Ontological and Machine Learning Approaches for Inspection of Facilities Using BIM

Fardin Bahreini

A Thesis
In the Department
of
Building, Civil, and Environmental Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy (Building Engineering) at
Concordia University
Montreal, Quebec, Canada

July 2022

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: *Fardin Bahreini*

Entitled: *Ontological and Machine Learning Approaches for Inspection of Facilities Using BIM*

and submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (*Building Engineering*)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

____Chair
Dr. Rolf Wuthrich

____External Examiner
Dr. Jeff Rankin

____External to Program
Dr. Farnoosh Naderkhani

____Examiner
Dr. Osama Moselhi

____Examiner
Dr. Sang Hyeok Han

____Thesis Supervisor
Dr. Amin Hammad

Approved by ____
Dr. Mazdak Nik-Bakht, Graduate Program Director

August/30/2022____
Dr. Mourad Debbabi, Dean, Gina Cody School of Engineering and Computer Science

ABSTRACT

Ontological and Machine Learning Approaches for Inspection of Facilities Using BIM

Fardin Bahreini, Ph.D.

Concordia University, 2022

Facilities should be kept in good condition throughout their lifecycle by rigorous inspection processes. The semantic relationships between multiple inspection information during lifecycle phases, and inspection result representation are among the most critical issues that need to be addressed. So far, many studies have been done to identify, analyze, repair, and prevent defects. However, after capturing the defect information, there is a need for an ontology to organize and integrate relevant information and future actions. Additionally, the availability of inspection robots in buildings' construction and operation phases has led to expanding the scope of applications and increasing technological challenges. BIM models comprise useful information about the building environment's representation, which can help the inspection robot overcome task complexity. However, the research in this area is still limited and fragmented, and there is a need to develop an integrated ontology to be used as a knowledge model for logic-based inspection of building defects.

Moreover, visual inspection using non-equipped eyes is the principal method of detecting structural surface defects, which is unsafe, time-consuming, expensive, and subjective to human errors. Using remote sensing, such as, cameras and LiDAR scanners, is one solution to overcome these shortcomings. The captured point cloud data from the real environment can assist in detecting the defects and taking further actions. Recently, machine learning methods attracted the attention of researchers for semantic segmentation and classification based on point clouds. However, no deep learning method is currently available for semantic segmentation of concrete surface defects based on raw point cloud data. Furthermore, the BIM model needs to be integrated with the results of defect semantic segmentation after the LiDAR-based inspection.

Addressing the above issues, this research has the following objectives: (1) Developing an ontology for concrete surface defects; (2) Developing BIM-based ontology to cover the different types of information and concepts related to robotic navigation and inspection tasks; (3) Developing a method for point cloud-based concrete surface defects semantic segmentation; and (4) Developing a semi-automated process for as-inspected modeling.

The first part of this research focused on the development of an ontology, called Ontology for Concrete Surface Defects (OCSD), to have a unified knowledge model where all the stakeholders can access information in a systematic manner. OCSD metrics include 333 classes, 51 relations, 27 attributes, and 31 individuals. OCSD comprises high-level knowledge of the concepts and relationships related to surface defects, inspection, diagnosis, and 3R (Repair, Rehabilitation, and Replacement) processes. The application of OCSD was investigated in a case study and a survey was designed to evaluate the semantic representation of OCSD. Based on the evaluation, OCSD was able to provide a clear understanding of the concepts and relationships in the domain, and it can help future asset management systems benefit from the provided knowledge.

The second part of this research focused on the development of an integrated ontology, called Ontology for BIM-based Robotic Navigation and Inspection Tasks (OBRNIT), to extend BIM

applications for robotic navigation and inspection tasks. OBRNIT metrics include 386 classes, 45 relations, 52 attributes, and 8 individuals. OBRNIT comprises high-level knowledge of the concepts and relationships related to buildings, robots, and navigation and inspection tasks. BIM is considered as a reference that is integrated with the knowledge model. The semantic representation of OBRNIT was evaluated through a case study and a survey. The evaluation demonstrates that OBRNIT covers the domain's concepts and relationships up to the point that satisfies the domain experts. Based on the evaluation, OBRNIT was able to give a clear understanding of the concepts and relationships in the domain, and it can help in the future in developing robotic inspection systems.

The last part of this research focused on a method for point cloud-based defect semantic segmentation based on Normal Vector Enhanced Dynamic Graph Convolutional Neural Network (NVE-DGCNN) to automate the inspection process of concrete surface defects, including cracks and spalls. This part investigates two main characteristics related to surface defects, including the normal vector and depth. The network's performance is improved by modifying the network and augmenting the dataset. Sensitivity analysis is applied to capture the best combination of hyperparameters and investigate their effects on the network performance. NVE-DGCNN resulted in 98.56% and 96.50% recalls for semantic segmentation of cracks and spalls, respectively. Furthermore, post-processing of the results of the defects semantic segmentation is done to semi-automate the process of as-inspected modeling. This semi-automated process made it possible to manage and visualize the detected defects by extracting their dimensions and identifying the conditions on the 3D model.

ACNOWLEDGEMENT

I would like to thank my supervisor, Dr. Amin Hammad, for his intellectual and personal support, encouragement, patient guidance, and advice he has provided during my studies. His insightful guidance and suggestions were the most valuable help for me, which softened the difficulties of my research study.

Also, I would like to thank the members of my thesis committee, Dr. Jeff Rankin, Dr. Osama Moselhi, Dr. Sang Hyeok Han, and Dr. Farnoosh Naderkhani for their valuable inputs and spending their precious time reading this thesis.

I must express my gratitude to my lovely wife, Saeede, for her continued support and encouragement. I would like to give my special thanks to my beloved parents, Fariba and Esmaeil, and my dear sister, Farnaz, for their continuous and unparalleled love, help, and support.

Finally, I would like to thank my research colleagues for their kind support. Data collection and manual annotation, as well as the implementation of PointNet++ were done in collaboration with my colleagues, Mr. Majid Nasrollahi, and Ms. Neshat Bolourian.

Dedicated to my lovely parents, Fariba and Esmaeil, my lovely wife, Saeede, and my dear sister Farnaz.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
3R	Repair, Rehabilitation, and Replacement
AEC/FM	Architecture, Engineering, Construction, and Facilities Management
API	Application Programming Interface
AUV	Autonomous Underwater Vehicles
BEO	Building Element Ontology
BIM	Building Information Modeling
BIMDO	BIM Design Ontology
BIMSO	BIM Shared Ontology
BMS	Building Management System
BOT	Building Topology Ontology
BrIM	Bridge Information Model
CDO	Concrete Damage Ontology
CIM	Civil Infrastructure Information Modelling
CNN	Convolutional Neural Network
CPL	CRAM Plan Language
CRAM	Cognitive Robot Abstract Machine
DGCNN	Dynamic Graph Convolutional Neural Network
DNN	Deep Neural Network
DOF	Degrees of Freedom
DOT	Damage Topology Ontology
FOG	Ontology for Geometry Formats
FOL	First Order Logic
GPS	Global Positioning System
HVAC	Heating, Ventilation, and Air Conditioning
HT	Hough Transform
IDEF	Integrated Definition
IFC	Industry Foundation Classes
IFCXML	IFC Extensible Markup Language
IoU	Intersection over Union
IRIM	Inspection and Repair Information Modeling
ISO	International Organization for Standardization
KIF	Knowledge Interchange Format
LiDAR	Light Detection and Ranging
LOAM	Lidar Odometry and Mapping
LOD	Level of Detail
MAS	Multi-Agent System
MEP	Mechanical, Electrical, and Plumbing
MLP	Multi-Layer Perceptron
MVD	Model View Definition
NVE-DGCNN	Normal Vector Enhanced Dynamic Graph Convolutional Neural Network
OBRNIT	Ontology for BIM-based Robotic Navigation and Inspection Tasks
OCSD	Ontology for Concrete Surface Defects

OMG	Ontology for Managing Geometry
OPM	Ontology for Property Management
OSIM	Ontario Structure Inspection Manual
OWL	Web Ontology Language
RANSAC	Random Sample Consensus
RDF	Resource Description Framework
RFID	Radio Frequency Identification
RGB	Red, Green, and Blue
ROS	Robot Operating System
SLAM	Simultaneous Localization and Mapping
TCEI	Time, Cost, and Environmental Impacts
TLS	Terrestrial Laser Scanning
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
UML	Unified Modeling Language
URI	Uniform Resource Identifier
XML	Extensible Markup Language

CHAPTER 1. INTRODUCTION

1.1 Background

Recent studies about Building Information Modeling (BIM) have demonstrated a strong potential for extending BIM applications to the operation and maintenance phase of facilities. On the other hand, ontology is a knowledge model that can help clarify and systematize implied knowledge in a way that is simple and logically understood by the user [1]. Ontology can create a unified knowledge model to exchange information among the construction industry and standardize the process and framework of using BIM for facilities management. Besides, inspection robots are used to automate inspection during the construction and operation phases. Advanced technologies (e.g. scanners, sensors) have made the inspection process more accurate and reliable [2]. Moreover, machine learning methods can be used for defect semantic segmentation using point clouds collected by a Light Detection and Ranging (LiDAR) scanner [3]. The detected surface defects information can be linked to the BIM model. Therefore, the inspection process can be more efficient by utilizing BIM-based inspection-related knowledge, and an integrated process of surface defect semantic segmentation and defect modeling.

1.2 Problem Statement and Research Gaps

The key problems of this research can be attributed to four main issues as follows:

(1) The first issue is regarding the inspection, diagnosis, and repair of concrete surface defects. Facilities should be kept in good condition throughout their lifecycle by rigorous inspection processes. The considerable amount of data resulting from inspection should be managed in an efficient manner to avoid errors, reduce cost, and make the best use of available resources. The semantic relationships between multiple inspection information during lifecycle phases, and inspection result representation are among the most critical issues that need to be addressed [4].

So far, many studies have been done to identify, analyze, repair, and prevent defects. However, after capturing the defect information, the need for an ontology to organize and integrate relevant information and future actions still requires further studies and development. There is a need to streamline the research to reduce duplications in efforts and provide a high-level approach to model the knowledge related to the inspection, diagnosis, and repair of specific types of defects.

(2) The second issue is regarding the challenges of robotic inspection in buildings' construction and operation phases. The complexity of the interactions with the surrounding building environment is the main challenge for inspection robots [5]. BIM models comprise useful information about the building environment's representation, which can help the inspection robot overcome task complexity. Moreover, an ontology can be used to overcome this challenge as a basis for the robot's task planning and execution. Therefore, a variety of knowledge, including the robot's low-level data related to perception and high-level data about the environment, objects, and tasks, needs to be integrated [6]. The robotic inspection must be performed in such a way that the process considers reliability, repeatability, and safety. Therefore, it is necessary to enhance operational consistency in the inspection environment [7]. Robotic inspection systems' capabilities have progressed over time, and these systems have become dependent on multiple components with diverse functions. In most developed systems, the modules are created independently by different individuals with different technical expertise. Thus, a clear definition of the relationships between the system's various components is needed. The system's structure and related

components must have a straightforward design and documentation to solve this problem [8]. A clear and accurate description of the environment and the task can help autonomous robotic inspection [9]. The robot declarative knowledge represents the task's objects, properties, and relationships in a semantic model [10]. The robot can use this declarative knowledge to perform the task more accurately. However, the research in this area is still limited and fragmented, and there is a need to develop an integrated ontology to be used as a knowledge model for logic-based inspection of building defects.

(3) The third issue is regarding to the automated concrete surface semantic segmentation of point cloud data. Visual inspection using non-equipped eyes is the principal method of detecting structural surface defects, which is unsafe, time-consuming, expensive, and subjective to human errors [11]. Using remote sensing such as cameras and LiDAR scanners, is one solution to overcome these shortcomings. The captured point cloud data from the real environment can assist in detecting the defects and taking further actions. It will also provide a database for other maintenance measures after creating an integrated 3D model for existing structures. Image-based inspection using cameras is based on pattern recognition techniques [12]. Several studies focused on detecting defects automatically (e.g. spalling [13] or cracks [14]) and determining some characteristics such as the width of cracks [15, 16]. Several crack detection algorithms have been developed, which can be practically used for real-time crack analysis [17, 18], crack classification [19], and automating crack sealing [20, 21]. There are several challenges in supporting concrete inspection using image-based methods. Such methods are mostly defined for simple flat concrete surfaces and may fail in analyzing more complex geometries and materials [20]. Good lighting condition is one of the main issues that should be considered during implementing these methods [22]. Another shortcoming is the necessity of providing supplementary information, such as camera lens, focal length, or the distance from the camera to the target surface, before analyzing the images [23]. Moreover, in comparison to image data, measuring the defects dimensions such as depth is more accurate and reliable in point cloud-based methods. Although the initial cost of LiDAR scanner is more than cameras, it may be more profitable and economical in the long term.

In order to automate the process of point cloud-based inspection, appropriate datasets and an efficient approach such as defect semantic segmentation are essential. Recently, machine learning methods attracted the attention of researchers for semantic segmentation and classification based on point clouds. Unlike other methods, such as the Hough Transform (HT) [24] and the Random Sample Consensus (RANSAC) approach [25], machine learning methods are robust and flexible. However, they rely on the point cloud density and size of the dataset. Moreover, training based on large datasets is time-consuming [26] and converting the point cloud into other representations increases the dataset size. Different methods such as classification, part segmentation, and semantic segmentation can be used to process the raw point cloud data [27]. This research focuses on semantic segmentation, which is based on the detailed information of each point. Although much work has been done for processing visual information with images, research on machine learning methods for semantic segmentation of raw point cloud data is still in its early stages [28]. Moreover, no deep learning method is currently available for semantic segmentation of the surface defects based on point clouds without converting the raw data to other representations (e.g. images).

(4) The fourth issue is related to as-inspected modeling. Most existing structures do not have a 3D model; and even when available, it is not a complete model. LiDAR technology is commonly used

to create as-is BIM models. However, the as-is model does not include the inspection results [29]. Therefore, the BIM model needs to be integrated with the results of defect semantic segmentation after the LiDAR-based inspection.

1.3 Research Objectives

Given the problems explained in Section 1.2, the main objectives of this research are defined as follows: (1) Developing an ontology for concrete surface defects; (2) Developing BIM-based ontology to cover the different types of information and concepts related to robotic navigation and inspection; (3) Developing a method for point cloud-based concrete surface defects semantic segmentation; and (4) Developing a semi-automated process for as-inspected modeling.

1.4 Thesis Organization

The structure of the thesis is presented as follows:

Chapter 2 Literature Review: This chapter reviews the literature related the major concepts used in this research, including facilities management using BIM, robotic inspection and navigation, inspection information modeling, ontology approach, and semantic segmentation of point clouds using deep learning.

Chapter 3 Research Framework: The overview of the proposed framework is discussed briefly in this chapter.

Chapter 4 Ontology for Concrete Surface Defects: This chapter elaborates on ontological approach to develop ontology for concrete surface defects.

Chapter 5 Ontology for BIM-Based Robotic Navigation and Inspection Tasks: This chapter elaborates on an ontological approach to develop ontology for BIM-based robotic navigation and inspection tasks.

Chapter 6 Point Cloud-Based Concrete Surface Defect Semantic Segmentation and As-Inspected Modeling: This chapter proposes a method for point cloud-based surface defect semantic segmentation using Normal Vector Enhanced Dynamic Graph Convolutional Neural Network (NVE-DGCNN). Moreover, this chapter proposes a semi-automated approach for the as-inspected modeling based on the results of semantic segmentation to integrate the defects with the BIM Model.

Chapter 7 Summary, Conclusions, and Future Work: This chapter summarizes the present research work, highlights its contributions, investigates the limitations, and suggests recommendations for future research.

CHAPTER 2. LITERATURE REVIEW

2.1 Introduction

In this chapter, literature related to major concepts including concrete infrastructure management, robotic inspection and navigation tasks, inspection information modeling, ontology approach, and semantic segmentation of point clouds using deep learning, is reviewed. First, a review is provided of concrete infrastructure management and different types of concrete surface defects. In addition to reviewing inspection tasks during the construction and operation phases, the literature related to concrete surface inspection, diagnosis, and repair processes, and robotic inspection and navigation-related tasks is reviewed. Then, research about using robots for inspection and IFC-based navigation is reviewed. Furthermore, the inspection information modeling literature and research gaps are reviewed. The ontology approach section explains ontology languages and tools, architecture, engineering and construction, and facilities management (AEC/FM) ontologies, and ontologies for robots. Finally, the machine learning approach for semantic segmentation of point clouds and local feature learning on point sets are reviewed.

2.2 Concrete Infrastructure Management

Regular inspection and appropriate Repair, Rehabilitation, and Replacement (3R) works are crucial for continued operations of infrastructure systems [30- 34].

2.2.1 Types of Defects

Gheitasi and Harris [35] evaluated the effect of subsurface delamination of the reinforced concrete deck. Deterioration of concrete elements often occurs in steel bars as corrosion and section loss due to leakage from expansion joints adjacent to supports. Cracks in concrete are caused by dead and live loads, stresses due to temperature changes, and shrinkage. Each of these cracks can provide a space for the penetration of chloride, moisture, or salt, resulting in the formation of new defects. Figure 2-1 includes possible damage because of rebar corrosion (i.e. longitudinal cracks, spalling, delamination).

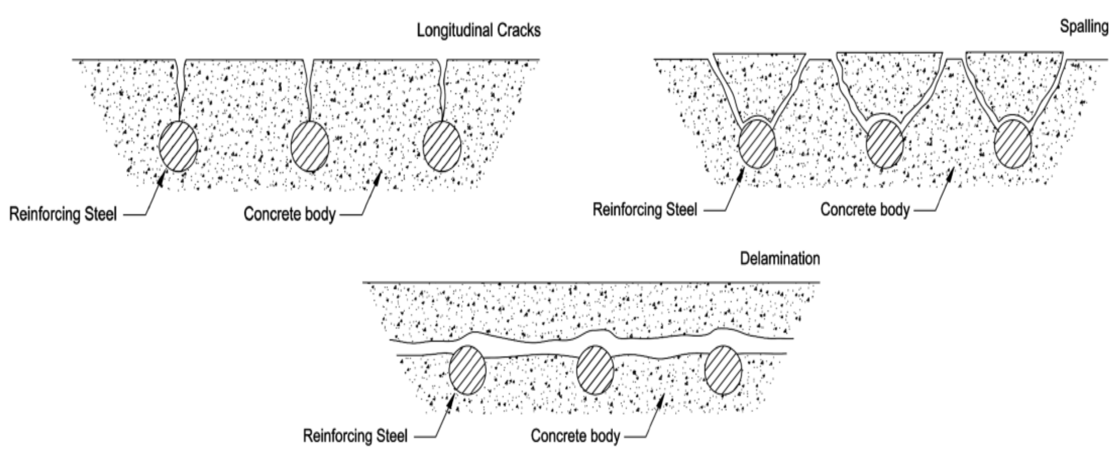


Figure 2-1. Possible damage because of rebar corrosion [35]

Nielsen et al. [36] presented a framework to gather the concrete bridge defects data, investigate defect severity, and recommend a prioritized repair portfolio for critical components. Le and Andrews [37] classified the various components of the bridge according to the component type and material, and defined the extent of repair actions (e.g. repair or replacement) and related condition states based on the severity and extent of defects. Brandon et al. [38] asserted that bridge failure can be the result of issues in one of the following processes: design, construction, operation, or inspection.

In some cases, the defect caused by a damage mechanism accelerates the process of other defect formation and leads to further damages. Therefore, the relationship between defects and the condition of the defected element requires further studies. For example, the acid reaction causes damage to the protective layer, and consequently, the hidden layer is exposed to the outside environment, which itself causes chloride penetration and corrosion. The aging process of concrete materials and various aggressive agents are among the factors that cause the damage of concrete over time [39]. The most common types of surface defects in concrete based on Concrete Structures Protection, Repair and Rehabilitation book [40] and Ontario structure inspection manual (OSIM) [41] are: (1) crack, (2) spalling, (3) delamination, (3) scaling, (4) disintegration, (5) erosion, (6) honeycombing, (7) pop-outs, (8) cold joints, (9) stratification, (10) segregation, (11) efflorescence, (12) exudation, (13) incrustation, (14) stalactite, (15) abrasion/wear, (16) slippery surface, and (17) stain. Table 2-1 shows the types of concrete defects based on OSIM.

Crack defects are characterized by a partial or complete linear fracture on the concrete surface. An oriented crack is a type of cracks in the concrete surface that usually has a particular slope and direction. A mapped crack is a crack in the concrete surface that occurs randomly at close distances and without a fixed direction. In addition, this crack usually covers a large area [42, 43]. Spalling is a defect characterized by a significant gap due to the local separation of concrete from a larger surface. Delamination is a defect characterized by the lack of bonding of the part of the separated concrete surface that is not entirely detached from the larger surface. Scaling is a defect characterized by the loss of part of the mortar or concrete surface in the form of surface peeling. Disintegration is a defect characterized by breaking concrete into smaller sections or parts, which usually occurs if the severe type of scaling is not controlled over time. Erosion is the mechanical damage and loss of mass caused by scrubbing sand and other particles in running water on a concrete surface. Honeycombing is a void between the coarse particles of concrete. Pop-out is a cone-shaped hole on the concrete surface. Cold joints are characterized by an interconnected linear separation at the joints between pouring two sets of concrete. Segregation is a defect due to the separation of cement and different sizes of aggregates. Stratification is characterized by separation in the form of the layered and horizontal structure due to high humidity and vibration. Efflorescence is characterized by white salt deposition on the concrete surface [44]. Exudation is the release of a substance from a compound, which in concrete is characterized by the release of gel-like material from surface pores [45]. Incrustation is characterized by the appearance of crusts or the accumulation of hard coating on the concrete surface [46]. Stalactites are a chemical reaction of water and minerals in concrete characterized by the accumulation of substances hanged from the surface. Abrasion is mechanical damage caused by scratching or rubbing the surface by vehicles or a sharp foreign object on the surface. Abrasion damage combined with solid particles such as sand can cause wear defects on the concrete surface. Slippery surface is a surface defect, which is characterized by the smoothness of the concrete surface [47]. This defect is dangerous and indicates a fair condition. Therefore, it should be fixed as soon as possible. Stain has different colors, shades, and textures. The main types of stains include biological growth stains (e.g. fungi, beetle), dust stains, chemical reaction stains, corrosion stains, and water stains [48, 49].

Other types of concrete surface defects are: graffiti, bugholes, and flatness defect. Graffiti is the intentional act of painting or writing on the concrete surface [50]. Bugholes are small holes in the concrete surface that are formed by air entrapment in fresh concrete [51]. The flatness of the surface is a defect characterized by deviations in elevation and irregularity of the surface [52].

Table 2-1. Common types of concrete surface defects based on OSIM (adapted from [41])

Surface defect types	Severity (all dimensions in mm)			
Crack	Hairline (width < 0.1)	Narrow (0.1 ≤ width ≤ 0.3)	Medium (0.3 < width ≤ 1.0)	Wide (1.0 < width)
Spalling	Light (Any direction < 150 or depth < 25)	Medium (150 ≤ Any direction ≤ 300 or 25 ≤ depth ≤ 50)	Severe (300 < Any direction ≤ 600 or 50 < depth ≤ 100)	Very severe (600 < Any direction or 100 < depth)
Delamination	Light (Any direction < 150)	Medium (150 ≤ Any direction ≤ 300)	Severe (300 < Any direction ≤ 600)	Very severe (600 < Any direction)
Scaling	Light (depth ≤ 5 without exposure of coarse aggregate)	Medium (5 < depth ≤ 10 with exposure of some coarse aggregates)	Severe (10 < depth ≤ 20 with aggregate particles standing out from the concrete and a few completely lost)	Very severe (Depth > 20)
Disintegration	Light (Depth ≤ 25 with some loss of coarse aggregate)	Medium (25 < depth ≤ 50 with considerable loss of coarse aggregate and exposure of reinforcement)	Severe (50 < depth ≤ 100 with substantial loss of coarse aggregate and exposure of reinforcement over a large area)	Very severe (100 < depth and extending over a large area)
Erosion	Light (Depth ≤ 25 with some loss of coarse aggregate)	Medium (25 < depth ≤ 50 with considerable loss of coarse aggregate and exposure of reinforcement)	Severe (50 < depth ≤ 100 with substantial loss of coarse aggregate and exposure of reinforcement over a large area)	Very severe (100 < depth and extending over a large area)
Honeycombing	Light (depth ≤ 25)	Medium (25 < depth ≤ 50)	Severe (50 < depth ≤ 100)	Very severe (100 < depth)
Pop-Outs	Light (depth ≤ 25)	Medium (25 < depth ≤ 50)	Severe (50 < depth ≤ 100)	Very severe (100 < depth)
Cold joints	N.A.			
Segregation	N.A.			
Stratification	N.A.			
Efflorescence	N.A.			
Exudation	N.A.			
Incrustation	N.A.			
Stalactite	N.A.			
Abrasion/wear	N.A.			
Slippery surface	N.A.			
Stain	N.A.			

1

2.2.2 Inspection Methods

Various methods can be used to inspect concrete surfaces. Computer vision methods can be used to detect anomalies in the collected data [3]. Crack measuring is a method to measure the characteristics of a crack on the concrete surface. Concrete cover and magnetic field measuring are interrelated methods to measure concrete cover by locating the rebars. Magnetic field measuring using devices such as magnetometer to measure the magnetic field in *X*, *Y*, and *Z*-axes [53]. Moisture/humidity measuring is another measurement method to inspect the concrete surface [54]. Infrared thermography is a measurement method which uses a thermal camera to measure the temperature of inspected concrete surfaces [55]. As most humidity measurement tools depend on the measurement of some other environmental features such as temperature, devices such as thermo-hygrometer can be used to measure the temperature and humidity of the air. A rubber hammer is a percussion tool. Usually, after macroscopic inspection of the concrete surface, a rubber hammer test is performed for secondary inspection. Colorimetric test strips are a tool to identify the types of salts on the concrete surface [56]. The surface absorption test evaluates the adsorption properties by measuring the amount of water that penetrates the concrete sample [57]. Half-cell potential test can be performed to specify the chance of corrosion within the reinforcement and the concrete surface [58].

2.2.3 Causes of Defects

Maksymowicz et al. [59] proposed a basic taxonomy of mechanisms for the causes of the damages to concrete bridges and classified them into the following three main categories: chemical, physical, and biological mechanisms. Each of these mechanisms has some consequences, which can result in unique damages (e.g. losses, deformations, displacements, etc.).

Concrete surface defects can appear for a variety of reasons. The primary issues that might produce defects on concrete surfaces during the design phase are poor design and design-related errors. Some instances of poor design are as follows: (1) Concrete formwork has a significant effect on the shape of the surface, and improper design of formwork can cause defects in the concrete surface; (2) Expansion joints play an important role in controlling cracks. Thus the improper design of these joints and inappropriate selection of materials can cause defects in the concrete surface; and (3) The release agent prevents fresh concrete from sticking to the formwork and facilitates the forming process. Improper selection of the release agent can cause defects such as stains on the concrete surface [60].

The main problems during the construction phase that can cause surface defects include: (1) Using defective formwork or incorrect use of formwork during construction can cause defects in the concrete surface; (2) Inappropriate transportation is a factor that can reduce the quality of concrete and cause defects such as cracking and segregation. (3) Too much or too little water in the concrete mix can cause defects such as surface cracking [61, 62]; (4) Inappropriate casting, as well as casting in inappropriate weather conditions, can also cause defects. For example, scorching temperatures cause the concrete to harden prematurely, and very high temperatures cause the concrete to lose strength before curing, which ultimately causes surface defects [63, 64]; (5) The inappropriate placement of casting joints causes deformation and defects in the concrete surface; (6) Inappropriate compaction causes defects in the concrete surface, such as honeycomb and bughole [65]; (7) Concrete needs to be cured for a sufficient time, and at suitable humidity and temperature to achieve the desired durability. Inappropriate curing can cause defects such as cracks in the concrete surface; (8) Inappropriate reinforcement placement in the correct position can lead to poor performance and defects in the concrete surface; (9) Lack of sufficient concrete cover on the reinforcement causes the reinforcement to corrode, resulting in surface defects [66]; (10) Improper application of the release agent during construction can cause defects such as stains on the concrete surface; and (11) Insufficient supervision and the use of poorly skilled workmanships are other factors that contribute to defects in the concrete surface.

The main problems during the operation phase that can cause surface defects include: (1) Environmental problems due to changes in temperature, humidity, and moisture, which can cause various factors (e.g. thermal stress) that play a role in forming surface cracks [67]; (2) Exposing concrete to fire can cause changes in the microstructures and defects such as cracks and delamination [68]; (3) Chemical mechanisms such as acid reactions, alkali-aggregate reactions, carbonation, chloride penetration, creation of composing salts, leaching, oil and fat influence, and sulfates reactions [59]; (4) Biological degradation mechanisms involve the accumulation of contamination and living organism activity; (5) Corrosion and expansion of reinforcement are other causes of surface defects during the operation that can be affected by environmental problems [69]; (6) Abrasion is the action of deterioration of concrete that causes wear. Wear is the loss of concrete mass caused by impact, friction, vehicle traffic, etc. [41, 70]; (7) Physical degradation mechanisms include creeping, shrinkage, material fatigue, extreme temperature influence, freezing actions, foundation displacement, and overloading [59]; (8) Load problems include static and dynamic over-loading, excessive vibration, and stress concentration that can lead to surface defects [71]; (9) Changes in the internal and external properties of concrete, including creep and shrinkage, can cause settlement or deformation, which are factors in the appearance of surface defects [72], [73]; and (10) Vandalism is also a cause of surface defects during the operation phase [74].

During the maintenance phase, problems such as lack of maintenance and insufficient frequency of applying surface protection will cause aging deterioration and, consequently, concrete surface defects [75, 76].

The diagnosis methods to find the causes of concrete surface defects include: (1) The remote-sensing-based diagnosis analyzes the results of inspection based on the remote sensing methods (e.g. analyzing temperature based on thermal images); (2) The magnetic-based diagnosis analyzes magnetic force information, rebar location, and size of the cover over time to notice the corrosion and other defects based on the results of inspection methods such as concrete cover measuring [77]; (3) The acoustic-based diagnosis, which is a simple method, analyzes the sound produced based on the results of the inspection methods, such as the rubber hammer test, to find the initial occurrence of some defects [78]; (4) The moisture/humidity-based diagnosis analyzes adsorption, moisture content, and humidity level data and finds the degree of porosity of concrete surfaces based on the results of inspection methods such as the initial absorption test; (5) The chemical-based diagnosis analyzes the presence of harmful substances based on the results of the inspection methods such as colorimetric strips. Analyzing the results of this test helps to identify the types of harmful substances and defects caused by them on the concrete surface; (6) The electrochemical-based diagnosis analyzes corrosion probability based on the results of inspection methods such as the half-cell potential test; (7) Crack monitoring diagnosis analyzes crack dimensions over time based on the results of inspection methods such as cracking measuring [79].

2.2.4 Condition Assessment

Based on OSIM's [41] classification of the severity and condition of concrete surface defects, the condition of defected element can be classified into excellent, good, fair, and poor. Excellent condition is usually a condition in which the element or structure is newly constructed, and the element is without surface defects and does not require treatment. Good condition is when surface defects are visible and minor, but since these defects do not affect the element's function, no corrective action is needed to treat the defect. Fair condition is a condition in which moderate defects are visible, and if 3R actions are cost-effective, preventive, corrective action will be taken (e.g. applying a protective surface coating). Poor conditions are conditions in which the defect is severe and seriously affects the performance of the element. In this case, there will be a need for treatment measures such as rehabilitation and replacement. The presence of medium cracks on the concrete surface indicates a fair condition, and the presence of wide cracks on the concrete surface indicates a poor condition. Since the presence of spalling in the concrete surface, regardless of the severity, indicates a serious problem and severe weakness in concrete, the appearance of this defect with any severity level indicates a poor condition that requires a corrective action. Moreover, the presence of delamination, disintegration, or erosion in the concrete surface with any severity level indicates a poor condition. In case of severe or very severe scaling, honeycombing or pop-out on the concrete surface, the element's condition can be evaluated as poor. The appearance of cold joint, segregation, stratification, efflorescence, exudation, incrustation, stalactite, abrasion damage or wear, and slippery surface on the concrete surface indicate the element's fair condition.

2.2.5 Repair, Rehabilitation, and Replacement

The 3R actions help to extend the actual useful life of a structure after the formation of defects or damages caused by defects. ISO (International Organization for Standardization) [80] provides a framework and fundamental principles for the maintenance and repair of existing concrete structures. The repair methods to deal with minor concrete surface defects are: (1) Surface cleaning is a method to remove water, dirt, debris, and stains from concrete surfaces [81]; (2) Irregularities related to non-flatness (bulge, roughness, waviness) can be repaired by methods such as surface grinding. Since excessive grinding also weakens the concrete surface, this issue should be considered when using this method [82]; (3) Protecting edges is a preventative method to protect the protruding edge of concrete surfaces from defects [60, 83]; (4) Using surface sealing or coating compounds, which is a preventative action, prevents the penetration of water and other destructive chemical solutions into the concrete surface and reduces damage caused by chemical reactions, corrosion of rebars, etc. The difference between surface sealers and surface coating is that the surface sealers penetrate the surface and usually apply in a thin layer. In contrast, surface coatings apply as a covering on top of the surface in a thicker layer than penetrating sealers. Minor concrete surface cracks can also be treated by applying the penetrating surface sealers (e.g. epoxy sealer, silicone sealer). The materials used for sealing and coating compounds have a wide variety of choices [84].

The methods of major repair of defective concrete include: (1) Rehabilitation and strengthening of concrete: excessive corrosion of the rebar weakens the strength and causes defects such as surface cracks. In such cases, in order to strengthen the concrete, the corroded parts are removed and replaced with new rebars [85]. Moreover, in cases where the concrete needs to be strengthened, fiber-reinforced polymers (e.g. glass fibers, steel fibers) can be used in concrete [86]; (2) Concrete repair or replacement: in order to repair or replace concrete, various measures are taken, including detaching and removing the loose part of the concrete, adding a new layer of concrete, and curing. curing is done for the added or replaced concrete to maintain the appropriate humidity and temperature conditions at depth and surface, which plays a vital role in developing the strength and durability of concrete [87].

Concrete repair or replacement method include: (1) Crack filling: resin injection is a method to repair defective concrete. Some types of penetrating surface sealers (e.g. polyurethane) can be used for this purpose [88]; (2) Shotcrete placement: in the shotcrete method, concrete mortar is thrown on the surface at high speed to place high-strength, low-permeability concrete without using forms [89]; (3) Adding or replacing mortar or concrete. Various types of concrete or mortar can be used in this method. Some types of mortar or concrete include: (1) Conventional mortar or concrete: the main components of conventional mortar or concrete are water, cement, and aggregate, which can be used to repair or replace defective parts of surfaces [90]; (2) Preplaced aggregate concrete: this concrete is made of coarse clean aggregates that are compacted together and cement grout injection. Due to the fluidity of the cement grout injected in this method, the forms must be made in such a way that they can withstand more pressure than conventional concrete [91]. The defective parts of the concrete surface can be repaired or replaced with this concrete; (3) Polymer-modified mortar or concrete: this mortar or concrete consists of a combination of polymer with cement, and aggregate and polymer concrete mortar consist of only polymer and aggregate. Polymer-modified mortar or concrete is made from a combination of water and polymer additives, cement, and sand that require less water than conventional concrete. This mortar or concrete has high strength, adhesion, and density. Another feature of this mortar or concrete is reducing permeability and shrinkage [92], [93]; and (4) Epoxy mortar or concrete: this mortar or concrete is made from a combination of epoxy and sand or epoxy, sand, and coarse aggregate. This mortar or concrete has characteristics to protect the rebar against corrosion [94].

Current approaches to inspecting and maintaining concrete surfaces face challenges due to subjectivity and inefficiency. In order to take appropriate 3R actions, defects must be identified simultaneously, and the characteristic of each defect must be considered for appropriate future action [48].

2.3 Inspection Tasks During the Construction and Operation Phases

2.3.1 Construction Inspection

Inspection during the construction phase is an important task in the construction industry. Lack of proper inspection will increase the cost of maintenance in the operation phase. Based on Tayeh et al. [95], the main factors causing construction defects are: (1) misinterpretation of design, (2) inaccurate measurement, (3) damaged formwork, (4) poor installation method, (5) improper installation, (6) early formwork removal, (7) excavation tools close to the building, and (8) painting in unsuitable conditions or on unsuitable surfaces. Kim et al. [96] proposed a framework for dimensional and surface quality assessment of precast concrete elements using BIM and Light Detection and Ranging (LiDAR) scanning.

Recent technologies (e.g. LiDAR scanner) are integrated with BIM to enhance the capabilities of construction inspection [97]. An accurate and comprehensive inspection of construction sites can be achieved with the aid of LiDAR scanner and sensors. These technologies can capture real-time data from the site [98, 99]. The use of computer vision techniques can help to inspect most of the surface defects [100, 101]. Bolourian and Hammad [102] considered the potential locations of the defects on the inspected surfaces and proposed a path planning method for LiDAR-equipped Unmanned Aerial Vehicle (UAV). Lundeen et al. [103] developed an adaptive inspection framework for construction robots to detect the location and geometry of joints and fill these joints. Freimuth et al. [104] used BIM for UAV flight path planning for construction inspection. The object boundaries were defined in the georeferenced BIM and represented as a set of voxels in three main categories: (1) building geometry, (2) occupied voxels, and (3) safety layer. The safety layer is based on the minimum distance between the UAV and the objects. A graph is generated to relate the voxels (nodes), and each transition between nodes indicates a movement of the UAV using way-nodes.

2.3.2 Inspection During Operation Phase

The other area of inspection is inspection during the operation phase. Facilities need regular inspection to satisfy their predetermined functions. Imperfections in the facilities are described as defects, errors, faults, failures, quality deviations, nonconformances, anomalies, snags, reworks, etc. [105]. Preventive maintenance, reactive maintenance, and emergency services are the three main types of maintenance. For preventive maintenance, safety and efficient functioning condition are two main factors that need to be inspected regularly [33]. Metni and Hamel [106] used visual inspection for monitoring the structures at the operation phase and discussed the challenges of considering the orientation limits for UAVs. The orientation limits help the UAV to focus on the inspected object within the field of view of the sensor.

To reflect the changes related to inspection during the operation phase, Chen et al. [107] focused on defect modeling. Aruga and Yabuki [108] proposed a cooperative management model for structures in the operation phase. Hammad et al. [109] developed an inspection ontology using BIM for lifecycle inspection and repair information modeling. This work focused on integrating all the inspection details into one model to facilitate accessing and updating the information at different life cycle phases. Kasireddy and Akinci [29] proposed integrating inspection data with IFC to support condition assessment.

2.3.3 Post-disaster Inspection

Post-disaster inspection is the third inspection type which should be done in the event of a disaster and before re-occupying the building to evaluate potential health and safety hazards. Search and rescue inspection are also done after disasters, which is a time-sensitive task, and it needs a quick action to reduce the potential injuries and damages [110- 112]. As an example, in case of a fire, inspection needs to be done to control the fire and rescue the people trapped in the building [113]. Inaccuracy, incompleteness, and poor communication are the key problems that affect this task [114].

2.4 Using Robots for Inspection

With the advancement of technology, autonomous robots have evolved and are equipped with advanced capabilities, including quality and production control, reducing workers' workload, enhancing the safety and efficiency in hazardous environments, and data gathering [115- 117]. Mobile robots are designed for sensing, navigation, inspection, and remote operation in dangerous situations. Service robots can be used for different purposes in the AEC/FM industry. Industrial cases show the variety in types, capabilities, and uses of robots. Autonomous unmanned systems including UAVs, Unmanned Ground Vehicles (UGVs), and Autonomous Underwater Vehicles (AUVs) can be used for quality inspection and measuring the installed quantities [118, 119]. For instance, Doxel is a liDAR-equipped robot that scans construction sites to monitor the work. This robot can go along a set route and scrambles up the stairs and beams to compile the report of site data [120]. Another example is Spot, which is a four-legged robot that can climb the stairs and move to the inspection spots, which are difficult for wheeled robots. This robot can perform tasks such as opening doors and grasping objects, progress inspection, thermal inspection, leak detection, noise detection, creating a digital twin, and comparing the surrounding environment with BIM [121]. Some types of climbing robots are designed to move on steel structures and inspect the surfaces. The lifting mechanism in these robots utilizes the magnet embedded in the robot wheels [122]. To perform inspection tasks, Cobalt is another type of robots which can detect a spill and leak in a pre-set zone and notify the specialists [123]. Denso [124] is a robot that consists of a robotic arm and a control module. It can be equipped with a LiDAR scanner for various purposes (e.g. inspection) [125]. DiddyBorg robot [126], which is a six-wheeled high-torque robotics platform, can be customized by adding different sensors, including a camera and a LiDAR scanner to perform different tasks [127]. Some robots are developed for cleaning purposes and can be modified for inspection purposes. For instance, Roomba is an autonomous robotic cleaner, which can navigate the floor with the aid of sensors and detect walls and obstacles [128]. Another area that industrial robots can help is underwater inspection. As an example, Hydro-Québec developed an underwater robot for dam inspection. This robot can accomplish visual inspection using defect measurement, acoustic imaging, and surface reconstruction using sonar data [129]. Autonomous robots can be used to find defects in an urgent situation and disaster relief and can also help the survival of people. As an example, the Bipedal robot developed by Honda has features such as picking objects, 180-degree rotation, and going up and down the stairs and ladders, which can help in the process of emergency response, inspection, and maintenance [130].

2.4.1 Robotic control system for building inspection

An autonomous robot control system enables robots to perform human activities in a building [131]. Cognitive Robot Abstract Machine (CRAM) is a software toolbox for designing and implementing cognitive-enabled autonomous robots, which is built using the Robot Operating System (ROS) framework. The two main parts of CRAM are: (1) CRAM Plan Language (CPL), and (2) Knowledge processing system (KnowRob). KnowRob contains a small core system and a large set of optional modules. It is based on SWI Prolog and Semantic Web library to access Web Ontology Language (OWL) files [132]. An autonomous system consists of a platform, mission computer, actuators, sensors, control system, a navigation system, datalink, and base station. The elements of an autonomous vehicle system that should be considered in the ontology are decision making, path planning, sensors, control, actuators, and robotic platform. Some examples of domain specifications in creating the ontology are different types of sensors (e.g. Global Positioning System (GPS), LiDAR), platform, task (e.g. navigation), and mission (e.g. inspection, rescue). Figure 2-2 demonstrates examples of UAVs domain specifications in creating the ontology. The entities' relationships of the domains are shown in Figure 2-3. Figure 2-4 shows KnowRob-Map diagram. Different relationships such as allocated-to, have, consists-of, executes, etc., can be defined among these entities. The elementary knowledge representation of UGVs provides reasoning capabilities for the robot's decision-making process. Sub-systems of this knowledge representation are locomotion (e.g. legged, wheeled), sensors, actuators, planning, and communication [133]. The decision-making concepts in the ontology can be extended for building inspection.

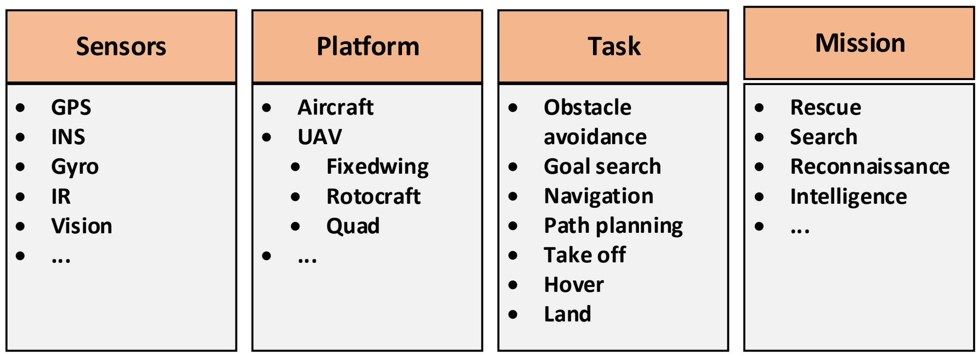


Figure 2-2. UAV domain specifications [133]

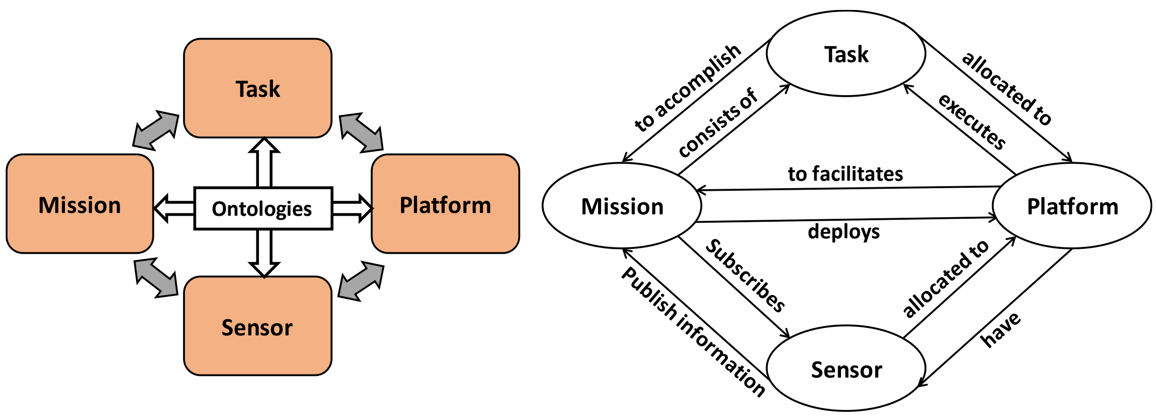


Figure 2-3. Entities' relationships of robotic ontology [133]

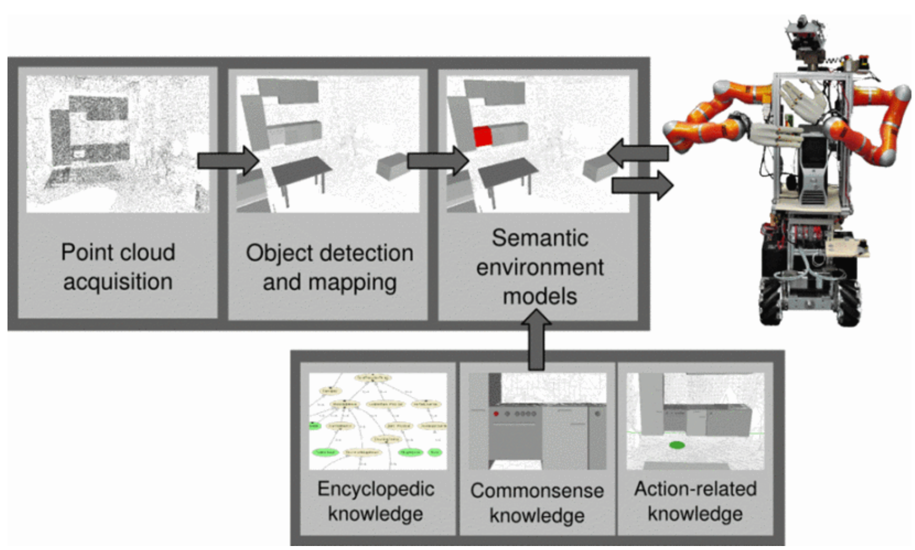


Figure 2-4. KnowRob-Map diagram [139]

2.4.2 Review of Robotic Inspection and Navigation Related Works

In this section, the research about integrating some aspects of robotic, inspection, and navigation with BIM is reviewed. Table 2-2 lists a summary of most related papers including the following information: (1) the objectives of the inspection; (2) the building lifecycle phase (e.g. construction, operation) in which the inspection is conducted; (3) in case of using a robot, the type of the robot (i.e. UAV, UGV); (3) the type of sensor (i.e. RGB (Red, Green, and Blue)/depth camera, thermal camera, or LiDAR); (4) using BIM model or IFC concepts in the process; and in case of using BIM, considering mismatch between actual structure and the model; (5) using the knowledge-based method (i.e. ontology); and (6) description of the objectives. The order of the papers is based on the most recent year of publication.

Some studies focused on using different sensors and robots for inspection purposes. All the studies considered at least one type of sensor (i.e. LiDAR or camera), which could be an element of the robot in an integrated platform or could be mounted on the robot. The target element of the inspection was building/infrastructure elements and specific defects, such as cracks on steel or concrete surfaces. In addition to the review of most related papers to robotic inspection in Table 2-2, Hamledari et al. [135] and Wang et al. [136] papers are added only because their works considered a mismatch between the actual structure and the BIM model.

Despite the great benefits of the reviewed papers, they have the following main limitations: (1) Robot awareness about the environment and the accuracy of objects' information and interactions during the task could be improved by considering a semantic description (i.e. ontology) [137- 143]; In most of these works, semantic description (i.e. ontology) was not considered (except [144]). (2) A standard BIM model was not used as a reference for navigation and localizing the defects. The comparison shows that several studies did not consider BIM in their work [137, 139, 140, 142, 144-148]; (3) In the case of having a BIM model, BIM elements were not updated based on mismatch consideration; and functional properties of BIM elements were not used for inspection [136, 149]. The BIM model is assumed to be complete and reliable [138], which is one of the main problems when using BIM-based robotic inspection. BIM models must be updated as the changes can occur in any building lifecycle phase. In some papers, the terms scan-to-BIM (e.g. [149]) and site-to-BIM (e.g. [135]) are often used interchangeably to refer to the process of creating as-built or as-is BIM models based on site conditions [150, 151]; and (4) Robotic navigation and obstacle detection were not considered in some studies [146-148, 152]. Navigation here refers to using a path generation method and obstacle avoidance based on sensor data.

From the studied literature, the integration of knowledge of robotic inspection and the construction domain is a key factor of an effective and efficient inspection, which needs more attention.

Table 2-2. Summary of most related papers to BIM-based robotic inspection and navigation tasks

Objectives	Year	Inspected Elements	Phase	Navigation	Robot: UAV	Robot: UGV	Sensor: LiDAR	Sensor: RGB / Depth Camera	Sensor: Thermal Camera	BIM: Model / IFC	BIM: Mismatch Consideration	Description
An autonomous thermal scanning system with which to obtain 3D thermal models of buildings [143]	2019	Indoor building elements	Operation	✓	-	✓	✓	✓	✓	-	-	Developed an autonomous platform using a LiDAR, RGB camera, and a thermal camera that navigates from one position to another to find the next best view position and provides a 3D thermal model by matching several 3D thermal images.
A framework for automated acquisition and processing of as-built data with autonomous unmanned aerial vehicles [149]	2019	Building elements	Constructio	✓	✓	-	✓	✓	-	✓	✓	Proposed a framework for extracting inspection target locations based on BIM and using UAVs to automate as-built data generation.
Automated robotic monitoring and inspection of steel structures and bridges [142]	2019	Steel cracks	Operation	✓	-	✓	✓	✓	-	-	-	Designed a climbing robot, equipped with sensors to collect and analyze camera data.
Automatic wall defect detection using an autonomous robot: a focus on data collection [145]	2019	Walls	Operation	✓	-	✓	✓	-	-	-	-	Developed an autonomous robot-enabled data collection system for indoor wall condition assessment.
Autonomous robotic exploration by incremental road map construction [137]	2019	Indoor building elements	Operation	✓	-	✓	✓	-	-	-	-	Proposed a framework for autonomous robotic exploration in 2D unknown environments considering both path planning and decision-making in the exploration process.

Table 2-2. Summary of most related papers to BIM-based robotic inspection and navigation tasks (continued)

Objectives	Year	Inspected Elements	Phase	Navigation	Robot		Sensor			BIM		Description
					UAV	UGV	LiDAR	RGB / Depth Camera	Thermal Camera	Model / IFC	Mismatch Consideration	
Planning and executing construction inspections with unmanned aerial vehicles [138]	2018	Building roofs	Construction	✓	✓	-	-	✓	-	✓	-	Developed an integrated concept for planning and executing UAV collision-free flight paths to improve inspection tasks with case study.
Tunnel structural inspection and assessment using an autonomous robotic system [139]	2018	Concrete cracks	Operation	✓	-	✓	✓	✓	-	-	-	Developed a multi-degree-of-freedom (multi-DOF) robotic system to automate data collection and inspection using different sensors and LiDAR.
Design and development of an inspection robotic system for indoor applications [140]	2018	Building elements (tested on inspecting walls)	Operation	✓	-	✓	-	✓	✓	-	-	Developed an inspection robotic system considering the requirements of mobility, sensorization, communication, and hardware and software reliability.
A semi-autonomous mobile robot for bridge inspection [146]	2018	Concrete cracks (tested on inspecting columns)	Operation	-	-	✓	-	✓	-	-	-	Proposed a robotic inspection system considering a customized truck and a robotic mechanism to capture pictures from the target area and associates them with the CAD model.
IFC-based development of as-built and as-is BIMs using construction and facility inspection data: site-to-BIM data transfer automation [135]	2018	Building elements: walls, doors, outlets, light fixtures	Operation	-	-	-	-	✓	-	✓	✓	Proposed a computational solution that uses IFC schema to automatically update as-designed BIM.

Table 2-2. Summary of most related papers to BIM-based robotic inspection and navigation tasks (continued)

Objectives	Year	Inspected Elements	Phase	Navigation	Robot		Sensor			BIM		Description
					UAV	UGV	LiDAR	RGB / Depth Camera	Thermal Camera	Model / IFC	Mismatch Consideration	
Automated quality assessment of precast concrete elements with geometry irregularities using terrestrial laser scanning [136]	2016	Precast concrete elements	Construction	-	-	-	✓	-	-	✓	✓	Estimated the dimensions of the elements using TLS LiDAR and compare it with as-designed BIM as a reference.
Infrared building inspection with unmanned aerial vehicles [152]	2015	Building elements (tested on inspecting roof and roof windows)	Operation	-	✓	-	-	-	✓	-	-	A case study using UAV equipped with an IR camera for inspection of detached houses.
Efficient search for known objects in unknown environments using autonomous indoor robots [144]	2015	Indoor building elements	Operation	✓	-	✓	-	✓	-	-	-	Developed an ontology-based system for detecting inspection targets using vision sensors and implemented the system in ROS.
A robotic crack inspection and mapping system for bridge deck maintenance [141]	2014	Concrete cracks	Operation	✓	-	✓	✓	✓	-	-	-	Proposed a robotic crack inspection and mapping system using a robot equipped with cameras to capture images for inspection tasks.

Table 2-2. Summary of most related papers to BIM-based robotic inspection and navigation tasks (continued)

Objectives	Year	Inspected Elements	Phase	Navigation	Robot		Sensor			BIM		Description
					UAV	UGV	LiDAR	RGB / Depth Camera	Thermal Camera	Model / IFC	Mismatch Consideration	
Low-cost aerial unit for outdoor inspection of building façades [147]	2013	Building facade and envelope elements (tested on facade openings)	Operation	-	✓	-	-	✓	-	-	-	Presented a platform to analyze the potential of using UAV for building geometric inspection and creating a 3D point cloud based on captured images.
Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel [148]	2007	Concrete cracks (tested on inspecting walls)	Operation	-	-	✓	-	✓	-	-	-	Proposed an image processing based mobile robotic system for crack detection.

2.5 BIM and IFC Applications in the Context of Inspection

BIM is a new approach to model all the information related to buildings and infrastructure systems, respectively, by integrating this information with 3D models representing the geometrical and spatial characteristics of these systems. The international standard of BIM is the *Industry Foundation Classes* (IFC), which was developed by buildingSMART and has a detailed semantic representation of the building model [153, 154]. The IFC schema represents objects and their semantic relationships. Any semantic concept for data modeling needs to define relationships, including: (1) aggregation relationships (e.g. a door in a wall), (2) topological relationships (e.g. connection of two walls), and (3) directional relationships (e.g. a floor that is above another floor) [155]. Model View Definition (MVD) breaks down the IFC schema into smaller parts. Based on different needs, many structures and Levels of Details (LOD) can be defined for diverse areas. buildingSMART International developed an ifcDoc tool for editing and generating MVD for IFC Extensible Markup Language (IFCXML) [156]. Regarding the different lifecycle phases, BIM models of a building include as-designed at the design phase, as-built at the construction phase [157, 158], and as-is at the Operation and Maintenance (O&M) phase. It should be noted that each of these models has several versions and should be continuously updated to reflect design, construction, deterioration, and repair changes in the different phases of the lifecycle.

Tang et al. [159] specified technological and organizational obstacles as the main challenges of a seamless sensing and modeling process. Kensek [160] studied the possibility of connecting sensing data and BIM. The case studies were implemented through scenarios using Ardunio, Dynamo, and Revit Application Programming Interface (API). Physical facility objects and their functions can be represented with the visual interface of BIM. This kind of representation in BIM can create a decision-making platform for FM. By integrating 3D models of buildings and sensing data, information such as real-time performance feedback can be displayed, and the decision support system for facility managers will be improved [161, 162]. This capability helps the facility managers to make better decisions during the lifecycle of assets [163, 164].

