsurface['ds'])

    ## Now we take our rules and apply them based on Mamdani. Rule 5: Level of Damage is 5 If 
wall failure is high OR Roof failure is high Or ....
    # The OR operator means we take the maximum of these two.
    active_rule5 =max([wf_high_level,tc_high_level,tw_high_level])
    # Now we apply this by clipping the top off the corresponding output
    # membership function with `np.fmin`
    level_five_damage = np.fmin(active_rule5, lwd_five)  # removed entirely to 0

    #Rule 1: Level of Damage is 1 If: non-structural wall failure is low OR wall spaling is 
low OR level one of crack on non-structural wall is low.
    active_rule1 =max([nwf_low_level,ws_low_level,l1nw_low_level])

    level_one_damage = np.fmin(active_rule1, lwd_one)

    #Rule 2: Level of Damage is 2 If level two of crack structural wall is low OR level one of 
crack on structural wall is moderate OR level one of crack on
    #strucrtural wall is low OR level three of non-structural wall is low Or level two of non-
structural wall is moderate OR level 2 of non-structural wall is low
    #OR level one of non-structural wall is moderate.
active_rule2=max([l2w_low_level,l1w_moderate_level,l1w_low_level,l3nw_low_level,l2nw_moderate_
```

```
level,l2nw_low_level,l1nw_moderate_level,nwf_moderate_level,ws_high_level])
    level_two_damage = np.fmin(active_rule2, lwd_two)

    #Rule 3: Level of Damage is 3 if :
    active_rule3
=max([l1nw_high_level,l2nw_high_level,l3nw_moderate_level,l3nw_high_level,l1w_high_level,l2w_m
oderate_level,l2w_high_level,l3w_low_level,l3w_moderate_level,cc_low_level,cc_moderate_level,c
b_low_level,cb_moderate_level,ds_low_level,cs_low_level,bs_low_level,tnw_low_level,wf_low_leve
l])
    level_three_damage = np.fmin(active_rule3, lwd_three)

    #Rule 4:
    active_rule4 =
max([ds_high_level,cb_high_level,cc_high_level,l3w_high_level,tc_low_level,tnw_high_level,tw_l
ow_level,wf_moderate_level,cs_high_level,bs_high_level])
    level_four_damage = np.fmin(active_rule4, lwd_four)
    level_wall_damage0 = np.zeros_like(array)


    # Visualize this
    fig, ax0 = plt.subplots(figsize=(8, 5))
    ax0.fill_between(array, level_wall_damage0, level_one_damage, facecolor='b', alpha=0.7)
    ax0.plot(array, level_one_damage, 'b', linewidth=0.5, linestyle='--')
    ax0.fill_between(array, level_wall_damage0, level_two_damage, facecolor='g', alpha=0.7)
    ax0.plot(array, level_two_damage, 'g', linewidth=0.5, linestyle='--')
    ax0.fill_between(array, level_wall_damage0, level_three_damage, facecolor='m', alpha=0.7)
    ax0.plot(array, level_three_damage, 'm', linewidth=0.5, linestyle='--')
    ax0.fill_between(array, level_wall_damage0, level_four_damage, facecolor='c', alpha=0.7)
    ax0.plot(array, level_four_damage, 'c', linewidth=0.5, linestyle='--')
    ax0.fill_between(array, level_wall_damage0, level_five_damage, facecolor='r', alpha=0.7)
    ax0.plot(array, level_five_damage, 'r', linewidth=0.5, linestyle='--')
    ax0.set_title('Damage membership in '+wallsurface['wall_name'])
    # Turn off top/right axes
    for ax in (ax0,):
        ax.spines['top'].set_visible(False)
        ax.spines['right'].set_visible(False)
        ax.get_xaxis().tick_bottom()
        ax.get_yaxis().tick_left()
    plt.tight_layout()
    one= mpatches.Patch(color='b', label='level one')
    two = mpatches.Patch(color='g', label='level two')
    three = mpatches.Patch(color='m', label='level three')
    four = mpatches.Patch(color='c', label='level four')
    five = mpatches.Patch(color='r', label='level five')
    plt.legend(handles=[one,two,three,four,five])
    plt.show()
    # Aggregate all five output membership functions together
    aggregated = np.fmax(level_one_damage,
                         np.fmax(level_two_damage,
                                 np.fmax(level_three_damage,
                                         np.fmax(level_four_damage,level_five_damage))))
    # Calculate defuzzified result
    try:
        level_damage = fuzz.defuzz(array,aggregated, 'centroid')
        level_damage_activation = fuzz.interp_membership(array, aggregated, level_damage)   #
for plot
        # Visualize this
   fig, ax0 = plt.subplots(figsize=(8, 5))
   ax0.plot(array, level_one_damage, 'b', linewidth=0.5, linestyle='--',label='levelone')
   ax0.plot(array, level_two_damage, 'g', linewidth=0.5, linestyle='--',label='level two')
   ax0.plot(array, level_three_damage, 'm', linewidth=0.5, linestyle='--',label='level three’)
   ax0.plot(array, level_four_damage, 'c', linewidth=0.5, linestyle='--',label='level four')
   ax0.plot(array, level_five_damage, 'r', linewidth=0.5, linestyle='--',label='level five')
   ax0.fill_between(array, level_wall_damage0, aggregated, facecolor='Orange', alpha=0.7)
   ax0.plot([level_damage, level_damage], [0, level_damage_activation], 'k', linewidth=1.5,
alpha=0.9)
   ax0.set_title('Aggregated membership and result of '+wallsurface['wall_name'])
   int_level_damage=int(level_damage)
   print(int_level_damage,"%")
   # Turn off top/right axes
   for ax in (ax0,):
            ax.spines['top'].set_visible(False)
            ax.spines['right'].set_visible(False)
            ax.get_xaxis().tick_bottom()
            ax.get_yaxis().tick_left()
```

```
    plt.tight_layout()
    plt.legend(loc=0,fontsize='medium')
    plt.show()
 except:
     level_damage = 0
    facade_damage.append(level_damage)

# put the result of "level_dame" of each facade in a list as input for second fuzzification.
#Generate universe function of second fuzzification
#for i in the list
level_wall_damage_final = np.arange(0,100,1)
level_building_damage = np.arange(0,100,1)
# Membership functions for wall_damage final value
lwdf_one= fuzz.trapmf(array, [1,1,10,20])
lwdf_two = fuzz.trimf(array, [10,20,35])
lwdf_three = fuzz.trimf(array, [20,35,50])
lwdf_four = fuzz.trimf(array, [35,50,75])
lwdf_five = fuzz.trapmf(array, [50, 75, 100,100])

# Membership functions for level_building_damage
lbd_one = fuzz.trapmf(array, [1,1 ,10, 20])
lbd_two = fuzz.trimf(array, [10, 20, 35])
lbd_three = fuzz.trimf(array, [20, 35, 50])
lbd_four = fuzz.trimf(array, [35,50,75])
lbd_five = fuzz.trapmf(array, [50, 75, 100,100])

#active membership function
#need a loop to call the number from list
#list of level_damage F
#first facade
lwdf = []
for damage in facade_damage:
    aux = {}
    aux['lwdf_one_level']= fuzz.interp_membership(array, lwdf_one,damage)
    aux['lwdf_two_level']= fuzz.interp_membership(array, lwdf_two,damage)
    aux['lwdf_three_level']= fuzz.interp_membership(array, lwdf_three, damage)
    aux['lwdf_four_level']= fuzz.interp_membership(array, lwdf_four, damage)
    aux['lwdf_five_level']= fuzz.interp_membership(array, lwdf_five, damage)
    lwdf.append(aux)
#roof
rf_low_level = fuzz.interp_membership(array, rf_low, roof[0]['rf'])
rf_moderate_level = fuzz.interp_membership(array, rf_moderate, roof[0]['rf'])
rf_high_level = fuzz.interp_membership(array, rf_high, roof[0]['rf'])
#rule1
active_rule1_final = lwdf[0]['lwdf_one_level']
for facade in lwdf:
    active_rule1_final = np.fmax(facade['lwdf_one_level'], active_rule1_final)
level_one_building_damage = np.fmin(active_rule1_final, lbd_one)
#rule2
active_rule2_final = lwdf[0]['lwdf_two_level']
for facade in lwdf:
    active_rule2_final = np.fmax(facade['lwdf_two_level'], active_rule2_final)
level_two_building_damage = np.fmin(active_rule2_final, lbd_two)
#rule3
active_rule3_final = rf_low_level
for facade in lwdf:
    active_rule3_final = np.fmax(facade['lwdf_three_level'], active_rule3_final)
level_three_building_damage = np.fmin(active_rule3_final, lbd_three)
#rule4
active_rule4_final = rf_moderate_level
for facade in lwdf:
    active_rule4_final = np.fmax(facade['lwdf_four_level'], active_rule4_final)
level_four_building_damage = np.fmin(active_rule4_final, lbd_four)
#rule5
active_rule5_final = rf_high_level
for facade in lwdf:
    active_rule5_final = np.fmax(facade['lwdf_five_level'], active_rule5_final)
level_five_building_damage = np.fmin(active_rule5_final, lbd_five)
level_building_damage0 = np.zeros_like(array)
# Visualize this
fig, ax0 = plt.subplots(figsize=(8, 5))
ax0.fill_between(array, level_building_damage0, level_one_building_damage, facecolor='b',
alpha=0.7)
ax0.plot(array, level_one_building_damage, 'b', linewidth=0.5, linestyle='--', )
ax0.fill_between(array, level_building_damage0, level_two_building_damage, facecolor='g',
```