2.5.1 IFC-based Navigation

Path planning in 3D spaces needs information related to spaces and their functions, geometry, and locations, assets and obstacles related to spaces, and accessibility of spaces. BIM can help to extract precise and up-to-date semantic and geometrical data from the building model [165, 166]. As an example, the wall is a building element which has the following main IFC entities: (1) IfcWallStandardCase for a wall when the thickness of the wall along the wall is constant or fully explained by a material layer set which represents the number of layers, the position, and the type of each layer. It is defining a wall with certain constraints to represent its parameters (e.g. height, thickness, offset from the axis) and geometric representation, (2) IfcWallElementedCase which is a combination of dependent elements. It is defined as a wall with certain constraints to represent its components and categorize all the parts, (3) IfcWall which is used to define all other walls (e.g. polygonal walls, L-shaped retaining walls, non-vertical walls). For floors and ceilings, the IFC standard uses IfcSlab as a core, and IfcCovering is used as a dependent element, which has the property sets of Pset_CoveringCeiling for ceiling and Pset_CoveringFlooring for floors.

IFC schema represents objects and their semantic relationships. Lin et al. [165] used an IFC file as the input for path planning. They extracted all geometric and semantic information from the IFC file and mapped them to a planar grid. In their study, IFC2x3 [167] was used as a reference file. In IFC2x3, a spatial structure is defined under the category of IfcSpatialStructureElement.

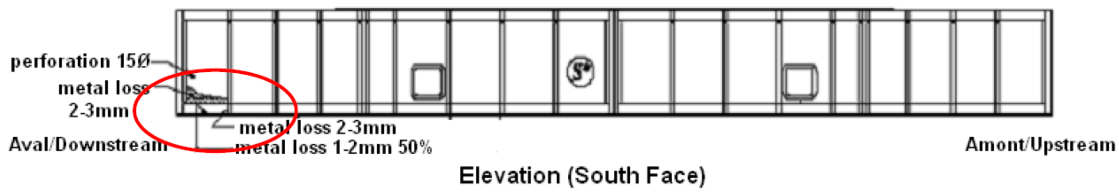
The spatial structure in IFC includes IfcSite, IfcBuilding, IfcBuildingStorey, and IfcSpace. IfcElement includes IfcBuildingElement (e.g. walls and doors), IfcFurnishingElement and IfcDistributionElement (e.g. ducts, pipes, and cables). IfcElement is a generalization of all AEC products, which are represented as IfcProduct. Lin et al. [165] used some of these entities from the IFC2x3 standard. However, they did not consider IfcPath, which is a topological entity including the collection of oriented edges.

The logical network is a representation of the full 3D building model and a detailed navigable network (e.g. spatial relations between floors, rooms with shared walls, etc.) [168]. Yan et al. [169] used a door-room connectivity graph method and a Revit file as a BIM model for collision-free path planning, considering walls and other objects as obstacles. The graph indicates the logical navigation between the centers of rooms' nodes and determined centers of doors and openings on the edges of spaces without considering the closed or open state and direction of the door. Hazardous zones, such as path hazards (e.g. slipping, tripping and falling hazards), environmental hazards (e.g. high temperature), and unexpected hazards (e.g. burst pipe, radiation) are other factors that can cause problems for the robot during the operation [165, 170, 171]. Zverovich et al. [111] developed a method to select the safest and balanced path by considering path length and hazard proximity (i.e. distance and number of obstacles) based on the BIM model. Strug and Ślusarczyk [172] proposed an IFC-based method for investigating the accessible path for disabled people by considering attributes such as the width of openings, door types, and door opening directions. Ivanov [173] used a context-aware BIM-based navigation model, which can search for an optimal path by considering both capabilities of users to navigate in an unfamiliar environment and the status of all sensors. Furthermore, the integration of motion sensors with BIM-based knowledge of the path will improve navigation reliability [174].

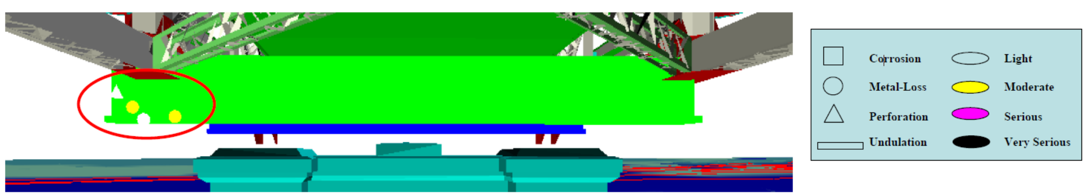
Eastman et al. [175] extracted the information from the IFC file and mapped them to nodes and edges of a graph with Solibri Model Checker. The accessibility rule checking was done based on several rules, such as the width of doors and corridors. In their work, security zones for public, restricted, and secure parts were defined as a property set for each space and circulation path in the IFC file. Then, they automatically mapped building elements into a topological graph and a metric graph. Checking the path with respect to the spatial elements was done based on the topological graph. Analyzing the moving distances and their visualization was done using the metric graph. All the instances of IFCRelSpaceBoundary, which define the relationship between surrounding building elements and a space in an IFC standard, can be used to indicate the topological representations in each level [176]. Connection, separation, intersection, containment, and adjacency are examples of spatial relationships between the building elements defined by topological representations [177, 178]. In another effort, Rasmussen et al. [179] stated that topological relationships between zones and elements of a BIM model can be described as *interface* class in their proposed minimal Building Topology Ontology (BOT) ontology. Furthermore, they stated that topological relationships can be used to specify restricted zones in the navigation. BOT ontology can be used in combination with other ontologies to define the building products such as walls and windows.

2.5.2 Inspection and Repair Information Modeling

Several studies explored extending Building and Civil Infrastructure Information Modelling (BIM/CIM) for Inspection and Repair Information Modeling (IRIM). For example, in the area of facilities management, Hassanain et al. [180] developed an integrated maintenance management prototype that demonstrated the potential uses of IFC to improve interoperability in the AEC/FM industry. Davila Delgado et al. [181] proposed an extension to the IFC data model standard for structural monitoring systems. Their extension could model the structural monitoring systems, store and retrieve obtained data, and visualize the BIM model's data. Chen et al. [182] proposed an approach to monitor the state of assets using an embedded sensing system and IFC-based BIM model. Hammad et al. [183] also proposed a framework for life-cycle infrastructure information modeling and management. However, they did not discuss the details of the formal definition of this information. Some previous work, such as Mailhot and Busuio [184], focused on manually recording of the inspection data in a 2D or 3D location-based sketching. Figure 2-5 shows the manual recording of inspection data using sketches. However, as these methods are fragmented, they were not useful.



(a) 2D Sketching



(b) 3D Location-based Model

Figure 2-5. Manual recording of inspection data using sketches [184]

Defects are considered in two different phases of the lifecycle of infrastructure facilities: the construction (or manufacturing) phase and the O&M phase. In the construction phase, defects are caused by errors or imperfections in the construction. In the O&M phase, defects are caused by factors such as loads applied on the structure, environmental effects, and natural aging. Although the causes of surface defects can be very different in these two phases, there are important similarities that can be exploited in developing IRIM from the point of view of type of defects (e.g. cracks, spalling) as well as the inspection processes and methods.

2.5.2.1 IRIM in the Construction Phase

Park et al. [185] proposed a framework for construction defect management using BIM and ontology-based data collection template. This framework proactively reduces the incidence of defects through an organized inspection plan. Figure 2-6 shows the proposed defect specific domain ontology including defect description, root cause analysis, impact analysis and control factor analysis. Kim et al. [96] proposed a framework for dimensional and surface quality assessment of precast concrete elements using BIM and LiDAR scanning. The proposed IFC-based entity-relationship model for the precast concrete element quality inspection is rather simple and does not cover all the details needed for modeling the defects information in a comprehensive way. For example, as shown in Figure 2-7, the location of the defects is represented using ifcDirection, which is obviously not enough to specify the location of the defect on the 3D model of the structure.

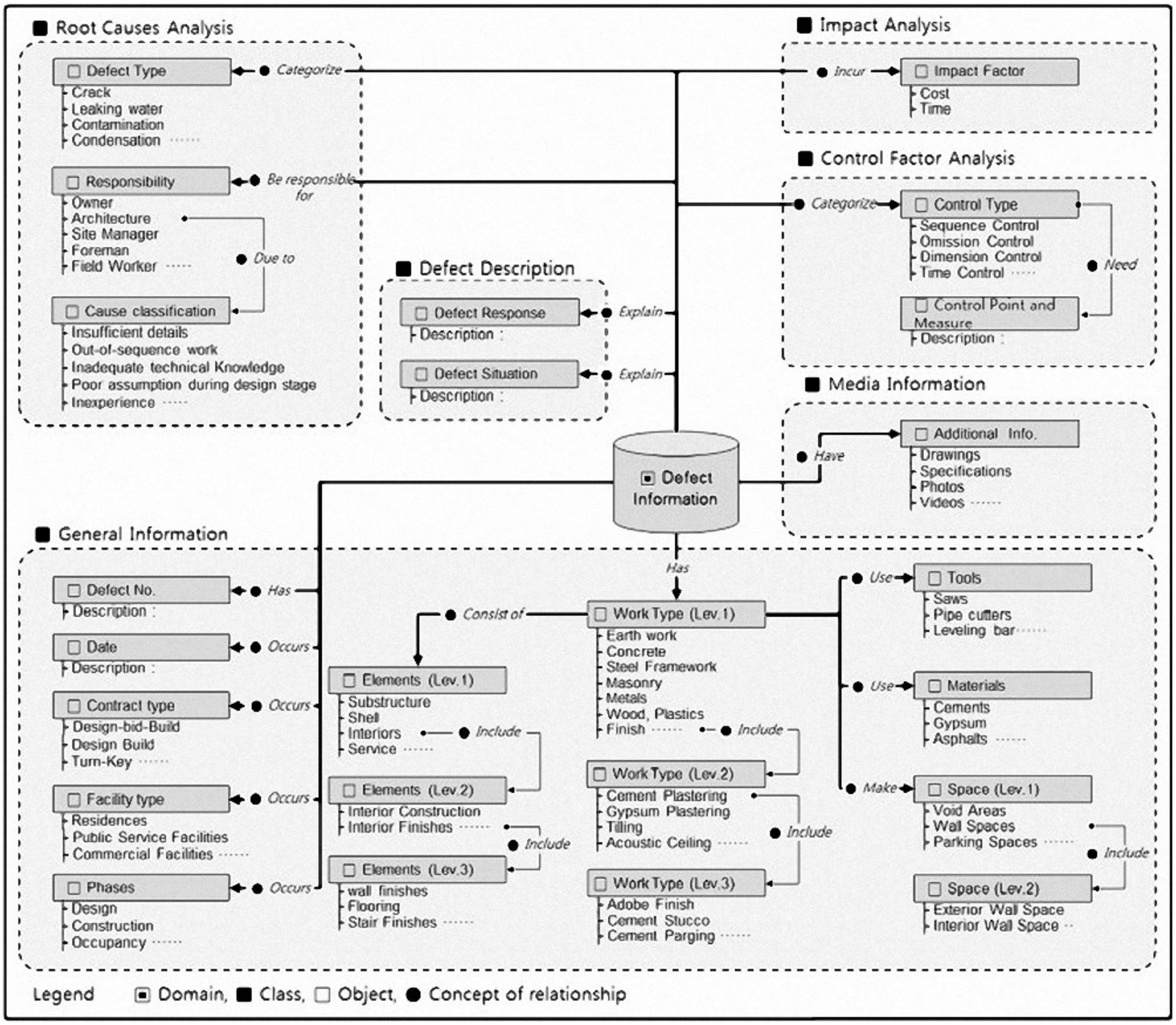


Figure 2-6. Defect domain ontology [185]

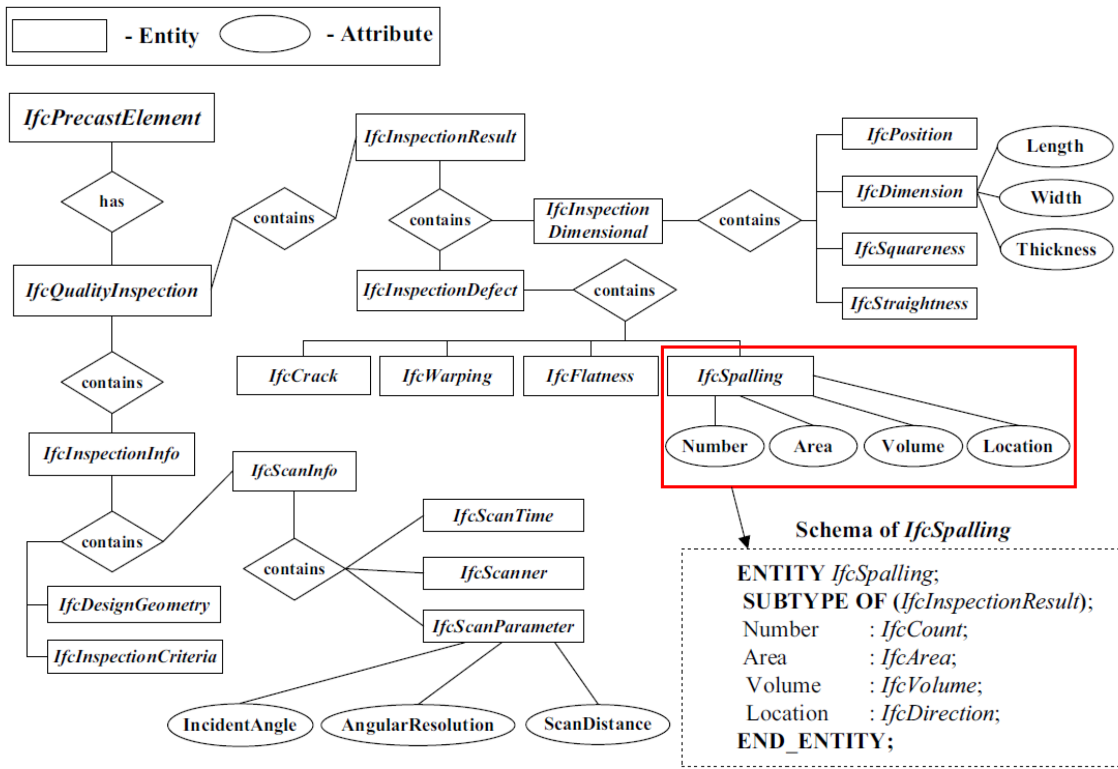
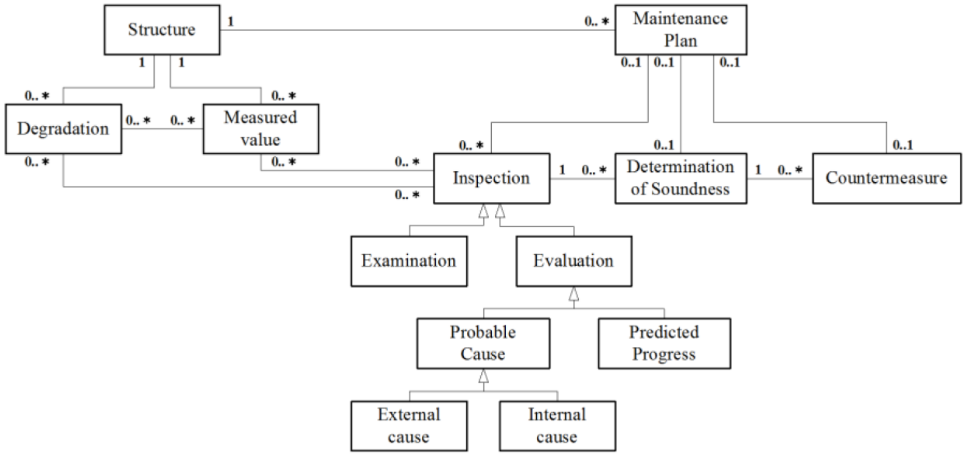


Figure 2-7. IFC-based entity-relationship model for precast concrete element quality inspection [96]

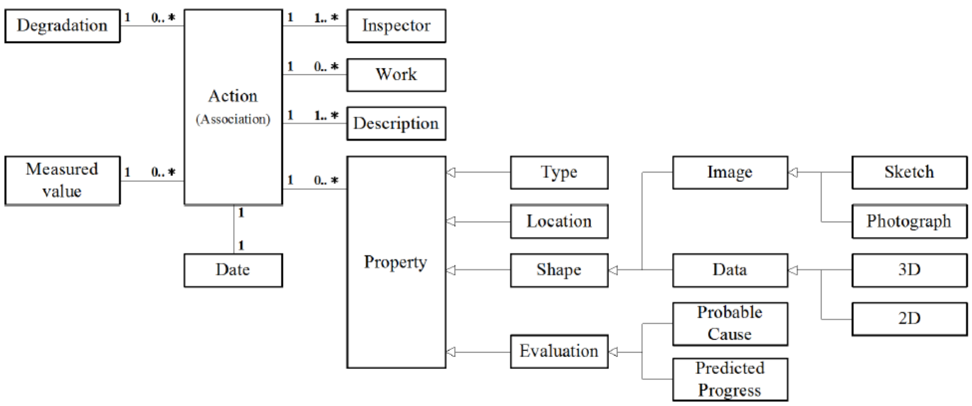
2.5.2.2 IRIM in the O&M Phase

Aruga and Yabuki [108, 186] proposed a cooperative management model for structures in the O&M phase. As shown in Figure 2-8(a), the maintenance management framework considers both the degradation level (i.e. condition assessment) and the measured values (i.e. inspection results). The evaluation based on inspection includes identifying the probable cause of the defect and predicts its future progress. Furthermore, the framework of the degradation and measured values shown in Figure 2-8(b) includes several inspection data types (e.g. sketch, photo, drawings) that could be used to identify the shape and location of the defects. However, this research did not discuss all the details of the IRIM.

Hammad et al. [187] demonstrated the applicability of 4D visualization of bridge lifecycle information based on a standard model. They proposed a framework for a mobile model-based bridge lifecycle management system to link all the information related to the design, construction, inspection, and maintenance to a 4D model of a bridge combining different scales of space and time. However, their proposed system did not include measured data.



(a) Framework of maintenance management



(b) Framework of degradation and measured values

Figure 2-8. Frameworks for inspection and maintenance [108, 186]

Kasireddy and Akinci [29] proposed integrating inspection data with IFC-Bridge. The advantages of this model are using IfcRepresentation and several contexts for representing the geometry of a defect from multiple inspections and using extended relationships from IFC and IFC-Bridge to link bridge element information with condition information. They stated that one limitation in their approach is that they used some classes from the present version of IFC-Bridge to represent other classes required for condition assessment. Motamedi et al. [188] proposed a defect/degradation model that includes various categories; defect types, relationships between elements and defects, and the processes related to inspection, evaluation and repair of defects. Their proposed model extended IFC model to include new required elements. However, they did not investigate an ontology related to inspection and repair modeling. Chen et al. [107] developed a product model for harbor structures degradation as shown in Figure 2-9. One of the main contributions of this work is that defects are classified according to the following types: surface degradation (e.g. change of color), addition degradation (e.g. corrosion), subtraction degradation (e.g. cracks), deformation, and material deterioration. However, this research focused on the defect modeling for harbor structures and did not attempt to provide a general approach for IRIM. Hamledari et al. [135] proposed a computational solution that uses IFC schema to automatically update as-designed BIM based on construction and facility inspection data.

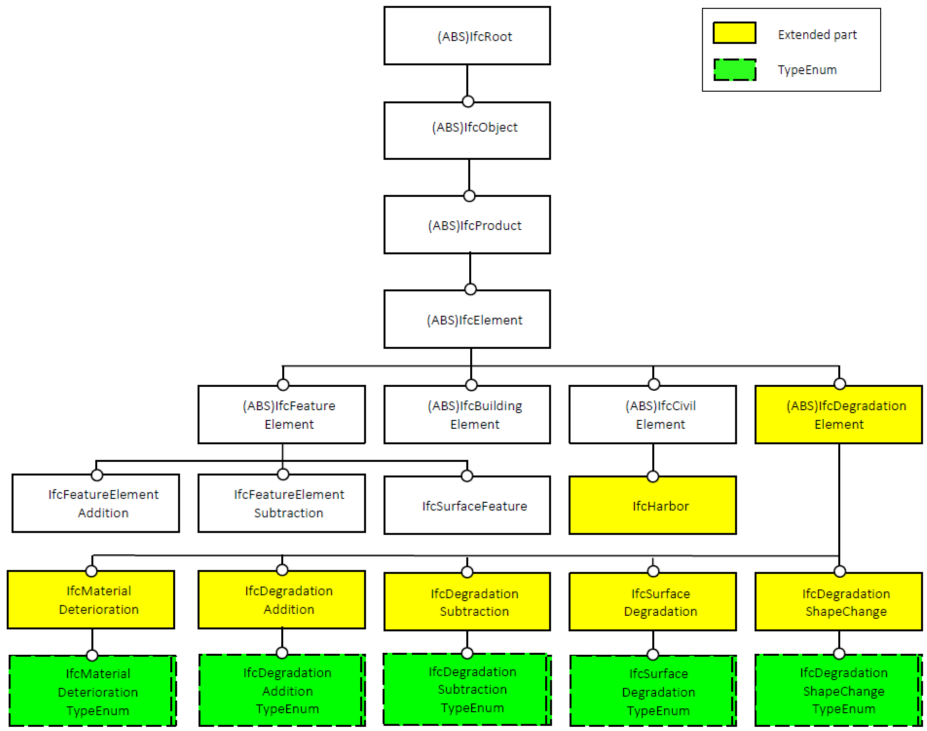


Figure 2-9. IFC-based entity-relationship model for the precast concrete element quality inspection [107]

Ma et al. [189] proposed an information model based on the IFC schema for damaged reinforced concrete structures from earthquake events. Their work was limited to structural elements. Based on their study, cracks can be represented as a texture on the element's surface, and structural damages such as breakage can be represented by trimming the building elements. Tanaka et al. [190] proposed an information model based on IFC to support the bridge inspection process. Moreover, a web-based system was developed based on WebGL to show the inspection information and images of degraded parts. Tanaka et al. [191] continued their work and proposed a system to extract inspection and repair reports. However, their work did not cover the detailed semantic information for inspection and repair processes. Sacks et al. [192] proposed a SeeBridge system for bridge information modeling based on inspection data. Their approach was not independent of the type of structure, and a detailed model for defect data was not covered in their work [193]. Hüthwohl et al. [194] proposed a framework to integrate bridge defect information with BIM. In their approach, some defect characteristics (e.g. type and size) were considered, and texture images were used to represent defects. Hamdan and Scherer [195] presented a framework for representing structural damages in BIM. Their approach was based on the multi-model approach, and in their study, a layered structure was developed to represent and visualize the damage geometry. In another study, Hamdan et al. [196] proposed a framework for semi-automatic generation of damage models and machine-based interpretation of the recorded structural damage data using ontology. Their work was based on a linked model approach, and a separate file was used to represent the damage model. This approach does not have all inspection data in one BIM model. Furthermore, their study relied on images that do not represent the defect's geometry as accurately as point cloud data. They developed a small Concrete Damage Ontology (CDO) and only focused on some concrete structural damages. They developed another separate small ontology for structural damage assessment based on the German guideline "Instruction of Road Information Databases for Constructions" [196]. However, their damage assessment focused on identifying index factors for assessing the damaging impact on structural health, durability, and traffic safety. The link for their used ontology for structural damage assessment is not publicly available, and their ontology was based on German terminology. Moreover, they did not develop inspection and 3R process ontology. Additionally, their developed ontologies were not unified and comprehensive, and they did not consider all the semantic relationships required for modeling.

Artus and Koch [197] presented two ways for modeling physical defects using IFC based on surface and void approaches. The texture images on top of the 3D component can represent the defect information in the surface-based approach. In the void-based approach, the defect geometry was subtracted from component geometry. Meshes from the point cloud can represent the spalling defect on the void element. Moreover, their void-based approach still has a problem with cracks geometry as cracks were modeled as extrusion of triangular profile. In another study, Artus et al. [198] presented a framework that generates spalling defect geometries from photos and saves them into a data model using IFC based on surface and void features. However, their work was based on image data, which does not contain the depth of defects. Isailović et al. [199] proposed a use case for enhancing an IFC-based bridge model using the image-based classification to identify the spalling defect features. Their approach depends on collected images from the inspection, and defect characteristics were directly identified from photos, which is not as accurate as point cloud data. The most related previous studies are summarized in Table 2-3, including whether or not the following information is included in the studies: (1) the inspection process; (2) the diagnosis process; (3) the 3R process; (4) using conceptual model (i.e. ontology); and (5) using BIM model or IFC concepts in the process.

Table 2-3. Summary of related works to inspection information modeling

Paper	Year	Type of defects	Process			Ontology	BIM
			Inspection	Diagnosis	3R		
A BIM Based Framework for Damage Segmentation, Modeling, and Visualization Using IFC [198]	2022	Spalling	-	-	-	-	✓
A semantic modeling approach for the automated detection and interpretation of structural damage [196]	2021	Structural damages (Concrete inhomogeneity, Crack, Spalling, Chemical damage, Moisture damage, Reinforcement damage, Tendon damage)	✓	✓	-	✓	✓
Modeling geometry and semantics of physical damages using IFC [197]	2020	Crack, Spalling	-	-	-	-	✓
Bridge damage: Detection, IFC-based semantic enrichment and visualization [199]	2020	Spalling	✓				✓
A generic model for the digitalization of structural damage [195]	2018	Non-specific structural damages	-	-	-	-	✓
Integrating RC bridge defect information into BIM model [194]	2018	Crack, Spalling, Scaling, Efflorescence, Rust staining, Abrasion/Wear, Exposed reinforcement	-	-	-	-	✓
SeeBridge as next generation bridge inspection: overview, information delivery manual and model view definition [192]	2018	Crack, Spalling, Scaling, Efflorescence, Rust staining, Abrasion/Wear	✓	-	-	-	✓
Bridge Information Modeling based on IFC for supporting maintenance management of existing bridges [191]	2018	Non-specific defects	✓	-	✓	-	✓
Bridge information model based on IFC standards and web content providing system for supporting an inspection process [190]	2016	Non-specific defects	✓	-	-	-	✓
Information modeling of earthquake-damaged reinforced concrete structures [189]	2015	Cracks, Structural damages (Braking, Buckling)	-	-	-	-	✓

2.5.2.3 Limitations of Previous Research Related to IRIM

Based on the review, in spite of the great benefits of the previous research related to IRIM, it has the following limitations:

(1) Duplication of efforts: Different researchers have focused on IRIM related to different types of civil infrastructures (e.g. bridges or tunnels), different types of material/elements, or at different phases of the lifecycle (e.g. construction or O&M). For example, comparing the models proposed by Chen et al. [107] for harbor concrete structures, Kasireddy and Akinci [29, 200] for bridges, and Kim et al. [96] for precast concrete elements, it can be seen that they used very different levels of detail for representing the properties of defects (e.g. location and geometry). This will result in duplication of efforts and less efficient research progress.

(2) Ad-hoc and shallow representation of concepts: One common aspect of most of the previous research works related to IRIM is that they focused on mapping a rudimentary data structure of the IRIM processes and products to the entities available in IFC or its derivatives (e.g. IFC-Bridge). This approach results in a rather ad-hoc and shallow models because not all the required entities are available in the current version of IFC. On the other hand, researchers are adding different entities that are duplicated but using variant terms. For example, the terms degradation and defect are used to represent the same concept.

(3) Limitations related to information modelling: Several researchers have discussed the link between the physical measurements of defects in the inspection process and the resulting condition assessment (or severity evaluation) in the diagnosis process, and the following decisions about the 3R actions. However, most of the previous research focused only on the modeling of defects. Therefore, more research is needed for modeling the other aspects of inspection, diagnosis and 3R information.

(4) Lack of comprehensive modelling: Some of the previous research focused on a specific inspection technology and the IRIM was developed only to demonstrate that technology (e.g. Kim et al. [96]).

2.6 Ontology Approach

One of the most widely used definitions of ontology is explicit shared knowledge and conceptualization of the domain [201, 202]. Another definition of ontology by Gaševic et al. [203] is that the ontology contains and presents two main elements: the related vocabulary to the domain of interest, and the knowledge representation using this vocabulary to describe this domain. The ontology, in simple words, is a set of relations between a set of concepts as shown in formula (1) [204].

$$\Omega = \{C, R\} \tag{1}$$

Where Ω is the ontology, C is the set of concepts of this ontology, and R is the set of relations between these concepts. The main types of concepts are: (1) Entities (e.g. project, operation, task, process, product, resource, and actor); (2) Attributes: Each entity has some attributes that make it different from other entities of the same type; (3) Relationships: El-Gohary and El-Diraby [205] classified the main types of relations among concepts as subsumption relations and partonomy relations. A subsumption relation reflects the is-a relationship between the concepts and is used to represent the relation between the general concept and a sub-concept. A Partonomy relation is a part-of relationship between the concept and its parts, which are built as patronymic hierarchies;

(4) Axioms: Axioms can be used to model and describe some constraints such as regulations, best-practices and client requirements; (5) Strategies: Strategies refer to the methods that are used to accomplish the operations and tasks in the project; and (6) Modalities: A modality is used as an umbrella to cover a variety of operation states and the conditions that describe them, such as stage modality, temporal modality, and situation modality. Stage modality can be used to describe a process belonging to one of the lifecycle phases (i.e. initiating, design and planning, construction, monitoring and control, and decommissioning). The steps for developing an ontology are: (1) Defining the purpose of the ontology (i.e. needs, scope, and users); (2) Building the taxonomy of concepts and their interrelations; (3) Developing the process model based on the taxonomy; (4) Ontology capturing and coding, where the terms referring to the relations and axioms are defined; (5) Ontology evaluation based mainly on experts' interviews.

2.6.1 Ontology Languages and Tools

There are different tools and languages, which are used to build ontologies. Protégé [206], and OntoEdit [207] are two examples of the ontology's editing environments [208]. Protégé provides various plug-ins that facilitate editing and visualizing the developed ontology. The Web Ontology Language (OWL) and Resource Description Framework (RDF) are examples of the languages that are used for representing the ontologies in human and machine-readable formats [209]. These languages provide a description of the complex relationships between the concepts in the ontology [210]. OWL provides the ability to describe complex concepts based on simpler ones available in the ontology. It has a reasoner that can be used for checking the consistency of the concepts defined in the ontology. Ontologies typically can be developed as XML-based files and can be represented in a computer using logic languages such as Knowledge Interchange Format (KIF) [203]. KIF is like the First Order Logic (FOL) and can provide the encoding of knowledge using a variety of logical operators.

2.6.2 Review of AEC/FM Ontologies

Ontologies are usually used for a specific domain to facilitate a specific application. In construction, ontologies are used to organize and represent the shared knowledge between the different entities in the domain in a way that can handle the dynamic construction environment. Ontologies are used to overcome the complexities and create interaction between the different disciplines at different levels of construction projects. In this context, ontologies are used for different proposes, such as improving safety and enhancing quality. One of the advantages of the ontology approach is that inferring new logical knowledge from a set of stated axioms can be directly applied by reasoning engines [211].

The entire IFC schema is available in a large ifcOWL ontology, representing building data using semantic web and linked data technologies [212]. In addition to ifcOWL ontology, many ontologies have been developed in the AEC/FM industry. Ding et al. [213] proposed a framework for risk knowledge management in BIM to improve the construction risk analysis process. In this framework, the ontology is used to model the risk knowledge and create the linkage between objects in BIM and the risks. Dibley et al. [214] proposed a framework for building monitoring using ontology and multi-agent systems (MAS). The framework supports real-time monitoring of the indoor environment using sensory data collected from various sensors. Adeleke and Moodley [215] proposed a framework to monitor and control air quality in the indoor environment, where data about the air and human activities are collected using sensors. Cacciotti et al. (2014) presented an ontology for the diagnosis of damage in order to process and manage cultural heritage damage information. However, the detailed taxonomies for damages and the causes of damages were not developed in their ontology. Moreover, their approach was domain-specific, and it is not widely applicable for other types of structures [196]. Jung et al. [216] proposed an ontological approach to infer the causes of concrete cracks. However, their study was limited to crack defects, and their proposed approach does not support BIM. Lee et al. [217] developed a linked data system framework for sharing construction defect information using ontologies and BIM environment.

El-Gohary and El-Diraby [205] proposed an infrastructure and construction process ontology that offers a formal representation of the process knowledge in the infrastructure and construction domain. El-Diraby [218] presented a domain ontology of construction knowledge, which contains the key terms' conceptual architecture, relationships, and behaviors in the construction domain. Park et al. [185] briefly discussed the benefits of developing an ontology for proactive construction defect management. Venugopal et al. [219] proposed an ontological approach to building information model exchanges in the precast/prestressed concrete industry. Zeb and Froese [220] developed a transaction ontology in the domain of infrastructure management. Hammad et al. [109] developed an ontology for modeling lifecycle inspection and repair information of civil infrastructure systems. Stroga et al. [221] proposed a taxonomy of relations in the product and design engineering knowledge domain. In their study, more than 40 relations containing seven main classes of compositional relationship, spatial relationships, role relationship, general relationship, dependency relationship, influence relationship, and temporal relationship are suggested. Table 2-4 shows the relations that can be used in the domain of product and design ontology.

Table 2-4. Relationships in product/design engineering ontology (adapted from [221])

Compositional relations	Spatial relations	Role relations	Dependency relations	Influence relations	Temporal relations	General relations
component of	has direct contact to	instrument	aim/purpose/reason	influence	after/follows	alternative
element of	has non-direct contact to	operand	base of	is opposing	before/ proceeds	criteria
material of	interacts with	operator	cause/factor/ stimuli	is supporting	co-occur	delivers
member of	contains	resource	consequence/ response/ result	-	-	has as an attribute
portion of	-	input	depends on/ presumption for	-	-	represents
-	-	output	-	-	-	realizes
-	-		-	-	-	satisfies

Moreover, many minimal ontologies were created for the linked building data purpose. Linked building data principle means using a web-compatible standard for the exchange of web-based information [179]. The purpose of the linked building data method is to present and identify elements and damages using HTTP URIs (Uniform Resource Identifiers) in order to connect a query language or a reasoning engine. Since this method describes the topology of a building without its geometry, it can be used to connect to other data sets [222].

BEO (Building Element Ontology) [223] and MEP (Mechanical, Electrical, and Plumbing) ontology [224] are two ontologies extracted from the IFC schema. These two ontologies do not include any relations, which can be used based on user requirements in different domains. Hamdan et al. [222] proposed Damage Topology Ontology (DOT), which is a small high-level ontology to describe any type of damage topology in general. Later on, Hamdan et al. [196] proposed a small ontology called Concrete Damage Ontology (CDO) to define some damages in concrete structures.

Rasmussen et al. [225] presented an Ontology for Property Management (OPM), which is a minimal high-level ontology for managing changes and property valuation over time. In another study, Rasmussen et al. [179] proposed BOT, which is a minimal ontology to describe building stories and space topology. Wenger et al. [226] developed Ontology for Managing Geometry (OMG) to connect the geometric description to the building element. Bonduel et al. [227] developed Ontology for Geometry Formats (FOG) to exchange descriptive geometric data.

On the other hand, many studies focused on integrating the ontologies with BIM information. Niknam and Karshenas [228] proposed BIM Shared Ontology (BIMSO) based on UNIFORMAT II (ASTM 2020) classification system to be extended with different building domain ontologies. For example, they proposed BIM Design Ontology (BIMDO) to extend BIMSO for expressing the design properties of building elements. Zhong et al. [229] proposed an ontology-based framework to support interior and exterior environmental monitoring and compliance checking. The framework integrates the building information from BIM, sensor data, and the related regulations, information and the design requirements. Zhu [230] developed TCEI-Ontology, which is the integration of time, cost, and environmental impacts to support multi-objective integrated analyses. TCEI-Ontology reused many of the available concepts from IFC because it has semantic-rich information related to the AEC/FM industry. In this regard, seven fundamental steps were considered to develop a multi-objective integrated ontology: (1) specifying the scope and domain of ontology, (2) adapting the available ontologies, (3) itemizing influential terms in the ontology, (4) specifying the hierarchy of classes, (5) specifying the properties of a class, (6) specifying facets which are properties to constrain property values, and (7) generating instances. Kim et al. [231] proposed an ontology to integrate FM maintenance work information of traditional FM system database and BIM-based data. Wang and Issa [232] proposed extracting relevant IFC information from the BIM model and integrating it with GIS ontology to create an integrated ontology. Table 2-5 shows the ontology metrics of some of the publicly available BIM-based ontologies.

Table 2-5. Ontology metrics of some of the publicly available BIM-based ontologies (adapted from [233])

Ontology	Metrics			
	Classes	Relations	Attributes	Individuals
BOT v.0.3.2 [179]	10	16	1	5
FOG v.0.4 [227]	3	14	119	4
ifcOWL v.4.1 [212]	1360	1644	5	1171
DOT v.0.8 [222]	13	13	3	1
MEP v0.1.0 [224]	484	0	1	1
OMG v.0.3 [226]	8	17	2	0
BEO v.0.1.0 [223]	183	0	1	1
OPM v.0.1.0. [225]	17	8	4	1

2.6.3 Ontologies for Robots

In the robotic area, ontologies can be used for different applications such as general robotic purposes (e.g. standardization [234, 235], and ontologies for autonomous robots (e.g. description/reasoning about the environment and tasks [9, 236]. The robotic system utilizes and processes the ontology as the robot's central data store [8]. To accomplish the tasks correctly, the autonomous robot needs to deal with high-level semantic data along with low-level sensory-motor data. Robot Operating System (ROS) [237] can also use several navigation methods, such as Lidar Odometry and Mapping (LOAM) and Simultaneous Localization and Mapping (SLAM), which help the robot to build its map based on the collected data about the environment [238].

Robotic ontologies embody the real-world description of objects, properties, and relationships in the domain [239]. KnowRob is an OWL-based robotic ontology that contains a small core system and a large set of optional modules, which are developed to perform human activities in a building. The KnowRob ontology v.1.0 has 742 classes, 176 relations, 119 attributes, and 23 individuals [132]. Bouguerra et al. [240] utilized semantic knowledge including environmental object description for execution monitoring of the robots in the indoor environments. Chella et al. [241] used a multi-perspective approach to represent qualitative and quantitative knowledge of an office indoor environment for robot operation. Wood [242] discussed the importance of combining task-oriented description with object-oriented knowledge of ontologies for autonomous robots. Robotic autonomy can benefit from ontologies by extracting the knowledge needed to execute the task. In addition, using ontologies can help in defining the constrains of the behavior of the robot based on specified knowledge (e.g. safety policies) [243]. Ontologies have also been used for autonomous object recognition of robots based on the description of objects characteristic and visual concepts (e.g. color, texture) or images features [244- 246].

Habib and Yuta [247] used hierarchical map representation of the robot's environment for collision-free path planning of the robot considering three structure levels of building, corridor, and room. In their work, all the objects are assumed to be flat and they did not consider vertical operational direction. Balakirsky and Scrapper [248]. Provided the robotic knowledge requirements for the basic development of a collision-free path. Understanding and representing knowledge about the obstacles in the ontology will enhance robot navigation to avoid potential accidents [249, 250]. Schlenoff et al. [251] developed an ontology for autonomous robots to assess the damage of collision with various objects during the navigation. Schlenoff and Messina [252] presented the basic requirements for search and rescue robot ontology considering the integration of sensing and navigation. Description knowledge can be extended by rules to be used as a knowledge representation for the sensor-based understanding of multifarious tasks [253].

2.7 LiDAR-based Defect Detection

LiDAR scanning is a non-contact measurement technology that has proven its potential in capturing accurate and instant point cloud data from object surfaces [254, 255]. However, the resolution and noise level of point cloud data pause some challenges in detecting small cracks [256]. Therefore, to overcome this limitation, an additional feature, which is the RGB color, is considered in deep learning models [11, 28]. Various methods have been applied to the point cloud data to detect surface defects. Geometry analysis and machine learning methods are two main approaches for detecting concrete surface defects.

Gaussian curvature distribution can be used to calculate volume loss [257]. Another method to detect concrete surface defects is the crossing section method [258]. Laefer et al. [23] used fundamental mathematics to define the smallest width of unit-based masonry cracks, which can be detected with LiDAR scanner by considering the main parameters of depth and orientation of crack, orthogonal offset, and interval scan angle. Anil et al. [259] focused on the performance of LiDAR scanners by using an automated algorithm on point cloud data from reinforced concrete surfaces and asserted the possibility of detecting 1 *mm* crack based on point cloud data. Xu and Yang [260] used the Gaussian filtering method and image-generated data from the point cloud to detect the cracks of a concrete tunnel structure. Teza et al. [257] proposed an automatic method for the inspection of damaged areas of concrete bridge surfaces using a LiDAR scanner and Gaussian mean curvature computation. Makuch and Gawronek [261] proposed an automatic inspection system for reinforced concrete cooling tower shells using point cloud data and local surface curvature computation. Olsen et al. [258] proposed using cross sectional analysis to detect surface damage based on LiDAR scanner data. Liu et al. [262] utilized the distance and gradient-based method to detect the defective area of bridge surfaces using laser scanner data. An automated classification algorithm for detecting historical building defects is suggested by Armesto-Gonzalez et al. [263]. Valença et al. [264] proposed a method combining image processing and LiDAR scanning technology to automate the process of capturing the geometrical characteristics of cracks on concrete bridges. Kim et al. [265] proposed a technique to indicate the location and measure the quantity of concrete surface spalling defects larger than 3 *mm* using LiDAR scanner data. Truong-Hong et al. [255] presented an approach to detect the bridge cracks using a LiDAR scanner and developed a tool to measure the length and width of cracks based on point cloud data and RGB color produced from an external camera. Tsai and Li [266] assessed the probability of using point cloud data to detect cracks with the dynamic-optimization-based segmentation method and assess the crack segmentation performance using the linear-buffered Hausdorff scoring method. Cabaleiro et al. [267] developed an automatic crack detection algorithm using LiDAR data for timber beams inspection to identify the crack geometrical characteristics. Mizoguchi et al. [268] proposed a customized region-growing algorithm along with an iterative closest point algorithm to detect the surface defects of concrete structures based on LiDAR scanner data. Nasrollahi et al. [269] proposed a method for detecting concrete surface defects based on collecting point cloud data from LiDAR scanners and using a Deep Neural Network (DNN). Guldur et al. [11] proposed a method to detect the defects using point clouds' intensity and RGB values to define a threshold and extract the defect's geometrical features [11]. However, their method was not based on using the point cloud geometrical features in the detection process and was not suitable for complicated structures [270]. Guldur and Hajjar [271] developed damage detection algorithms for automatic surface normal-based defect detection and quantification using LiDAR scanner data.

So far, different deep learning methods have been used to identify concrete surface defects using images, and progress in this area has reached an acceptable level [272]. Image-based methods are usually affected by the consistency and stability of light conditions for the captured images, and these methods are usually suitable only for simple flat surfaces [20, 22]. Table 2-6 compares some examples of the results of image-based deep learning methods for concrete surface crack detection.

Although the papers discussed in this section have significant value in the field of defect detection, there is no deep learning method for semantic segmentation of concrete surface defects using raw point cloud data.

Table 2-6. Examples of the results of image-based deep learning crack detection methods

Author	Year	Paper	Surface material	CCN input data	Precision (%)	Recall (%)	F1-Score (%)
Yang et al. [273]	2018	Automatic pixel-level crack detection and measurement using fully convolutional network	Concrete wall, pavement	Image	81.73	78.97	79.95
Liu et al. [274]	2019	Computer vision-based concrete crack detection using U-net fully convolutional networks	Concrete	Image	90	91	90
Ali et al. [275]	2021	Performance evaluation of deep CNN-based crack detection and localization techniques for concrete structures	Concrete	Image	99.7	85	91.8
Le et al. [276]	2021	Development of deep learning model for the recognition of cracks on concrete surfaces	Concrete	Image	96.5	98.8	97.7
Vignesh et al. [277]	2021	Concrete bridge crack detection using convolutional neural network	Concrete	Image	96.69	99.55	98.1

2.7.1 Deep Learning Approach

Deep Neural Networks (DNNs) or deep feedforward networks utilize multiple deep layers along with highly optimized algorithms to learn from trained data sets without the process of manual feature extraction [278]. A DNN is formed by linking many functions. The DNN hidden layers represent the functions as the links in the network. The length of all links determines the network's depth [279]. A DNN attempts to map a function $y=f(x;\theta)$ through learning the θ value from the optimal value function f over the x value as an input. Training samples are dispersed in mini-batches. An epoch is a single pass through all the training samples, and a step or iteration refers to the training process over a mini-batch. Furthermore, the sum of the size of all data samples divided by the number of data samples in a mini-batch (batch size) equals the number of steps in an epoch.

A Convolutional Neural Network (CNN) is a class of DNNs containing input, convolutional, subsampling, and output layers [280]. Each CNN layer receives inputs through the previous layer's local receptive fields. Neurons extract basic geometric features such as edges, boundaries, and corners using local receptive fields, which are also known as filters or kernels. A feature map would be created by convolving a kernel with a unique weight vector over the entire image, and a feature detector that works for a part of an input image is supposed to work for the entire image. In order to extract multiple features from the input image, a convolutional layer creates multiple feature maps [280]. Each feature map has the same weight vector, which will enhance through the backpropagation process.

Traditional neural networks utilize matrix multiplication on the entire input data in one point to generate an output unit, but CNN utilizes a kernel that is smaller than the input and results in fewer operations [279]. In the parameter sharing process, a weight vector is used over and over in a CNN's layer to compute a layer's results. Parameter sharing has no impact on forwarding propagation time but significantly impacts computational resources and model efficiency [281]. Parameter sharing reduces the number of free parameters, which ultimately enhances a model's generalization capability [282]. In terms of computation volume, CNNs outperform traditional neural networks significantly. The first step of a CNN is the operation of multiplying a weight vector by the input data by convolving a kernel and establishing linear activations in a feature map [281]. The other two stages are nonlinear activation and subsampling feature mapping utilizing pooling functions.

2.7.2 Semantic Segmentation of Point Clouds

The use of point cloud-based deep learning methods is a breakthrough in identifying concrete surface defects in 3D points data. Point cloud-based deep learning methods are currently in their early stages, and very little research has been done in this area [272]. Data quality is critical in determining the best fitting function for any neural network. The datasets should reflect the appropriate parameters and provide various cases depending on the requirements. As a result, gathering sufficient datasets is required to obtain an accurate model. A point cloud is a set of data that includes the geometric information of sparse points collected in three dimensions. RGB and density information could also be included in point cloud datasets. Pixel-based, voxel-based, and 3D point-based approaches are three main categories of CNN approaches based on data representation. 3D data is transformed into 2D representation in pixel-based approaches [283]. In voxel-based approaches, the 3D points are used to create voxels [284]. Qi et al. [285] presented 3D point-based approaches to process point cloud data using 3D CNN and utilizing 3D recognition tasks such as object classification, part segmentation, and semantic segmentation. Since pixel and voxel-based methods are more common than point-based methods, 3D point cloud datasets are frequently converted into images or 3D-voxel grids before being used in deep learning. In certain circumstances, the transformation produces enormous data with uncertain invariances. Furthermore, point cloud data are easier to learn because of their clarity and consistent structure, but meshes are complicated and contain contradictory compositional patterns [286]. Although there are increasingly more recently developed networks that surpass PointNet performance on various datasets, PointNet remains a benchmark for point cloud semantic segmentation studies [287].

PointNet is a cutting-edge CNN model for point cloud research that can be utilized straightly in classification, part segmentation, and semantic segmentation using multi-dimensional NumPy Python library arrays. PointNet was established in 2017 to address issues involving displaying point clouds and voxelization [286]. PointNet's input is a set of points with three major features, which are considered when creating the PointNet architecture. First, these points are not in any particular sequence. The relationship of adjacent points is the next most essential feature of such datasets. The points are not apart from the other points, and the semantic content of the data points is influenced by the local structure of the composition of adjacent points. The third aspect of point clouds is their invariance under transformation. The semantic scene segmentation of the PointNet network was validated using the Stanford Large-Scale 3D Indoor Spaces dataset (S3DIS) [286, 288].

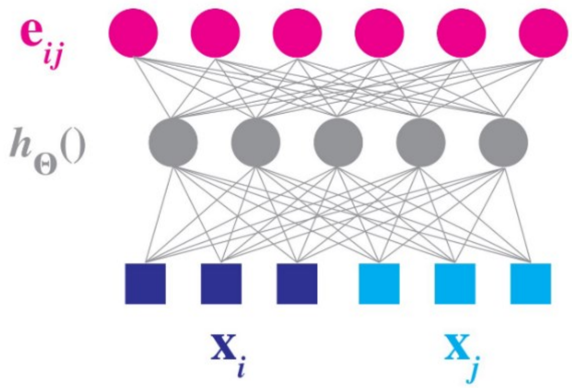
2.7.3 Local Feature Learning on Point Sets

PointNet++, the second version of PointNet, outperforms PointNet for point cloud semantic segmentation because it extracts the point's local characteristics utilizing multi-scale sampling [28, 289]. In order to feed the CNN, the PointNet network normalizes the number of points in pre-set geometrical blocks. On the other hand, points have diverse densities in different sections of a point cloud, which may impair the segmentation process and result in the loss of valuable information. It is necessary to study the feasible minor groups of point sets to achieve the ideal aim of capturing every class's important features in point clouds. In this regard, PointNet++ [289] feeds the CNN a mixture of non-uniform density data points. Moreover, considering a greater sample of the data points is also important, as the small sample of the data points in low-density areas does not provide valuable information.

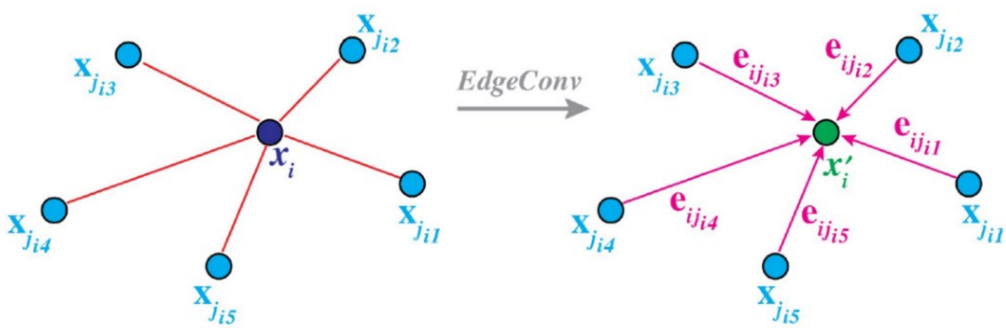
PointNet++ can extract the point's local features using multi-scale sampling [289]. The multi-scale sampling is extremely powerful and useful for semantic segmentation with several labels, particularly when segmenting small objects. The network extracts local features from small samples of data points. Then the network groups the small samples into greater samples. This procedure is repeated until all of the point set's local features have been extracted. However, Pointnet++ still treats individual points in local point sets independently.

The Dynamic Graph CNN (DGCNN) is a recent network proposed by Wang et al. [27]. It is a new point-based CNN suitable for high-level tasks, such as object classification and semantic segmentation. DGCNN can improve capturing local geometric functions as it creates a local neighborhood graph and dynamically updates the graph with the nearest neighbors after each layer of the network. Compared to PointNet++, DGCNN rather than operating on individual points, iteratively performs convolution on edges, associating the neighborhood point pairs.

Operation layer for edge feature generation in DGCNN is called EdgeConv, which can define the relationships between a point and its neighbors [27]. Figure 2-10 shows the mechanism of DGCNN edge feature generation. As shown in Figure 2-10(a), X_i and X_j are a point pair, and e_{ij} is $h_\theta(X_i, X_j)$, which is the edge feature function; h is the function parameterized by the set of learnable parameters θ. Figure 2-10(b) shows the aggregation operation on the edge features associated with all the edges originating from each vertex, where X_i' is the EdgeConv operation, which is defined by applying aggregation operation at the *i-th* vertex.



(a) Computing an edge feature e_{ij} from a point pair X_i, X_j



(b) Aggregation operation on the edge features associated with all the edges originating from each vertex

Figure 2-10. Mechanism of DGCNN edge feature generation [27]

The segmentation model of DGCNN involves a series of three EdgeConv layers and three fully connected layers. The parameter K in the model is the number of the edge features for each point, which is computed in each EdgeConv layer for the input of n points. Edge feature is the most important feature in concrete surface defect semantic segmentation. Wang et al. [27] stated that the model with their developed DGCNN improved the accuracy for classification task in comparison of PointNet++ for the same ModelNet40 [290] dataset. Furthermore, for the semantic segmentation task, they used Stanford large-scale 3D indoor spaces dataset (S3DIS) [291] and compared their work with PoinNet, for which their work achieved a higher accuracy. In another study, Pierdicca et al. [292] compared the performance of PointNet++ and DGCNN for semantic segmentation of historical architectural elements. They used a publicly available digital cultural heritage dataset with 11 labeled points clouds. Table 2-7 shows the results of the tests performed to compare the performance of PointNet++ and DGCNN.

Table 2-7. Results of the tests performed on an unknown scene based on different network [292]

Network	Test Accuracy	Precision	Recall	F1-Score
DGCNN	74%	77%	74%	74%
PointNet++	53%	53%	53%	48%
PointNet	35%	54%	35%	27%

2.8 Summary

This chapter reviewed the concepts, technologies, and methods that are used in the current research. The literature review covered the review of concrete infrastructure management and different types of concrete surface defects. Inspection information modeling and limitations of previous research were explained. Furthermore, it explained the ontological approach and the related concepts of BIM-based inspection and robotic inspection and navigation tasks. The machine learning approach was also covered in this chapter, including object detection in point clouds dataset and deep neural network applications in semantic segmentation.

Based on the reviewed literature, it is concluded that the main usage of BIM at the time being is limited to the design and construction phases of the lifecycle. Although several studies have proposed extending the usage of BIM to model defects in the construction and O&M phases, the research in this area is still limited and fragmented. In order to take full advantage of BIM throughout the infrastructure's lifecycle, inspection-related information should be integrated with BIM to have a semantically unified knowledge model where all the stakeholders can access information in a systematic manner. Furthermore, robotic inspection needs to be integrated with the construction domain's knowledge to be more effective and efficient. Moreover, LiDAR scanner is an emerging technology to detect surface defects. The literature review indicated that utilizing BIM-based inspection-related knowledge along with an integrated process of surface defect semantic segmentation and defect modeling have the potential of enhancing the efficiency of the inspection process.

CHAPTER 3. RESEARCH FRAMEWORK

3.1 Introduction

An overview of the proposed framework is presented in this chapter. In addition to the ontological approach, explained in Chapters 4 and 5, Chapter 6 focuses on the machine learning approach for inspecting surface defects.

First, the research framework for an ontology for concrete surface defects is explained in Section 3.3. Then, the research framework of an ontology for BIM-based robotic navigation and inspection tasks is explained in Section 3.4. Finally, Section 3.5 explained the research framework for point cloud-based concrete surface defect semantic segmentation and as-inspected modeling. Detecting accurate defects is the main objective of this section. Understanding the accurate information about defective areas will help to perform further actions more efficiently.

3.2 Overview of the Research Framework

This section gives a brief overview of the proposed framework, which consists of three parts. The first part uses an ontological approach dedicated to developing ontologies, including an ontology for concrete surface defects and an ontology for BIM-based robotic navigation and inspection tasks. The first part uses an ontological approach to develop an ontology for concrete surface defects in the first phase, and the next phase is as-inspected modeling, which is dedicated to updating the as-is BIM model with analyzed defect information. The second part is dedicated to developing an ontology for BIM-based robotic navigation and inspection tasks. The third part focuses on one of the remarkable robotic inspection methods, LiDAR-based defect semantic segmentation using NVE-DGCNN. This part uses a machine learning approach for the inspection of surface defects using a LiDAR scanner. The last part is as-inspected modeling, which is dedicated to updating the as-is BIM model with analyzed defect information. An overview of the framework is shown in Figure 3-1.

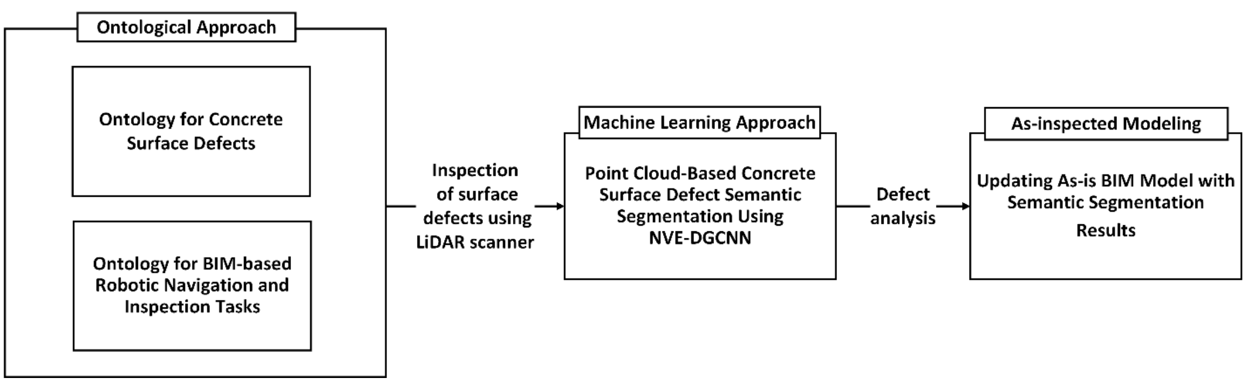


Figure 3-1. Overview of the proposed framework

3.3 Ontology for Concrete Surface Defects

The BIM model will evolve during the different phases of the lifecycle to reflect the changes related to quality inspection and repair processes during the construction phase, as well as the inspection and maintenance processes during the O&M phase. As discussed in Section 2.5.2, In order to take full advantage of BIM or CIM throughout the infrastructure's lifecycle, related defect information should be integrated with BIM to have a semantically unified knowledge model where all the stakeholders can systematically access information in a systematic manner. Chapter 4 aims to develop an ontology for concrete surface defects (OCSD). The developed ontology covers four main groups of concepts: (1) inspection concepts, (2) diagnosis concepts, (3) 3R (Repair, Rehabilitation, and Replacement) concepts, and (4) defect concepts. Figure 3-2 shows the main types of OCSD concepts. The proposed method for developing OCSD uses the general approach and tools discussed in Section 2.2 and is based on the following steps: (1) defining the competency questions by analyzing the previous related research to identify the common aspects and limitations of available models; (2) identifying the steps for developing the ontology at a level of abstraction that can be applied to different structures/materials; and (3) extending the basic ontology to cover all the requirements defined in Step 1. Chapter 4 will provide an in-depth discussion of developing OCSD.