```
alpha=0.7)
ax0.plot(array, level_two_building_damage, 'g', linewidth=0.5, linestyle='--')
ax0.fill_between(array, level_building_damage0, level_three_building_damage, facecolor='m',
alpha=0.7)
ax0.plot(array, level_three_building_damage, 'm', linewidth=0.5, linestyle='--')
ax0.fill_between(array, level_building_damage0, level_four_building_damage, facecolor='c',
alpha=0.7)
ax0.plot(array, level_four_building_damage, 'c', linewidth=0.5, linestyle='--')
ax0.fill_between(array, level_building_damage0, level_five_building_damage, facecolor='r',
alpha=0.7)
ax0.plot(array, level_five_building_damage, 'r', linewidth=0.5, linestyle='--')
ax0.set_title('Damage membership of building')

# Turn off top/right axes
for ax in (ax0,):
    ax.spines['top'].set_visible(False)
    ax.spines['right'].set_visible(False)
    ax.get_xaxis().tick_bottom()
    ax.get_yaxis().tick_left()
plt.tight_layout()

one= mpatches.Patch(color='b', label='level one')
two = mpatches.Patch(color='g', label='level two')
three = mpatches.Patch(color='m', label='level three')
four = mpatches.Patch(color='c', label='level four')
five = mpatches.Patch(color='r', label='level five')
plt.legend(handles=[one,two,three,four,five])
plt.show()

# Aggregate all five output membership functions together
aggregated = np.fmax(level_one_building_damage,
                         np.fmax(level_two_building_damage,
                                 np.fmax(level_three_building_damage,
np.fmax(level_four_building_damage,level_five_building_damage))))
# Calculate defuzzified result
try:
    level_building_damage_final = fuzz.defuzz(array,aggregated, 'centroid')
    level_building_damage_final_activation = fuzz.interp_membership(array, aggregated,
level_building_damage_final)  # for plot
    # Visualize this
    fig, ax0 = plt.subplots(figsize=(8, 5))
    ax0.plot(array, level_one_building_damage, 'b', linewidth=0.5, linestyle='--',label='level
one' )
    ax0.plot(array, level_two_building_damage, 'g', linewidth=0.5, linestyle='--',label='level
two')
    ax0.plot(array, level_three_building_damage, 'm', linewidth=0.5, linestyle='--
',label='level three')
    ax0.plot(array, level_four_building_damage, 'c', linewidth=0.5, linestyle='--
',label='level four')
    ax0.plot(array, level_five_building_damage, 'r', linewidth=0.5, linestyle='--
',label='level five')
    ax0.fill_between(array, level_building_damage0, aggregated, facecolor='Orange', alpha=0.7)
    ax0.plot([level_building_damage_final, level_building_damage_final], [0,
level_building_damage_final_activation], 'k', linewidth=1.5, alpha=0.9)
    ax0.set_title('Aggregated membership and result of final damage of building ')
    int_level_building_damage_final=int(level_building_damage_final)
    print(int_level_building_damage_final,"%")
    # Turn off top/right axes
    for ax in (ax0,):
        ax.spines['top'].set_visible(False)
        ax.spines['right'].set_visible(False)
        ax.get_xaxis().tick_bottom()
        ax.get_yaxis().tick_left()

    plt.tight_layout()
    plt.legend(loc=0,fontsize='medium')
    plt.show()
except:
    int_level_building_damage_final = 0;
    print(int_level_building_damage_final,"%")
    print('The building is intact')
```