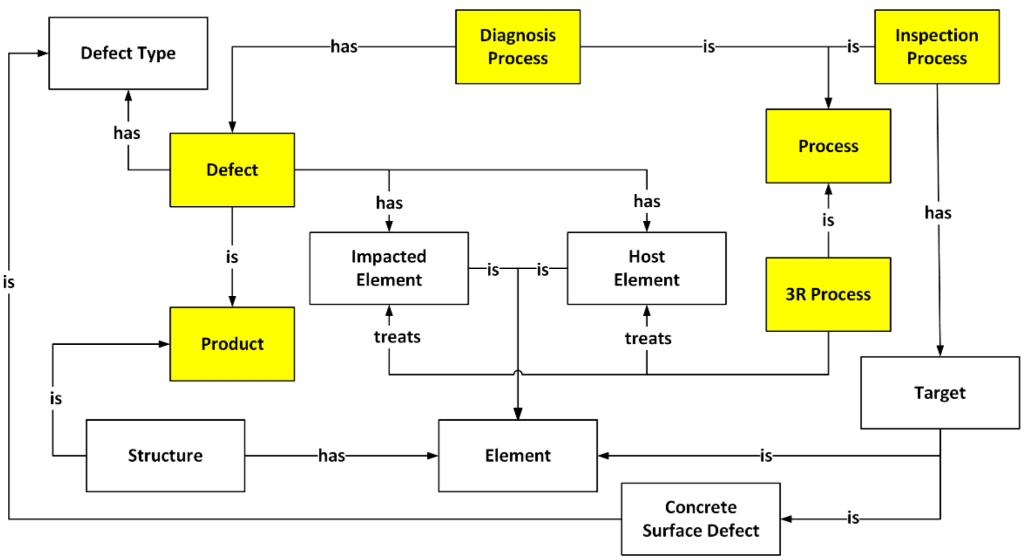


Figure 3-2. Main groups of OCSD concepts (main entities marked in yellow)

3.4 Ontology for BIM-based Robotic Navigation and Inspection Tasks

As discussed in Section 2.5, several studies have used BIM for navigation purposes. Also, some studies focused on developing a knowledge-based ontology to perform activities in a robotic environment (e.g. CRAM). Chapter 5 aims to integrate robotic inspection with the knowledge of the construction domain. In this regard, an Ontology for BIM-based Robotic Navigation and Inspection Tasks (OBRNIT) is developed. This ontology can help system engineers involved in developing robotic inspection systems by identifying the different concepts and relationships about robotic inspection and navigation tasks based on BIM information. The developed ontology covers four main types of concepts: (1) robot concepts, (2) building concepts, (3) navigation tasks concepts, and (4) inspection tasks concepts. Figure 3-3 shows the main types of OBRNIT concepts. Developing an integrated ontology is a first step towards logic-based inspection. The use case is an inspection robot that is navigating in a building with partial knowledge of the environment because of changes in the available information due to construction and renovation scheduling issues, unexpected obstacles in the building, etc. As shown in Figure 3-4, to define the requirements of OBRNIT, UML (Unified Modeling Language) use case diagram is presented. The actor is a robot, and the associations between the actor and the use cases are shown with solid lines. Dependency relationships are shown with dotted lines. *Includes* relationships indicate that the involved use case is a part of the base use case. *Extends* relationships indicate that the base use case does not depend on the extending use case, and specific criteria are needed for the occurrence of the extending use case. The details of the proposed method will be explained in Chapter 5.

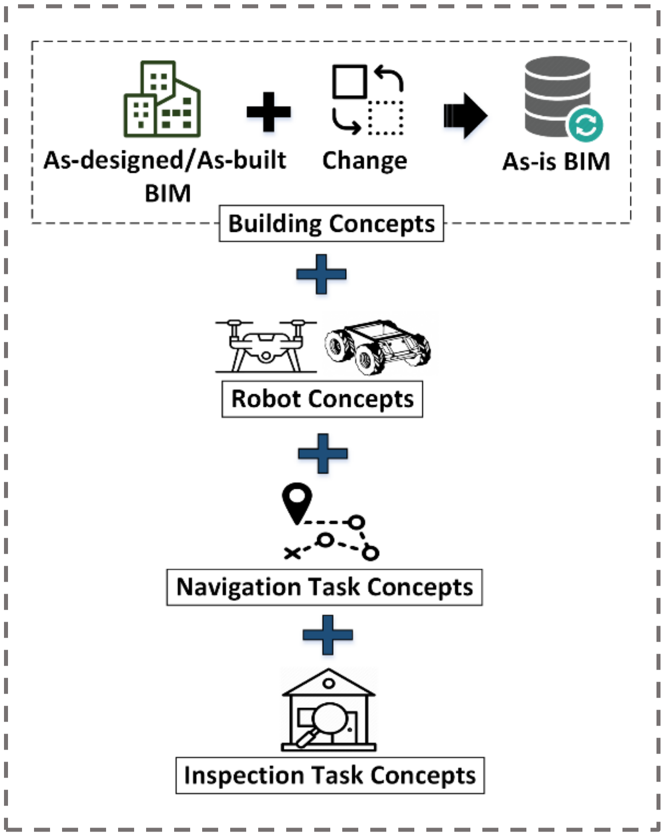


Figure 3-3. OBRNIT main types of concepts

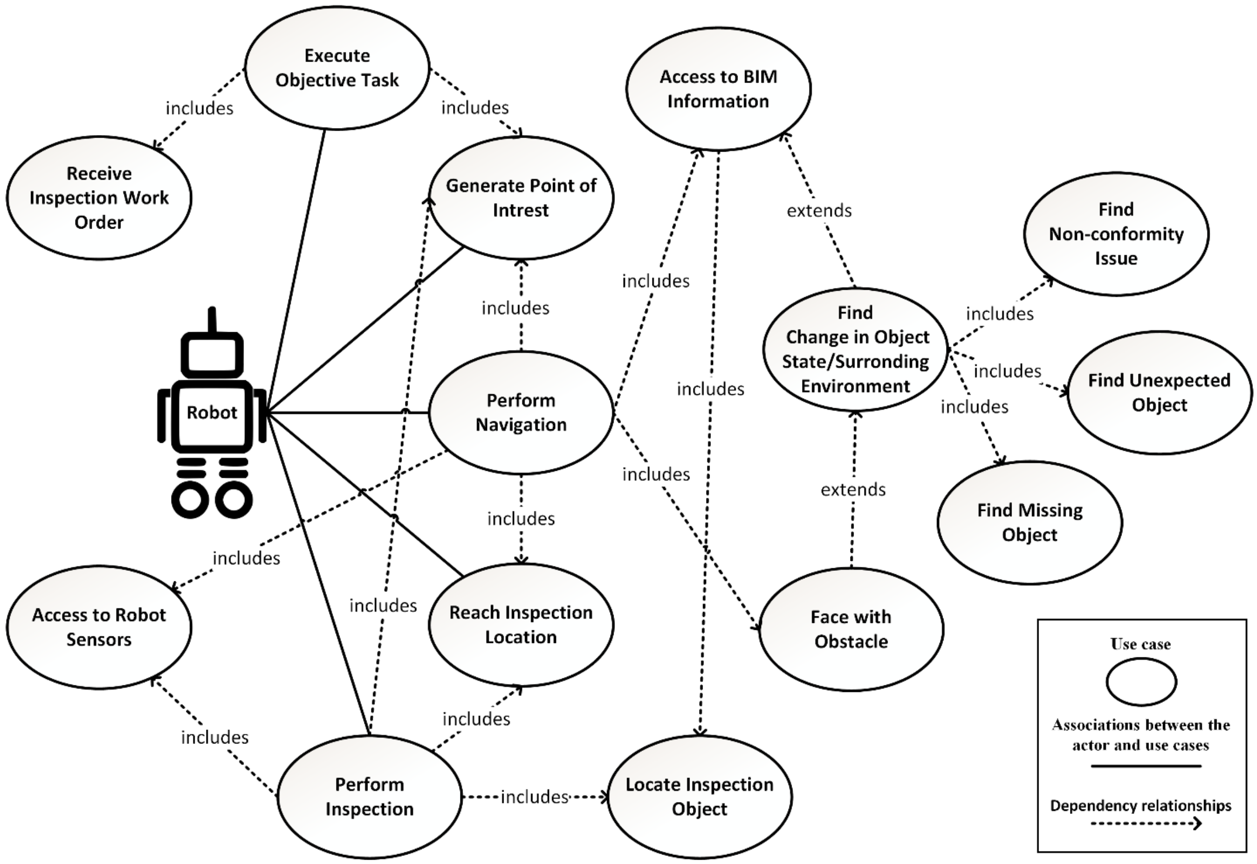


Figure 3-4. Use case diagram of OBRNIT

3.5 Point Cloud-Based Concrete Surface Defect Semantic Segmentation and As-Inspected Modeling

LiDAR scanners can collect high-quality 3D point cloud datasets. In order to automate the process of concrete surface inspection, it is important to collect proper datasets and use an efficient approach to analyze them and find the defects. Deep Neural Networks (DNNs) have been recently used for detecting 3D objects within point clouds. Based on Section 2.7.2, in order to detect concrete surface defects, the CNN approach can be applied on point cloud datasets. Each neural network is trained for a unique purpose. Adaption of the right algorithm to a specific purpose can greatly improve the performance.

Chapter 6 starts with investigating the adapted PointNet++ in the first phase. Then the DGCNN's [27] ability to detect the edges is considered in the next phase. For the DGCNN, the work started with an adapted DGCNN and then the main network of this research, which is NVE-DGCNN was investigated. DGCNN is a deep neural network for classification, part segmentation, and semantic segmentation of point clouds, which is modified and adapted in this study to detect concrete surface defects. This algorithm is originally designed to detect indoor building elements. The semantic segmentation of DGCNN is adapted to detect surface defects using point cloud datasets from scanning concrete bridge surfaces. DGCNN can improve capturing local geometric functions as it creates a local neighborhood graph and dynamically updates the graph with the nearest neighbors after each layer of the network. DGCNN, rather than operating on individual points, iteratively performs convolution on edges associating the neighborhood point pairs. As the edge is an important feature of the surface defects (e.g. cracks), using a deep learning method that can consider the edge feature and the relationship between neighboring points can improve the learning model's accuracy and efficiency. Figure 3-5 shows the overall framework of the proposed for point cloud-based concrete surface defect semantic segmentation using NVE-DGCNN. There are six main steps for the inspection of surface defects using LiDAR scanner: (1) data collection, (2) manual annotation, (3) data pre-processing, (4) training and evaluation, (5) testing, and (6) sensitivity analysis. The detailed framework of the proposed method will be explained in Section 6.2.2.

Chapter 6 also proposes a method that includes post-processing of semantic segmentation results for the automated as-inspected modeling purpose. The results of the defect semantic segmentation will be used to locate the defects in the BIM Model. As explained in Section 2.5.2.3, the current BIM information does not support the representation and integration of defect information. Integrating detected defect information with BIM will facilitate accessing and updating the inspected defect information at different phases of the lifecycle resulting in improved efficiency and reduced rate of data input errors. As-inspected modeling will help to store the inspection in an efficient and precise way. It can also enable the tracking and analysis of the changes throughout the lifecycle. The as-inspected BIM model not only contains the basic geometry of defects, but also semantic information about their type, severity, etc. Chapter 6 will provide an in-depth discussion of LiDAR-based defect semantic segmentation using NVEDGCNN and as-inspected modeling.

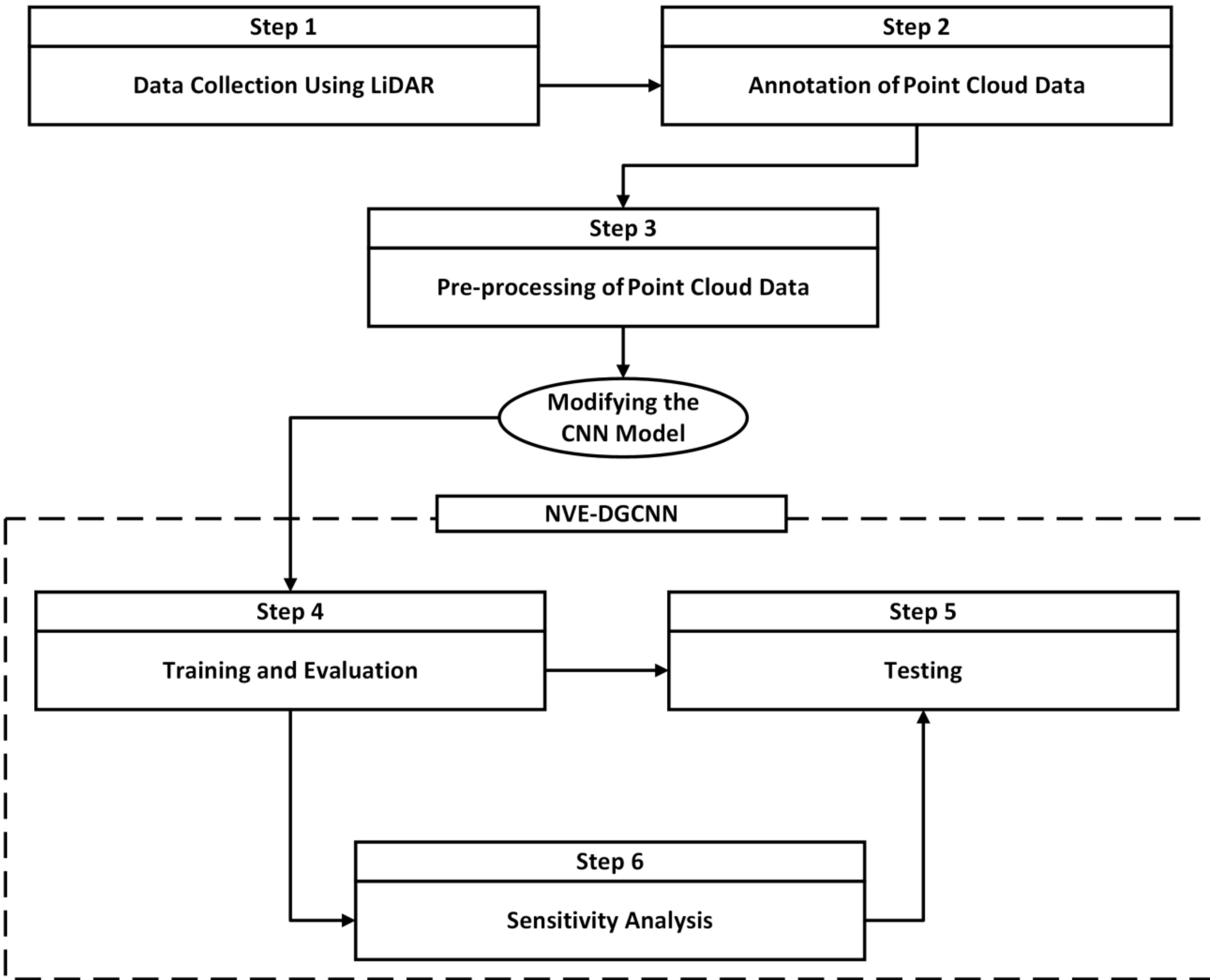


Figure 3-5. Overall framework of the proposed method for point cloud-based concrete surface defect semantic segmentation using NVE-DGCNN

3.6 Summary

This chapter provided an overview of the proposed methodology of this thesis. The ontological approaches for developing an ontology for concrete surface defects and an ontology for BIM-based robotic navigation and inspection tasks were explained. The LiDAR-based method, which is an automated measurement method for robotic inspection, was explained. Moreover, the as-inspected modeling approach considered the integration of inspected surface defects with the 3D model and updating the as-is BIM.

CHAPTER 4. ONTOLOGY FOR CONCRETE SURFACE DEFECTS

4.1 Introduction

The quality of buildings and infrastructure systems should be inspected for defects that are beyond the tolerance level, and the detected defects should be repaired. The main objective of this chapter is to develop an ontology to cover the different types of information and concepts related to the inspection, diagnosis, and repair, rehabilitation, and replacement (3R) of concrete surface defects. This chapter focuses on concrete surface defects regardless of the type of the structure and can be applied at different phases of the lifecycle (i.e. construction and O&M). The ontology is called OCSD (Ontology for Concrete Surface Defects).

The rest of chapter is organized as follows. In Section 4.2, competency questions for the ontology are defined. Then, in Section 4.3, the methodology workflow is discussed. Section 4.4 focuses on the development of the ontology. Finally, Section 4.5 and Section 4.6 present the evaluation of the ontology and the conclusions, respectively.

4.2 Competency Questions for OCSD

The competency questions are defined to clarify the requirements of the inspection, diagnosis, and 3R processes of concrete surface defects domain [293]. The following competency questions are defined for developing a unified ontology based on the reviewed literature and the limitations of previous research.

(1) OCSD should follow a top-down approach where the common aspects of defects are molded at a higher level so that they can be shared by several types of structures and used at different phases of the lifecycle. For example, reinforced concrete surface cracks are very similar in tunnels and bridges although they are caused by different types of loads. This modeling approach will not only avoid duplicating efforts but will also provide a better-quality model, which grasps the essence of IRIM and can be further extended to cover the specific details related to the specific type of structure and the phase of lifecycle.

(2) OCSD requires a comprehensive modeling. OCSD should cover as much details as possible about the generic aspects of the inspection, diagnosis, and repair processes (i.e. process modeling) and the resulting defect model (i.e. product modeling). This requires developing a clear taxonomy considering all the semantic relationships required for modeling.

(3) OCSD should satisfy the needs of the state-of-the-art infrastructure management systems and guidelines. OCSD should reflect the common aspects of guidelines at an abstract level that can be applied to the widest category of structures. On the other hand, it is expected that the product and process models that can be developed based on OCSD will influence the current infrastructure management practices by creating an opportunity to re-engineer the processes used in these systems and enhancing additional aspects of IRIM in these systems (e.g. defect modelling).

(4) OCSD should not be restricted to the resources available in the current modeling standard (i.e. IFC). In other words, OCSD can be used as a starting point to extend IFC. Therefore, before extending any BIM-based standard (i.e. IFC) for inspection purposes, it is necessary to understand the defects and inspection-related concepts at the abstract level.

(5) OCSD should have the ability to accommodate new data collection technologies. The amount of inspection data is expected to grow exponentially with the availability of new technologies (e.g.

LiDAR, photogrammetry, etc.). OCSD should support these technologies and provide the means to accommodate the collected raw data and the resulting inspection information.

4.3 Methodology Workflow

This section explains the main steps for developing an ontology for concrete surface defects. OCSD development methodology is METHONTOLOGY. METHONTOLOGY is a clear, mature, and well-documented method [294, 295].

As shown in Figure 4-1, the initial, development, and final stages are three main steps of ontology development in METHONTOLOGY. The best practices and knowledge in the inspection, diagnosis and 3R processes of concrete surface defects domain are used to develop OCSD.

Determining the scope and main concepts and taxonomies of OCSD are the steps that should be considered in the initial stage. The scope of OCSD is defined based on the competency questions defined in Section 4.2. Moreover, the required level of covered details and the size of development is considered in this step. In the step of defining concepts and taxonomies, the related knowledge to OCSD is gathered based on literature from many sources such as textbooks, research papers, and online resources. At all steps of this stage, communication with end-users and professionals and getting feedback are essential. The list of requirements not only helps in the defining scope step but also helps in other stages of development.

Constructing and verifying the initial structure of OCSD are considered in the development stage. The first step of this stage uses a formal language (e.g. OWL) to implement and represent the conceptual model. The formal language helps the ontology to be easily used by different systems. [296]. Based on the availability and maturity level of ontologies and to fulfill the competency questions defined in Section 4.2, OCSD is developed from scratch. In the next step of the development stage, ontology verification is technically examined based on the developed ontology's consistency checking and competency questions.

The final stage involves improving OCSD through experts' and end-users suggestions and real-world needs. Criteria-based evaluation method and a case study is used to evaluate OCSD. The entire ontology development life cycle involves knowledge acquisition, evaluation, and documentation. The final step is documenting the developed OCSD. The IDEF5 (Integrated DEFinition) [297] ontology description method is used to presents the details of input, output, control, and mechanism in each of the methodology steps (Figure 4-1).

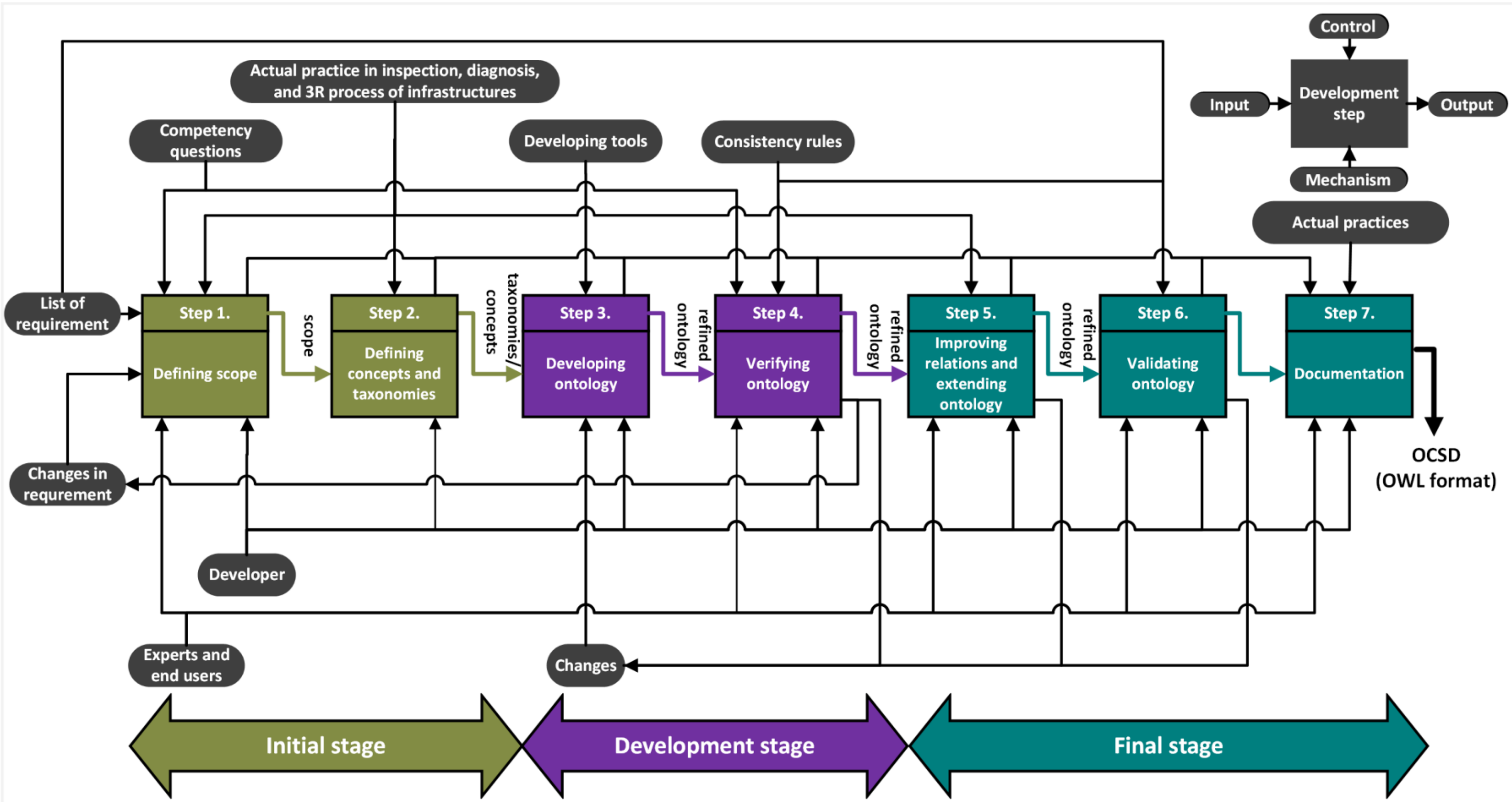


Figure 4-1. Development workflow of OCSD (adapted from Taher et al. [298])

4.4 Developing OCSD

A few concepts from CDO ontology [196], which is a small ontology and mainly developed for concrete structural damages, are used as parts of this study. OCSD is developed using Protégé [206]. OCSD has 333 classes, 51 relations, 27 attributes, and 31 individuals. The current version of OCSD is available at https://github.com/OCSD-OWL/OCSD.

OCSD covers five main groups of concepts related to process and product modeling, including: (1) inspection concepts, (2) diagnosis concepts, (3) 3R concepts, and (4) defect concepts, which are explained in the following sections. The concepts of ontology are semantically interrelated by the relationships defined between them. The types of relations used in OCSD are: is (e.g. *point cloud is collected data*), has (e.g. *inspection process has target*), uses (e.g. *remote sensing method uses LiDAR*), captures (e.g. *image sensor captures image*), performs (e.g. *inspector performs inspection process*), causes (e.g. *temperature change causes thermal stress*), affects (e.g. *environmental problem affects reinforcement expansion*), analyzes (e.g. *crack monitoring analyzes crack dimension*), evaluates (e.g. *condition assessment evaluates extent of damage*), determines (e.g. *condition assessment determines condition*), depends on (e.g. *condition depends on severity*), treats (e.g. *3R process treats host element*), and chooses (e.g. *3R process chooses repair material*).

4.4.1 Process Modeling Concepts

OCSD covers three main types of processes: (1) inspection concepts, (2) diagnosis concepts, and (3) 3R concepts, as explained below.

4.4.1.1 Inspection Concepts

Concrete surface *inspection* should be performed systematically and regularly to identify existing surface defects and detect possible future anomalies. The *inspection* concepts of OCSD cover the main concepts related to the *inspection* of *concrete surface defects*. Specific relationships are defined in OCSD to semantically interrelate different *inspection methods* and associated *inspection results*. Figure 4-2 shows OCSD *inspection process's* main concepts and relationships. The main *inspection* concepts and relationships in Figure 4-2 are summarized in Table 4-1. The *inspection process* has an *inspection method*, which can be *visual inspection*, *testing*, or a *method for measuring* defects. The information of the *inspector* and *inspection work schedule* is covered in OCSD. Some concepts are duplicated in Figures 4-2, 4-3, 4-4, and 4-5 to improve the readability of the figures. Furthermore, the main concepts are marked in yellow.

The *inspection method* can be chosen based on the order of complexity. As explained in Section 2.2.2, *measurement methods* for the *inspection* of concrete surface defects are *remote sensing methods* (e.g. *LiDAR*), *health monitoring* (e.g. *fiber-optic sensors*), or methods to measure *defects* (e.g. *crack*), *magnetic field*, and *environmental conditions* (e.g. *temperature, moisture, humidity*). The *collected data* depend on the *inspection method*. For example, a *visual inspection* will produce *images*, and an *inspection* using *LiDAR* will produce *point clouds*. *Post-processing of inspection* data includes *edge detection, shape extraction*, and *clustering*. *Inspection tools* (e.g. *binoculars*) and *measurement devices* are used during the *inspection* to accomplish the process. Measurement devices for *inspection* include *image sensors* (e.g. *RGB camera*), *LiDAR scanners*, etc. Several devices can be used for *crack measuring*, including *crack measuring magnifiers, crack width meters, vibrating wire crack meters, crack monitor gauges, crack measuring microscopes*, and *digital strain gauge deformation meters* [299].

Testing includes *destructive, semi-destructive*, or *non-destructive testing*. Moreover, *safety-related testing* is mainly using to determine the serviceability of existing or repaired concrete elements. As discussed in Section 2.2.2, the *testing* methods that can be used for the *inspection* of concrete surface defects include *rubber hammer test*, *half-cell potential test*, *initial surface absorption test*, and *colorimetric test strips*. Each of the measurement methods and *inspection* tests has a result that will be used in the *diagnosis process*. At the end of the *inspection*, the *inspector* will prepare an *inspection report*. *Inspection frequency* is another important factor that can help to detect the defects at early stage. The inspected data can be archived in a time series format that allows easy retrieval and processing.

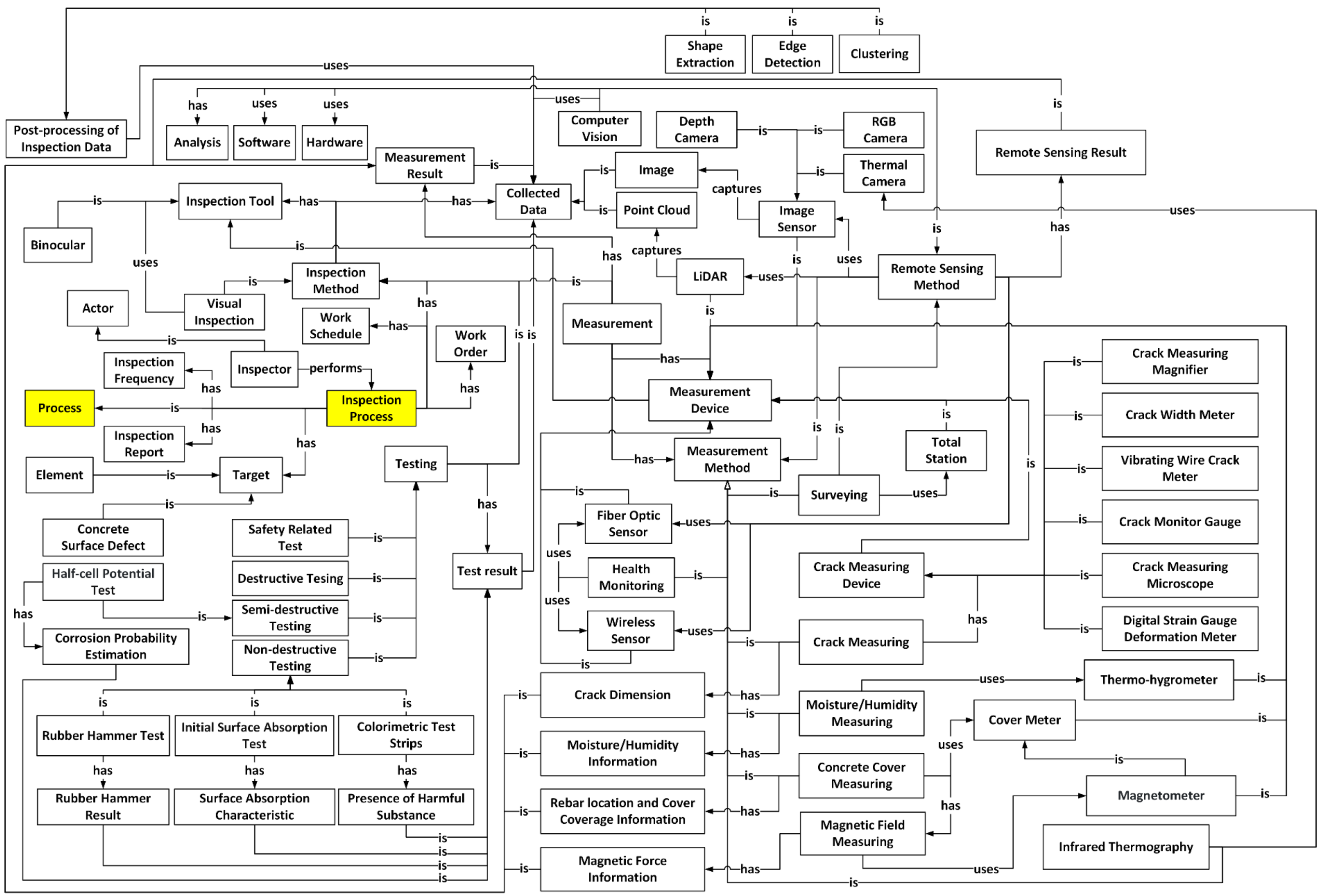


Figure 4-2. The main inspection process concepts and relationships

Table 4-1. The main inspection concepts and relationships

Concepts	Relationship	Concepts
Inspection Process	*is*	*Process*
	has	*Inspection Frequency, Inspection Method, Inspection Report, Target, Work Order, Work Schedule*
Inspector	*Performs*	*Inspection Process*
	is	*Actor*
Concrete Surface Defect, Element	*is*	*Target*
Inspection Method	*has*	*Collected Data, Inspection Tool*
Measurement, Testing, Visual Inspection	*is*	*Inspection Method*
Visual Inspection	*uses*	*Inspection Tool*
Binocular	*is*	
Measurement	*has*	*Measurement Device, Measurement Method, Measurement Result*
Concrete Cover Measuring, Crack Measuring, Health Monitoring, Infrared Thermography, Moisture/Humidity Measuring, Remote Sensing Method, Surveying	*is*	*Measurement Method*
Concrete Cover Measuring	*has*	*Magnetic Field Measuring, Rebar Location and Cover Coverage Information*
	uses	*Cover Meter*
Magnetometer	*is*	
Magnetic Field Measuring	*has*	*Magnetic Force Information*
	uses	*Magnetometer*
Crack Measuring	*has*	*Crack Dimension, Crack Measuring Device*
Moisture/Humidity Measuring	*has*	*Moisture/Humidity Information*
	uses	*Thermo-hygrometer*
Infrared Thermography	*uses*	*Thermal Camera*
Health Monitoring	*uses*	*Fiber Optic Sensor, Wireless Sensor*
Remote Sensing Method	*has*	*Remote Sensing Result*
	uses	*Fiber Optic Sensor, Image Sensor, LiDAR, Wireless Sensor*
Computer Vision, Surveying	*is*	*Remote Sensing Method*
Computer Vision	*has*	*Analysis*
	uses	*Collected Data, Hardware, Software*
Surveying	*uses*	*Total Station*
Measurement Device	*is*	*Inspection Tool*
Crack Measuring Device, Cover Meter, Fiber Optic Sensor, Image Sensor, LiDAR, Magnetometer, Thermo-hygrometer, Total Station, Wireless Sensor	*is*	*Measurement Device*

Table 4-1. The main inspection concepts and relationships (continued)

Concepts	Relationship	Concepts
Crack Measuring Magnifier, Crack Measuring Microscope, Crack Monitor Gauge, Crack Width Meter, Digital Strain Gauge Deformation Meter, Vibrating Wire Crack Meter	*is*	*Crack Measuring Device*
LiDAR	*Captures*	*Point Cloud*
Image Sensor	*Captures*	*Image*
Depth Camera, RGB Camera, Thermal Camera	*is*	*Image Sensor*
Crack Dimension, Magnetic Force Information, Moisture/Humidity Information, Rebar location and Cover Coverage Information Remote Sensing Result	*is*	*Measurement Result*
Testing	*has*	*Test Result*
Destructive Testing, Non-destructive Testing, Safety Related Test, Semi-destructive Testing	*is*	*Testing*
Half-cell Potential Test	*is*	*Semi-destructive Testing*
	has	*Corrosion Probability Estimation*
Colorimetric Test Strips, Initial Surface Absorption Test, Rubber Hammer Test	*is*	*Non-destructive Testing*
Colorimetric Test Strips	*has*	*Presence of Harmful Substance*
Initial Surface Absorption Test	*has*	*Surface Absorption Characteristic*
Rubber Hammer Test	*has*	*Rubber Hammer Result*
Corrosion Probability Estimation, Presence of Harmful Substance, Rubber Hammer Result, Surface Absorption Characteristic	*is*	*Test Result*
Image, Measurement Result, Point Cloud, Test Result	*is*	*Collected Data*
Post-processing of Inspection Data	*uses*	
Clustering, Edge Detection, Shape Extraction	*is*	*Post-processing of Inspection Data*

4.4.1.2 Diagnosis Concepts

The *diagnosis process* is an auxiliary process that evaluates the information obtained from the *inspection*. OCSD defines specific relationships to semantically link diverse *diagnosis methods*, *cause analysis*, and *condition assessment*. Figure 4-3 shows OCSD *diagnosis* process's main concepts and relationships. The main *diagnosis* concepts and relationships in Figure 4-3 are summarized in Table 4-2. The information of this process plays an important role in deciding the necessity of executing the *3R processes*. The *diagnosis process* can be done at the office by an *engineer* different from the inspector. Therefore, the information of the *engineer* needs to be covered in OCSD. The *diagnosis process* is based on processing the collected *inspection* data and the information about the surrounding conditions.

As shown in Figure 4-3, the *diagnosis* concepts of OCSD cover concepts related to analyzing the *cause* of the defect, *predicting the defect progress*, *analyzing the impact of the defect on other elements* of the structure and evaluating the *extent of damage*, *assessing the condition* of a concrete element based on *inspection* results, assessing the *condition* of connected elements based on gathered data from the surrounding environment [200], and evaluating the *need for 3R processes*.

The *diagnosis process* includes using *tools* and *heuristic methods* to interpret the *inspection* data. As discussed in Section 2.2.3, *diagnosis methods* will analyze the *inspection results*, whether *remote sensing-based*, *magnetic-based*, *acoustic-based*, *chemical-based*, etc., to find the causes of defects.

OCSD covers various causes of concrete defects. The appearance of the defect on the concrete surface has a *formation mechanism*. As discussed in Section 2.2.3, the *formation mechanism* of the defect is initiated by one or more *causes* [300]. The *actual cause* of the defect is finally determined by analyzing the *potential causes* in the *diagnosis process*. *Cause analysis* of detected defects considers the relationships with the surrounding conditions, which can be reflected in the design, construction, operation, and maintenance phases.

As discussed in Section 2.2.3, the main *problems during the design phase* that can cause defects on the concrete surface include *poor design of formwork*, *expansion joints*, etc. The main problems during the construction phase that can cause surface defects include *non-conformity issues between design and the built structure*, *inappropriate mixing*, *poor workmanship*, etc. *Non-conformity issues* refer to design and built structure discrepancies concerning elements' attributes, such as location or dimensions. The main *problems during the operation phase* that can cause surface defects include *environmental problems*, *load problems*, etc. *Lack of maintenance* and *insufficient frequency of surface protection* are *problems during the maintenance phase*.

Surface defects are often the result of a combination of *causes*. For example, *suspended solids*, such as *soil, dirt, debris*, and *fine sand*, can accumulate on the surface, causing problems for the bonding coats. Eventually, coating problems allow water and chemicals to penetrate the surface and cause *defects* [301, 302]. The presence of *water, the effect of freeze and thaw* (e.g. *aggregate extension*), attacks due to the *presence of chemicals* (*sulfates, chlorides*), and *biological agents* such as microorganisms, fungi, etc., are factors that cause surface defects (e.g. cracks) [303, 304].

At the end of the *diagnosis process*, the *engineer* will prepare a *diagnosis report* that includes information about the *condition* of the *defect* and the need for further actions and performing the *3R processes*.

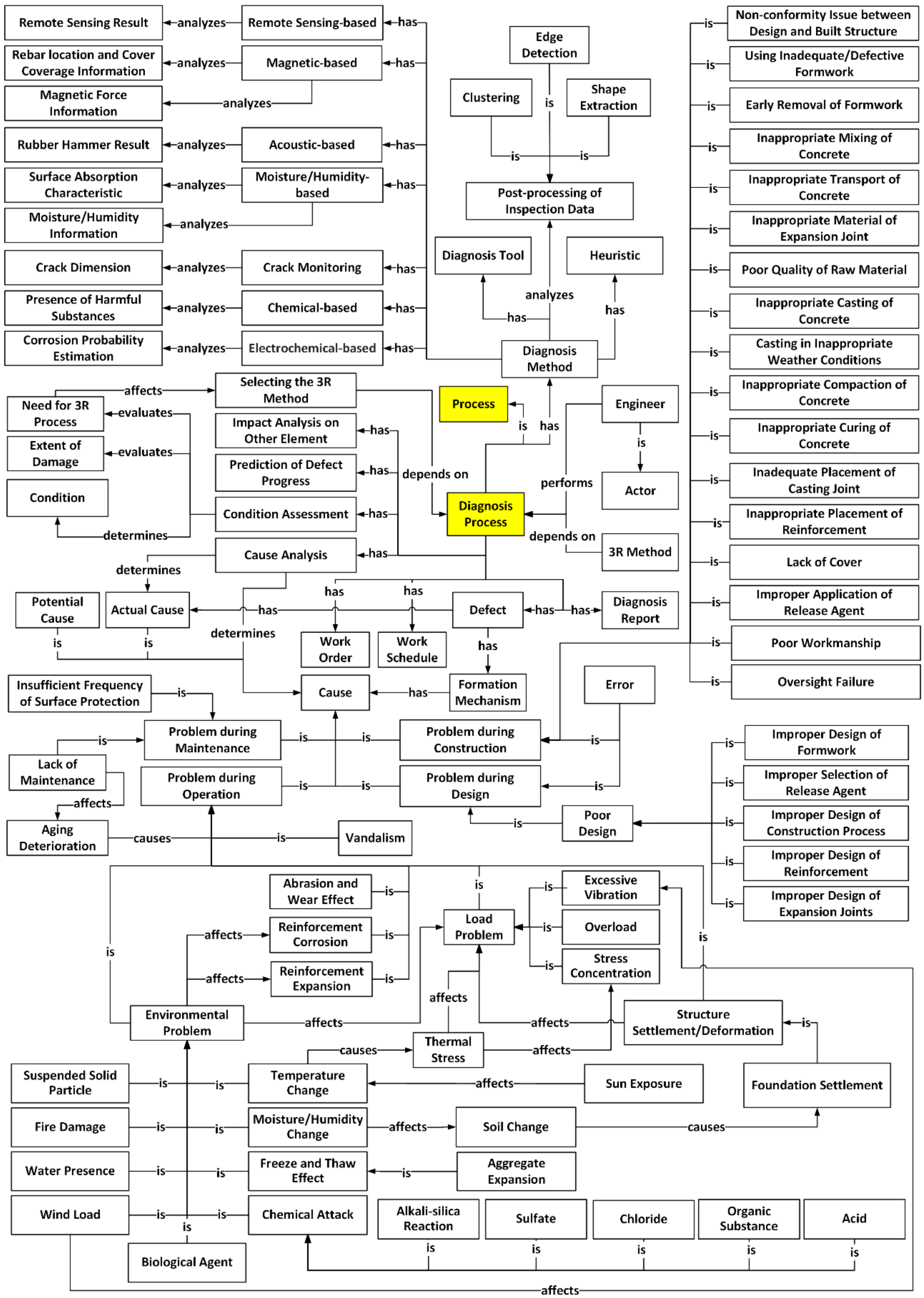


Figure 4-3. The main diagnosis process concepts and relationships

Table 4-2. The main diagnosis concepts and relationships

Concepts	Relationship	Concepts
Diagnosis Process	*is*	*Process*
	has	*Cause Analysis, Condition Assessment, Defect, Diagnosis Method, Diagnosis Report, Impact Analysis on Other Element, Prediction of Defect Progress, Work Order, Work Schedule*
Selecting the 3R Method	*depends on*	*Diagnosis Process*
Engineer	*performs*	
	is	*Actor*
Diagnosis Method	*has*	*Acoustic-based, Chemical-based, Crack Monitoring, Diagnosis Tool, Electrochemical-based, Heuristic, Magnetic-based, Moisture/Humidity-based, Remote Sensing-based*
	analyzes	*Post-processing of Inspection Data*
Clustering, Edge Detection, Shape Extraction	*is*	
Acoustic-based	*analyzes*	*Rubber Hammer Result*
Crack Monitoring	*analyzes*	*Crack Dimension*
Chemical-based	*analyzes*	*Presence of Harmful Substance*
Electrochemical-based	*analyzes*	*Corrosion Probability Estimation*
Magnetic -based	*analyzes*	*Magnetic Force Information, Rebar Location and Cover Coverage Information*
Moisture/Humidity-based	*analyzes*	*Surface Absorption Characteristic, Moisture/Humidity Information*
Remote Sensing-based	*analyzes*	*Remote Sensing Result*
Condition Assessment	*evaluates*	*Extent of Damage, Need for 3R Process*
	evaluates	*Condition/Sate*
Need for 3R Process	*affects*	*Selecting the 3R Method*
Cause Analysis	*determines*	*Actual Cause, Cause*
Defect	*has*	*Actual Cause, Formation Mechanism*
Formation Mechanism	*has*	*Cause*
Actual Cause, Potential Cause, Problem During Construction, Problem During Design, Problem During Maintenance, Problem During Operation	*is*	

Table 4-2. The main diagnosis concepts and relationships (continued)

Concepts	Relationship	Concepts
Casting in Inappropriate Weather Conditions, Early Removal of Formwork, Error, Improper Application of Release Agent, Inadequate Placement of Expansion Joint, Inappropriate Casting of Concrete, Inappropriate Compaction of Concrete, Inappropriate Curing of Concrete, Inappropriate Material of Expansion Joint, Inappropriate Mixing of Concrete, Inappropriate Placement of Reinforcement, Inappropriate Transport of Concrete, Lack of Cover, Non-conformity Issue between Design and Built Structure, Oversight Failure, Poor Quality of Raw Material, Poor Workmanship, Using Inadequate/Defective Formwork	is	Problem During Construction
Error, Poor Design	is	Problem During Design
Improper Design of Construction Process, Improper Design of Expansion Joints, Improper Design of Formwork, Improper Design of Reinforcement, Improper Selection of Release Agent	is	Poor Design
Insufficient Frequency of Surface Protection, Lack of Maintenance	is	Problem During Maintenance
Lack of Maintenance	affects	Aging Deterioration
Aging Deterioration	causes	Problem During Operation
Abrasion and Wear Effect, Environmental Problem, Load Problem, Reinforcement Corrosion, Reinforcement Expansion, Structure Settlement/Deformation, Vandalism	is	
Environmental Problem	affects	Load Problem, Reinforcement Corrosion, Reinforcement Expansion
Biological Agent, Chemical Attack, Fire Damage, Freeze and Thaw Effect, Moisture/Humidity Change, Suspended Solid Particle, Temperature Change, Water Presence, Wind Load	is	Environmental Problem
Temperature Change	causes	Thermal Stress
Thermal Stress	affects	Load Problem, Stress Concentration
Sun Exposure	affects	Temperature Change
Moisture/Humidity Change	affects	Soil Change
Soil Change	causes	Foundation Settlement
Foundation Settlement	is	Structure Settlement/Deformation

Table 4-2. The main diagnosis concepts and relationships (continued)

Concepts	Relationship	Concepts
Structure Settlement/Deformation	*affects*	*Load Problem*
Excessive Vibration, Overload, Stress Concentration	*is*	
Aggregate Expansion	*is*	*Freeze and Thaw Effect*
Acid, Alkali-silica Reaction, Chloride, Organic Substance, Sulfate	*is*	*Chemical Attack*
Wind Load	*affects*	*Excessive Vibration*

4.4.1.3 3R Concepts

In general, the term *repair* refers to restoring, renewing, or replacing concrete surface or element after primary placement [305]. OCSD presents specific relationships to semantically link various *3R methods* and related *repair materials* for defective components. Figure 4-4 shows OCSD *3R processes*' main concepts and relationships. The main *3R* concepts and relationships in Figure 4-4 are summarized in Table 4-3. In OCSD, *repair* refers to the specific actions that need to be done to treat the defective elements of the structure. *Rehabilitation* refers to the major repair of critical elements of the structure to reach the suitable service level. *Replacement* refers to the removal and replacement of defective areas or damaged elements of the structure. The *3R processes* are based on the results of the *diagnosis process* and the *condition* of the defect, *host elements*, and *impacted elements*. After the *diagnosis process*, if further actions are required to maintain the element and the structure, the *3R processes* will be done to treat the element.

As shown in Figure 4-4, the *3R processes* can be done by a *3R company*. The information of the *3R company* and *3R work order* are covered in OCSD. The *3R work order* includes *request* and *component ID*, *team* or assigned *person ID*, *date* and *time*, *location*, estimated *cost*, *status*, and *emergency level*. The *3R processes* include using *material*, *tools*, and *methods* to perform an acceptable level of concrete surface treatment. The quality-related *specifications of materials*, including *bonding strength* and *durability* of materials, are considered in OSCD. The *3R methods* for treating concrete surface defects are *surface cleaning, repair of surface irregularities, protecting protruding edges, surface sealing or coating, rehabilitation and strengthening of concrete, concrete repair or replacement* as explained Section 2.2.5.

As discussed in Section 2.2.5, methods used to *repair* or *replace* concrete with surface defects include *filling cracks, placing shotcrete on the surface, and adding or replacing mortar or concrete*. Different types of *mortar or concrete*, such as *conventional mortar or concrete*, *preplaced aggregate concrete, polymer-modified mortar or concrete, and epoxy mortar or concrete*, can be used to repair defective surfaces. At the end of the *3R processes*, the *actor* will prepare *3R execution report* that includes information about the actual *3R date*, *cost*, etc. In addition, information about the treated surface defects can be archived in a way to allow relating this information to the future *inspection* data to track the element *condition* and reduce the *potential cause* of the defect by appropriate maintenance.

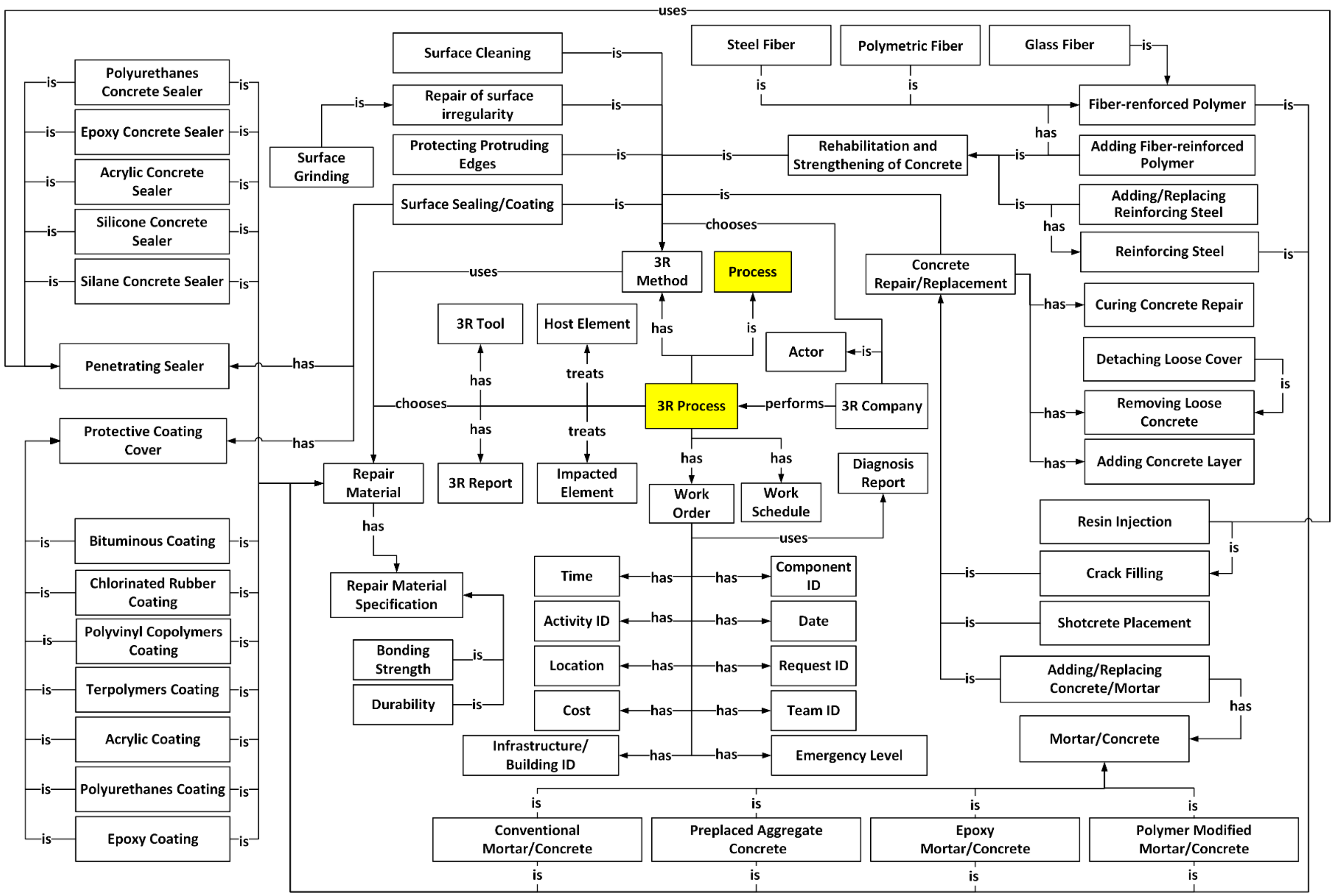


Figure 4-4. The main 3R processes concepts and relationships

Table 4-3. The main 3R concepts and relationships

Concepts	Relationship	Concepts
3R Process	*is*	*Process*
	has	*3R Method, 3R Report, 3R Tool, Work Order*
	chooses	*Repair Material*
	treats	*Host Element, Impacted Element*
3R Company	*performs*	*3R Process*
	is	*Actor*
	chooses	*3R Method*
3R Method	*uses*	*Repair Material*
Repair Material	*has*	*Repair Material Specification*
Bonding Strength, Durability	*is*	
Work Order	*uses*	*Diagnosis Report*
Work Order	*has*	*Activity ID, Component ID, Cost, Date, Emergency Level, Infrastructure/Building ID, Location, Request ID, Time*
Concrete Repair/Replacement, Protecting Protruding Edges, Rehabilitation and Strengthening of Concrete, Repair of Surface Irregularities, Surface Cleaning, Surface Sealing/Coating	*is*	*3R Method*
Surface Grinding	*is*	*Repair of Surface Irregularity*
Surface Sealing/Coating	*has*	*Penetrating Sealer, Protective Coating Cover*
Acrylic Concrete Sealer, Epoxy Concrete Sealer, Polyurethane Concrete Sealer, Silane Concrete Sealer, Silicone Concrete Sealer	*is*	*Penetrating Sealer*
Acrylic Coating, Bituminous Coating, Chlorinated Rubber Coating, Epoxy Coating, Polyurethane Coating, Polyvinyl Copolymers Coating, Terpolymers Coating	*is*	*Protective Coating Cover*
Adding Fiber-reinforced Polymer, Adding/Removing Reinforcing Steel	*is*	*Rehabilitation and Strengthening of Concrete*
Adding Fiber-reinforced Polymer	*has*	*Fiber-reinforced Polymer*
Glass Fiber, Polymetric Fiber, Steel Fiber	*is*	
Adding/Removing Reinforcing Steel	*has*	*Reinforcing Steel*
Concrete Repair/Replacement	*has*	*Adding Concrete Layer, Curing Concrete Repair, Removing Loose Concrete*
Detaching Loose Cover	*is*	*Removing Loose Concrete*
Adding/Replacing Concrete/Mortar, Crack Filling, Shotcrete Placement	*is*	*Concrete Repair/Replacement*
Resin Injection	*is*	*Crack Filling*
	uses	*Penetrating Sealer*

Table 4-3. The main 3R concepts and relationships (continued)

Concepts	Relationship	Concepts
Adding/Replacing Concrete/Mortar	*has*	*Mortar/Concrete*
Conventional Mortar/Concrete, Epoxy Mortar/Concrete, Polymer Modified Mortar/Concrete, Preplaced Aggregate Concrete	*is*	
Acrylic Coating, Acrylic Concrete Sealer, Bituminous Coating, Chlorinated Rubber Coating, Conventional Mortar/Concrete, Epoxy Coating, Epoxy Concrete Sealer, Epoxy Mortar/Concrete, Fiber-reinforced Polymer, Polymer Modified Mortar/Concrete, Polyurethane Coating, Polyurethane Concrete Sealer, Polyvinyl Copolymers Coating, Preplaced Aggregate Concrete, Reinforcing Steel, Silane Concrete Sealer, Silicone Concrete Sealer, Terpolymers Coating	*is*	*Repair Material*

4.4.2 Product Modeling Concepts

The additional product-related concepts of OCSD cover the main concepts related to defects and repair product modeling, as explained below.

4.4.2.1 Defect Concepts

The *defect* is the final product of the *inspection process*. The *diagnosis process* examines the *defect*, and finally, if necessary, the *3R processes* will focus on treating the defect. Since *defects* play a key role in all of these processes, OCSD should cover the concepts of defects as the main product of these processes. Detailed semantic relationships are defined in OCSD to connect different types of *concrete surface defects* and their impact on defective elements. OCSD *concrete surface defects* and the *condition* of the defected concrete surfaces main concepts and relationships are shown in Figure 4-5. The main concepts and relationships in Figure 4-5 are summarized in Table 4-4.

As discussed in Section 2.2.1, the attributes of defects are defined based on common types of *concrete surface defects* [40, 41, 306, 307]. As shown in Figure 4-5, the defected product in OCSD covers information related to *host* and *impacted elements*, *defect types*, and *condition* of the defected concrete surfaces. The *host element* is the defective element. When there is a *defect*, the *host element* is usually weakened, which affects other elements, leading to the formation of new defects related to this process. As discussed in Section 4.4.1.2, the *actual cause* of the defect will be determined in the process of *cause analysis* from the *potential causes*. *Defects* are defined by features such as *generation period, orientation, location, dimensions, shape or patterns,* and *severity*. Depending on the *defect types*, the changes in concrete surface forms include *addition, deformation, section loss,* and *subtraction*.

As discussed in Section 2.2.1, common types of surface cracks include: *cracks, spalling, delamination, scaling, disintegration, erosion, honeycombing*, etc. A *functional defect* is a defect that disrupts the expected performance of an *element* or *structur*e. Issues caused by any, or a combination, of defects in the concrete surface can change the *condition* of the *element*, causing a *functional defect* of the *element* or *structure.*

As explained in Section 2.2.4, the definition of levels of *severity* and *conditions* of some specific concrete surface defects in OCSD are based on the Ontario structure inspection manual (OSIM) [41]. The value of severity of each defect based on Table 2-1 is added as an individual's property set in OCSD. The severity of surface defects can be categorized as *light, medium, sever,* and *very severe*, and the c*ondition* of an element can be categorized into *excellent, good, fair,* and *poor*. In *cracks*, the *severity* can be divided into *hairline, narrow, medium,* and *wide*. The *condition* of the element depends on the *severity* of the defect.

The presence of some *concrete surface defects* indicates a specific *condition* in the *element*. For example, the presence of *cold joints* is a *fair element condition*. Moreover, some *defects*, such as *stains*, can have different *conditions* based on specific information. For example, some *stains,* such as those caused by *biological growth* and *dust*, do not indicate the weakness of the *element*, and the *condition* of the *element* can be assessed as *good*. However, some *stains*, such as *stains* caused by *chemical reactions*, *water,* and *corrosion*, indicate an abnormal *condition* in the *element*, and the *condition* of the *element* can be assessed as *fair*. *Graffiti*, *bughole*, and *flatness defects* only affect the appearance of the concrete and do not affect the strength of the concrete, so the *condition* of the defective *element* is considered *good* in the presence of these *defects* [40, 41].

4.4.2.2 Repair Product Modeling Concepts

Repair product modeling should cover the following information: (1) *host* and *impacted elements*; and (2) modified model of the *element* after the *3R process* including changes in the geometry and materials.

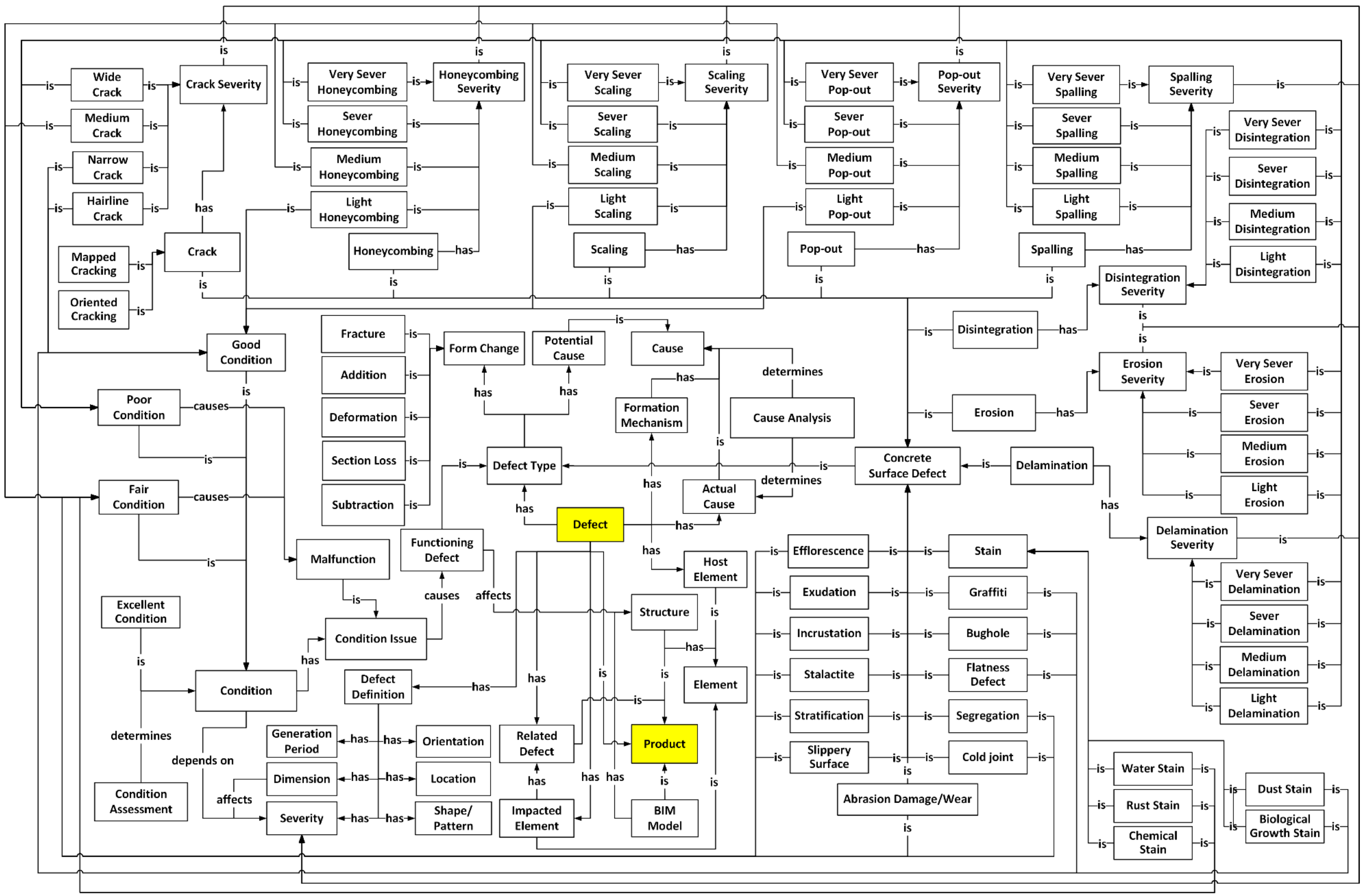


Figure 4-5. The main defects and the condition of the defected concrete surfaces concepts and relationships

Table 4-4. The main defects and the condition of the defected concrete surfaces concepts and relationships

<table>
<tr><th>Concepts</th><th>Relationship</th><th>Concepts</th></tr>
<tr><td>BIM Model, Defect, Related Defect, Structure</td><td>is</td><td>Product</td></tr>
<tr><td>BIM Model</td><td>has</td><td>Structure</td></tr>
<tr><td>Defect</td><td>has</td><td>Actual Cause, Defect Definition, Defect Type, Formation Mechanism, Host Element, Impacted Element, Related Defect</td></tr>
<tr><td>Structure</td><td>has</td><td rowspan="2">Element</td></tr>
<tr><td>Host Element, Impacted Element</td><td>is</td></tr>
<tr><td>Impacted Element</td><td>has</td><td>Related Defect</td></tr>
<tr><td>Formation Mechanism</td><td>has</td><td rowspan="2">Cause</td></tr>
<tr><td>Actual Cause, Potential Cause</td><td>is</td></tr>
<tr><td>Cause Analysis</td><td>determines</td><td>Actual Cause, Cause</td></tr>
<tr><td>Defect Definition</td><td>has</td><td>Dimension, Generation Period, Location, Orientation, Severity, Shape/Pattern</td></tr>
<tr><td>Dimension</td><td>affects</td><td>Severity</td></tr>
<tr><td>Defect Type</td><td>has</td><td>Form Change, Potential Cause</td></tr>
<tr><td>Addition, Deformation, Section Loss, Subtraction</td><td>is</td><td>Form Change</td></tr>
<tr><td>Concrete Surface Defect, Functioning Defect</td><td>is</td><td>Defect Type</td></tr>
<tr><td>Condition Assessment</td><td>determines</td><td rowspan="2">Condition</td></tr>
<tr><td>Excellent Condition, Fair Condition, Good Condition, Poor Condition</td><td>is</td></tr>
<tr><td rowspan="2">Condition</td><td>depends on</td><td>Severity</td></tr>
<tr><td>has</td><td rowspan="2">Condition Issue</td></tr>
<tr><td>Malfunction</td><td>is</td></tr>
<tr><td>Condition Issue</td><td>causes</td><td>Functioning Defect</td></tr>
<tr><td>Fair Condition, Poor Condition</td><td>causes</td><td>Malfunction</td></tr>
<tr><td>Abrasion Damage/Wear, Bughole, Cold Joint, Crack, Delamination, Disintegration, Efflorescence, Erosion, Exudation, Flatness Defect, Graffiti, Honeycombing, Incrustation, Pop-out, Scaling, Segregation, Slippery Surface, Spalling, Stain, Stalactite, Stratification</td><td>is</td><td>Concrete Surface Defect</td></tr>
<tr><td>Biological Growth Stain, Chemical Stain, Dust Stain, Rust Stain, Water Stain</td><td>is</td><td>Stain</td></tr>
<tr><td>Mapped Cracking, Oriented Cracking</td><td>is</td><td>Crack</td></tr>
<tr><td>Crack</td><td>has</td><td rowspan="2">Crack Severity</td></tr>
<tr><td>Hairline Crack, Medium Crack, Narrow Crack, Wide Crack</td><td>is</td></tr>
</table>

Table 4-4. The main defects and the condition of the defected concrete surfaces concepts and relationships (continued)

Concepts	Relationship	Concepts
Delamination	*has*	*Delamination Severity*
Light Delamination, Medium Delamination, Severe Delamination, Very Severe Delamination	*is*	
Disintegration	*has*	*Disintegration Severity*
Light Disintegration, Medium Disintegration, Severe Disintegration, Very Severe Disintegration	*is*	
Erosion	*has*	*Erosion Severity*
Light Erosion, Medium Erosion, Severe Erosion, Very Severe Erosion	*is*	
Honeycombing	*has*	*Honeycombing Severity*
Light Honeycombing, Medium Honeycombing, Severe Honeycombing, Very Severe Honeycombing	*is*	
Pop-out	*has*	*Pop-out Severity*
Light Pop-out, Medium Pop-out, Severe Pop-out, Very Severe Pop-out	*is*	
Scaling	*has*	*Scaling Severity*
Light Scaling, Medium Scaling, Severe Scaling, Very Severe Scaling	*is*	
Spalling	*has*	*Spalling Severity*
Light Spalling, Medium Spalling, Severe Spalling, Very Severe Spalling	*is*	
Crack Severity, Delamination Severity, Disintegration Severity, Erosion Severity, Honeycombing Severity, Pop-out Severity, Scaling Severity, Spalling Severity	*is*	*Severity*
Light Delamination, Light Disintegration, Light Erosion, Light Spalling, Medium Delamination, Medium Disintegration, Medium Erosion, Medium Spalling, Severe Delamination, Severe Disintegration, Severe Erosion, Severe Honeycombing, Severe Pop-out, Severe Scaling, Severe Spalling, Very Severe Delamination, Very Severe Disintegration, Very Severe Erosion, Very Severe Honeycombing, Very Severe Pop-out, Very Severe Scaling, Very Severe Spalling, Wide Crack	*is*	*Poor Condition*

Table 4-4. The main defects and the condition of the defected concrete surfaces concepts and relationships (continued)

Concepts	Relationship	Concepts
Abrasion Damage/Wear, Chemical Stain, Cold Joint, Efflorescence, Exudation, Incrustation, Medium Crack, Medium Honeycombing, Medium Pop-out, Medium Scaling, Rust Stain, Segregation, Slippery Surface, Stalactite, Stratification, Water Stain	*is*	*Fair Condition*
Biological Growth Stain, Bughole, Dust Stain, Flatness Defect, Graffiti, Hairline Crack, Light Honeycombing, Light Pop-out, Light Scaling, Narrow Crack	*is*	*Good Condition*

4.5 Evaluation of OCSD

Ontology tools perform the consistency evaluation during the verification process [308, 309], [310]. In this regard, to evaluate the consistency and identify the subsumption relationships, HermiT OWL Reasoner, which is based on the hypertableau algorithm is applied [311]. Two other evaluation methods are used for evaluating the usefulness of OCSD: (1) application-based evaluation, and (2) qualitative criteria-based evaluation. The application-based evaluation is used to demonstrate the benefits of the ontology using a case study. This approach judges whether the ontology is suitable and meets the objectives. The qualitative criteria-based evaluation assesses the correctness and presentation of the main concepts and relationships of the developed ontology.

4.5.1 Consistency Evaluation Using Protégé

In protégé, a description logic reasoner is used to perform the verification process and test the consistency criteria for OCSD [308, 309]. OWL HermiT Reasoner explores the relationships and discovers the implicit relationships between classes. Furthermore, it verifies the concepts hierarchy and clarifies any inconsistencies in the ontology. For example, no individual can be at the same time an instance of two classes, which the reasoner can check. HermiT OWL reasoner was utilized during OCSD development stage and clarified some inconsistencies in the ontology. These results were utilized as feedback and input to rectify problems before going on to the final step.

4.5.2 Case Study

The application-based evaluation is used to demonstrate the benefits of OCSD using a case study. This approach judges whether the ontology is suitable and meets the objectives. In the ontology development process, evaluating the content of the ontology is an essential step towards improving the developed ontology [312, 313]. The evaluation of OCSD is investigated in a case study in Section 6.6, where a specific inspection method based on deep learning using point clouds is used to semi-automatically create the as-inspected BIM model.

4.5.3 Criteria-based Evaluation

The adequacy of the semantic representation of OCSD was evaluated through a survey comprising of 11 questions related to different components of OCSD. The information of the respondents was collected through the first question.

The second question was about the benefits of inspection, diagnosis, and repair related information modeling of concrete surface defects. The third and fourth questions were about the clarity and comprehensiveness of the main inspection concepts and relationships of OCSD. The fifth and sixth questions investigated the clarity and comprehensiveness of the main diagnosis concepts and relationships of OCSD. The seventh and eighth questions investigated the clarity and comprehensiveness of the 3R concepts and relationships of OCSD. The ninth and tenth questions investigated the clarity and comprehensiveness of the defect concepts and relationships of OCSD. The eleventh question considered OCSD capabilities to influence the future BIM-based asset management systems. The questions of this survey are listed in Table 4-5.

The five-point Likert scale was used to obtain the qualitative values of the answers. Figures 4-2, 4-3, 4-4, and 4-5 were used in the survey to present some details of OCSD. The survey was sent to 101 internationally recognized experts selected based on their knowledge in BIM, concrete construction, inspection, diagnosis, and repair.

The results of OBRNIT survey including the total number of participants, the respondents' profiles, and the survey answers are available at https://github.com/OCSD-OWL/Survey-Result.

Table 4-5. The evaluation questions of OCSD

Q_1	Name, organization/university, area of expertise, and years of experience.
Q_2	Developing a unified ontology for modeling inspection, diagnosis, and repair related information of concrete surface defects will facilitate accessing and updating the information, and streamlining the processes at different phases of the lifecycle resulting in improved efficiency and reduced rate of data input errors. Do you agree with this statement? ○ Strongly agree ○ Agree ○ Neither agree nor disagree ○ Disagree ○ Strongly disagree ○ No answer Comments: ……………………………………………………………………………………………...
Q_3	Figure 4-2 represents the high-level concepts and relationships of the ontology for the inspection process of concrete surface defects. Do you find this representation clear and provide good understanding of the concepts in the domain? ○ Very clear ○ Clear ○ Somewhat clear ○ Not so clear ○ Not clear at all ○ No answer Comments: ……………………………………………………………………………………………...
Q_4	Based on Figure 4-2, do you find the representation comprehensive? Comprehensiveness here means representing the main concepts and relationships for modeling the inspection-related information of concrete surface defects. ○ Very comprehensive ○ Comprehensive ○ Somewhat comprehensive ○ Not comprehensive ○ Missing lots of concepts ○ No answer Comments: ……………………………………………………………………………………………...
Q_5	Figure 4-3 represents the high-level concepts and relationships of the ontology for the diagnosis process of concrete surface defects. The diagnosis process is based on processing the collected inspection data and the information about the surrounding conditions. Do you find this representation clear and provide good understanding of the concepts in the domain? ○ Very clear ○ Clear ○ Somewhat clear ○ Not so clear ○ Not clear at all ○ No answer Comments: ……………………………………………………………………………………………...

Table 4 5. The evaluation questions of OCSD (continued)

Q_6	Based on Figure 4-3, do you find the representation comprehensive? Comprehensiveness here means representing the main concepts and relationships for modeling the diagnosis-related information of concrete surface defects. ○ Very comprehensive ○ Comprehensive ○ Somewhat comprehensive ○ Not comprehensive ○ Missing lots of concepts ○ No answer Comments: …………………………………………………………………………………………...
Q_7	Figure 4-4 below represents the high-level concepts and relationships of the ontology for the 3R (Repair, Rehabilitation, and Repair) processes of concrete surface defects. Do you find this representation clear and provide good understanding of the concepts in the domain? ○ Very clear ○ Clear ○ Somewhat clear ○ Not so clear ○ Not clear at all ○ No answer Comments: …………………………………………………………………………………………...
Q_8	Based on Figure 4-4, do you find the representation comprehensive? Comprehensiveness here means representing the main concepts and relationships for modeling the 3R-related information of concrete surface defects. ○ Very comprehensive ○ Comprehensive ○ Somewhat comprehensive ○ Not comprehensive ○ Missing lots of concepts ○ No answer Comments: …………………………………………………………………………………………...
Q_9	Figure 4-5 below represents the high-level concepts and relationships of the ontology for the defects and condition of the defected concrete surfaces. Do you find this representation clear and provide good understanding of the concepts in the domain? ○ Very clear ○ Clear ○ Somewhat clear ○ Not so clear ○ Not clear at all ○ No answer Comments: …………………………………………………………………………………………...
Q_{10}	Based on Figure 4-5, do you find the representation comprehensive? Comprehensiveness here means representing the main concepts and relationships for modeling concrete surface defects and condition of the defected surfaces. ○ Very comprehensive ○ Comprehensive ○ Somewhat comprehensive ○ Not comprehensive ○ Missing lots of concepts ○ No answer Comments: …………………………………………………………………………………………...
Q_{11}	OCSD provided knowledge is expected to influence the future BIM-based asset management systems and allow a new level of coordination and collaboration among the stakeholders of the project. Do you agree with this statement? ○ Strongly agree ○ Agree ○ Neither agree nor disagree ○ Disagree ○ Strongly disagree ○ No answer Comments: …………………………………………………………………………………………...

4.6 Summary and Conclusions

This chapter focused on the development of an ontology, called OCSD, for concrete surface defects to have a unified knowledge model where all the stakeholders can access information in a systematic manner. There are 333 classes, 51 relations, 27 attributes, and 31 individuals in OCSD. OCSD comprises high-level knowledge of the concepts and relationships related to concrete surface defects, inspection, diagnosis, and 3R processes. In addition, the consistency of OCSD was evaluated using HermiT OWL reasoner. A survey was designed and conducted to evaluate the semantic representation of OCSD. The evaluation proves that OCSD satisfies the domain experts and covers the domain's main concepts and relationships. Based on the evaluation, OCSD was able to provide a clear understanding of the concepts and relationships in the domain. The application of OCSD is investigated in a case study in Section 6.6. The evaluation demonstrates that OCSD can answer all the competency questions defined in Section 4.2 as follows: (1) OCSD was developed based on a top-down approach, and it is not dependent on specific types of structures and can be used at different phases of the lifecycle; (2) OCSD covers the main information about the generic aspects of the inspection, diagnosis, and repair processes, and based on the evaluation, it has comprehensive modeling; (3) OCSD reflects the common aspects of OSIM guidelines at an abstract level and can be applied to all types of concrete structures; (4) OCSD has the main inspection-related concepts at the abstract level that can be used to extend the IFC standard; and (5) OCSD is able to accommodate new data collection technologies and the associated inspection data.

OCSD is expected to provide the following benefits: (1) OCSD can help future asset management systems benefit from the provided knowledge and efficiently develop, modify, and process the ontological knowledgebase. The knowledge model can be used as the first step for the purpose of re-engineered processes of infrastructure management systems, analysis reflecting the defects and repair changes of the structure, and visual analytics to support diagnosis processes [314]. Moreover, OCSD knowledgebase can be used to develop concrete surface inspection expert systems, software or checklists; (2) All the details of the inspection, diagnosis, and 3R are integrated in OCSD, and this integration will facilitate accessing and updating the information, and streamlining the processes at different phases of the lifecycle resulting in improved efficiency and reduced rate of data input errors; (3) OCSD allows a new level of coordination and collaboration among the stakeholders of the project; and (4) It can be used as a starting point to extend IFC for missing inspection-related information.

CHAPTER 5. ONTOLOGY FOR BIM-BASED ROBOTIC NAVIGATION AND INSPECTION TASKS

5.1 Introduction

Inspection is indispensable in the construction industry. Robots are used to automate the process of inspection during the construction and operation phases. The use of advanced technologies (e.g. LiDAR scanners and sensors) has made the inspection process more accurate and reliable [2]. As explained in Section 2.6.3, the robotic system utilizes and processes the ontology as the robot's central data store [8]. The BIM-based approach is also expected to improve the inspection process. Besides, unstructured and unknown environments (e.g. post-disaster situations) can be better inspected with the help of robotic inspection. The objective of this chapter is to develop BIM-based ontology to cover the different types of information and concepts related to robot navigation and inspection tasks. This ontology aims to help system engineers involved in developing robotic inspection systems by identifying the different concepts and relationships about robotic inspection and navigation tasks based on BIM information. The navigation concepts in this chapter are dependent on using the semantic knowledge based on the BIM concepts for navigation tasks. The use case is an inspection robot that is navigating in a building with partial knowledge of the environment because of changes in the available information due to construction and renovation scheduling issues, unexpected obstacles in the building, etc.

5.2 Methodology Workflow

In this section, the methodology workflow steps to develop OBRNIT are similar to what is explained in Section 4.3. The competency questions need to be defined as a part of the requirements of the robotic inspection domain [293]. The competency questions are identified based on the use case diagram as explained in Section 3.4 (Figure 3-4) and reviewed literature to define the key challenges OBRNIT can address, as shown in Table 5-1.

Table 5-1. Competency questions of OBRNIT

Q1	How to locate the defect in the BIM model?
Q2	How to relate the mobility characteristics of the robot with the conditions of the building based on the BIM model?
Q3	How to benefit from the BIM model in defining the path of the robot?
Q4	How to use the sensors of the robot to find the mismatches with the BIM model for replanning the path of the robot?
Q5	How to select the suitable sensors of the robot for the specific inspection task?

The methodology for developing OBRNIT is METHONTOLOGY, which is clear, well-documented, mature, and based on the experience of other domains ontology development [294], [295]. OBRNIT development based on METHONTOLOGY includes the initial, development, and final stages as shown in Figure 5-1. The best practices and knowledge in the robotic inspection domain are used to develop OBRNIT.

The initial stage involves steps to specify the scope, main concepts, and the taxonomies of OBRNIT. The scope of OBRNIT is defined based on the requirements. Research papers, textbooks, and online resources are used as sources for the requirements (e.g. properties). The ontology needs to cover all the concepts about the robot characteristics, building characteristics, and inspection and navigation tasks.

Furthermore, this step helps to consider the size of the development and the level of detail that needs to be covered in OBRNIT. The next step is defining the concepts and taxonomies for OBRNIT. The data related to OBRNIT are gathered in this step. The list of requirements from the defining scope step helps the process of ontology development. Communication with experts and end-users along with getting feedback from them is essential at the whole cycle of this stage.

The development stage is devoted to constructing and verifying the initial structure of OBRNIT. In the first step of the development stage, the conceptualization model is clearly represented and implemented in a formal language (e.g. OWL) to be later accessible by computers and used by different systems [296]. The development of OBRNIT involves reusing and adapting BIM concepts. BEO v.0.1.0 [223, 233], which is based on the IfcBuildingElement subtree in the IFC specification and ifcOWL ontology [212], is a good starting point for including the relevant BIM concepts to OBRNIT. BEO is available in OWL format, which facilities the integration process. Moreover, BOT v.0.3.2 [179] and DOT v.0.8 [222] ontologies are integrated and adapted in the development of OBRNIT to represent the required concepts related to damages and building topology, respectively.

The ontology integration in the METHONTOLOGY method can be done at the conceptualization level [315]. The methods to reuse available ontologies are: (1) ontologies merging, (2) ontologies alignment, and (3) ontologies integration. Ontologies merging refers to unifying two or more available ontologies by comparing the available ontologies and finding similarities between their domain information [315]. Ontologies alignment refers to mapping the concepts and relationships in two or more available ontologies to find equivalency between them. This method requires the smallest number of changes, and it is a simpler form of merging [316, 317]. Ontologies integration refers to integrating one or more available ontologies in the process of developing a new ontology by adapting, extending, specializing, or assembling [315]. The ontology integration method is selected in this chapter as it saves the effort to reuse and adapt the components that are needed to complete OBRNIT [318]. The next step of the development stage is verifying the developed ontology. Based on the consistency rules and competency questions, this process examines the ontologies from the technical perspective.

The final stage is to add new, or modify existing, relationships and evaluate OBRNIT with experts and end-users through evaluation questions. In this stage, the ontology is improved with the suggestions of the domain experts and end-users to fulfill the real-world requirements. OBRNIT evaluation is done through a case study and a criteria-based evaluation method [319]. Similar ontologies in the robotic inspection domain are not available to compare the developed ontology with a benchmark ontology or high-level standards in the domain. The final step is documenting the developed OBRNIT. Obtaining knowledge, evaluation, and documentation are involved throughout the whole life cycle of ontology development. Each step of the METHONTOLOGY is presented using IDEF5 (Integrated DEFinition) [297] ontology description method, which includes detailed information about the input, output, control, and mechanism. The next section explains in detail about the ontology development. The following section focuses on the verification and evaluation steps.

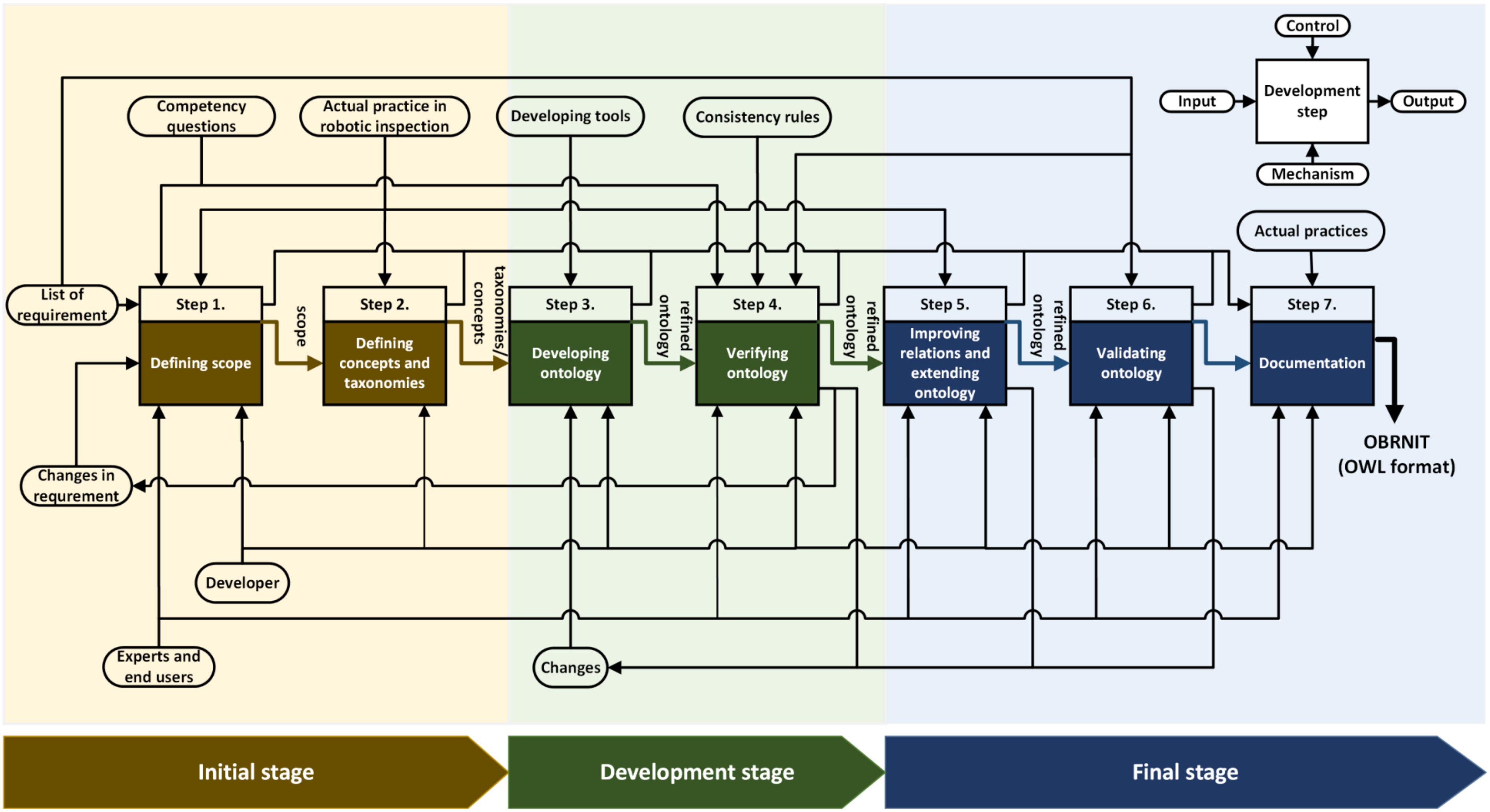


Figure 5-1. Development workflow of OBRNIT (adapted from Taher et al. [298])

5.3 Developing OBRNIT

Some concepts from BIM and KnowRob ontology [132] are used as parts of this study. Protégé [206] is used to develop OBRNIT and to integrate it with BEO, BOT, and DOT [320]. OBRNIT has 386 classes, 45 relations, 52 attributes, and 8 individuals. The current version of OBRNIT is available at https://github.com/OBRNIT/OBRNIT.

OBRNIT covers four main groups of concepts including: (1) robot concepts, (2) building concepts, (3) navigation task concepts, and (4) inspection task concepts, which are explained in the following sections. Figure 5-2(a) shows the main concepts and relationships of OBRNIT. Figure 5-2(b) shows the inspection task concepts. Some concepts are duplicated in Figure 5-2(a) and (b) to improve the readability of the figures. Color coding is used to group the concepts pertaining to each of the four groups. However, the figures are simplified by adding the colors only to the main concepts of the ontology. The relationships between concepts show how the ontology components are semantically interrelated. The types of relations used in the developed ontology are: *is*, *has*, *uses*, *affects*, *performs*, *causes*, *captures*, *has state*, *has time*, *has target*, and *measures* (e.g. thermal camera *measures* temperature).

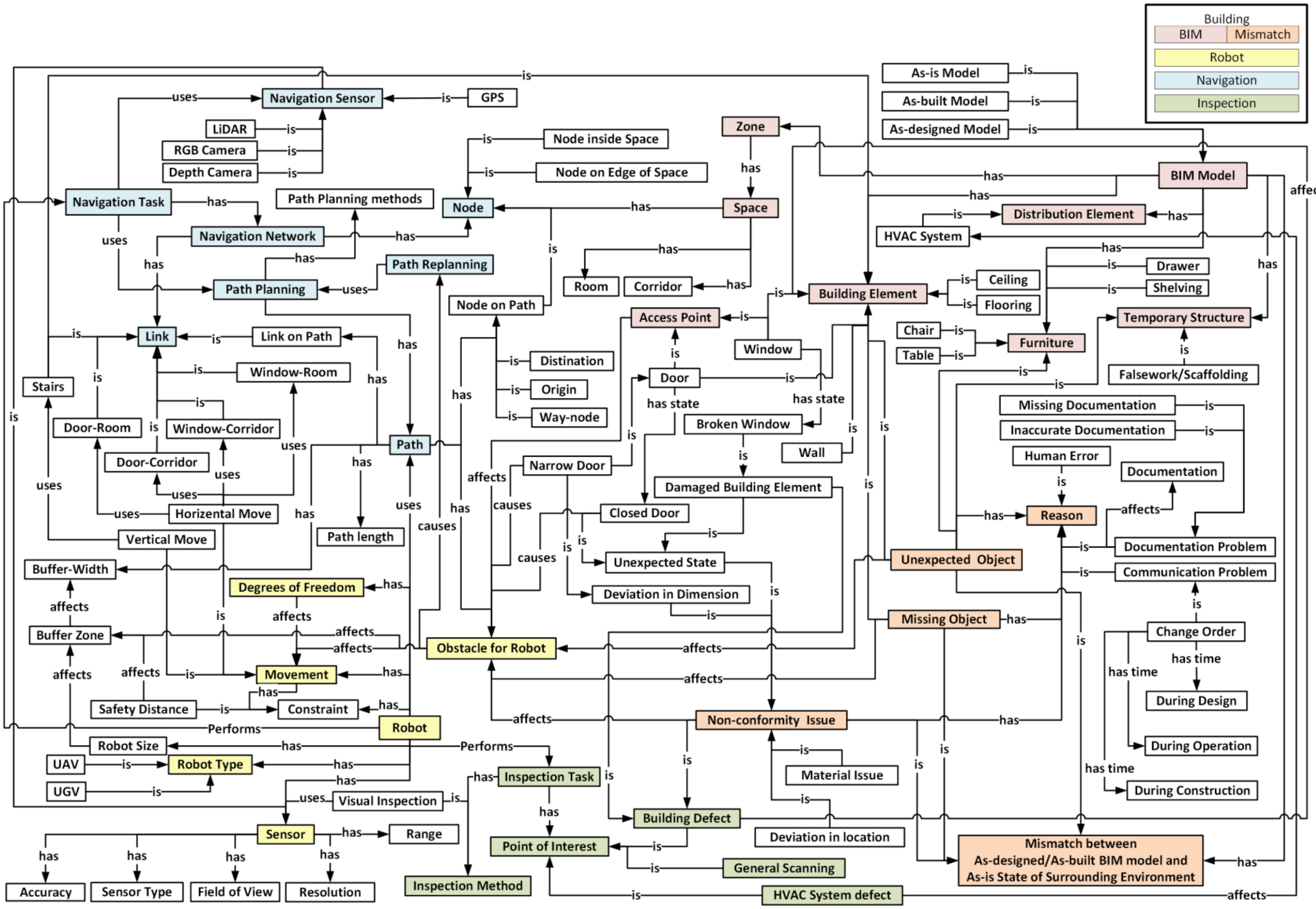


Figure 5-2. Concepts and relationships of OBRNIT: (a) Main concepts

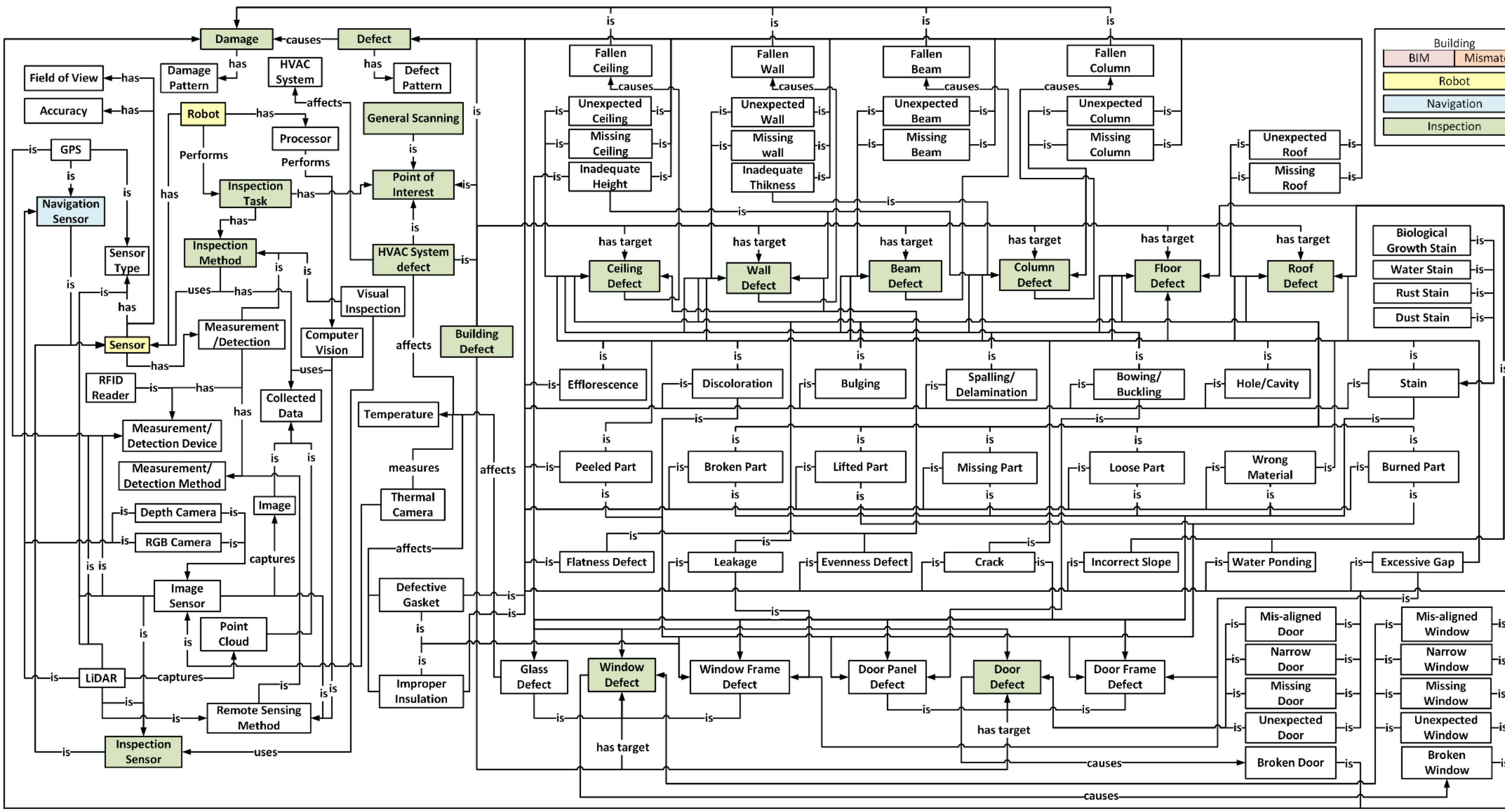


Figure 5-2. Concepts and relationships of OBRNIT (continued): (b) Inspection task concepts

5.3.1 Robot Concepts

The robot concepts of OBRNIT cover the main functions of a robot along with the related knowledge of the inspection and navigation tasks. Declarative abstract knowledge about the tasks and environment should be encoded in the robot controller and used to determine proper actions for a specific task.

KnowRob ontology represents semantic models using object detection applied to the acquired point clouds enriched by encyclopedic, common-sense, and action-related knowledge [134]. From the BIM point of view, this ontology is primitive and does not provide full support of building elements. For example, the concept of a wall is only mentioned as a part of the edges of a region's surface and does not have dimensions, material, connectivity, type, etc. Walls may play a major role in inspection and navigation tasks because they define the boundaries of robots' movements or can be obstacles, or the main target of inspection. Other building elements, such as ceilings, columns, and windows, are not covered in KnowRob.

As shown in Figure 5-2(a), mobility and sensing are the two main functions of robots. The mismatches between the path found based on the non-updated BIM model (Section 0) and the as-is state of the surrounding environment (Section 5.3.3) will cause an obstacle for the robot movement, and consequently its performance. Robot concepts cover basic attributes (e.g. type, size), robots' performance (e.g. movements, degrees of freedom (DOF)), robots' constraints (e.g. safety distance), and sensors for navigation and inspection tasks. The DOF define the modes for the motion capability of the robot. The types of robots considered In OBRNIT are UAV and UGV. UGV refers to any type of crawling, climbing, and other ground-based robots. The movement of UAVs is in the 3D spaces of the building. However, UGVs move following the floors and may be able to climb the stairs. In this case, there are some constraints on the movement, such as the maximum height of a stair step that they can climb. Also, the flying movement of a UAV has constraints, which mainly depend on the size of the UAV.

Sensors can be used for inspection (e.g. RGB camera, thermal camera) and navigation purposes (e.g. depth camera, GPS). LiDAR and cameras are two different types of sensors. Cameras collect images, which can be RGB/depth/thermal images. LiDAR scanners is a remote sensing method, which collects point cloud from the environment. The accuracy and field of the view of the robots' sensor, as well as its type, affect the robot's inspection performance. The concepts related to inspection tasks are explained in Section 5.3.4. The main robot concepts and relationships in Figure 5-2(a) are summarized in Table 5-2.

Table 5-2. Main robot concepts and relationships

Concepts	Relationship	Concepts
Robot	*performs*	*Inspection Task, Navigation Task*
	has	*Constraint, Degrees of Freedom, Movement, Processor, Robot Size, Robot Type, Sensor*
	uses	*Path*
UAV, UGV	*is*	*Robot Type*
Obstacle for Robot	*affects*	*Buffer Zone, Movement*
	causes	*Path Replanning*
Robot Size, Safety Distance	*affects*	*Buffer Zone*
Buffer Zone	*affects*	*Buffer-Width*
Safety Distance	*is*	*Constraint*
Processor	*performs*	*Computer Vision*
Sensor	*has*	*Accuracy, Field of View, Measurement, Range, Resolution, Sensor Type*
Degrees of Freedom	*affects*	*Movement*
Movement	*has*	*Constraint*
Horizontal Move, Vertical Move	*is*	*Movement*
Vertical Move	*uses*	*Stairs*
Horizontal Move	*uses*	*Door-Corridor, Door-Room, Window-Corridor, Window-Room*

5.3.2 Building Concepts

The BIM model can provide information about the environment of the robotic inspection. Every building element that affects the robot navigation and inspection processes should be included in OBRNIT. As explained in Section 5.3, the integration process starts with integrating BEO. The required concepts, which are not included in BEO, are added from ifcOWL ontology, or defined based on the required concepts for robotic navigation and inspection. The process of integrating BIM concepts with OBRNIT aims to link the available BIM concepts with the developed OBRNIT concepts, including related building concepts (e.g. BIM mismatch concepts), robot concepts, and inspection and navigation tasks concepts. Some research focused on robots that can open a closed-door with specific access control or use a handle, knob, or button [321]. For example, Cobalt Access [123] can open locked doors by using the door's access control reader. However, passing through locked doors without human intervention is still one of the main issues for most of the robots. Figure 5-3 shows robot access control concepts in OBRNIT. The state of the door can be open or closed, locked or unlocked, mechanically locked, or electronically locked.

Table 5-3 shows examples of building concepts reused from BEO, BOT, ifcOWL, and new concepts defined in OBRNIT. Building concepts of OBRNIT includes the following: (1) Concepts reused from BEO ontology; (3) Concepts reused from BOT; (4) Concepts reused from ifcOWL: Some necessary concepts, which are not included in BEO (e.g. the *furniture* concept), are added from ifcOWL ontology. HVAC elements are also added from ifcOWL ontology in order to consider HVAC system defects; (5) Concepts adopted from Building Management Systems (BMS): Some concepts related to the state of the door are required for navigation purposes. These concepts are adopted from BMS; and (6) New building concepts defined based on OBRNIT needs: These concepts include BIM mismatch concepts. In addition, the following relationships are defined to link building-related concepts to navigation and inspection concepts: (1) Relationships to define the links between spaces for navigation paths (e.g. door-corridor), (2) Relationships to define a BIM object as the point of interest of inspection, and (3) Relationships to define obstacles or constraints for the robot movement (e.g. a narrow door).

Table 5-3. Examples of reused, adopted, and new building concepts in OBRNIT

Concept's source	Example concepts
Concepts reused from BEO	Beam, column, covering (ceiling and flooring), door, stair, wall, window
Concepts reused from BOT	Space, zone
Concepts reused from ifcOWL	HVAC system
	Room, corridor
	Furniture (e.g. table, shelving)
Concepts adopted from BMS	Open door, closed door, locked door, unlocked door
New building concepts	Access point
	Temporary structure (e.g. falsework/scaffolding)
	BIM model (as-designed, as-built, as-is)
	Mismatch between as-designed/as-built BIM and as-is state of surrounding environment (missing objects, unexpected objects, non-conformity issues), deviation in dimension, deviation in location, material issue, unexpected state, damaged building element
	Mismatch reason (communication problem, documentation problem, human error), change order, inaccurate documentation, missing documentation

Furthermore, the mismatches between the as-designed or as-built BIM model and the as-is state of the surrounding environment should be semantically represented in OBRNIT. By implementing a BIM model of a building, all the information about the elements is available through this model. Identifying the potential types of mismatches is the first step to define a logic-based robotic inspection system that can reduce delays and reworks. Having a rich semantic database about the spaces and building components can enhance the overall efficiency of the robot. Also, the information about the path has a major role when the goal is finding the optimal route and avoiding collisions with existing barriers. Different spaces in the building can form different zones. Spaces (e.g. rooms) can be used to generate nodes for generating the path of the robot, which is explained in the Section 5.3.3. The dimensions of a space can be used to define these nodes inside or on the edges of the space. The main building spaces for robot path planning are rooms, corridors, and stairs. The functionality of rooms and specifications of spaces can be different (e.g. security level for access to public/restricted room) [175].

The mismatches between the information in the available BIM model and the reality cause navigation problems for robots. The preliminary model of the BIM at the design phase is called as-designed BIM. As-built BIM includes all the changes during the construction phase. As-is BIM includes the updated information of the facility and all the changes (e.g. repair, replacement, etc.) at the time of data collection. In some cases, the lack of adequate communication in the design phase, insufficient documentation, or errors of the contractor can turn into unexpected results including information mismatches between the as-designed BIM model and the as-is state of the building. The same problem can occur in the operation phase, where renovation issues can cause mismatches between the non-updated as-built BIM and the as-is state of the building. The assumption in OBRNIT is that the path planning is based on a reference BIM model, but this model is not as-is and reliable. The semantic mismatch between the as-designed BIM model (or as-built

BIM model) and the as-is state of the surrounding environment could be caused by one of the following problems: (1) there is an object in the BIM model, which does not exist in reality. This problem can be the result of design changes during the construction phase (e.g. removing a door) where the changes are not applied in the BIM model; (2) there is an object in the building which is not included in the last updated BIM model; or (3) there is a discrepancy between the BIM model and the actual building with respect to objects' attributes, such as location or dimensions. As shown in Figure 5-2(a), these problems that the robot can face in a building are classified as missing objects, unexpected objects, and non-conformity issues. Each of these issues could be linked with fixed or mobile objects. For instance, building elements (e.g. access points) can be missing objects, and furniture and temporary structures (e.g. falsework) can be unexpected objects. Also, classes related to non-conformity should cover material issues, unexpected states (e.g. damaged building element, a closed-door which is expected to be open), and deviation in location or deviation in dimensions (e.g. narrow door), etc. As shown in Figure 5-2(a), each of the main mismatch entities has one or more causes and effects. For instance, some of the causes are human errors, documentation problems (e.g. change request was not documented), and communication problems during the different phases of AEC/FM. Each of these reasons causes a problem that can be described as an effect (e.g. obstacles for a robot). A narrow door (i.e. deviation in dimensions) or a closed-door (i.e. different states from what is expected) are examples of non-conformity that can cause problems for a robot during its operation. The main robot concepts and relationships in Figure 5-2(a) are summarized in Table 5-4.

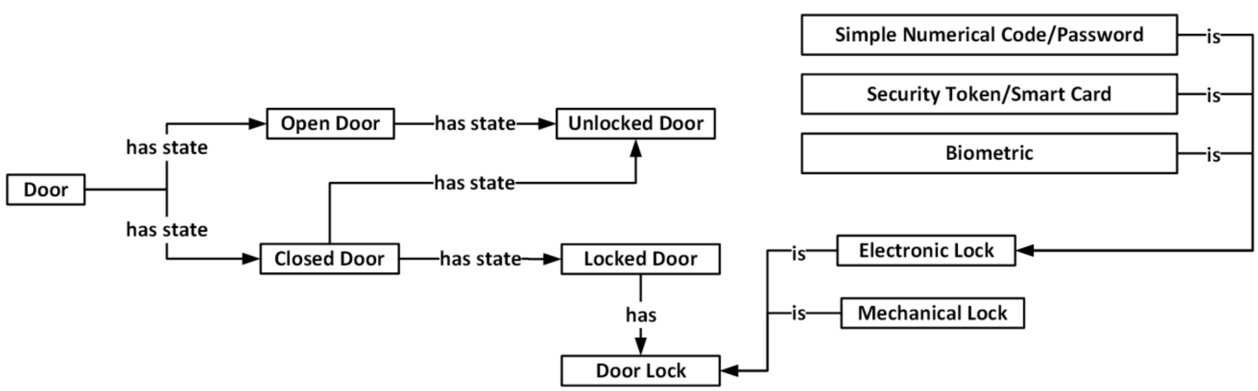


Figure 5-3. Robot access control concepts

Table 5-4. Main building concepts and relationships

Concepts	Relationship	Concepts
As-built Model, As-designed Model, As-is Model	is	BIM Model
BIM Model	has	Building Element, Distribution Element, Furniture, Mismatch between As-designed/As-built BIM model and As-is state of surrounding Environment, Zone, Temporary Structure
Ceiling, Door, Floor, Stairs, Wall, Window	is	Building Element
HVAC	is	Distribution Element
Zone	has	Space
Space	has	Corridor, Node, Room
Chair, Drawer, Shelving, Table	is	Furniture
Falsework/Scaffolding	is	Temporary Structure
Door, Window	is	Access Point
Window	has state	Broken Window
Broken Window	is	Damaged Building Element
Damaged Building Element	is	Building Defect
Door	has state	Closed Door, Locked Door, Open Door, Unlocked Door
Closed Door, Narrow Door	causes	Obstacle for Robot
Missing Object, Non-conformity Issue, Unexpected Object	is	Mismatch between As-designed/As-built BIM model and As-is state of surrounding Environment
	has	Reason
	affects	Obstacle for Robot
Access Point	affects	
Deviation in Dimension, Deviation in Location, Material Issue, Unexpected State	is	Non-conformity Issue
Closed Door, Damaged Building Element	is	Unexpected State
Narrow Door	is	Deviation in Dimension
Non-conformity Issue	is	Building Defect
Missing Object	is	Building Element
Unexpected Object	is	Building Element, Furniture, Temporary Structure
Communication Problem, Documentation Problem, Human Error	is	Reason
Inaccurate Documentation, Missing Documentation	is	Documentation Problem
Documentation Problem	affects	Documentation
Change Order	is	Communication Problem
	has time	During Construction, During Design, During Operation

5.3.3 Navigation Concepts

The navigation task in OBRNIT refers to the act of performing navigation by the robot. As shown in Figure 5-2(a), navigation concepts cover the main information related to the path of the robot including nodes and links, which can be used for path planning. The navigation task has a network, and it uses the information of this network for path planning. Different types of navigation sensors can be used including GPS, LiDAR scanner, and depth camera. A LiDAR scanner can be used to support both the inspection task (Section 5.3.4) and the navigation task. The robot uses the path for performing the navigation task. A path has attributes including the length, direction, and buffer-width. A node can be the origin or destination of a path, or a way-node on the path. Spaces (e.g. room, corridor) and access point elements of a building (e.g. doors, windows) can be nodes of a path. For example, if a robot must move from a corridor to a room, the center point of the corridor is the origin node, the center point of the room is the destination node, and the door of the room is a way-node. Positions of the way-nodes vary based on the obstacles on the way of the robot. These obstacles may be unexpected objects detected by the robot as explained in the Section 0. New links on the path connect these way-nodes to the origin and the destination nodes and each other [165]. Links connect nodes and define the direction of the path. Examples of links are the links connecting a window to a room (in case of UAV), a door to a corridor, or a door to a room, based on the defined building elements and spaces as explained in Section 0. Links can be horizontal or vertical (e.g. stairs' links are vertical). Figure 5-4 shows a simple example, where Node 1 at the center of Room 1 is the origin node and Node 2 at the center of Room 2 is the destination node. Link 1 is the shortest link to connect the origin to the destination nodes; but it crosses two obstacles (i.e. the walls of the rooms). Nodes 3 to 7, which are way-nodes on the path, and the links between them are added to create an obstacle-free path (path A). As explained in the Section 0, the state and dimensions of access points (e.g. doors and windows) are important to enable the robot movement over the path. For example, a closed or narrow door can be an obstacle for the robot's navigation. The main navigation concepts and relationships in Figure 5-2(a) are summarized in Table 5-5.

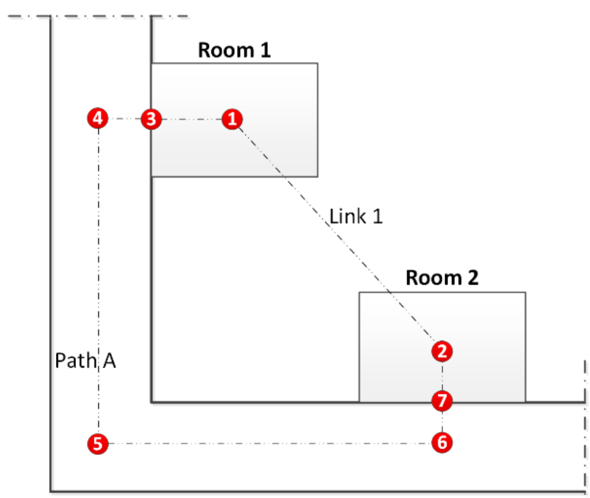


Figure 5-4. Example of using BIM for path planning

Table 5-5. The main navigation concepts and relationships

Concepts	Relationship	Concepts
Navigation Task	*has*	*Navigation Network*
	uses	*Navigation Sensor*
Navigation Network	*has*	*Node, Link*
Depth Camera, LiDAR, RGB Camera	*is*	*Inspection Sensor, Navigation sensor, Sensor Type*
GPS	*is*	*Navigation Sensor, Sensor Type*
Navigation Sensor	*is*	*Sensor*
Node inside Space, Node on Edge of Space, Node on Path	*is*	*Node*
Space	*has*	
Destination, Origin, Way-node	*is*	*Node on Path*
Door-Corridor, Door-Room, Link on Path, Stair, Window-Corridor, Window-Room	*is*	*Link*
Path Planning	*has*	*Path, Path Planning Method*
Path Replanning	*uses*	*Path Planning*
Path	*has*	*Buffer-Width, Link on Path, Node on Path, Obstacle for Robot, Path Length*

5.3.4 Inspection Concepts

Section 4.4.1.1 explained the main inspection concepts of OCSD, which is mainly dedicated to the inspection of concrete surface defects. Similar to OCSD, OBRNIT covers some types of concrete surface defects such as cracks and spalls. However, OBRNIT is not only about concrete surface defects and covers different types of building element defects (e.g. missing parts). Moreover, different inspection methods can be used based on OCSD to inspect concrete surface defects. OBRNIT mainly focuses on robotic inspection based on the remote sensing method to perform the robotic inspection.

The inspection is the main task of the robot in OBRNIT and is mostly performed using vision sensors (e.g. LiDAR scanners, cameras). As explained in Section 5.3, DOT concepts are integrated to link with building defects concepts of OBRNIT. Examples of concepts reused from DOT are: *damage*, *damage pattern*, *documentation*, and *defect*. In this section, the attributes of inspection-related tasks of OBRNIT are defined based on common defects in buildings [322]. OBRNIT covers only major types of defects related to ceiling, beam, column, wall, floor, roof, door, and window elements. However, it does not cover all types of building defects. Building elements can have different types of defects that the robot can inspect based on their material. For example, concrete surfaces can have defects such as cracks, spalling, and efflorescence. Moreover, some types of defects such as missing roofs can be detected after a disaster occurrence. As shown in Figure 5-2(b), The inspection task has an inspection method, which can be visual inspection or a method for the measurement/detection of physical conditions (e.g. broken glass) or environmental conditions (e.g. temperature). The method of inspection is based on the sensor's measurement/detection and acquired datasets. Measurement/detection devices for inspection are radio-frequency ID (RFID) readers, image sensors (i.e. RGB and thermal cameras), and LiDAR scanners. RFID is a technology that uses radio frequencies to detect the objects. RFID tags can be attached to separate object instances and linked with BIM information [323]. Inspection using cameras produces images while inspection using LiDAR scanners produces point clouds. These images and point clouds can be used to detect surface defects, deformations, non-conforming elements, etc. The quality of LiDAR data is defined based on two main parameters of density and [324]. The density of a point cloud is represented by the number of points in a specific area. The distance between the two points which are next to each other defines the point spacing. Computer vision methods can be used for anomaly detection on the collected data [3]. Also, the information of computer vision methods can be used for obstacle detection and navigation tasks (Section 5.3.3).

Defects can cause damage to the building elements. A damage occurs when a defective element loses its function. For example, water leakage from the ceiling is a defect, which can cause damage to the ceiling elements over time. There are several causes for defects formation and damage occurrence. Defects and damages have various patterns and characteristics.

OBRNIT covers two main types of defects, including building defects and HVAC system defects. The point of interest of the inspection task is defined by the inspection purpose, which can be general scanning, inspecting mechanical systems (e.g. HVAC), or detecting building defects. General robotic scanning aims to update the BIM model or to collect data of a hazardous building, which is unsafe to inspect by human inspectors. The malfunctions of the HVAC system affect the environment temperature and air quality. Defected HVAC elements or related building elements (e.g. improper insulation) can be evaluated by thermal cameras. In the case of inspecting building defects (e.g. surface/material defects), specific building elements (e.g. doors, walls, floors, etc.) are the points of interest, and each of them can be a target for the inspection task. For example, defective gasket and improper insulation are some types of window frame defects; and the ceiling can be inspected for different types of defects such as leakage, stain, discoloration, bulging, spalling, delamination, and efflorescence. Some issues related to non-conformity can be considered as building defects, as discussed in Section 0. Furthermore, the detected defects can be used to update the available BIM model to create an up-to-date as-is BIM model. The inspection concepts and relationships in Figure 5-2(b) are summarized in Table 5-6.

Table 5-6. The main inspection concepts and relationships

Concepts	Relationship	Concepts
Inspection Task	*has*	*Inspection Method, Point of Interest*
Measurement/Detection, Visual Inspection	*is*	*Inspection Method*
Visual Inspection	*uses*	*Inspection Sensor*
Inspection Method	*has*	*Collected Data*
	uses	*Sensor*
Computer Vision	*is*	*Remote Sensing Method*
	uses	*Collected Data*
Measurement/Detection	*has*	*Measurement/Detection Device, Measurement/Detection Method*
Remote Sensing Method	*is*	*Measurement/Detection Method*
GPS, Image Sensor, LiDAR, RFID Reader	*is*	*Measurement/Detection Device*
Depth Camera, RGB Camera, Thermal Camera	*is*	*Image Sensor*
Thermal Camera	*measures*	*Temperature*
Image Sensor	*captures*	*Image*
	is	*Inspection Sensor, Navigation Sensor, Remote Sensing Method, Sensor Type*
Image, Point Cloud	*is*	*Collected Data*
LiDAR	*captures*	*Point Cloud*
	is	*Inspection Sensor, Navigation Sensor, Remote Sensing Method, Sensor Type*
Inspection Sensor	*is*	*Sensor*
Building Defect, General Scanning, HVAC System Defect	*is*	*Point of Interest*
Bowing/Buckling, Broken Part, Building Defect, Bulging, Burned Part, Crack, Defective Gasket, Discoloration, Efflorescence, Evenness Defect, Excessive Gap, Flatness Defect, HVAC System Defect, Hole/cavity, Improper Insulation, Inadequate Height, Inadequate Thickness, Incorrect Slope, Leakage, Lifted Part, Loose Part, Mis-aligned Door, Mis-aligned Window, Missing Beam, Missing Ceiling, Missing Column, Missing Door, Missing Part, Missing Roof, Missing Wall, Missing Window, Narrow Door, Narrow Window, Peeled Part, Spalling/Delamination, Stain, Unexpected Beam, Unexpected Ceiling, Unexpected Column, Unexpected Door, Unexpected Roof, Unexpected Wall, Unexpected Window, Water Ponding, Wrong Material	*is*	*Defect*
Broken Door, Broken Window, Fallen Beam, Fallen Ceiling, Fallen Column, Fallen Wall	*is*	*Damage*
Defect	*causes*	
	has	*Defect Pattern*

Table 5-6. The main inspection concepts and relationships (continued)

Concepts	Relationship	Concepts
Damage	*has*	*Damage Pattern*
HVAC System Defect	*affects*	*HVAC System, Temperature*
Building Defect	*has target*	*Beam Defect, Ceiling Defect, Column Defect, Door Defect, Floor Defect, Roof Defect, Wall Defect, Window Defect*
Bowing/Buckling, Broken Part, Crack, Discoloration, Efflorescence, Hole/Cavity, Missing Beam, Peeled Part, Spalling/Delamination, Stain, Unexpected Beam, Wrong Material	*is*	*Beam Defect*
Bowing/Buckling, Broken Part, Bulging, Burned Part, Crack, Discoloration, Efflorescence, Evenness Defect, Flatness Defect, Hole/Cavity, Leakage, Lifted Part, Loose Part, Missing Ceiling, Missing Part, Peeled Part, Spalling/Delamination, Stain, Unexpected Ceiling, Wrong Material	*is*	*Ceiling Defect*
Bowing/Buckling, Broken Part, Crack, Discoloration, Efflorescence, Hole/Cavity, Inadequate Height, Inadequate Thickness, Missing Column, Peeled Part, Spalling/Delamination, Stain, Unexpected Column, Wrong Material	*is*	*Column Defect*
Door Frame Defect, Door Panel Defect, Inadequate Height, Mis-aligned Door, Missing Door, Narrow Door, Unexpected Door	*is*	*Door Defect*
Broken Part, Burned Part, Crack, Discoloration, Lifted Part, Loose Part, Missing Part, Peeled Part, Stain, Wrong Material	*is*	*Door Panel Defect*
Broken Part, Burned Part, Crack, Discoloration, Excessive Gap, Lifted Part, Loose Part, Missing Part, Peeled Part, Stain, Wrong Material	*is*	*Door Frame Defect*
Bowing/Buckling, Broken Part, Bulging, Burned Part, Crack, Discoloration, Efflorescence, Excessive Gap, Hole/Cavity, Incorrect Slope, Leakage, Lifted Part, Loose Part, Missing Part, Peeled Part, Spalling/Delamination, Stain, Water Ponding, Wrong Material	*is*	*Floor Defect*
Broken Part, Bulging, Burned Part, Crack, Discoloration, Efflorescence, Hole/Cavity, Incorrect Slope, Leakage, Lifted Part, Loose Part, Missing Roof, Missing Part, Peeled Part, Stain, Unexpected Roof, Water Ponding, Wrong Material	*is*	*Roof Defect*

Table 5-6. The main inspection concepts and relationships (continued)

Concepts	Relationship	Concepts
Bowing/Buckling, Broken Part, Bulging, Burned Part, Crack, Discoloration, Efflorescence, Evenness Defect, Flatness Defect, Hole/Cavity, Inadequate Height, Leakage, Lifted Part, Loose Part, Missing Part, Missing Wall, Peeled Part, Spalling/Delamination, Stain, Unexpected Wall, Wrong Material	*is*	*Wall Defect*
Glass Defect, Inadequate Height, Mis-aligned Window, Missing Window, Narrow Window, Unexpected Window, Window Frame Defect	*is*	*Window defect*
Broken Part, Burned Part, Crack, Defective Gasket, Discoloration, Excessive Gap, Improper Insulation, Leakage, Lifted Part, Loose Part, Missing Part, Peeled Part, Stain, Wrong Material	*is*	*Window Frame defect*
Broken Part, Crack, Loose Part, Missing Part, Stain, Wrong Material	*is*	*Glass Defect*
Biological Growth Stain, Chemical Stain, Dust Stain, Rust Stain, Water Stain	*is*	*Stain*
Broken Glass, Defective Gasket, Improper Insulation	*affects*	*Temperature*

5.4 Evaluation and Discussion

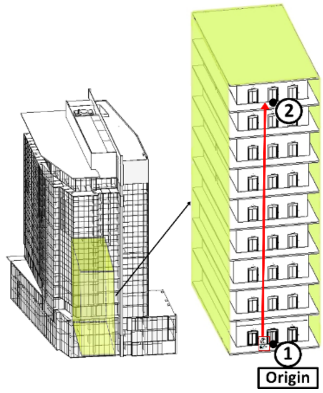
Ontology evaluation is a main step in the ontology development, which refers to the process of evaluating if the developed ontology is correct and if it represents the main concepts and relationships [312, 313]. Two evaluation methods are used for evaluating the usefulness of OBRNIT: (1) application-based evaluation, and (2) qualitative criteria-based evaluation. The application-based evaluation is the evaluation of a developed ontology using a case study. This approach judges whether the ontology is suitable to perform the task and meets the objectives. However, it is not used to evaluate the design or the contents of the ontology [312]. On the other hand, the qualitative criteria-based evaluation approach is used to evaluate the ontology based on criteria such as clarity, coherence, correctness, and expandability. As explained in Section 4.5.1 consistency criteria are tested using the HermiT OWL Reasoner in the verification process [308, 309]. HermiT OWL Reasoner, which is based on the hypertableau algorithm, is used for identifying subsumption relationships and consistency evaluation [311]. The reasoner clarified some inconsistencies in the ontology. As described in Section 5.2, these results were utilized as feedback and input to Step 3 to fix the problems before going to the final step.

5.4.1 Case Study

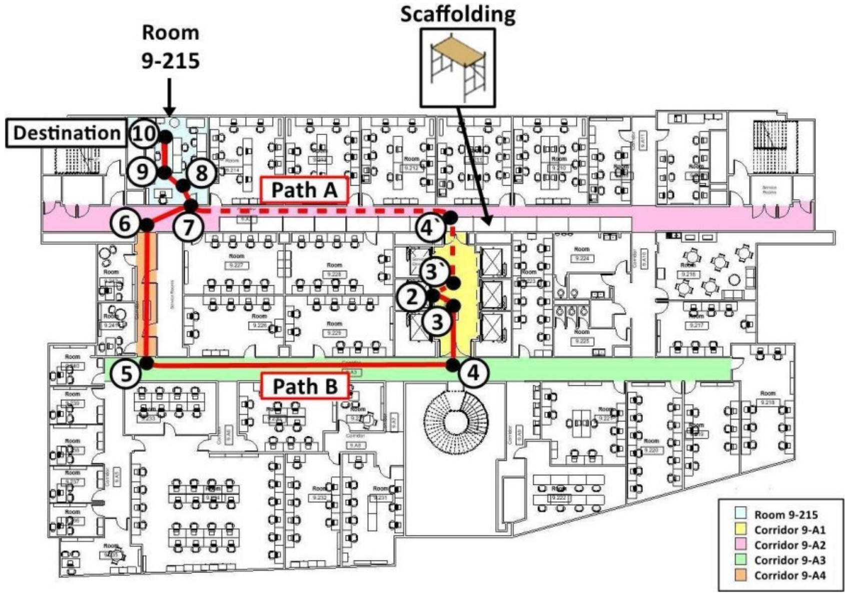
Figure 5-5 shows a hypothetical case study of using an inspection robot to find the leakage in one of the rooms on the ^{9t}h floor in a building at Concordia University. The aim of the case study is to demonstrate the applicability of OBRNIT based on specific information about the building extracted from a BIM model and information about the inspection robot. The assumption is that the robot partially knows the environment based on a non-updated BIM model. After defining the inspection point of interest in Room 9-215, which is leakage in the ceiling, the robot navigates to reach this point of interest to perform the inspection task. Path planning is based on a reference as-built BIM model. The inspection robot will use an image sensor to capture images of the ceiling. Table 5-7 shows the inspection task specifications. The FLIR PackBot robot [325] is assumed as the robot used in the case study. The robot type is UGV, and it has horizontal and vertical (e.g. climbing the stairs) mobility. The inspection robot specifications are shown in Table 5-8.

Examples of BIM-based information are shown in Table 5-9. This information includes the objects in Room 9-215 and the inspection point of the interest, as well as the spaces/objects from the elevator on the 9th floor to the door of Room 9-215. This table only contains the walls of Room 9-215, and it does not show the other walls of the entire 9th floor. The navigation network and path planning concepts for the desired path are shown in Table 5-10. The origin node is in front of the elevators on the 1st floor, and the destination node is inside Room 9.215. The path has three parts. The first part is the vertical movement in the elevator from the origin node to the 9th floor (Figure 5-5(a)). The second part of the path is the horizontal movement from the 9th floor elevator hall to the door of Room 9-215 (Figure 5-5(b)). The shortest path (Path A) uses Corridors 9-A1 and 9-A2 (Nodes 2, 3`, 4`, and 7). However, this path is blocked with scaffoldings, which are used for a renovation project, and create an obstacle for the robot. Therefore, the robot must follow a longer path (Path B) to reach the room. The robot obtains information about the scaffoldings from an up-to-date BIM model, if available, or from its sensing ability. Having an up-to-date BIM model (i.e. as-is model) results in a higher confidence level with respect to obstacles. After detecting the obstacle, the robot replans a new path (Path B). The involved corridors to reach Room 9-215 in Path B are Corridor 9-A1, Corridor 9-A3, Corridor 9-A4, and Corridor 9-A2, which contain Nodes 2,3,4,5, 6, and 7. The last part of the path is the horizontal movement inside the room from the door to the destination node (i.e. the inspection point of interest) as shown in Figure 5-5(c). The robot will learn from performing the navigation task. After finding the mismatch with the as-built BIM model (i.e. the scaffoldings), the robot stores them as a reference point for performing the next tasks. Figure 5-5(d) shows the robot collecting images of leakage in the room.

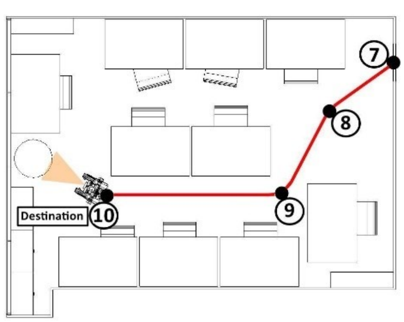
The case study demonstrates that OBRNIT can answer all the competency questions (Table 5-1) and it covers all the concepts necessary for the planning of the robotic building navigation and inspection. Integrating mobility characteristics of the robot and the knowledge about the surrounding environment have been used to help the robot define the appropriate path based on the robot type and constraints and meet the requirement for the inspection task. The movement of the robot has constraints, which mainly depend on the size of the robot (e.g. a door, which is narrower than the width of the robot, is a constraint for the robot movement). The ontology has been used to help the robot use a suitable sensor for the specific inspection task. Furthermore, the field of view of the camera of the robot is a constraint for the inspection task (for this specific case is not a problem). Moreover, the robot benefits from the BIM model to define the path based on defining the nodes and links of the path. In addition, the robot benefits from the BIM model information to locate the inspection object. The case study shows how several concepts are extracted from the BIM model of the building. Examples of these concepts include concepts related to the navigation task (moving to the specific floor and the specific room, and then moving to the point of interest in the room), as well as the inspection task (orienting the camera to the leakage area based on the field of view and collecting images).

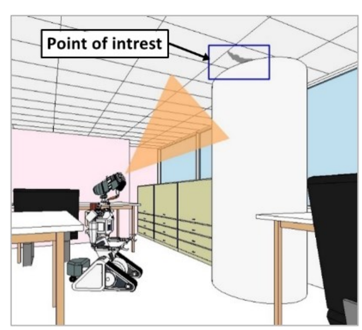


(a) BIM model of the whole building with the green part showing the vertical movement of the robot from the ground floor to the 9th floor



(b) BIM model of the ^{9t}h floor and involved corridor spaces for the path from elevator to Room 9-215





(c) Path of inspection robot in Room 9-215 **(d) Robot collecting images of leakage in the room**

Figure 5-5. Case study of using an inspection robot

Table 5-7. Inspection task specifications

Concept in OBRNIT	
Point of interest	Ceiling defect
Type of defect	Leakage
Inspection method	Measurement/Detection
Measurement/Detection device	Image sensor

Table 5-8. Inspection robot main specifications

Concept in OBRNIT		Specifications
Robot type (UGV)		FLIR PackBot
Movement		Horizontal move Vertical move
Sensor type (camera)		4 RGB wide angel cameras with zoom and illumination (supports an optional 5^{th} camera with thermal capability)
Field of view		60° to 110°
Degrees of freedom		8
Size	Length	88.9 cm
	Width	52.1 cm
	Height	17.8 cm

Table 5-9. Examples of IFC-based information of Room 9.215 and the spaces/objects outside the room

IfcEntity		Name	Tag	Concept in OBRNIT
Room 9-215	IfcColumn	M_Round Column: 610mm Diameter	364991	Column
	IfcCovering	Compound Ceiling: 600 x 600 mm grid 2, white	378778	Ceiling (point of interest for leakage inspection)
	IfcCurtainWall	Curtain Wall: Storefront	363008	Curtain wall
	IfcDoor	M_Single-Flush:0915 x 2134mm:379291	379291	Door
	IfcFurniture	M_Furniture_System-Standing_Desk-Rectangular: 1500x750 mm	372571	Table
	IfcFurniture		373006	
	IfcFurniture		373129	
	IfcFurniture		373192	
	IfcFurniture		373239	
	IfcFurniture		373486	
	IfcFurniture		373630	
	IfcFurniture		374087	
	IfcFurniture		374640	
	IfcFurniture		374723	
	IfcFurniture	M_Chair–- Executive	376992	Chair
	IfcFurniture		377394	
	IfcFurniture		377583	
	IfcFurniture		377646	
	IfcFurniture		377711	
	IfcFurniture		377776	
	IfcFurniture		377859	
	IfcFurniture		377916	
	IfcFurniture		377983	
	IfcFurniture		378050	
	IfcFurniture	M_Shelving: 1240 x 0305 x 1500 mm	368134	Shelving
	IfcFurniture		370460	
	IfcFurniture	M_Cabinet-File 4 Drawer:1000 x 0457 mm	367042	Drawer
	IfcFurniture		367118	
	IfcFurniture		368542	
	IfcSlab	Floor: Generic Floor–- 400mm	359802	Flooring
	IfcSpace	Room – 9-215		Room
	IfcWallStandardCase	Basic Wall: Interior–- 138mm Partition	360817	Wall
	IfcWallStandardCase		360875	
	IfcWallStandardCase		360745	
	IfcWallStandardCase		361005	
	IfcWallStandardCase		361035	
	IfcWallStandardCase	Basic Wall: steel- 200 mm concrete masonry unit (CMU)	361214	
IfcBuildingElementProxy		Elevator: 1300 x 950mm	263782	Transport element - elevator
IfcBuildingElementProxy			263642	
IfcBuildingElementProxy			263853	
IfcBuildingElementProxy		Site_Scaffolding	321511	Falsework/scaffolding
IfcSpace		Corridor – 9-A1	-	Corridor
IfcSpace		Corridor – 9-A2	-	
IfcSpace		Corridor – 9-A3	-	
IfcSpace		Corridor – 9-A4	-	
IfcStair		Assembled Stair: ”" max riser 1”" tread	258349	Stair

Table 5-10. Navigation network and path planning concepts

<table>
<tr><th colspan="4">Concept in OBRNIT</th></tr>
<tr><th colspan="2">Parts of the path to reach inspection point of interest</th><th>Links connecting nodes</th><th>Obstacles for robot</th></tr>
<tr><td colspan="2">Vertical path in the elevator shaft from the 1st floor to the 9th floor</td><td>1-2</td><td>-</td></tr>
<tr><td rowspan="2">Horizontal path in corridors on the 9th floor</td><td>Path A</td><td>2-3`-4`-7</td><td>Scaffoldings, walls, door</td></tr>
<tr><td>Path B</td><td>2-3-4-5-6-7</td><td>Walls, door</td></tr>
<tr><td colspan="2">Horizontal path inside Room 9-215</td><td>7-8-9-10</td><td>Chairs, tables</td></tr>
</table>

5.4.2 Criteria-based Evaluation

A survey was conducted to evaluate the adequacy of the semantic representation of the concepts and relationships of OBRNIT. The survey includes eight questions, which are related to the different components of OBRNIT. These questions reflect the coverage of the concepts and semantic relationships between the classes and aim to measure the clarity and comprehensiveness of OBRNIT. The first question was about the respondents' information. The second question was about BIM and its benefits for inspection robots. The third and fourth questions considered the clarity and comprehensiveness of the main concepts of OBRNIT. The fifth and sixth questions were about the clarity and comprehensiveness of the inspection part of OBRNIT. The seventh question was about a statement related to the complexity of interactions between components in OBRNIT. Finally, the last question considered OBRNIT capability for system development. The survey questions of OBRNIT are listed in Table 5-11. A five-point Likert scale is used to get quantitative values of the answers. The survey was sent to 105 internationally recognized experts selected based on their knowledge in BIM, construction, and robotic inspection.

The results of OBRNIT survey including the total number of participants, the respondents' profiles, and the survey answers are available at https://github.com/OBRNIT/Survey-Result.

Table 5-11. The evaluation questions of OBRNIT

Q_1	Name, organization/university, area of expertise, and years of experience.
Q_2	BIM (Building Information Modeling) information extends the declarative knowledge of the environment for the performance of the cognitive robot during navigation tasks. Do you agree with this statement? ○ Strongly agree ○ Agree ○ Neither agree nor disagree ○ Disagree ○ Strongly disagree ○ No answer Comments: ……………………………..……………………………………………………………………...
Q_3	Figure 5-2(a) represents the high-level concepts and relationships of the ontology for BIM-based robotic navigation and inspection tasks. Do you find this representation clear? ○ Very clear ○ Clear ○ Somewhat clear ○ Not so clear ○ Not clear at all ○ No answer Comments: ……………………………..……………………………………………………………………...
Q_4	Based on Figure 5-2(a), do you find the representation comprehensive in integrating the main concepts related to OBRNIT? Comprehensiveness here means representing the main concepts to robotic navigation and inspection using BIM, and not replicating all the concepts of building elements, robots, etc. ○ Very comprehensive ○ Comprehensive ○ Somewhat comprehensive ○ Not comprehensive ○ Missing lots of concepts ○ No answer Comments: ……………………………..……………………………………………………………………...
Q_5	Figure 5-2(b) shows the high-level concepts and relationships of the ontology for inspection-related tasks. Do you find this representation clear? ○ Very clear ○ Clear ○ Somewhat clear ○ Not so clear ○ Not clear at all ○ No answer Comments: ……………………………..……………………………………………………………………...
Q_6	Based on Figure 5-2(b), do you find the representation comprehensive? ○ Very comprehensive ○ Comprehensive ○ Somewhat comprehensive ○ Not comprehensive ○ Missing lots of concepts ○ No answer Comments: ……………………………..……………………………………………………………………...
Q_7	OBRNIT defines complex declarative knowledge where the same concept is used in multiple relationships related to different tasks. For example, a door can be an obstacle for the robot in a navigation task or the main target of inspection in an inspection task. Another example, a LiDAR scanner can be a sensor for navigation or inspection tasks. Do you agree with this statement? ○ Strongly agree ○ Agree ○ Neither agree nor disagree ○ Disagree ○ Strongly disagree ○ No answer Comments: ……………………………..……………………………………………………………………...
Q_8	OBRNIT is expected to help the development of robotic navigation and inspection systems. Do you agree with this statement? ○ Strongly agree ○ Agree ○ Neither agree nor disagree ○ Disagree ○ Strongly disagree ○ No answer Comments: ……………………………..……………………………………………………………………...

5.5 Summary and Conclusions

This chapter developed an integrated ontology, called OBRNIT, to extend BIM applications for robotic navigation and inspection tasks. There are 386 classes, 45 relations, 52 attributes, and 8 individuals in OBRNIT. OBRNIT comprises high-level knowledge of the concepts and relationships related to buildings, robots, and navigation and inspection tasks. BIM is considered as a reference that is integrated with the knowledge model. The HermiT OWL reasoner was used to evaluate the consistency of OBRNIT. The evaluation demonstrates that the ontology is consistent, and all implicit relationships have been represented. The application of OBRNIT was investigated in a case study. In addition, a survey was designed and conducted to evaluate the semantic representation of OBRNIT. The evaluation demonstrates that OBRNIT covers the domain's concepts and relationships up to the point that satisfies the domain experts. Based on the evaluation, OBRNIT was able to give a clear understanding of the concepts and relationships in the domain, and it can be applied for developing robotic inspection systems. OBRNIT is expected to provide the following benefits: (1) OBRNIT can help system engineers involved in developing robotic inspection systems by identifying the different concepts and relationships about robotic inspection and navigation tasks based on BIM information; (2) capturing the essential information from BIM can help to develop a seamless knowledge model to cover the missing parts of BIM; and (3) Using ontological knowledge can help overcome the complexity in interactions between components in the robotic inspection system. OBRNIT can be used as a first step towards logic-based inspection, which can help robots to perform inspection tasks autonomously without the help of human judgment. It is difficult to prove that an ontology enables additional capabilities for systems that would not be possible without it [8]. However, using a central ontological knowledgebase can facilitate the development of robotic inspection systems.

CHAPTER 6. POINT CLOUD-BASED CONCRETE SURFACE DEFECT SEMANTIC SEGMENTATION AND AS-INSPECTED MODELING

6.1 Introduction

As explained in Chapter 2, DNNs have been recently used for detecting 3D objects within 3D point clouds. This chapter proposes an approach for detecting concrete surface defects (i.e. cracks and spalls) using adapted Normal Vector Enhanced DGCNN (NVE-DGCNN). This chapter started with exploring the adapted PointNet++ in the first phase. Then the DGCNN's ability to detect the edges is considered in the next phase. For the DGCNN, the work started with an adapted DGCNN, and then the main network of this research, which is the NVE-DGCNN was investigated in the next step. The proposed method is applied to a point cloud dataset from four concrete bridges in Montreal. The experimental results show the usefulness and robustness of the proposed NVE-DGCNN in detecting concrete surface defects from 3D point cloud data. Furthermore, post-processing of semantic segmentation results was done in this chapter to create an as-inspected BIM model.

6.2 Methodology

6.2.1 Modified CNN Models

DGCNN, which was originally designed to detect indoor building elements, is modified and adapted to automate the inspection process of concrete surface defects, including cracks and spalls. DGCNN is selected in this research because it considers the edge feature, which is the most valuable feature in concrete surface defects semantic segmentation. As explained in Section 6.1, exploring adapted PointNet++ in this research was an initial step. Therefore, in the first phase, PointNet++ was adapted, and sensitivity analysis was performed to identify the effect of hyperparameters and compare the network results with the adapted DGCNN, which is the main network of this research. Therefore, this research is based on three modified CNN models, including adapted PointNet++, adapted DGCNN, and NVE-DGCNN, which are explained as follows:

(a) Adapted PointNet++: The following modifications have been done:

- **Network input parameters:** The network input parameters such as the class number, number of points per block, block size, and stride are modified based on annotated segments, sizes of the structural defects, and density of segmented parts.
- **Sampling size:** In contrast with the original PointNet++, the sampling size is modified based on the smallest dimension of segments in the prepared dataset. The original PointNet++ considered the network's sampling sizes of 10, 20, 40, and 80 *cm*. In this study, the smallest dimension of segments in the prepared dataset is 46 cm. Therefore, the sampling sizes are decreased to 5, 10, 20, and 30 *cm*.
- **Convolving direction:** The *X*-axis is set along the concrete surface, the *Z*-axis is set in the vertical direction of the canonical coordinate system, and the *Y*-axis is set perpendicular to the surface and in the direction of the depth of the defects. The depth of defects is set to have positive *Y* values.
- **Modification related to the normalized coordinates:** The normalized values of the *X* and *Z* coordinates are removed from the PointNet++ network, and the input variables for the adapted PointNet++ are *XYZ*, *RGB*, and *Y'*.

- **Loss function:** In adapted PointNet++, the method of cost-sensitive loss function was selected. The cost-sensitive loss function is a method based on assigning different costs to the imbalanced data classes [326]. This method was selected as this network considers weighted sampling in the model. In the weighted sampling process, different sampling probabilities are assigned to each sample during the training. In this method, every set of batches may have different weight distribution and the weight vector of classes is calculated for every set of batches separately. Therefore, the contribution of classes with higher cost weights will increase by updating the sampling weights [327]. This process makes the model learn from each class equally. The probability of occurrence of class *i* and the cost weight of label *i* are calculated dynamically in the training process by using Equation 6-1 [328-330]. The cost weights of labels tend to be inversely related to the sample's class percentage in the overall dataset.

$$\text{Cost weight of Label}_{\mathrm{i}} = \frac{1}{\log(1.05 + \frac{\mathrm{C_i}}{\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{C_i}})} \qquad \text{Equation 6-1}$$

  Where N is number of classes and C_i is total number of points for each class.

As shown in Figure 6-1, the first hidden layer of adapted PointNet++ has four sub-layers for sampling N_1, N_2, N_3, and N_4 points out of n points in each sub-layer and grouping the points in the radius of 5, 10, 20, and 30 *cm* from the centroid of a region to feed the mini PointNet MLPs. Then four mini PointNet MLPs are applied to interpolate the learning features from previous MLPs. The last layers are two Fully-Connected (FC) layers with a dropout layer in the middle to classify the n points in three classes using the cost-sensitive loss function. The implementation of adapted PointNet++ is explained in Section 6.4.

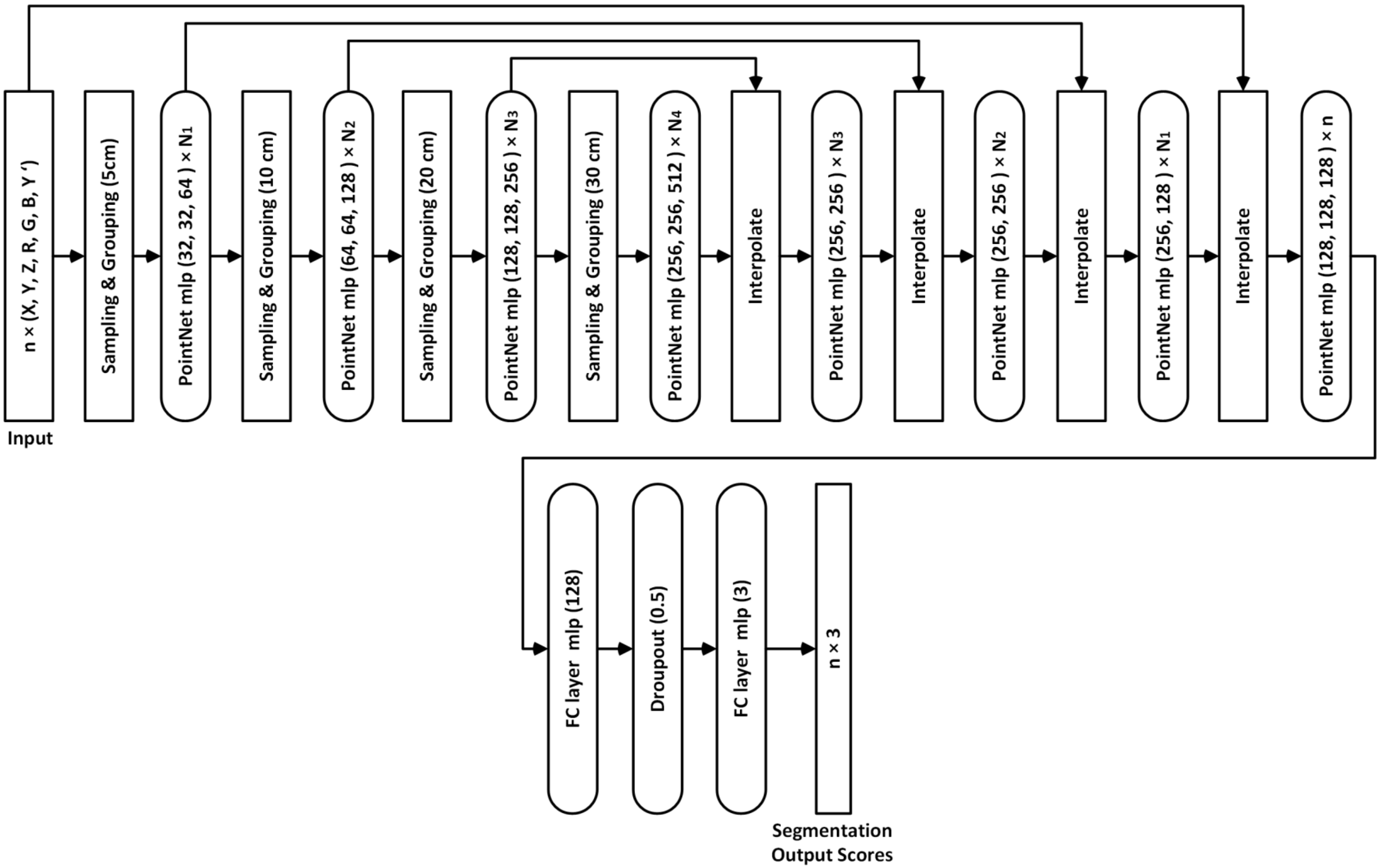


Figure 6-1. Architecture of adapted PointNet++ (adapted from [289])

(b) Adapted DGCNN: The following modifications have been done:

- **Network input parameters:** Similar to the adapted PointNet++, the adapted DGCNN network input parameters such as the class number, number of points per block, block size, and stride are modified based on annotated segments, sizes of the structural defects, and density of segmented parts. Wang et al. [27] used the block size of 1 *m* × 1 *m* on the *XY* surface for rooms with a height of 3 *m* to detect indoor building elements using DGCNN. The number of points of 4,096 is used for their training process. This setting results in a very low density of points for detecting most types of defects in this study (e.g. medium-size spalls). In the adapted DGCNN, the block size of 40 *cm* × 40 *cm* is set based on the sizes of the structural defects in the dataset. Moreover, the density of points in each block is increased by raising the number of points.
- **KNN of the EdgeConv layer:** In this study, in contrast with the original DGCNN, using the normalized *X*, *Y*, and *Z* coordinates of points for the KNN of the EdgeConv layer was considered to be unsuitable because normalization can destroy the critical information about the depth in the *Y* direction, which is much smaller than *X* and *Z* coordinates. A test was performed based on the original DGCNN to examine the effect of using the normalized location values for the KNN, which will be shown in Section 6.4.2. The test result showed that the models' performance declined significantly by considering the normalized location values for KNN. Therefore, the KNN for the EdgeConv layer of the adapted DGCNN is modified to compute the KNN based on the *XYZ* coordinates.
- **Convolving direction:** Similar to the adapted PointNet++, in adapted DGCNN, the depth of defects is set in the direction of the *Y* coordinate value.
- **Modification related to the normalized coordinates:** Similar to the adapted PointNet++, the input point variables of adapted DGCNN are changed from a 9-dimensional vector (*XYZ*, *RGB*, and *X'Y'Z'*) to a 7-dimensional vector (*XYZ*, *RGB*, and *Y'*) by removing the normalized values of the *X* and Z coordinates, and this vector is fed to the network.
- **Loss function:** As the defects' number of points in this research is less than the non-defect number of points (defect points are almost 14% of the whole point cloud), which is known as the issue of *imbalanced datasets*, a weighted softmax cross entropy loss function is utilized to adapt the DGCNN models to the prepared dataset, and the corresponding weight vector is set based on the distribution of points label among the three classes. The method of distribution-based loss function is selected as a label weight score of classes is not dependent on how the instances are sampled, which can be more practical in this case. The label weight score of class *i* is calculated statically and added to the model using Equation 6-2 and Equation 6-3 [281, 331].

$$\text{Distribution of points with class}_\text{i}\text{ label} = \frac{\sum_{\text{i}=1}^{\text{N}} \text{C}_\text{i}}{\text{C}_\text{i}} \qquad \text{Equation 6-2}$$

$$\text{Label weight score of class}_\text{i} = \frac{\text{Distribution of points with class}_\text{i}\text{ label}}{\sum_{\text{i}=1}^{\text{N}} \text{Distribution of poInts with class}_\text{i}\text{ label}} \qquad \text{Equation 6-3}$$

Where *N* is the number of classes and C_i is the total number of points for each class.

Figure 6-2 shows the architecture of the adapted DGCNN. The parameter *K* in the model is the number of edge features calculated for each point in each EdgeConv layer with an input of *n* points.

The segmentation model of adapted DGCNN involves a series of three EdgeConv layers, a max-pooling layer to extract global features of the block, and three FC layers with a dropout layer in the middle to classify the *n* points in three classes using the weighted softmax cross entropy loss function.

(c) NVE-DGCNN: The only modification in addition to the those in adapted DGCNN is the consideration of normal vector in NVE-DGCNN. The adapted DGCNN method was enhanced by considering the normal vector feature in the NVE-DGCNN. The NVE-DGCNN model is modified to compute the KNN based on *XYZ* coordinates, and the contribution of the minority classes is increased during the training of the network by modifying the loss function, as in the case of adapted DGCNN. NVE-DGCNN is modified to consider a 10-dimensional vector by adding the components of the normal vector (*Nx*, *Ny*, and *Nz*) as additional point features to the 7-dimensional vector in adapted DGCNN. The input variables are *XYZ*, *RGB*, *NxNyNz,* and *Y'*. The segmentation model of NVE-DGCNN also involves a series of three EdgeConv layers, a max-pooling, and three fully connected layers. Figure 6-3 shows the architecture of the NVE-DGCNN. The adapted DGCNN and NVE-DGCNN hyperparameters are shown in Table 6-1.

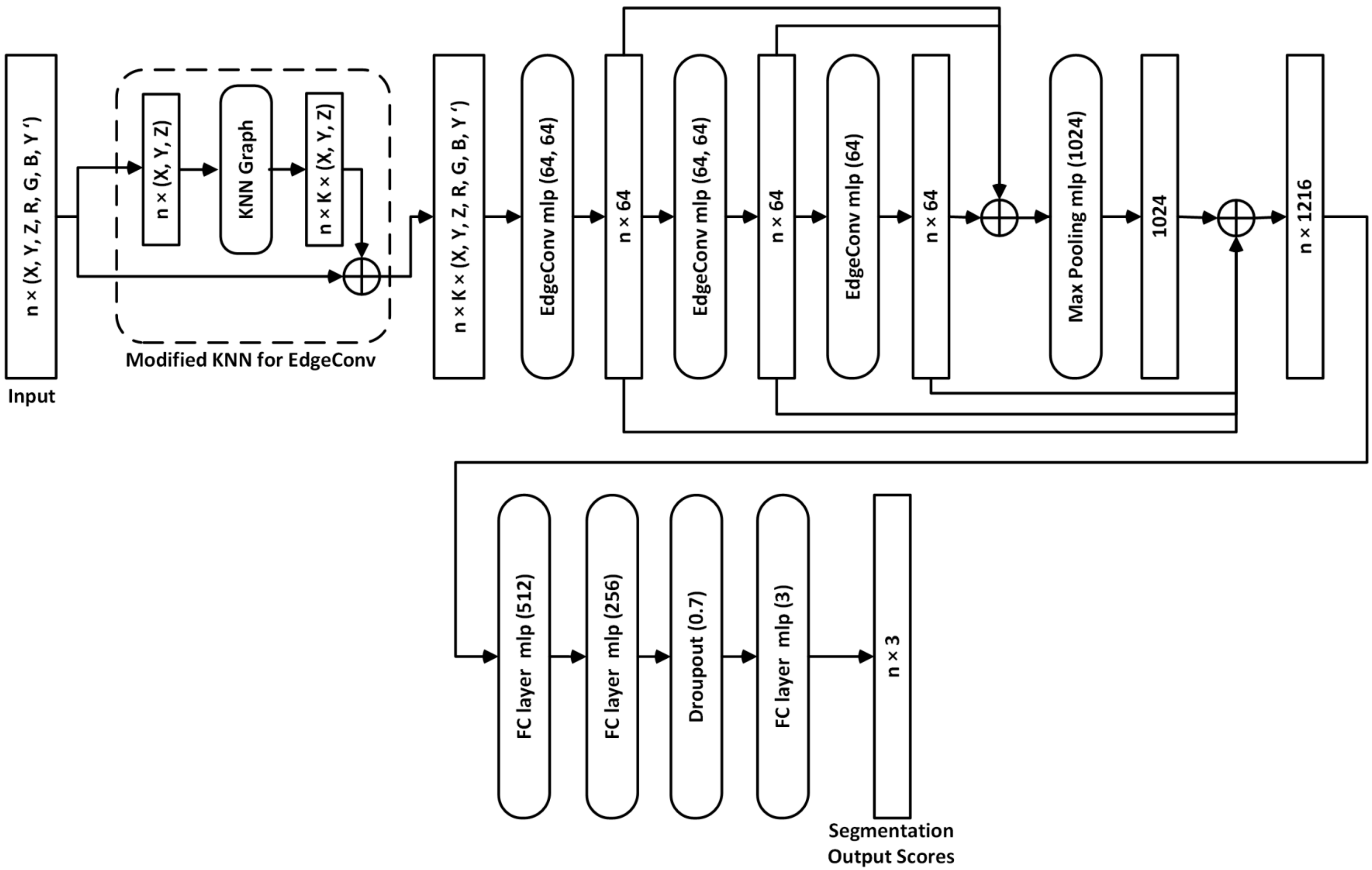


Figure 6-2. Architecture of adapted DGCNN (adapted from [27])

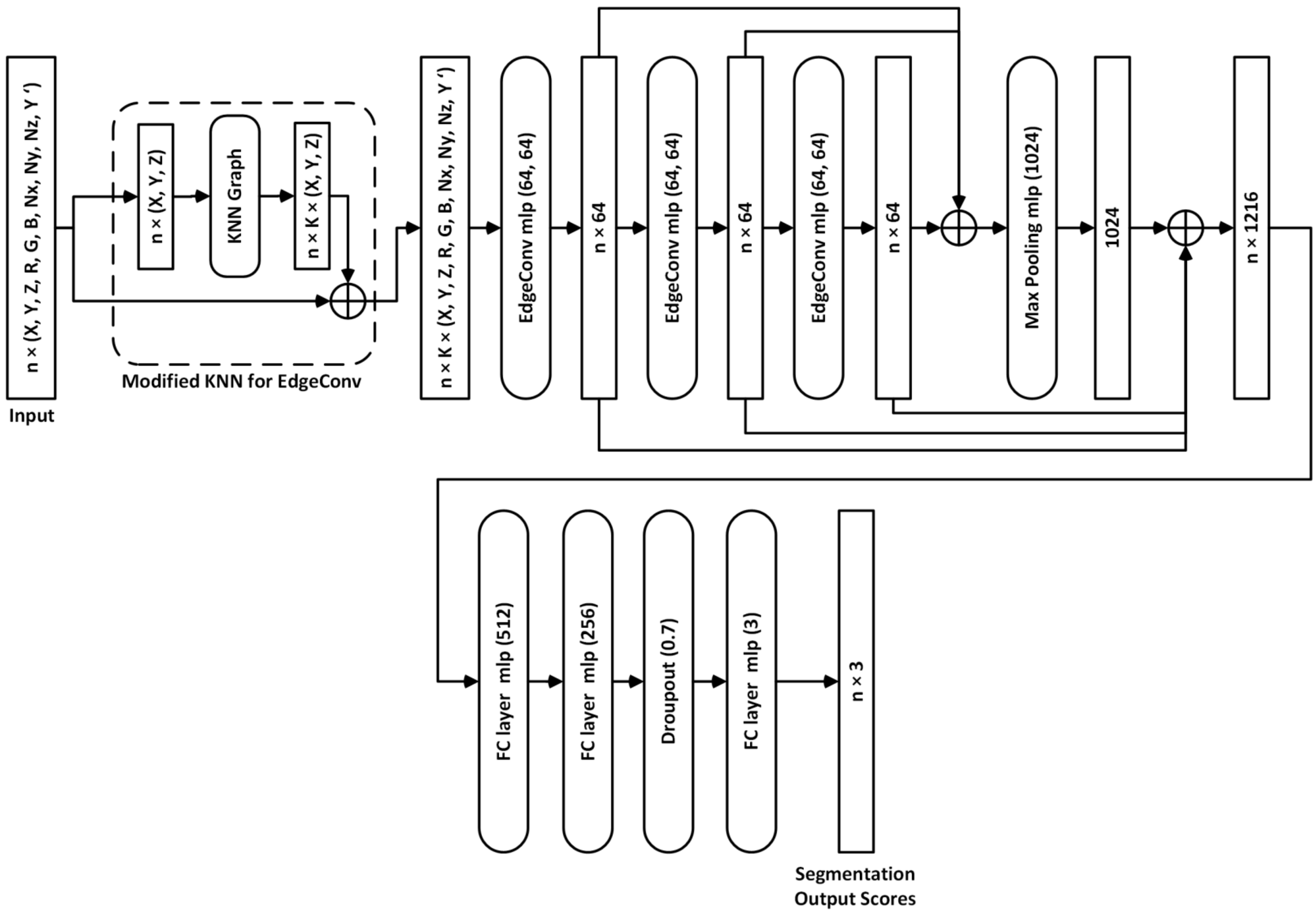


Figure 6-3. Architecture of NVE-DGCNN (adapted from [27])

Table 6-1. DGCNN, adapted DGCNN and NVE-DGCNN hyperparameters

Parameter	DGCNN	Adapted DGCNN	NVE-DGCNN
Classes	Building indoor objects (11 classes)	Cracks, spalls, no-defect	Cracks, spalls, no-defect
Input Variables	*X, Y, Z, R, G, B, X', Y', Z'*	*X, Y, Z, R, G, B, Y'*	*X, Y, Z, R, G, B, Nx, Ny, Nz,Y'*
Number of points in each block	4,096 *pts*	8,192 *pts*	8,192 *pts*
Size of blocks (m)	1 m × 1 m × Z_{max}	0.4 m × Y_{max} × 0.4 m	0.4 m × Y_{max} × 0.4 m
Stride	N.A.	0%	25%
Convolving direction	*XY* surface	*XZ* surface	*XZ* surface
Number of nearest neighbors (k)	20	20	20
Number of epochs	100	50	50
Optimizer	Adam	Adam	Adam
Data augmentation	Random rotation	Flipping horizontally	Flipping horizontally
Weight vector for loss function	Softmax cross entropy	Weighted Softmax cross entropy	Weighted Softmax cross entropy
Learning rate	1e-3 (decays exponentially to a minimum of 1e-5)		

6.2.2 Steps of Applying the Modified CNN Models

There are six main steps in applying the modified CNN models: (1) data collection, (2) manual annotation, (3) data pre-processing, (4) training and evaluation, (5) testing, (6) sensitivity analysis. Figure 6-4 shows the proposed method for concrete surface defect semantic segmentation using adapted DGCNN and NVE-DGCNN in detail. However the normal vector estimation and sensitivity analysis steps are only implemented for NVE-DGCNN.

6.2.2.1 Data Collection

The geometric features of defects, particularly the depth, play a significant role in extracting important features and having accurate results. Currently, most of available online datasets for concrete surface defects are image-based. Therefore, data collection is an important step and has to be done accurately. The scanner position and the scanning parameters, such as resolution, quality, Field of View (FOV), and the number of scanned points, are the factors that can affect the visibility of defects in the collected point cloud data.

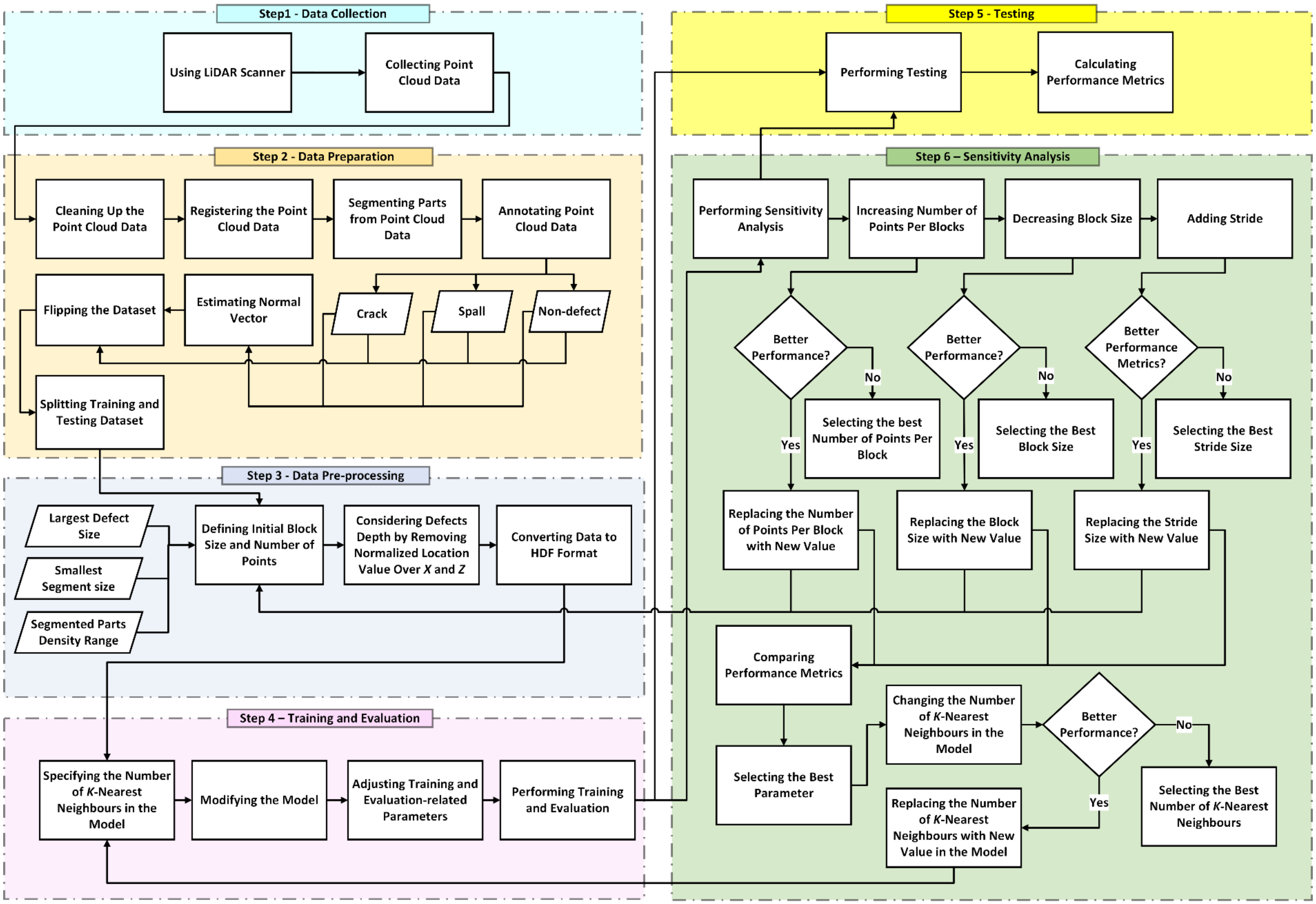


Figure 6-4. Proposed method for concrete surface defect semantic segmentation using adapted DGCNN and NVE-DGCNN

6.2.2.2 Data preparation

After data collection, irrelevant points of the point cloud data in each scan need to be eliminated, and all the scans will be prepared for registration. Then, different areas are cut from the registered point cloud data, and different parts are segmented in each area. The selected parts need to be manually annotated based on the types of targeted surface defects. As shown in Figure 6-4, in this research, two main types of surface defects, which are cracks and spalling, are considered. Each part of the dataset is annotated into three categories of crack, spalling, and non-defect. Furthermore, in this research, to enlarge the size of the dataset, the augmentation method of flipping the point cloud is used, where the annotated parts are flipped with respect to the YZ plane.

6.2.2.3 Data pre-processing

This research considers two approaches for preparing the dataset and feeding the MLP classifier of the CNN network. In the first approach, the original dataset files are converted into data label files, which are *2D* matrices with *XYZRGBL* in each line. Then, each part is split into blocks, and for each block, normalized location values on the *Y* surface are added [27]. Each point is represented as a 7-dimensional vector of *XYZ*, *RGB*, and *Y'*. These features were used for the training process of the adapted PointNet++ and the adapted DGCNN. In the second approach, an additional hand-crafted point feature, which is the normal vector (*Nx, Ny,* and *Nz*) is added to feed a MLP classifier of NVE-DGCNN. Previous works, such as Hyeon et al. [332] investigated the effect of considering normal vectors as an additional feature for semantic segmentation of building elements and stated that considering the normal vectors in CNN networks can improve the model's performance.

Then, the sizes of blocks are defined based on the sizes of the structural defects (smallest segment or the largest defect) in the dataset. In this research, the smallest dimension of segments is 46 *cm*, and the largest defect size is 60 *cm*. Hence, the selected block size in the data pre-processing step is assumed to be at least 40 *cm* × 40 *cm* on the *XZ* surface, with the depth of the defects as the third dimension, which is equal to the depth of the deepest defect in each segment. As shown in Figure 6-4, in this step, the wrapped and normalized points inside the blocks are converted to Hierarchical Data Format (HDF) [333], and HDF5 files are used for the training process in the next step.

6.2.2.4 Training and evaluation

As discussed in Section 6.2.1, a series of three EdgeConv layers followed by three fully-connected layers are included in the segmentation model of DGCNN, and the number of the *K*-nearest neighbors of a point for EdgeConv layers is specified for the input of *n* points in the model. The number of the *K*-nearest neighbors of a point for EdgeConv layers is set equal to 20 following the suggested value by Wang et al. [27].

6.2.2.5 Testing

To validate the model accuracy, the unseen parts of the dataset, which are not used in the training and evaluation steps, are used for the testing step. The confusion matrix is used to describe the model's performance using the equations presented in Table 6-2. In this research, the term *overall accuracy* refers to the percentage of correct predictions for the test data. Furthermore, the *recall* is assumed to be more relevant than precision as the process of concrete surface inspection aims to minimize the chance of missing actual defect points, which can be achieved by minimizing the *False Negative* prediction of the model. *Recall* is more important when the outlay of missing the correct prediction is more significant than the outlay of the wrong prediction. *Precision* is more important when the outlay of the wrong prediction is more significant than the outlay of missing the correct prediction. *F1 score*, which is the harmonic mean of *precision* and *recall*, is used when the model's resistance to the outliers is more critical. *Intersection over Union (IoU)* is used when computing the area of overlap between the bounding boxes of prediction and ground-truth is more important [328].

Table 6-2. Model performance metrics

Performance metrics	Equation
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F1 score	$2 \times \frac{Precision \times Recall}{Precision + Recall}$
Intersection over Union (IoU)	$\frac{TP}{FP+TP+FN}$
Overall accuracy	$\frac{TP_{Crack} + TP_{Spall} + TP_{Non-defect}}{All\ perdiction\ of\ the\ points}$

Note: *TP* refers to true positives, *FP* refers to false positives, and *FN* refers to false negatives

6.2.2.6 Sensitivity analysis

As shown in Figure 6-4, sensitivity analysis was done in this research to investigate the effect of different input variables on the network's performance. The dataset's density depends on the block size and the number of points per block. Blocks with densities more than the pre-set value result in up-sampling. On the other hand, densities less than the pre-set value for each block result in down-sampling. Therefore, the distribution of segments based on their densities should be considered. The hyperparameters related to the dataset, which are considered in this research, are: (1) number of points, (2) size of the block, and (3) size of stride. The number of points per block is selected based on the density range and the pre-set default block size (*40 cm × 40 cm*). The number of points per block of 8,192 is selected as the first acceptable value. Then the density of points in each block is increased by raising the number of points. The density of most segments of the prepared dataset is between 9,049 and 329,369 pts/m^2. Figure 6-5 shows the list of number of points per block, block sizes, and their densities. Increasing the number of points to more than 12,288 could not be applied due to computation resource limitations (i.e., RAM). Moreover, the density of the block with the size of *20 cm × 20 cm* and number of points of 14,336 is more than 329,369 pts/m^2, which is out of the range for this research. Then the block size is decreased for the same pre-set number of points to increase the density of points in each block. Stride can be added to shift the number of points per block over the input matrix. Considering the overlapped number of points per block may improve the results. Therefore, the effect of stride on the results is considered in the sensitivity analysis.

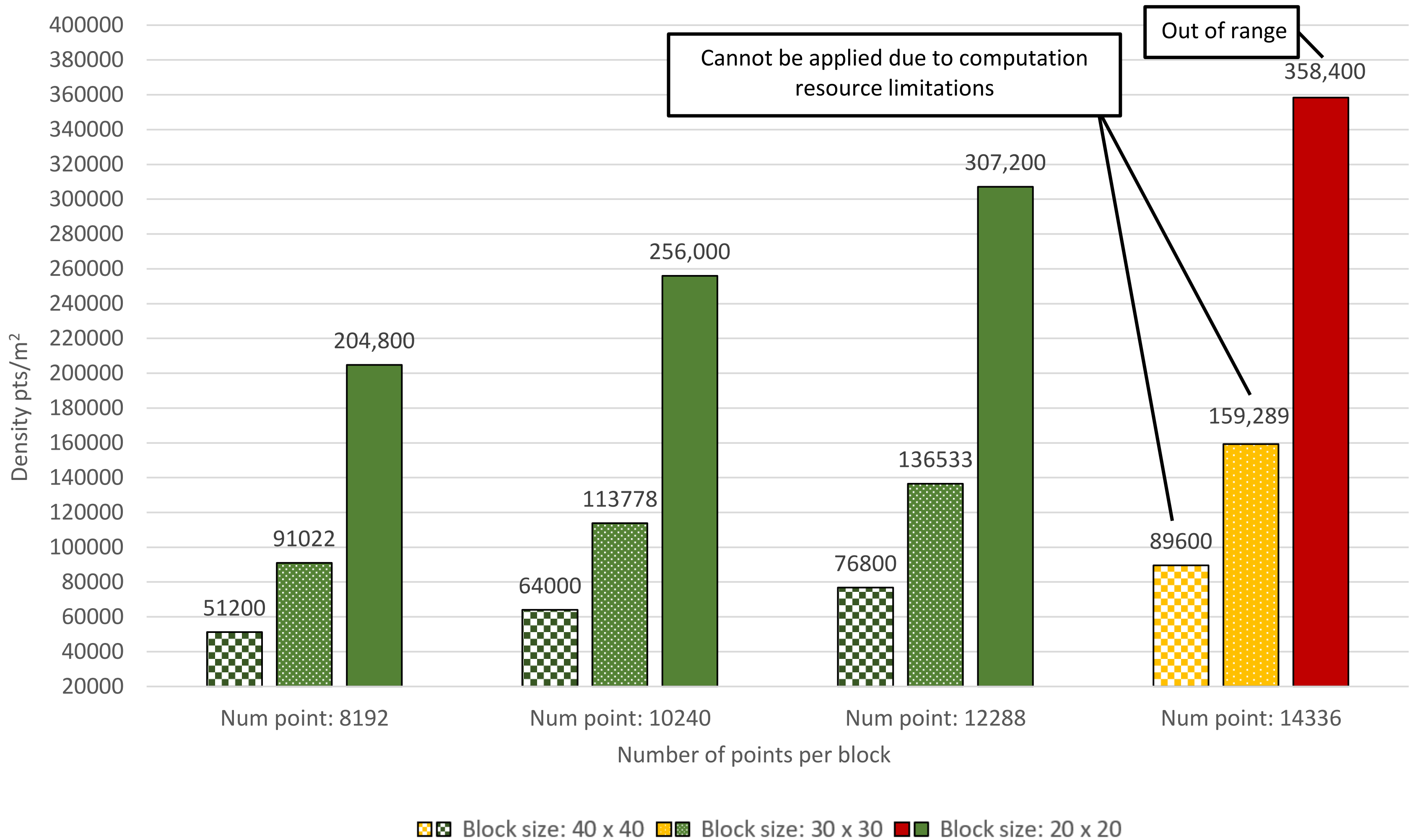


Figure 6-5. List of number of points per block, block sizes, and their densities

6.2.3 As-Inspected Modeling

Figure 6-6 shows the workflow of the as-inspected modelling process. The following main steps are used to semi-automate the process of as-inspected modeling:

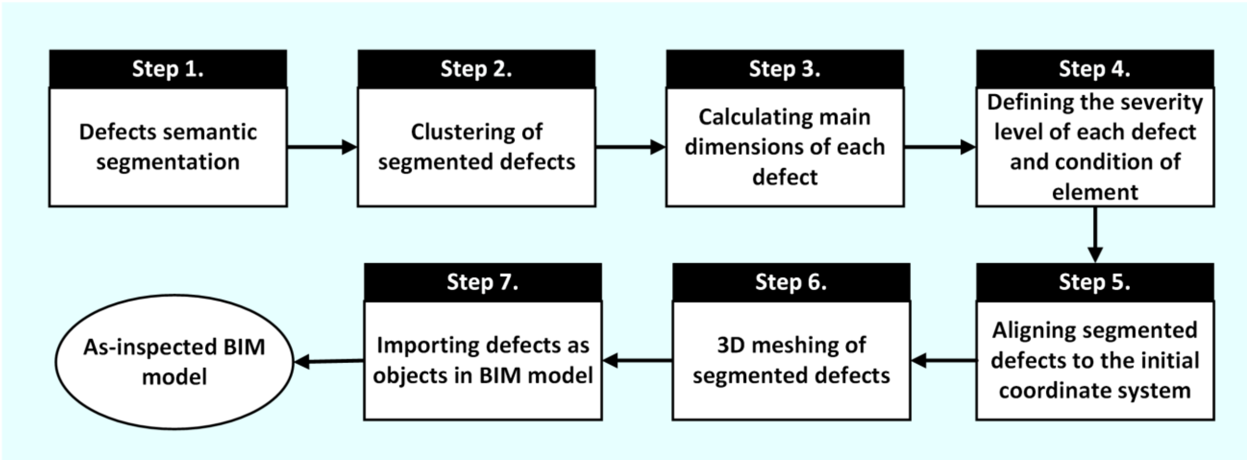


Figure 6-6. Workflow of as-inspected modeling

(1) Defects semantic segmentation: The as-built bridge model is assumed to be available. The semantic segmentation results of NVE-DGCNN for crack and spall defects are used for the as-inspected modeling purpose.

(2) Clustering of segmented defects: Density-based spatial clustering, which is proposed by Liu et al. [334], is used in this step to create spatial proximity relationships to cluster the defects [335].

(3) Calculating main dimensions of each defect: An algorithm is used to find the Minimum Bounding Box (MBB) for each cluster [336, 337], and geometrical information of cracks and spalls including the defects' length, width, and depth are calculated based on the Euclidean distance between the corners of the associated bounding box.

(4) Defining the severity level of each defect and condition of element: The severity level of each defect is defined based on Table 6-3. This table shows the severity levels of crack and spall defects used in OCSD (Section 4.4), which are based on the OSIM [41]. The element condition is defined based on the severity level.

Table 6-3 Severity of crack and spall defects in OCSD based on OSIM [41]

Surface defect types	Severity (all dimensions in mm)			
Crack	Hairline (width < 0.1)	Narrow (0.1 ≤ width ≤ 0.3)	Medium (0.3 < width ≤ 1.0)	Wide (1.0 < width)
Spall	Light (Any direction < 150 or depth < 25)	Medium (150 ≤ Any direction ≤ 300 or 25 ≤ depth ≤ 50)	Severe (300 < Any direction ≤ 600 or 50 < depth ≤ 100)	Very severe (600 < Any direction or 100 < depth)

(5) Aligning segmented defects to the initial coordinate system: The segmented defects are aligned to the initial coordinate system by considering each point's normal vector using the Normal Iterative Closest Point (NICP) algorithm [338], which is proposed by Serafin and Grisetti [339]. Although some tools such as CloudCompare software can be used to align the point clouds manually by picking at least four pairs of reference points in the source and target point clouds, this process is time consuming.

(6) 3D meshing of segmented defects: In this step, the clusters of detected defects are converted into the 3D mesh product using 3DReshaper software to have an accurate model of the defect objects.

(7) Importing defects as objects in BIM model: In this step, a Dynamo script using Mesh Toolkit [340] is utilized to import the 3D mesh defects into the BIM model (Appendix A). Although the case study that will be explained in Section 6.5 is about bridge inspection, where the structure model is refer to as Bridge Information Model (BrIM), the term BIM will be used in the rest of this chapter.

6.3 Data Collection and Dataset Pre-processing

This study used point cloud datasets from four reinforced concrete bridges in Montreal, scanned using a FARO Focus3D scanner [341]. The specifications of this scanner are presented in Table 6-4. The images of the scanned bridges are shown in Figure 6-7. Table 6-5 shows the scanning parameters. CloudCompare software [342] is used to register and eliminate the irrelevant points of the point cloud data. The scanned data's quality depends on the two main parameters of density and accuracy [324]. The number of points in a specific area represents the point cloud density [343]. The resolution parameter represents the number of points that the scanner uses to measure the environment during the scanning process (between *1* (710.7 million points) - *1/32* (11.1 million points)), and the quality represents the number of times the scanner hits the same point during the scanning (between–*1x* - *8x*) [344]. Therefore, the distance between two points next to each other depends on the resolution parameter.

The scanning process in this step is affected by several factors, such as the battery capacity and performance limitations especially in severe weather conditions, scanning time, and traffic constraints. For this reason, different settings, including different numbers of stations, were used to scan each of the bridges. In some scans, the FOV was reduced to avoid scanning irrelevant objects (e.g. moving vehicles).

Table 6-4. FARO Focus3D LiDAR scanner specifications [354]

LiDAR	Points per Second	Field of View		Angular Resolution	Accuracy	Measurement Range
		Vertical	Horizontal			
FARO Focus 3D	976,000	305°	360°	0.009°	±2 mm	1.5 m – 120 m

(a) Bridge 1: Guy Street

(b) Bridge 2: Lucian L'Allier Street

(c) Bridge 3: Avenue Atwater

(d) Bridge 4: Chemin Macdonald

Figure 6-7. Scanned bridges

Table 6-5. Scanning parameters of four scanned bridges in Montréal

Scans		Number of Stations	Resolution	Quality	Horizontal FoV	Vertical FoV	Number of Points (Mpts)
Bridge 1	Scan 1	8	1/4	6x	23° to 259°	-42.5° to 71°	25.5
	Scan 2	4	1/4	6x	23° to 259 °	-42.5° to 71°	25.5
Bridge 2	Scan 3	6	1/1	2x	0° to 360°	-60° to 90°	710.7
Bridge 3	Scan 5	4	1/2	4x	0° to 360°	-45° to 71°	134.5
Bridge 4	Scan 6	2	1/2	4x	0° to 360°	-60° to 90°	177.7

The prepared dataset includes 102 selected segmented parts from the scanned bridges. The number of annotated cracks in the selected parts is 595, and the number of annotated spalls is 773. The annotation process is done manually in CloudCompare software using the following rules based on experience: (1) a specific range of 150,000 pts to 400,000 pts is considered for the number of points of each selected part; (2) the scanned surfaces are classified into rectangular parts because of the box shape of the blocks in the model; and (3) the part size should consider the higher density of points in some parts and it should not contain more than the maximum defined number of points, which is 400,000 pts. Furthermore, the handcrafted normal vector feature (*Nx, Ny,* and *Nz*) is computed in CloudCompare software to prepare the NVE-DGCNN dataset. The annotated datasets are split into five areas. Area 1 to 3 are used for training, Area 4 is used for evaluation, and Area 5 is dedicated to testing. The total number of segmented parts after adding the flipped data is 204 parts. The statistical information of the dataset, including the flipped data, is given in Table 6-6. Figure 6-8 shows the structure of preparing the dataset, including a sample of an annotated segment, where yellow, red, and blue represent the non-defect, crack, and spalling, respectively.

Table 6-6. The statistics of the prepared dataset

Dataset		Number of segmented parts	Number of points	Defects				Non-defects
				Crack		Spalling		
				Number of cracks	Number of points	Number of spalls	Number of points	Number of points
Training (59.5%)	Area 1	32	10,418,902	264	104,256	226	715,768	9,598,878
	Area 2	44	11,003,768	334	112,436	266	282,822	10,608,510
	Area 3	42	10,651,316	160	67,714	356	744,356	9,839,246
Evaluation (19.6%)	Area 4	44	10,552,584	192	80,454	328	762,156	9,709,974
Testing (20.9%)	Area 5	42	11,257,240	240	128,538	370	1,365,228	9,763,474
Total		204	53,883,810	1,190	493398	1,546	3,870,330	49,520,082

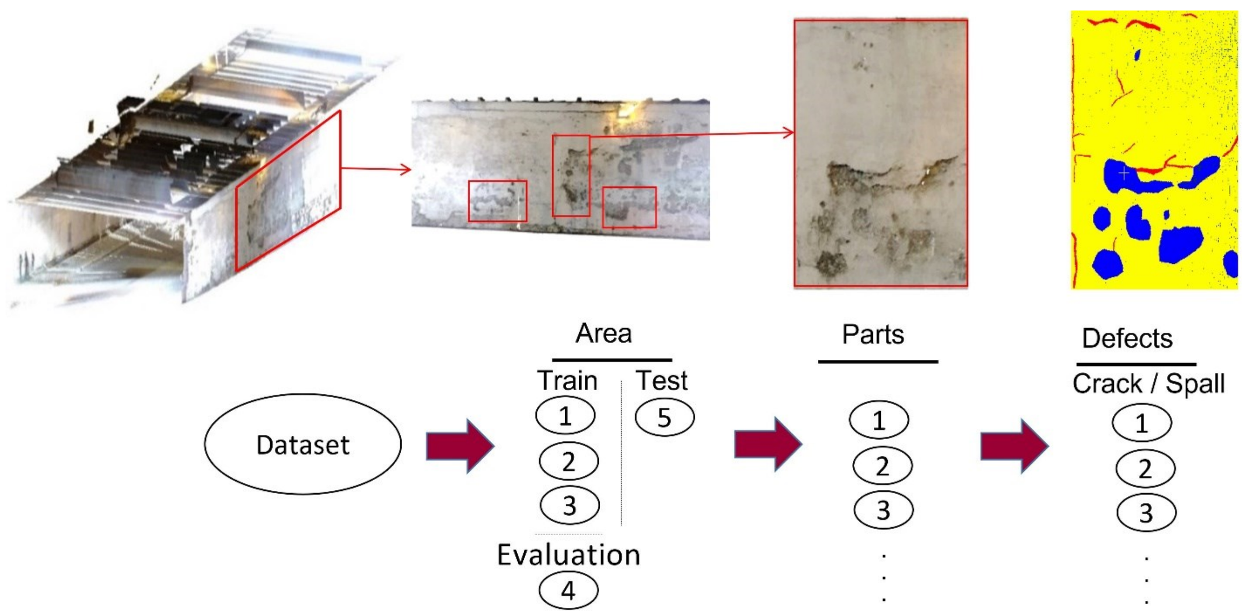


Figure 6-8. The structure of the dataset

6.4 Implementation of the Modified CNN Models

6.4.1 Adapted PointNet++

The adapted PointNet++ was studied to identify the effect of the hyperparameters mainly the number of points and strides. In this sensitivity analysis, nine cases are defined and validated using three numbers of points (8,192, 10,240, 12,288) and three stride sizes (0%, 25%, 50%). As discussed in Section 6.2.2.6, the values for these two parameters in the sensitivity analysis are related to the values of the raw dataset. The convolving direction is set to *XZ* surface, and the initial block size is set to $40 \times Y_{max} \times 40\ cm$. Training and testing were performed on a cloud computing platform using 2 NVIDIA P100 Pascal GPUs, 24 GB RAM per GPU, and a 32-core CPU. The number of epochs is set to 50. The initial learning rate is 0.001 and the learning rate decays exponentially to a minimum of 1e-5.

Three cases were with 0% (40 cm) stride (A to C), which means there is no overlapping in the training datasets and convolutions. In all cases with the same block size and stride, the number of points increased from 8,192 to 12,288. In three cases (D to F), the stride was decreased by 25% (30 cm), and in the remaining cases (G to H), the stride considered was 50% (20 cm).

Table 6-7 shows the output results of calculated accuracies and mean losses for training and evaluation, and Table 6-8 shows the testing results. Figure 6-9 shows recall test results for PointNet++ cases. Figure 6-10 shows the effect of stride based on the average recall value. As shown in this figure, both crack and spalling recalls were improved by decreasing the stride value. Decreasing the stride to 50% improved the crack recall by almost 2%. Figure 6-11 shows the effect of the number of points based on the average recall value. As shown in this figure, increasing the number of points will increase the recall of defects in the adapted PointNet++ network.

Table 6-7. Training and evaluation results for adapted PointNet++

Case	Number of points in each block	Block size (cm)	Stride (cm)	Training		Evaluation		Training time
				Mean loss	Overall Accuracy	Mean loss	Overall Accuracy	
A	8,192	40×40	40 (0%)	0.120	98.4%	0.135	96.2%	4h 42m
B	10,240	40×40	40 (0%)	0.087	98.7%	0.102	96.8%	5h 42m
C	12,288	40×40	40 (0%)	0.094	98.7%	0.107	96.7%	6h 36m
D	8,192	40×40	30 (25%)	0.089	98.7%	0.134	96.3%	7h 32m
E	10,240	40×40	30 (25%)	0.079	98.8%	0.114	96.7%	8h 57m
F	12,288	40×40	30 (25%)	0.074	98.9%	0.115	96.7%	10h 34m
G	8,192	40×40	20 (50%)	0.108	98.5%	0.127	96.5%	14h 33m
H	10,240	40×40	20 (50%)	0.071	99.0%	0.100	97.1%	17h 36m
I	12,288	40×40	20 (50%)	0.086	98.7%	0.094	97.0%	20h 35m

Table 6-8. Testing results for adapted PointNet++

Case	Cracks				Spalling				Non-defect			
	Precision	Recall	F1 score	IOU	Precision	Recall	F1 score	IOU	Precision	Recall	F1 score	IOU
A	27.6	21.9	24.4	13.9	66.9	69.6	68.3	51.8	95.3	95.0	95.2	90.8
B	38.4	45.2	41.6	26.2	67.6	77.5	72.2	56.5	96.5	94.4	95.5	91.3
C	39.7	48.1	43.5	27.8	68.0	80.4	73.7	58.4	96.9	94.4	95.6	91.6
D	29.5	26.5	27.9	16.2	75.8	75.7	75.7	60.9	96.0	96.2	96.1	92.5
E	42.9	43.5	43.2	27.5	74.9	79.6	77.2	62.8	96.6	95.8	96.2	92.7
F	41.7	49.3	45.2	29.2	76.6	75.6	76.1	61.4	96.2	96.1	96.1	92.6
G	22.2	24.4	23.3	13.2	70.2	79.4	74.5	59.4	96.3	94.3	95.3	91.0
H	46.6	50.1	48.3	31.8	80.0	77.7	78.9	65.1	96.3	96.6	96.5	93.2
I	42.4	50.1	45.9	29.8	75.2	80.6	77.8	63.6	96.9	95.7	96.3	92.8

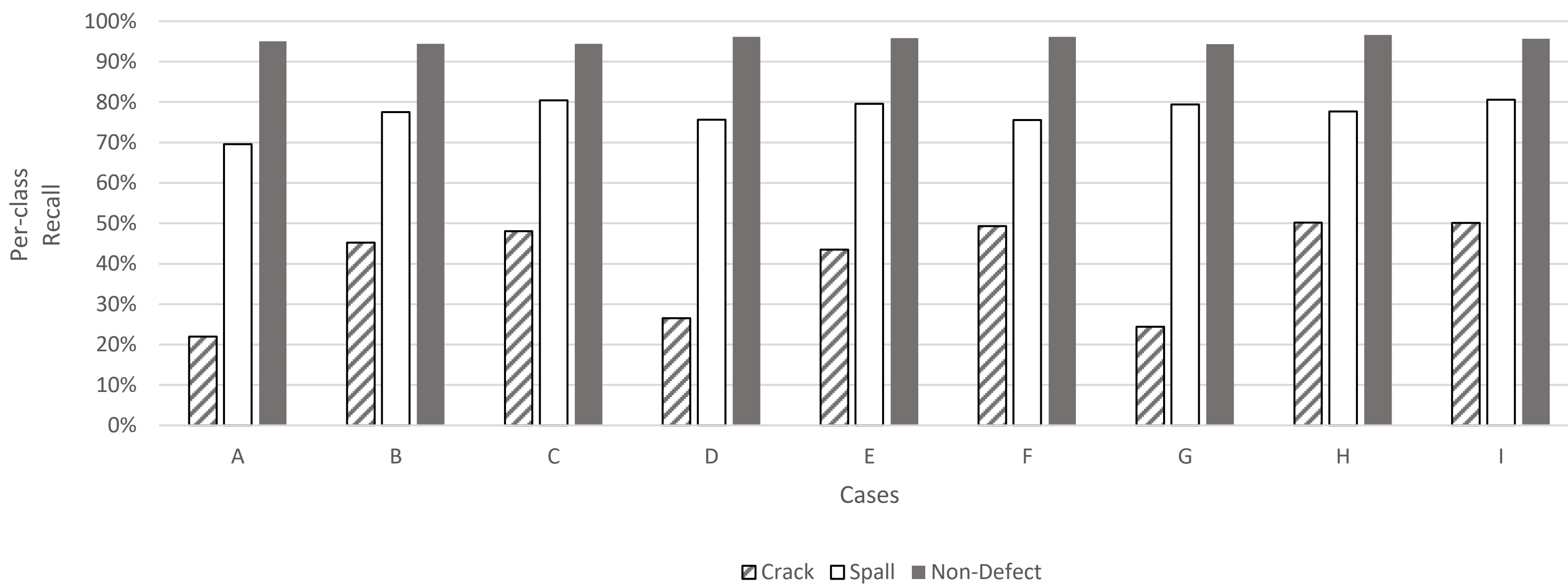


Figure 6-9. Recall testing results for adapted PointNet++

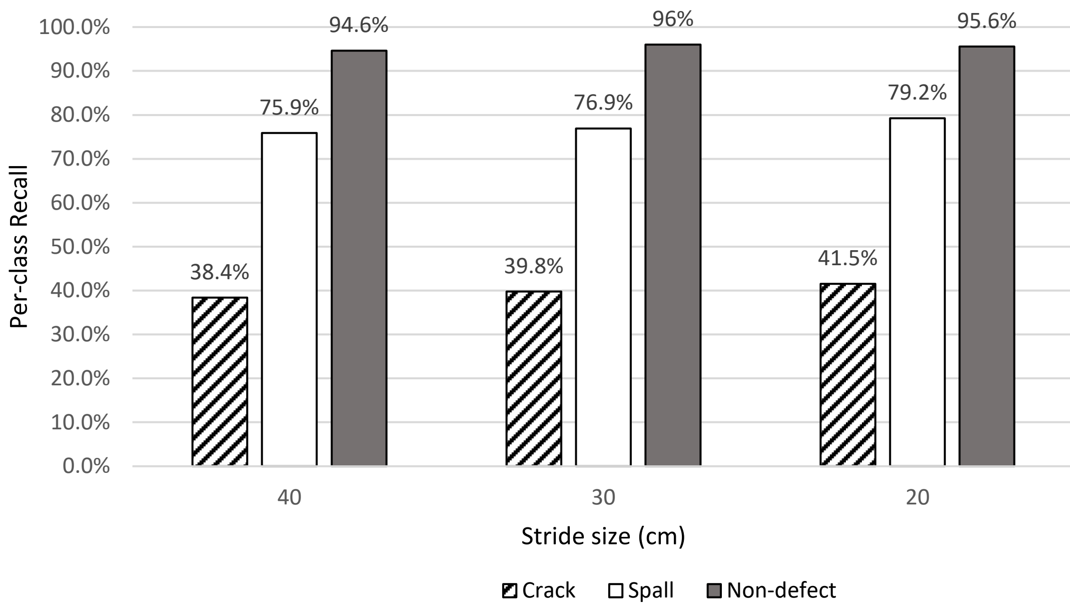


Figure 6-10. Effect of stride based on the average recall value for adapted PointNet++

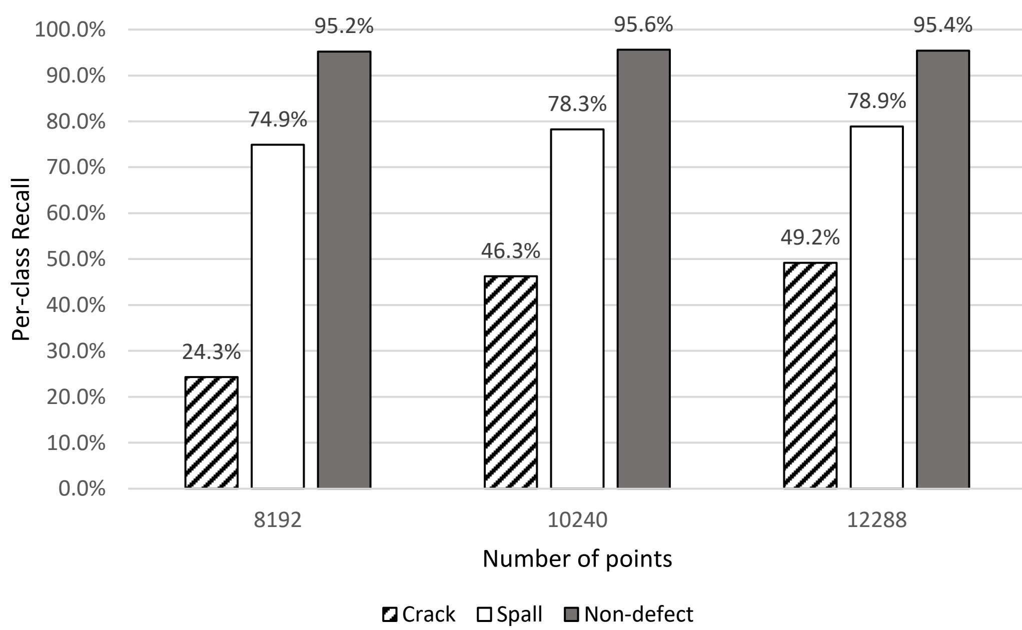


Figure 6-11. Effect of the number of points based on the average recall value for adapted PointNet++

6.4.2 Adapted DGCNN

A Compute Canada cluster is used to implement this case study using 4 NVIDIA V100 Volta GPUs with 32 GB RAM per GPU, 24 CPUs, and 123 GB of memory. The available memory on the hardware constraints the possible volume of computation. The number of epochs is set to 50. The initial learning rate is 0.001 and it decays exponentially to a minimum of 1e-5. The percentage of the defect points is almost 14% of the whole point cloud and is much less than the no-defect points (86%). Therefore as discussed in Section 6.2.1, a weighted softmax cross entropy loss function is defined in the model based on the points distribution of the classes (crack, spalls, and non-defect), which is [0.714, 0.271, 0.016]. By using a weighted loss function, the effective weight of points of each class in the correcting process of backpropagation can be adjusted.

In the first step, to implement the adapted DGCNN, three cases are defined with different numbers of input points of 8,192, 10,240, and 12,288 (Case A1 to C1), which are sampled for each block during the training process. The training and evaluation results, including the overall accuracy and mean loss of Cases A1 to C1, are presented in Table 6-9. Precision, recall, F1 score, IoU, and overall accuracy are calculated to evaluate the semantic segmentation results for Cases A1 to C1. The test results of the adapted DGCNN (Table 6-10) show the detecting recall for cracks and spalls for Case C1 (12,288 points) are 58.67% and 87.40%, respectively. Figure 6-12 shows recall test results for adapted DGCNN. Increasing the number of points from 8,192 to 12,288 improved the crack semantic segmentation recall from 55.20% to 58.67%. However, this increase resulted in decreasing the spall recall from 89.77% to 87.40%, and non-defect recall from 97.17% to 96.64%. This is because increasing the number of points sometimes can cause overfitting [345].

Table 6-9. Training and evaluation results for adapted DGCNN

Case	Number of sampled points for each block	Block size (cm)	Training		Evaluation		Training time
			Mean loss	Overall accuracy (%)	Mean loss	Overall accuracy (%)	
A1	8,192	40×40	0.0022	97.54	0.0081	97.50	13h 44m
B1	10,240	40×40	0.0024	97.39	0.0090	97.65	16h 35m
C1	12,288	40×40	0.0030	97.04	0.0082	96.88	20h 18m

Table 6-10. Testing results for adapted DGCNN (%)

Case	Overall accuracy	Crack				Spalling				Non-defect			
		Precision	Recall	F1 score	IOU	Precision	Recall	F1 score	IOU	Precision	Recall	F1	IOU
A1	95.94	69.98	55.20	61.76	44.68	79.30	89.77	84.2	72.72	98.54	97.17	97.85	95.79
B1	95.59	68.95	55.31	61.38	44.28	77.47	89.41	83.0	71.0	98.48	96.82	97.64	95.39
C1	95.24	49.73	58.67	53.83	36.83	77.00	87.40	81.9	69.3	98.48	96.64	97.55	95.22

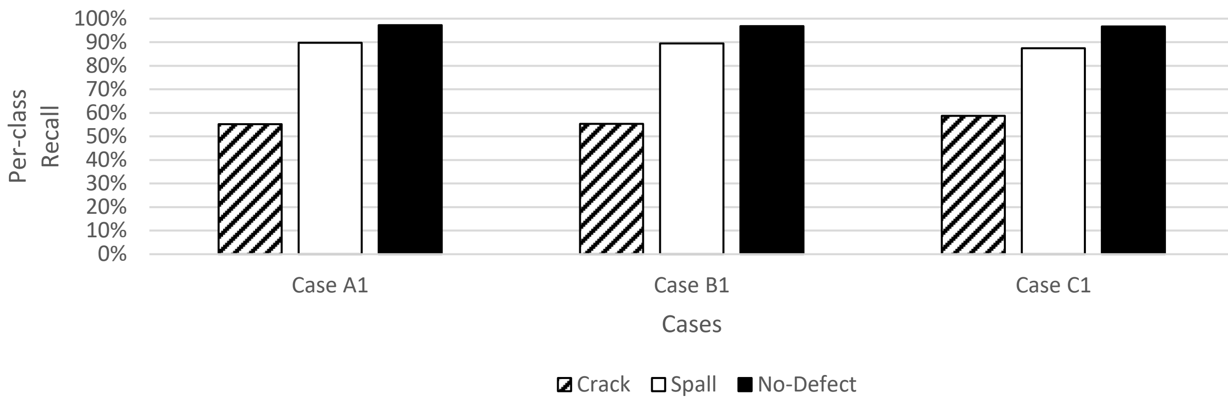


Figure 6-12. Recall test results for adapted DGCNN

As the depths of segmented parts are different, and the learning process depends on the maximum depth of the part's defects, the recall result of the tests is categorized based on the depth of segmented parts used in the test as shown in Table 6-11. As shown in this table, deeper parts can increase recall up to 80.04% for crack and 93.33% for spall.

Table 6-11. Defect semantic segmentation recall based on the depth of defects for adapted DGCNN (%)

Case	Number of sampled points for each block	Depth (cm)					
		D≤3		3<D<7		7≤D	
		Crack	Spall	Crack	Spall	Crack	Spall
A1	8,192	35.22	90.78	44.87	87.59	76.91	92.68
B1	10,240	36.65	88.48	42.52	86.99	79.00	93.22
C1	12,288	39.22	81.91	48.07	84.39	80.04	93.33

As shown in Table 6-12, the comparison of the adapted PointNet++ and adapted DGCNN shows that for the same number of points of 8,192, 10240, and 12288, and block size of 40 × 40 *cm* with 0% stride, DGCNN achieved a higher recall.

Table 6-12. Comparison of the recall results of adapted PointNet++ and adapted DGCNN (%)

Case	Number of sampled points for each block	Block size (cm)	Adapted PointNet++		Adapted DGCNN	
			Crack	Spall	Crack	Spall
A1	8,192	40×40	21.9	69.6	55.20	89.77
B1	10,240	40×40	45.2	77.5	55.31	89.41
C1	12,288	40×40	48.1	80.4	58.67	87.40

Furthermore, as explained in Section 6.2.1, a test was performed based on the original DGCNN to examine the effect of using the normalized *X, Y, and Z* values for the KNN. The test was performed for the number of points of 8,192, and block size of 40 × 40 *cm* with no stride. As shown in Table 6-13, the comparison of the DGCNN with original KNN and adapted DGCNN with modified KNN shows that the models' performance declined significantly by considering the normalized location values for KNN.

Table 6-13. Comparison of the recall results of DGCNN with original KNN and Adapted DGCNN (%)

Case	Number of sampled points for each block	Block size (cm)	DGCNN with original KNN		Adapted DGCNN with modified KNN	
			Crack	Spall	Crack	Spall
A1	8,192	40×40	41.17	41.27	55.20	89.77

Three samples of the test results for Case C1 from the adapted DGCNN are shown in Figure 6-13.

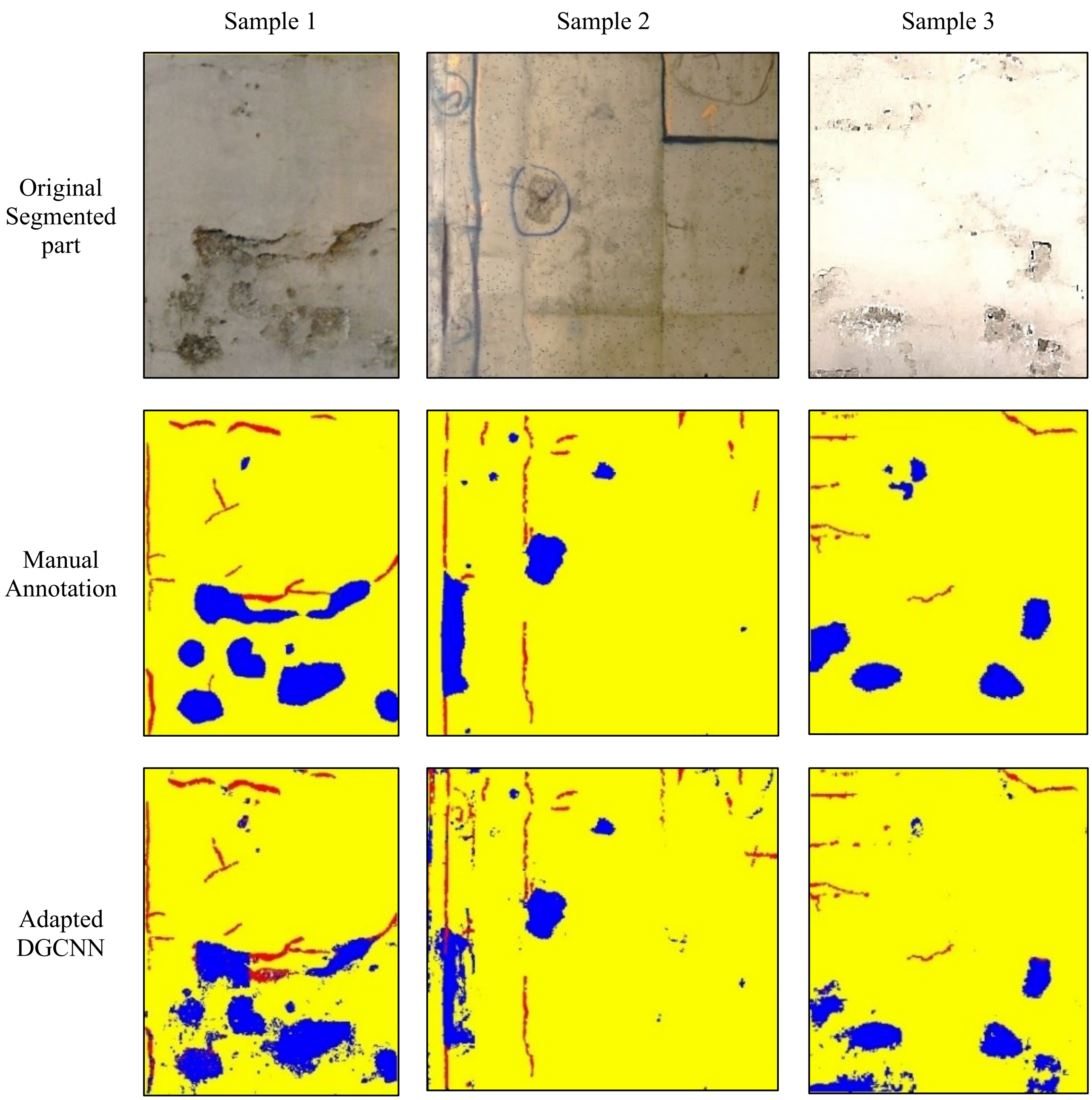


Figure 6-13. Test results from three samples of adapted DGCNN (Case C1)

6.4.3 NVE-DGCNN

This part of the implementation defines fifteen cases based on the NVE-DGCNN to identify the effect of the hyperparameters on the performance.

The training and evaluation results, including the overall accuracy and mean loss of Cases A2 to O2 are presented in Table 6-14. Precision, recall, F1 score, IoU, and overall accuracy are calculated to evaluate the testing of the cases as shown in Table 6-15. The results show that decreasing the block size to 20 × 20 *cm* or adding the 50% stride for the block with the size of 40 × 40 *cm* will decrease the crack and spall recall. The best test results of the NVE-DGCNN show the recall for cracks and spalls were for Case J2 (8,192 points and 25% stride), which are 98.56% and 96.50%, respectively. Figure 6-14 shows recall test results for NVE-DGCNN.

Table 6-14. Training and evaluation results for NVE-DGCNN

Case	Number of points per block	Block size (cm)	Stride (cm)	Training		Evaluation		Training time
				Mean loss	Overall accuracy (%)	Mean loss	Overall accuracy (%)	
A2	8,192	40×40	40 (0%)	0.0012	98.60	0.0097	98.58	14h 46m
B2	10,240			0.0008	99.12	0.0077	98.36	18h 35m
C2	12,288			0.0009	98.76	0.0070	98.62	20h 26m
D2	8,192	30×30	30 (0%)	0.0013	98.74	0.0092	95.16	21h 01m
E2	10,240			0.0007	99.13	0.0104	95.50	27h 05m
F2	12,288			0.0013	98.67	0.0069	95.26	31h 46m
G2	8,192	20×20	20 (0%)	0.0011	98.97	0.0090	94.84	41h 44m
H2	10,240			0.0008	99.14	0.0117	98.14	52h 50m
I2	12,288			0.0016	98.46	0.0148	97.58	65h 14m
J2	8,192	40×40	30 (25%)	0.0004	99.53	0.0064	97.47	19h 50m
K2	10,240			0.0006	99.34	0.0077	98.05	24h 35m
L2	12,288			0.0006	99.29	0.0087	96.63	29h 4m
M2	8,192	40×40	20 (50%)	0.0003	99.62	0.0084	97.88	36h 41m
N2	10,240			0.0003	99.62	0.0091	97.56	44h 55m
O2	12,288			0.0004	99.59	0.0080	97.28	55h 31m

Table 6-15. Testing results for NVE-DGCNN (%)

Case	Number of points per block	Block Size (cm)	Stride (cm)	Overall accuracy	Crack				Spalling				Non-defect			
					Precision	Recall	F1 score	IOU	Precision	Recall	F1 score	IOU	Precision	Recall	F1	IOU
A2	8,192	40×40	40 (0%)	98.28	92.71	96.20	94.42	89.43	89.97	95.30	92.56	86.15	99.44	98.67	99.06	98.13
B2	10,240			98.34	92.97	95.49	94.22	89.06	90.95	94.60	92.74	86.46	99.36	98.83	99.09	98.20
C2	12,288			98.13	93.75	95.32	94.53	89.63	88.57	95.53	91.92	85.04	99.47	98.48	98.97	97.97
D2	8,192	30×30	30 (0%)	98.12	94.16	97.57	95.84	92.01	88.95	94.91	91.83	84.90	99.39	98.52	98.95	97.93
E2	10,240			97.96	96.49	94.94	95.71	91.76	90.11	91.53	90.81	83.17	98.96	98.79	98.87	97.77
F2	12,288			98.29	95.43	97.78	96.59	93.40	83.76	95.31	89.16	80.44	99.64	98.53	99.08	98.18
G2	8,192	20×20	20 (0%)	93.14	96.26	95.47	95.86	92.05	62.69	92.99	74.89	59.86	99.08	93.13	96.01	92.33
H2	10,240			97.54	93.31	97.52	95.37	91.15	88.52	89.53	89.02	80.22	98.72	98.53	98.63	97.30
I2	12,288			97.07	97.38	90.36	93.74	88.21	87.81	85.32	86.55	76.29	98.18	98.61	98.39	96.84
J2	8,192	40×40	30 (25%)	98.88	95.98	98.56	97.25	94.65	95.55	96.50	96.02	92.35	99.46	99.27	99.36	98.73
K2	10,240			98.17	93.67	97.64	95.61	91.60	92.83	94.18	93.50	87.79	99.11	98.83	98.97	97.95
L2	12,288			98.34	96.72	95.87	96.29	92.85	91.35	94.52	92.91	86.76	99.30	98.86	99.08	98.18
M2	8,192	40×40	20 (50%)	98.42	97.32	97.68	97.50	95.13	93.15	93.78	93.47	87.74	99.16	99.06	99.11	98.24
N2	10,240			98.44	96.41	97.23	96.82	93.84	94.10	92.92	93.51	87.80	99.06	99.22	99.14	98.29
O2	12,288			98.47	96.93	94.02	95.45	91.30	93.92	93.22	93.57	87.91	99.10	99.24	99.17	98.36

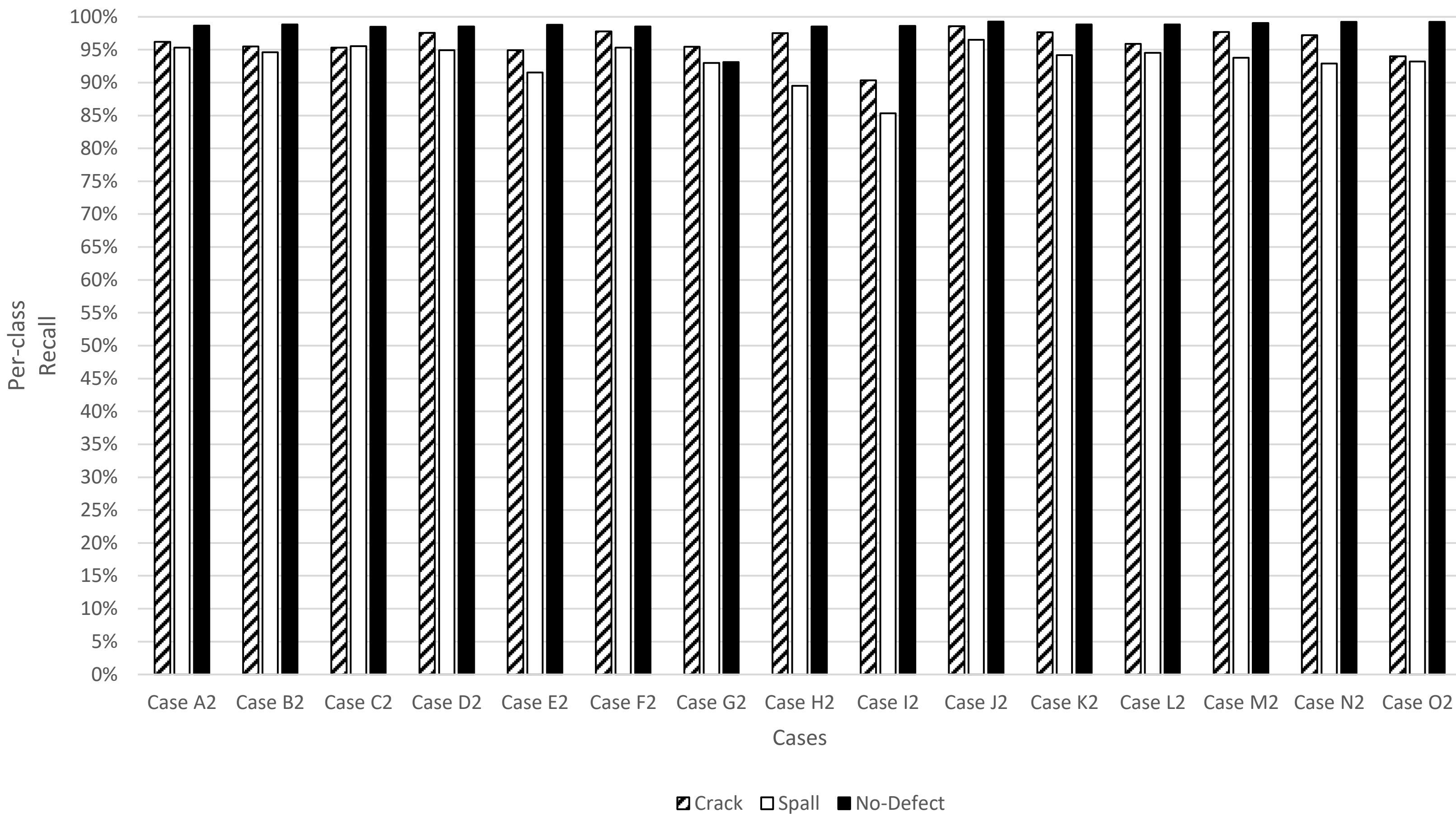


Figure 6-14. Recall test results for NVE-DGCNN

1

For sensitivity analysis of NVE-DGCNN, fifteen cases (Case A2 to O2) are defined and validated using three numbers of points of 8,192, 10,240, and 12,288, and three block sizes of 40 × 40 *cm*, 30 × 30 *cm*, and 20 × 20 *cm*. Moreover, the stride could not be applied on 20 × 20 *cm* due to the computation resource limitations (i.e., RAM size). Therefore, the effect of stride size is investigated for the block size of 40 × 40 *cm* by applying 0%, 25%, and 50% strides (40, 30, and 20 *cm*). Nine cases were studied with 0% stride (A2 to I2). In all cases with the same block size and stride, the number of points increased from 8,192 to 12,288 (Case A2 to O2). In three cases (J2 to L2), the stride was decreased by 25%, and in the remaining cases (M2 to O2) the stride considered was 50%.

Figure 6-15 shows the effect of the number of points based on the average recall value. As shown in this figure, increasing the number of points will decrease the recall of defects in the NVE-DGCNN network. Based on [27], this can be explained by the mismatch between the density and the value of the number of the *K*-nearest neighbors. Moreover, increasing the number of points may occasionally result in overfitting [345]. Figure 6-16 shows the effect of block size based on the average recall value. As shown in this figure, decreasing the block size to 25% (30 *cm* × 30 *cm*) will increase the crack recall by 1.17%. However, the accuracy of spall decreased by 0.17%. The recall of both crack and spall was decreased by decreasing the block size to 50% (20 *cm* × 20 *cm*). Figure 6-17 shows the effect of decreasing the stride based on the average recall value. As shown in this figure, decreasing the stride to 25% (30 *cm*) improved the crack recall by 1.63%. The recall of both crack and spall was decreased by decreasing the stride to 50% (20 *cm)*.

NVE-DGCNN improves the semantic segmentation performance of cracks a little more than spalls (almost 2% in Case J2). Adding the normal vector feature to the points specifies additional geometric information. On the other hand, the range of the change of normal vector in crack is less than in spalls. Therefore considering the normal vector may increase the chance of detecting cracks more than spall. In the testing phase, most of unforeseen crack points were considered as non-defects in adapted DGCNN, but where correctly classified as cracks in NVE-DGCNN.

Furthermore, the EdgeConv layer can detect the edges by applying an operation on edges to define the relationships between a point and its neighbors [27, 346]. Three samples of the best results for Case J2 from the NVE-DGCNN are shown in Figure 6-18. As shown in this figure, the NVE-DGCNN detected most of the cracks when the normal vector feature was added to the points. In addition, in Sample 2, the network efficiently detected a line that looks like a crack as a non-defect. This example indicates the advantage of point cloud-based methods over image-based methods.

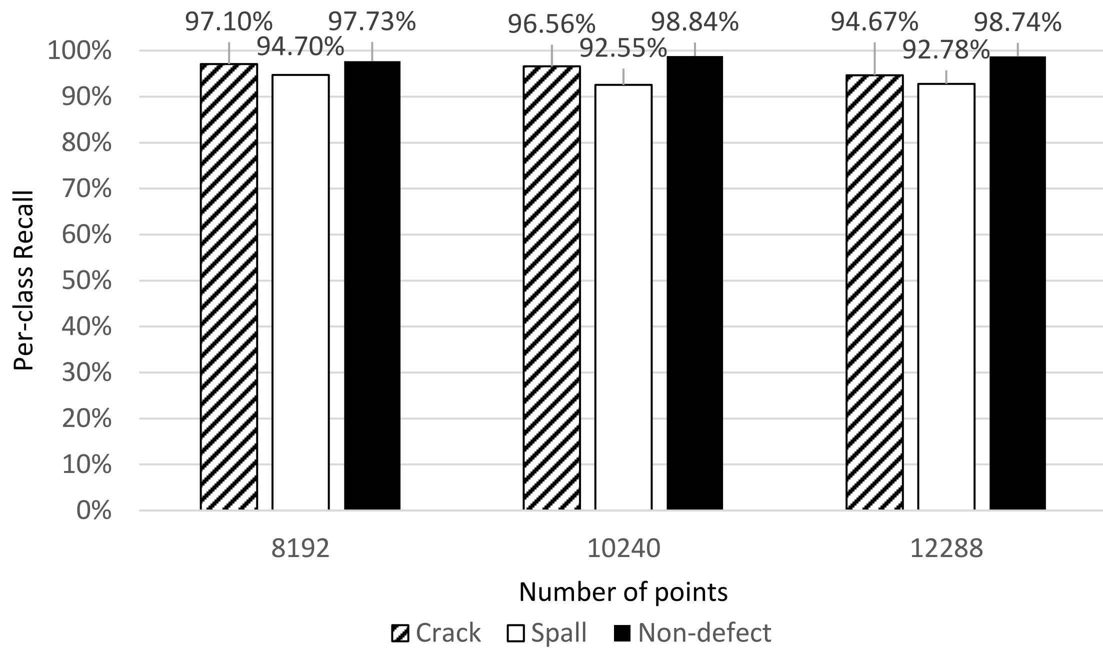


Figure 6-15. Effect of the number of points based on the average recall value for NVE-DGCNN (all cases)

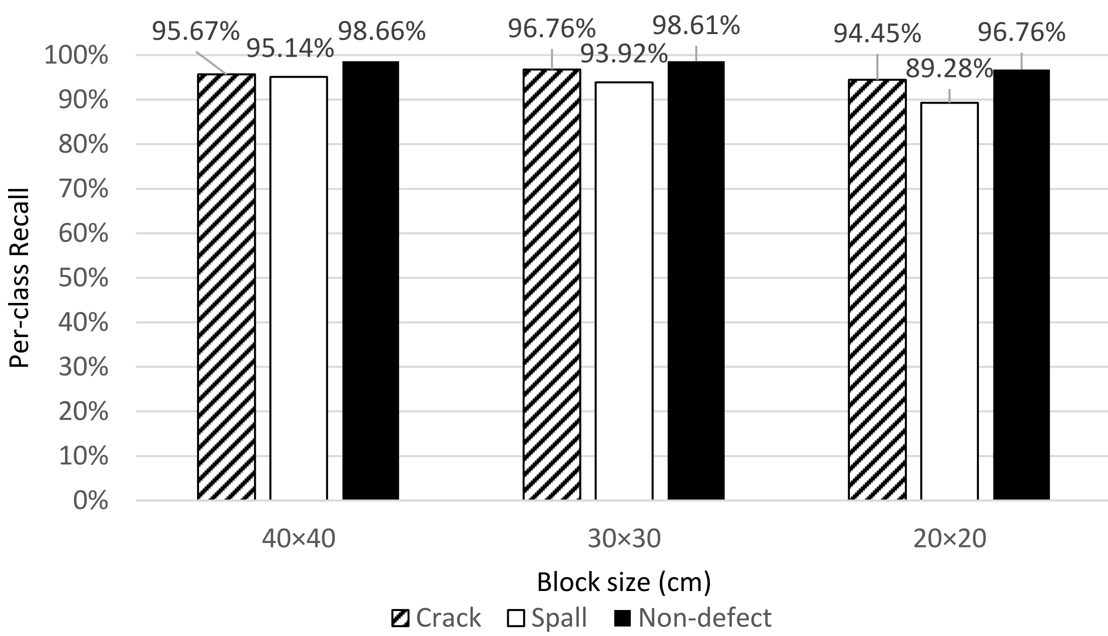


Figure 6-16. Effect of block size based on the average recall value for NVE-DGCNN (cases with 0% stride)

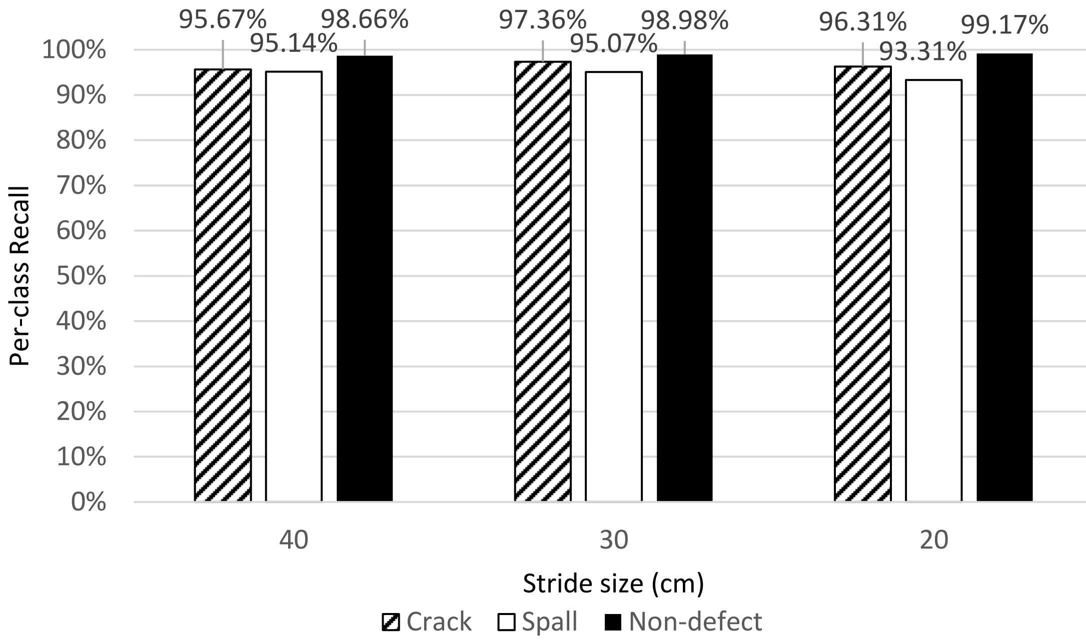


Figure 6-17. Effect of stride size based on the average recall value for NVE-DGCNN

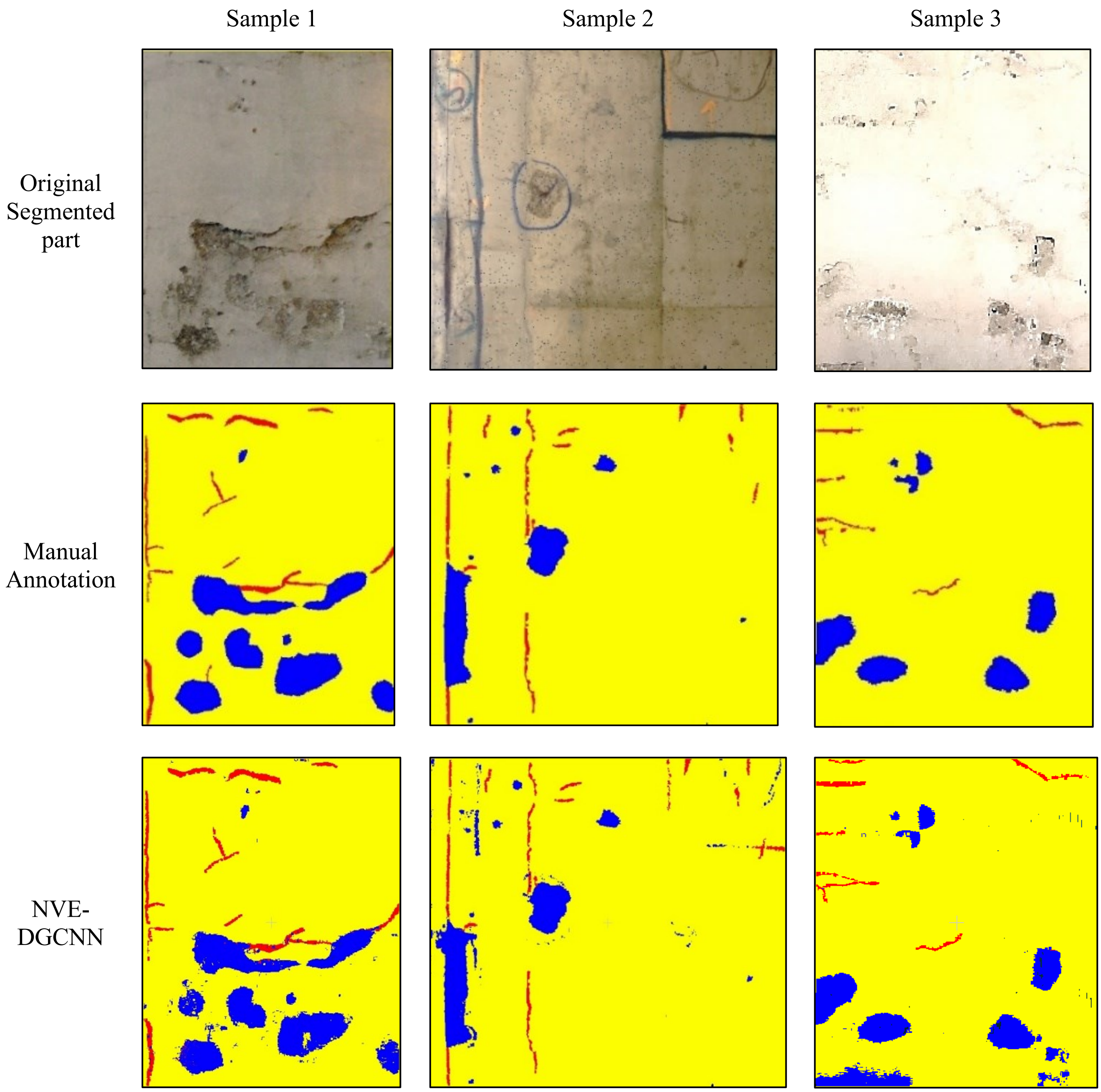


Figure 6-18. Test results from three samples of NVE-DGCNN (Case J2)

As shown in Table 6-16, deeper parts can increase recall up to 99.38% for crack and 99.41% for spall for Cases J2.

Table 6-16. Semantic segmentation recall based on the depth of defects for NVE-DGCNN (%)

Case	Number of sampled points for each block	Depth (cm)					
		D≤3		3<D<7		7≤D	
		Crack	Spall	Crack	Spall	Crack	Spall
J2	8,192	94.67	94.75	99.32	97.36	99.38	99.41

As shown in Table 6-17, the comparison between adapted DGCNN and NVE-DGCNN shows that using normal vector as an additional point feature improved the model's accuracy for the same number of points of 8,192, 10240, and 12288, and block size of 40 × 40 *cm* with 0% stride.

Table 6-17. Comparison of the results of adapted DGCNN and NVE-DGCNN (recall %)

Number of sampled points for each block	Block size (cm)	Adapted DGCNN		NVE-DGCNN	
		Crack	Spall	Crack	Spall
8,192	40×40	55.20	89.77	96.20	95.30
10,240	40×40	55.31	89.41	95.49	94.60
12,288	40×40	58.67	87.40	95.32	95.53

In addition, to determine the effect of *K*-nearest neighbors in the model, five cases with different numbers of *K* (5, 10, 15, 20, and 25) are defined for Case J2, which has the best performance. As Table 6-18 shows, the case of *K* equal to 20 (suggested value by Wang et al. [27]) still has the best performance and increasing the number of *K* will decrease the network's performance. As explained earlier in this section, mismatch between the density and the value of the number of the *K*-nearest neighbors may decrease the performance [27].

Table 6-18. Results of best NVE-DGCNN case with different numbers of *K*-nearest neighbors

Case	Number of sampled points for each block	Stride (cm)	Number of nearest neighbours (K)	Crack	Spalling
				Recall	Recall
J2-1	8192	30 (25%)	5	98.14	95.22
J2-2			10	97.67	94.23
J2-3			15	98.21	94.21
J2-4			20	98.56	96.50
J2-5			25	97.34	94.98

6.5 Case Study of As-inspected Modeling

This section aims to automate the process of as-inspected modeling based on the results of the concrete surface defects semantic segmentation, including cracks and spalls. The goal of the case study is to implement as-inspected modelling. The case study involves one of the scanned bridges in Section6.3. The scan stations were located under the bridge on *Rue Lucien-L'Allier* between *Rue*

Saint-Antoin West and *René-Lévesque Boulevard West, Montreal*. Figure 6-19 shows the images of bridge.

Figure 6-19. Images of the bridge

The point cloud registration process is done with Trimble RealWorks, and the point cloud of the bridge is cleaned up from most unrelated data using Recap Pro. Figure 6-20 shows the registered scan of the bridge.

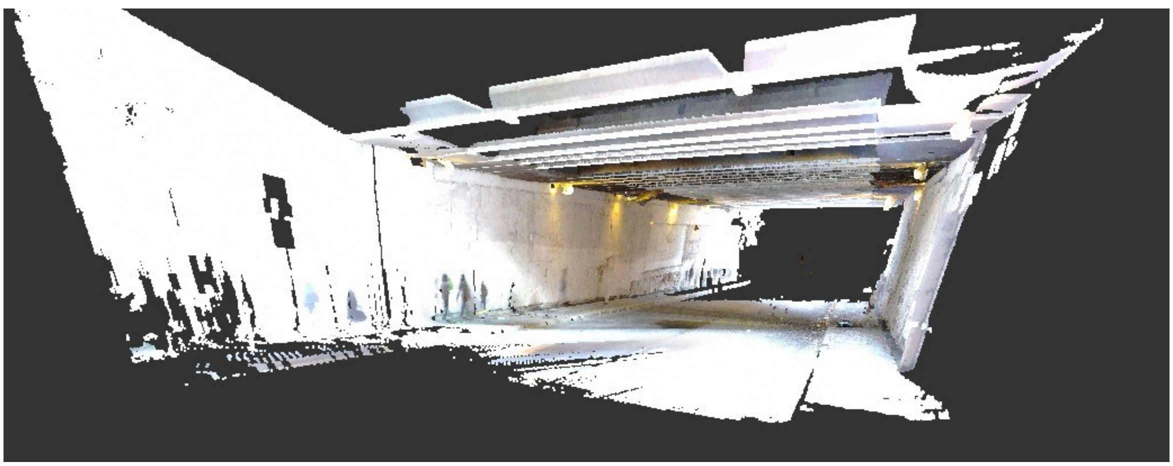
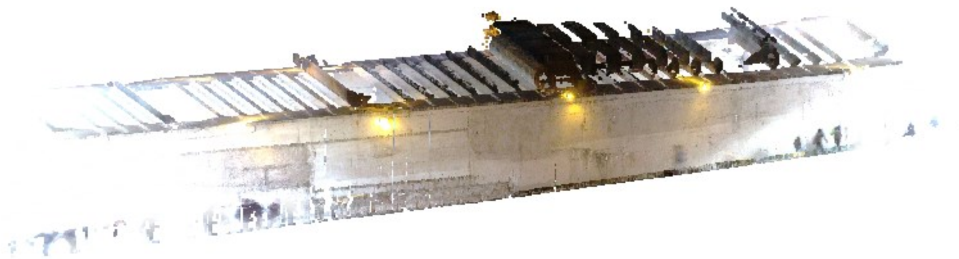
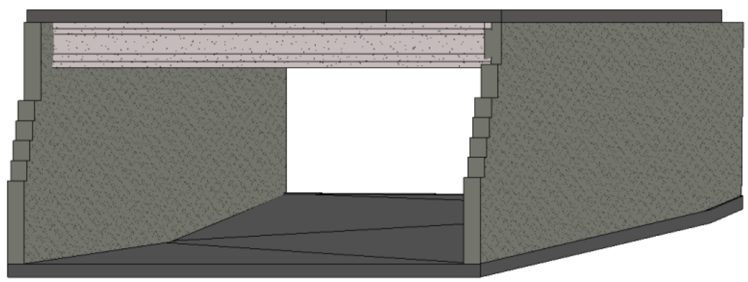


Figure 6-20. Cleaned up scan of the bridge

The inspector usually focuses on scanning the areas that are expected to have defects. However, in this study, an additional step of scan-to-BIM was done, as the 3D model of the bridge was not available. In this step, the registered scan file was exported to ". rcs" format, and then imported into Autodesk Revit 2019 software. Figure 6-21(a) shows the imported cloud data in Revit software. Figure 6-21(b) shows the 3D model of the bridge based on point cloud data.



(a) Imported point cloud data in Revit



(b) 3D model of the bridge

Figure 6-21. 3D model of the bridge based on point cloud data

As Figure 6-22 shows, a sample of the point cloud on the abutment surface of the bridge is selected for defect semantic segmentation.

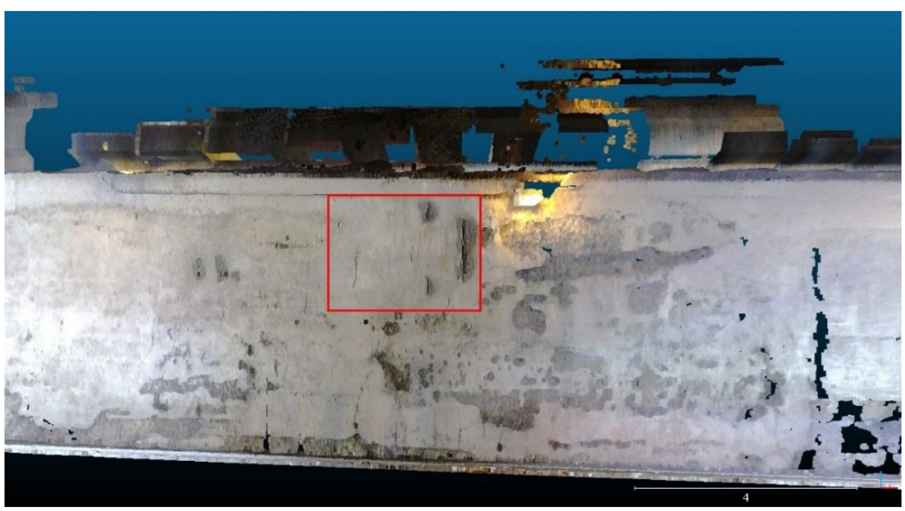
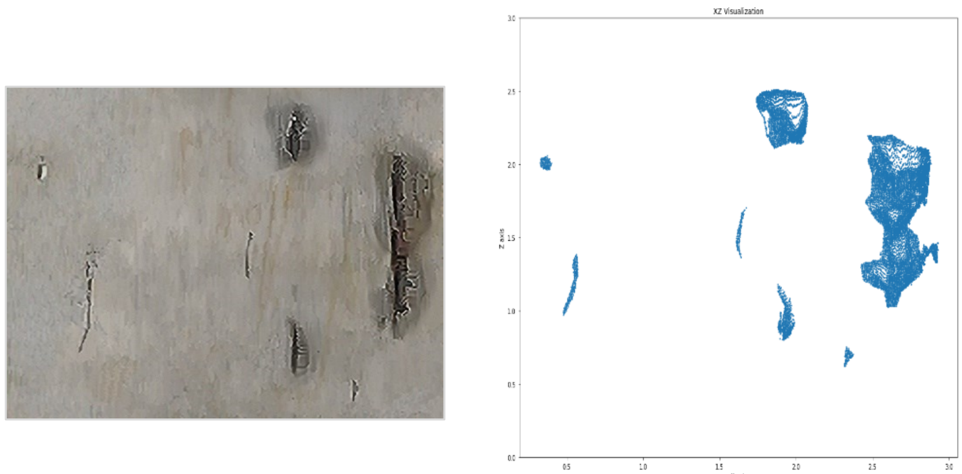


Figure 6-22. Selected part from point cloud data

Figure 6-23(a) shows crack and spall defects on point cloud data. Figure 6-23(b) shows detected crack and spall defects using NVE-DGCNN. As discussed in Section 6.2.3, detected crack and spall defects' length, width, and depth were calculated using density-based spatial clustering and the MBB algorithm. Figures 6-24(a) and (b) show the clusters of the cracks and spalls, respectively. Figure 6-25(a) shows an example of the MBB of crack cluster Number 0 of in the sample. Figure 6-25(b) shows an example of the MBB for spall cluster Number 2 in the sample. Tables 6-19 and 6-20 show the geometrical and semantic information of cracks and spalls of the sample, respectively.

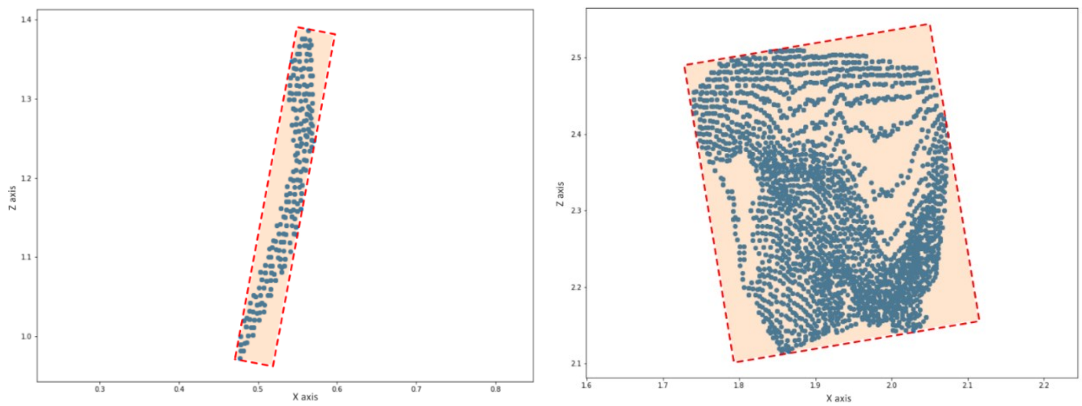


(a) Original point cloud **(b) Detected defects using NVE-DGCNN**

Figure 6-23. Visualization of detected spalls and cracks of the sample

(a) Clusters of cracks **(b) Clusters of spalls**

Figure 6-24. Clusters in the sample



(a) cluster Number 0 of cracks **(b) cluster Number 2 of spalls**

Figure 6-25. An example of MBBs for cracks and spalls clusters in the sample

Table 6-19. Geometric and semantic information of crack defects in the sample

Cluster number	Length (mm)	Width (mm)	Depth (mm)	Severity	Condition
0	423	45	4	Wide	Poor condition
1	339	41	4	Wide	

Table 6-20. Geometric and semantic information of spall defects in the sample

Cluster number	Length (mm)	Width (mm)	Depth (mm)	Severity	Condition
0	90	69	8	Light	Poor condition
1	376	108	35	Medium	
2	391	324	68	Medium	
3	127	53	3	Light	
4	1,168	495	64	Very sever	

The 3D model of the as-inspected bridge based on point cloud data is used for defect product modeling. After applying clustering and MBB algorithms, the segmented defects are aligned to the initial coordinate system using NICP algorithms, as explained in Section 6.2.3. Then, the segmented defects with an aligned coordinate system are saved as new files to be used in the 3D modeling step. The results of aligned segmented defects look similar to Figure 6-22.

The clusters of detected defects are converted to 3D mesh products to have an accurate model of the defect objects. A Dynamo script using Mesh Toolkit [340] is utilized to import the 3D mesh defects into the BIM model. Figure 6-26 shows the imported defect objects into the model of the bridge abutment in Revit. Finally, the semantic information of each defect, including defects type, the severity level, and the condition of the defected element, are added manually to the BIM model. Figure 6-27 shows an example of semantic information for cluster Number 4 of spalls in the BIM model.

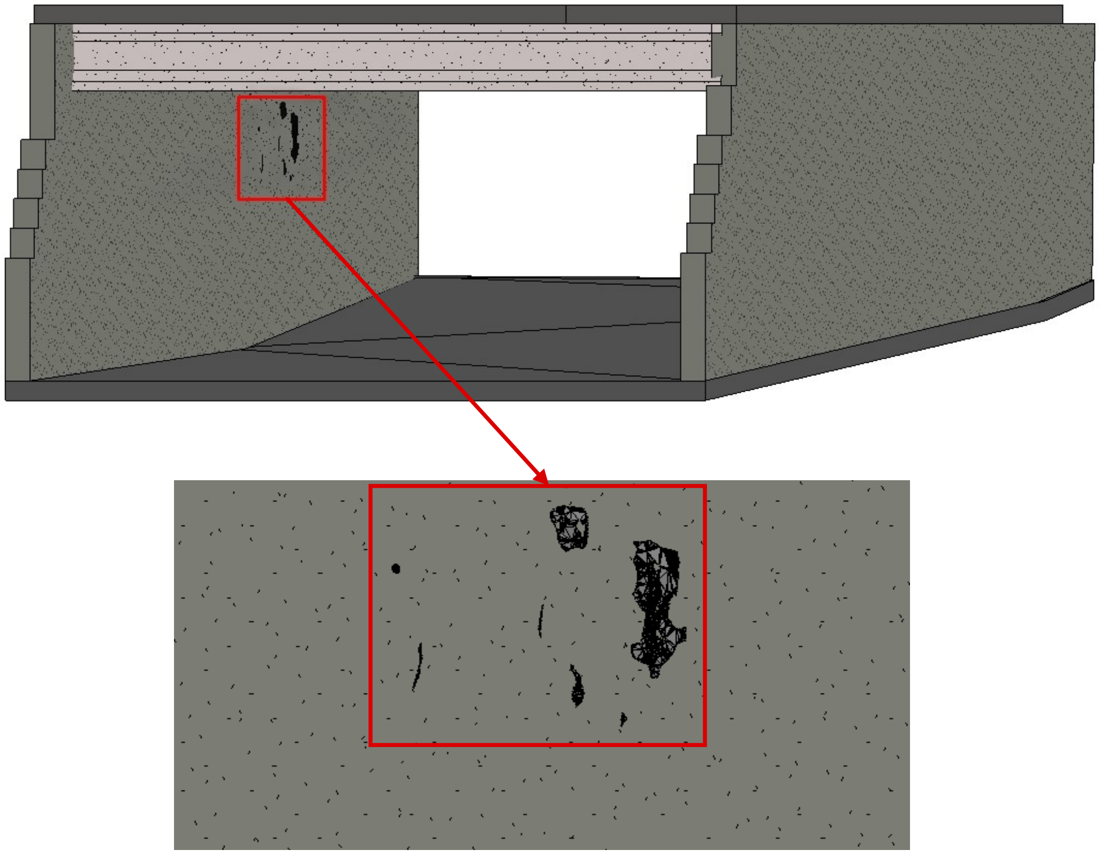


Figure 6-26. Imported bridge abutment's defect objects in Revit

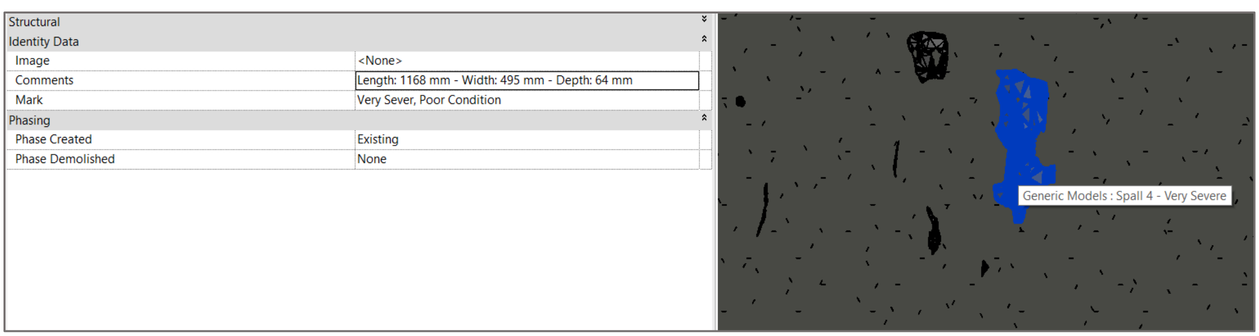


Figure 6-27. An example of semantic information for cluster Number 4 of spalls in BIM

6.6 OCSD Evaluation Based on Point Cloud Semantic Segmentation and As-inspected Modeling

This section aims to evaluate the usage of OCSD based on the case study explained in Section 6.5. The remote sensing method is used in the inspection process. Table 6-21 shows examples of the related OCSD inspection process concepts and relationships used in the case study.

Table 6-21. Related OCSD inspection process concepts and relationships

Concept	Relationship	Concept
Concrete surface defects	is	Inspection target
Remote sensing	is	Measurement method
Computer vision (semantic segmentation)	is	Remote sensing method
LiDAR (Faro Focus3D scanner)	is	Measurement device
Point cloud	is	Collected data
Clustering (density-based Spatial clustering)	is	Method of processing inspection data
Crack, spalling	is	Concrete surface defect types
Crack defect form change	is	Fracture
Spall defect form change	is	Section loss

In this section, the diagnosis and 3R processes are based on the engineering judgment. The conditions of concrete surfaces are determined based on defect dimensions and semantic information of OCSD. Cause analysis is a heuristic method based on inspector's experience. The causes of defects in this case study are the environmental problems during the operation phase, including water, freeze and thaw effect, and well as the lack of maintenance during the maintenance phase. As the condition of the sample is poor, there is a need for 3R processes. Table 6-22 shows examples of the related OCSD diagnosis process concepts and relationships used in the case study.

Table 6-22. Related OCSD diagnosis process concepts and relationships

Concept	Relationship	Concept
Remote sensing-based method	is	Diagnosis method
Cause analysis	is	Heuristic method
Problem during operation (water presence, freeze and thaw effect)	is	Defects Cause
Problem during maintenance (lack of maintenance)		
Poor condition	is	Condition
Condition assessment	evaluates	Need for 3R processes (in this case YES)

The last step is dedicated to the 3R actions. The bridge is located in harsh weather conditions, and the concrete surface is exposed to water presence. Therefore, this step should be done in two phases. The first phase is fixing with mortar compound, and the next phase is waterproofing the surface with epoxy coating. Conventional mortar is selected as the repair material as a matching compound, which has thermal expansion close to the existing material of the bridge. The first step before repairing the surface is removing the loose material and cleaning the surface. As the cracks are wide, they should be filled with conventional mortar. Conventional mortar is also applied to the cleaned surface to repair the spall defects. After applying a new surface layer, the next step is curing the new material to gain the required strength. Due to the bridge's location and its exposure to harsh weather and water, epoxy coating is applied as a waterproofing compound. Table 6-23 shows the examples of related OCSD 3R processes concepts and relationships used in this case study.

Table 6-23. Related OCSD 3R processes concepts and relationships

Concept	Relationship	Concept
Concrete repair	is	3R method
Surface sealing/coating		
Conventional mortar, epoxy	is	Repair material
Removing loose concrete	is	Concrete repair
Adding mortar		
Crack filling		
Epoxy Coating	is	Protective coating cover

The above evaluation confirms that OCSD covers the domain's concepts and relationships needed in the case study for inspection, diagnosis, and 3R processes of concrete surface defects. With the aid of OCSD and as-inspected product model, any changes can be tracked and visualized throughout the lifecycle.

6.7 Discussion

In another research, Bolourian et al. [347] investigated the effect of adding normal vectors, to the adapted PointNet++ (SNEPointNet++). The best results of this network along with the best results of the NVE-DGVNN are shown in Table 6-24. Compared to SNEPointNet++, NVE-DGCNN recalls in detecting cracks and spalls are 5.56 % and 4.50 % better, respectively.

Table 6-24. Comparison between best results of the NVE-DGCNN and SNEPointNet++ (%)

Method	Cracks				Spalls				Non-defects			
	Precision	Recall	F1 score	IOU	Precision	Recall	F1 score	IOU	Precision	Recall	F1 score	IOU
SNEPointNet++ [347]	73.3	93.0	81.9	69.2	89.9	92.0	90.6	82.5	99.3	98.8	99.0	98.1
NVE-DGCNN	95.98	98.56	97.25	94.65	95.55	96.50	96.02	92.35	99.46	99.27	99.36	98.73

Table 6-25 shows that the current image-based classification methods have reached 99.5% recall in concrete surface crack defect classification. However, this study aimed to determine the semantic information of each point separately, while the classification methods are not suitable for this purpose. Moreover, the proposed method focused on multiclass point cloud semantic segmentation while the image-based methods focused on binary classification or binary semantic segmentation. To this end, the performance of NVE-DGCNN recall is higher than previous image-based semantic segmentation methods.

Table 6-25. Comparison between NVE-DGCNN and image-based computer vision methods

Author	Year	Image classification	Image semantic segmentation	Point cloud semantic segmentation	Surface material	CCN input data	Type of defect	Precision (%)	Recall (%)	F1-Score (%)
Yang et al. [273]	2018	-	✓	-	Concrete wall, pavement	Image	Crack	81.73	78.97	79.95
Liu et al. [274]	2019	-	✓	-	Concrete	Image	Crack	90	91	90
Lopez Droguett et al. [348]	2020	-	✓	-	Concrete	Image	Crack	N/A	97.9	97.5
Ali et al. [275]	2021	✓	-	-	Concrete	Image	Crack	99.7	85	91.8
Le et al. [276]	2021	✓	-	-	Concrete	Image	Crack	96.5	98.8	97.7
Vignesh et al. [277]	2021	✓	-	-	Concrete	Image	Crack	96.69	99.55	98.1
Mohammed Abdelkader et al. [349]	2021	-	✓	-	Concrete	Image	Spall	N/A	N/A	90.98
NVE-DGCNN	ours	-	-	✓	Concrete	Point cloud	Crack	95.98	98.56	97.25
							Spall	95.55	96.50	96.02

Moreover, the previous point cloud-based works for concrete defects such as [255, 264, 265, 271], utilized different metrics (i.e. error) or visualization approaches to show the results. Therefore, the results are incomparable.

6.8 Summary and Conclusions

This chapter developed a new method for point cloud-based defect semantic segmentation (NVE-DGCNN) to automate the inspection process of concrete surface defects, including cracks and spalls, without transforming the point cloud into other representations. Moreover, this chapter investigated two main characteristics related to surface defects (i.e. normal vector and depth). The challenges related to the size of the dataset and imbalanced classes were studied. Sensitivity analysis was applied to capture the best combination of hyperparameters and investigate their effects on the network performance. In addition, post-processing of the semantic segmentation was done to automate the process of as-inspected modeling.

The network's performance was improved by modifying the network (e.g., KNN for EdgeConv and the loss function) and by augmenting the dataset (i.e. by flipping the point cloud data). The testing showed the usefulness and robustness of the proposed method in detecting concrete surface defects from 3D point cloud data. Moreover, the results showed that the normal vector can be an important factor in the learning process of the model and detecting the edge of cracks.

It is concluded that: (1) NVE-DGCNN resulted in 98.56% and 96.50% recalls for semantic segmentation of cracks and spalls, respectively. NVE-DGCNN is more accurate than other point cloud-based methods; (2) The sensitivity analysis results showed that decreasing the size of blocks to less than 30 × 30 *cm* decreased the recall, as increasing the density of blocks can cause overfitting or failure in Euclidean distance computing. Moreover, decreasing the stride to 25%

improved the network performance in terms of recall for the block size of 40 × 40 *cm*. However, decreasing the stride to 50% was not beneficial and decreased the recall. Finally, the sensitivity analysis showed that NVE-DGCNN is not very sensitive to the points density; (3) The case study showed that deeper cracks and spalls in the dataset are easier to detect. In deeper samples, the recalls for cracks and spalls reached 99.38% and 99.41%, respectively; (4) The semi-automated process of as-inspected modeling made it possible to manage and visualize the detected defects by collecting their dimensions and identifying the conditions on the 3D model; and (5) The semantic representation of OCSD was evaluated through the case study to demonstrate its benefits, and the evaluation indicates that the domain's concepts and relationships related to the case study are covered in OCSD. Moreover, the case study validates the systematic approach of OCSD and expands the usability of concrete infrastructure management through the defect product model as a critical component of OCSD.

CHAPTER 7. SUMMARY, CONTRIBUTIONS, AND FUTURE WORK

7.1 Summary of Research

This research aims to enhance the efficiency of facilities inspection by proposing BIM-based inspection-related Knowledge models along with an integrated process of surface defect semantic segmentation and defect modeling.

In Chapter 2, facilities management, and different types of concrete surface defects were reviewed. Furthermore, the ontological approach and the related concepts of BIM-based inspection and robotic inspection and navigation tasks were covered. Moreover, defect information modeling and limitations of previous research were explained. Finally, point cloud-based deep neural network applications in semantic segmentation were discussed. Chapter 3 explained the overall proposed framework of this research.

Chapter 4 focused on the development of an ontology, called OCSD, for concrete surface defects to have a unified knowledge model where all the stakeholders can access information in a systematic manner. OCSD metrics include 333 classes, 51 relations, 27 attributes, and 31 individuals. OCSD comprises high-level knowledge of the concepts and relationships related to surface defects, inspection, diagnosis, and 3R processes. Consistency of OCSD was evaluated using HermiT OWL reasoner. The application of OCSD was investigated in a case study and a survey was designed to evaluate the semantic representation of OCSD. OCSD considers the BIM model of the defect as a critical component through a systematic approach. The defect product model will help the stakeholders benefit from accessing the defect information and condition of defected elements, which will result in enhancing the efficiency of inspection and repair processes.

Chapter 5 focused on the development of an integrated ontology, called OBRNIT, to extend BIM applications for robotic navigation and inspection tasks. OBRNIT metrics include 386 classes, 45 relations, 52 attributes, and 8 individuals. OBRNIT comprises high-level knowledge of the concepts and relationships related to buildings, robots, and navigation and inspection tasks. BIM is considered as a reference that is integrated with the knowledge model. The HermiT OWL reasoner was used to evaluate the consistency of OBRNIT. The semantic representation of OBRNIT was evaluated through a case study and a survey. Although having an actual case study using an inspection robot is interesting, for the purpose of OBRNIT evaluation, considering the specifications of the inspection robot in the case study is enough.

Chapter 6 focused on developing a method for LiDAR-based defect semantic segmentation based on NVE-DGCNN to automate the inspection process of concrete surface defects, including cracks and spalls. This chapter investigated two main characteristics related to surface defects, including normal vector and depth. The network's performance was improved by modifying the network (e.g., KNN for EdgeConv and the loss function) and by augmenting the dataset (i.e. by flipping the point cloud data). The challenges related to the size of the dataset and imbalanced classes were studied. Sensitivity analysis was applied to capture the best combination of hyperparameters and investigate their effects on the network performance. Furthermore, post-processing of the results of the concrete surface defects semantic segmentation was done to semi-automate the process of as-inspected modeling. As-inspected BIM includes the updated information of the facilities at the time of data collection. The BIM model was used to capture and visualize the semantic segmentation results by reflecting the geometric and semantic information of defects and identifying the element conditions on the BIM model.

7.2 Research Contributions and Conclusions

This research results in the following contributions:

(1) Developing an ontology (OCSD) for inspection, repair, and 3R processes of concrete surface defects. The current version of OCSD is available at https://github.com/OCSD-OWL/OCSD. The following conclusions can be drawn from this contribution:

- The evaluation proves that OCSD satisfies the domain experts and covers the domain's main concepts and relationships. OCSD was able to provide a clear understanding of the concepts and relationships in the domain.
- OCSD can help future asset management systems benefit from the provided knowledge and efficiently develop, modify, and process the ontological knowledgebase.
- The semantic representation of OCSD was evaluated through the case study to demonstrate its benefits, and the evaluation indicates that the domain's concepts and relationships related to the case study are covered in OCSD. Moreover, the case study validates the systematic approach of OCSD and expands the usability of concrete infrastructure management through the defect product model as a critical component of OCSD.

(2) Developing BIM-based ontology (OBRNIT) to cover the different types of information and concepts related to robot navigation and inspection tasks. The current version of OBRNIT is available at https://github.com/OBRNIT/OBRNIT. The following conclusions can be drawn from this contribution:

- The evaluation demonstrates that OBRNIT covers the domain's concepts and relationships up to the point that satisfies the domain experts. Based on the evaluation, OBRNIT was able to give a clear understanding of the concepts and relationships in the domain, and it can be applied for developing robotic inspection systems.
- OBRNIT extends the BIM application for robotic navigation and inspection tasks. OBRNIT can help system engineers involved in developing robotic inspection systems by identifying the different concepts and relationships about robotic inspection and navigation tasks based on BIM information.
- **(3) Developing a method (NVE-DGCNN) for point cloud-based concrete surface defects semantic segmentation.** The following conclusions can be drawn from this contribution:
- NVE-DGCNN resulted in 98.56% and 96.50% recalls for semantic segmentation of cracks and spalls, respectively. NVE-DGCNN is more accurate than other point cloud-based methods.
- The sensitivity analysis results showed that decreasing the size of blocks to less than 30 × 30 *cm* decreased the recall, as increasing the density of blocks can cause overfitting or failure in Euclidean distance computing. Moreover, decreasing the stride to 25% improved the network performance in terms of recall for the block size of 40 × 40 *cm*. However, decreasing the stride to 50% was not beneficial and decreased the recall. Finally, the sensitivity analysis showed that NVE-DGCNN is not very sensitive to the points density.
- The case study showed that deeper cracks and spalls in the dataset are easier to detect. In deeper samples, the recalls for cracks and spalls reached 99.38% and 99.41%, respectively.

(4) Developing a semi-automated process for as-inspected modeling. The following conclusion can be drawn from this contribution:

- The semi-automated process of as-inspected modeling made it possible to manage and visualize the detected defects by collecting their dimensions and identifying the conditions on the 3D model.

7.3 Limitations and Future Work

Despite the above-mentioned contributions, this research has some limitations that should be addressed in the future. The limitations can be organized in four categories as follows:

(1) Limitations related to OCSD ontology

- The scope of this research does not cover all possible concepts of concrete surface inspection, diagnosis, and 3R processes. For example, some types of inspection, such as underwater inspection, are not covered in OCSD. In the future, OCSD will be extended to cover other types of related concepts.
- OCSD knowledgebase can be used in the future to develop concrete surface inspection expert systems, software, or checklists.

(2) Limitations related to OBRNIT ontology

- The scope of this research does not address the low-level path planning and the problem of SLAM [350]. Future work will focus on further development of OBRNIT to integrate it with low-level robotic capabilities to make the robot more autonomous. The abstract knowledge can be combined with robot action-related procedural knowledge to make the tasks executable [10].
- A planning language system for reasoning over the ontological knowledgebase for the execution of plans can be developed, benefiting from previous research [149].
- Extending OBRNIT will be considered by linking it with other available ontologies (e.g. Sensor Ontology) [351].

(3) Limitations related to the point cloud-based semantic segmentation

- A larger dataset is expected to improve the learning process resulting in better performance of the model. Therefore, Future work will focus on collecting and preparing more data to enlarge the dataset. Moreover, due to computing resource limitations (i.e. memory and processors limitation), it was impossible to study the effect of increasing the number of input points of the model to more than 12,288 per block. Having more computing resources will make the opportunity to expand the sensitivity analysis.
- The proposed method only considered flat surfaces. The performance of the network on curved surfaces needs more investigation.
- In the future, the dataset can be classified into more classes in order to consider the levels of severity as described in OCSD. However, more data in terms of variety and quantity needs to be considered.
- The effects of each point feature on the performance of the NVE-DGCNN network were not examined independently (e.g. color). Therefore, in the future, it is important to investigate the impact of each feature on network performance.

(4) Limitations related to the as-inspected modeling

- As-inspected modeling will help store the inspection results efficiently and precisely, resulting in tracking and analysis of the defect changes throughout the lifecycle. The version control (i.e. time-series) of as-inspected modeling and tracking of the changes of the as-inspected BIM models will be investigated in the future. Furthermore, the BIM model can be modified after performing repairs. In this regard, time-series of as-repaired models needs to be considered.

- In this research, the defect's semantic information, such as severity level, is added manually to the BIM model. Future work will focus on a fully automated approach to integrate the semantic knowledge of OCSD with as-inspected modeling.

REFERENCES

[1] Nepal, M. P., Staub-French, S., Pottinger, R., & Zhang, J., "Ontology-based feature modeling for construction information extraction from a building information model," *Journal of Computing in Civil Engineering,* vol. 27, no. 5, pp. 555-569, 2012.

[2] Balaguer, C., Gimenez, A. and Abderrahim, C.M., "ROMA robots for inspection of steel based infrastructures," *Industrial Robot,* vol. 29, no. 3, pp. 246-251, 2002.

[3] Bao, Y., Tang, Z., Li, H. and Zhang, Y., "Computer vision and deep learning–based data anomaly detection method for structural health monitoring," *Structural Health Monitoring,* vol. 18, no. 2, pp. 401-421, 2019.

[4] Biagini, C., Capone, P., Donato, V., & Facchini, N., "Towards the BIM implementation for historical building restoration sites," *Automation in Construction,* vol. 71, pp. 74-86, 2016.

[5] Kim D, Goyal A, Newell A, Lee S, Deng J, Kamat VR., "Semantic relation detection between construction entities to support safe human-robot collaboration in construction," in *Proceedings of the ASCE International Conference on Computing in Civil Engineering: Data, Sensing, and Analytics*, Reston, VA, 2019.

[6] Lim, G.H., Suh, I.H. and Suh, H, "ontology-based unified robot knowledge for service robots in indoor environments," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans,* vol. 41, no. 3, pp. 492-502, 2010.

[7] Lattanzi, D. and Miller, G., "Review of robotic infrastructure inspection systems," *Journal of Infrastructure Systems,* vol. 23, no. 3, pp. 04017004/1-16, 2017.

[8] Saigol, Z., Wang, M., Ridder, B. and Lane, D.M., "The Benefits of Explicit Ontological Knowledge-Bases for Robotic Systems," in *Proceedings of the 16th Annual Conference on Towards Autonomous Robotic Systems*, Liverpool, United Kingdom, 2015a.

[9] Brunner, S., Kucera, M. and Waas, T., 2017, June., "Ontologies used in robotics: A survey with an outlook for automated driving," in *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, Vienna, Austria, 2017.

[10] Stulp, F. and Beetz, M., "Combining declarative, procedural, and predictive knowledge to generate, execute, and optimize robot plans," *Robotics and Autonomous Systems,* vol. 56, no. 11, pp. 967-979, 2008.

[11] Guldur, B., Yan, Y. and Hajjar, J.F., "Condition assessment of bridges using terrestrial laser scanners," in *Proceedings of Structures Congress on Bridges and Other Structures*, Portland, Oregon, 2015.

[12] Jahanshahi, M.R., Kelly, J.S., Masri, S.F. and Sukhatme, G.S., "A survey and evaluation of promising approaches for automatic image-based defect detection of bridge structures," *Structure and Infrastructure Engineering,* vol. 5, no. 6, pp. 455-486, 2009.

[13] Koch, C. and Brilakis, I., "Pothole detection in asphalt pavement images," *Advanced Engineering Informatics,* vol. 25, no. 3, pp. 507-515, 2011.

[14] Hutchinson, T.C. and Chen, Z., "Improved image analysis for evaluating concrete damage," *Journal of Computing in Civil Engineering,* vol. 20, no. 3, pp. 210-216, 2006.

[15] Barazzetti, L. and Scaioni, M., "Crack measurement: Development, testing and applications of an automatic image-based algorithm," *SPRS Journal of Photogrammetry and Remote Sensing,* vol. 64, no. 3, pp. 285-296, 2009.

[16] Hampel, U. and Maas, H.G., "Cascaded image analysis for dynamic crack detection in material testing," *ISPRS journal of Photogrammetry and Remote Sensing,* vol. 64, no. 9, pp. 345-350, 2009.

[17] Adhikari, R.S., Moselhi, O. and Bagchi, A., "Image-based retrieval of concrete crack properties," *Automation in Construction,* vol. 1, no. 34, pp. 180-194, 2012.

[18] Huang, Y. and Xu, B., "Automatic inspection of pavement cracking distress," *Journal of Electronic Imaging,* vol. 5, no. 1, p. 013017, 2006.

[19] Sun, Y., Salari, E. and Chou, E., "Automated pavement distress detection using advanced image processing techniques," in *Proceedings of IEEE International Conference on Electro/Information Technology*, Windsor, Ontario, Canada, 2009.

[20] Koch, C., Georgieva, K., Kasireddy, V., Akinci, B. and Fieguth, P., "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," *Advanced Engineering Informatics,* vol. 29, no. 2, pp. 196-210, 2015.

[21] Haas, C., "Evolution of an automated crack sealer: a study in construction technology development," *Automation in Construction,* vol. 4, no. 4, pp. 293-305, 1996.

[22] Smith, C.J. and Adendorff, K.K., "Advantages and limitations of an automated visual inspection system," *The South Africa Journal of Industrial Engineering,* vol. 5, no. 1, pp. 27-36, 1991.

[23] Laefer, D.F., Truong-Hong, L., Carr, H. and Singh, M., "Crack detection limits in unit based masonry with terrestrial laser scanning," *Ndt & E International,* vol. 62, no. 1, pp. 66-76, 2014.

[24] Ballard, D.H., "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition,* vol. 13, no. 2, pp. 111-122, 1981.

[25] Fischler, M.A. and Bolles, R.C., "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM,* vol. 24, no. 6, pp. 381-395, 1981.

[26] Grilli, E., Menna, F. and Remondino, F., "A review of point clouds segmentation and classification algorithms," in *Proceedings of the ISPRS International Archives on the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Wuhan, China, 2017.

[27] Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M. and Solomon, J.M., "Dynamic graph cnn for learning on point clouds," *Acm Transactions On Graphics,* vol. 38, no. 5, pp. 1-12, 2019.

[28] Liu, W., Sun, J., Li, W., Hu, T. and Wang, P., "Deep learning on point clouds and its application: A survey," *Sensors,* vol. 19, no. 9, pp. 4188/1-22, 2019.

[29] Kasireddy, V. and Akinci, B., "Towards the integration of inspection data with bridge information models to support visual condition assessment," in *Proceedings of the ASCE International Workshop on Computing in Civil Engineering*, Austin, Texas, 2015.

[30] Orcesi, A.D., Chemineau, H., Lin, P.H., van Gelder, P. and van Erp, N., "A risk analysis for asset management considering climate change," *Transportation Research Procedia,* vol. 14, no. 1, pp. 105-114, 2016.

[31] Venkittaraman, A. and Banerjee, S., "Enhancing resilience of highway bridges through seismic retrofit," *Earthquake Engineering & Structural Dynamics,* vol. 43, no. 8, pp. 1173-1191, 2014.

[32] Pennsylvania State University, "Enhancing Disaster Resilience of Highway Bridges to Multiple Hazards," 2014. [Online]. Available: https://www.transportation.gov/utc/enhancing-disaster-resilience-highway-bridges-multiple-hazards. [Accessed Jan. 2021].

[33] Slaichova, E. and Marsikova, K., "The effect of implementing a maintenance information system on the efficiency of production facilities," *Journal of competitiveness,* vol. 5, no. 3, pp. 60-75, 2013.

[34] Kumar, R. and Gardoni, P., "Modeling structural degradation of RC bridge columns," *Journal of Structural Engineering,* vol. 138, no. 1, pp. 45-51, 2012.

[35] Gheitasi, A. and Harris, D.K., "Effect of deck deterioration on overall system behavior, resilience and remaining life of composite steel girder bridges," in *Proceedings of the Structures Congress*, Boston, Massachusetts, United States , 2014.

[36] Nielsen, D., Chattopadhyay, G. and Raman, D., "Life cycle management of railway bridges: defect management," in *Proceedings of Conference on Railway Engineering*, Brisbane, Australia, 2012.

[37] Le, B. and Andrews, J., "Modelling railway bridge degradation based on historical maintenance data," *In Safety and Reliability,* vol. 35, no. 2, pp. 32-55, 2015.

[38] Brandon, W.C. and Yadlosky, J.M., "Framework for Improving Resilience of Bridge Design," FHWA Report, Report No. FHWA-IF-11-016, Federal Highway Administration, Washington, DC, 2011.

[39] Pereira, C., Hamadyk, E. and Silva, A., "Probabilistic analysis of the durability of architectural concrete surface," *Applied Mathematical Modelling,* vol. 77, no. 1, pp. 199-215, 2020.

[40] Woodson, R.D, Concrete structures: protection, repair and rehabilitation, Oxford, United Kingdom: Butterworth-Heinemann, 2009.

[41] Ontario Ministry of Transportation Ontario, "Ontario Structure Inspection Manual (OSIM)," Ontario ministry of transportation, Provincial highways management division, Highway standards branch, Bridge office, Ontario, Canada, 2018.

[42] Etse, G., Caggiano, A. and Vrech, S., "Multiscale failure analysis of fiber reinforced concrete based on a discrete crack model," *International Journal of Fracture,* vol. 178, no. 1-2, pp. 131-146, 2012.

[43] Hagelia, P., "Origin of map cracking in view of the distribution of air-voids, strength and ASR-gel," in *Proceedings of the International Conference on Alkali-Aggregate Reaction in Concrete*, Beijing, China, 2004.

[44] Allahverdi, A., Kani, E.N., Hossain, K.M.A. and Lachemi, M., "Methods to control efflorescence in alkali-activated cement-based materials," in *Handbook of alkali-activated cements, mortars and concretes*, Sawston, United Kingdom, Woodhead Publishing, 2015, pp. 463-483.

[45] Ramachandran, V.S. and Beaudoin, J.J., "Concrete science," in *Handbook of analytical techniques in concrete science and technology: principles, techniques and applications*, Norwich, NY, USA, William Andrew, 2001, pp. 1-62.

[46] ACI CT-13, *ACI Concrete Terminology,* Farmington Hills, MI, USA: American concrete institute, 2013.

[47] Mazzotta, F., Lantieri, C., Vignali, V., Simone, A., Dondi, G. and Sangiorgi, C., "Performance evaluation of recycled rubber waterproofing bituminous membranes for concrete bridge decks and other surfaces," *Construction and Building Materials,* vol. 136, no. 1, pp. 524-532, 2017.

[48] Hüthwohl, P. and Brilakis, I., "Detecting healthy concrete surfaces," *Advanced Engineering Informatics,* vol. 37, no. 1, pp. 150-162, 2018.

[49] Hassanain, M.A., Al-Hammad, A.M. and Fatayer, F., "Assessment of architectural defects attributed to lack of maintenance feedback to the design team," *Science Review,* vol. 57, no. 2, pp. 132-138, 2014.

[50] Matsui, I., Onda, A., Shinozaki, S., Kyo, E., Nagai, K. and Yuasa, N., "Method of Removing Graffiti from Surface of Concrete," *Key Engineering Materials,* vol. 32, no. 1, pp. 363-370, 2006.

[51] Yao, G., Wei, F., Yang, Y. and Sun, Y., "Deep-learning-based bughole detection for concrete surface image," *Advances in Civil Engineering,* vol. 2019, no. 1, pp. 1-12, 2019.

[52] Li, F., Li, H., Kim, M.K. and Lo, K.C., "Laser scanning based surface flatness measurement using flat mirrors for enhancing scan coverage range," *Remote Sensing,* vol. 13, no. 4, pp. 714/1-17, 2021.

[53] The World Wide Web Consortium (W3C), "Magnetometer: W3c working draft," 2019. [Online]. Available: https://www.w3.org/TR/magnetometer/. [Accessed May 2020].

[54] De Carufel, S., "Understanding Relative Humidity in Concrete," Giatec Scientific Inc., 2019. [Online]. Available: https://www.giatecscientific.com/education/understanding-relative-humidity-in-concrete/. [Accessed May 2021].

[55] Azarsa, P., Gupta, R. and Biparva, A., "Detection and characterization of surface cracks and defects in concrete structures using various NDTs," in *Proceedings of the CSCE Conference on Leadership in Sustainable Infrastructur*, Vancouver, BC, Canada, 2017.

[56] Silva, C.D., Coelho, F., de Brito, J., Silvestre, J. and Pereira, C., "Inspection, diagnosis, and repair system for architectural concrete surfaces," *Journal of Performance of Constructed Facilities,* vol. 31, no. 5, pp. 04017035/1-11, 2017.

[57] Folagbade, S.O., "Initial Surface Absorption of Cement Combination Concretes Containing Portland Cement, Fly Ash, Silica Fume and Metakaolin," *nternational Journal of Sustainable Construction Engineering and Technology,* vol. 8, no. 2, pp. 46-56, 2017.

[58] Yodsudjai, W. and Pattarakittam, T., " Factors influencing half-cell potential measurement and its relationship with corrosion level," *Measurement,* vol. 104, no. 1, pp. 159-168, 2017.

[59] Maksymowicz, M., Cruz, P.J., Bien, J. and Helmerich, R., "Concrete railway bridges: Taxonomy of degradation mechanisms and damages identified by NDT methods," in *Proceedings of the International Conference on Bridge Maintenance, Safety and Management*, Porto, Portugal, 2006.

[60] Silva, C., Coelho, F., de Brito, J., Silvestre, J. and Pereira, C., "Statistical survey on inspection, diagnosis, and repair of architectural concrete surfaces," *Journal of Performance of Constructed Facilities,* vol. 31, no. 6, pp. 04017097/1-9, 2017.

[61] Djelal, C., Vanhove, Y., Azzi, A. and Madec, O., "Recommendation for concrete mix design to prevent bleed channels on diaphragm walls," *European Journal of Environmental and Civil Engineering,* vol. 0, no. 0, pp. 1-13, 2020.

[62] Yehia, S., Qaddoumi, N., Farrag, S. and Hamzeh, L., "Investigation of concrete mix variations and environmental conditions on defect detection ability using GPR," *NDT & E International,* vol. 65, no. 1, pp. 35-46, 2014.

[63] Nassif, A.Y. and Petrou, M.F., "Influence of cold weather during casting and curing on the stiffness and strength of concrete," *Construction and building materials,* vol. 44, no. 1, pp. 161-167, 2013.

[64] Almusallam, A.A., "Effect of environmental conditions on the properties of fresh and hardened concrete," *Cement and Concrete composites,* vol. 23, no. 4-5, pp. 353-361, 2001.

[65] Eghtesadi, S. and Nokken, M., "Effect of cracking and improper consolidation as important concrete defects on water absorption and electrical conductivity," *Journal of Materials in Civil Engineering,* vol. 29, no. 11, pp. 04017201/1-8, 2017.

[66] Vazquez, E.G., Haddad, A.N., Qualharini, E.L., Alves, L.A. and Féo, I.A., "Pathologies in Reinforced Concrete Structures," in *Sustainable Construction*, Singapore, Springer, 2016, pp. 213-228.

[67] Tran, Q.H., Han, D., Kang, C., Haldar, A. and Huh, J., "Effects of ambient temperature and relative humidity on subsurface defect detection in concrete structures by active thermal imaging," *Sensors,* vol. 17, no. 8, pp. 1718/1-8, 2017.

[68] Felicetti, R., " Assessment methods of fire damages in concrete tunnel linings," *Fire Technology,* vol. 49, no. 2, pp. 509-529, 2013.

[69] Chen, J., Diao, B., He, J., Pang, S. and Guan, X., "Equivalent surface defect model for fatigue life prediction of steel reinforcing bars with pitting corrosion," *International Journal of Fatigue,* vol. 10, no. 1, pp. 53-161, 2018.

[70] Riding, K.A., Peterman, R.J., Guthrie, S., Brueseke, M., Mosavi, H., Daily, K. and Risovi-Hendrickson, W., "Environmental and track factors that contribute to abrasion damage," in *Proceedings of Joint Rail Conference on Railroad Infrastructure Engineering*, Pittsburgh, Pennsylvania, USA, 2018.

[71] Yuan, Y., Han, W., Xu, X., Wang, J. and Sun, J., "Permit checking of overloaded customized transport vehicle based on serviceability limit state reliability of concrete bridges," *Advances in Structural Engineering,* vol. 24, no. 5, pp. 884-896, 2021.

[72] Tahir, R.M., "Building Component Defects Due to Land Settlement: A Case Study of Miri Industrial Training Institute," *Journal of Industry, Engineering and Innovation,* vol. 1, no. 1, pp. 1-8, 2019.

[73] Yamov, V. and Belyaeva, Z., "Analysis of main causes of defects of multi-storey reinforced concrete frame buildings," in *Proceedings of MATEC Web of Conferences on Mechanical and Materials Engineering*, EDP Sciences, Les Ulis, France, 2019.

[74] Pereira, C., de Brito, J. and Silvestre, J.D., "Harmonised classification of the causes of defects in a global inspection system: Proposed methodology and analysis of fieldwork data," *Sustainability,* vol. 12, no. 14, pp. 5564/1-27, 2020.

[75] Chen, T.T., "Factors in bridge failure, inspection, and maintenance," *Journal of Performance of Constructed Facilities,* vol. 31, no. 9, pp. 04017070/1-11, 2017.

[76] Muhammad, N.Z., Keyvanfar, A., Majid, M.Z.A., Shafaghat, A. and Mirza, J., "Waterproof performance of concrete: A critical review on implemented approaches," *Construction and Building Materials,* vol. 101, no. 1, pp. 80-90, 2015.

[77] Zhang H, Liao L, Zhao R, Zhou J, Yang M, Xia R., "The Non-Destructive Test of Steel Corrosion in Reinforced Concrete Bridges Using a Micro-Magnetic Sensor," *Sensors,* vol. 16, no. 9, pp. 1439/1-9, 2016.

[78] Louhi Kasahara, J.Y., Yamashita, A. and Asama, H., "Acoustic inspection of concrete structures using active weak supervision and visual information," *Sensors,* vol. 20, no. 3, pp. 629/1-15, 2020.

[79] Gao, Y. and Sun, H., "Influence of initial defects on crack propagation of concrete under uniaxial compression," *Building Materials,* vol. 227, no. 1, p. 122361, 2021.

[80] International Organization for Standardization, *ISO 16311-1, Maintenance and repair of concrete structures,* Geneva, Switzerland: ISO, 2014.

[81] Abdelkhalek, S. and Zayed, T., "Comprehensive inspection system for concrete bridge deck application: Current situation and future needs," *Journal of Performance of Constructed Facilities,* vol. 34, no. 5, pp. 03120001/1-16, 2020.

[82] Guyer J.P., An Introduction to Thin Repairs of Concrete, Sacramento, California,Usa: Guyer Partners, 2020.

[83] Yufeng, S. and Enqiang, Z., "Summary of research on pretreatment methods of recycled aggregate," in *Proceedings of E3S Web of Conferences on Ecological Environment Protection and Pollution Control*, EDP Sciences, Les Ulis, France, 2021.

[84] Fay, K.F., Guide to Concrete Repair, Denver, Colorado, USA: US Department of the Interior, Bureau of Reclamation, Technical Service Center, 2015.

[85] Al-Ostaz A., *Diagnostic evaluation and repair of deteriorated concrete bridges,* Mississippi, USA: Department of Civil Engineering-University of Mississippi, 2004.

[86] Benmokrane, B., Robert, M. and Youssef, T., "Reinforcement of concrete using fibre-reinforced polymer composites," in *Durability of composites for civil structural applications*, Sawston, United Kingdom, Woodhead Publishing, 2007, pp. 225-246.

[87] Kang, S.H., Hong, S.G. and Moon, J., " Importance of drying to control internal curing effects on field casting ultra-high performance concrete," *Cement and concrete research,* vol. 108, no. 1, pp. 20-30, 2018.

[88] Safan, M.A., Etman, Z.A. and Konswa, A., "Evaluation of polyurethane resin injection for concrete leak repair," *Studies in Construction Materials,* vol. 11, no. 1, pp. e00307/1-12, 2019.

[89] Naidenov, V. and Mironova, M., "Innovative hybrid fiber-reinforced shotcrete for thin repairing concrete overlays," *Innovations,* vol. 8, no. 2, pp. 73-78, 2020.

[90] Aslani, F. and Nejadi, S., "Mechanical properties of conventional and self-compacting concrete: An analytical study," *Construction and Building Materials,* vol. 36, no. 1, pp. 330-347, 2012.

[91] Cheng, Y., Liu, S., Zhu, B., Liu, R. and Wang, Y., "Preparation of preplaced aggregate concrete and experimental study on its strength," *Construction and Building Materials,* vol. 229, no. 1, pp. 116847/1-11, 2019.

[92] Wang, M., Wang, R., Yao, H., Farhan, S., Zheng, S., Wang, Z., Du, C. and Jiang, H., "Research on the mechanism of polymer latex modified cement," *Construction and Building Materials,* vol. 111, no. 1, pp. 710-718, 2016.

[93] Aggarwal, L.K., Thapliyal, P.C. and Karade, S.R., "Properties of polymer-modified mortars using epoxy and acrylic emulsions. Construction and Building Materials," *Construction and Building Materials,* vol. 21, no. 2, pp. 379-383, 2007.

[94] Woodson, R. D., "Chapter 6 - Materials and methods for repair and rehabilitation," in *Concrete structures protection, repair and rehabilitation*, Oxford, United Kingdom, Butterworth-Heinemann, 2009, pp. 61-81.

[95] Tayeh, B.A., Maqsoom, A., Aisheh, Y.I.A., Almanassra, M., Salahuddin, H. and Qureshi, M.I., "Factors affecting defects occurrence in the construction stage of residential buildings in Gaza Strip," *SN Applied Sciences,* vol. 2, no. 2, pp. 1-12, 2020.

[96] Kim, M.K., Cheng, J.C., Sohn, H. and Chang, C.C., "A framework for dimensional and surface quality assessment of precast concrete elements using BIM and 3D laser scanning," *Automation in Construction,* vol. 1, no. 49, pp. 225-238, 2015.

[97] Ding, L., Li, K., Zhou, Y. and Love, P.E., "An IFC-inspection process model for infrastructure projects: Enabling real-time quality monitoring and control," *Automation in Construction,* vol. 1, no. 84, pp. 96-110, 2017.

[98] Boukamp, F. and Akinci, B., "Automated processing of construction specifications to support inspection and quality control," *Automation in Construction,* vol. 17, no. 1, pp. 90-106, 2007.

[99] De Melo, R.R.S., Costa, D.B., Alvares, J.S. and Irizarry, J., "Applicability of unmanned aerial system (UAS) for safety inspection on construction sites," *Safety Science,* vol. 1, no. 98, pp. 174-185, 2017.

[100] Phung, M.D., Quach, C.H., Dinh, T.H. and Ha, Q., "Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection," *Automation in Construction,* vol. 1, no. 81, pp. 25-33, 2017.

[101] Oh, J.K., Jang, G., Oh, S., Lee, J.H., Yi, B.J., Moon, Y.S., Lee, J.S. and Choi, Y., "Bridge inspection robot system with machine vision," *Automation in Construction,* vol. 18, no. 7, pp. 929-941, 2009.

[102] Bolourian, N. and Hammad, A., "LiDAR-equipped UAV path planning considering potential locations of defects for bridge inspection," *Automation in Construction,* vol. 1, no. 117, pp. 103250/1-16, 2019.

[103] Lundeen, K.M., Kamat, V.R., Menassa, C.C. and McGee, W., "Autonomous motion planning and task execution in geometrically adaptive robotized construction work," *Automation in Construction,,* vol. 1, no. 100, pp. 24-45, 2019.

[104] Freimuth, H., Müller, J., & Konig, M., "Simulating and Executing UAV-Assisted Inspections on Construction Sites," in *Proceedings of the 34th ISARC International Symposium on Automation and Robotics in Construction*, Taipeh, Taiwan, 2017.

[105] Bortolini, R. and Forcada, N., "Building inspection system for evaluating the technical performance of existing buildings," *Journal of Performance of Constructed Facilities,* vol. 32, no. 5, pp. 04018073/1-14, 2018.

[106] Metni, N. and Hamel, T., "A UAV for bridge inspection: visual servoing control law with orientation limits," *Automation in Construction,* vol. 17, no. 1, pp. 3-10, 2007.

[107] Chen, W., Yabuki, N., Fukuda, T., Michikawa, T., & Motamedi, A., "Development of product model for harbor structures degradation," in *Proceedings of the 2nd International Conference on Civil and Building Engineering Informatics (ICCBEI)*, Tokyo, Japan, 2015.

[108] Aruga, T., Yabuki, N., "Application of a Product Model of Degradation for Civil Engineering Structures Management," *Journal of Japan Society of Civil Engineers, Ser.F3 (Civil Engineering Informatics),* vol. 69, no. 2, pp. 71-81, 2013.

[109] Hammad, A., Motamedi, A., Yabuki, N., Taher, A. and Bahreini, F., "Towards unified ontology for modeling lifecycle inspection and repair information of civil infrastructure systems," in *Proceedings of the 17th ICCCBE International Conference on Computing in Civil and Building Engineering*, Tampere, Finland, 2018.

[110] Calisi, D., Farinelli, A., Iocchi, L. and Nardi, D., "Multi-objective exploration and search for autonomous rescue robots," *Journal of Field Robotics,* vol. 24, no. 8-9, pp. 763-777, 2007.

[111] Zverovich, V., Mahdjoubi, L., Boguslawski, P. and Fadli, F., "Analytic prioritization of indoor routes for search and rescue operations in hazardous environments," *Computer-Aided Civil and Infrastructure Engineering,* vol. 32, no. 9, pp. 727-747, 2017.

[112] Delmerico, J., Mueggler, E., Nitsch, J. and Scaramuzza, D., "Active autonomous aerial exploration for ground robot path planning," *IEEE Robotics and Automation Letters,* vol. 2, no. 2, pp. 664-671, 2017.

[113] Miyazawa, K., "Fire robots developed by the Tokyo Fire Department," *Advanced Robotics,* vol. 16, no. 6, pp. 553-556, 2002.

[114] Cheng, M.Y., Chiu, K.C., Hsieh, Y.M., Yang, I.T., Chou, J.S. and Wu, Y.W., "BIM integrated smart monitoring technique for building fire prevention and disaster relief," *Automation in Construction,* vol. 1, no. 84, pp. 14-30, 2017.

[115] Equipmentjournal, "Boston Dynamics' Spot Robot is finding a home in construction," 21 Nov 2019. [Online]. Available: https://www.equipmentjournal.com/tech-news/boston-dynamics-spot-robot-is-finding-a-home-in-construction/. [Accessed 20 Jan 2020].

[116] Feng, C., Xiao, Y., Willette, A., McGee, W. and Kamat, V.R., "Vision guided autonomous robotic assembly and as-built scanning on unstructured construction sites," *Automation in Construction,* vol. 1, no. 59, pp. 128-138, 2015.

[117] Lundeen, K.M., Kamat, V.R., Menassa, C.C. and McGee, W., "Scene understanding for adaptive manipulation in robotized construction work," *Automation in Construction,* vol. 1, no. 82, pp. 16-30, 2017.

[118] Carra, G., Argiolas, A., Bellissima, A., Niccolini, M. and Ragaglia, M., "Robotics in the construction industry: state of the art and future opportunities," in *Proceedings of the 35th ISARC on Automation and Robotics in Construction*, Berlin, Germany, 2018.

[119] Montero, R., Victores, J.G., Martinez, S., Jardón, A. and Balaguer, C., "Past, present and future of robotic tunnel inspection," *Automation in Construction,* vol. 1, no. 59, pp. 99-112, 2015.

[120] Dormehl, L., "98 percent of construction projects go over budget. These robots could fix that," 26 Jan 2018. [Online]. Available: https://www.digitaltrends.com/cool-tech/doxel-construction-monitoring-robots/. [Accessed 20 Jan 2020].

[121] BostonDynamics, "Spot," 24 Sep 2019. [Online]. Available: https://www.bostondynamics.com/spot. [Accessed 20 Jan 2020].

[122] Pham, N.H., La, H.M., Ha, Q.P., Dang, S.N., Vo, A.H. and Dinh, Q.H., "Visual and 3D mapping for steel bridge inspection using a climbing robot," in *Proceedings of the 33rd ISARC on Automation and Robotics in Construction*, Auburn, AL, USA, 2016.

[123] Cobalt robotics, "The first security industry solution to allow robots to open doors," 04 Mar 2019. [Online]. Available: https://cobaltrobotics.com/2019/03/cobalt-robotics-announces-door-integration-solution/. [Accessed 30 jan 2020].

[124] DENSO robotics, 2020. [Online]. Available: https://www.densorobotics.com/. [Accessed 06 May 2020].

[125] Paul, G., Webb, S., Liu, D. and Dissanayake, G., "Autonomous robot manipulator-based exploration and mapping system for bridge maintenance," *Robotics and Autonomous Systems,* vol. 59, no. 7-8, pp. 543-554, 2011.

[126] PiBorg, "DiddyBorg v2 - Build Instructions," 25 Oct 2017. [Online]. Available: https://www.piborg.org/blog/diddyborg-v2-build-instructions. [Accessed 5 Jul 2020].

[127] Peel, H., Luo, S., Cohn, A.G. and Fuentes, R., "Localisation of a mobile robot for bridge bearing inspection," *Automation in Construction,* vol. 1, no. 94, pp. 244-256, 2018.

[128] iRobot Corporation, "Roomba® Robot Vacuums," May 2020. [Online]. Available: https://www.irobot.com/roomba/s-series. [Accessed 31 Jan 2020].

[129] Croteau, A. and Duguay, N., "The Maski underwater robot: Technology, field experience and benefits," in *Proceedings of 1st IEEE International Conference on Applied Robotics for the Power Industry*, Montreal, QC, Canada, 2010.

[130] Ackerman, E., "Honda Unveils Prototype E2-DR Disaster Response Robot," Oct 02 2017. [Online]. Available: https://spectrum.ieee.org/automaton/robotics/humanoids/iros-2017-honda-unveils-prototype-e2dr-disaster-response-robot. [Accessed 24 01 2020].

[131] Beetz, M., Mösenlechner, L., & Tenorth, M., "CRAM—A Cognitive Robot Abstract Machine for everyday manipulation in human environments," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, 2010.

[132] Tenorth, M., & Beetz, M., "KnowRob—knowledge processing for autonomous personal robots," in *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*, St. Louis, MO, USA, 2009.

[133] Paull, L., Severac, G., Raffo, G.V., Angel, J.M., Boley, H., Durst, P.J., Gray, W., Habib, M., Nguyen, B., Ragavan, S.V. and Saeedi, S.,, "Towards an Ontology for Autonomous Robots," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Portugal, 2012.

[134] Tenorth, M., Kunze, L., Jain, D., & Beetz, M., "Knowrob-map-knowledge-linked semantic object maps," in *Proceedings of the 10th IEEE-RAS International Conference on Humanoid Robots*, Nashville, TN, USA, 2010.

[135] Hamledari, H., Rezazadeh Azar, E. and McCabe, B., "IFC-based development of as-built and as-is BIMs using construction and facility inspection data: Site-to-BIM data transfer automation," *Journal of Computing in Civil Engineering,* vol. 32, no. 2, pp. 04017075/1-15, 2018.

[136] Wang, Q., Kim, M. K., Cheng, J. C., and Sohn, H., "Automated quality assessment of precast concrete elements with geometry irregularities using terrestrial laser scanning," *Automation in Construction,* vol. 1, no. 68, pp. 170-182, 2016.

[137] Wang, C., Chi, W., Sun, Y. and Meng, M.Q.H., "Autonomous robotic exploration by incremental road map construction," *IEEE Transactions on Automation Science and Engineering,* vol. 16, no. 4, pp. 1720-1731, 2019.

[138] Freimuth, H., and Konig, M., "Planning and executing construction inspections with unmanned aerial vehicles," *Automation in Construction,* vol. 96, no. 1, pp. 540-553, 2018.

[139] Menendez, E., Victores, J.G., Montero, R., Martínez, S. and Balaguer, C., "Tunnel structural inspection and assessment using an autonomous robotic system," *Automation in Construction,* vol. 1, no. 87, pp. 117-126, 2018.

[140] Rea, P., and Ottaviano, E., "Design and development of an Inspection Robotic System for indoor applications," *Robotics and Computer-Integrated Manufacturing,* vol. 49, no. 1, pp. 143-151, 2018.

[141] Lim, R.S., La, H.M. and Sheng, W., "A robotic crack inspection and mapping system for bridge deck maintenance," *IEEE Transactions on Automation Science and Engineering,* vol. 11, no. 2, pp. 367-378, 2014.

[142] La, H. M., Dinh, T. H., Pham, N. H., Ha, Q. P., and Pham, A. Q, "Automated robotic monitoring and inspection of steel structures and bridges," *Robotica,* vol. 37, no. 5, pp. 947-967, 2019.

[143] Adan, A., Prieto, S. A., Quintana, B., Prado, T., and Garcia, J., "An autonomous thermal scanning system with which to obtain 3D thermal models of buildings," in *Proceedings of the 35th CIB W78 Conference on IT in Design, Construction, and Management*, Chicago, IL, USA, 2019.

[144] Saigol, Z., Ridder, B., Wang, M., Dearden, R., Fox, M., Hawes, N., Lane, D.M. and Long, D., "Efficient search for known objects in unknown environments using autonomous indoor robots," in *Proceedings of the IROS Workshop on Task Planning for Intelligent Robots in Service and Manufacturing*, Hamburg, Germany, 2015b.

[145] Wang, J., and Luo, C., "Automatic Wall Defect Detection Using an Autonomous Robot: A Focus on Data Collection," in *Proceedings of the ASCE International Conference on Computing in Civil Engineering: Data, Sensing, and Analytics*, Reston, VA, 2019.

[146] Sutter, B., Lelevé, A., Pham, M.T., Gouin, O., Jupille, N., Kuhn, M., Lulé, P., Michaud, P. and Rémy, P., "A semi-autonomous mobile robot for bridge inspection," *Automation in Construction,* vol. 1, no. 91, pp. 111-119, 2018.

[147] Roca, D., Laguela, S., Diaz-Vilarino, L., Armesto, J. and Arias, P., "Low-cost aerial unit for outdoor inspection of building facades," *Automation in Construction,* vol. 1, no. 36, pp. 128-135, 2013.

[148] Yu, S. N., Jang, J. H., and Han, C. S., "Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel," *Automation in Construction,* vol. 16, no. 9, pp. 255-261, 2007.

[149] Freimuth, H., & Konig, M., "A framework for automated acquisition and processing of As-built data with autonomous unmanned aerial vehicles," *Sensors,* vol. 19, no. 20, pp. 4513/1-22, 2019.

[150] Bosche, F., Ahmed, M., Turkan, Y., Haas, C.T. and Haas, R., "The value of integrating Scan-to-BIM and Scan-vs-BIM techniques for construction monitoring using laser scanning and BIM: The case of cylindrical MEP components," *Automation in Construction,* vol. 1, no. 49, pp. 201-213, 2015.

[151] Xiong, X., Adan, A., Akinci, B. and Huber, D., "Automatic creation of semantically rich 3D building models from laser scanner data," *Automation in Construction,* vol. 1, no. 31, pp. 325-337, 2013.

[152] Krawczyk, J.M., Mazur, A.M., Sasin, T. and Stokłosa, A.W., "Infrared building inspection with unmanned aerial vehicles," *Prace Instytutu Lotnictwa,* vol. 3, no. 240, pp. 32-48, 2015.

[153] Eastman, C.M., Eastman, C., Teicholz, P., Sacks, R. and Liston, K., BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors, 2nd ed., John Wiley & Sons, 2011.

[154] buildingSMART, "Industry Foundation Classes Release 4.1," 2018. [Online]. Available: https://standards.buildingsmart.org/IFC/RELEASE/IFC4_1/FINAL/HTML/. [Accessed 1 JUL. 2020].

[155] Belsky, M., Sacks, R., & Brilakis, I., "Semantic enrichment for building information modeling," *Computer-Aided Civil and Infrastructure Engineering,* vol. 31, no. 4, pp. 261-274, 2016.

[156] buildingSMART, "ifcDoc Tool Summary," 2018. [Online]. Available: http://www.buildingsmart-tech.org/specifications/specification-tools/ifcdoc-tool. [Accessed 21 Dec. 2018].

[157] Abudayyeh, O. and Al-Battaineh, H.T., "As-built information model for bridge maintenance," *Journal of Computing in Civil Engineering,* vol. 17, no. 2, pp. 105-112, 2003.

[158] Akinci, B. and Boukamp, F., "Representation and integration of as-built information to IFC based product and process models for automated assessment of as-built conditions," in *Proceedings of the 19th ISARC International Symposium on Automation and Robotics in Construction*, Washington, USA, 2003.

[159] P. Tang, H. M. Aktan and J. S. Polasek, "Integrating sensing technology and Building Information Modeling into a construction engineering curriculum," in *Proceedings of 119th ASEE Annual Conference and Exposition on BIM and Virtual Construction Environments*, San Antonio, Texas, 2012.

[160] Kensek, K. M., "Integration of environmental sensors with BIM: case studies using Arduino, Dynamo, and the Revit API," CSIC - Instituto de Ciencias de la Construcción Eduardo Torroja (IETCC), Madrid, Spain, 2014.

[161] Zhang, J., Seet, B. C., Lie, T. T. , "Building information modelling for smart built environments," *Buildings,* vol. 5, no. 1, pp. 100-115, 2015.

[162] Teicholz, P. ed., BIM for facility managers, New Jersey, USA: John Wiley & Sons, 2013.

[163] Chang, K. M., Dzeng, R. J., & Wu, Y. J., "An automated IoT visualization BIM platform for decision support in facilities management," *Applied sciences,* vol. 8, no. 7, p. 1086, 2018.

[164] Schlueter, A., Geyer, P., and Cisar, S, "Analysis of georeferenced building data for the identification and evaluation of thermal microgrids," in *Proceedings of the IEEE on Optimization for Data-Driven Learning and Control*, New Jersey, USA, 2016.

[165] Lin, Y. H., Liu, Y. S., Gao, G., Han, X. G., Lai, C. Y., & Gu, M. , "The IFC-based path planning for 3D indoor spaces," *Advanced Engineering Informatics,* vol. 27, no. 2, pp. 189-205, 2013.

[166] Wong, M.O. and Lee, S., "A Technical Review on Developing BIM-Oriented Indoor Route Planning," in *Proceedings of the ASCE International Conference on In Computing in Civil Engineering: Visualization, Information Modeling, and Simulation*, Reston, VA, 2019.

[167] buildingSMART, "IFC2x Edition 3," 2007. [Online]. Available: http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/. [Accessed 10 Jun. 2020].

[168] Boguslawski, P., Mahdjoubi, L., Zverovich, V. and Fadli, F., "Automated construction of variable density navigable networks in a 3D indoor environment for emergency response," *Automation in Construction,* vol. 1, no. 72, pp. 115-128, 2016.

[169] Yan, W., Culp, C., & Graf, R., "Integrating BIM and gaming for real-time interactive architectural visualization," *Automation in Construction,* vol. 20, no. 4, pp. 446-458, 2011.

[170] Mitka, E., Gasteratos, A., Kyriakoulis, N., & Mouroutsos, S. G., "Safety certification requirements for domestic robot," *Safety science,* vol. 50, no. 9, pp. 1888-1897, 2012.

[171] Boguslawski, P., Mahdjoubi, L., Zverovich, V. and Fadli, F., "A dynamic approach for evacuees' distribution and optimal routing in hazardous environments," *Automation in Construction,* vol. 1, no. 91, pp. 11-21, 2018.

[172] Strug, B. and Ślusarczyk, G., "Reasoning about accessibility for disabled using building graph models based on BIM/IFC," *Visualization in Engineering,* vol. 5, no. 1, pp. 1-12, 2017.

[173] Ivanov, R., "An approach for developing indoor navigation systems for visually impaired people using Building Information Modeling," *Journal of Ambient Intelligence and Smart Environments,* vol. 9, no. 4, pp. 449-467, 2017.

[174] Park, J., Chen, J. and Cho, Y.K., "Self-corrective knowledge-based hybrid tracking system using BIM and multimodal sensors," *Advanced Engineering Informatics,* vol. 1, no. 32, pp. 126-138, 2017.

[175] Eastman, C., Lee, J. M., Jeong, Y. S., & Lee, J. K., "Automatic rule-based checking of building designs," *Automation in Construction,* vol. 18, no. 8, pp. 1011-1033, 2009.

[176] Taneja, S., Akinci, B., Garrett, J.H., Soibelman, L. and East, B., "Transforming IFC-based building layout information into a geometric topology network for indoor navigation assistance," in *Proceedings of the ASCE International Workshop on Computing in Civil Engineering*, Miami, Florida, United States, 2011.

[177] Khalili, A. and Chua, D.H., "IFC-based graph data model for topological queries on building elements," *Journal of Computing in Civil Engineering,* vol. 29, no. 3, pp. 04014046/1-14, 2015.

[178] Nguyen, T.H., Oloufa, A.A. and Nassar, K., "Algorithms for automated deduction of topological information," *Automation in Construction,* vol. 14, no. 1, pp. 59-70, 2005.

[179] Rasmussen, M.H., Lefrançois, M., Schneider, G.F. and Pauwels, P., "BOT: The building topology ontology of the W3C linked building data group," *Semantic Web,* vol. 12, no. 1, pp. 1-9, 2021.

[180] Hassanain, M.A., Froese, T.M. and Vanier, D.J., "Implementation of a distributed, model-based integrated asset management system," *Journal of Information Technology in Construction (ITcon),* vol. 8, no. 10, pp. 119-134, 2003.

[181] Davila Delgado, J. M., N. De Battista, I. Brilakis, and C. Middleton., "Design and data modelling of fibre optic systems to monitor reinforced concrete structural elements," in *Proceedings of Construction Research Congress on Old and New Construction Technologies* , San Juan, Puerto Rico , 2016.

[182] Chen, J., Bulbul, T., Taylor, J. E., and Olgun, G., "A case study of embedding real-time infrastructure sensor data to BIM," in *Proceedings of Construction Research Congress on Construction in a Global Network*, Atlanta, Georgia, 2014.

[183] Hammad, A., Vahdatikhaki, F., Albahri, A., Salimi, S., Moghaddam, D., Jahromi, A.,, "Framework for life-cycle infrastructure information modeling and management," in *Proceedings of Annual Conference of the Canadian Society for Civil Engineering on Construction*, Montreal, Canada, 2013.

[184] Mailhot G, Busuioc S., "Application of long range 3D laser scanning for remote sensing and monitoring of complex bridge structures," in *Proceedings of the 7th International Conference of Short & Medium*, Montreal, Canada, 2006.

[185] Park, C. S., Lee, D. Y., Kwon, O. S., & Wang, X., "A framework for proactive construction defect management using BIM, augmented reality and ontology-based data collection template," *Automation in Construction,* vol. 1, no. 33, pp. 61-71, 2013.

[186] Aruga, T. and Yabuki, N., "Cooperative Information Management of Degradation of Structures in Operation and Management," in *Proceedings of International Conference on Cooperative Design, Visualization and Engineering*, Springer, Berlin, 2012.

[187] Hammad, A., Zhang, C., Hu, Y. and Mozaffari, E., "Mobile model-based bridge lifecycle management system," *Computer-Aided Civil and Infrastructure Engineering,* vol. 21, no. 7, pp. 530-547, 2006.

[188] Motamedi, A., Yabuki, N. and Fukuda, T., "Extending BIM to include defects and degradations of buildings and infrastructure facilities," in *Proceedings of the 3rd International Conference on Civil and Building Engineering Informatics*, Taipei, Taiwan, 2017.

[189] Ma, L., Sacks, R. and Zeibak-Shini, R., "Information modeling of earthquake-damaged reinforced concrete structures," *Advanced Engineering Informatics,* vol. 29, no. 3, pp. 396-407, 2015.

[190] Tanaka, F., Hori, M., Onosato, M., Date, H. and Kanai, S., "Bridge information model based on IFC standards and web content providing system for supporting an inspection process," in *Proceedings of the 16th International Conference on Computing in Civil and Building Engineering*, Osaka, Japan, 2016.

[191] Tanaka, F., Tsuchida, M., Onosato, M., Date, H., Kanai, S., Hada, Y., Nakao, M., Kobayashi, H., Hasegawa, E., Sugawara, T. and Oyama, T., "Bridge Information Modeling based on IFC for supporting maintenance management of existing bridges," in *Proceedings of the 17th International Conference on Computing in Civil and Building Engineering*, Tampere, Finland, 2018.

[192] Sacks, R., Kedar, A., Borrmann, A., Ma, L., Brilakis, I., Hüthwohl, P., Daum, S., Kattel, U., Yosef, R., Liebich, T. and Barutcu, B.E., "SeeBridge as next generation bridge inspection: overview, information delivery manual and model view definition," *Automation in Construction,* vol. 90, no. 1, pp. 134-145, 2018.

[193] Artus, M. and Koch, C., "State of the art in damage information modeling for RC bridges–A literature review," *Advanced Engineering Informatics,* vol. 46, no. 1, pp. 101171/1-16, 2020.

[194] Hüthwohl, P., Brilakis, I., Borrmann, A. and Sacks, R., "Integrating RC bridge defect information into BIM model," *American Society of Civil Engineers,* vol. 32, no. 3, pp. 04018013/1-14, 2018.

[195] Hamdan, A. and Scherer, R.J., "A generic model for the digitalization of structural damage," in *Proceedings of the Sixth International Symposium on Life-Cycle Civil Engineering*, Ghent, Belgium, 2018.

[196] Hamdan, A.H., Taraben, J., Helmrich, M., Mansperger, T., Morgenthal, G. and Scherer, R.J., "A semantic modeling approach for the automated detection and interpretation of structural damage," *Automation in Construction,* vol. 128, no. 1, pp. 103739/1-17, 2021.

[197] Artus, M. and Koch, C., "Modeling geometry and semantics of physical damages using IFC," in *EG-ICE 2020 Workshop on Intelligent Computing in Engineering*, Universitätsverlag der TU Berlin, 2020.

[198] Artus, M., Alabassy, M.S.H. and Koch, C., "A BIM Based Framework for Damage Segmentation, Modeling, and Visualization Using IFC," *Applied Sciences,* vol. 12, no. 6, pp. 2772/1-11, 2021.

[199] Isailović, D., Stojanovic, V., Trapp, M., Richter, R., Hajdin, R. and Döllner, J., "Bridge damage: Detection, IFC-based semantic enrichment and visualization," *Automation in Construction,* vol. 112, no. 1, pp. 103088/1-22, 2020.

[200] Kasireddy, V., Akinci, B., "Challenges in generation of as-is bridge information model: a case study," in *Proceedings of the International Symposium on Automation and Robotics in Construction*, Oulu, Finland, 2015.

[201] Gruber, T. R., "Toward principles for the design of ontologies used for knowledge sharing," *International Journal of Human-computer Studies,* vol. 43, no. 5-6, pp. 907-928, 1995.

[202] Viljamaa, E., and Peltomaa, I., "Intensified construction process control using information integration," *Automation in Construction,* vol. 39, pp. 126-133, 2014.

[203] Gasevic, D., Djuric, D., & Devedzic, V., Model Driven Engineering and Ontology Development, Berlin, Germany: Springer, 2009.

[204] Thomopoulos, R., Destercke, S., Charnomordic, B., Johnson, I. and Abécassis, J., "An iterative approach to build relevant ontology-aware data-driven models," *Information Sciences,* vol. 221, pp. 452-472, 2013.

[205] El-Gohary, N.M. and El-Diraby, T.E., "Domain ontology for processes in infrastructure and construction," *Journal of Construction Engineering and Management,* vol. 136, no. 7, pp. 730-744, 2010.

[206] Stanford university, "Stanford Center for Biomedical Informatics Research," Protégé, 2019. [Online]. Available: https://protege.stanford.edu/products.php. [Accessed 2 Jan. 2019].

[207] Staab, S., and Maedche, A., "Ontology engineering beyond the modeling of concepts and relations," in *Proceedings of the ECAI Workshop on Ontologies and Problem-Solving Methods*, Berlin, 2000.

[208] Ingh, A., & Anand, P., "State of art in ontology development tools," *International Journal,* vol. 2, no. 7, pp. 96-101, 2013.

[209] El-Diraby, T. E., & Kashif, K. F., "Distributed ontology architecture for knowledge management in highway construction," *Journal of Construction Engineering and Management,* vol. 131, no. 5, pp. 591-603, 2005.

[210] McGuinness, D. L., and Van Harmelen, F., "OWL web ontology language overview," W3C recommendation 10, 2004.

[211] Terkaj, W., & Sojic, A., "ontology-based representation of IFC EXPRESS rules: An enhancement of the ifcOWL ontology," *Automation in Construction,* vol. 1, no. 57, pp. 188-201, 2015.

[212] buildingSMART, "ifcOWL," 2019. [Online]. Available: https://technical.buildingsmart.org/standards/ifc/ifc-formats/ifcowl/. [Accessed 22 JUL. 2020].

[213] Ding, L. Y., Zhong, B. T., Wu, S., & Luo, H. B., "Construction risk knowledge management in BIM using ontology and semantic web technology," *Safety Science,* vol. 1, no. 87, pp. 202-213, 2016.

[214] Dibley, M., Li, H., Rezgui, Y., & Miles, J., "An ontology framework for intelligent sensor-based building monitoring," *Automation in Construction,* vol. 28, pp. 1-14, 2012.

[215] Adeleke, J. A., & Moodley, D., "An ontology for proactive indoor environmental quality monitoring and control," in *Proceedings of the Annual Research Conference on South African*

Institute of Computer Scientists and Information Technologists, Stellenbosch, South Africa, 2015.

[216] Jung, S., Lee, S. and Yu, J., "Ontological Approach for Automatic Inference of Concrete Crack Cause," *Applied Sciences,* vol. 11, no. 1, pp. 252/1-20, 2021.

[217] Lee, D.Y., Chi, H.L., Wang, J., Wang, X. and Park, C.S., "A linked data system framework for sharing construction defect information using ontologies and BIM environments," *Automation in Construction,* vol. 68, no. 1, pp. 102-113, 2016.

[218] El-Diraby, T. E., "Domain ontology for construction knowledge," *Journal of Construction Engineering and Management ,* vol. 137, no. 7, pp. 768-784, 2012.

[219] Venugopal, M., Eastman, C. M., Teizer, J., Kandil, A., & Hastak, M., "An ontological approach to building information model exchanges in the precast/pre-stressed concrete industry," in *Proceedings of Construction research congress on Construction Challenges in a Flat World*, West Lafayette, Indiana, USA, 2012.

[220] Zeb, J. and Froese, T., "Transaction ontology in the domain of infrastructure management," *Canadian Journal of Civil Engineering,* vol. 39, no. 9, pp. 993-1004, 2012.

[221] Storga, M., Marjanovic, D., and Andreasen, M. M., "Relationships between the concepts in the design ontology," in *Proceedings of 16th International Conference on Engineering Design*, Paris, France, 2007.

[222] Hamdan, A.H., Bonduel, M. and Scherer, R.J., "An ontological model for the representation of damage to constructions," in *Proceedings of the 7th Linked Data in Architecture and Construction Workshop*, London, United Kingdom, 2019.

[223] Pauwels, P., "Building Element Ontology," 2018. [Online]. Available: https://pi.pauwel.be/voc/buildingelement/index-en.html. [Accessed 26 Feb 2021].

[224] Pauwels, P., "Distribution Element Ontology," 2019. [Online]. Available: https://pi.pauwel.be/voc/distributionelement/index-en.html. [Accessed 20 Dec 2021].

[225] Rasmussen, M., Lefrançois, M., Bonduel, M., Hviid, C. and Karlshøj, J., "PM: An ontology for describing properties that evolve over time," in *Proceedings of the 6th Linked Data in Architecture and Construction Workshop*, London, United Kingdom, 2018.

[226] Wagner, A., Bonduel, M., Pauwels, P. and Uwe, R., "Relating geometry descriptions to its derivatives on the web," in *European Conference on Computing in Construction*, Chania, Crete, Greece, 2019.

[227] Bonduel, M., Wagner, A., Pauwels, P., Vergauwen, M. and Klein, R., "Including widespread geometry formats in semantic graphs using RDF literals," in *Proceeding of European Conference on Computing in Construction*, Chania, Crete, Greece, 2019.

[228] Niknam, M. and Karshenas, S., "A shared ontology approach to semantic representation of BIM data," *Automation in Construction,* vol. 80, no. 1, pp. 22-36, 2017.

[229] Zhong, B., Gan, C., Luo, H., & Xing, X., "Ontology-based framework for building environmental monitoring and compliance checking under BIM environment," *Building and Environment,* vol. 1, no. 141, pp. 127-142, 2018.

[230] Zhu, Y., "Ontology to Support Multi-Objective Integrated Analyses for Sustainable Construction: A Conceptual Framework," in *Ontology in the AEC Industry, Cahapter 4*, Reston, Virginia, American Society of Civil Engineers (ASCE), 2015, pp. 73-95.

[231] Kim, K., Kim, H., Kim, W., Kim, C., Kim, J., & Yu, J., "Integration of ifc objects and facility management work information using Semantic Web," *Automation in Construction,* vol. 87, no. 1, pp. 173-187, 2018.

[232] Wang, N. and Issa, R.R., "Ontology-Based Integration of BIM and GIS for Indoor Routing," in *Proceedings of Construction Research Congress on Computer Applications*, Reston, VA, USA, 2020.

[233] Bourreau, P., Charbel, N., Werbrouck, J., Senthilvel, M., Pauwels, P. and Beetz, J., "Multiple inheritance for a modular BIM," in *In Le BIM et l'évolution des pratiques: Ingénierie et architecture, enseignement et recherche*, Eyrolles, 2020.

[234] Haidegger, T., Barreto, M., Gonçalves, P., Habib, M.K., Ragavan, S.K.V., Li, H., Vaccarella, A., Perrone, R. and Prestes, E., "Applied ontologies and standards for service robots," *Robotics and Autonomous Systems,* vol. 61, no. 11, pp. 1215-1223, 2013.

[235] Kokar, M.M., Matheus, C.J. and Baclawski, K., "Ontology-based situation awareness," *Information Fusion,* vol. 10, no. 1, pp. 83-98, 2009.

[236] Barbera, T., Albus, J., Messina, E., Schlenoff, C. and Horst, J., "How task analysis can be used to derive and organize the knowledge for the control of autonomous vehicles," *Robotics and Autonomous Systems,* vol. 49, no. 1-2, pp. 67-78, 2004.

[237] Open Robotics, "ROS," 2020. [Online]. Available: https://www.ros.org/. [Accessed 07 JUL 2020].

[238] Zhang, J., and Singh, S., "LOAM: Lidar Odometry and Mapping in Real-time," *In Robotics: Science and Systems,* vol. 2, no. 9, pp. 401-416, 2014.

[239] Kostavelis, I. and Gasteratos, A., "Semantic mapping for mobile robotics tasks: A survey," *Robotics and Autonomous Systems,* vol. 1, no. 66, pp. 86-103, 2015.

[240] Bouguerra, A., Karlsson, L. and Saffiotti, A., "Semantic Knowledge-Based Execution Monitoring for Mobile Robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Roma, Italy, 2007.

[241] Chella, A., Cossentino, M., Pirrone, R. and Ruisi, A., "Modeling ontologies for robotic environments," in *Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering*, New York, United States, 2002.

[242] Wood, S., "Representation and purposeful autonomous agents," *Robotics and Autonomous Systems,* vol. 51, no. 2-3, pp. 217-228, 2005.

[243] Jung, H., Bradshaw, J.M., Kulkarni, S., Breedy, M., Bunch, L., Feltovich, P., Jeffers, R., Johnson, M., Lott, J., Suri, N. and Taysom, W., "An ontology-based representation for policy-governed adjustable autonomy," in *Proceedings of the AAAI Spring Symposium on Knowledge Representation and Ontology for Autonomous Systems*, Palo Alto, California, 2004.

[244] Fukuda, H., Mori, S., Kobayashi, Y., Kuno, Y. and Kachi, D., "Object recognition based on human description ontology for service robots," in *Proceedings of the 40th IEEE Annual Conference on Industrial Electronics Society (IECON)*, Dallas, TX, USA, 2014.

[245] Maillot, N.E. and Thonnat, M., "Ontology based complex object recognition," *Image and Vision Computing,* vol. 26, no. 1, pp. 102-113, 2008.

[246] Dasiopoulou, S., Mezaris, V., Kompatsiaris, I., Papastathis, V.K. and Strintzis, M.G., "Knowledge-assisted semantic video object detection," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 15, no. 10, pp. 1210-1224, 2005.

[247] Habib, M.K., Yuta, S., "Map representation of a large in-door environment with path planning and navigation abilities for an autonomous mobile robot with its implementation on a real robot," *Automation in Construction,* vol. 2, no. 2, pp. 155-179, 1993.

[248] Balakirsky, S. and Scrapper, C., "Knowledge representation and planning for on-road driving," *Obotics and Autonomous Systems,* vol. 49, no. 1-2, pp. 57-66, 2004.

[249] Provine, R., Schlenoff, C., Balakirsky, S., Smith, S. and Uschold, M., "Ontology-based methods for enhancing autonomous vehicle path planning," *Robotics and Autonomous Systems,* vol. 49, no. 1-2, pp. 123-133, 2004.

[250] Schlenoff, C., "Linking sensed images to an ontology of obstacles to aid in autonomous driving," in *In Proceedings of the 18th National Conference on Artificial Intelligence: Workshop on Ontologies for the Semantic Web*, Edmonton, Canada, 2002.

[251] Schlenoff, C., Balakirsky, S., Uschold, M., Provine, R. and Smith, S., "Using ontologies to aid navigation planning in autonomous vehicles," *The knowledge Engineering Review,* vol. 18, no. 3, pp. 243-255, 2003.

[252] Schlenoff, C. and Messina, E., "A robot ontology for urban search and rescue," in *Proceedings of the ACM Workshop on Research in knowledge Representation for Autonomous Systems*, Bremen, Germany, 2005.

[253] Hummel, B., Thiemann, W. and Lulcheva, I., "Description logic for vision-based intersection understanding," in *Proceedings of the International Conference on Cognitive Systems with Interactive Sensors (COGIS07)*, Stanford University, Palo Alto, CA, USA, 2007.

[254] Rashidi, M., Mohammadi, M., Sadeghlou Kivi, S., Abdolvand, M.M., Truong-Hong, L. and Samali, B., "A decade of modern bridge monitoring using terrestrial laser scanning: Review and future directions," *Remote Sensing,* vol. 12, no. 22, pp. 3796/1-34, 2020.

[255] Truong-Hong, L., Falter, H., Lennon, D. and Laefer, D.F., "Framework for bridge inspection with laser scanning," in *Proceedings of the EASEC-14 Conference on Structural Engineering and Construction*, Minh City, Vietnam, 2016.

[256] Tang, P., Akinci, B. and Huber, D., "Quantification of edge loss of laser scanned data at spatial discontinuities," *Automation in Construction,* vol. 18, no. 8, pp. 1070-1083, 2009.

[257] Teza, G., Galgaro, A. and Moro, F., "Contactless recognition of concrete surface damage from laser scanning and curvature computation," *NDT & E International,* vol. 42, no. 4, pp. 240-249, 2009.

[258] Olsen, M.J., Kuester, F., Chang, B.J. and Hutchinson, T.C., "Terrestrial laser scanning-based structural damage assessment," *Journal of Computing in Civil Engineering,* vol. 24, no. 3, pp. 264-272, 2010.

[259] Anil, E.B., Akinci, B., Garrett, J.H. and Kurc, O., "Characterization of laser scanners for detecting cracks for post-earthquake damage inspection," in *Proceedings of the International Symposium on Automation and Robotics in Construction*, Montreal, Canada, 2013.

[260] Xu, X. and Yang, H., "Intelligent crack extraction and analysis for tunnel structures with terrestrial laser scanning measurement," *Advances in Mechanical Engineering,* vol. 11, no. 9, pp. 1-7, 2019.

[261] Makuch, M. and Gawronek, P., "3D point cloud analysis for damage detection on hyperboloid cooling tower shells," *Remote Sensing,* vol. 12, no. 10, pp. 1542/1-23, 2020.

[262] Liu, W., Chen, S. and Hauser, E., "LiDAR-based bridge structure defect detection," *Experimental Techniques,* vol. 35, no. 6, pp. 27-34, 2011.

[263] Armesto-González, J., Riveiro-Rodríguez, B., González-Aguilera, D. and Rivas-Brea, M.T., "Terrestrial laser scanning intensity data applied to damage detection for historical buildings," *Journal of Archaeological Science,* vol. 73, no. 12, pp. 3037-3047, 2010.

[264] Valença, J., Puente, I., Júlio, E., González-Jorge, H. and Arias-Sánchez, P., "Assessment of cracks on concrete bridges using image processing supported by laser scanning survey," *Construction and Building Materials,* vol. 146, no. 1, pp. 668-678, 2017.

[265] Kim, M.K., Sohn, H. and Chang, C.C., "Localization and quantification of concrete spalling defects using terrestrial laser scanning," *Journal of Computing in Civil Engineering,* vol. 29, no. 6, p. 04014086, 2015.

[266] Tsai, Y.C.J. and Li, F., "Critical assessment of detecting asphalt pavement cracks under different lighting and low intensity contrast conditions using emerging 3D laser technology," *Journal of Transportation Engineering,* vol. 138, no. 5, pp. 649-656, 2012.

[267] Cabaleiro, M., Lindenbergh, R., Gard, W.F., Arias, P. and Van de Kuilen, J.W.G., "Algorithm for automatic detection and analysis of cracks in timber beams from LiDAR data," *Construction and Building Materials,* vol. 130, no. 1, pp. 41-53, 2017.

[268] Mizoguchi, T., Koda, Y., Iwaki, I., Wakabayashi, H., Kobayashi, Y., Shirai, K., Hara, Y. and Lee, H.S., "Quantitative scaling evaluation of concrete structures based on terrestrial laser scanning," *Automation in Construction,* vol. 35, no. 1, pp. 263-274, 2013.

[269] Nasrollahi, M., Bolourian, N. and Hammad, A., "Concrete surface defect detection using deep neural network based on lidar scanning," in *Proceedings of the CSCE Annual Conference on AI and machine learning*, Montreal, Canada, 2019.

[270] Te, G., Hu, W., Zheng, A. and Guo, Z., "Rgcnn: Regularized graph cnn for point cloud segmentation," in *Proceedings of the 26th ACM International Conference on Multimedia*, New Yor, kNY, USA, 2018.

[271] Erkal, B.G. and Hajjar, J.F., "Laser-based surface damage detection and quantification using predicted surface properties," *Automation in Construction,* vol. 83, no. 1, pp. 285-302, 2017.

[272] Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L. and Bennamoun, M., "Deep learning for 3d point clouds: A survey," *IEEE transactions on pattern analysis and machine intelligence,* vol. 43, no. 12, pp. 4338 - 4364, 2020.

[273] Yang, X., Li, H., Yu, Y., Luo, X., Huang, T. and Yang, X., "Automatic pixel-level crack detection and measurement using fully convolutional network," *Computer-Aided Civil and Infrastructure Engineering,* vol. 33, no. 12, pp. 1090-1109, 2018.

[274] Liu, Z., Cao, Y., Wang, Y. and Wang, W., "Computer vision-based concrete crack detection using U-net fully convolutional networks," *Automation in Construction,* vol. 104, no. 1, pp. 129-139, 2019.

[275] Ali, L., Alnajjar, F., Jassmi, H.A., Gochoo, M., Khan, W. and Serhani, M.A., "Performance evaluation of deep CNN-Based crack detection and localization techniques for concrete structures," *Sensors,* vol. 21, no. 5, pp. 1688/1-22, 2021.

[276] Le, T.T., Nguyen, V.H. and Le, M.V., "Development of deep learning model for the recognition of cracks on concrete surfaces," *Applied Computational Intelligence and Soft Computing,* vol. 2021, no. 1, pp. 1687-9724, 2021.

[277] Vignesh, R., Narenthiran, B., Manivannan, S., Murugan, R.A. and RajKumar, V., "Concrete Bridge Crack Detection Using Convolutional Neural Network," in *Proceedings of the*

International Conference Materials, Design, and Manufacturing for Sustainable Environment, Springer, Singapore, 2021.

[278] Shrestha, A. and Mahmood, A., "Review of deep learning algorithms and architectures," *IEEE Access,* vol. 7, no. 1, pp. 53040-53065, 2019.

[279] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., "Generative adversarial nets," in *Proceedinf of the 28th Conference on Neural Information Processing Systems*, Montreal, CA, 2014.

[280] LeCun, Y. and Bengio, Y., "Pattern recognition and neural networks," in *The Handbook of Brain Theory and Neural Networks*, Cambridge, MA, USA, MIT Press, 1995, pp. 1-24.

[281] Goodfellow, I., Bengio, Y., Courville, A. and Bengio, Y., Deep learning, Cambridge, MA, USA: MIT press, 2016.

[282] LeCun, Y., Generalization and network design strategies, Zurich, Switzerland: Connectionism in perspective Elsevier, 1989.

[283] Su, T., Li, H., Zhang, S. and Li, Y., "Image segmentation using mean shift for extracting croplands from high-resolution remote sensing imagery," *Remote Sensing Letters,* vol. 6, no. 12, pp. 952-961, 2015.

[284] Brock, A., Lim, T., Ritchie, J.M. and Weston, N., "Generative and discriminative voxel modeling with convolutional neural networks," in *arXiv preprint arXiv:1608.04236*, 2016.

[285] Qi, C.R., Su, H., Nießner, M., Dai, A., Yan, M. and Guibas, L.J., "Volumetric and multi-view cnns for object classification on 3d data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, NV, USA, 2016.

[286] Qi, C.R., Su, H., Mo, K. and Guibas, L.J., "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Honolulu, HI, USA, 2017.

[287] Xie, Y., Tian, J. and Zhu, X.X., "A review of point cloud semantic segmentation," in *arXiv preprint arXiv:1908.08854*, 2019.

[288] Armeni, I., Sax, S., Zamir, A.R. and Savarese, S., "Joint 2d-3d-semantic data for indoor scene understanding," in *arXiv preprint arXiv:1702.01105*, 2017.

[289] Qi, C.R., Yi, L., Su, H. and Guibas, L.J., "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Red Hook, NY, United States, 2017.

[290] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X. and Xiao, J., "3D shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Boston, MA, USA, 2015.

[291] Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M. and Savarese, S., "3D semantic parsing of large-scale indoor spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016.

[292] Pierdicca, R., Paolanti, M., Matrone, F., Martini, M., Morbidoni, C., Malinverni, E.S., Frontoni, E. and Lingua, A.M., "Point cloud semantic segmentation using a deep learning framework for cultural heritage," *Remote sensing,* vol. 12, no. 6, pp. 1005/1-23, 2020.

[293] Suarez-Figueroa, M.C., Gomez-Perez, A. and Villazon-Terrazas, B., "How to write and use the ontology requirements specification document," in *in Proceedings of the OTM Confederated International Conferences on the Move to Meaningful Internet Systems*, Springer, Berlin, Heidelberg, 2009.

[294] Fernández-López, M., Gómez-Pérez, A. and Juristo, N., "Methontology: from ontological art towards ontological engineering," in *Proceedings of the AAAI97 Spring Symposiumon on Ontological Engineering*, Stanford, USA, 1997.

[295] Prestes, E., Carbonera, J.L., Fiorini, S.R., Jorge, V.A., Abel, M., Madhavan, R., Locoro, A., Goncalves, P., Barreto, M.E., Habib, M. and Chibani, A., "Towards a core ontology for robotics and automation," *Robotics and Autonomous Systems,* vol. 61, no. 11, pp. 1193-1204, 2013.

[296] Yun, H., Xu, J., Xiong, J. and Wei, M., "A knowledge engineering approach to develop domain ontology," *International Journal of Distance Education Technologies (IJDET),* vol. 9, no. 1, pp. 57-71, 2011.

[297] KBSI, "IDEF – Integrated DEFinition Methods (IDEF)," Knowledge Based Systems, Inc., 2020. [Online]. Available: https://www.idef.com/. [Accessed 18 JUL 2020].

[298] Taher, A., Vahdatikhaki, F., and Hammad, A, "Towards Developing an Ontology for Earthwork Operation," in *Proceedings of the ASCE International Workshop on Computing in Civil Engineering*, Seattle, Washington, USA, 2017.

[299] PCTE, "Crack Measuring Device," PCTE, 2020. [Online]. Available: https://www.pcte.com.au/crack-measuring. [Accessed May 2021].

[300] Perfilov, V.A., Oreshkin, D.V. and Zemlyanushnov, D.Y., "Concrete strength and crack resistance control," *Procedia Engineering,* vol. 150, no. 1, pp. 1474-1478, 2016.

[301] Zhang, J., Cui, X., Li, L. and Huang, D., "Sediment transport and pore clogging of a porous pavement under surface runoff," *Road Materials and Pavement Design,* vol. 18, no. 3, pp. 240-248, 2017.

[302] Bissonnette, B., Vaysburd, A.M. and Fay, K.F., "Best Practices for Preparing Concrete Surfaces Prior to Repairs and Overlays—No MERL 12–17," US Department of the Interior Bureau of Reclamation Technical Service Center, Denver, CO, USA, 2012.

[303] Wei, S., Jiang, Z., Liu, H., Zhou, D. and Sanchez-Silva, M., "Microbiologically induced deterioration of concrete: a review," *Brazilian Journal of Microbiology,* vol. 44, no. 1, pp. 1001-1007, 2013.

[304] Kovler, K. and Chernov, V., "Types of damage in concrete structures," in *Failure, distress and repair of concrete structures*, Sawston, United Kingdom, Woodhead Publishing, 2009, pp. 32-56.

[305] Grantham, M., Mechtcherine, V. and Schneck, U., Concrete Solutions, Leiden, Netherland: CRC Press, 2011.

[306] Alberta infrastructure and transportation, "Alberta Bridge Inspection and Maintenance System - Inspection Manual," Goverment of the province of Alberta, Alberta, Canada, 2008.

[307] New York State Department of Transportation, "Bridge Inspection Manual," Office of structures, New York, USA, 2017.

[308] Yu, J., Thom, J.A. and Tam, A., "Requirements-oriented methodology for evaluating ontologies," *Information Systems,* vol. 34, no. 8, pp. 766-791, 2009.

[309] Gomez-Perez, A., "Towards a framework to verify knowledge sharing technology," *Expert Systems with Applications,* vol. 11, no. 4, pp. 519-529, 1996.

[310] Vrandecic, D., "Ontology evaluation," in *Handbook on Ontologies*, Berlin, Heidelberg, Germany, Springer, 2009, pp. 293-313.

[311] Oxford university, "The Knowledge Representation and Reasoning Group," HermiT OWL Reasoner, 2019. [Online]. Available: http://www.hermit-reasoner.com/. [Accessed JUN 10 2020].

[312] Haghighi, P.D., Burstein, F., Zaslavsky, A. and Arbon, P., "Development and evaluation of ontology for intelligent decision support in medical emergency management for mass gatherings," *Decision Support Systems,* vol. 54, no. 2, pp. 1192-1204, 2013.

[313] Obrst, L., Ceusters, W., Mani, I., Ray, S. and Smith, B., "The evaluation of ontologies," in *Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences*, Boston, MA, USA, Springer, 2007, pp. 139-158.

[314] Motamedi, A., Hammad, A. and Asen, Y., "Knowledge-assisted BIM-based visual analytics for failure root cause detection in facilities management," *Automation in construction,* vol. 43, no. 1, pp. 73-83, 2014.

[315] Pinto, H.S., Gomez-Perez, A. and Martins, J.P., "Some issues on ontology integration," in *Proceedings of IJCAI Workshop on Workshop on Ontologies and Problem Solving Methods: Lessons Learned and Future Trends*, Stockholm, Sweden, 1999.

[316] Choi, N., Song, I.Y. and Han, H., "A survey on ontology mapping," *ACM Sigmod Record,* vol. 35, no. 3, pp. 34-41, 2006.

[317] Noy, N.F., Griffith, N. and Musen, M.A., "Collecting community-based mappings in an ontology repository," in *Proceedings of 7th ISWC International Conference on Semantic Web* , Karlsruhe, Germany, 2008.

[318] Pinto, H.S. and Martins, J.P., "A methodology for ontology integration," in *Proceedings of the 1st international conference on Knowledge capture*, Victoria, British Columbia, Canada, 2001.

[319] Brank, J., Grobelnik, M. and Mladenic, D., "A survey of ontology evaluation techniques," in *In Proceedings of the conference on data mining and data warehouses (SiKDD 2005)*, Citeseer Ljubljana, Slovenia, 2005.

[320] Selvaraj, S. and Choi,, "TKM ontology integration and visualization," in *Proceedings of the 3rd International Conference on Software Engineering and Information Management*, Sydney, Australia, 2020.

[321] Klingbeil, E., Saxena, A. and Ng, A.Y., "Learning to open new doors," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, 2010.

[322] B. Richardson, Defects and Deterioration in Buildings: A Practical Guide to the Science and Technology of Material Failure, 2nd ed., London, United Kingdom: Routledge, 2002.

[323] Motamedi, A., Soltani, M.M. and Hammad, A., "Localization of RFID-equipped assets during the operation phase of facilities," *Advanced Engineering Informatics,* vol. 27, no. 4, pp. 566-579, 2013.

[324] Lohani, B. and Singh, R., "Effect of data density, scan angle, and flying height on the accuracy of building extraction using LiDAR data," *Geocarto International,* vol. 32, no. 2, pp. 81-94, 2008.

[325] FLIR Systems, "FLIR PackBot," 2020. [Online]. Available: https://www.flir.ca/products/packbot/?utm_source=robots.ieee.org. [Accessed 22 Jan 2021].

[326] Zhou, Z.H. and Liu, X.Y., "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Transactions on knowledge and Data Engineering,* vol. 8, no. 1, pp. 63-77, 2015.

[327] Kukar, M. and Kononenko, I., "Cost-sensitive learning with neural networks," *ECAI,* vol. 15, no. 27, pp. 88-94, 1998.

[328] Griffiths, D. and Boehm, J., *Weighted point cloud augmentation for neural network training data class-imbalance,* arXiv preprint arXiv:1904.04094, 2019.

[329] Cui, Z., Henrickson, K., Ke, R. and Wang, Y., "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *IEEE Transactions on Intelligent Transportation Systems,* vol. 21, no. 11, pp. 4883-4894, 2019.

[330] Oza, P. and Patel, V.M., "One-class convolutional neural network," *IEEE Signal Processing Letters,* vol. 26, no. 2, pp. 277-281, 2018.

[331] Jadon, S., "A survey of loss functions for semantic segmentation," in *In proceedings of IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology*, Santiago, Chile, 2020.

[332] Hyeon, J., Lee, W., Kim, J.H. and Doh, N., "NormNet: Point-wise normal estimation network for three-dimensional point cloud data," *International Journal of Advanced Robotic Systems,* vol. 16, no. 4, pp. 1729881419857532/1-11, 2019.

[333] HDF-Group, "HDF5 User's Guide," 2018. [Online]. Available: https://portal.hdfgroup.org/display/HDF5/HDF5+User%27s+Guide. [Accessed May 2021].

[334] Liu, Q., Deng, M., Shi, Y. and Wang, J., "A density-based spatial clustering algorithm considering both spatial proximity and attribute similarity," *Computers & Geosciences,* vol. 46, no. 1, pp. 296-309, 2012.

[335] Ahmed, K.N. and Razak, T.A., "A comparative study of different density based spatial clustering algorithms," *International Journal of Computer Applications,* vol. 99, no. 8, pp. 18-25, 2014.

[336] Sidlauskas, D., Chester, S., Zacharatou, E.T. and Ailamaki, A., "Improving spatial data processing by clipping minimum bounding boxes," in *Proceedings of IEEE 34th International Conference on Data Engineering*, Paris, France, 2018.

[337] Barequet, G. and Har-Peled, S., "Efficiently approximating the minimum-volume bounding box of a point set in three dimensions," *Journal of Algorithms ,* vol. 38, no. 1, pp. 91-109, 2001.

[338] "NICP algorithm," Pyoints Documentation, [Online]. Available: https://laempy.github.io/pyoints/tutorials/icp.html#References. [Accessed May 2022].

[339] Serafin, J. and Grisetti, G., "NICP: Dense normal based point cloud registration," in *In proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Kyoto, Japan, 2015.

[340] Misol Monzo, M., "Dynamo Mesh Toolkit," May 2020. [Online]. Available: https://github.com/DynamoDS/Dynamo/wiki/Dynamo-Mesh-Toolkit. [Accessed June 2021].

[341] FARO Technologies Inc., "Faro Laser Scanner Focus 3D X 130 -NEO-Tech," 2010.

[342] Girardeau-Montaut, D., Roux, M., Marc, R. and Thibault, G., "Change detection on points cloud data acquired with a ground laser scanner," in *Proceedings of the International Archives on Photogrammetry, Remote Sensing and Spatial Information Sciences,*, Hannover, Germany, 2005.

[343] FARO Technologies Inc., "Quality Setting Function on the Focus3D," 2019. [Online]. Available:

https://knowledge.faro.com/Hardware/3D_Scanners/Focus/Quality_Setting_Function_on_the_Focus3D. [Accessed 5 JUN 2020].

[344] FARO Technologies Inc., "User Manual FARO Laser Scanner Focus 3D," 2011.

[345] Brownlee, J., "Overfitting and underfitting with machine learning algorithms," Machine Learning Mastery, 2016. [Online]. Available: https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/. [Accessed 2021 May 03].

[346] Wang, X., Girshick, R., Gupta, A. and He, K., "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Salt Lake City, UT, USA, 2018.

[347] Bolourian, N., Nasrollahi, M., Bahreini, F., and Hammad, A., "Point-based Concrete Surface Defect Semantic Segmentation," *Journal of Computing in Civil Engineering,* (submitted).

[348] Lopez Droguett, E., Tapia, J., Yanez, C. and Boroschek, R., "Semantic segmentation model for crack images from concrete bridges for mobile devices," *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability,* vol. 236, no. 4, pp. 570-583, 2020.

[349] Mohammed Abdelkader, E., Moselhi, O., Marzouk, M. and Zayed, T., "Entropy-based automated method for detection and assessment of spalling severities in reinforced concrete bridges," *Journal of Performance of Constructed Facilities,* vol. 35, no. 1, pp. 04020132/1-25, 2021.

[350] Thrun, S. and Leonard, J.J., "Simultaneous Localization and Mapping," in *Springer Handbook of Robotics*, Berlin, Heidelberg, Germany, Springer, 2008, pp. 871-889.

[351] Compton, M., Barnaghi, P., Bermudez, L., Garcia-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A. and Huang, V., "The SSN ontology of the W3C semantic sensor network incubator group," *Journal of Web Semantics,* vol. 17, pp. 25-32, 2012.

APPENDIX A PYTHON CODE FOR POINT CLOUD ALIGNMENT

NICP algorithm (adapted from [338]):

```
import numpy as np
from pyoints import (storage,
   Extent,
   transformation,
   filters,
   registration,
   normals,)
from pyntcloud import PyntCloud
import numpy as np
import pandas as pd
import os
import sys
import pdb

A = storage.loadPly('bridge-sample.ply')
print(A.shape)
print(A.dtype.descr)
B = storage.loadPly('Defect.ply')
print(B.shape)
print(B.dtype.descr)
output_dir = “./Alighned/”
if not os.path.exists(output_dir):
   os.makedirs(output_dir)

coords_dict = {
   'A': A.coords,
   'B': B.coords,}
n_th = np.sin(15 * np.pi / 180)
radii = [d_th, d_th, d_th, n_th, n_th, n_th]
nicp = registration.ICP(
   radii,
   max_iter=300,
   max_change_ratio=0.0003,
   update_normals=True,
   k=1)
T_dict, pairs_dict, report = nicp(coords_dict, normals_dict)
for key in coords_dict:
   coords = transformation.transform(coords_dict[key], T_dict[key])
PyntCloud.to_file(B, output_dir+'aligned_defect.ply')
```

APPENDIX B. DYNAMO SCRIPT USING MESHTOOLKIT

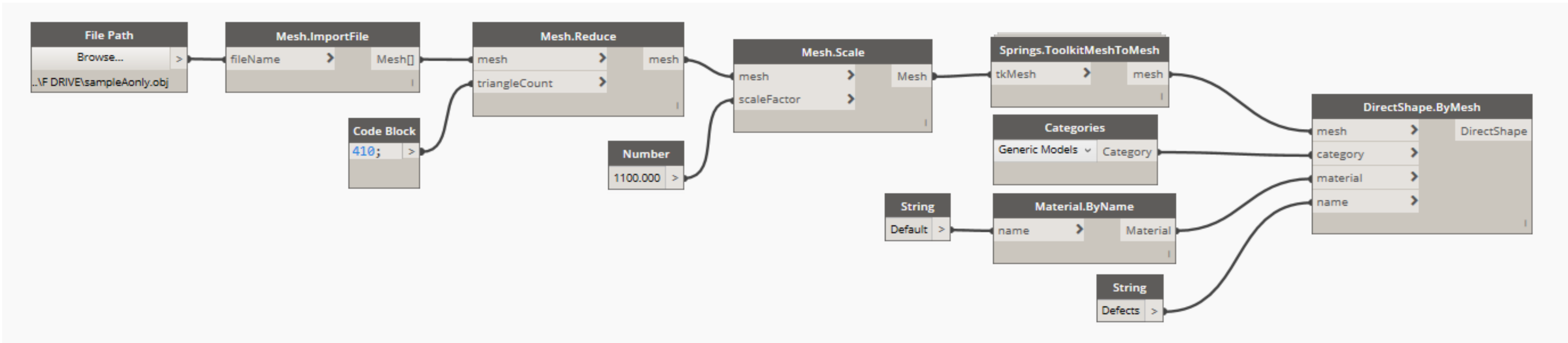


Figure B-1. Dynamo Script using Mesh Toolkit [340]